

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

INGENIERÍA DE TELECOMUNICACIÓN



Universidad de Alcalá

Trabajo Fin de Carrera

**Sistema de Detección y Reconocimiento
Facial de Conductores Mediante
Sistemas de Visión Computacional**

Sebastián Bronte Palacios
2008

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

INGENIERÍA DE TELECOMUNICACIÓN

Trabajo Fin de Carrera

**Sistema de Detección y Reconocimiento
Facial de Conductores Mediante
Sistemas de Visión Computacional**

Alumno: Sebastián Bronte Palacios

Director: Dr. D. Luis Miguel Bergasa Pascual

Tribunal:

Presidente: D. Miguel Ángel Sotelo Vázquez.

Vocal 1º: D. Manuel Ocaña Miguel.

Vocal 2º: D. Luis Miguel Bergasa Pascual.

CALIFICACIÓN:.....

FECHA:

A mis padres

“ El nueve no es sino un seis que se superó a sí mismo ”

Agradecimientos

En primer lugar tengo que agradecer haber llegado hasta aquí a mis padres y mi abuela. Sin su apoyo, sus ánimos, sus ideas, consejos y la orientación recibida cuando lo necesitaba esto no habría sido posible en absoluto, pues en los momentos de desánimo ellos siempre estuvieron allí para levantarme la moral y seguir hasta el final.

En segundo lugar agradecer a mi tutor Luis Miguel Bergasa el apoyo recibido, por sus consejos y sus lecciones, puesto que, además de tutor en este proyecto de fin de carrera, ha sido tutor y guía en los primeros años de carrera, y profesor en la asignatura de Control Neuronal y Borroso, la cual ha sido una de mis asignaturas favoritas de toda la carrera, puesto que combinaba dos aspectos fundamentales que buscaba cuando me matriculé en esta carrera: Control de sistemas e Inteligencia Artificial. Además también agradecer a otros profesores, que a lo largo de la carrera me han enseñado y valorado algo más que aspectos puramente académicos.

Como no, agradecer a mis compañeros del laboratorio donde he desarrollado el proyecto, Juanpa, Miguel, Balky, Álvaro, Jesús, Diego, Pablo, José Luis, Amaia, y todos los demás, por su colaboración, su ayuda, los buenos momentos de fiesta que hemos pasado juntos, y el hecho de haberles quitado parte de su tiempo para dejarse grabar y hacer las distintas pruebas que en este proyecto se presentan, además de las ideas dadas para realizar el proyecto.

También quiero dedicar unas palabras para mis amigos más cercanos: Fran y Cris, Adrián, Víctor, David, Murcia, Manzanares..., sin los cuales no habría tenido ratos de relax, e incluso de inspiración en este trabajo y por supuesto de diversión, pues todo no va a ser trabajo académico, sino que también el esparcimiento es necesario a la hora de estudiar una carrera.

Aquí he de agradecer a Alberto, Óscar, Luis, José, Marta, Juan Eulogio, Dani Blaquez, Esteban, Juan Antonio, etc..., amigos, compañeros de carrera y de prácticas, con los que también he pasado muy buenos ratos, y que también han tenido que aguantar mi mal humor en ocasiones.

También quisiera hacer una mención especial a Mazo, Santi y todos los demás compañeros de informática, mi carrera "adoptiva", a la que hubiese entrado si no me hubiesen aceptado en la que estoy terminando y con los que he compartido asignaturas de libre elección y también muy buenos momentos, de fiesta y prácticas muy llevaderas.

No quisiera olvidarme de la gente que conocí en mis cursos de verano de inglés en Londres y Dublín, en los cuales conocí a mucha gente, y algunos de ellos siguen siendo amigos. Mención especial a las tres "teachers" por su ayuda en la revisión del artículo para el WAF2008.

Por último, no quiero olvidarme de mis compañeros de la asignatura de Control Neuronal y Borroso, Francisco Huerta y Sara Robisco, con los pasé buenos momentos durante la asignatura y de los que aprendí mucho. Gracias Sara por mostrarme LyX y enseñarme el principio del camino de cómo se pueden hacer las cosas con L^AT_EX bien y de manera sencilla.

Sé que hay algunos a los que no he podido nombrar o se me ha podido olvidar. Por ello mis sinceras disculpas, pero sabed que no os olvido.

Índice general

I	Resumen	21
II	Memoria	27
1.	Introducción	29
1.1.	Motivación	30
1.2.	Objetivos	30
1.3.	Problemas asociados	31
2.	Estado del arte	33
2.1.	Sistemas antirrobo para automóviles	33
2.1.1.	Centralita inmovilizadora	35
2.1.2.	GPS	36
2.1.3.	GSM	36
2.2.	Sistemas de control de acceso	37
2.2.1.	Sistemas basados en reconocimiento facial	37
2.2.2.	Sistemas de control de arranque	40
2.2.3.	Sistemas implantados en aeropuertos, casinos y otros lugares	42
2.2.4.	Internet	43
2.3.	Reconocimiento con imágenes de Infrarrojos	45
3.	Técnicas empleadas	47
3.1.	Umbralizado de Otsu	47
3.2.	Tratamiento de la iluminación	48
3.2.1.	Introducción	48
3.2.2.	Ecualización de histograma uniforme	50
3.2.3.	Filtrado de mediana	52
3.2.4.	SSR (Single Scale Retinex)	53
3.3.	Operaciones morfológicas: dilatación y erosión	56
3.4.	Obtención de bordes	58
3.4.1.	Máscaras	58
3.4.2.	Canny	61
3.5.	Detección de caras: Viola & Jones	63
3.6.	Extracción de características	67
3.6.1.	PCA (Principal Component Analysis)	67
3.6.2.	2DPCA (2 Dimensional PCA)	70
3.7.	Algoritmo de clasificación: K-vecinos más cercanos	71
3.8.	Medidas de distancia	72
3.8.1.	Distancia euclídea	72
3.8.2.	Medida de distancia de Mahalanobis	73

3.9.	Métodos de fusión de medidas: sistemas borrosos	74
3.9.1.	Visión general	74
3.9.2.	Normalización y desnormalización	75
3.9.3.	Conjuntos borrosos	76
3.9.4.	Reglas y operadores	76
3.9.5.	Borrosificación y desborrosificación	77
3.9.6.	Tipos de sistemas borrosos	78
4.	Arquitectura del sistema de partida	79
4.1.	Introducción	79
4.2.	Detección de caras	80
4.3.	Tracking	81
4.4.	Extracción de características: 2DPCA	82
4.5.	Clasificador de vecino más cercano con criterio de mayoría	84
4.6.	Problemas asociados al sistema de partida	85
5.	Arquitectura del sistema propuesto	87
5.1.	Descripción general del sistema de reconocimiento	87
5.2.	Rechazo de candidatos externos	89
5.2.1.	Introducción	89
5.2.2.	Clasificador	91
5.3.	Tratamiento de la iluminación	95
5.3.1.	Introducción	95
5.3.2.	Ecuilibración uniforme y filtrado de mediana	98
5.3.3.	Recorte de cara a zonas de interés más invariantes	99
5.3.4.	Máscaras de Iluminación	101
5.3.5.	Retinex / SSR (Single Scale Retinex)	103
5.4.	Detección de gafas	104
5.4.1.	Introducción	104
5.4.2.	Método propuesto para la detección de gafas	105
5.5.	Reconocimiento diurno y nocturno	111
5.5.1.	Introducción	111
5.5.2.	Selector de noche / día	113
6.	Resultados experimentales	119
6.1.	Resultados del clasificador	120
6.1.1.	Tablas	121
6.1.2.	Gráficas	123
6.1.3.	Conclusiones Parciales	129
6.2.	Resultados de los tratamientos de la iluminación	130
6.2.1.	Secuencias de prueba en condiciones de iluminación controladas	130
6.2.1.1.	Tabla de resultados con el conjunto de secuencias de prueba	131
6.2.1.2.	Gráficas	132
6.2.2.	Vídeos de prueba en condiciones de iluminación exteriores	135
6.2.2.1.	Tabla de resultados del conjunto de secuencias de prueba	136
6.2.2.2.	Gráficas	139
6.2.3.	Tiempos de ejecución	141
6.2.4.	Conclusiones parciales	142
6.3.	Resultados del detector de gafas de sol	143
6.3.1.	Tablas	143

6.3.2.	Gráficas	144
6.3.3.	Tiempos de cómputo	147
6.3.4.	Conclusiones parciales	147
6.4.	Resultados de reconocimiento diurno y nocturno	148
6.4.1.	Tabla de resultados de reconocimiento diurno y nocturno	148
6.4.2.	Gráficas	151
6.4.3.	Tiempo de cómputo	155
6.4.4.	Conclusiones parciales	157
7.	Conclusiones y Trabajos futuros	159
7.1.	Conclusiones finales	159
7.2.	Trabajos futuros en iluminación	160
7.3.	Trabajos futuros en tracking	161
7.4.	Trabajos futuros en inmunización frente a intentos de engaño del sistema	161
7.5.	Trabajos futuros respecto de la apariencia de las imágenes de entrada	161
7.6.	Trabajos futuros en el clasificador	162
7.7.	Trabajos futuros en la base de datos	162
III	Pliego de condiciones	165
8.	Pliego de condiciones	167
8.1.	Condiciones generales	167
8.1.1.	Equipos	167
8.1.1.1.	Cámara	167
8.1.1.2.	Tarjeta de adquisición	168
8.1.1.3.	Cable de conexión	168
8.1.1.4.	Iluminador de infrarrojos	169
8.1.1.5.	PC	169
8.1.2.	Software	170
8.1.2.1.	S.O. Linux	170
8.1.2.2.	Generación de base de datos de caras	170
8.1.2.3.	Reconocimiento de usuarios y extracción de muestras	171
8.2.	Condiciones de ejecución	171
8.3.	Especificaciones técnicas	171
8.3.1.	Cámara de vídeo	171
8.3.2.	Diodos de infrarrojos	173
8.3.3.	Tarjeta capturadora (PCI)	173
8.3.4.	PC	174
IV	Presupuesto	175
9.	Presupuesto	177
9.1.	Cantidades necesarias, precios unitarios y presupuestos parciales	177
9.1.1.	Elementos Hardware	177
9.1.2.	Elementos de software comercial	178
9.1.3.	Ingeniería y montaje	179
9.2.	Presupuesto general	179

V	Apéndice	181
A.	Manual de puesta a punto e instalación	183
A.1.	Instalación de Linux	183
A.2.	Configuración de Linux	187
A.2.1.	Compiladores	187
A.2.2.	Firewire (IEEE 1394)	188
A.2.3.	Librerías	189
A.2.4.	Ffmpeg	189
A.2.5.	OpenCV	190
A.3.	Preparación de las aplicaciones	191
B.	Adquisición de imágenes desde cámara de vídeo	193
B.1.	Introducción	193
B.2.	Firewire (IEEE1394)	193
B.2.1.	Características	193
B.2.2.	Funciones de acceso y control en Linux	194
B.2.3.	Modos de adquisición	196
B.3.	Inclusión en el sistema	197
C.	Manual de uso	199
C.1.	Captador de muestras	199
C.2.	Base de datos faciales	200
C.2.1.	Procedimiento para la obtención de vídeos de entrenamiento	200
C.2.2.	Estructura de la base de datos faciales	201
C.2.3.	Programa de administración	203
C.2.4.	Añadir usuario	203
C.2.5.	Eliminar usuario	204
C.2.6.	Entrenar base de datos	204
C.2.7.	Eliminar información de entrenamiento	204
C.3.	Aplicación de reconocimiento	204
C.3.1.	Aplicación básica y variaciones con el clasificador	204
C.3.2.	Aplicación con captura desde cámara de vídeo	205
C.3.3.	Aplicaciones de tratamiento de iluminación	205
C.3.3.1.	Zona de interés reducida	205
C.3.3.2.	Ecualización	206
C.3.3.3.	Máscaras de iluminación	207
C.3.3.4.	Retinex	207
C.3.4.	Aplicación de detección de gafas	208
C.3.5.	Aplicación de reconocimiento nocturno y diurno	208
D.	Contenido del DVD	211
	Bibliografía	213

Índice de figuras

1.1. Sistemas de seguridad implantables en un coche utilizando técnicas de visión artificial	30
2.1. Sistemas antirrobo mecánicos que bloquean pedales y volante	34
2.2. Dispositivos antirrobo mecánicos que bloquean volante con cuadro de mandos o volante con asiento	34
2.3. Dispositivos antirrobo mecánicos que bloquean la palanca de cambios	35
2.4. Localizadores GPS para coche	36
2.5. Localizador híbrido GPS+GSM	36
2.6. Face Recognition System: FxAlert 2	37
2.7. Face Recognition Time-Attendance Management System	38
2.8. IC Face Recognition lock Certification	38
2.9. Portátiles Toshiba con tecnología de reconocimiento facial	39
2.10. Portátil Asus con tecnología de reconocimiento facial	39
2.11. Alcoholímetro con lector de retina	40
2.12. Sensor de alcohol en el aire situado en el asiento del conductor	41
2.13. Sensor de alcohol en el sudor situado en la palanca de cambios	41
2.14. Cámara e imágenes de la misma para la evaluación del estado del conductor	41
2.15. MyHeritage, reconocimiento facial por internet	44
2.16. Página principal de Polar Rose	44
2.17. Distintos tipos de Imágenes infrarrojas, por cámara térmica (a) y (b), y por reflexión con iluminador y cámara normal mas un filtrado de mediana para eliminar ruido (c) y (d)	45
2.18. Efecto de las imágenes infrarrojas térmicas sobre las gafas	46
3.1. Ejemplo de histograma de una imagen	50
3.2. Ejemplo de aplicación de ecualización de histograma a una imagen	52
3.3. Ejemplo de extracción de mediana	52
3.4. Ejemplo de aplicación de filtro de mediana a una imagen	53
3.5. Retinex aplicado a imágenes de prueba con iluminación variada	56
3.6. Aplicación de Retinex para la eliminación de niebla	56
3.7. SSR aplicado a imágenes médicas con distintos valores de varianza	57
3.8. Ejemplo de dilatación y erosión, a la derecha, la imagen original, en el centro, el resultado de una erosión, y a la derecha, el resultado de la dilatación	58
3.9. Máscaras paso bajo	59
3.10. Máscaras de Prewitt	60
3.11. Máscaras de Sobel	60
3.12. Obtención de imágenes de bordes	61
3.13. Imágenes de módulo y fase	61
3.14. Imágenes de bordes con distintos umbrales	62

3.15. Resultado de aplicar Canny a la imagen anterior	63
3.16. Ejemplos de variación de parámetros en el algoritmo de Canny	64
3.17. Función Haar unidimensional	64
3.18. Características principales que comprueba el detector de Viola & Jones	66
3.19. Primeras características seleccionadas por la cascada de detectores de Viola & Jones	67
3.20. Ejemplos de recuperación de las imágenes con 3 y con 1 autovector con PCA	70
3.21. Esquema básico de un sistema borroso	75
3.22. Funciones básicas borrosas	76
4.1. Esquema principal del sistema	79
4.2. Resultado de aplicar Viola & Jones en este sistema	80
4.3. Funcionamiento de la máquina de estados del sistema	82
4.4. Escalado implementado y corrección de coordenadas	83
4.5. Esquema implementado en el entrenador de la base de datos	84
4.6. Esquema del clasificador del sistema inicial	85
5.1. Esquema principal del sistema	87
5.2. Casos posibles en espacio de proyección	91
5.3. Funcionamiento básico del subsistema clasificador	93
5.4. Representación simplificada de los 2 primeros filtrados	94
5.5. Ejemplos de iluminación variable en un túnel	96
5.6. Ejemplos de iluminación al aire libre	97
5.7. Ejemplo de variación de iluminación en entornos controlados	98
5.8. Ejemplos de imágenes antes y después del tratamiento con ecualización de histograma y filtrado de mediana	99
5.9. Procedimiento para la selección de la zona de interés de los ojos y la nariz	100
5.10. Ejemplos de imágenes recortadas	101
5.11. Aplicación del método a las imágenes de la base de datos	102
5.12. Función de ajuste de niveles de gris de las plantillas	102
5.13. Plantillas de iluminación utilizadas	103
5.14. Imágenes combinadas con las máscaras de iluminación	103
5.15. Ejemplos de imágenes tratadas con Retinex	104
5.16. Primer caso posible de detección de elipses	105
5.17. Segundo caso de detección de contornos	106
5.18. Tercer caso de detección de contornos	106
5.19. Cuarto caso de detección de contornos	106
5.20. Quinto caso de detección de contornos	107
5.21. Sexto caso de detección de contornos	107
5.22. Séptimo caso de detección de contornos	107
5.23. Esquema de funcionamiento del algoritmo de detección de gafas	108
5.24. Imágenes preprocesadas para la obtención de bordes	110
5.25. Integración en el sistema del algoritmo de detección de gafas	111
5.26. Ejemplos de imágenes de entrada diurnas (a) y nocturnas (b)	112
5.27. Conjuntos borrosos pertenecientes a la media de nivel de gris de la imagen	114
5.28. Valores de salida posibles para la salida sugeno implementada	115
5.29. Esquema de implantación del detector en el sistema general	117
6.1. Muestras del conjunto de entrenamiento de la base de datos utilizada en condiciones de iluminación interior, sin tratamiento de la iluminación	121
6.2. Muestras de los candidatos en la secuencia de prueba de rechazo de candidatos	124

6.3.	Ejemplo de rechazo en la primera etapa del clasificador. Se representa el umbral en línea discontinua y distancia obtenida al centro del universo en línea continua.	125
6.4.	Distancias a los centros de las constelaciones del test, reconocimiento 1.	126
6.5.	Ejemplo de aceptación por parte de la primera etapa del clasificador. Se representa el umbral en línea discontinua y distancia obtenida al centro del universo en línea continua.	127
6.6.	Distancias a los centros de las constelaciones de la prueba	128
6.7.	Muestras tomadas para reconocimiento del vídeo de prueba 2	133
6.8.	Distancias mínimas a usuarios de la BB.DD. en la prueba 2	133
6.9.	Distancias mínimas medias a usuarios de la BB.DD. en la prueba	134
6.10.	Muestras del conjunto de entrenamiento de la base de datos utilizada en condiciones de iluminación exterior, sin tratamiento de la iluminación	136
6.11.	Ejemplos de caras contenidas en los distintos vídeos del coche	137
6.12.	Muestras tomadas para el reconocimiento 2, vídeo 6, cámara izquierda	139
6.13.	Distancias generadas por el reconocimiento con distintos tratamientos de la iluminación, Vídeo de coche 6 cámara 1 reconocimiento 2	140
6.14.	Medias de distancia mínima media a los usuarios, con los tratamientos de iluminación analizados, para el vídeo 6, cámara 1 y reconocimiento 2	140
6.15.	Secuencia 5 de prueba del detector de gafas	145
6.16.	Secuencia 6 de prueba del detector de gafas	146
6.17.	Secuencia 7 de prueba del detector de gafas	146
6.18.	Muestras del conjunto de entrenamiento de la base de datos utilizada en condiciones de iluminación interior, sin tratamiento de la iluminación	149
6.19.	Muestras del conjunto de entrenamiento de la base de datos utilizada en condiciones de iluminación interior, sin tratamiento de la iluminación	150
6.20.	medias de nivel de gris en las imágenes iniciales, tanto diurnas como nocturnas	152
6.21.	Muestras del primer y segundo reconocimientos en condiciones diurnas	152
6.22.	Experimento primero de reconocimiento en condiciones diurnas	153
6.23.	Distancias medias obtenidas en el experimento primero de reconocimiento en condiciones diurnas	153
6.24.	Experimento segundo de reconocimiento en condiciones diurna	154
6.25.	Distancias medias obtenidas en el experimento sexto de reconocimiento en condiciones diurnas	154
6.26.	Muestras del tercer reconocimiento en condiciones nocturnas	155
6.27.	Experimento tercero de reconocimiento en condiciones nocturnas	155
6.28.	Distancias medias obtenidas en el experimento tercero de reconocimiento en condiciones nocturnas	156
8.1.	Cámaras utilizadas	167
8.2.	Tarjetas adaptadoras para bus Firewire a distintos buses de PC o Mac, de la firma Unibrain	168
8.3.	Cable Firewire	169
8.4.	Características en monocromo de la cámara utilizada, Bastler Scout	172
8.5.	Características del modo de color de la cámara	173
B.1.	Esquema de implementación del driver IEEE1394 en las distintas capas del S.O.	195
C.1.	Ejemplos de adquisición de gestos variados para la base de datos	200
C.2.	Ejemplos de adquisición de posibles movimientos de la cara	200
C.3.	Estructura de directorios en la base de datos básica	201
C.4.	Estructura de directorios en la base de datos modificada	202

C.5. menú principal del programa gestor de la base de datos	203
C.6. Muestra de pantalla de la ejecución del programa base	205
C.7. muestra de pantalla del programa de cara recortada	206
C.8. Muestra de pantalla del programa de tratamiento con ecualización de histograma	206
C.9. Muestra de pantalla del programa de máscaras de iluminación	207
C.10.Muestra de programa con tratamiento del algoritmo Retinex	208
C.11.Muestra de pantalla del programa de detección de gafas	209
C.12.Muestra de pantalla del programa de detección diurna y nocturna con imágenes diurnas	209
C.13.Muestra de pantalla de imágenes nocturnas y diurnas con imágenes nocturnas . .	210

Índice de tablas

5.1. Tabla de reglas utilizadas	115
6.1. Umbrales utilizados en el clasificador utilizando distancia euclídea	122
6.2. Umbrales utilizados en el clasificador utilizando distancia de Mahalanobis	122
6.3. Porcentaje de imágenes rechazadas con la distancia euclídea	122
6.4. Porcentaje de imágenes rechazadas con distancia de Mahalanobis	123
6.6. Resultados obtenidos con distancia de Mahalanobis en los secuencias grabadas en la sala	131
6.8. Resultados obtenidos mediante distancia de Mahalanobis, vídeos coche	138
6.9. Tiempos de ejecución del sistema para distintos tratamientos de iluminación	141
6.10. Tiempos de proceso equivalentes por frame con distancia de Mahalanobis	141
6.11. Estadísticas del detector de gafas	143
6.12. Sobrecarga introducida en el sistema debida a la inclusión del subsistema de detección de gafas	147
6.13. Resultados del reconocimiento para imágenes diurnas y nocturnas con distancia euclídea	151
6.14. Tiempos de ejecución del reconocimiento para imágenes nocturnas y diurnas sin tratamiento de iluminación	156
8.1. Tabla de características de la cámara utilizada	172
9.2. Tabla de precios de elementos hardware	178
9.3. Tabla de precios de elementos de software comercial	179
9.4. Tabla de horas de ingeniería invertidas para el desarrollo del proyecto	179
9.5. Presupuesto final	180
B.1. Comparativa entre USB y Firewire	194

Parte I

Resumen

Resumen

En este proyecto se ha implementado un sistema de reconocimiento facial flexible para ser utilizado a bordo de un vehículo en cualquier momento del día, con el objetivo final de crear un sistema antirrobo. Este sistema utiliza una cámara montada en el interior del habitáculo, apuntando al asiento donde irá el conductor y una matriz de leds de infrarrojos para iluminar por la noche. El hardware utilizado en este sistema se basa en el que utilizan otros proyectos desarrollados en el grupo Robe-Safe orientados a la asistencia al conductor.

El sistema dispone de una base de datos de fotos de las caras de los conductores autorizados, en distintas posturas y gestos, tanto en condiciones de iluminación normales como en condiciones de iluminación nocturna. En la fase de funcionamiento se extraen mediante el algoritmo 2DPCA las características faciales de las imágenes del conductor, obtenidas mediante la cámara, y de las imágenes de los conductores autorizados, disponibles en la base de datos. Éstas se compararán para comprobar si el conductor está o no dentro de la base de datos y en caso afirmativo se comprueba cuál de todos los conductores autorizados es el candidato.

Se presentan los resultados obtenidos con el prototipo y los problemas que se presentan en condiciones reales.

Palabras clave: Reconocimiento facial, Tratamiento de la iluminación, imágenes faciales infrarrojas, 2DPCA, SSR.

Abstract

An on board facial recognition system is presented in this project. This system tries to be flexible against variable conditions because it will work outdoor, into a car, in order to build an antitheft device. This system uses an on board camera inside the car, in front of the driver, and an array of infrared leds to illuminate the scene at night. The hardware which is used in this system is based on another project's hardware, developed by Robe-Safe group, which is oriented to driver's assistance.

This system is based on a database of face's pictures of authorized drivers, in several gestures and positions, in standard illumination and night conditions. When this system is running, face database and driver's face characteristics are extracted with 2DPCA algorithm. This characteristics will be compared to check if driver is in the database and who of all is the driver in the database.

System prototipe's results and problems in real conditions will be presented.

Key words: Facial recognition, Illumination independence problem, facial infrared imagery, 2DPCA, SSR.

Parte II

Memoria

Capítulo 1

Introducción

El presente proyecto se enmarca dentro de la línea de investigación de los Sistemas de Transporte Inteligente (STI) desarrollada en el grupo Robe-Safe Departamento de Electrónica de la Universidad de Alcalá.

El objetivo de este grupo de investigación, es mejorar la seguridad, eficiencia y confort del transporte mejorando la funcionalidad de los coches y las carreteras usando las tecnologías de la información.

Para poder alcanzar los objetivos que se han formulado anteriormente será preciso definir módulos de percepción, control, procesamiento de datos, interfaces con el usuario, etc. Son muchos los sensores utilizados en los vehículos comerciales con la finalidad de obtener información del entorno: ultrasonidos, radar, láser, cámaras de vídeo, etc. La visión por computador, basada en el tratamiento digital de imágenes, se diferencia de los anteriores en la riqueza de información que proporciona, y además, permite interpretar la escena de trabajo en mayor medida que los otros sensores referidos anteriormente.

En un vehículo se pueden integrar múltiples sistemas para incrementar la seguridad del mismo, tanto seguridad pasiva como activa, aunque la mayoría se engloban en el primer caso. En el siguiente esquema se muestran algunos aspectos que podemos cubrir con técnicas de tratamiento de imagen, como pueden ser control de dirección de los faros, evitar colisiones, lectura e interpretación de paneles informativos y señales viales, detección del estado del conductor, etc (imagen extraída de [44]).

En nuestro caso, el sistema que se propone no es solo aplicable en el ámbito del automóvil, sino también a otros ámbitos, como el de la seguridad. Por ejemplo, es aplicable a la seguridad ciudadana, en la que se procura detectar a aquellos individuos que pueden ser peligrosos, o también seguridad en los casinos, en la que los estafadores pueden ser detectados evitando pérdidas en un casino.

En el mundo se está experimentando con este tipo de sistemas desde hace unos años. Las investigaciones iniciales para este tipo de sistemas se realizaron por universidades americanas, con fondos del Ministerio de Defensa (DOD). Estos sistemas se pensaron inicialmente para implementar un sistema automático de reconocimiento de sospechosos en las calles. Como veremos más tarde, se han hecho pruebas de este tipo en ciudades como en Los Ángeles o Londres, a través de los sistemas de cámaras en circuito cerrado (CCTV) que monitorizan calles, metros, edificios, etc. También estos sistemas se están probando en aeropuertos, en este caso con 2 objetivos: por

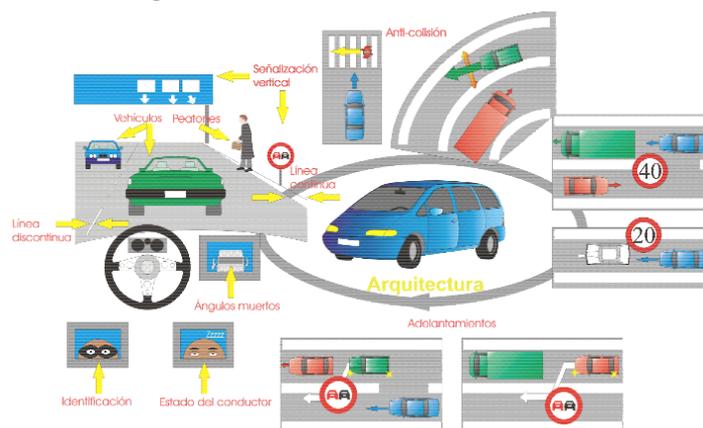


Figura 1.1: Sistemas de seguridad implantables en un coche utilizando técnicas de visión artificial

un lado, agilizar la verificación de identidad de múltiples personas para reducir el tiempo de espera en la cola de entrada al país correspondiente, y por otro lado detectar el paso de personas buscadas por los cuerpos de seguridad en los aeropuertos. Uno de los aeropuertos donde se están realizando este tipo de experiencias desde el año 2005 es el de Barajas, utilizando las cámaras de los arcos de seguridad en los puestos de control. Estos sistemas son controvertidos, pues grupos que piensan que se viola la privacidad de la gente que pasa por un determinado lugar. No obstante, en determinadas ocasiones, las cuestiones de seguridad deben prevalecer respecto a la privacidad, puesto que el fin es un beneficio más general que los inconvenientes individuales que pueden ocasionar.

1.1. Motivación

Las estadísticas de 2006 [12] revelan que en España ocurrió un robo de un coche cada 3 minutos, esto es, aproximadamente 175.000 coches en un año. El parque automovilístico de España superó en el mismo año los 25.75 millones de vehículos (asegurados). Por tanto, esta cifra apenas llega al 1% del total. Sin embargo la cifra en valor absoluto de robos de coche es tan elevada que nos obliga a reflexionar. El coste de este tipo de siniestros para las compañías de seguros, rondan los 80 millones de euros en indemnizaciones. Además, jamás se recuperan una parte de los vehículos, en torno a 50000, al ser coches de lujo y ser revendidos en el extranjero.

Se pueden ver las soluciones al problema de los robos de coche desde dos puntos de vista: por un lado, evitar que el vehículo sea robado, con sistemas de seguridad que comprueben que el usuario está autorizado a coger el coche; por otro lado, si han conseguido llevárselo, se ha de avisar de su localización para proceder a su recuperación.

Con el sistema propuesto se enfoca al primer punto de vista, puesto que el segundo punto ya está bastante desarrollado, al estar basado en las tecnologías de telefonía móvil o de localización por satélite, las cuales están muy implantadas y muy probadas para sus usos habituales.

1.2. Objetivos

Los objetivos perseguidos con el sistema que se presenta son, por un lado, rechazar usuarios externos a los permitidos, incluidos en la base de datos, y por otro lado, reconocer correctamente

al conductor de un automóvil con objeto de implementar un sistema antirrobo. Este sistema no dejaría arrancar al coche si el conductor no está autorizado, es decir, sus características faciales no se encuentran almacenadas en la base de datos de posibles conductores.

En el sistema que se presenta en este documento se integran varios subsistemas. En primer lugar es necesario detectar la presencia de caras en la imagen [39]. A continuación, hay que hacer un seguimiento de la misma para no perderla durante el reconocimiento y tenerla centrada. Cuando se tiene una imagen estable de la cara, se evalúa la región de interés para realizar el reconocimiento. Posteriormente se realiza la comparación con las caras que están en una base de datos, siendo éstas realizadas calculando las distancias entre características de las imágenes de la base de datos y las que llegan desde el dispositivo de entrada al sistema. Por último, se evalúa cuál es el conductor que más se parece dentro de la base de datos. Este sistema, aparentemente ideal, presenta una serie de problemas que se presentarán más adelante.

Además de realizar un reconocimiento correcto en condiciones de iluminación diurnas, se procederá a realizar el reconocimiento con imágenes nocturnas, iluminando al sujeto con un iluminador de infrarrojos ya desarrollado. Para ello, tendremos que tener 2 bases de datos adecuadas a cada tipo de iluminación y además se ha de poder detectar el hecho de encontrar un entorno de iluminación nocturna o diurna para poder seleccionar la base de datos adecuada.

1.3. Problemas asociados

Introducir un reconocedor a bordo de un coche supone unos nuevos retos que hay que estudiar. Los sistemas básicos para el reconocimiento de caras funcionan bien en entornos controlados. Algunos de los retos que se presentan son el rechazo de usuarios no permitidos, independizar el reconocimiento de la iluminación, poder realizar el reconocimiento tanto de día como de noche y la detección de gafas de sol para que el usuario se las quite en el caso que quiera arrancar el coche. Estos problemas se analizarán con más detalle posteriormente.

Los sistemas de reconocimiento facial se basan fundamentalmente en 2 técnicas: Biometría [40] y PCA/2DPCA (Principal Component Analysis /2 Dimensional PCA) [27, 42].

La biometría se basa en calcular distancias entre partes significativas de la cara, como pueden ser la distancia entre ojos, dimensiones de la nariz, tamaño de la boca, etc. Este cálculo de distancias puede fallar si las cámaras se descalibran o si están mal enfocadas o si el objetivo se mueve mucho obteniendo unas imágenes borrosas no aptas para un buen cálculo de distancias. Otras posibles variaciones que nos podemos encontrar a la hora de medir distancias en una cara son los gestos, con los cuales podemos modificar sustancialmente las dimensiones de la boca y de los ojos. Si queremos unas buenas medidas de distancia entre puntos característicos de la cara, no se puede confiar únicamente en una proyección 2D de la cara. Una buena opción es realizar un modelado en 3D a partir de distintas proyecciones de la cara en distintas cámaras o con otros métodos. Esta técnica es más cara de implementar, puesto que necesitamos planos láser con barrido en la otra dimensión, un escáner 3D, o un sistema de visión binocular con la correspondiente calibración y la realización de las correspondencias entre las imágenes de las 2 cámaras. Además, independientemente de la técnica utilizada (visión o láser), es más costosa computacionalmente porque hay mucho volumen de información a procesar.

Respecto a los sistemas basados en PCA, tienen el inconveniente de que se basan en la apariencia de las imágenes para proceder a la comparación, esto es, que influye en gran medida el gesto que tenga la cara de la persona en el instante que se proceda al reconocimiento. Este problema se soluciona enriqueciendo la base de datos con suficientes gestos de una misma persona.

En estas dos técnicas de reconocimiento facial aparece el problema de la iluminación de la escena, la cual hace que varíen los colores de ciertas zonas de la cara dependiendo de dónde esté el foco, y esta es una alinealidad que no es fácilmente eliminable. De hecho el problema de la iluminación, es un problema no resuelto todavía debido a la gran cantidad de alinealidades que introducen las variaciones de iluminación en una superficie. En los algoritmos basados en apariencia, la iluminación puede cambiarla drásticamente, con lo que podemos equivocarnos con facilidad a la hora de realizar el reconocimiento.

El caso de la iluminación es un reto importante, puesto que para realizar un reconocimiento correcto, ésta ha de ser uniforme o debe seguir un cierto patrón que ha de estar incluido en la base de datos. En condiciones reales esta condición va a ser difícil de cumplir.

Respecto a los problemas mencionados anteriormente en el clasificador para el rechazo de usuarios, si tomamos exclusivamente un criterio de distancia mínima, no se es capaz de decidir si el usuario pertenece o no a la base de datos. El sistema lo que hace en realidad es buscar el más parecido dentro de la misma, aunque el parecido sea muy bajo, lo cual no es aceptable, por lo que se plantea un subsistema para evitar este inconveniente.

De día, un usuario cualquiera, puede llevar puestas gafas de sol. Este hecho para nosotros es un inconveniente, puesto que las gafas de sol alteran sustancialmente la apariencia de un usuario y el sistema podría fallar. Por tanto, se ha desarrollado un sistema de detección de gafas de sol para hacer que el usuario se las quite antes de efectuar un reconocimiento válido.

Una vez se hayan conseguido estas correcciones, se reconocerá la cara con otros tipos de imágenes, como las infrarrojas, útiles en ausencia de luz captadas con cámaras específicas, atendiendo a lo sugerido en [8,37].

En caso de que el sistema siguiese en desarrollo, se hará que éste sea capaz de reconocer al usuario en distintas condiciones, como por ejemplo, si el usuario se ha dejado crecer barba o se ha cortado el pelo u otras no linealidades en la imagen captada que fuesen detectables, controlables y salvables.

En un futuro, este sistema, además de servir de antirrobo, como se pretende en este proyecto, se podría utilizar de "caja negra" pudiendo controlar lo que ocurre en el interior del vehículo, reconocer a los acompañantes del conductor, monitorizar su estado, etc.

Capítulo 2

Estado del arte

En esta sección hablaremos en primer lugar de los distintos sistemas comerciales que podemos encontrar en la actualidad que realizan algún tipo de reconocimiento de usuarios, exceptuando los lectores de huella digital. A continuación se expondrán las tecnologías utilizadas actualmente para resolver algunos de los problemas que se han podido comprobar en este sistema. Posteriormente nos centraremos en dos problemas fundamentales, que son, el reconocimiento en condiciones no controladas de iluminación, y el reconocimiento en condiciones de iluminación muy reducida o nula. La primera situación la encontraremos si nos encontramos en el interior del coche, durante el día al aire libre, mientras que la segunda la encontraremos de noche, o de día en un aparcamiento o en un garaje en los que la iluminación es bastante reducida en comparación con la que obtenemos al aire libre.

Actualmente podemos encontrar múltiples sistemas de seguridad que se basan en reconocimiento facial, de retina o iris, de huella dactilar, etc. Los que nos van a interesar en nuestro proyecto son los relacionados con la cara, con lo que nos centraremos en los sistemas de reconocimiento facial, aunque daremos un breve repaso a los demás. Existen sistemas que utilizan ciertas entidades para proveer seguridad, y otros que se comercializan para que cualquiera que lo necesite, lo pueda comprar y satisfaga sus necesidades de seguridad.

Estos dispositivos suelen estar basados en biometría en la cual, como se ha avanzado en la introducción, se realizan mediciones sobre la imagen para poder identificar a partir de esas medidas, quién de todos los usuarios de la base de datos es el que más se parece, o si el sujeto está o no incluido.

2.1. Sistemas antirrobo para automóviles

Hace algunos años, la industria automovilística no se preocupaba demasiado en los aspectos de seguridad para evitar que el coche fuera robado. Ejemplos de estos son los coches de los años anteriores a los 90. En la mayoría de estos coches, es fácil realizar un puente para poder llevarse el coche sin ningún problema. A partir de entonces la industria se empezó a preocupar por este problema dada la facilidad con la que se podían llevar a cabo los robos de sus vehículos.

En la actualidad, para evitar o por lo menos disuadir a los ladrones no expertos, existen varios métodos:

1. Unos están basados en bloquear el coche mediante un dispositivo mecánico

Dentro de estos últimos, existen cuatro modelos que pasamos a describir brevemente:

- a) Dispositivos que unen y bloquean los pedales con el volante.
- Este tipo de dispositivos, eran comunes hace unos años, aunque en la actualidad no lo son tanto, puesto que han sido desbancados por otros más sencillos. Con éste, conseguimos inmovilizar los actuadores más importantes del coche y por tanto el movimiento del mismo si intentan llevárselo.

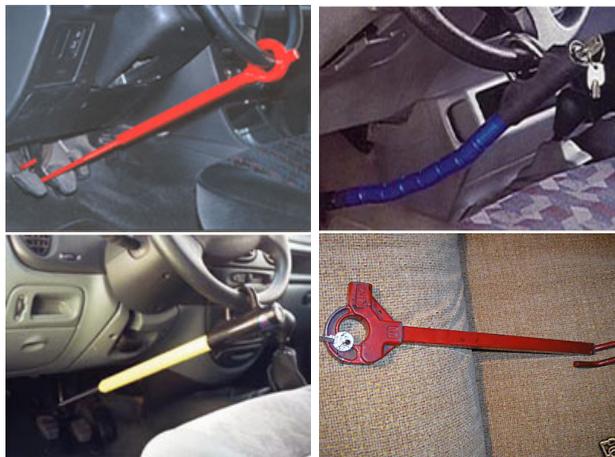


Figura 2.1: Sistemas antirrobo mecánicos que bloquean pedales y volante

- b) Los sistemas que bloquean el giro del volante mediante una barra que hace de tope con el panel de mandos, el asiento o las piernas del conductor.
- Éste es el más demandado en la actualidad, al ser sencillo y rápido de colocar.



Figura 2.2: Dispositivos antirrobo mecánicos que bloquean volante con cuadro de mandos o volante con asiento

- c) Otros sistemas unen y bloquean la palanca de cambios con el freno de mano.
- A diferencia del primero, dejamos los pedales libres, pero no pueden mover el coche cuando lo arranquen a menos que inutilicen los frenos manualmente o corten la cadena que une el volante con el freno de mano.



Figura 2.3: Dispositivos antirrobo mecánicos que bloquean la palanca de cambios

a) Por último, se tienen sistemas que bloquean cualquier movimiento de los pedales.

Como el primero de los dispositivos, pero sin unirlos al volante. En caso de hacer un puente y arrancar el coche, no podemos actuar sobre él.

Todos estos dispositivos son por lo general económicos, pero tampoco suponen un gran obstáculo para los amigos de lo ajeno si estos son expertos. Si se tiene suerte y damos con un ladrón no experto, este tipo de sistemas disuadirán a un ladrón e irán a por otro coche que no tenga esta protección.

2. Otros están basados en dispositivos electrónicos que de alguna forma evitan el arranque. Este tipo es en el que nos centraremos posteriormente.

Por último, si el coche consigue ser robado, tenemos también varios métodos de alerta para localizar el coche:

1. El primero y más básico es una simple sirena que avisa de que el coche está en movimiento, ha sido golpeado o levantado, estando cerrado.
2. Uno de los métodos basados ya en localización es utilizando la red GSM, el cual explicaremos con algo más de detalle a continuación.
3. Por último, el segundo método más utilizado está basado en la utilización de GPS para poder localizar el vehículo una vez sustraído.

2.1.1. Centralita inmovilizadora

La mayoría de los coches, por no decir todos los modelos actuales tienen este sistema de protección de serie. El coche se desbloquea a partir de un código numérico introducido por un teclado, una llave con un código insertado, o un sistema de desbloqueo similar. Este sistema es difícil de romper por los ladrones poco avezados, puesto que la centralita inhibe la capacidad de hacer un puente al coche al modo clásico, impidiendo que el coche arranque aún haciendo el cortocircuito con los 2 cables correspondientes. Sin embargo, este sistema puede ser vulnerado con una centralita maestra, que proporciona los códigos correctos al coche, y por tanto poder arrancar sin ningún problema. Para vehículos de gama baja o media, este dispositivo es útil a la hora de disuadir, puesto que si quieren llevárselo no van a poder hacerlo de manera sencilla, y las mafias de robo de coches, a la hora de conseguir una centralita maestra se centrarán en las de coches más caros, por los cuales obtendrán más beneficio.

2.1.2. GPS

Este sistema se basa en la localización por satélite del vehículo, que llevará instalado en un sitio no visible por los ladrones una baliza para que, en caso de que se detecte que el vehículo ha sido sustraído, enviar continuamente la posición en la que se encuentra, para poder ser localizado por los cuerpos de seguridad.

Los modelos más completos ofrecen opciones como el corte del encendido, sensores de choque y anti-levantamiento o inmovilización del coche.

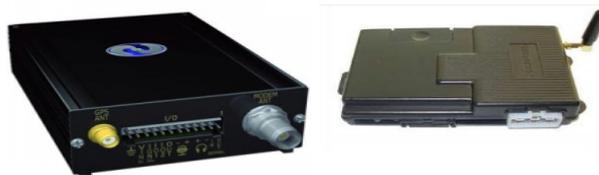


Figura 2.4: Localizadores GPS para coche

2.1.3. GSM

En estos casos, el medio utilizado para la detección del movimiento y transmisión de la información, es la red de telefonía móvil GSM. En este caso, en lugar de utilizar la simple información de encaminamiento a través de redes, lo que se hace es utilizar la información de posición, la cual se obtiene a través de trilateración. Sabiendo el retardo de la señal del transmisor GSM desde que sale del transmisor hasta que llega a varios receptores, podemos calcular la posición de una forma precisa, puesto que se sabe exactamente dónde se localizan las antenas receptoras.



Figura 2.5: Localizador híbrido GPS+GSM

Como en el caso anterior, el sistema integrado en el vehículo es capaz de saber si éste ha sido robado o no, puesto que se ha puesto en movimiento estando supuestamente cerrado o puede reconocer una situación de forzado.

En este caso, el usuario es avisado desde la centralita directamente a su teléfono móvil, bien mediante llamada o mensaje de texto, indicando el tipo de alarma producida y el estado del

vehículo. En algunos modelos más sofisticados se incluye la velocidad del vehículo y la calle y número de su posición.

2.2. Sistemas de control de acceso

2.2.1. Sistemas basados en reconocimiento facial

A través de internet, en algunas páginas dedicadas a la venta de dispositivos de seguridad para edificios, apartamentos, etc (<http://www.alibaba.com>), se pueden encontrar distintos dispositivos que realizan algún tipo de identificación facial. Algunos ejemplos de estos sistemas se pasan a ilustrar a continuación.

De la empresa Hoya electronics CO LTD (INDIA) tenemos un dispositivo con las siguientes características técnicas:

- Hoya electronics CO LTD (INDIA): Face Recognition System. FxAlert 2.



Figura 2.6: Face Recognition System: FxAlert 2

- 4-16 canales de vídeo de hasta 25 fps
- Algoritmo rápido y exacto
- Mecanismo innovador y auto-adaptativo a los entornos dinámicos
- Rápido post-proceso de búsqueda y detección.
- Detección automatizada de movimiento o eventos

- FIRS Intelligent Technology (Shenzhen) Co., Ltd. (CHINA): Face Recognition Time-Attendance Management System.



Figura 2.7: Face Recognition Time-Attendance Management System

Porcentaje de aciertos: >99 %, si FAR <0.01 %

Requerimientos de iluminación: cualquier condición

Tasa de falso rechazo: <0.1 %

Velocidad de reconocimiento: <1 seg, si 1:1; <2 segs, si 1:todos

Requerimientos del sistema: Basado en PC o embebido

Grupos de usuarios / Gestión departamental

Modos de operación: solo con la tarjeta, solo reconocimiento facial, tarjeta y reconocimiento facial o contraseña y reconocimiento facial.

Servidor de base de datos interna, plantillas y registro de eventos, generación de informes y capacidad de exportar datos a excel mediante xls o en modo texto.

Acceso a través de navegador web (disponible para el sistema embebido)

- ICEC Technology Limited (CHINA), IC Face Recognition lock Certification: CE,FCC



Figura 2.8: IC Face Recognition lock Certification

Estructura de fácil instalación y alta compatibilidad con otros sistemas de control de acceso y monitorización.

Interfaz flexible para posibilidad de conexión de distintos periféricos: lector de tarjetas RFID, mini-teclado, monitor, etc.

Control de acceso local y combinado con tarjeta o código.

Acceso manual e inteligente a las funciones de control Monitor en tiempo real a través de la red Basado en Linux

Modos de gestión online, medio-online y offline Bajo consumo Capacidad del sistema: 100 en modo offline, 1500 en modo medio-online y 6000 en modo online.

- Portátiles

En este año se están empezando a comercializar portátiles con esta tecnología incluida de serie para hacerlos más innovadores. Hay que decir que esta tecnología no es difícil de instalar y que ya lleva algún tiempo funcionando debido a que la mayoría de portátiles actuales llevan incorporada una cámara web, la cual puede ser usada como lector biométrico.

- Toshiba Satellite A305 y U405.

Estos 2 portátiles son los primeros que esta marca ha lanzado al mercado con esta tecnología de reconocimiento facial en 2008. Se ha impulsado esta tecnología para salir de los estados de hibernación o suspensión de manera más rápida que introduciendo la contraseña, y por supuesto más fácil pues lo único que tenemos que hacer es estar en frente del monitor.



Figura 2.9: Portátiles Toshiba con tecnología de reconocimiento facial

- Asus F8Sn

En este caso, esta marca ha preferido lanzar esta gama de portátiles, en principio, solo en corea. El objetivo por el que se ha implantado esta tecnología en este modelo es el mismo que en Toshiba.



Figura 2.10: Portátil Asus con tecnología de reconocimiento facial

2.2.2. Sistemas de control de arranque

A continuación mostramos algunos ejemplos de otros dispositivos que podemos encontrar en el mercado que utilizan técnicas de reconocimiento basadas en biometría de esta zona particular de la cara.

- Alcoholímetro con lector de retina

Se ha encontrado en Internet un sistema de inmovilización para coches compuesto de un alcoholímetro combinado con un lector de iris. Este sistema lo ha desarrollado la empresa española ATURVITE. La parte de inmovilizado del coche ligada a los usuarios consiste en una base de datos que almacena hasta 10 individuos, lo cual está dentro de las especificaciones técnicas de este proyecto.



Figura 2.11: Alcoholímetro con lector de retina

En el folleto de publicidad de este producto, vemos cómo se ven fotos de autobuses y camiones. En el caso de los camiones puede ser suficiente con tener un máximo de 10 usuarios, pero en el caso de los autobuses, sobre todo los urbanos, solo 10 posibles conductores podría no ser suficiente, puesto que un mismo autobús puede ser conducido por un número mayor de conductores. Para coches de empresa o camiones, en el que el número de conductores posibles es más reducido, es una aplicación interesante.

- Nissan: Concept car to showcase anti drunk driving technology

Esta firma japonesa está apostando por implementar tecnología para aumentar la seguridad en sus automóviles con su sistema de desarrollo ITS (Intelligent Transport Systems). Proyectos desarrollados por esta división fueron publicados en agosto de 2007 en la página web [32]. En uno de los prototipos se implementó un sistema de control de arranque utilizando varios tipos de sensores para detectar si el conductor ha bebido alcohol.

- El primero de los sensores utilizados es el de alcohol en el aire. Este sensor está situado en asiento del conductor, lo más cerca posible de los hombros, para poder captar el aliento del conductor y poder evaluar si su espiración contiene alcohol.



Figura 2.12: Sensor de alcohol en el aire situado en el asiento del conductor

- Otro de los sensores utilizados es el que detecta alcohol en el sudor. Este sensor se encuentra en la palanca de cambios, y es capaz de detectar si hay presencia de alcohol en el sudor de la palma de la mano, puesto que para cambiar de marcha en el coche, se necesitará apoyar la mano en la palanca. Si se detecta la presencia del alcohol, se encenderá una alarma en el coche y no se permitirá su arranque.



Figura 2.13: Sensor de alcohol en el sudor situado en la palanca de cambios

- Por último, para más seguridad, se utiliza una cámara para evaluar el estado del conductor. Esta cámara se ha montado en el cuadro de mandos del conductor para poder captar su cara y sus gestos. Este subsistema se encarga de evaluar el nivel de cansancio a partir de la frecuencia de parpadeo del conductor. Si se ha detectado que el conductor se está durmiendo, lo que se ha optado por hacer es apretar el cinturón más fuerte y rápido, para llamar la atención del conductor lo más rápido posible.

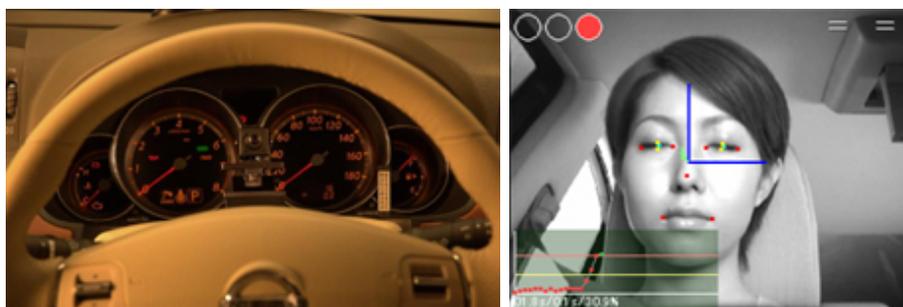


Figura 2.14: Cámara e imágenes de la misma para la evaluación del estado del conductor

2.2.3. Sistemas implantados en aeropuertos, casinos y otros lugares

Como diferencia fundamental con los anteriores, vemos que estos sistemas tienen que utilizar bases de datos de rasgos o caras más grandes, pues se ha de reconocer a una persona entre muchas al ser lugares con más flujo de personas, y por tanto mayor número de posibles candidatos a reconocer satisfactoriamente.

Estos sistemas se basan en el establecimiento previo de una base de datos con una serie de personas que se quieren detectar por una razón determinada, normalmente de acceso. En las imágenes de entrada de la cámara se buscan usuarios y se compararán con las imágenes de caras de la base de datos. A continuación pasamos a describir ciertas experiencias realizadas.

En el aeropuerto de Barajas se vienen probando desde 2005 sistemas de reconocimiento facial desarrollados por profesores de la universidad Carlos III. Los algoritmos utilizados para el reconocimiento son de apariencia (PCA, 2DPCA, 2DLDA...) aunque en uno de sus artículos digan que se utilizan algoritmos basados en biometría [9].

Respecto a Estados Unidos, este tipo de sistemas se están empezando a implantar en aeropuertos y casinos. En el caso de los casinos, se prohíbe la entrada a las personas que se sabe que hacen algún tipo de trampa. Para este tipo de locales, en este país, existen ejemplos de software que realiza reconocimiento facial:

- BioBouncer

Este sistema utiliza el reconocimiento facial para reconocer a los individuos dentro de un local lleno de gente, como por ejemplo en los casinos más populares.

Los casinos lo usan para identificar, a ciertas personas previamente introducidas en una base de datos, y con esto saber de su presencia para poder obrar en consecuencia. Este sistema de reconocimiento lo podemos encontrar también en algunos bares y clubes de Estados Unidos.

El coste inicial de este sistema es de 7.500 dólares por el hardware, configuración y software. Además se ha de pagar 6.000 dólares al año por mantenimiento, lo cual hace que sólo ciertos sitios puedan permitírselo. Respecto de la seguridad, se utilizan estos sistemas para atrapar a las personas que tienen múltiples carnets de conducir según el estado.

- Vilsage

Utilizado en casinos, este software realiza las comparaciones con 128 caras que se consideran patrón. Los resultados de la comparación se muestran al supervisor en orden descendente, para que él sea el que tenga la última palabra en la decisión.

- Visionics

Software que para realizar el reconocimiento utiliza rasgos localizados del rostro, esto es, se detectan puntos característicos de la cara, y se comparan con otros que existan en la base de datos predeterminadas para localizar el mapeado de puntos que más se ajusta al sujeto a analizar.

También se ha probado en las calles con no demasiado éxito al ser detenidas varias personas inocentes, siendo la tasa de reconocimiento de un 55 % aproximadamente. Sin embargo, con otros programas utilizados más recientemente, se han obtenido resultados positivos al probarlo en la final de la Super Bowl, al conseguir arrestar a personas que tenían antecedentes penales y que no habían sido reconocidos previamente por el personal.

En Reino Unido están haciendo lo propio con su red CCTV, en sus aeropuertos e incluso la cadena de supermercados Budgens. Estos sistemas se quieren utilizar para agilizar la identificación de pasajeros en los arcos de seguridad, para evitar la libre circulación de personas buscadas por motivos de seguridad. Estos sistemas primero se implantarán en aeropuertos pequeños para luego implantarlos en aeropuertos mayores en el caso de que las pruebas den resultados positivos. Se generarán falsos negativos para que los pasajeros dudosos o inocentes se deriven a filas convencionales para asegurar una correcta identificación y por lo menos agilizar en gran parte el tránsito de personas. Respecto al caso concreto de Budgens estos sistemas se utilizan para detectar menores de edad que intentan comprar alcohol, para facilitar su reconocimiento y alertar a tiempo a los cajeros.

En el casino Esplanade, en Alemania, se está empezando a implantar este tipo de programas de reconocimiento facial por biometría. Un ejemplo de este tipo de programa es FaceSnap / FaceCheck, desarrollado por Cross Match.

Este programa se aproxima al BioBouncer en funcionamiento, con la diferencia de que no sabemos sus costes de implantación ni la inversión inicial necesaria.

En China, en la ciudad de Shenzhen, con 12,4 millones de habitantes, se van a instalar unas 20.000 cámaras de seguridad, asociadas a software de reconocimiento facial para la detección de delincuentes y además para el control de personas y su ficha de natalidad, para que, en el caso de que se detecte alguna persona que haya infringido esta ley, detenerla y hacerla pagar la multa.

Por último, como curiosidad, también se están implantando este tipo de sistemas en los aeropuertos, puertos, y puestos fronterizos de Uruguay. En este caso se quieren utilizar para luchar contra el narcotráfico y atrapar a sospechosos habituales relacionado con este tipo de delitos.

2.2.4. Internet

En internet existen algunas aplicaciones de reconocimiento de caras, aunque estas no sean estrictamente de control de acceso.

Una de estas es MyHeritage, en la que, además de obtener información sobre nuestro árbol genealógico, una de sus aplicaciones es poder comparar cualquier foto que subamos con alguna de las 3200 que se encuentran en su base de datos. Estas imágenes son fotos de gente famosa. Un ejemplo de pantalla se encuentra en la figura 2.15.

Para reconocer caras en internet, existe otro proyecto con el que se puede reconocer caras existentes en imágenes publicadas en internet. Polar Rose es un proyecto que utiliza las bases de datos de imágenes de otros buscadores o portales de internet y localiza las caras y las compara con las que contiene en su base de datos. Se pueden etiquetar algunas caras manualmente si estas no se encuentran en la base. Por el momento, la base de datos supera los 200000 nombres de caras que se pueden buscar. También se puede hacer el proceso inverso, esto es, subir una foto y que el buscador devuelva el nombre de la persona si ésta se encuentra en su base de datos. Para entrar en ciertas funciones de la página hay que estar registrado. Se muestra la página principal de esta web en la figura 2.16.

Dejando a un lado el reconocimiento de caras, últimamente se está poniendo de moda sistemas de reconocimiento de objetos a partir de una palabra clave, estaríamos ante el ejemplo de buscar imágenes en google, en el cual ya busca por el contenido de la imagen y no solo por el nombre del archivo de la imagen, aunque el nivel de abstracción actual no es demasiado alto comparado con el que tienen los seres humanos. Hay más buscadores de imágenes (pretendidos por el gigante

anterior, pero que no terminó su adquisición) como Riya, al que se le introduce una palabra clave y él devuelve los posibles resultados dentro de lo que él conoce. Existen también otros buscadores a los que se introduce una imagen y el buscador se dedica a encontrar el mismo objeto o parecido en las distintas imágenes en su base de datos.



Figura 2.15: MyHeritage, reconocimiento facial por internet

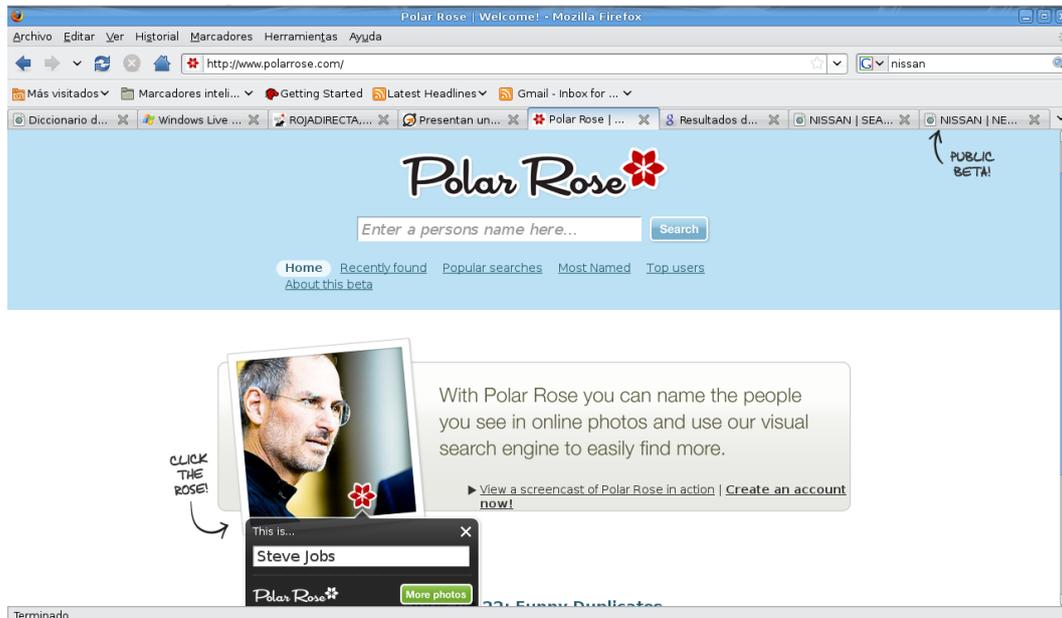


Figura 2.16: Página principal de Polar Rose

2.3. Reconocimiento con imágenes de Infrarrojos

A la hora de realizar un reconocimiento facial, existen varias situaciones en cuanto a las condiciones de iluminación. En ausencia de la misma, se ha de utilizar radiación infrarroja para iluminar la escena. En cuanto a la adquisición de imágenes infrarrojas, existen dos opciones: imágenes infrarrojas térmicas, en las que es el cuerpo el que emite la radiación, o imágenes por reflexión de luz infrarroja.

En estos 2 tipos de imágenes la diferencia está en la longitud de onda que se captura en los sensores. Existen 3 bandas dentro de este tipo de luz.

- Infrarrojo cercano ($0,78 - 1,1\mu m$)
- Infrarrojo medio ($1,1 - 15\mu m$)
- Infrarrojo lejano ($15 - 100\mu m$)

El infrarrojo cercano es el correspondiente a las imágenes obtenidas por reflexión, mientras que las imágenes térmicas están localizadas en el infrarrojo medio y lejano.

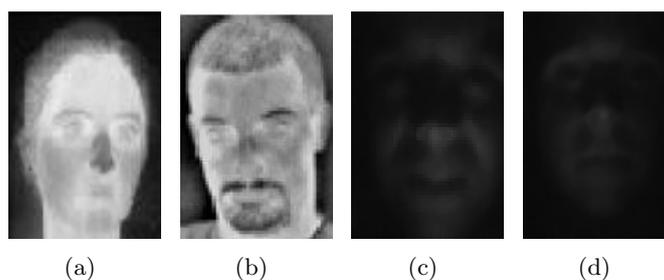


Figura 2.17: Distintos tipos de Imágenes infrarrojas, por cámara térmica (a) y (b), y por reflexión con iluminador y cámara normal mas un filtrado de mediana para eliminar ruido (c) y (d)

Antes de continuar, apuntar que todos los fotodiodos que se integran en las cámaras tienen sensibilidad a la radiación infrarroja, algunos más que otros, dependiendo del diseño y de la aplicación para la que estén diseñados. En las cámaras normales, que no se quiere que tengan sensibilidad a este tipo de radiación, se incluye un filtro para no dejar pasar estas frecuencias, en un intento de imitar la curva de sensibilidad del ojo humano.

Respecto de la figura 2.17, tenemos que destacar varias cosas. En primer lugar, las dos primeras fotos, que han sido extraídas de los papers [23, 38], se utilizan imágenes infrarrojas para proceder con este tipo de detección, mientras que las otras 2 imágenes se han tomado para las pruebas iniciales de nuestro sistema. Estas últimas 2 imágenes han sido previamente tratadas, puesto que en la cámara utilizada inicialmente se introducía mucho ruido al trabajar en condiciones de muy baja iluminación, por tanto ha habido que tratar la imagen previamente con un filtrado de mediana de 5×5 .

Las imágenes infrarrojas térmicas tienen una particularidad respecto a las gafas, ya sean de sol o no, es que siempre aparecen negras, como podemos apreciar en el artículo [37]. La razón de que esto ocurra es que las gafas suelen estar a una temperatura mucho más baja que el cuerpo,

más bien a temperatura ambiente. De ahí que las gafas se vean aproximadamente del color del fondo de la imagen. Se ha extraído del artículo citado anteriormente una imagen para que sirva como ejemplo visual de lo que se pretende decir:

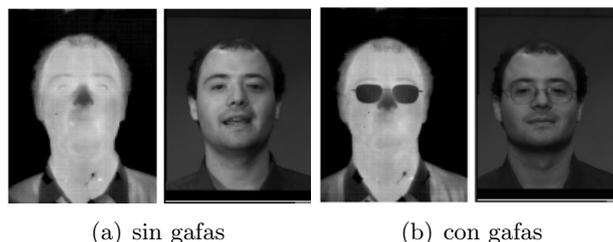


Figura 2.18: Efecto de las imágenes infrarrojas térmicas sobre las gafas

En este tipo de imágenes vemos que podemos detectar muy fácilmente, como si de gafas de sol se tratasen, todo tipo de gafas. En el caso de luz visible, dependiendo del tipo de gafas, estas pueden variar sustancialmente la apariencia del usuario o no. El detector de gafas que posteriormente se explicará, utiliza los bordes como medida de si existen realmente gafas de sol o no, pero con este tipo de imágenes se pueden detectar cualquier tipo de gafas.

En el artículo antes citado, se realiza reconocimiento facial con imágenes térmicas. Respecto al problema de las gafas, se detectan y se evita el entorno cercano de las mismas, pero se utiliza el resto de la imagen térmica para realizar el reconocimiento.

En [1] se propone utilizar una fusión de varias fuentes de imágenes, visuales y térmicas, para comprobar la identidad de un usuario. Este sistema, como en el anterior artículo del que se han extraído imágenes (aparentemente pertenece al mismo grupo de trabajo), tiene en cuenta que las gafas normales son un problema para el reconocimiento, y trata de evitar esta zona.

En [2] se utilizan las imágenes térmicas para hallar una huella única para cada persona (parecida a una huella digital) a partir de la imagen de la cara, mediante técnicas de extracción de bordes. Esta técnica totalmente diferente se puede utilizar para el reconocimiento de una persona mediante biometría, puesto que las características térmicas de una persona, en condiciones normales, son siempre las mismas.

Literatura sobre imágenes de infrarrojos obtenidas por reflexión no se ha encontrado demasiada, en el caso de realizar un reconocimiento facial sobre el individuo. Esta técnica de obtención de imágenes de caras por infrarrojos y reflexión se utiliza más a la hora de la detección de somnolencia del individuo [4], con iluminadores de infrarrojos de poca potencia, para poder realizar un seguimiento sobre las pupilas para poder detectar la frecuencia con la que se abren y cierran y por tanto el nivel de somnolencia del individuo. Además, basado en este sistema de detección de somnolencia es en el que se va a implementar el sistema que se presenta en este documento.

Capítulo 3

Técnicas empleadas

A continuación se expondrán las técnicas de visión computacional e inteligencia artificial utilizadas en este sistema.

Estas estarán ordenadas desde el nivel más bajo de abstracción hasta los niveles más altos. Los niveles más bajos se corresponden con el tratamiento de píxeles, al ir subiendo de nivel de abstracción, se tratarán zonas reducidas de la imagen para obtener alguna característica, y posteriormente, funciones más complejas para realizar tareas más cercanas a las que hacen los humanos de forma natural, como es la detección de caras o la comparación entre patrones.

Empezaremos por el umbralizado utilizado para el detector de gafas de sol. Posteriormente se hablará de la teoría perteneciente a los métodos de tratamiento de la iluminación probados. A continuación, se hablará sobre las operaciones morfológicas y la detección de bordes, utilizados ambos métodos en el detector de gafas. Seguidamente se hablará sobre el algoritmo utilizado para la detección de caras, el método de extracción de características, el clasificador de patrones, las medidas de distancia y por último una pequeña introducción a los sistemas borrosos.

3.1. Umbralizado de Otsu

Para extraer información de una imagen de forma elemental, para luego por ejemplo realizar una segmentación o extracción de bordes, realizamos una umbralización. Este método consiste en hacer que, a partir de cierto valor de nivel de gris llamado umbral, los píxeles de la imagen valgan 1 y por debajo de este nivel los píxeles de la imagen destino valgan 0. Expresado matemáticamente nos queda la siguiente ecuación

$$I_{umb}(x, y) = \begin{cases} 0 & 0 \leq I_{in}(x, y) \leq \alpha \\ 1 & \alpha < I_{in}(x, y) \leq 255 \end{cases} \quad (3.1)$$

Como se puede ver en la ecuación anterior, el problema fundamental es obtener el valor de α , para alcanzar los objetivos que se pretenden. El valor de este umbral puede ser constante a lo largo de toda la imagen o puede estar en función de la zona de la misma, aunque en la ecuación no se vea explícito, como se explicará en el siguiente párrafo.

Existen distintas formas de hallar el umbral. Una de ellas es asignar un umbral fijo que funcione aproximadamente bien en todos los casos. Este método no tiene en cuenta condiciones de iluminación ni otros aspectos que se pueden dar en la imagen, que es que los objetos en la escena pueden venir dados por colores diferentes, y por tanto pueden ser diferenciados por ello.

Existen otros métodos de hallar este umbral como puede ser el adaptativo, en el que, se toma una zona de la imagen como referencia y se calcula el umbral cuya probabilidad de nivel de gris es igual hacia un lado que hacia otro (en el caso de un umbralizado en 2 clases, como es lo que se pretende).

Los métodos de obtención del umbral anteriormente explicados son simples, pero dependiendo de la aplicación, suficientes, puesto que otros métodos de cálculo de umbrales, suponen mayor complejidad computacional, lo cual en ocasiones no es admisible. Si se pretende hacer una umbralización con un umbral que permita una separación de clases óptima, tenemos que recurrir al método de Otsu [33].

El método de cálculo de umbral de Otsu nos da, en función del histograma de la imagen pasada como argumento, el mejor umbral teórico para la decisión entre clases, o lo que es lo mismo entre objetos de distintos colores en la imagen. El umbral obtenido es el resultado de la maximización de un parámetro, que en el método se llama varianza entre clases. Este parámetro nos da una idea de la separación entre las clases en las que se intenta dividir el histograma de la imagen. Para calcularlo, primero se obtienen las medias de las clases en las que se quiere dividir el histograma, y después, se define la varianza entre clases como:

$$\sigma = \omega_1 (\mu_1 - \mu_T)^2 + \omega_2 (\mu_2 - \mu_T)^2 \quad (3.2)$$

siendo ω_1 la probabilidad de que el nivel de grises de la clase 'i' esté por debajo del umbral, ω_2 la probabilidad de que esté por encima del umbral, μ_i es la media de la clase 'i', y μ_T es la media global de la imagen.

La ventaja fundamental de este método, es que asegura un umbral de segmentación que va a separar de la manera estadística mejor posible 2 clases entre niveles de una imagen, por tanto estadísticamente es un método correcto e incluso óptimo.

Otra de las ventajas de este método es que viene implementado en buena parte de las librerías más utilizadas en visión artificial: la Image Processing Toolbox de Matlab y OpenCV, entre otras. En el caso de OpenCV, hasta la versión 1.0.0 no estaba disponible este método de umbralizado, e incluso, en esta versión y posteriores, este método no está reflejada su existencia en la documentación, pero si vamos a las modificaciones realizadas en las distintas versiones, se vemos que se ha incluido este método de cálculo de umbral. Para poder utilizarlo, en la función `cvThreshold`, se han de introducir los parámetros como si se utilizase un umbral normal, pero en las opciones se ha de introducir la opción `CV_THRESH_OTSU`.

3.2. Tratamiento de la iluminación

3.2.1. Introducción

La superficie que define la piel que se encuentra en la cara tiene ciertas propiedades que pasamos a describir a continuación. Las propiedades de la piel respecto de la reflexión de luz, se pueden aproximar a las de una superficie Lambertiana, la cual cumple una ley un tanto diferente a la que se está acostumbrado con Snell. En este tipo de superficies, la cantidad de luz radiada en una determinada dirección, es proporcional al coseno del ángulo que forma el rayo con la normal de la superficie. Este patrón de reflexión difuso nos es útil, puesto que independientemente de la dirección de la luz incidente vamos a poder ver en cualquier dirección la luz reflejada de la cara. Como en todos los casos, dependiendo de la dirección de la luz, veremos el objeto con zonas

más iluminadas que otras y con sombras, aunque, en un caso especial, si la luz sigue un patrón difuso, podremos verlo igual en todas las direcciones.

Como se ha dicho en la introducción, el problema de la iluminación no controlada todavía no se ha resuelto satisfactoriamente, puesto que es un problema altamente no lineal visto desde la perspectiva de dos dimensiones. Los efectos de una iluminación no controlada en una imagen es, por un lado, la saturación (nivel muy alto de gris, cercano al blanco) de algunas zonas en la que podemos tener puntos de interés de la cara, con la posibilidad de perderlos debido a que no se es capaz de diferenciar demasiados niveles de gris en la zona y no poder detectar el punto correctamente. Por otro lado, la iluminación puede generar sombras dependiendo de la dirección de incidencia de la luz, con el mismo resultado que antes. Estos 2 efectos tienen además como consecuencia la variación totalmente aleatoria de la apariencia de la cara, lo cual, para un algoritmo de la familia de PCA resulta muy perjudicial.

A pesar de no ser un problema resuelto todavía, se han probado diversas aproximaciones para resolverlo y algunas de estas soluciones pasamos a enumerarlas a continuación.

La primera forma (la más habitual) de paliar este problema, es aplicar una ecualización de histograma, para adaptar el histograma de las imágenes a un histograma patrón, en nuestro caso plano. El resultado de aplicar la ecualización de histograma a una imagen es que se produce el realce en los niveles de la misma que nos interese. El problema de este método es que, si tomamos el espectro del histograma como una función de densidad de probabilidad, observamos que el ruido blanco tiene la misma forma de espectro. De hecho, en ocasiones, se introducirá ruido de sal y pimienta al aplicarlo. Este problema añadido se soluciona aplicando posteriormente un filtro de Mediana, el cual borrosifica un poco la imagen, pero a cambio eliminamos las no linealidades que introduce el ruido de sal y pimienta.

Otro procedimiento probado para solucionar este gran problema, se basa en el artículo [20]. En él se sugiere que se puede dividir la información de una imagen facial en dos subespacios diferentes: el del movimiento de la cara y el de la iluminación. Con esto, se dice que se puede extraer la iluminación media muestreando unas cuantas imágenes representativas. Con esto, se dice poder extraer las correspondientes bases de las imágenes para poder normalizar el espacio de iluminación y por último poder independizar así el reconocimiento de las imágenes del espacio de iluminación. Si tomamos esta idea, pero en vez de extraer características como se indica, introducimos nuevos patrones de iluminación a los patrones básicos, podemos enriquecer la base de datos y por tanto ser capaces de soportar más situaciones de iluminación diferente. Como se comentará posteriormente, este método es prometedor, salvo por el tiempo de computación que emplea que es bastante superior, pues estamos incrementando linealmente el número de patrones con el que tenemos que comparar cada imagen de entrada.

Buscando otra solución para este problema, se ha probado un algoritmo que imita el funcionamiento del ojo para filtrar en parte los problemas de la iluminación. El sistema visual humano, se comporta de manera logarítmica con los cambios de iluminación, lo cual nos va a dar cierta invariabilidad para cambios de iluminación no muy bruscos en la escena. El algoritmo que implementa este tratamiento para un único canal (el de niveles de gris en nuestro caso) se llama SSR (Single Scale Retinex).

En otro artículo [34], se ha utilizado una técnica basada en el cálculo de los primeros términos de las ecuaciones de reflectancia en las superficies lambertianas, a partir de las cuales se puede estimar la dirección de la iluminación de la cara, para, posteriormente aplicar métodos para contrarrestar esa iluminación que nos distorsiona la cara.

En [35] se realizan varias pruebas de corrección de iluminación por zonas, mediante varios métodos, como son la ecualización de histograma, la corrección gamma, y otros métodos, y se

expone la comparación entre los mismos, cuando se aplica a imágenes de la base de datos de caras de Yale.

En [11] se propone un sistema de identificación facial independiente de la iluminación mediante ecualización de histograma y una transformada Wavelet. También está probado con la base de datos de caras de Yale. Con su mezcla de métodos dicen conseguir porcentaje de reconocimientos positivos cercanos superiores, en media, al 95 %.

3.2.2. Ecualización de histograma uniforme

Esta técnica es la más comúnmente utilizada para realizar las primeras aproximaciones a la resolución del problema del realce. Antes de explicar cómo hacer la ecualización, comenzaremos por explicar qué es un histograma.

El histograma es la representación de la cantidad de los niveles de gris que podemos encontrar en la imagen, esto es, cuántos píxeles de un determinado nivel de gris podemos encontrar en una imagen, y este contador, calculado para todos los niveles posibles de gris en la imagen. Si vamos un poco más allá en esta definición, nos damos cuenta de que esta función se puede equiparar, previa normalización, a una función de densidad de probabilidad, lo cual utilizaremos posteriormente para explicar ciertos efectos.

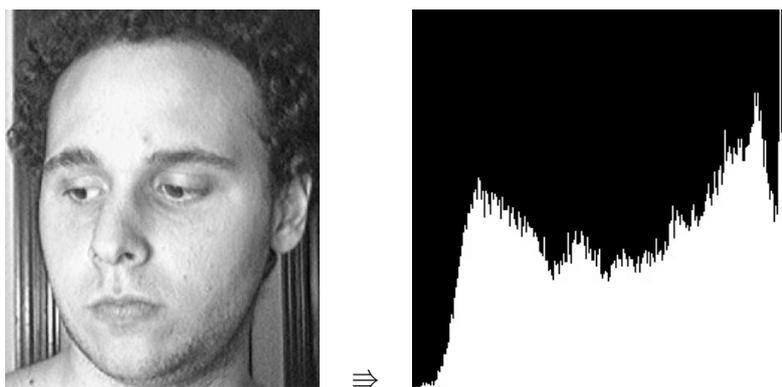


Figura 3.1: Ejemplo de histograma de una imagen

La ecualización de histograma, no solo uniforme, sino en general, pretende el realce o mejora de una imagen variando su histograma, esto es, modificar la distribución de probabilidad que tenemos de encontrar un determinado nivel de gris en la imagen hasta alcanzar una función de densidad deseada. Esta función de densidad es en realidad el histograma normalizado.

Ahora pasamos a definir la operación formalmente: Sea $f(u, v)$ la imagen de entrada y $H_f(n)$ su histograma asociado, definido entre los niveles f_0 y f_k . Sea también $g(u, v)$ la imagen de destino y $H_g(n)$ el histograma de destino definido entre g_0 y g_k . Para trabajar con funciones de densidad de probabilidad en lugar de histogramas, normalizamos por la superficie de la imagen.

$$\sum_{i=f_0}^{f_k} H_f(i) = \sum_{i=g_0}^{g_k} H_g(i) = NxM \quad (3.3)$$

$$p_f(n) = \frac{H_f(n)}{NxM} \quad p_g(n) = \frac{H_g(n)}{NxM} \quad (3.4)$$

Queremos obtener una función tal que transforme el histograma de la imagen de entrada al histograma deseado, formulado matemáticamente:

$$g(u, v) = T(f(u, v)) \quad (3.5)$$

Para obtener esta función T , aplicamos la ecuación de ecualización:

$$\int_{f_0}^f p_f(s) ds = \int_{g_0}^g p_g(r) dr \quad (3.6)$$

Las funciones de densidad de probabilidad en nuestro caso, son discretas, con lo que posteriormente, sustituiremos la integral por el sumatorio equivalente en el dominio discreto.

En nuestro caso, queremos obtener un histograma de destino plano, lo cual obedece a una función de distribución de probabilidad uniforme, la cual viene descrita por la siguiente ecuación:

$$p_g(g) = \frac{1}{g_k - g_0} \quad (3.7)$$

Por tanto, el segundo miembro de la ecuación 3.6 combinándola con la ecuación anterior será:

$$\int_{g_0}^g p_g(r) dr = \int_{g_0}^g \frac{1}{g_k - g_0} dr = \frac{(g - g_0)}{g_k - g_0} \quad (3.8)$$

Entonces podemos obtener una función que nos dé una expresión de la salida en función de las entradas y de los datos del histograma deseado, a partir de la ecuación anterior y de la ecuación de ecualización (3.6), con lo que nos queda:

$$g = (g_k - g_0) \int_{f_0}^f p_f(s) ds + g_0 \quad (3.9)$$

Escribiendo esta ecuación en forma continua:

$$g = T(f) = (g_k - g_0) \sum_{i=f_0}^f p_f(i) + g_0 \quad (3.10)$$

Con este método hemos conseguido realzar la imagen de entrada hasta tener el máximo contraste posible (cuanto más plano es el histograma, el contraste es mayor). En cambio, al tener esta distribución de densidad de probabilidad, la cual se asemeja al ruido blanco, se puede llegar a provocar en la imagen ruido de tipo de sal y pimienta, en algunos píxeles. En la siguiente sección veremos cómo solucionar en parte este problema.

El resultado de aplicar este tratamiento a una imagen de prueba es el mostrado en la figura 3.2

En la figura vemos los detalles de la imagen se ven ahora más claramente, como los dibujos hechos en la pizarra o las letras de las máquinas. Además también vemos cómo la zona donde el

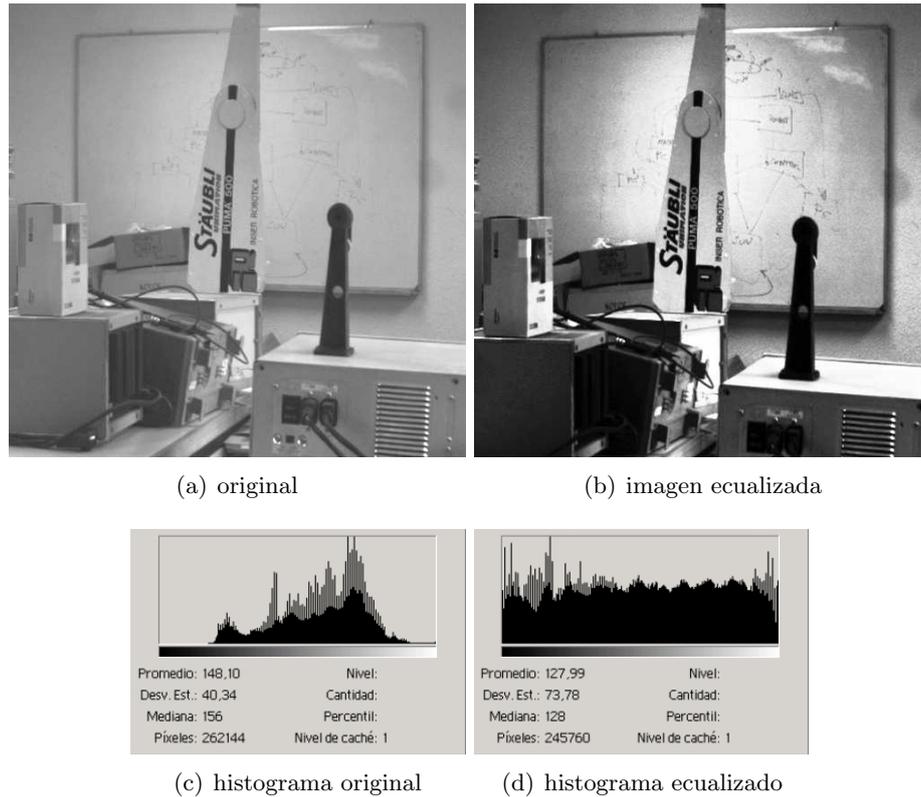


Figura 3.2: Ejemplo de aplicación de ecualización de histograma a una imagen

foco donde ha iluminado más, resalta más que el resto de la escena, que se ha visto oscurecida. Respecto al histograma, se aprecia que no es exactamente plano pero en buena medida se parece. Una de las causas por las que no es exactamente plano es porque la función no es continua, sino discreta, por lo que se puede perder información debido a esa discretización. En la imagen tratada podemos observar un aumento muy grande del contraste, puesto que esta es la propiedad que tiene una imagen con histograma plano. Dicho de otro modo y tomando el histograma como una función de densidad de probabilidad, la entropía en una función de densidad uniforme es máxima al tener todos los valores de la función la misma probabilidad de aparecer.

3.2.3. Filtrado de mediana

Para solucionar el problema de la introducción de ruido en algunos píxeles por el hecho de aplicar una ecualización uniforme a las imágenes de destino, recurrimos a un filtrado de mediana con la mínima máscara (3×3), por el tamaño de la imagen. Este filtrado no lineal, consiste en la elección del valor de la mediana de los valores de los píxeles que están dentro de la máscara. La mediana se define como el valor que queda en el elemento central del array una vez ordenado este de mayor a menor valor.

$$\begin{array}{|c|c|c|} \hline 3 & 5 & 2 \\ \hline 6 & 4 & 9 \\ \hline 1 & 8 & 7 \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline 7 & 8 & 9 \\ \hline \end{array}$$

Figura 3.3: Ejemplo de extracción de mediana

Con este tipo de filtrado se consigue la eliminación de este ruido tan molesto puesto que, el ruido de sal y pimienta supone una variación muy brusca en los píxeles de alrededor, por tanto, estos píxeles al ordenar el array irán a los extremos, y por tanto lo alejaremos de los elementos centrales, eliminando la probabilidad de que el píxel de destino contenga información de este ruido aleatorio.

Como inconveniente de la aplicación de este método tenemos que la imagen se emborrona de manera considerable a medida que aumentamos el tamaño de la máscara, igual que pasa cuando aplicamos otros filtros paso bajo en la imagen, como pueden ser los de media u otros no lineales como el filtro paso bajo gaussiano, que elimina el ruido, pero en este caso cuando ese ruido tiene otras distribuciones de densidad de probabilidad.

A continuación se muestra en una imagen de ejemplo el efecto de aplicar un filtrado de mediana.

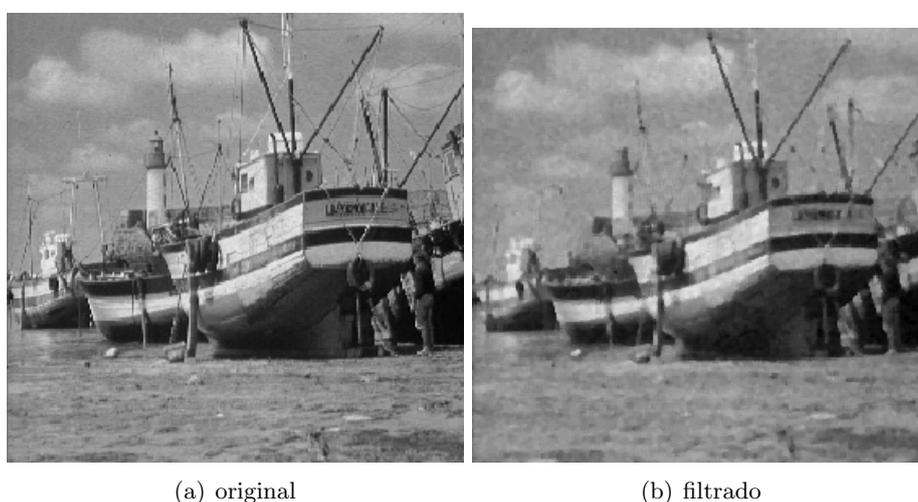


Figura 3.4: Ejemplo de aplicación de filtro de mediana a una imagen

En la figura no se observa bien el efecto real para el que se ha diseñado el filtro, puesto que la imagen original no ha sido contaminada con ruido de sal y pimienta. Lo que sí se puede apreciar con facilidad es el efecto vidrioso que deja este filtro en la imagen final, así como la reducción de bordes finos de la imagen y la disminución del nivel de detalle debido al tamaño de la máscara.

3.2.4. SSR (Single Scale Retinex)

Con este algoritmo se intenta convertir las imágenes que tenemos en la base de datos y las de la cámara en imágenes tales que su variación respecto a cambios de luz sea mínima. El algoritmo Retinex (Retina y Cortex) [6] está basado en la respuesta logarítmica que posee en conjunto el sistema visual humano. Con SSR (versión reducida de este método), se tratará de convertir una imagen en escala de grises que puede estar afectada de cualquier forma por la iluminación cambiante en otra imagen cuya variación con la iluminación sea mínima.

Lo que se intenta realmente con el algoritmo Retinex es tener una aproximación al problema real del color constante. Este problema consiste en lo siguiente: Dependiendo de la iluminación, del color de la luz con la que se vea, de las sombras generadas, del tipo de superficie sobre la que se refleja la luz, y otros factores, un mismo color real, en una imagen se puede apreciar con una variada gama de colores. Podemos describir este fenómeno, de manera matemática de la siguiente forma:

Sea $E(\lambda)$ la distribución de potencia para la longitud de onda λ , y la función de reflexión de un punto de la superficie $S(\lambda, x)$. Entonces la función de la señal luminosa que llega al ojo, reflejada desde ese punto de la superficie viene dada por la siguiente expresión:

$$C(\lambda, x) = E(\lambda) S(\lambda, x) \quad (3.11)$$

Una vez tenemos esta potencia en el ojo, los foto-receptores sensibles a cada color (conos) tienen una función de sensibilidad para distintas frecuencias luminosas, por tanto, para obtener la potencia recibida por cada uno de estos receptores, tendremos que acumular todas las longitudes de onda para saber la potencia total luminosa adquirida por cada receptor (la ecuación que obtenemos, solo es válida para la luz reflejada desde el punto x , para obtener la total obtenida por toda la superficie, tendremos que calcular la integral):

$$\rho_i(x) = \int_{\lambda_0}^{\lambda_1} C(\lambda, x) R_i(\lambda) d\lambda \quad (3.12)$$

siendo $\rho_i(x)$ la respuesta del receptor de clase i a la potencia recibida desde el punto x , $C(\lambda, x)$ descrita por la ecuación 3.11, y $R_i(\lambda)$ la sensibilidad del receptor i respecto de la longitud de onda λ . En el artículo podemos ver la versión discreta de estas ecuaciones, puesto que en él, nos exponen el desarrollo para una serie de frecuencias representativas, no para todo el espectro. En este desarrollo, para el cálculo de la potencia reflejada, no se ha tenido en cuenta en ningún momento dobles reflexiones ni reflexiones especulares, ni otros efectos que pueden complicar considerablemente todo lo que acabamos de describir.

Con las ecuaciones anteriores, podemos obtener un factor que nos da una idea de la iluminación recibida por cada píxel en una imagen. Este factor viene dado por la siguiente ecuación:

$$l_i(x) = \log\left(\frac{\rho_i(x)}{G_i}\right) \quad (3.13)$$

Donde G_i es la media geométrica del receptor para la clase i de receptor.

Una vez hemos mostrado las ecuaciones que rigen el comportamiento de la iluminación en una imagen y viendo que su comportamiento no es lineal (tenemos logaritmos en la última ecuación), el problema del color constante en una escena es bastante más complicado de lo que parece (todavía no ha sido resuelto satisfactoriamente).

Este problema, en los algoritmos que están basados en el color, supone una gran desventaja si tenemos en cuenta el problema del color constante, puesto que un color, en una imagen puede estar representado por una gama muy grande de colores en la realidad, dependiendo del color e intensidad de la luz con la que se ilumine, de las sombras que se generen o de las reflexiones que pueda tener, además del nivel de transparencia de la superficie (aunque en este caso entraríamos en el campo de la radiosidad), por tanto pueden fallar muchos de estos algoritmos si no realizamos un tratamiento previo para paliar este problema.

Si realizamos varias aproximaciones y asumimos que las funciones de sensibilidad y de emisión son gaussianas, podemos llegar a las ecuaciones que se van a detallar de SSR particularizando para una banda estrecha de longitudes de onda.

Si particularizamos el algoritmo para un solo canal, como se ha probado en [16, 17], y partiendo de las ecuaciones anteriores, podemos obtener una imagen de destino con una cierta corrección de la iluminación respecto de la original.

SSR se aplica a una imagen mediante la siguiente ecuación:

$$I_R(x, y) = \alpha (\log(A(x, y)) - \log(A(x, y) * F(x, y))) - \beta \quad (3.14)$$

Siguiendo con la notación anterior, A es la imagen de entrada, * es el operador de convolución, y G es la gaussiana generada a partir de:

$$F(x, y) = ke^{-\frac{x^2+y^2}{c}} \quad (3.15)$$

Donde c es la varianza de la gaussiana empleada para el algoritmo y k es una constante que normaliza el tamaño de la gaussiana:

$$k = \frac{1}{\iint e^{-\frac{x^2+y^2}{c}} dx dy} \quad (3.16)$$

A la vista de la ecuación fundamental de SSR 3.14, vemos que tenemos varios parámetros libres, los cuales se hallarán experimentalmente, α , β y c, de los cuales c da buenos resultados con $c=4$, puesto que más, nos produce una gaussiana tan amplia que se introducen las no linealidades que queríamos evitar por la iluminación, y menos no se introduce la suficiente información y termina degenerando en una imagen de bordes.

La diferencia entre SSR y MSR (Multi Scale Retinex) es que se aplica a varios canales y esta aplicación está ponderada por unos pesos como se muestra en la siguiente ecuación

$$R_{MSR_i} = \sum_{n=1}^N \omega_n SSR_i \quad (3.17)$$

siendo N el número de escalas, ω_n el peso de cada una de las escalas y SSR_i la imagen resultado de aplicar SSR al canal i.

Las aplicaciones del algoritmo Retinex son variadas, tanto de SSR como de MSR. Algunas de las aplicaciones de SSR se pasan a mostrar a continuación.

Una aplicación es el tratamiento de iluminación, razón por la cual se ha probado este método en el sistema. En [24] se utiliza este algoritmo para el tratamiento de la iluminación en las imágenes, junto con una corrección de histograma, para adecuarlo a uno de destino.

Otra aplicación de algoritmos derivados de Retinex aplicados a realce de imágenes, es el tratamiento en imágenes aéreas para hacer más claros los objetivos o la eliminación de niebla.

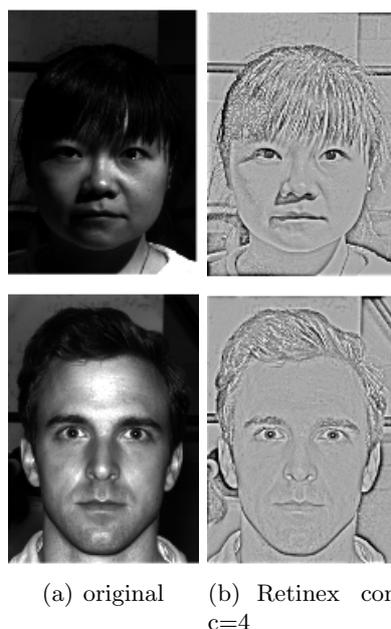


Figura 3.5: Retinex aplicado a imágenes de prueba con iluminación variada

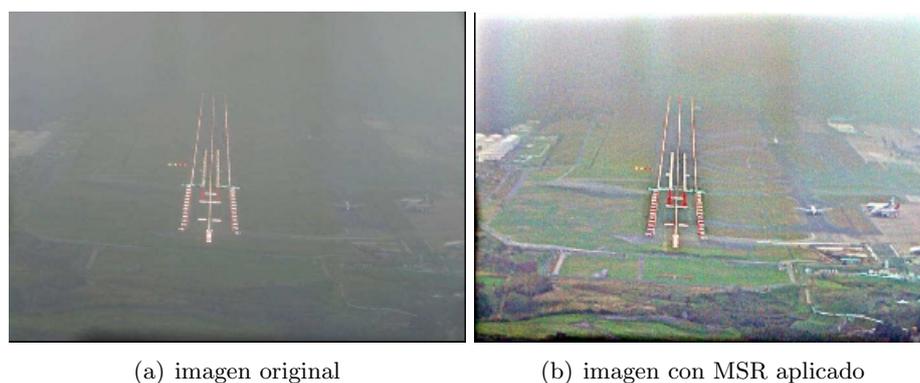


Figura 3.6: Aplicación de Retinex para la eliminación de niebla

Como podemos observar en la figura, con este algoritmo se es capaz de eliminar parte de la niebla existente en una imagen aérea. En este caso se ha aplicado MSR a los 3 canales de color y con una varianza de la gaussiana utilizada en la convolución en torno a de 100, la cual es varios órdenes superior a la utilizada en nuestro caso.

También se utiliza en otras aplicaciones donde se necesita un realce de la imagen, como es el caso de las imágenes médicas, dependiendo de la naturaleza de éstas y del efecto deseado, se aplicará SSR o MSR. Un ejemplo de esta aplicación se muestra en la figura 3.7

3.3. Operaciones morfológicas: dilatación y erosión

Estas operaciones se utilizan para modificar algunos aspectos de una imagen binaria para eliminar bordes finos con la erosión o para hacerlos más gruesos en caso de que se quieran resaltar

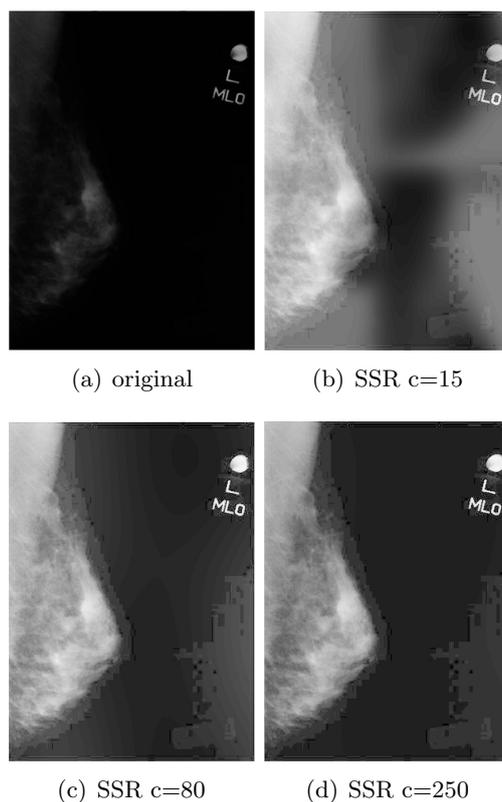


Figura 3.7: SSR aplicado a imágenes médicas con distintos valores de varianza

los bordes finos con la dilatación.

Formalmente, estos operadores se describen matemáticamente de la siguiente manera:

Dilatación:

$$I \oplus S = \{p \in E^2 : p = i + s \forall i \in I, s \in S\} \quad (3.18)$$

Erosión:

$$I \ominus S = \{p \in E^2 : p + s \in I \forall s \in S\} \quad (3.19)$$

Estas expresiones, por si solas no aparentan decir demasiado, sin embargo, con un ejemplo sencillo podremos ver mejor el efecto de los operadores sobre un vector simple:

$$\begin{aligned} I &= \{(0, 2), (1, 2), (2, 0), (2, 1), (2, 2), (2, 3), (3, 2), (4, 2)\} \\ S &= \{(0, 0), (0, 1)\} \\ I \oplus S &= \{(0, 2), (1, 2), (2, 0), (2, 1), (2, 2), (2, 3), (3, 2), (4, 2), (0, 3), (1, 3), (2, 4), (3, 3), (4, 3)\} \\ I \ominus S &= \{(2, 1), (2, 2), (2, 3)\} \end{aligned}$$

Como podemos observar, en la dilatación se añaden píxeles en la dirección que indique el operador S . En la erosión solo tomamos los puntos que, una vez sumados con el operador pertenecen a la imagen I , con lo cual, estamos eliminando píxeles. Si en la imagen se presentan bordes verticales, y en el caso de que el operador S sea horizontal, como en el ejemplo, éstos se eliminarán, puesto que solo perduran aquellos que la vecindad en la dirección del operador sea la suficiente.

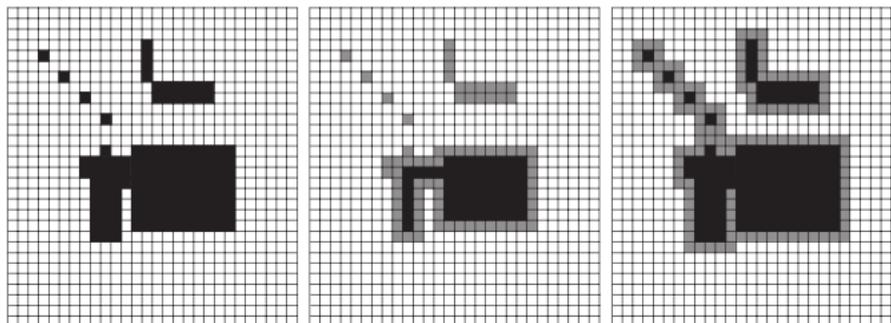


Figura 3.8: Ejemplo de dilatación y erosión, a la derecha, la imagen original, en el centro, el resultado de una erosión, y a la derecha, el resultado de la dilatación

Si aplicamos estas 2 operaciones en el orden correcto sobre una imagen umbralizada podemos eliminar o resaltar características que sean adecuadas. Si hacemos una dilatación sobre una imagen en la que los bordes están próximos, lo que hacemos es juntar unos cuantos para formar un objeto, y si luego se aplica una erosión, se eliminan los bordes más débiles. Este tipo de técnica la utilizaremos en la detección de gafas, puesto que los bordes que se generan en estos casos debidos a las reflexiones, tienen las características anteriormente comentadas. Por tanto, si realizamos estas operaciones morfológicas antes de aplicar un filtro de bordes, podremos agrupar aquellos bordes finos con la superficie que ocupa la gafa y reducir el número de bordes que no conducen a unos buenos resultados.

3.4. Obtención de bordes

Una vez hemos visto los tratamientos a bajo nivel, esto es a nivel de píxeles, pasamos a tratamientos en regiones más amplias de las imágenes, regiones de interés dentro de una imagen, o a la imagen completa. Este tipo de tratamientos puede consistir, como es nuestro caso, en calcular los bordes de una región para luego hallar, con tratamientos de un mayor nivel de abstracción, formas, figuras, segmentar regiones, detectar contornos y averiguar sus características, etc.

En una imagen podemos obtener 2 clases de información, en función de la frecuencia con la que se da en la imagen, paso bajo y paso alto. La información paso bajo nos da una idea de la distribución de la iluminación y de otros tipos de información cuya frecuencia de variación es muy baja. Este tipo de información se corresponde, en una imagen real, al fondo de la imagen, nivel medio o información correspondiente a la iluminación. En cambio, la información paso alto, es la correspondiente a una frecuencia de variación muy alta. Esto último se corresponde con la información de variación rápida de una imagen, esto es, normalmente con los bordes o con el ruido de la imagen.

3.4.1. Máscaras

Los dos tipos de información se pueden extraer con el mismo procedimiento: las máscaras. Éstas se pasan por la imagen y se genera otra imagen con el tipo de información adecuada. Las máscaras para extraer un tipo u otro de información varían considerablemente. En el caso de extraer información paso bajo, podemos utilizar una máscara de media o una gaussiana. En el

caso de querer extraer características paso alto, utilizaremos máscaras con nombre propio (Sobel, Prewitt), dependiendo de las propiedades paso alto que queramos extraer.

Para aplicar las máscaras a una imagen se ha de aplicar la convolución bidimensional, que consiste en lo siguiente: se da la vuelta a la máscara, se aplica al principio de la imagen, y se multiplican los elementos de la imagen por los de la máscara, se suman, y este será el resultado del píxel de la imagen de destino. Expresado matemáticamente la ecuación de convolución bidimensional

$$g(u, v) = h(u, v) * f(u, v) = \sum_i \sum_j f(i, j) h(u - i, v - j)$$

Con esta convolución bidimensional se aplica genéricamente a dos imágenes dadas, cuando se quiere componerlas, pero también se puede aplicar las máscaras, si vemos estas como imágenes pequeñas que realizan una función determinada sobre pequeñas porciones de imagen para extraer alguna característica. Las máscaras suponen, por tanto, un tratamiento de la imagen local.

A continuación pasamos a describir dos máscaras paso bajo comúnmente utilizadas.

$$M(u, v) = \sum_{k=u-1}^{u+1} \sum_{j=v-1}^{v+1} I(u, v) \quad G(u, v) = \frac{1}{2\pi\sigma} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

1/9	1/9	1/9	1	4	1
1/9	1/9	1/9	4	12	4
1/9	1/9	1/9	1	4	1

media

gaussiana ($\sigma=0.391$)

Figura 3.9: Máscaras paso bajo

La máscara de media se utiliza para el filtrado de imágenes para eliminación de ruido y la gaussiana también, cuando tenemos la certeza de que este sigue esa distribución y además, cuando se sabe que la imagen va a ser posteriormente tratada para la extracción de bordes y queremos eliminar el ruido introducido.

Estas máscaras normalmente se utilizan para niveles más bajos de abstracción

Las máscaras paso alto se utilizan para detección de gradiente, por lo tanto, son capaces de detectar variaciones bruscas dentro de la imagen. Se aprovecha esta circunstancia para detectar ciertas características particulares como son los bordes, los cuales obtenemos a partir de las ecuaciones del gradiente bidimensional, realizando las aproximaciones oportunas.

$$G[f(u, v)] = \begin{pmatrix} G_u \\ G_v \end{pmatrix} = \begin{pmatrix} \frac{\partial f(u, v)}{\partial u} \\ \frac{\partial f(u, v)}{\partial v} \end{pmatrix} = \begin{cases} |G| = \sqrt{G_u^2 + G_v^2} & |G| \approx |G_u| + |G_v| \\ \angle G = \arctan\left(\frac{G_v}{G_u}\right) \end{cases} \quad (3.20)$$

A partir de la ecuación anterior y aproximando las derivadas parciales de distintas maneras, obtenemos las distintas máscaras, como podemos ver a continuación:

$$\text{Prewitt} \begin{cases} G_u = \frac{\partial f(u, v)}{\partial u} \approx f(u, v) - f(u - 1, v) \\ G_v = \frac{\partial f(u, v)}{\partial v} \approx f(u, v) - f(u, v - 1) \end{cases} \quad (3.21)$$

0	-1	0	0	0	0
0	1	0	-1	1	0
0	0	0	0	0	0

Figura 3.10: Máscaras de Prewitt

Vemos que la máscara de Prewitt se puede computar fácilmente con un par de operaciones sobre los píxeles locales, debido a la simplificación del gradiente que se ha tomado. Si tomamos una simplificación del gradiente algo más compleja, tenemos el ejemplo de la máscara de Sobel, en la cual los distintos píxeles tienen distintos pesos a la hora de computar el valor de la imagen de bordes. En el caso de la máscara de Prewitt, se necesita 2 multiplicaciones y una suma sobre los vecinos más cercanos del píxel que estamos calculando en la imagen de destino, mientras que en el método de Sobel se necesitan 6 multiplicaciones y 5 sumas. Las diferencias de uso entre los 2 dependen de la aplicación.

$$\text{Sobel} \begin{cases} G_u = \frac{\partial f(u,v)}{\partial u} \approx [f(u+1, v-1) + 2f(u+1, v) + f(u+1, v+1)] \\ \quad - [f(u-1, v-1) + 2f(u-1, v) + f(u-1, v+1)] \\ G_v = \frac{\partial f(u,v)}{\partial v} \approx [f(u+1, v-1) + 2f(u+1, v) + f(u+1, v+1)] \\ \quad - [f(u-1, v-1) + 2f(u-1, v) + f(u-1, v+1)] \end{cases} \quad (3.22)$$

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

Figura 3.11: Máscaras de Sobel

Una vez se ha descrito estas máscaras, pasamos a ver ejemplos de lo que se puede llegar a obtener para las distintas imágenes. En la figura 3.12, se muestra un ejemplo de la extracción por separado de los bordes horizontales y verticales (G_u y G_v) y a partir de las anteriores la imagen de módulo, y en la figura 3.13, se representa un ejemplo de extracción de la imagen de módulo y fase, obtenida a partir de la ecuación 3.20. La máscara utilizada para calcular los bordes horizontales y verticales ha sido la de Prewitt.

Una vez hemos obtenido estas imágenes (sobre todo la de módulo es la más utilizada), se umbraliza para resaltar los bordes más significativos. Dependiendo del umbral utilizado se resaltarán más o menos bordes, en función de la aplicación.

Decir que existen más tipos de máscaras, pero en realidad no se explican al no ser frecuentemente usados, como por ejemplo Roberts, que se utiliza para destacar bordes diagonales, o en una determinada dirección (dependiendo del tamaño de la máscara utilizada). En una máscara de Roberts de 3x3, tenemos la posibilidad de distinguir diferencias de ángulo de 45°.

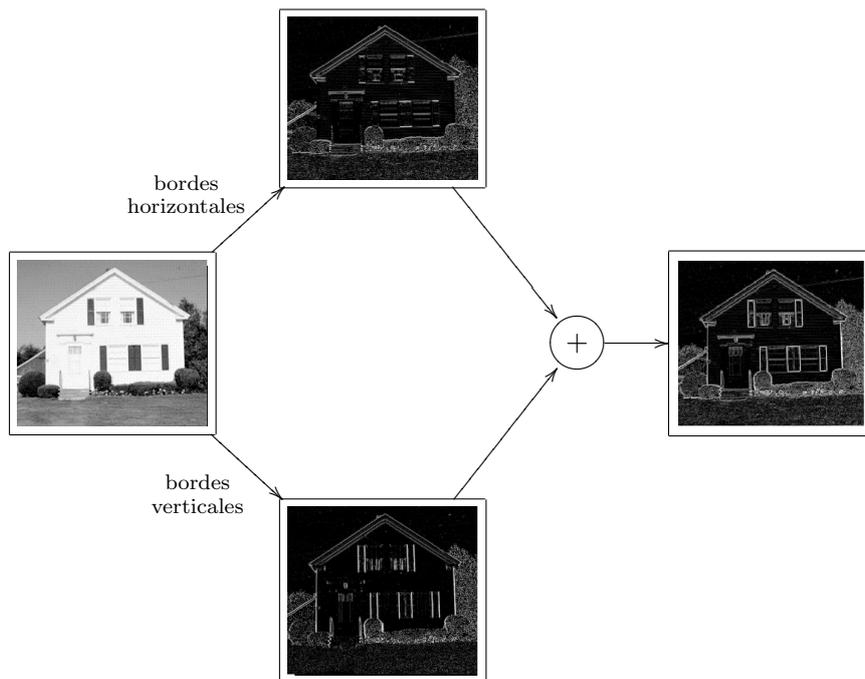


Figura 3.12: Obtención de imágenes de bordes

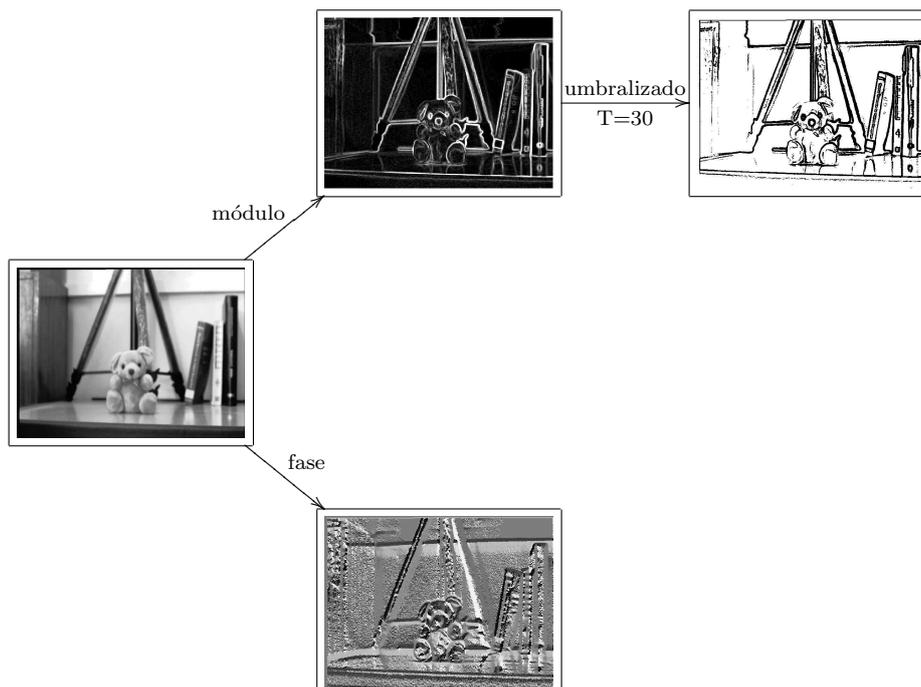


Figura 3.13: Imágenes de módulo y fase

3.4.2. Canny

Además de las máscaras de obtención de bordes, existen otros métodos más elaborados para su obtención. El método más extendido se basa en el algoritmo de Canny.

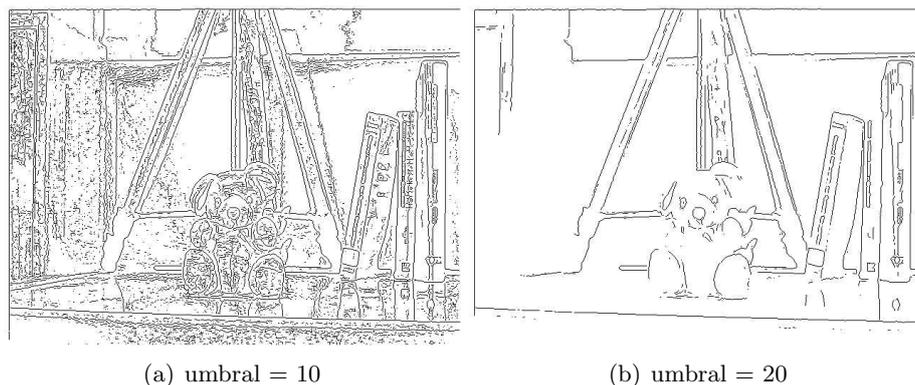


Figura 3.14: Imágenes de bordes con distintos umbrales

Este algoritmo se basa en parte en los anteriores pasos, puesto que se necesita, como se verá a continuación, información acerca del gradiente, a partir de alguna de las máscaras comentadas anteriormente. Además, este método no utiliza sólo la información de módulo, sino que también utiliza la de fase para optimizar el umbralizado posterior que se hace a la imagen de bordes, como anteriormente se ha visto, con un umbral fijo.

Este método consiste en varios pasos que se pasan a describir a continuación

1. Suavizado de la imagen con un filtro gaussiano, para no afectar a los bordes que queremos detectar.

Al ser el filtro gaussiano un filtro de media pero ponderando más la posición central de la máscara del filtro que el resto de posiciones. Con esto se consigue que la muestra actual pese más y por tanto, si hay un borde, este no se difumine tanto como con el filtro de media.

2. Se obtiene el gradiente para cada punto con cualquiera de los métodos vistos anteriormente (tanto módulo como fase, pues la fase se necesita para saber la dirección del gradiente en cada punto)
3. Supresión no máxima del gradiente, adelgazamiento de bordes mediante umbralizado con histéresis

La imagen de módulo umbralizada con un único umbral no es suficientemente buena para poder obtener los bordes, puesto que se suelen introducir máximos locales debido al ruido. Debido a esto, para obtener mejor los bordes se utiliza este método de adelgazamiento.

- a) Se fijan 2 umbrales $T_1, T_2, T_1 < T_2$
- b) Para todo u, v y recorriendo la imagen en un orden fijo, se realizan varias comprobaciones.
 - 1) Si es el valor del píxel es $>T_2$, se continúan las comprobaciones, si no se salta al siguiente paso.
 - 2) Si el valor del píxel vecino en la dirección del gradiente es mayor que él, se pone a 0, y si no, se deja como está para cuando se pase a evaluar sus vecinos y se verá si se tiene que poner a 0 o no.
- c) Una vez hemos adelgazado los bordes a un píxel de ancho al haber recorrido toda la imagen, pueden haber quedado componentes no conectadas. Para ello, utilizamos el

umbral T_1 , haciendo que las componentes conectadas sean las que, en la vecindad del píxel, superen este umbral.

Un inconveniente de este algoritmo es que en ocasiones, al ir eliminando algunos cuadros, puede que eliminen las esquinas o puntos de unión de múltiples segmentos. Si necesitamos detectarlas para otra sección de una aplicación, habrá que tener cuidado con este método, o utilizar otros métodos.

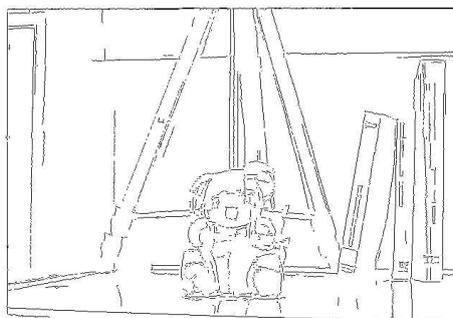


Figura 3.15: Resultado de aplicar Canny a la imagen anterior

Variando los umbrales descritos anteriormente y la varianza de la gaussiana utilizada para suavizar la imagen, podemos obtener distintos resultados. Si se aumenta el umbral inferior T_1 y se mantienen fijos los otros 2 parámetros, se descartan bordes que no son lo suficientemente conexos, esto es, que tienen demasiados cortes, con lo que quedan solo lo suficientemente remarcados. Si disminuimos el umbral superior, se relaja la condición para el procesamiento en el algoritmo, por lo que pasan bordes menos significativos y en la imagen final aparecerán más bordes. Por último, si aumentamos la varianza de la gaussiana utilizada para el filtrado previo, se eliminan aquellos bordes muy juntos o muy finos en la imagen, difuminándolos, con lo que solo quedan aquellos bordes lo suficientemente abruptos o significativos.

Existen otros métodos de obtención de bordes tan elaborados como el anterior, pero por ejemplo, en el caso de la Laplaciana o la Laplaciana de la gaussiana, este operador funciona con segunda derivada, lo cual hace que sea mucho más sensible al ruido, y por tanto, no sea comúnmente utilizado.

3.5. Detección de caras: Viola & Jones

Este detector fue desarrollado por Paul Viola y Michael Jones en 2001 [39]. Viene implementado directamente en OpenCV, y se caracteriza por una rapidez en la detección de objetos en la escena, por un alto índice de aciertos y un bajo porcentaje de falsos positivos. Éste se puede ver como un detector hacia un tipo de objetos muy específico, dependiente totalmente del conjunto de entrenamiento utilizado. El problema en el que más extendido ha sido el uso de este detector ha sido la detección de caras, aunque su uso es extensible a cualquier otro tipo de objetos, siempre que tengamos el conjunto de entrenamiento adecuado.

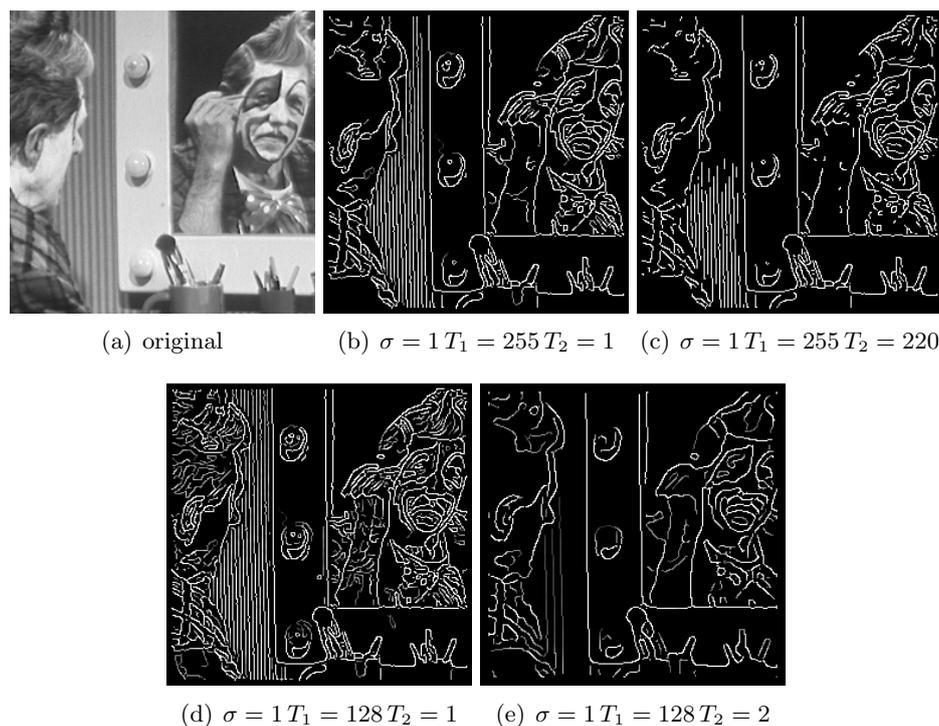


Figura 3.16: Ejemplos de variación de parámetros en el algoritmo de Canny

Pasamos a escribir las partes fundamentales de las que consta este algoritmo:

En primer lugar calculamos la imagen integral que permite la extracción rápida de características y esta operación consume en realidad muy pocas instrucciones máquina. Esta representación permite que la posterior extracción de características con independencia del tamaño del mismo y con tiempo constante.

La extracción de características se realiza mediante Haar Wavelets. Como toda transformada Wavelet [14], necesita de dos funciones, la generadora y la de escala. En el caso de este detector, esta función es bidimensional. La función generadora (unidimensional) normalizada está definida de la siguiente forma:

$$\psi(x) = \begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ -1 & \frac{1}{2} \leq x < 1 \end{cases} \quad x \in \mathfrak{R} \quad (3.23)$$

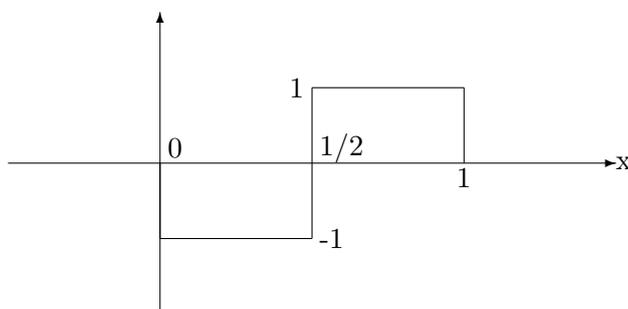


Figura 3.17: Función Haar unidimensional

Y la función de escala necesaria para ajustar la función base es la siguiente:

$$\phi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{resto} \end{cases} \quad (3.24)$$

Con estas dos ecuaciones tenemos definida prácticamente la transformada wavelet que se usa en este detector para la extracción de características.

En segundo lugar utilizamos AdaBoost (Adaptative Boosting) para quedarnos en cada comprobación con las características más importantes extraídas en cada etapa del algoritmo.

Por último agrupamos las características extraídas que nos devuelve la etapa anterior según la zona de la imagen y vamos clasificando éstas en cascada para poder seleccionar aquellas zonas de la imagen más prometedoras e ir eliminando aquellas que no tienen características relevantes.

Para poder extraer las características que nos servirán de patrón en el clasificador, debemos realizar un entrenamiento con un conjunto de imágenes adecuado, tanto con imágenes de los objetos a detectar como con varias imágenes que contengan o no este tipo de objetos, además para poder asegurar su correcto funcionamiento.

Primero se realiza una pasada de extracción de características por la porción de imagen y con un simple clasificador de características invariantes de 2 ramas, somos capaces de eliminar la mitad de las localizaciones posibles donde buscar posteriormente. Los tamaños de estas porciones de imagen son de 24x24.

A continuación particularizamos en las zonas donde hemos encontrado ciertas características y vamos clasificando características de la zona según criterios más complejos en cada etapa que en la anterior. Este proceso de detección y clasificación es similar a los árboles de decisión que podemos tener en otros algoritmos. Si durante una búsqueda una región no se ha rechazado, con una probabilidad muy alta, estará el objeto a detectar.

- Características

Las características a extraer son invariantes respecto al tamaño de la zona de búsqueda de la imagen. Las más comunes son las que se asemejan a los bordes (2 rectángulos con distintos valores de nivel de gris en las distintas posiciones posibles), de línea (3 rectángulos con una variación central suficiente en las distintas posiciones posibles y con distintos grosores de línea) y de contornos concéntricos (representados por una zona cuadrada dentro de la otra con las posibles rotaciones). Son posibles otras formas características de la imagen, pero en realidad no las mostramos porque en nuestro caso de detección de caras tampoco se utilizan. Las más utilizadas son las de borde y las de línea. Estas son algunas de las representaciones de las funciones Haar bidimensionales utilizadas.

- Imagen integral

Esta imagen es el resultado de sumar todos los píxeles desde la esquina superior izquierda (referencia de coordenadas). Puesto en término de ecuación:

$$I_{int}(x, y) = \sum_{i=0}^x \sum_{j=0}^y I_{orig}(i, j) \quad (3.25)$$

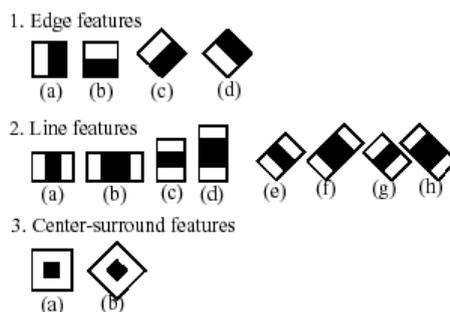


Figura 3.18: Características principales que comprueba el detector de Viola & Jones

Si se aplica un filtrado posterior a la imagen, éste se elegirá para que no deforme los bordes ni las líneas ni ninguna característica de la imagen contra la que se pueda comparar.

■ Clasificador

Una vez extraídas las características de la imagen, el número de éstas que podríamos llegar a manejar es bastante superior incluso al número de píxeles, por lo que se ha de encontrar una manera inteligente de clasificar y procesar este volumen de datos con agilidad. Para ello se emplea un entrenador débil para cada característica. Este entrenador es capaz de calcular el umbral óptimo para la clasificación, a través de las muestras positivas (en las que está el objeto que debería) y negativas (en las que no está).

Este clasificador puesto en forma matemática queda de la siguiente manera:

$$h_i(x) = \begin{cases} 1 & \text{si } p_i f_i(x) < p_i \theta_i \\ 0 & \text{en otro caso} \end{cases} \quad (3.26)$$

donde $h_i(x)$ es el clasificador en función de x , que es la imagen de 24x24 píxeles, p_i la paridad o la dirección de la inecuación, θ_i el umbral, y $f_j(x)$ la característica que estamos evaluando.

Las 2 primeras características seleccionadas por el clasificador AdaBoost son:

- la diferencia de niveles entre los ojos y la parte superior de los pómulos
- La diferencia de niveles entre los ojos y el puente formado por la nariz

■ Cascada de clasificadores

Por último se construye la cascada de clasificadores intentando optimizar el rechazo de los negativos y el rechazo de falsos positivos. Los clasificadores se encadenan y en cada etapa se rechaza o se acepta una imagen para continuar en la siguiente etapa. Si atraviesa todas las etapas existe una alta probabilidad de que en esa zona se encuentre el objeto requerido.

Cada etapa de la cascada de clasificadores se entrena haciendo que la tasa de falsos positivos y la tasa de detección se encuentren. Cuando se encadenan estas etapas, lo que ocurre es que la tasa de aciertos aumenta lo suficiente y la de falsos positivos desciende drásticamente.

Un ejemplo de las características en cascada que se toman para el caso de la detección de caras es el mostrado en la figura 3.19.

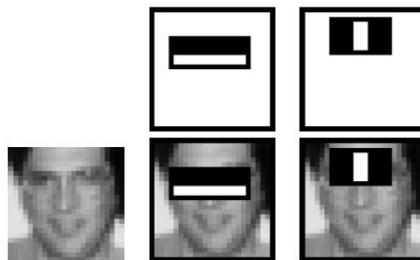


Figura 3.19: Primeras características seleccionadas por la cascada de detectores de Viola & Jones

Se puede ver en esta figura cómo primero se seleccionan las características que ocupan una mayor superficie de la imagen. Se observa cómo la primera de ellas es la diferencia de niveles entre la parte superior de los ojos y la parte inferior, o sea, entre la frente y la cuenca de los ojos, que es la que mayor superficie de la cara ocupa en realidad. Se puede observar también cómo la segunda de las características principales es capaz de detectar el tabique nasal, como diferencia de niveles entre la parte central y los laterales de la parte superior de la nariz.

3.6. Extracción de características

Existen multitud de métodos para la extracción de características de una imagen. Las características pueden venir dadas desde distintas magnitudes físicas, aunque al final todo se reduce a una serie de coeficientes recogidos en vectores o en matrices. Según el criterio que utilicemos para la extracción de características estaremos ante un algoritmo u otro, teniendo una gran variedad de métodos para caracterizar una imagen.

La mayoría de estas transformaciones tienen la capacidad de ser reversibles, esto es, que a partir de la imagen original se pueden obtener los coeficientes transformados, y a partir de éstos, se puede obtener la imagen original en una forma parecida. Al primer procedimiento se le suele llamar transformada directa y al último transformada inversa.

Cada una de estas técnicas tiene aplicaciones concretas, no siendo adecuada para el mismo problema todas las transformadas. A continuación se presentan, sin entrar en demasiada profundidad, algunas técnicas utilizadas en visión artificial para el problema reconocimiento de caras, a la hora de la extracción de características.

3.6.1. PCA (Principal Component Analysis)

Ahora pasamos a explicar el algoritmo que ha supuesto la base del que se ha utilizado en el sistema propuesto.

Este método, también llamado transformada de Hotelling, es ampliamente utilizado en el ámbito del reconocimiento facial, en una gran variedad de artículos y en la mayor parte de la bibliografía. Además es la base de muchos otros algoritmos de reconocimiento facial, como veremos más adelante.

Este algoritmo base se basa en lo siguiente:

Se tienen una serie de M imágenes de dimensión $N \times N$. Para procesarla mediante este método, se pone la imagen completa en forma de vector.

$$I_k = (I_{k1}, I_{k2}, \dots, I_{kN}, I_{k(N+1)}, \dots, I_{k2N}, \dots, I_{kN^2})^T \quad (3.27)$$

Por tanto, las M imágenes se podrán representar como vectores de N^2 componentes:

$$I_1 = \begin{pmatrix} I_{11} \\ I_{12} \\ \vdots \\ I_{1N^2} \end{pmatrix} \quad I_2 = \begin{pmatrix} I_{21} \\ I_{22} \\ \vdots \\ I_{2N^2} \end{pmatrix} \quad \dots \quad I_M = \begin{pmatrix} I_{M1} \\ I_{M2} \\ \vdots \\ I_{MN^2} \end{pmatrix} \quad (3.28)$$

A partir de aquí, se calcula el vector de componentes medio:

$$\psi = E[I] = \frac{1}{M} \sum_{i=1}^M I_k = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_{N^2} \end{pmatrix} = \begin{pmatrix} I_{11} + I_{21} + \dots + I_{M1} \\ I_{12} + I_{22} + \dots + I_{M2} \\ \vdots \\ I_{1N^2} + I_{2N^2} + \dots + I_{MN^2} \end{pmatrix} \quad (3.29)$$

Con esta imagen media, calculamos la diferencia con cada una de la serie de imágenes de entrenamiento:

$$\Phi_k = I_k - \psi = \begin{pmatrix} \Phi_{k1} \\ \Phi_{k2} \\ \vdots \\ \Phi_{kN^2} \end{pmatrix} = \begin{pmatrix} I_{k1} - \psi_1 \\ I_{k2} - \psi_2 \\ \vdots \\ I_{kN^2} - \psi_{N^2} \end{pmatrix} \quad (3.30)$$

Con esta imagen de diferencia, calculamos la matriz de covarianza:

$$C_{PCA} = \frac{1}{M} \sum_{k=1}^M \Phi_k \Phi_k^T = \frac{1}{M} A A^T \quad (3.31)$$

La matriz C_{PCA} es de dimensión $N^2 \times N^2$, y además es real y simétrica, por tanto, se pueden calcular los autovalores y autovectores asociados de la misma.

Los autovalores vienen dados por la solución de la ecuación

$$|\lambda I - C_{PCA}| = 0 \quad (3.32)$$

Asociados a los autovalores $\lambda_1, \lambda_2, \dots, \lambda_{N^2}$, se calculan los autovectores mediante la siguiente igualdad

$$C_{PCA} u_i = \lambda_i u_i, \quad |u_i| = 1 \quad (3.33)$$

Una vez calculados todos los autovalores y autovectores, se introducen éstos en la matriz U por orden creciente del valor del autovector. Esta matriz U es la que nos da la base del espacio transformado PCA.

$$\Omega = U^T A \quad (3.34)$$

Siendo Ω la matriz de características en el espacio transformado y A la imagen de origen (previa transformación en vector de la imagen).

Al ser la matriz U cuadrada, la matriz Ω , tiene las mismas dimensiones que la matriz A , esto es, $N^2 \times M$.

El nombre de Principal viene dado porque, la información fundamental de las características de la imagen está contenida en los autovalores de mayor valor. Si ponemos un umbral a estos autovalores, el algoritmo se queda únicamente con los autovectores asociados a los autovalores mayores, que son en realidad las componentes principales que dan nombre al algoritmo.

Ahora, si realizamos la correlación entre las matrices de características, podemos encontrar la siguiente relación:

$$C_{\Omega} = \frac{1}{M} \Omega \Omega^T = \frac{1}{M} U^T A A^T U = U^T C_{PCA} U \quad (3.35)$$

Esta matriz tiene la propiedad de que, al ser U una matriz diagonalizante de C_{PCA} , entonces la forma de la matriz C_{Ω} es diagonal y su diagonal está compuesta por los autovalores de C_{PCA} .

Ya hemos visto la transformación directa desde el espacio imagen al espacio transformado 3.34, ahora solo queda ver cómo realizar la transformación inversa:

Deshaciendo todas las transformaciones realizadas hasta ahora, vistas en las ecuaciones anteriores, y teniendo en cuenta que $U^{-1} = U^T$, se deduce que:

$$A = (U^T)^{-1} \Omega = U \Omega \quad (3.36)$$

Y finalmente I_i se obtiene como:

$$I_i = [Columna_i(A)] + \psi \quad (3.37)$$

Siendo esta A la recuperada mediante la ecuación 3.36.

En el caso de recuperar con menos autovalores o autovectores, no se tiene toda la información, y a veces en la reconstrucción, se encontrará algún problema, mezclando alguna de las imágenes.

Ejemplos del algoritmo aplicado a una serie de imágenes y el resultado de la recuperación, se presenta en la figura 3.20:

La base del espacio PCA se ha obtenido mediante las imágenes originales, esto es la primera columna de la tabla de la figura anterior. A partir de ella calculamos la matriz media, la matriz de covarianza, y los autovalores y autovectores. Generando las bases del espacio con uno o tres autovectores, calculamos la reconstrucción anteriormente descrita y observamos varios fenómenos que ocurren entre las imágenes reconstruidas y las originales.

En primer lugar, vemos que si solo utilizamos un autovector, la imagen reconstruida tiende a ser muy parecida a la imagen media, lo cual nos indica que con un solo autovector no tenemos información suficiente que diferencie una imagen de las demás. En la columna central, vemos las imágenes reconstruidas con 3 autovectores contenidos en la base. En este caso, vemos que las imágenes son muy parecidas a las originales, aunque en algunos casos se aprecian los efectos de partir de una imagen media. En algunos casos aparece la silueta de uno de las 2 personas que estaban anteriormente debido, como hemos dicho antes, al efecto de partir de una imagen media y no tener suficiente información. Además se pueden apreciar cambios en la iluminación

en algunas de las fotografías reconstruidas, sobre todo en el segundo caso, donde la iluminación que afecta a uno de los 2 es distinta que la de la imagen original.



Figura 3.20: Ejemplos de recuperación de las imágenes con 3 y con 1 autovector con PCA

3.6.2. 2DPCA (2 Dimensional PCA)

La diferencia de este algoritmo con respecto al PCA normal es que no es necesaria la vectorización inicial de la imagen, lo cual hace que la matriz de covarianza sea más fácilmente computable y por tanto los autovectores y autovalores obtenidos a partir de ésta, sea más fácilmente computables, con el consiguiente ahorro de tiempo. Uno de los inconvenientes, y no muy grave, es que para reconstruir una imagen con la suficiente calidad, se necesitan más autovectores que con PCA.

Como en la ecuación 3.34, con este algoritmo se pasa de un espacio imagen a un espacio transformado a partir de una matriz de transformación calculada a partir de la matriz de covarianza, a través de sus autovectores y autovalores.

$$Y = AX \quad (3.38)$$

Para obtener la mejor X que pueda representar las características de A en nuestro espacio, empezamos calculando la matriz de covarianza de todos los usuarios de nuestra base de datos:

$$\bar{A} = \frac{1}{N} \sum_{i=1}^N A_i \quad (3.39)$$

$$C = \frac{1}{N} \sum_{i=1}^N (A_i - \bar{A}) (A_i - \bar{A})^T \quad (3.40)$$

Siendo \bar{A} la imagen media, N el número de imágenes en nuestra base de datos, A_i la imagen i -ésima de la base de datos, y C la matriz de covarianza.

$$C = PQP^T \quad (3.41)$$

A partir de la matriz de covarianza, calculamos sus autovalores Q (matriz diagonal) y sus autovectores P (matriz de paso), los cuales utilizaremos para formar la base de nuestro espacio, como lo hacíamos anteriormente en PCA. La diferencia fundamental entre estos 2 métodos es el tamaño de la matriz de covarianza calculada. A diferencia de PCA, cuya matriz de covarianza tiene un tamaño de $N^2 \times N^2$, lo cual no es posible de encontrar fácilmente sino que hay que recurrir a métodos especiales del estilo de SVD, en 2DPCA se tiene una matriz cuyas dimensiones son de $N \times N$. Esta razón es la fundamental por la que se utiliza este método en nuestro sistema frente a otros, puesto que se necesita agilidad en el cálculo. En comparación con este método, con PCA se tiene una tasa de éxito menor que con 2DPCA. No tomamos todos los autovectores, sino que, como hemos visto antes tomamos los autovectores correspondientes con los primeros autovalores, los cuales son los más representativos, los que concentran la mayor parte de la energía de la foto, los que nos dará más información acerca de las características de las caras, como hemos visto antes en la figura 3.20, en la que con pocos autovectores se pueden recuperar los aspectos básicos de la imagen completa. Esta última matriz de autovectores seleccionados será nuestra matriz X . En el caso de 2DPCA, son necesarios más autovectores que en PCA, como podemos leer en el artículo original sobre este método [42].

Por último, calculamos las proyecciones de las imágenes de los conductores permitidos en la base del espacio obtenida anteriormente, como hemos explicado en la ecuación 3.38.

Se puede observar que este método no es invariante ante rotaciones y translaciones de la cara, condiciones de iluminación, variación de la apariencia ni variación de gestos, por lo que habrá que introducir suficientes variaciones en la base de datos para poder utilizar el algoritmo correctamente.

Existen otros métodos derivados de PCA como ICA, LDA, Kernel-PCA e IPCA, basados en otras hipótesis, los cuales no se van a explicar, puesto que exceden los objetivos de este TFC.

3.7. Algoritmo de clasificación: K-vecinos más cercanos

En la literatura existen muchos métodos para la clasificación de patrones y para su posterior comparación para la identificación de señales.

Estos sistemas ayudan a realizar tareas en las que hay que seleccionar, a partir de unas muestras, el conjunto o clase a la que pertenecen los objetos que hay a la entrada.

Para realizar esta labor de clasificación, existen varios métodos. Algunos de ellos se basan en la estadística, como el clasificador bayesiano, ML, EM, Correlación, etc. y otros basados en las regiones que generan las distintas clases en el espacio de decisión, como K-medias, redes neuronales, SVM, K-vecinos más cercanos, etc.

El que se va a utilizar en el sistema es la particularización para $k=1$ de los k -vecinos más cercanos, el cual se pasa a explicar a continuación.

Este algoritmo se basa en calcular la distancia a todos los vecinos, se queda con los K que menor distancia tienen. De estos, se evalúan las clases a las que pertenecen. La clase a la que pertenezcan más de los vecinos seleccionados será la clase seleccionada.

Su uso se justifica cuando solo los centros de las distribuciones no son representativos puesto que la varianza sea muy dispar. Cuanto mayor sea K menor será el error será menor, pero más tiempo de cómputo se necesitará.

Para $K=1$ se obtiene el clasificador del vecino más cercano, en el cuál solo contamos con el vecino que más cerca esté de la muestra de entrada.

3.8. Medidas de distancia

3.8.1. Distancia euclídea

Para tener una idea de la bondad de los resultados y para poder realizar comparaciones en el clasificador, se han de utilizar medidas de distancia, lo cual va a dar una idea de lo cerca que están una serie de características de otras, o lo que es lo mismo, de la diferencia entre las mismas.

La medida de distancia euclídea es la más clásica, básica y general, o sea, la más implementada cuando los problemas a resolver son sencillos o tienen características normales.

La distancia euclídea se define formalmente como el producto escalar de la diferencia de 2 vectores de posición (referidos al mismo origen de coordenadas), esto es, como la norma del vector diferencia de posición. Expresado matemáticamente, nos queda lo siguiente:

$$d(\vec{a}, \vec{b}) = \left(\langle \vec{a} - \vec{b}, \vec{a} - \vec{b} \rangle \right)^{\frac{1}{2}} = \left\| \vec{a} - \vec{b} \right\| = \sqrt{(a_1 - b_1)^2 + \dots + (a_n - b_n)^2} \quad (3.42)$$

A la hora de calcularla en el programa, utilizamos la siguiente ecuación derivada de la anterior

$$d_{euclidea} = \sum_{i=1}^w \sqrt{(a_i - b_i)^T (a_i - b_i)} \quad (3.43)$$

Siendo a_i y b_i las columnas i -ésimas de las matrices de características de la imagen de entrada y de la base de datos respectivamente, y w la anchura de la imagen.

En esta medida de distancia, todas las componentes o elementos de los vectores utilizados en la ecuación anterior, contribuyen de la misma forma a la medida final. Esto puede que, en ocasiones nos sea perjudicial, puesto que pueden haber componentes algo más significativas a la

hora de calcular el peso que otras componentes. Por tanto, este sistema de cálculo de distancias no es el óptimo, pero es el más homogéneo.

Para mejorar este tipo de medidas, se suelen utilizar matrices de pesos para cada una de las componentes de los vectores, que queden afectadas por distintos factores para ajustar la medida mejor a nuestros propósitos, o lo que es lo mismo, utilizar una medida de distancia euclídea ponderada. Existen diversas formas de obtener los pesos adecuados para ajustar la medida a las necesidades. Es posible obtener una estimación de esta matriz a mano, pero tenemos el inconveniente de que la dificultad que esto supone crece proporcionalmente con el número de componentes del vector y el número de muestras, por tanto, para nuestro caso, este método quedaría totalmente descartado.

Otras medidas de distancia posibles, sin recurrir a métodos estadísticos, son las de Chebychev y la de Minkowsky. La distancia de Chebychev se obtiene como el máximo de todos los términos de distancia, la cual no parece demasiado apta para nuestros propósitos, puesto que, en principio no nos vale solo como el máximo, pues existen muchos términos en los patrones y no podría dar una medida adecuada de la disparidad entre los patrones utilizados. La distancia de Minkowsky es la más general de todas estas distancias, puesto que para su medida se utiliza un parámetro λ que da el orden al que tenemos que elevar los términos y también el de la raíz que tenemos que hacer. Para $\lambda = 1$, tenemos la distancia de Manhattan (en bloques), para $\lambda = 2$ la distancia euclídea y para $\lambda = \infty$, la distancia de Chebychev.

Una de las mejores maneras posibles de hacer esta ponderación es realizarla estadísticamente, como veremos en el capítulo dedicado a las mejoras en el clasificador 3.8.2.

3.8.2. Medida de distancia de Mahalanobis

Este método de medida estadística fue descubierta por un matemático Indio llamado Prasanta Chandra Mahalanobis en 1936. Esta medida de distancia se basa en métodos estadísticos para calcular la mejor matriz de pesos posible en un contexto estadístico. Este método de cálculo de distancia se basa en la matriz de covarianza de los datos de entrada, de salida o de los dos. Con la matriz de covarianza calculada a partir de los datos, tenemos una idea de la varianza que va a tener cada una de las componentes de los vectores de entrada. Aquellas que tengan más varianza son aquellas que van a tener más incertidumbre, y por tanto, las medidas en las que menos se ha de confiar. Por tanto, la matriz de pesos que tenemos que utilizar para ponderar cada una de las componentes es la matriz inversa de la matriz de covarianza.

Para el cálculo del método de medida de distancia de Mahalanobis, se ha utilizado la siguiente ecuación:

$$d_{mahalanobis} = \sum_{i=1}^w \sqrt{(a_i - b_i)^T C^{-1} (a_i - b_i)} \quad (3.44)$$

Donde C es la matriz de covarianza de las imágenes de características de entrada al sistema, calculada de igual forma que en la matriz de covarianza para generar la base de datos. Si hacemos que la matriz C sea la identidad, la distancia de Mahalanobis se reduce a la distancia euclídea. En la siguiente ecuación, en la que se explica cómo calcular la matriz C , A_i es la matriz de características i -ésima y \bar{A} es la media de las matrices de características.

$$C = \frac{1}{N} \sum_{i=1}^N (A_i - \bar{A}) (A_i - \bar{A})^T \quad (3.45)$$

Se ha de tener cuidado a la hora de calcular esta matriz para no sea singular, ni cercana a serlo porque los valores de covarianza no se ajustarían a los que son correctos para el correcto cálculo de la distancia. Si la matriz de resultados es cercana a ser singular, el resultado que devolverá la distancia de Mahalanobis se puede desviar lo suficiente para confundir, en nuestro sistema, un usuario con otro, o que un usuario no válido pase a serlo.

También hay que tener en cuenta que las variables aleatorias empleadas, en este caso las imágenes de entrada, han de tener una distribución de probabilidad similar.

Esta última distancia mejorará los resultados como se verá en el apartado de resultados, pues estamos multiplicando las distintas componentes de distancia por un peso que viene dado por la matriz de covarianza, con lo que estamos adaptando mejor las distancias a la base de datos y al usuario de entrada.

Esta distancia ya ha sido utilizada para el propósito buscado en este sistema en el artículo [41], y en otra bibliografía específica sobre PCA [5], aunque el método de cálculo de esta distancia no ha sido exactamente igual que el utilizado en el sistema presentado, sino una versión simplificada.

3.9. Métodos de fusión de medidas: sistemas borrosos

Para saber exactamente ante qué tipo de imágenes estamos, no se puede confiar en un único parámetro, sino que hay que observar varios parámetros que podemos encontrar en la cámara. Estos pueden corresponder a la imagen capturada o a los parámetros de la cámara en el instante de captura, como la ganancia, el tiempo de exposición, u otros parámetros que sean dependientes del nivel de iluminación de la escena y por tanto nos puedan dar una idea de ante qué tipo de entorno se encuentra el sistema.

Para la fusión de la información de estas medidas, se ha de incluir cierta información que se conozca previamente para poder, sin mucho esfuerzo, realizar una detección correcta y, por tanto, se ha de seleccionar un sistema de fusión en el que se pueda incorporar el conocimiento del diseñador de manera sencilla.

Un tipo de sistema que permite incorporar el conocimiento del diseñador directamente son los sistemas borrosos, los cuales pasamos a explicar teóricamente, haciendo una breve introducción, y posteriormente entraremos en más detalle acerca del subsistema propuesto.

El conocimiento del diseñador se introduce a través de las funciones de pertenencia a conjuntos borrosos y a través de las reglas de inferencia borrosa, que son los pilares de este tipo de sistemas basados en lógica borrosa.

Como se ha dicho anteriormente, al poder aplicar intuitivamente el conocimiento del diseñador a este tipo de sistemas, son más flexibles a la hora del diseño, puesto que en ocasiones es intuitivo saber cuáles han de ser los valores límite sobre los que se definen los conjuntos borrosos, tanto a la entrada como a la salida.

3.9.1. Visión general

Este tipo de sistemas siguen funciones basadas en la lógica, con la diferencia de que extendemos el dominio de esta lógica con la modificación de las variables de entrada, las cuales dejarán de ser conjuntos que puedan valer únicamente 0 o 1, sino conjuntos que puedan abarcar todo el rango de valores entre los dos valores anteriores.

Esto último ha supuesto un aumento de la flexibilidad y facilidad para implementar sistemas

basados en reglas lógicas, lo cual ha extendido su uso a varios campos, uno de los más importantes es el campo del control automatizado, el cual, durante una época se vio revolucionado por el uso de este tipo de controladores.

Para hacer funcionar este tipo de sistemas se ha de aplicar el conocimiento del diseñador, a través de las reglas que son capaces de gobernar su comportamiento. El resto de etapas de los sistemas borrosos son prácticamente como cualquier otro sistema de control, salvo que, en principio, no se incluyen elementos temporales, como elementos de retardo, acumuladores, etc. Este tipo de sistemas se utilizan para realizar funciones no lineales que actúen sobre otros sistemas. Para implementar funciones lineales existen otros métodos, como los sistemas P (Proporcional), PI (Proporcional e Integral), PD (Proporcional y Derivativo) o PID (Proporcional, Integral y Derivativo) o sistemas basados en variables de estados, siendo el comportamiento de éstos definido por matrices.

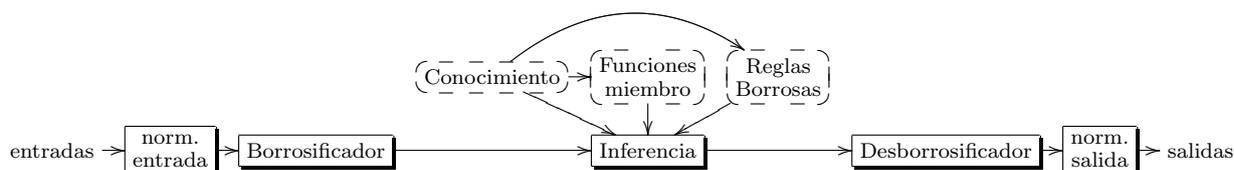


Figura 3.21: Esquema básico de un sistema borroso

El resto de bloques que componen el sistema son similares a los podemos encontrar en cualquier otro.

1. En primer lugar tendremos una o varias entradas, las cuales normalizaremos para adaptarlas al rango de entrada.
2. A continuación estas entradas hay que borrosificarlas para adaptarlas a la lógica que posteriormente utilizaremos en el sistema. El proceso de borrosificación consiste en pasar del dominio real a distintos niveles de pertenencia de los conjuntos borrosos de entrada.
3. Después de la borrosificación de las entradas se aplica el mecanismo de inferencia, por el cual, mediante antecedentes correspondientes a los conjuntos de entrada activados, y aplicando las reglas establecidas, se calculan los consecuentes de salida.
4. Una vez obtenidos los consecuentes de salida se desborrosifican para obtener la salida real.
5. Por último se desnormaliza la salida para adaptarla al rango necesario.

En los siguientes apartados describiremos con más detalle cada uno de estos puntos.

3.9.2. Normalización y desnormalización

A la hora de implementar un controlador borroso se puede hacer de dos formas bien distintas. Se puede implementar un controlador borroso en el cual, el rango de variables de entrada esté contenido en un margen reducido o que el rango abarque el rango completo de la salida. Si estamos en el primer caso, se ha de adaptar el rango de valores de la entrada real al rango de las

entradas borrosas. Estaremos en una situación parecida si hablamos de las variables de salida, puesto que se puede tener definido el rango de salida en un margen que no sea el adecuado para nuestros propósitos, por lo que se tendrá que adaptar de alguna forma a la salida.

Existen dos maneras de realizar esta adaptación, una por métodos lineales y la otra a través de funciones no lineales.

Para el caso de las funciones lineales los parámetros a ajustar son la constante de proporcionalidad y la componente continua, mientras que para el caso de las funciones no lineales, tenemos funciones basadas en logaritmos o exponenciales, parábolas o radicales, funciones trigonométricas directas o inversas que pueden adaptar el rango si las variables de entrada tienen un rango que se desee comprimir o expandir en ciertas zonas.

3.9.3. Conjuntos borrosos

Las funciones comúnmente utilizadas para los conjuntos borrosos, tanto de entrada como de salida, son las siguientes:

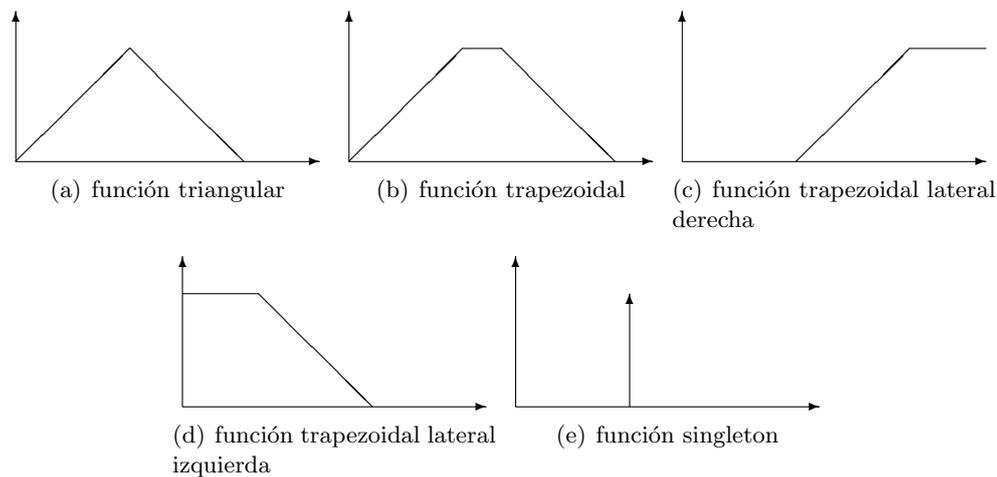


Figura 3.22: Funciones básicas borrosas

Existen multitud más de funciones posibles, como las sigmoideas, gaussianas, y otras, pero no son tan comunes a la hora de utilizarlas en un sistema real, puesto que son más difíciles de computar y por tanto consumen bastante tiempo, lo cual es perjudicial en un sistema de tiempo real, aunque este tipo de funciones proporcionan mayor estabilidad entre otras cosas.

3.9.4. Reglas y operadores

Las reglas se pueden definir como se hace en la lógica booleana, mediante proposiciones, y operadores, para formar los antecedentes y los consecuentes. Los operadores lógicos and or y not se han de redefinir para adaptarlos a las nuevas circunstancias, al tener que trabajar con valores no discretos:

and existen principalmente dos métodos para definir este operador binario: mínimo y producto de los grados de activación de las funciones de pertenencia.

or análogamente, tenemos otras dos formas para definirlo: máximo o suma algebraica ($a+b-ab$, siendo a y b los grados de activación de las funciones de pertenencia que acompañan al operador).

not en este caso, como en la lógica booleana, el operador es unario, y para el caso de la lógica borrosa solo hay una forma de definirlo, la cual es $1-a$.

Dependiendo del tipo de sistema que se implemente o de las necesidades del mismo, se utilizarán máximos y mínimos (sistemas rápidos, empotrados, de tiempo real), o productos y sumas (sistemas precisos, exactos donde ni la potencia ni el tiempo de cálculo son un problema)

Las reglas tienen la forma de las reglas utilizadas en la lógica habitual:

Si A es A_1 y B es B_1 , **entonces** C es C_1

Podemos apreciar que todo lo que va antes de entonces, es el antecedente y lo que va después el consecuente, como en una implicación normal. Los operadores and y or aplicados a la salida equivalen a varias reglas, con el mismo antecedente que afectan a variables de salida diferentes, dicho de otro modo, desglosamos el consecuente en varias reglas que afectan cada una a una variable distinta de salida.

3.9.5. Borrosificación y desborrosificación

La borrosificación transforma una variable real en conjuntos borrosos activados a niveles diferentes en función de los valores de la variable entrada que se han introducido.

La desborrosificación realiza la transformación contraria, esto es, pondera los conjuntos borrosos de salida y calcula el valor de la variable de salida real.

Para la borrosificación tenemos básicamente 2 posibilidades, que la función que borrosifique las variables reales de entrada sea un singleton o delta, o que sea una gaussiana o una exponencial cuyo centro sea el valor de la variable en ese instante. Lo más normal que se utiliza para borrosificar es la delta, puesto que con ella es fácil determinar el nivel de activación de los conjuntos borrosos de entrada, y además solo se tendrá 1 único valor por cada variable, mientras que si utilizamos el otro, tenemos una gaussiana multiplicada por cada conjunto de entrada, lo cual es bastante más complicado de construir.

Para la desborrosificación existen varios métodos. Dependiendo de en qué tipo de sistema borroso utilicemos o cómo queramos su comportamiento, existen varias alternativas:

- Centro del área

$$y_{CoA} = \frac{\int_{support(D)} \mu_D(x) x dx}{\int_{support(D)} \mu_D(x) dx}$$

- Centro de gravedad

$$y_{CoG} = \frac{\sum_i b_i \int_{support(D_i)} \mu_{D_i}(x) dx}{\sum_i \int_{support(D_i)} \mu_{D_i}(x) dx}$$

- Media del máximo

$$y_{MoM} = \frac{\int_{core(D)} \mu_D(x) x dx}{\int_{core(D)} \mu_D(x) dx}$$

- Bisector del área

En este caso Y_{BoA} se extrae de la solución a la siguiente ecuación

$$\int_{min(support(D))}^{Y_{BoA}} \mu_D(x) dx = \int_{Y_{BoA}}^{max(support(D))} \mu_D(x) dx$$

- Centros ponderados (sugeno)

$$y_{CP} = \frac{\sum b_i \omega_i}{\sum \omega_i}$$

3.9.6. Tipos de sistemas borrosos

En cuanto a los tipos de sistemas borrosos, dependiendo de cómo calculen su salida tenemos 2 tipos:

- Mamdani

Este sistema se caracteriza por tener a la salida del bloque de inferencia conjuntos borrosos, que luego mediante métodos de desborrosificación se convertirán en salidas no borrosas.

Este sistema es más estable frente a cambios en la entrada, pero es más lento pues el número de cálculos para obtener la salida real es mayor que el que veremos a continuación.

El método de desborrosificación utilizado en este tipo de controlador es el de centro del área o el de gravedad.

- Sugeno

Este método transforma directamente la salida del bloque de inferencia a una salida real, no de conjuntos borrosos. Esto agiliza el cálculo de la salida al eliminar uno de los bloques del sistema básico, pero a cambio el sistema, es más inestable.

Existen varios tipos de controlador sugeno, siendo el más común el de tipo cero, el cual tiene a su salida variables de tipo singleton, por lo cual, su salida será directamente una combinación lineal de las entradas. Los controladores sugeno de tipo uno tienen a su salida funciones lineales, no solo deltas, lo cual hace que no sea tan utilizado como el primero.

El método utilizado para la inferencia es el de centros ponderados, el cual únicamente utiliza los centros de las funciones de salida y los grados de activación de los antecedentes de las reglas activadas.

Capítulo 4

Arquitectura del sistema de partida

4.1. Introducción

El sistema de reconocimiento facial del que se parte, tiene las partes que se muestran a continuación:

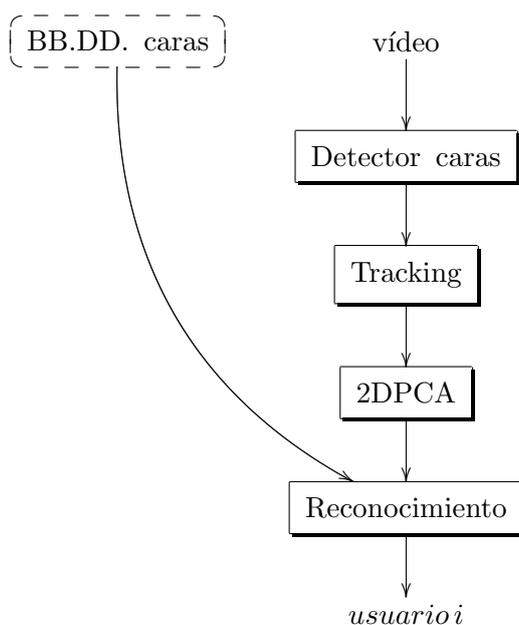


Figura 4.1: Esquema principal del sistema

En ella podemos ver que está formado por dos partes. Por un lado, tenemos la base de datos y por otro el software de reconocimiento.

En este software de reconocimiento, la base de datos faciales se carga al inicio del programa, pero no se utiliza la información hasta que no se realiza la comprobación en el clasificador. En la base de datos, mediante un programa que la genera, se calculan las proyecciones en el espacio 2DPCA de las fotos de caras introducidas por cada usuario permitido, por tanto en la carga, lo único que tenemos que hacer es ir cargando los archivos de características o proyecciones.

Uno de los subsistemas más importantes que encontramos en el diagrama de bloques es el detector de caras, basado en un algoritmo rápido, en el que, con pocas instrucciones aplicadas a una imagen reducida, es capaz de obtener si existe en la imagen una cara o no. Por supuesto, este clasificador ha de ser entrenado convenientemente, pero éste no ha sido nuestro trabajo, puesto que la suite de visión computacional empleada ya incluye unos archivos con los umbrales de cada una de las características a extraer, detectar y decidir.

Otro subsistema, que estabiliza en parte el funcionamiento del sistema es el tracking, en el cual, realizamos el seguimiento de la cara y las correcciones de coordenadas correspondientes, para adaptar a la aplicación propuesta las coordenadas devueltas por el detector.

El siguiente subsistema más importante del software que se incluye es el de extracción de características, mediante el algoritmo 2DPCA. Se ha seleccionado este algoritmo por su velocidad y su sencillez, además de por una alta tasa de acierto.

Por último, para poder decidir quién es el usuario que está ante la cámara es necesario implementar un clasificador, basado en distancia euclídea y en el método de búsqueda del vecino más cercano.

Todos los bloques que se han mostrado en la figura anterior, se pasan a explicar con más detalle a continuación.

4.2. Detección de caras

Ahora pasamos a ver la implementación de este detector de Viola & Jones en el sistema que se presenta.

No se ha generado un conjunto de entrenamiento puesto que en OpenCV ya vienen dado los parámetros de distintos detectores ya entrenados, para caras frontales, laterales, para cuerpo entero, etc.

El resultado de aplicar el algoritmo de detección a las imágenes de entrada lo mostramos en la figura 4.2.



Figura 4.2: Resultado de aplicar Viola & Jones en este sistema

El resultado devuelto por el detector de Viola & Jones queda representado por el recuadro rojo en ambos casos. Este cuadro, nos sirve de área reducida de búsqueda posteriormente para no tener que evaluar toda la imagen, sino un área más reducida, con la consiguiente reducción del tiempo de cómputo.

La línea interior (blanca en la subfigura 4.2.a y verde en la subfigura 4.2.b) se corresponde con la zona reducida de la cara, la cual va a ser la zona que utilizemos para el reconocimiento. El método de cálculo de este área de interés interna se explicará posteriormente en la descripción del tracking 4.3.

En la segunda secuencia de vídeo, correspondiente a la imagen de la subfigura 4.2.b, debido al contraste de luces y sombras, puesto que la iluminación de la escena proviene de la ventana, provoca que este detector falle un número considerable de veces y no se llegue a realizar una función de tracking correcta. En el caso de la subfigura 4.2.a el detector funciona mejor, puesto que se tiene una imagen con una iluminación uniforme, donde la luz viene desde arriba e ilumina mejor la escena global.

Este algoritmo, es capaz de ejecutarse en tiempo real. Se ha llegado hasta 30 fps en aquellos vídeos que se han grabado a esta frecuencia, ejecutado en un hardware de doble núcleo.

La implementación del mismo ha sido modificada en OpenCV con las modificaciones propuestas en el paper [25], para extender el tipo de características que es capaz de detectar con este detector e intentar realizar las detecciones más rápido.

En ocasiones, este detector presenta el inconveniente de los falsos positivos, lo cual hace que la imagen devuelta por el detector no sea la imagen de una cara. Este inconveniente no es importante mientras estos falsos positivos no sean en varias imágenes consecutivas y éstos no se den durante un reconocimiento, una vez pasada la fase de tracking. Si este caso se da, se hará el reconocimiento sobre la imagen de la zona cuyas coordenadas haya devuelto el detector, por lo cual el reconocimiento fallará.

4.3. Tracking

El algoritmo de detección de caras, trabaja en su implementación, con imágenes reducidas por varias razones, entre ellas para incrementar la velocidad de ejecución (de ahí que se pueda ejecutar en tiempo real, como se ha dicho antes). Las coordenadas que nos devuelve éste son respecto de la imagen que se ha pasado, luego se necesita escalarla y adaptarla con los desplazamientos verticales correspondientes a las necesidades de recorte de la cara que se requieren en la aplicación.

Como fuente predeterminada de posición de la cara detectada se utiliza el algoritmo de detección antes descrito, pero sí el algoritmo pierde en alguna imagen la cara, la máquina de estados global retrocede algunos estados según van pasando N imágenes sin caras detectadas (N prefijado experimentalmente). En el caso de que haya suficientes fotogramas sin una cara, el sistema asume que la cara se ha perdido y volverá al estado inicial de la máquina.

Si se ha encontrado una cara en la imagen anterior, utilizamos las coordenadas del anterior, puesto que se asume que la cara en un corto espacio de tiempo no se mueve, por tanto no realizamos la corrección hasta que la posición cambia lo suficiente como para tener que calcularla de nuevo. Si se ha alejado lo suficiente de la anterior detección, se toman las coordenadas del detector y se corrigen escalándola y dando el offset vertical adecuado, en nuestro caso de 20 píxeles en una imagen de 320x240. El factor de escalado, para adecuar las imágenes al detector es de 2. El diagrama de bloques correspondiente a esta función de tracking se presenta en la figura 4.3 y la representación gráfica del escalado realizado se muestra en la 4.4.

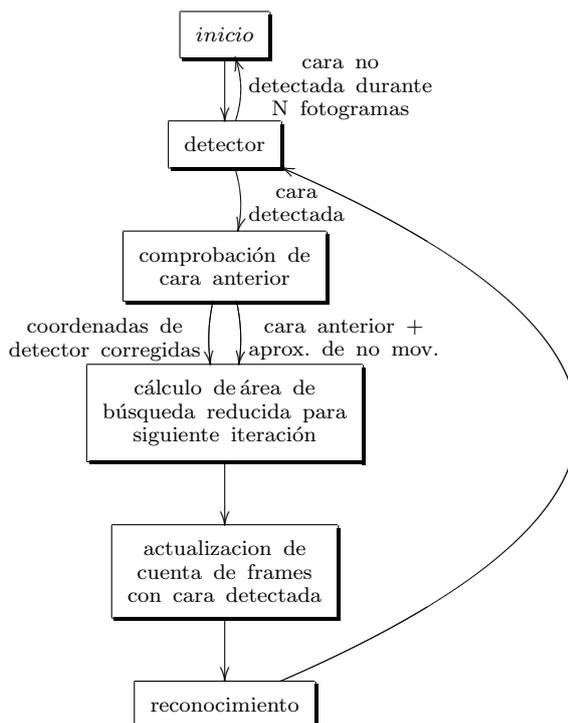


Figura 4.3: Funcionamiento de la máquina de estados del sistema

Una vez se han calculado y corregido las coordenadas, se recalcula el área de interés a partir del rectángulo obtenido anteriormente para enviar al detector una imagen reducida y para tratar también de introducir una única cara, la del conductor, que es la que interesa reconocer. Esta selección del área de interés se realizará en función de la posición de la cara detectada dentro de la imagen, pues se ha de tener cuidado con si se ha detectado demasiado cerca de los bordes de la imagen o si se tiene unas dimensiones adecuadas.

Por último, una vez se ha detectado la cara, se ha realizado su seguimiento según lo explicado anteriormente, se envía al módulo de reconocimiento para la extracción de características y su comparación, según sea necesario.

4.4. Extracción de características: 2DPCA

Como se ha dicho en el capítulo de Técnicas utilizadas, este algoritmo está basado en PCA, a la hora de caracterizar patrones para poder clasificarlos, reconstruir imágenes, comprimir las mismas u otros propósitos. Este algoritmo, propuesto en 2004 por Jian Jang y otros [42], pretende aumentar la eficiencia y la tasa de reconocimiento de PCA modificando algunos de los aspectos del anterior.

Uno de los aspectos fundamentales que cambia de este algoritmo con relación al PCA es que en 2DPCA no es necesario transformar una imagen bidimensional en un vector de dimensión $1 \times N^2$, con lo cual, no se reduce el tamaño, pero sí un paso previo, y además, se reducirá la dimensión de la matriz de covarianza a calcular para obtener la base de autovectores, aumentando la compresión de la información y por tanto el número de componentes necesario para realizar

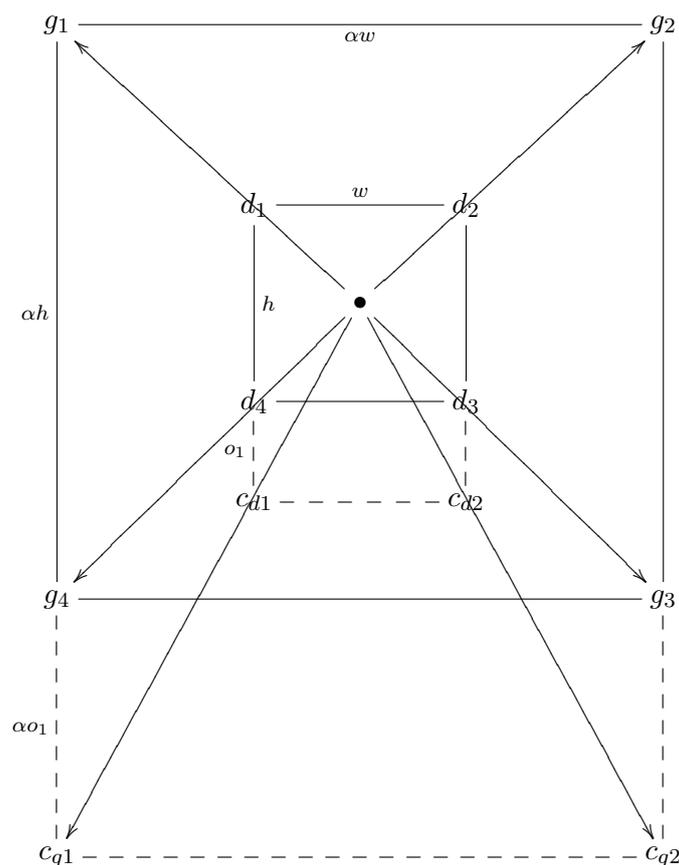


Figura 4.4: Escalado implementado y corrección de coordenadas

un reconocimiento facial correcto. Además, el hecho de no tener que vectorizar la imagen, hace mucho más sencilla la manipulación posterior.

En este caso se ha optado por no seleccionar todos los posibles autovectores, como se ha dicho anteriormente, sino tomar los 35 primeros. Estos son suficientes para tener tasas de reconocimiento más que aceptables, a la vista de las pruebas realizadas en [27]. En este trabajo podemos ver cómo con estos autovectores somos capaces de tener tasas de reconocimiento superiores al 90 %, lo que para nuestra aplicación es suficiente. Como se ha dicho antes, la iluminación va a jugar un papel importante a la hora de aumentar o reducir considerablemente este porcentaje de aciertos.

Además, hay que tener en cuenta que las imágenes utilizadas para el reconocimiento son distintas de las que hay en la base de datos, con lo que las tasas de acierto serán diferentes de las incluidas en los papers, puesto que en estos se utilizan las mismas imágenes tanto para reconocimiento como para generar la base. En este sistema se tiene un conjunto reducido de imágenes de muestra, tomadas desde secuencias de entrenamiento de entrada. Con ellas se calculan las características con las que se van a comparar posteriormente las muestras del candidato de entrada.

Para generar la base de datos se han ideado dos sistemas: uno que toma muestras automáticamente para la base de datos periódicamente, y un generador de la base de datos a partir de las muestras de entrada.

Una vez se tiene estabilizada la cara, como en el sistema principal, se procede a la toma de N muestras periódicas cada T segundos, para almacenar en la base de datos. En la implementación

inicial se tomaban 10 muestras cada 5 segundos y en la final 30 muestras con el mismo período.

En el segundo sistema, se recogen la información generada de cada usuario por el primer programa, y se procede al cálculo de todas las matrices descritas en la sección que describe el algoritmo. La base del espacio 2DPCA, junto con las proyecciones de cada una de las imágenes, son guardadas en la base de datos y son cargadas en el arranque del programa para realizar las posteriores comparaciones con el clasificador.

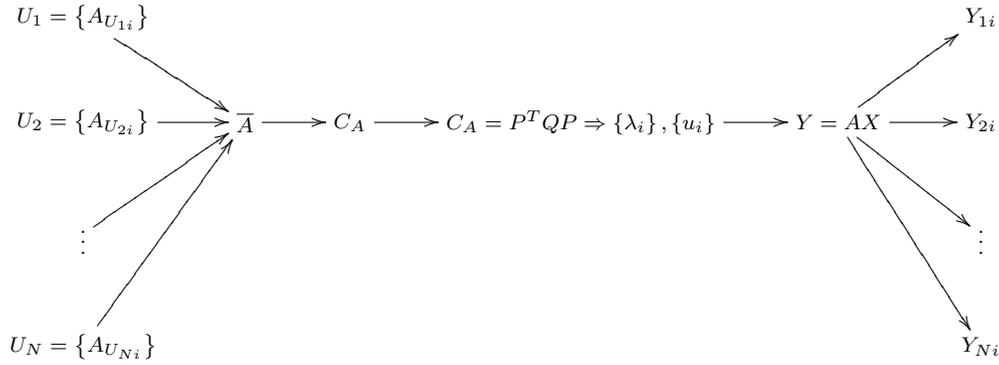


Figura 4.5: Esquema implementado en el entrenador de la base de datos

Los datos a tratar en el sistema serán, en lugar de vectores unidimensionales como en PCA, matrices de 2 dimensiones, como se define en el algoritmo 2DPCA, por tanto, la forma de definir la distancia se hará en función de la suma de todo lo obtenido en cada fila o en cada columna, no es la simple norma de la resta de dos vectores.

Antes de continuar, decir que las imágenes incluidas en la base de datos, son dependientes del método de tratamiento de la iluminación utilizado en cada caso, esto es, las imágenes incluidas son las que se han tratado con el método de independización de la iluminación que se haya probado. Se ha implementado de esta manera para tener compatibilidad absoluta con el programa generador de la base de datos. De esta forma, el programa que genera la base de datos es siempre el mismo, solo hay que cambiar el archivo de configuración de la base de datos y las que están en las carpetas de los usuarios. También la estructura de la base de datos y esta vez sí, el programa que lo genera varía ligeramente cuando se están haciendo pruebas con el sistema que es capaz de reconocer usuarios tanto de día como de noche.

En el programa de reconocimiento, se calcula la proyección de las imágenes de entrada multiplicándolas por base del espacio. Con estas proyecciones, se calcularán las distancias respecto del resto de proyecciones de la base de datos y, con el clasificador, se decidirá cuales de todas las muestras de la base son las más cercanas a las muestras de entrada.

4.5. Clasificador de vecino más cercano con criterio de mayoría

Tomando la distancia euclídea como punto de partida, y calculando la distancia de la proyección de la muestra de entrada a la de cada una de las proyecciones en el espacio 2DPCA de la base de datos, evaluamos cuál de todas es la que menor distancia tiene. Teniendo esta distancia mínima, podemos saber cuál de todas las muestras contenidas en la base de datos es la corres-

pondiente a esa distancia, y por tanto, el usuario dentro de la base de datos más parecido a la misma.

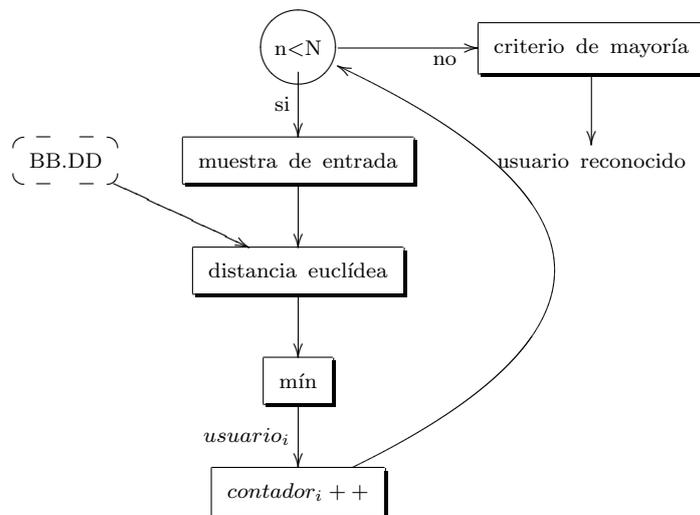


Figura 4.6: Esquema del clasificador del sistema inicial

Se considera también que un único reconocimiento no nos proporciona la suficiente información como para poder tomar una decisión correcta, con lo que repetimos este procedimiento N veces (20 en el sistema implementado) y utilizando un criterio de mayoría seleccionamos el usuario de todos los que se incluyen en la base de datos que más veces ha sido votado con las distintas muestras.

En este sistema se asume que todos los usuarios que entrarán en el coche, serán los que estén en la base de datos, pero en la realidad, por supuesto, esto no será así, por tanto, este clasificador lo que nos dirá es, dentro de la base de datos, cuál es el que parece más a la cara de entrada, aunque su parecido sea mínimo. Este problema se tratará de resolver en el capítulo siguiente, en el cual se propone un método para rechazar a usuarios externos a la base de datos.

4.6. Problemas asociados al sistema de partida

Los problemas fundamentales que se han encontrado en este sistema son varios.

1. El sistema no es capaz de discernir si el usuario que está enfrente de la cámara, está incluido en la base de datos. Lo que hacía en realidad era decir cuál de todos los usuarios que se encontraban en el sistema es el que más se parece al que se encuentra delante de la cámara.
2. Se comprobó que la iluminación es un problema muy importante a tener en cuenta, pues puede hacer que el sistema se confunda.
3. El hecho de que un usuario quiera conducir con unas gafas de sol, provoca un cambio de apariencia en la imagen de la cara muy grande, por lo que puede hacer que el sistema se confunda o se puede introducir un usuario externo.

4. El sistema no estaba preparado para hacer una detección en condiciones de iluminación nocturna, puesto que cuando se desarrolló, todavía no estaba disponible el sistema de visión nocturna creado por el grupo.
5. Existen ciertos problemas para el tracking debidos al funcionamiento intrínseco del detector y siendo éstos debidos a la iluminación. Este es un problema muy difícil de resolver, puesto que el detector está diseñado para imágenes nocturnas y no para imágenes con ciertos tratamientos para la iluminación. Por tanto, las mejoras incluidas en la iluminación que se verán no se pueden aplicar directamente para mejorar la tasa de aciertos del detector.
6. Si la apariencia del usuario varía considerablemente, debido a que se ha dejado barba, o se ha cambiado el corte de pelo, u otro cambio de apariencia similar, el sistema puede equivocarse al estar basado en un algoritmo de extracción de características por apariencia.

En este documento no se han podido resolver todos los problemas que se presentan, pero sí algunos de ellos y en otros se ha intentado hacer mejoras. En los próximos capítulos se expondrá el problema concreto, cómo afecta al sistema y los métodos que se han planteado para intentar resolverlo.

Capítulo 5

Arquitectura del sistema propuesto

5.1. Descripción general del sistema de reconocimiento

En este capítulo mostraremos el funcionamiento del sistema completo a través del diagrama de bloques y después comentaremos brevemente cada uno de los mismos para que quede claro su funcionamiento. La explicación más amplia y detallada de algunos de los bloques, se realizará en los apartados siguientes dentro de este capítulo.

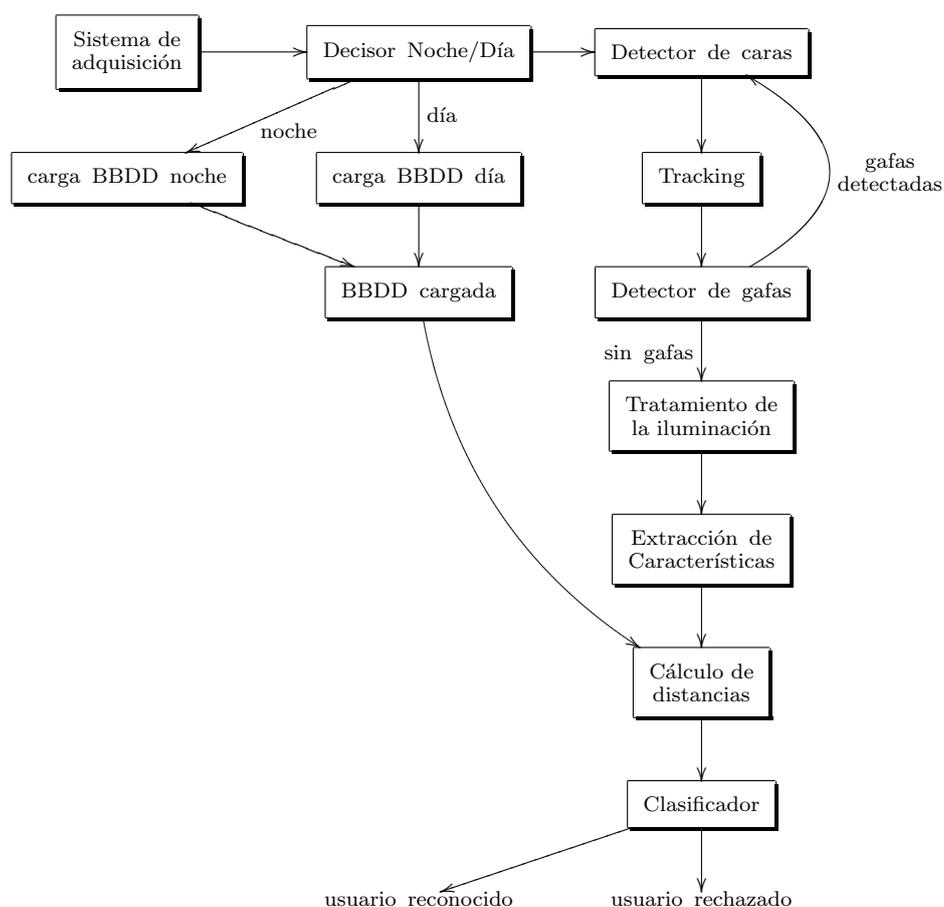


Figura 5.1: Esquema principal del sistema

A la vista del esquema podemos observar claramente 2 ramas: por un lado la carga de la base de datos adecuada, y por otro lado, la detección de caras y gafas, el tratamiento y el proceso de las imágenes de entrada para poder realizar la comparación.

Desde la cámara se capturan las imágenes de entrada, tanto para las imágenes de entrenamiento como para el sistema de salida. La cámara funcionará bajo el bus Firewire (IEEE 1394), por compatibilidad con la plataforma OpenCV de desarrollo de programas de visión artificial, y por la facilidad de uso de los drivers que se emplean bajo Linux. Además, este bus es más fiable y más veloz (en el caso de utilizar múltiples cámaras) que el bus USB. Sobre las particularidades de este sistema de adquisición se podrá encontrar más información en el Apéndice B.2.

La cámara utilizada para las primeras pruebas ha sido una Unibrain fire-i. Esta cámara funciona correctamente en condiciones diurnas, pero al tener un filtro de infrarrojos a la entrada del CCD, la hace poco sensible a este tipo de iluminación. En la implementación final de la cámara, se utilizará una Bastler Scout, la cual tiene múltiples modos, entre ellos diurno a 3 colores y nocturno, además de esta última se pueden extraer correctamente todos los parámetros utilizados por la cámara en cada instante de tiempo, sus sensores son menos ruidosos y su sensibilidad al infrarrojo es superior. Para la obtención de imágenes nocturnas se utilizará un iluminador de luz infrarroja, desarrollado para otros sistemas creados por el departamento, en el que se utilizan diodos infrarrojos de tipo OPE5685, dispuestos en forma circular con dos ramas y con 5 diodos por cada lado del círculo, para repartir la tensión suficiente entre las 2 ramas. Esta corona de diodos, será controlada, tanto en iluminación como en temporización del encendido y apagado, por una tarjeta de sincronismo también desarrollada por el departamento. Más detalles sobre este sistema de adquisición se pueden encontrar en el pliego de condiciones.

El resto del hardware necesario para la implementación del sistema es un ordenador con las características suficientes para poder ejecutar el software que se proporciona en el proyecto. En un sistema monoprocesador, el hecho de ejecutar el sistema operativo, el programa, las librerías y todo el demás software necesario, hace que en vídeos a 30 fps el procesado realizado por Viola & Jones y el resto de procesados posteriores, no sea suficiente y el sistema tarde algo más, no siendo por tanto posible la correcta ejecución en tiempo real. Sin embargo, pruebas realizadas en un ordenador con doble núcleo y una frecuencia de 2.4 GHz, el sistema ha sido capaz de ejecutarse en tiempo real con menos una carga en los 2 procesadores en torno al 60% (con una correcta repartición de la carga realizada por el kernel de Linux, incluido en las versiones de ubuntu 7.10 y 8.04).

Después del sistema de adquisición de datos, en nuestro caso las imágenes desde la cámara, tenemos ya la etapa software, la cual dividimos en varios subsistemas. En primer lugar, se ha de detectar si la imagen de entrada corresponde con una imagen de infrarrojos o con una imagen diurna. Si se ha detectado que se está en condiciones de iluminación nocturna, se ha de encender el iluminador de infrarrojos e iluminar la escena adecuadamente. En cambio, si las condiciones de iluminación son diurnas, se pasa directamente a las siguientes fases, eso sí, previa carga de la base de datos correspondiente a imágenes nocturnas o diurnas. Este subsistema se explicará detalladamente en el apartado correspondiente de este capítulo.

A continuación se procederá a la detección de caras como se describió en la implementación inicial del sistema, puesto que ya tenemos todos los datos necesarios para el reconocimiento cargados a la espera de encontrar al usuario del coche.

Posteriormente se realizará el tracking de esta cara, para evitar falsos positivos del detector de caras, y para, a la hora de realizar un reconocimiento, tener una imagen estable de la cara.

El siguiente paso trata de evitar que la apariencia de la cara cambie sustancialmente si el sujeto lleva puestas gafas de sol, por lo que se introducirá un detector de gafas de sol (las gafas normales no alteran sustancialmente la apariencia si los soportes no son gruesos, e incluso, si los bordes de las gafas son lo suficientemente gruesos este sistema sería capaz de detectarlas), para hacer que si el usuario las lleva, se las quite, y realizar un correcto reconocimiento facial.

A continuación, una vez se tiene una imagen estable de la cara, se pasa a la extracción de características de las muestras adquiridas, mediante la proyección en el espacio 2DPCA calculado en la base de datos (y cargado al inicio del programa, junto con las proyecciones de las muestras incluidas en la base de datos).

Por último se realiza, mediante el clasificador implementado, la comprobación de si un usuario se encuentra o no en la base de datos y si ese usuario está incluido, cuál de los que está es el más parecido.

Si el reconocimiento ha sido positivo, el programa sacará por pantalla la identidad del mismo y el porcentaje de confianza que se tiene sobre el reconocimiento realizado. Si el usuario ha sido rechazado, la máquina de estados se reiniciará cuando el usuario salga de escena, pues el sistema de tracking seguirá funcionando independientemente.

5.2. Rechazo de candidatos externos

5.2.1. Introducción

El primer problema importante que se detectó en el sistema base con el que se empezó a trabajar fue que no era capaz de rechazar por sí solo usuarios no pertenecientes a la base de datos. Al estar implementado en el sistema un clasificador mediante el vecino más cercano, sólo era capaz de decir cuál de todas las fotos que se tienen en la base se parece más a la que se introduce en la entrada.

A partir de ahora, al estar hablando de sistemas de detección, se hará referencia a las matrices de características como puntos en el espacio transformado, al conjunto de puntos que pertenecen a un usuario como la constelación del usuario, y al conjunto de puntos de todos los usuarios como el universo de decisión, para simplificar y agilizar el lenguaje utilizado en este capítulo.

La necesidad de la inclusión de este subsistema de decisión en el sistema global viene motivada principalmente por dos razones.

En primer lugar, se necesita poder elegir, dentro de una serie de usuarios permitidos, cuál de todos es el que está enfrente. Esto se necesitará posteriormente para tener perfiles de usuario y para adaptar el coche a las necesidades del usuario autorizado a conducir.

La segunda función, pero fundamental para el sistema de seguridad, es la capacidad de este sistema de rechazar a usuarios que no pertenecen a la base de datos. Esta función no se puede realizar con facilidad a través de algoritmos de agrupamiento y clasificación normales, puesto que este problema no se ha encontrado resuelto en la bibliografía consultada, con lo que hubo que realizar un estudio estadístico acerca de las distancias obtenidas. Se descubrió empíricamente la propiedad de que las distancias de usuarios externos a la base de datos son mayores que las de los usuarios internos a la base (sobre todo utilizando distancia de Mahalanobis). Se utilizará esta propiedad para organizar un clasificador adecuado a las necesidades que tenemos de rechazo de

usuarios.

Es importante señalar que en el sistema que se presenta, pesa mucho más el hecho de que se acepte un conductor no permitido, que rechazar a un usuario permitido o confundirlo con otro usuario de la misma base de datos.

Inicialmente se propuso establecer umbrales a las distancias calculadas entre las proyecciones en el espacio transformado de las imágenes de entrada y los puntos correspondientes a toda la base de datos, pero este sistema no tenía demasiada capacidad de rechazar usuarios externos y los umbrales que se tenían que utilizar para que aquello funcionase con los vídeos de prueba eran demasiado altos, por lo que se mezclaban las regiones de decisión y no se funcionaba correctamente, con lo que se decidió separar la parte de rechazo de candidatos externos de la parte de reconocimiento, una vez se asegura que el usuario que está en frente de la cámara pertenece a la base.

Cuando se pensó en las posibilidades que se podrían dar en el espacio de proyección, se observó que teníamos varios supuestos que nos podríamos encontrar acerca de las posiciones de los puntos proyectados, de las imágenes de entrada respecto de los puntos contenidos en el universo de decisión. A la hora de pensar acerca del espacio de decisión, se supuso que las clases estaban agrupadas en torno a una media y que la distancia entre clases era suficientemente amplia para poder diferenciar claramente las distintas clases. Los 4 supuestos que se pensaron solucionar fueron los siguientes:

1. El punto correspondiente a la imagen de entrada puede estar tan alejado del resto de puntos del universo de decisión que su parecido sea ínfimo, por tanto este tipo de puntos se deberían rechazar automáticamente. Este caso se representa gráficamente en la subfigura 5.2.a
2. El punto de entrada puede estar entre 2 constelaciones, pero puede no pertenecer a ninguna de ellas, porque esté suficientemente alejado de las mismas, por lo que se debería rechazar automáticamente. Para que quede más claro, se realiza una representación 2D del caso en la subfigura 5.2.b
3. En la subfigura 5.2.c se muestra el tercer caso. Si hay constelaciones cuyos puntos se entremezclan, si está lo suficientemente cerca de dos puntos de distintas constelaciones podemos confundirnos con facilidad, con lo que habría que descartar este punto de entrada.
4. Por último, si está lo suficientemente cerca de un punto de una constelación y lejos de los demás, claramente el usuario de entrada será el correspondiente a la constelación donde esté el punto de la base de datos. Este caso se representa en el dibujo de la subfigura 5.2.d

Para implementar la funcionalidad de rechazo de usuarios externos, se optó por utilizar varias referencias externas a los puntos que componen las constelaciones de los usuarios, los cuales ayudarán a decidir si las características detectadas están suficientemente cercanas a las de alguna de las constelaciones existentes. Para ello, tomamos como puntos estratégicos para evaluar mejor las distancias el centro del universo de decisión, y los centros de las constelaciones. Como centro del universo de la constelación, tomamos la característica media de todas las que existan en la base de datos. Como centro de la constelación, utilizamos la característica media de las de un único usuario de la base de datos. Con estos puntos estratégicos, tenemos una idea más cercana de cuándo podemos rechazar a un usuario y cuándo no.

Además, la distancia euclídea presentaba el problema de que en ocasiones se confundía de usuario, o que, con el detector, a veces usuarios externos los tomaba como internos, al asignar

el mismo peso a todos los elementos de la matriz de características. Se decidió implementar la distancia de Mahalanobis, la cual se vio en varios artículos que funcionaba bien con este tipo de sistemas de reconocimiento. El hecho de implementar esta distancia, mejoró profundamente el comportamiento del detector. Parecía que a los usuarios que no pertenecían a la base incrementaban las distancias mientras que los que estaban dentro de la base se reducían considerablemente, lo cual proporciona una mejora a este subsistema.

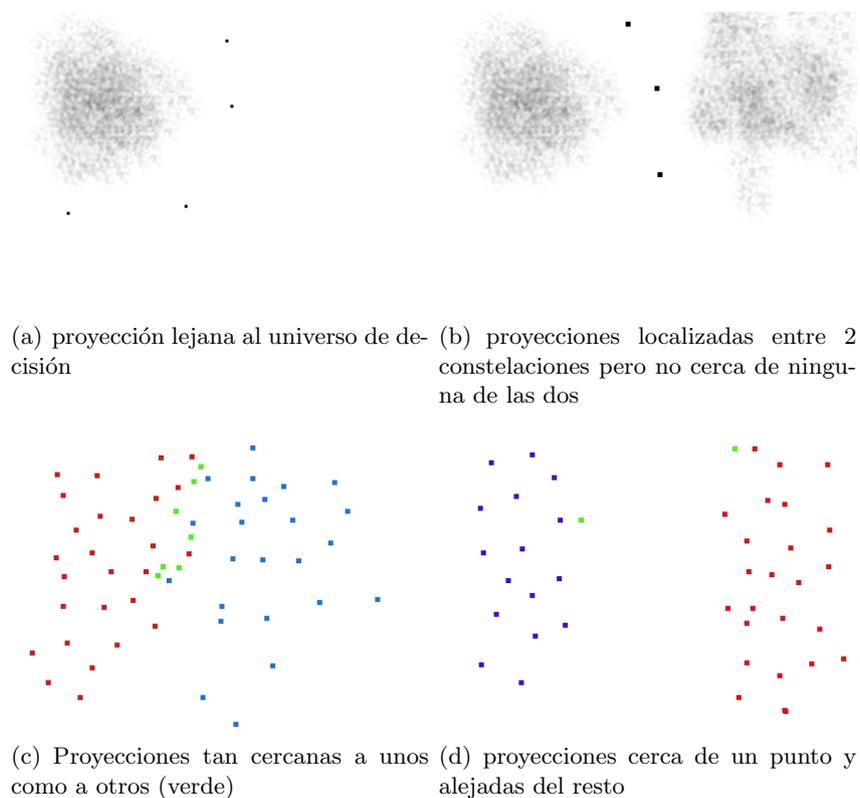


Figura 5.2: Casos posibles en espacio de proyección

El efecto de la introducción de la medida de distancia de Mahalanobis en nuestro sistema, respecto a la distancia euclídea, es, como se verá en los resultados, que en caso de que el usuario no se encuentre en la base de datos, la distancia aumentará considerablemente, mientras que en el caso de que el usuario se encuentre en la base de datos, la distancia disminuirá, y podemos reconocer mejor a un usuario, puesto que las diferencias entre las distancias se acentúan.

5.2.2. Clasificador

Para implementar el clasificador, se parte de las imágenes características calculadas con 2DPCA como puntos en el espacio de decisión. Utilizando los métodos de cálculo de distancias anteriores, se tiene una medida de cómo estamos de cerca o lejos de una región del espacio que define el conjunto de puntos de cada conductor.

Para implementarlo, se necesita el cálculo previo de varios puntos significativos. Hemos seleccionado este cambio de punto de vista para tener una visión más general acerca del universo

conocido. Los puntos significativos seleccionados como referencias son, como se ha dicho antes, el centro del universo de decisión y el centro de las constelaciones. Estos puntos se han calculado como la media de las matrices de características de todos los puntos y de los puntos correspondientes al usuario.

Después de calcular estos puntos de interés, se han de calcular medidas las distancias de estos puntos al resto de puntos del universo. Primero se calcula cuál es la distancia máxima desde todos los puntos al centro, para poder tener la primera referencia de distancia. Posteriormente se calcula la distancia máxima y mínima desde los puntos de cada constelación al centro de la misma puesto que puede haber constelaciones alejadas unas de otras, pero unas pueden estar muy alejadas del centro y otras muy cercanas, por tanto, hay que tener los dos parámetros en cuenta. Cada constelación tendrá dos medidas independientes, lo cual habrá que tenerlo en cuenta posteriormente.

Estos datos son calculados inmediatamente después de la carga de la base de datos en el programa.

Como se puede observar en el diagrama de bloques de la figura 5.3, este clasificador tiene varias etapas de funcionamiento, las cuales se pasan a describir a continuación:

1. Con el centro del universo tenemos una referencia para poder realizar un rechazo sencillo de los usuarios externos a la base de datos, pues normalmente éstos generarán distancias mayores que las de la base de datos. En este primer filtrado se calculará la distancia de este punto al centro de la constelación para comprobar si está lejos de este punto y por tanto rechazar inmediatamente la muestra de entrada como usuario no válido. Para tener cierta flexibilidad ante entradas válidas pero que excedan por poco la distancia máxima permitida, se aceptarán imágenes que cumplan que su distancia calculada al centro sea menor que $d_{cuni\ max} (1 + Th_1)$.
2. Si pasa el primer filtrado, se comprueba que la proyección de la imagen de características del conductor actual se ha quedado en una región intermedia entre las agrupaciones formadas por los conductores permitidos por el sistema. Para ello, se ha calculado para cada conductor, el punto central que le representa y la distancia máxima que le caracteriza. Si el punto obtenido desde el sistema está entre 2 centros de conductores de la base de datos, pero está muy alejado de los 2, el sistema se dará cuenta y rechazará esta imagen para el reconocimiento. Así como se ha hecho anteriormente, para que una imagen sea aceptada, tendrá que cumplir que sea menor que al anterior $d_{cusr\ max} (1 + Th_2)$ y también mayor que $d_{cusr\ max} (1 - Th_2)$. La justificación de utilizar una distancia mínima y máxima es que puede haber unas constelaciones de usuarios permitidos incluidas dentro de otras regiones y éstas tener una distancia máxima menor, por lo que se pueden introducir y ser detectadas como un usuario, aunque en realidad este sea otro.
3. Si ha pasado el filtro de las 2 etapas anteriores, si las regiones de decisión se entremezclan o son cercanas, comprobaremos a cuál de todas las imágenes de la base de datos se parece más, esto es, cuál es la que más cerca está. También calcularemos cuál es el segundo conductor en la base más cercano. Si está por encima de $d_{min} (1 + Th_3)$ el reconocimiento será válido. Si no, la imagen también será rechazada.

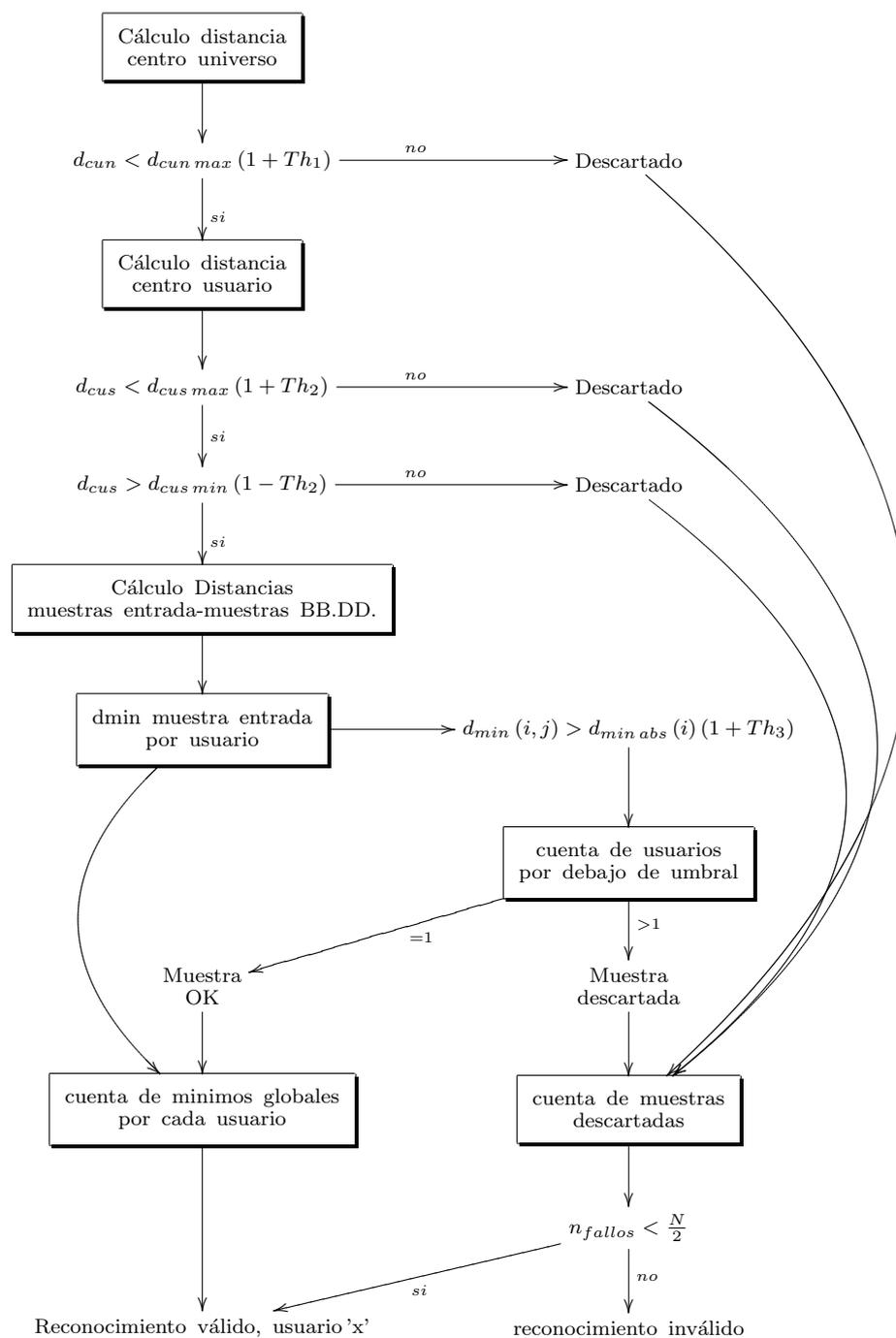


Figura 5.3: Funcionamiento básico del subsistema clasificador

4. Repetimos todo el proceso anterior con varias imágenes tomadas como muestras y se comprueba si en sucesivos intentos el conductor está dentro de la base de datos o no. Si se confirma el mismo conductor en las suficientes repeticiones, daremos el proceso por válido y tendremos seguridad sobre la identidad del conductor. Para asegurar mejor el reconocimiento se calculará la media de las distancias mínimas de cada usuario a las muestras de entrada. De estas distancias medias, calculando la mínima, tendremos otra medida fiable de cuál es el usuario que está en frente del sistema. Por último, comprobamos si el usuario devuelto por la medida de la media de las distancias es igual que el obtenido

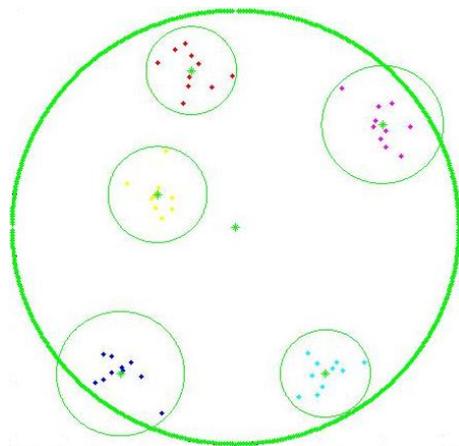


Figura 5.4: Representación simplificada de los 2 primeros filtrados

por el criterio de mayoría. Si es así, el usuario será correcto, y si no, el reconocimiento será fallido.

Vemos en el funcionamiento de este clasificador que, si un usuario se ha rechazado, se ahorran bastantes cálculos que tendríamos que hacer respecto del sistema anterior, puesto que se calculaban todas las distancias a todos los puntos del espacio transformado. Esta cascada de filtros supone una optimización importante a la hora del tiempo de cómputo y nos permite un funcionamiento ágil del sistema en caso de que el usuario no pertenezca a la base de datos.

También observamos que en este clasificador se pueden implementar otros métodos de cálculo de puntos de referencia, tanto del centro del universo como de centro de las constelaciones o clases. Un ejemplo de algoritmo que se puede cambiar para el cálculo del centro de las constelaciones es el de las K-Medias. Este algoritmo calcula los centros de separación de clases de manera iterativa, siendo circunferencias las superficies de separación entre clases.

Se puede sustituir la última etapa de clasificación por otro algoritmo más complejo pero más fiable, como SVM, el cual supone un aumento de la carga computacional bastante importante, a cambio de una mayor seguridad a la hora de realizar un reconocimiento correcto de cuál de todos los usuarios de la base es asignado al habernos asegurado que el candidato que tenemos en frente está dentro de la base de datos.

Este clasificador funcionará mejor con espacios que maximicen la separación entre clases y reduzcan la separación intra-clase. Esta característica la posee el espacio de proyección de LDA, en el que, a la vez que se procura maximizar la separación inter-clase, se minimiza la separación intra-clase, esto es, en el espacio de proyección separamos las constelaciones, y concentramos éstas mucho más alrededor del centro. Para incluir este algoritmo en el sistema tenemos dos opciones, una es sustituir 2DPCA por LDA, o concatenar los 2 espacios de proyección.

También se puede sustituir el clasificador completo por otros más eficientes, como una red neuronal convenientemente entrenada. El problema de este clasificador es encontrar el conjunto de entrenamiento adecuado para que el entrenamiento converja.

Este clasificador dará lugar, dependiendo de los umbrales introducidos, a tasas de reconocimiento más bajas o incluso rechazo de usuarios que estén en la base de datos, puesto que existe la probabilidad de rechazar imágenes correctas.

En una implementación real, se exigen tasas de reconocimiento superiores al 85 % para hacer que este sistema sea más robusto y estar completamente seguros de cuál es el usuario correcto.

5.3. Tratamiento de la iluminación

5.3.1. Introducción

A la hora de implementar este tipo de sistemas en un entorno real, aparecen varios problemas, como se ha comentado anteriormente. Uno de los problemas más importantes para algunos algoritmos de visión artificial es el problema de la iluminación. Un mismo color, iluminado de distinta manera se puede observar como colores muy distintos dentro de su misma gama. Este problema afecta de distinta forma a los distintos algoritmos de reconocimiento, sobre todo a los basados en apariencia. Los algoritmos basados en biometría, tienen el problema de que pueden perder los puntos característicos o los bordes utilizados para medir las distancias o contornos, tanto por exceso de iluminación como por defecto. Respecto a los algoritmos basados en apariencia, la iluminación varía el nivel de gris de la cara según la dirección de la fuente de luz, pudiéndose genera sombras en la cara. Por tanto, se varía la apariencia en esa zona y como consecuencia, variarán los coeficientes posteriores que obtendremos de las medidas a partir de estas imágenes, llegando a la confusión o el rechazo del candidato.

Este problema no ha sido resuelto satisfactoriamente hasta ahora, por lo menos en lo que a reconocimiento facial en 2D se refiere, aunque se han realizado diversas aproximaciones como se ha comentado en ???. Este problema es no lineal, puesto que la iluminación puede venir de infinidad de sitios, de distinto modo en cada instante, de distinto color según la fuente de luz, etc.

Las proyecciones en la cara pueden ser diferentes según la dirección y las sombras generadas en la cara también dependen, no solo de la forma de la superficie de la propia cara, sino de la forma del chasis del coche, puesto que estos bordes opacos que soportan los cristales, pueden generar sombras sobre la cara nada previstas, y además, estas sombras cambian de forma según el modelo de coche, la proyección desde el sol, etc. Por tanto, vemos que la variedad de posibilidades que se pueden presentar en la realidad es muy grande, y además, totalmente incontrolada, aunque el reconocimiento se haga con el coche sin arrancar, puesto que el coche se puede parar en cualquier sitio, por tanto las condiciones de iluminación pueden considerarse aleatorias.

En las siguientes páginas se muestran imágenes que ejemplifican lo que se pretende explicar acerca de la variedad de situaciones que se pueden dar con la iluminación, tanto en entornos oscuros, como en contornos claros e incluso en entornos en los que cambiamos es la orientación del coche respecto del sol en una hora en la que no entra directamente por las ventanas laterales. La diferencia entre un entorno cerrado y uno abierto es que, en el cerrado las condiciones de iluminación se pueden controlar un poco más que en el abierto.

Dentro de esta figura, en las subfiguras 5.5.a y 5.5.d, tenemos iluminación lateral que genera brillo en un lateral de la cara, mientras que el otro lateral se encuentra poco iluminado. Dependiendo de la dirección del foco, nos encontraremos con algunas de las condiciones de iluminación que se han expuesto anteriormente.

En la subfigura 5.5.e, vemos el contraste de iluminación de fondo con la de la cara, puesto que, en ese caso se está cerca de la salida del túnel, aunque también se generan sombras y el pelo queda oculto por las sombras que genera la luz.

En 5.5.b tenemos un rayo de luz del foco que molesta a la iluminación global de la escena, introduciendo brillo en una parte de la misma alterando zonas que no debería y empobreciendo por tanto la calidad de la imagen.

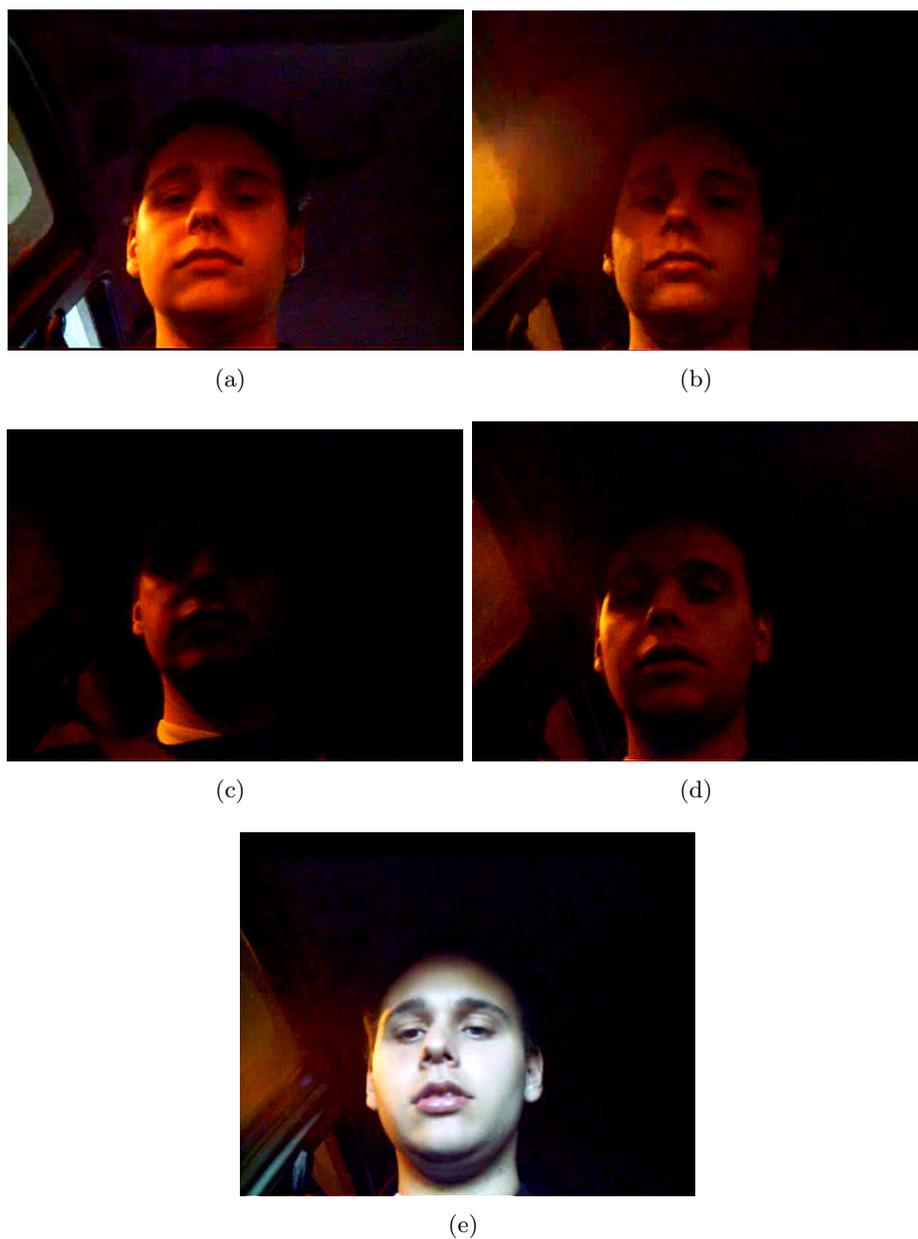


Figura 5.5: Ejemplos de iluminación variable en un túnel

Por último, en la 5.5.c encontramos un ejemplo en el que las condiciones de iluminación son paupérrimas. En este último caso sería necesario un suplemento de la iluminación o incluso proceder al reconocimiento mediante infrarrojos como se verá posteriormente.

En el caso de la iluminación exterior, vemos cómo la variabilidad es todavía mayor. En el caso exterior podemos tener una iluminación a la sombra, aproximadamente uniforme, la cual nos beneficia, pero también podemos tener cambios de iluminación muy acusados.

En los ejemplos de las subfiguras 5.6.a y 5.6.b se puede apreciar que la zona de la cara está muy saturada, al entrar toda la luz por las ventanas del vehículo.

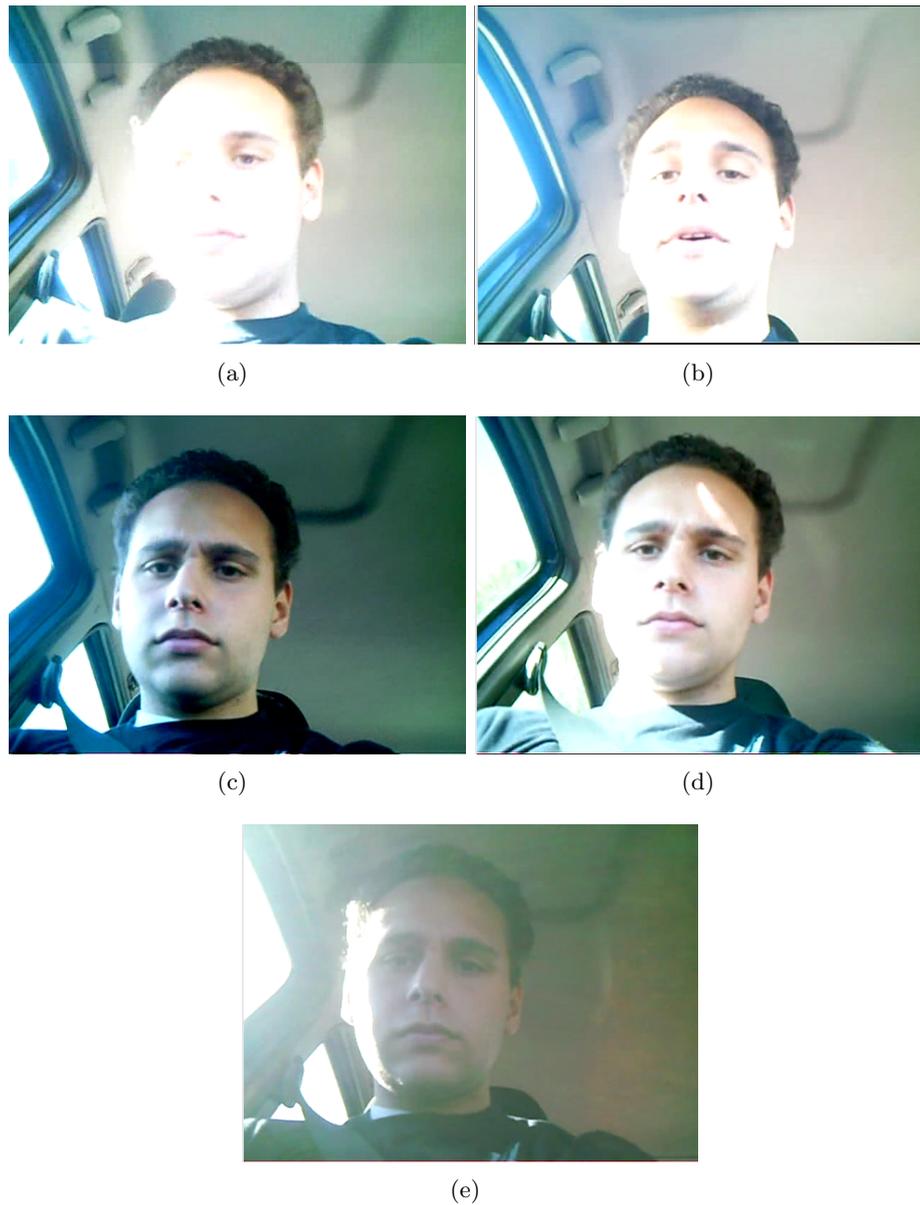


Figura 5.6: Ejemplos de iluminación al aire libre

En la subfigura 5.6.c se presenta un ejemplo de una iluminación general uniforme, se tiene una zona de la cara más iluminada, lo cual modifica la apariencia a la hora de aplicar el algoritmo de reconocimiento.

En 5.6.d se tiene un ejemplo claro de cómo la iluminación puede ser tan variable y alineal como se quiera, puesto que además de saturar un lateral de la cara, también se introduce en la frente en el otro lateral un haz de luz que proviene del hueco entre el retrovisor interior y el quitasol.

Por último en la foto correspondiente a la subfigura 5.6.e, vemos cómo la luz del sol que entra en la cámara altera completamente la escena, además de una fuerte iluminación lateral en la cara. En estas condiciones el reconocimiento que se realizaría sería paupérrimo, confundiendo o rechazando al candidato de entrada con casi total seguridad.

Para terminar, ilustramos el efecto del cambio de orientación del coche en la misma hora en

condiciones de sombra en los 2 primeros casos y un poco orientado hacia el sol en el tercero.

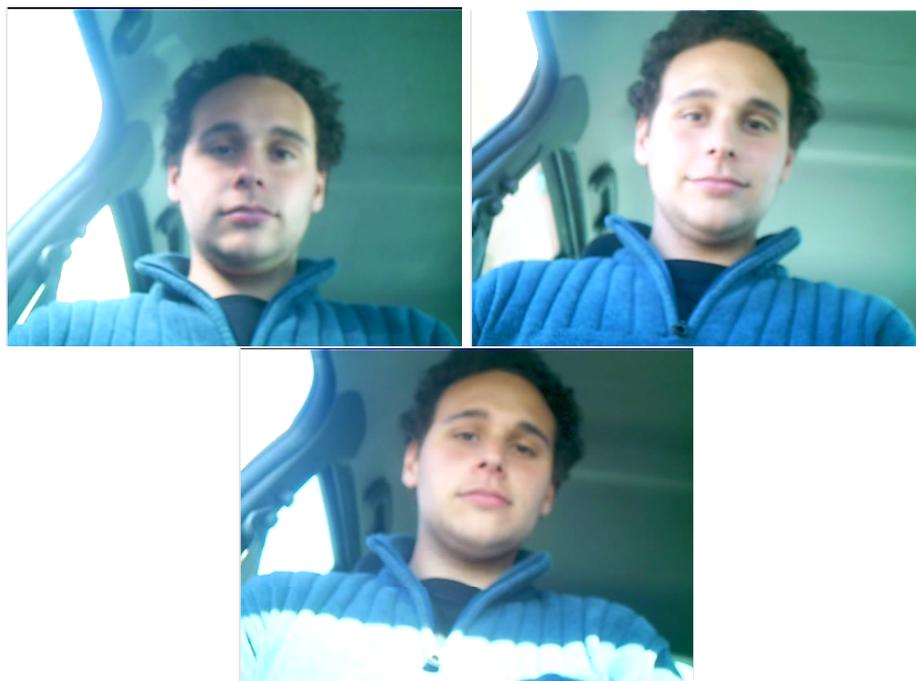


Figura 5.7: Ejemplo de variación de iluminación en entornos controlados

Se puede apreciar, cómo, en este caso la apariencia de la imagen también cambia, pues, aún encontrándonos en condiciones de iluminación cercanas a la uniforme, sobre todo en la subfigura 5.7.c y por tanto a la que consideramos ideal, debido a ligeras variaciones de posición de la cabeza y a que la iluminación entra por uno o los dos laterales, existen variaciones de nivel entre los distintos laterales de la cara.

5.3.2. Ecuación uniforme y filtrado de mediana

El método clásico para eliminar problemas de iluminación es la ecualización de histograma. Como se ha dicho en los apartados anteriores, este método transforma la imagen, para pasar de un histograma de origen a uno de destino, que en el caso normal de realce de características de una imagen, es un histograma plano, por tanto tiene un alto contraste. Al ser un histograma plano, puede suceder que en algunos píxeles de la imagen de la cara aparezca ruido de sal y pimienta. Frente a este ruido, se aplica el filtrado de mediana para eliminarlo, aunque la imagen se difumina un poco dependiendo del tamaño del filtro de mediana que se utilice.

En este caso de aplicación, el método de ecualización de histograma ha sido el implementado en OpenCV, en su función `cvEqualizeHist`, con los parámetros predeterminados. En cuanto al filtrado de mediana, se ha utilizado la función `cvSmooth`, con el parámetro `CV_MEDIAN`, en el cual se implementa un filtrado de mediana de 3×3 .

Aplicados los 2 métodos sobre unas imágenes de muestra, se obtienen los resultados que se muestran en la siguiente figura, en la que tenemos en la primera fila, las imágenes de las que se parte, y en la siguiente fila las imágenes obtenidas al aplicar el tratamiento.



Figura 5.8: Ejemplos de imágenes antes y después del tratamiento con ecualización de histograma y filtrado de mediana

Esta combinación de tratamientos solo se ha aplicado al canal de luminancia, no al resto de canales de crominancia, de ahí que estas imágenes tengan un aspecto extraño cuando se representan en 3 colores, aunque lo que realmente importa a la hora de realizar el reconocimiento es el nivel de gris de la imagen.

Vemos también cómo las variaciones en la iluminación son más acusadas en las imágenes tratadas que en las imágenes normales, por tanto, pueden hacer que el sistema funcione incluso peor de cómo funcionaba el anterior, como se verá en la sección de resultados.

Por el hecho de aumentar el contraste de las imágenes, aquellas zonas que aparecían un poco iluminadas, se iluminan bastante, y las zonas poco iluminadas aparecen más negras, no con el mismo valor de negro, pero con tonos muy similares. Respecto a las zonas intermedias, el tratamiento hace que la intensidad de gris varíe muchísimo en los mismos.

5.3.3. Recorte de cara a zonas de interés más invariantes

Este método se basa en seleccionar zonas de la cara que se han observado más invariantes a cambios de iluminación.

El método propuesto en esta sección tiene 2 objetivos:

1. Por un lado, se pretende, para el tratamiento de la iluminación, buscar una zona en la cara en la que la iluminación pueda provocar los menores cambios y se pueda realizar un reconocimiento correcto de la persona. La zona seleccionada corresponde a los ojos, la nariz y una pequeña parte de la frente. Además, esta cara es más invariante ante los gestos, pues el único grado de libertad que tiene esta zona son las posibles aperturas y cierres de los ojos. También esta zona contiene una buena parte de la información necesaria para el reconocimiento, aunque el resto de la cara también la contiene, esta zona es más invariante a cambios de peinado, barba u otras variaciones posibles que se pueden aplicar. El hecho de encontrar un método sencillo para reconocer correctamente a los usuarios e independizarlos en la medida de lo posible de la iluminación fue la motivación por la que se implementó y se probó el método.
2. Por otro lado, este método de selección de esta zona de interés de la cara nos sirve para poder detectar la presencia de gafas de sol, las cuales afectan a la apariencia del sujeto y

por tanto, pueden alterar el reconocimiento.

El problema observado al implementar este método para funciones de reconocimiento es que no hay suficiente información con el área recortada para realizar un reconocimiento correcto, y si además el usuario utiliza gafas y estas no están incluidas en la base de datos, se producirán rechazos o errores.

Además, el área total que es capaz de recortar este método para realizar el reconocimiento también influye a la hora de realizar un reconocimiento correcto, puesto que la cantidad de información que podemos extraer según el método de reconocimiento utilizado es proporcional a la cantidad de píxeles que se encuentran en la región y por tanto del área de la misma.

Para recortar esta zona de la cara, se utiliza el sistema de ajuste descrito en la sección de seguimiento de la cara en la implementación del sistema (4.3). A partir de las coordenadas corregidas del sistema, realizamos unos ajustes de offset superiores, inferiores y laterales.

Para calcular los puntos de interés, partimos de las coordenadas corregidas del apartado de tracking, una vez realizado el escalado de la imagen del detector.

Para seleccionar el área de interés, se calcula el ancho y alto de la región de interés corregida por el detector. A continuación, para los bordes laterales, se calcula una proporción de ese ancho, se suma al lado izquierdo y se resta al lado derecho. Respecto de los bordes horizontales, estas proporciones varían, pero desde el borde superior se suma una porción de la altura y desde el borde inferior se resta otra porción de la altura.

El procedimiento para la selección de este área de interés es el que se muestra en la figura 5.9.

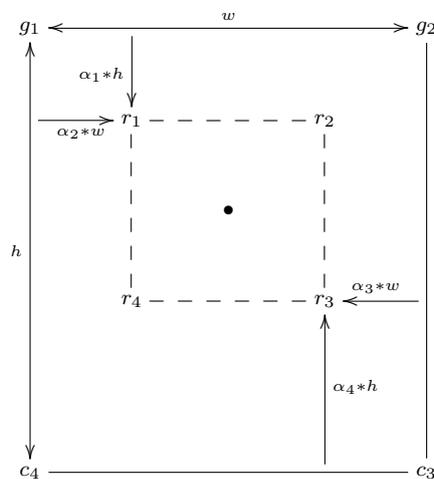


Figura 5.9: Procedimiento para la selección de la zona de interés de los ojos y la nariz

Después de ver el procedimiento de recorte de la imagen, pasamos a ver ejemplos reales de recorte con imágenes extraídas de los vídeos de prueba.



Figura 5.10: Ejemplos de imágenes recortadas

Se puede apreciar que no todas las relaciones de aspecto de las caras de las personas no son iguales, por tanto, al no hacer el tracking respecto de puntos característicos de la cara sino de las características devueltas por el detector, puede variar y por tanto el recorte no ser igual para unas personas o para otras. Vemos cómo los desplazamientos se ajustan para tomar la zona de ojos y nariz. En ocasiones se introduce parte de la boca en la zona de interés, dependiendo de la persona que esté delante.

5.3.4. Máscaras de Iluminación

En algunos artículos mencionados en la bibliografía se proponen ideas similares a este algoritmo. En [34] y otros artículos parecidos del mismo autor se realiza una estimación de la iluminación de la escena mediante técnicas basadas en el cálculo del gradiente y de los primeros términos de la ecuación de reflectancia y gradiente, y después se aplicaba una renormalización a la imagen en función de dónde estuviese apuntando el foco, para corregir en la medida de lo posible la iluminación que se tenía en la imagen de la cara.

El método propuesto en este proyecto se basa en el artículo de J. M. Buenaposada [20]. En este artículo se realiza la suposición de que una cara se puede separar en dos espacios: el espacio de iluminación y el espacio de la apariencia de la cara, debido a gestos, rotaciones o giros. Cada espacio se entrena con un vídeo diferente, en el que una de las dos condiciones se dejaba fija y la otra era la que se alteraba.

La iluminación en una cara se considera una variable aleatoria, con lo que no es posible entrenarla para todos los casos.

Con esta idea se realiza la siguiente propuesta: se entrena el espacio de gestos mediante secuencias de cada usuario dejando las condiciones de iluminación uniformes, y posteriormente se aplica a estas imágenes plantillas de iluminación con los patrones que ésta siga habitualmente.

Se generan una serie de imágenes que se correspondan con patrones de iluminación típicos en un coche. Estas imágenes patrón se utilizarán como plantilla para fusionarla luego a cada una de nuestras imágenes de la base de datos del programa. Este método es el que más recursos consume al ejecutar el algoritmo, tanto de memoria como de tiempo, pues el tamaño de la base de datos crece proporcionalmente al número de plantillas utilizadas para la iluminación, y por tanto el número de comparaciones se multiplica en función del número de plantillas de iluminación que se introduzcan.

El método seguido para la fusión de las imágenes de gestos con las plantillas de iluminación es el mostrado en la figura 5.11.

Los niveles de ajuste se introducen como entradas en el programa para realizar la composición de las 2 imágenes.

Como se puede observar, la iluminación no es todo lo realista que se pudiese esperar, aunque ayuda a entender el problema. Las plantillas de iluminación utilizadas han sido realizadas con

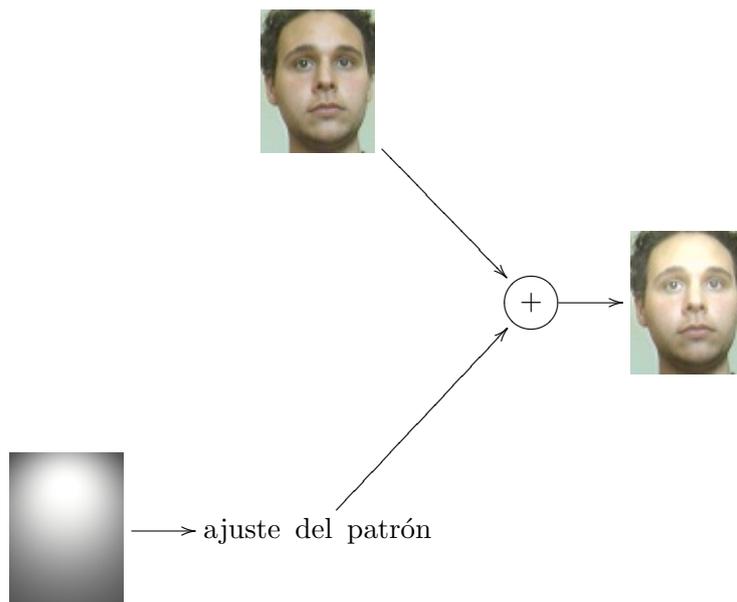


Figura 5.11: Aplicación del método a las imágenes de la base de datos

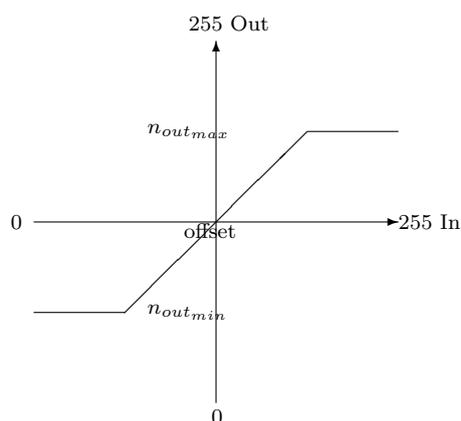


Figura 5.12: Función de ajuste de niveles de gris de las plantillas

programas de dibujo como gimp o Adobe Photoshop, con la herramienta de gradiente de niveles de grises.

Para entrenar al sistema con este método, lo que hacemos es tomar todas las imágenes de los usuarios de la base de datos y las combinamos con las plantillas que se hayan generando, enriqueciendo la base de datos.

Para implementar este sistema, se han generado unas máscaras de prueba, que intentan imitar la luz que viniese desde unos hipotéticos focos en varias dimensiones del espacio. Las plantillas utilizadas para las pruebas realizadas son las mostradas en la figura 5.13.

Estas máscaras, han sido realizadas con gimp con un degradado en escala de grises, con una imagen con un fondo de gris uniforme de valor 128. A la hora de aplicar el degradado, se ha elegido que este se realice de manera elíptica, pero con cierta perspectiva, imitando el efecto de un foco cuando impacta con una superficie. Se han generado estas máscaras desde distintos ángulos, para abarcar las zonas en las que más probablemente se encontrará una iluminación fuerte.

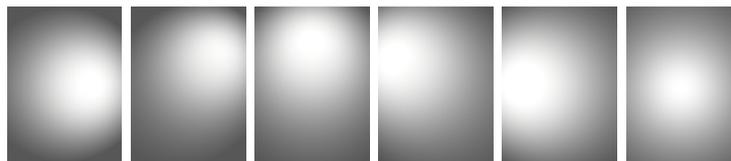


Figura 5.13: Plantillas de iluminación utilizadas

Con este método se consigue un comportamiento parecido al que se obtiene con el sistema sin tratamiento alguno, aunque se aprecian ligeras mejoras como se verá en la sección de resultados. A continuación se muestran ejemplos de la aplicación de las máscaras a una imagen de la base de datos.



Figura 5.14: Imágenes combinadas con las máscaras de iluminación

Por supuesto, el tiempo para realizar todas las comparaciones de las imágenes incluidas en la base de datos aumenta proporcionalmente con el número de máscaras utilizadas, lo que hace que este método no sea capaz de ser ejecutado con requisitos de tiempo real, sino que estos se deben relajar.

5.3.5. Retinex / SSR (Single Scale Retinex)

Como se ha visto anteriormente, este algoritmo se ha probado para mejorar la tasa de reconocimiento de caras, puesto que en el paper del que se tomó la idea [24] se dice que se consigue cierta independencia a condiciones de iluminación adversas, tomando la varianza de la gaussiana adecuada, para que, por un lado, exista suficiente información en la imagen para realizar un reconocimiento válido, y por otro lado, no tener una varianza tan grande que se permita la introducción de la información de iluminación.

Una vez implementado, se han hecho pruebas con valores de varianza cercanos a los recomendados en el paper. En la siguiente figura se presenta el resultado de aplicar el algoritmo a las imágenes de la base de datos con $c=4$, con un offset adecuado para hacer que la media de la imagen sea de 128 y aumentando el margen de valores de nivel de gris contenidos en la imagen, mediante una función de expansión de histograma, puesto que el margen de valores que deja el algoritmo es cercano al negro y el histograma que devuelve es estrecho, por tanto se añade una componente continua y se expande el histograma.

Las imágenes, como se puede apreciar tienen un aspecto distinto del que se ve aparentemente, tendiendo casi a una imagen de bordes, con apariencia casi de imagen de bordes. A cambio la



Figura 5.15: Ejemplos de imágenes tratadas con Retinex

información de iluminación, si esta no es muy extrema, esto es, con muchas luces y sombras, se verá suprimida. Si la iluminación es muy fuerte en una zona de la cara aparecerá un borde correspondiente con el borde que genera en la cara ese área con demasiada iluminación.

Algunos de los detalles de la cara se ven resaltados con este algoritmo, como por ejemplo la rugosidad en las pestañas, los pelos, superficies rugosas, etc.

5.4. Detección de gafas

5.4.1. Introducción

Para esta tarea particular se pueden utilizar algunos métodos basados en distintas técnicas de visión artificial.

Un algoritmo desarrollado a partir de ajuste por mínimos cuadrados se utiliza en [1]. En este caso se utiliza para descartar la zona de las gafas para el reconocimiento y utilizar el resto de la imagen para realizarlo, aunque se pierda eficacia y seguridad en su uso. Para llevar a cabo la detección de las gafas, umbraliza la imagen, realiza una extracción de bordes, extrae las elipses por mínimos cuadrados, y realiza comparaciones de similitud entre los parámetros de las elipses. Si éstos siguen determinadas proporciones, se asegura que se ha detectado una gafa.

En este sistema se requiere que capaz de ejecutarse en el menor tiempo posible, para poder incluirlo en un sistema de tiempo real. Por tanto, no podemos andar calculando los parámetros de las elipses, al consumir este proceso mucho tiempo. En el método de detección propuesto se propone bajar un poco el nivel de abstracción e ir a tratar directamente con las características que pueden tener las imágenes en las que en su interior tengan gafas cerca de los ojos.

La idea en la que se basa este subsistema es que en, al haber un contraste entre el color de las gafas y el color de la piel, los algoritmos de detección de bordes y posteriormente de contornos, podrán localizarlos. Además, se sabe que éstos bordes pasarán cerca del cruce que existe entre la línea vertical formada por el tabique nasal y la línea horizontal formada por los ojos. Por tanto lo único que tendremos que hacer será calcular la distancia mínima del contorno a este punto de interés. Por último, los contornos cuyo perímetro es largo, pero su área es pequeña, nos dan una idea de que son estrechos y por ende no corresponden a unas gafas.

Con este algoritmo, no se exige que la forma de las gafas sea elíptica, por tanto es más flexible en este aspecto que otros algoritmos desarrollados para esta tarea, aunque el porcentaje de falsos positivos sea superior.

Empezaremos explicando cómo se realizan dos operaciones importantes sobre la imagen para la extracción de bordes, y posteriormente explicaremos detalladamente el método propuesto.

5.4.2. Método propuesto para la detección de gafas

Como hemos podido ver en el anterior ejemplo de algoritmo utilizado, si en la cara tenemos unas gafas, lo que ocurrirá es que se generarán una serie de bordes a lo largo de la misma, por el contraste entre el color de la cara y el de las gafas.

Este algoritmo se basa en la detección de bordes en una zona restringida de la cara, la de los ojos y la nariz. En este caso no se realiza un ajuste por elipses del contorno para corroborar perfectamente si es una gafa y de qué tamaño es, para hacer que este detector vaya lo más rápido posible.

La casuística con los bordes posibles detectados es amplia y no la describiremos en su totalidad, pero sí nos meteremos en algunos casos particulares. En una cara normal pueden aparecer bordes que no sean los de una elipse.

1. Se puede tener 2 contornos alrededor de las gafas (la situación menos frecuente en una imagen en condiciones reales), lo cual hará que la detección sea muy sencilla si estos son lo suficientemente grandes y poco degradados debido a los reflejos que se van a producir en la gafa. Este tipo de situación puede darse cuando tengamos sombras en los ojos que abarquen la cuenca del ojo o cuando tengamos gafas con lentes circulares con soportes y patillas finas. En este último caso y como veremos posteriormente los bordes generados por estas últimas partes no nos afectarán demasiado cuando realicemos sobre la imagen las operaciones de dilatación y erosión. Un ejemplo visual de lo que se pretende decir se encuentra en la figura 5.16.



Figura 5.16: Primer caso posible de detección de elipses

2. Se puede tener un único contorno que rodee la gafa, no teniendo porqué ser este elíptico, puesto que las gafas grandes utilizadas en la actualidad no son dos elipses sino que se componen de un único cristal, por tanto, no es posible guiarnos solo por el ajuste por elipses. También se puede dar el caso de la detección de un único contorno no elíptico si, por la perspectiva, el puente que une los 2 cristales es muy amplio. Para poder ilustrarlo mejor se muestra un ejemplo en la figura 5.17.
3. En la figura 5.18, mostramos el tercer caso que se puede dar en este detector. El borde detectado puede unirse con el borde generado entre la cara y la sección de pelo que podemos encontrar en la sección de cara seleccionada. Por tanto, este borde puede alargarse hacia los extremos de la imagen. Con un ajuste por mínimos cuadrados o transformada de



Figura 5.17: Segundo caso de detección de contornos

Hough, este problema podría corregirse con facilidad, pues solo se seleccionarían los puntos adecuados del borde y se descartarían todos los que se alejan de los parámetros elegidos por la mayoría. En la práctica, se comprueba que este caso es el más frecuente, puesto que, normalmente, para gente con el pelo moreno, al ser las gafas de color oscuro, el borde en lugar de cortarse se alarga y se extiende a lo largo de la imagen.

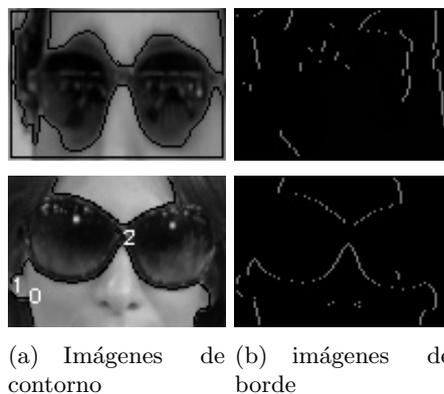


Figura 5.18: Tercer caso de detección de contornos

4. En una imagen como la de la figura 5.19, se puede detectar solo un borde, pero al ser éste demasiado alejado o demasiado largo para ser un contorno de una gafa, entonces se descartaría.

Se puede apreciar cómo, además de no captar correctamente el borde único de la sombra, puesto que se junta este borde con el formado por el pelo y la cara, toma como borde conexo con él, el borde exterior de la fotografía, con lo cual, por exceso de longitud, después se descartará o se penalizará en la función fitness.



Figura 5.19: Cuarto caso de detección de contornos

5. En la figura 5.20 mostramos otra de las posibles situaciones a la hora de detectar bordes. El brillo de las gafas puede ocasionar la generación de muchos más bordes de los previstos, lo cual puede alterar los resultados, aunque si los bordes son lo suficientemente cercanos al contorno exterior no tenemos problemas por el algoritmo utilizado. Además, si uno de los bordes es el predominante y se ajusta bien al contorno que deseamos, el resto de bordes generados lo único que ocasionarán es más trabajo para el detector a la hora de calcular funciones fitness de los bordes candidatos a ser gafas.



Figura 5.20: Quinto caso de detección de contornos

6. También se puede dar el caso de encontrar varios bordes, pero muy alejados del punto de referencia, por lo que esos contornos quedarían descartados. En la figura 5.21 podemos apreciar el punto utilizado de referencia para calcular las distancias y los bordes obtenidos una vez pasados los filtrados iniciales. Estos bordes se encuentran muy alejados del punto de referencia, lo cual va a penalizar bastante. Además, los bordes característicos del pelo, son largos, pero su contorno es alargado, por tanto, tiene menos superficie, lo cual penalizará en la función que se aplicará a esos parámetros, dando un reconocimiento negativo.



Figura 5.21: Sexto caso de detección de contornos

7. Por último, en el caso claro de no llevar gafas y tener una iluminación uniforme, se obtendrán bordes correspondientes a pequeñas sombras o salientes de la cara, los cuales no tienen ni los valores de superficie ni longitud ni la suficiente cercanía al punto de origen de medida de distancias como para pasar correctamente el umbral impuesto en la función fitness, por lo que se rechazarán. Un ejemplo de este caso, lo encontramos en la figura 5.22.



Figura 5.22: Séptimo caso de detección de contornos

Además, si la sombra generada en esa zona de interés es muy fuerte, fallará tanto este como cualquier otro algoritmo que se base en la detección de bordes en la imagen. De todos modos, si existe una sombra en la cuenca de los ojos, no registrada en la base de datos, el sistema en cierto modo nos ayudará a prevenir sobre estas situaciones adversas respecto de la iluminación.

A continuación pasamos a mostrar las principales etapas de las que se compone el algoritmo de detección de gafas, para una sola imagen.

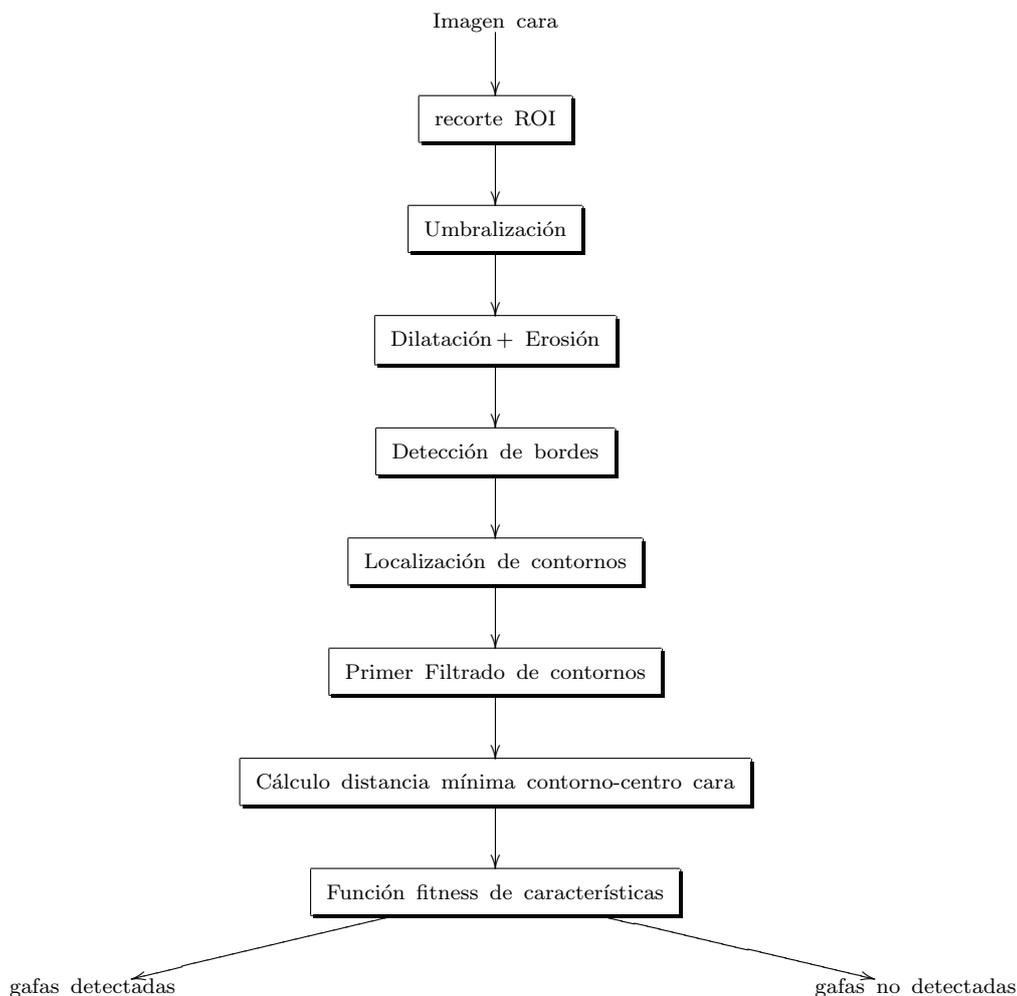


Figura 5.23: Esquema de funcionamiento del algoritmo de detección de gafas

Ahora pasamos a describir los principales bloques que hemos incluido en la figura anterior.

1. Se convierte la imagen procedente de la fuente a una imagen en escala de grises (suponiendo que la cámara graba en color).
2. Recortamos la zona de interés dentro de la cara, compuesta por los ojos, la nariz y parte de la frente. El modo en el que se recorta esta zona de interés se cuenta con mayor detalle en 5.3.3. De este modo reducimos el área en el que tenemos que buscar y se restringen bastante las condiciones en las que queremos encontrar bordes.

3. Umbralizamos la imagen de manera que podamos detectar correctamente las zonas oscuras. Después de probar con distintos tipos de umbrales (adaptativo, Otsu, fijo, etc), nos quedamos con el umbral de Otsu al ser el umbral óptimo si queremos dividir en dos clases los datos que tenemos en el histograma. En las pruebas finales ha sido el que mejores resultados generales ha dado para este tipo de casos, distintos tipos de gafas y distintos tipos de piel (el contraste con muestras de personas de raza africana - afro americana, que llevaban puestas gafas de sol era suficiente).
4. A continuación, con la imagen umbralizada, se realizan operaciones de dilatación y erosión para eliminar zonas que podrían generar bordes no deseados que pueden interferir con los principales. Con la primera pasada de dilatación se ensanchan los bordes, con lo que conseguimos que los bordes gruesos se resalten y los finos, apenas varíen. Con la erosión posterior, conseguimos eliminar todos los bordes que no son significativos, manteniendo únicamente los principales. También si hay muchos bordes finos juntos, se consigue agruparlos, generando una zona más gruesa en la imagen y con la erosión posterior sigue quedando un borde significativo, en lugar de múltiples bordes, lo que hace que reduzcamos el tiempo de proceso del sistema.
5. Después, pasamos un filtrado para obtener los contornos de la imagen anterior. En este caso el filtro utilizado ha sido el predeterminado de la función de encontrar contornos, que muy posiblemente sea Canny, aunque se podría haber pasado perfectamente una imagen umbralizada con Canny o Sobel o Prewitt.
6. El siguiente paso es un filtrado ligero de candidatos para no tener tanta carga posteriormente. En este paso descartamos los perímetros con una longitud mayor que un 80 % del perímetro total de la imagen ($2 \cdot (\text{altura} + \text{anchura})$) y también descartamos los perímetros cuya longitud sea inferior a 20.
7. Para continuar, extraemos las características de cada contorno, esto es, centroide, superficie, longitud (aunque ya lo habíamos hecho antes), y también calculamos la distancia a un punto central definido en la foto, donde vamos a tener una probabilidad muy alta de obtener bordes correspondientes a gafas. Este punto de referencia se obtiene a partir del punto central del rectángulo dado por el detector, bajándolo un offset vertical proporcional a la altura de la imagen, con lo que nos aseguramos que independientemente de la relación de aspecto de la cara, el punto de referencia quede aproximadamente donde se requiere. La posición vertical más adecuada (el factor de proporcionalidad del offset) ha sido obtenido empíricamente para maximizar los aciertos.
8. Por último aplicamos la función. En la foto, como mínimo siempre se va a encontrar el borde exterior, con lo que si solo se ha detectado un único borde, ignoramos la muestra. Sin embargo, si existen dos o más bordes principales en la imagen se procede a calcular la función fitness. Para detectar realmente si los bordes detectados son en realidad pertenecientes a las gafas utilizamos una función fitness (similar a las utilizadas en las funciones de coste en los algoritmos genéticos) compuesta por tres componentes que pasamos a ver a continuación:

$$f_i = \frac{2 \cdot (w + h)}{c_i} + \frac{w \cdot h}{s_i} + \frac{1}{d_i} \quad (5.1)$$

donde w es la anchura, h es la altura, c_i es el contorno i -ésimo detectado en la imagen a evaluar, s_i es la superficie asociada al contorno i -ésimo, y d_i es la distancia desde un punto característico fijado en el centro de la imagen.

La justificación de calcular la distancia del contorno a este punto es, porque cerca del mismo vamos a localizar la mayoría de contornos asociados con las gafas de sol que queramos detectar. Cuanto más cerca estén estos contornos de este punto mayor posibilidad tienen éstos de pertenecer a las gafas de sol.

Respecto de la longitud del contorno, la utilizamos para desechar los contornos demasiado pequeños o demasiado grandes que se aproximen más al contorno exterior que a contornos que estén contenidos dentro de la imagen.

Por último utilizamos la superficie para desechar contornos largos pero de superficie pequeña, o contornos muy grandes que se aproximen a la superficie total que ocupa la imagen.

Umbralizando esta función fitness, se puede saber si en una imagen hay gafas o no. El umbral utilizado (0.7) ha sido obtenido empíricamente a partir de un conjunto de prueba anterior a las pruebas con vídeos reales.

Para ilustrar alguna de las etapas, mostramos algunas imágenes a continuación:

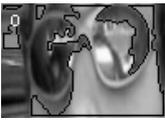
	Imágenes de entrada	Umbralización	Obtención de bordes	Contornos finales
Caso a: Sin gafas				
Caso b: con reflejos intensos				
Caso c: con reflejos suaves				
Caso d: Otras gafas				

Figura 5.24: Imágenes preprocesadas para la obtención de bordes

Como se puede ver en las imágenes de esta figura, en el caso 5.24.a, que no se llevan gafas, el número de bordes generados es bajo y su amplitud no es lo suficientemente grande, además, solo se ha generado un borde alrededor de un ojo, por tanto será el único significativo y será rechazado por los umbrales impuestos de la función fitness.

Respecto al resto de imágenes, casos 5.24.b, 5.24.c y 5.24.d; vemos que el número de bordes generados y la distancia al punto significativo es inferior, por tanto la función fitness es mayor y se aceptará como una imagen con gafas.

Analizando el caso particular 5.24.b, observamos que la gafa tiene muchos reflejos, lo cual puede perjudicar bastante a nuestro sistema, pero si vamos un poco más allá, vemos como el borde formado por la lente izquierda (visto en la imagen, no en la realidad) es lo suficientemente grande como para darnos un positivo en la detección de gafas.

En el caso 5.24.c, al reducirse la cantidad de reflejos, el resultado de la imagen umbralizada es mejor y por tanto el de la obtención de bordes también, reduciéndose su cantidad y aumentando su "calidad", apareciendo muchas menos zonas intermedias.

Para terminar de analizar casos, en el 5.24.d, se aprecia cómo el resultado de la umbralización ha sido muy bueno, también debido a que las gafas tienen un color más oscuro que las anteriores. Por tanto, en este experimento, la detección de gafas de sol no tendrá demasiados problemas, al generarse 2 bordes claros.

La integración final de este detector en el sistema completo será la siguiente:

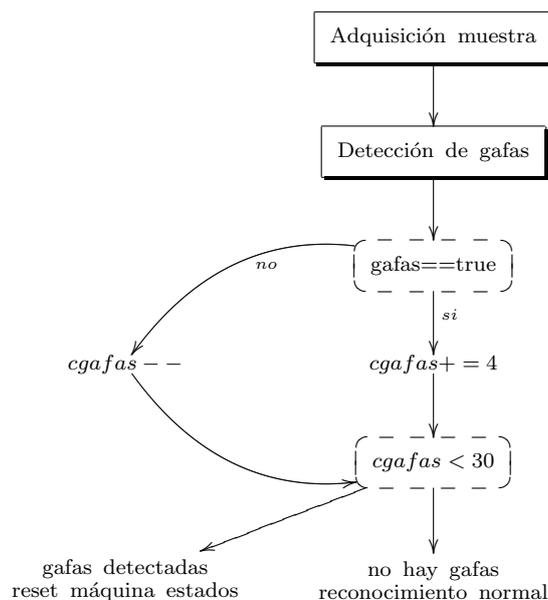


Figura 5.25: Integración en el sistema del algoritmo de detección de gafas

Para terminar, en la máquina de estados del sistema, se contarán cuántas de las imágenes de las de entrada se han detectado gafas. Para hacer que esta detección evolucione ante falsos positivos, se decrementará el contador en una unidad, mientras que si se han detectado gafas, el acumulador se incrementará con mayor velocidad, así, varios positivos seguidos harán que la detección sea positiva, mientras que unos pocos casos aislados no producirán que se detecte.

Por supuesto, se preferirá por seguridad, tener más falsos positivos a que se introduzca en las etapas posteriores un usuario que tenga las gafas de sol puestas, por lo que afectaría al sistema completo y podría pasar como otro usuario, con las consecuencias que ello supone.

5.5. Reconocimiento diurno y nocturno

5.5.1. Introducción

Uno de los problemas fundamentales que se tiene en este sistema es que, aun habiendo tratado de paliar el problema de la variación de la iluminación, nos encontramos ante el problema de que no es posible realizar un reconocimiento correcto en condiciones de baja iluminación o que esta sea nula.

Este tipo de casos los encontraremos cuando se esté en un aparcamiento subterráneo, en el cual se dan varios factores para que la iluminación sea tan pobre que no se pueda realizar un reconocimiento correcto. Entre la oscuridad debida a estar bajo tierra, a que la iluminación es pobre y que además, la poca que hay normalmente no entra en el coche debido a que el techo del coche genera sombra como para que la zona que da a la cara esté suficientemente oscura como para que el detector de Viola & Jones no funcione.

En este caso se utiliza un sistema de iluminación interna basado en infrarrojo.

Las características de las imágenes nocturnas son distintas que las de las imágenes diurnas. Esto es debido al tipo de iluminación, a la sensibilidad y al modo de trabajo de la cámara (distinto para imágenes diurnas que para nocturnas), a que la iluminación en este caso viene solo desde un punto cuando en imágenes diurnas la iluminación es más global. Por estos motivos hay que realizar una base de datos diferente según las condiciones de iluminación que se den en cada situación.

A la hora de inicializar el programa tenemos que saber en cuál de las dos condiciones está el entorno, si en una imagen nocturna o diurna, para cargar una base de datos u otra. Por tanto, se necesita un subsistema, basado en los datos recibidos desde la cámara, que sea capaz de decidir con certeza si las imágenes de entrada iniciales son nocturnas o diurnas, y a partir de entonces, proceder con la carga de la base correspondiente y de encender los leds iluminadores infrarrojos para realizar el reconocimiento nocturno.

Para implementar esta funcionalidad de decisión, se ha decidido utilizar un sistema de fusión de distintas medidas, puesto que solo con un origen de datos de medida no podemos estar totalmente seguros de que una imagen sea diurna o nocturna. Esta parte es fundamental pues de ella depende que se cargue la base de datos correcta o no y por tanto el sistema rechace a los usuarios. Se ha de dar también cierta flexibilidad al sistema, puesto que los umbrales no son fijos, sino que están definidos en un intervalo y la frontera entre los dos tipos de imágenes no es clara del todo.



(a) imagen diurna

(b) imagen nocturna

Figura 5.26: Ejemplos de imágenes de entrada diurnas (a) y nocturnas (b)

5.5.2. Selector de noche / día

A partir de las primeras imágenes de la cámara tenemos que detectar el hecho de tener imágenes nocturnas o diurnas, para poder encender la iluminación infrarroja o no automáticamente.

Se ha optado por implementar un sistema basado en parámetros intrínsecos de la cámara y además, con la propia imagen de la cámara.

Para hacer esto, a partir del nivel de gris medio de la imagen de origen se podría haber realizado, pero para asegurarnos de que se está en condiciones nocturnas, modificaremos el nivel de exposición de la cámara, si esta tiene esa posibilidad. Entonces, con estos 2 parámetros de entrada y aplicando unas reglas sobre el conjunto de entrada, se puede implementar un controlador borroso, como los implementados en algunas cámaras fotográficas que regulan el nivel de exposición o la ganancia de la misma.

Como primera aproximación, se pensó en utilizar como parámetros solo los que se pudiesen obtener directamente de la imagen. Observando imágenes tomadas de infrarrojos y las diurnas, se pudo observar que la media daba información, pero no la suficiente, por lo que se optó por introducir la varianza de los canales de crominancia basándonos en la siguiente justificación: los infrarrojos hacen que la imagen tienda al gris en una imagen a color, por tanto, la varianza de los canales de crominancia ha de ser mucho menor que la que se da en condiciones diurnas, pero se vio, que en una implementación real esta condición no se daba tal y como se pensaba, por lo que se descartó.

Se intentó otra aproximación extraída a partir de la ganancia de la cámara y la iluminación media de la imagen, pero se comprobó que la ganancia de la cámara no era tan dependiente de la iluminación como se esperaba, con lo que se buscó el parámetro por el que la cámara hacía un ajuste automático de la iluminación obtenida cuando había un cambio, y ese parámetro resultó ser el tiempo de exposición.

Además, respecto de las variables de entrada, no se puede asegurar directamente que una imagen sea nocturna o diurna asignando un umbral fijo, puesto que no se sabe exactamente en la práctica cuál es el nivel de gris para el que una imagen es nocturna o diurna. Puede darse el caso de haber imágenes nocturnas con un foco muy intenso en una zona reducida de la imagen, y también puede darse el caso de encontrar una imagen diurna con una media baja debido a la baja iluminación que se tiene.

Ante este nivel de indeterminación se ha optado por definir el siguiente conjunto borroso para la variable de nivel de gris medio de la imagen de entrada.

Además de este nivel de gris, información que procede de la cámara que no es directamente la imagen, nos puede ser útil para deducir si estamos ante una imagen nocturna o diurna. Con el tiempo de exposición de la cámara (shutter) o con la ganancia de la misma (aunque con este parámetro no se puede asegurar tanto la certidumbre del detector) podemos hacer un sistema con varias mediciones, el cual es más fiable que utilizando una sola medición.

Los conjuntos borrosos pertenecientes a las variables de entrada son los que se muestran en la figura 5.27.

Las variables lingüísticas de los conjuntos borrosos son las siguientes:

- MO: Media Oscura. Indica que el nivel medio de gris de la imagen es bajo y por tanto la imagen es oscura.

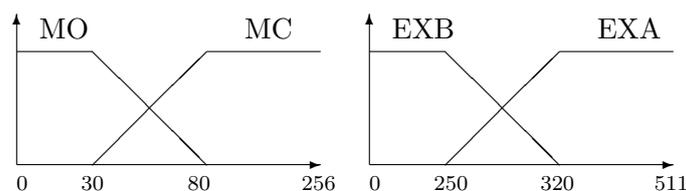


Figura 5.27: Conjuntos borrosos pertenecientes a la media de nivel de gris de la imagen

- MC: Media Clara. Indica que el nivel medio de gris es alto y por tanto la imagen es clara.
- EXB: Exposición Baja. Indica que el nivel de exposición de la cámara es bajo, esto es, que la cantidad de luz absorbida por la cámara no es muy alta, lo cual nos servirá para saber si la cámara se está iniciando o ya se encuentra en un funcionamiento en régimen permanente.
- EXA: Exposición Alta. Indica que el nivel de exposición es alto, o sea, el tiempo que se deja al sensor adquiriendo es el suficiente como para que, si encontramos una imagen oscura, sea porque estemos en condiciones nocturnas.

Se han seleccionado los valores característicos que se ven en la figura por los siguientes motivos:

Niveles de grises inferiores a 30 o valores similares son muy parecidos al negro para el ojo humano y además es más que suficiente margen para que una imagen nocturna quede por debajo de él, a menos que tenga varios focos puntuales que puedan subir la media.

A partir de este valor de nivel de gris y avanzando hacia valores superiores, el tono de gris se va aclarando. Se ha seleccionado 80 como valor de nivel de gris límite para el conjunto MO (Media Oscura) por si toda la imagen es oscura salvo un foco de luz muy brillante que exista en alguna parte de la imagen, o incluso luces de baja intensidad que puedan subir el nivel de gris, aunque la imagen corresponda a una imagen oscura. Sin embargo en el atardecer, que tengan características cercanas a las imágenes infrarrojas, pero todavía se consideran imágenes diurnas.

En cuanto al conjunto MC, podemos tener imágenes oscuras diurnas, cuya media esté entre el margen de 30 a 80, si estamos en condiciones de baja iluminación, ya sea con sombra, o estar atardeciendo o anocheciendo. Si el nivel medio de gris está entre 80 y 255, son más que suficientes para asumir que la imagen se encuentra en condiciones diurnas, pues si fuese una imagen nocturna, se necesitaría un foco apuntando directamente a la cámara para hacer que la imagen tenga el suficiente brillo para que se cargue la base incorrecta y que el sistema falle.

En cuanto a los niveles de exposición EXA y EXB, se han seleccionado estos niveles experimentalmente, en un entorno en el que había zonas oscuras y claras a las que apuntar con la cámara y manipulando manualmente el parámetro shutter. Si el parámetro de shutter (o nivel de exposición) está al máximo, significa que el tiempo de exposición de los sensores es pequeño, y si la imagen que aparece es clara, y por tanto estamos en condiciones diurnas. Si el nivel de exposición es pequeño, puede ser por 2 motivos:

- la cámara se está inicializando y está tendiendo a valores superiores de shutter, con lo que la imagen es oscura
- Si se consiguen obtener imágenes claras de la cámara con tiempos de exposición altos, significa que estamos en el caso de imágenes diurnas, pero con iluminación baja.

A continuación se muestran las reglas utilizadas en el sistema borroso implementado.

	MO	MC
EXA	IR	VIS
EXB	ERR	VIS

Tabla 5.1: Tabla de reglas utilizadas

Se ha elegido esta tabla de reglas, algo peculiar que se pasa a explicar a continuación.

En primer lugar, tenemos una variable que representa una salida correspondiente a una imagen con iluminación diurna (VIS), una correspondiente a iluminación nocturna (IR) y una salida especial correspondiente a error.

Si la imagen es clara, si tiene un tiempo de exposición bajo, la iluminación es muy fuerte y por ese motivo la cámara ha tenido que reducir ese tiempo, sin embargo, si la imagen es clara y se tiene un tiempo de exposición alto, significa que, aun siendo clara la imagen corresponde con una imagen a la sombra, pero de día.

En cambio, si la imagen es oscura y tiene un tiempo de exposición alto, estaremos en el caso de que la cámara está haciendo lo que puede para captar luz, pero no es suficiente para obtener una imagen en la que se detecten objetos claros, por tanto estaremos ante una imagen nocturna, susceptible de iluminar la escena con infrarrojos.

Un caso especial, que se ha de tratar a parte, es el aquel que la imagen es oscura pero el tiempo de exposición bajo. En este caso no somos capaces de decidir si la imagen obtenida es diurna o nocturna. Si el tiempo de exposición es bajo, entonces poca luz es capaz de llegar al sensor, y por tanto, podríamos estar ante una situación de arranque de la cámara, en la cual se vaya autoajustando el valor del tiempo de exposición, y no esté todavía en sus valores altos.

Respecto del tipo de sistema utilizado, en esta ocasión, debido a los requisitos de velocidad requeridos, se utiliza un tipo sugeno, es decir, el desborrosificador es de centros ponderados.

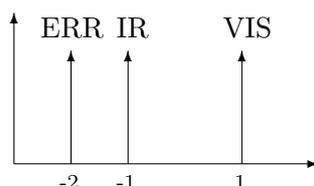


Figura 5.28: Valores de salida posibles para la salida sugeno implementada

Se ha asignado -2 a la salida de error, para que la transición sea menos abrupta en el caso de que, en la cámara, el tiempo de exposición aumente, mientras que el nivel medio de gris en la imagen se mantiene bajo, y la función de salida siga siendo negativo.

- Posible simplificación

En el caso de que el tiempo de exposición permaneciese suficientemente alto, es decir, el valor de activación del conjunto EXA fuese máximo (1), este controlador borroso se puede reducir a una simple umbralización como pasamos a calcular a continuación.

Al ser conjuntos trapezoidales laterales, podemos utilizar las siguientes ecuaciones para describir el valor de activación de cada uno de los conjuntos.

$$\mu_{MO} = \begin{cases} 1 & 0 \leq m \leq 30 \\ 1 - \frac{(m-30)}{50} = \frac{8}{5} - \frac{m}{50} & 30 \leq m \leq 80 \\ 0 & 80 \leq m \leq 255 \end{cases}$$

$$\mu_{MC} = 1 - \mu_{MO} = \begin{cases} 0 & 0 \leq m \leq 30 \\ \frac{m}{50} - \frac{3}{5} & 30 \leq m \leq 80 \\ 1 & 80 \leq m \leq 255 \end{cases}$$

Aplicando las reglas de inferencia borrosas y calculando la salida del sistema sugeno a partir de los valores de salida obtenemos la siguiente función

$$u = \begin{cases} \frac{-1*1+0*1}{1+0} = -1 & 0 \leq m \leq 30 \\ \frac{-1*\left(\frac{8}{5} - \frac{m}{50}\right) + \left(\frac{m}{50} - \frac{3}{5}\right)*1}{\frac{8}{5} - \frac{m}{50} + \frac{m}{50} - \frac{3}{5}} = -\frac{11}{5} + \frac{m}{25} & 30 \leq m \leq 80 \\ \frac{-1*0+1*1}{0+1} = 1 & 80 \leq m \leq 255 \end{cases}$$

Si queremos obtener el punto donde más incertidumbre se puede encontrar, es igualando la función a 0, puesto que en ese punto exacto no se puede saber con certeza si el detector, tal y como está descrito está detectando una imagen nocturna o diurna.

$$u_{\text{mite}} = 0 = -\frac{11}{5} + \frac{m}{25} \implies m = 55$$

Por tanto, si estamos en condiciones en el que el shutter de la cámara es suficientemente alto, el controlador borroso se va a comportar como un sistema de umbral normal con valor 55, en el que, si la media de nivel de grises de la imagen de entrada supera este valor, se tomará como imagen diurna y en caso contrario como imagen nocturna.

Una vez comentada esta simplificación, se pasa a describir cómo se implementará el subsistema dentro del sistema general. Utilizaremos este sistema no en una única imagen, sino que se tomaran 10 imágenes de muestra para comprobar el estado de iluminación de la escena y estar seguros de la decisión final del sistema. Después de la toma de esas 10 muestras se realiza la media y umbraliza el resultado a la mitad del margen entre los dos centros que se han tomado de salida.

Si la salida ha resultado ser errónea (<-1), no se contará esa muestra a la hora de calcular la media.

Como podemos ver en la figura, asumimos que la imagen es diurna si el valor del acumulador es positivo y nocturna si éste es negativo.

Con este sistema se pueden tomar decisiones de manera más fiable que utilizando un umbral fijo, mediante el cual podemos equivocarnos en ocasiones.

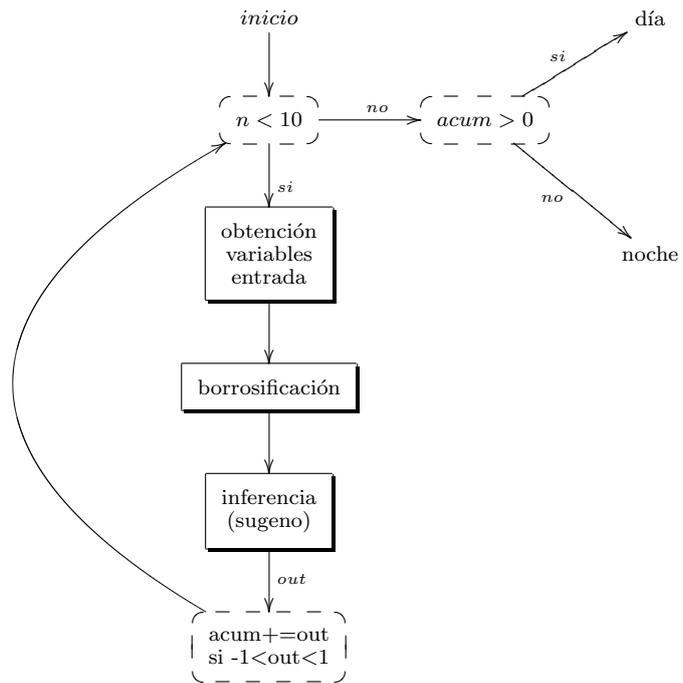


Figura 5.29: Esquema de implantación del detector en el sistema general

Capítulo 6

Resultados experimentales

En este capítulo pasamos a exponer los resultados obtenidos en las distintas partes del sistema presentadas anteriormente.

En primer lugar se muestran los resultados obtenidos en el clasificador, que es capaz de rechazar los usuarios de la base de datos, por lo cual se considera uno de los subsistemas fundamentales, al no dejar arrancar el coche si los usuarios detectados no están incluidos en la base de datos.

Después, se presentan los resultados del tratamiento de la iluminación, en los cuales se analizará, a la vista de los resultados, cuál es el mejor método o el método que se podría implementar en el prototipo final para tratar de mejorar el porcentaje de aciertos.

A continuación se presentarán los resultados obtenidos para el detector de gafas de sol, el cual, no es un sistema crítico pero influye en la detección final al proporcionar un sistema para evitar confusiones posibles debidas al cambio de apariencia que sufre una cara cuando lleva gafas de sol.

Por último, se presentarán los resultados obtenidos para el sistema que decide si se está de noche o de día, y los resultados de reconocimiento en el caso de las bases de datos nocturna y diurna, previa selección de la base de datos adecuada.

6.1. Resultados del clasificador

A continuación se presentan los resultados del clasificador. En primer lugar incluimos los resultados obtenidos de varios reconocimientos realizados sobre una secuencia en la que los candidatos que se sientan en el coche no están incluidos en la base de datos para ver cómo se comporta el clasificador en situaciones en las que tiene que rechazar candidatos. El resultado de una de las secuencias se presenta en la figura 6.3. Se han representado los resultados obtenidos utilizando la distancia euclídea y la de Mahalanobis como medidas de distancia en el clasificador.

Se han presentado los resultados en función de los tratamientos de la iluminación empleados, puesto que las distancias generadas al evaluar las imágenes en cada método son diferentes, y por tanto, el comportamiento del clasificador, al basarse en las distancias, también lo será.

En todos los casos de tratamiento de la iluminación, para la medida de distancia de Mahalanobis, se observa cómo el comportamiento del clasificador es similar, y en el caso de que los usuarios no estén incluidos en la base de datos, las distancias obtenidas, respecto del centro son muy superiores al caso de que los usuarios estén dentro de la base de datos.

Respecto de los resultados de la distancia euclídea, se aprecia como no todos los métodos son adecuados para realizar la labor de rechazo, puesto que con algunos métodos los usuarios son rechazados, mientras que con otros no.

Respecto al comportamiento del clasificador con usuarios que pertenecen a la base, podemos observar en las figuras 6.5 y 6.6, que pasan estas dos etapas de comprobación de distancias a los puntos característicos calculados del universo de decisión.

La primera etapa de clasificación realiza la mayor parte del filtrado para los usuarios que no están incluidos en la base, puesto que las distancias resultantes son muy superiores a las que se encuentran para usuarios incluidos en la misma. En el caso de la distancia de Mahalanobis, las diferencias entre las distancias de imágenes de usuarios internos comparadas con las generadas por los usuarios externos serán superiores, al tener una matriz de pesos que regula las distintas componentes del vector de características.

Las imágenes introducidas en la base de datos, son independientes del método del cálculo de distancia utilizado, no así del método de tratamiento de la iluminación.

Las características de las secuencias de entrenamiento y test son las siguientes:

- Cámara: Unibrain fire-i
- Resolución: 320x240
- Imágenes por segundo: 15
- Color: RGB

En cuanto a las características de la base de datos son las siguientes:

- Resolución: 60x90
- Número de imágenes por usuario: 30
- Número de usuarios: 9
- Número de autovectores: 35

El método de la cara recortada, posee una resolución inferior. La anchura se mantiene, pero la altura se adapta para que mantenga una relación de aspecto similar a las imágenes de entrada al algoritmo, esto es 60x35.

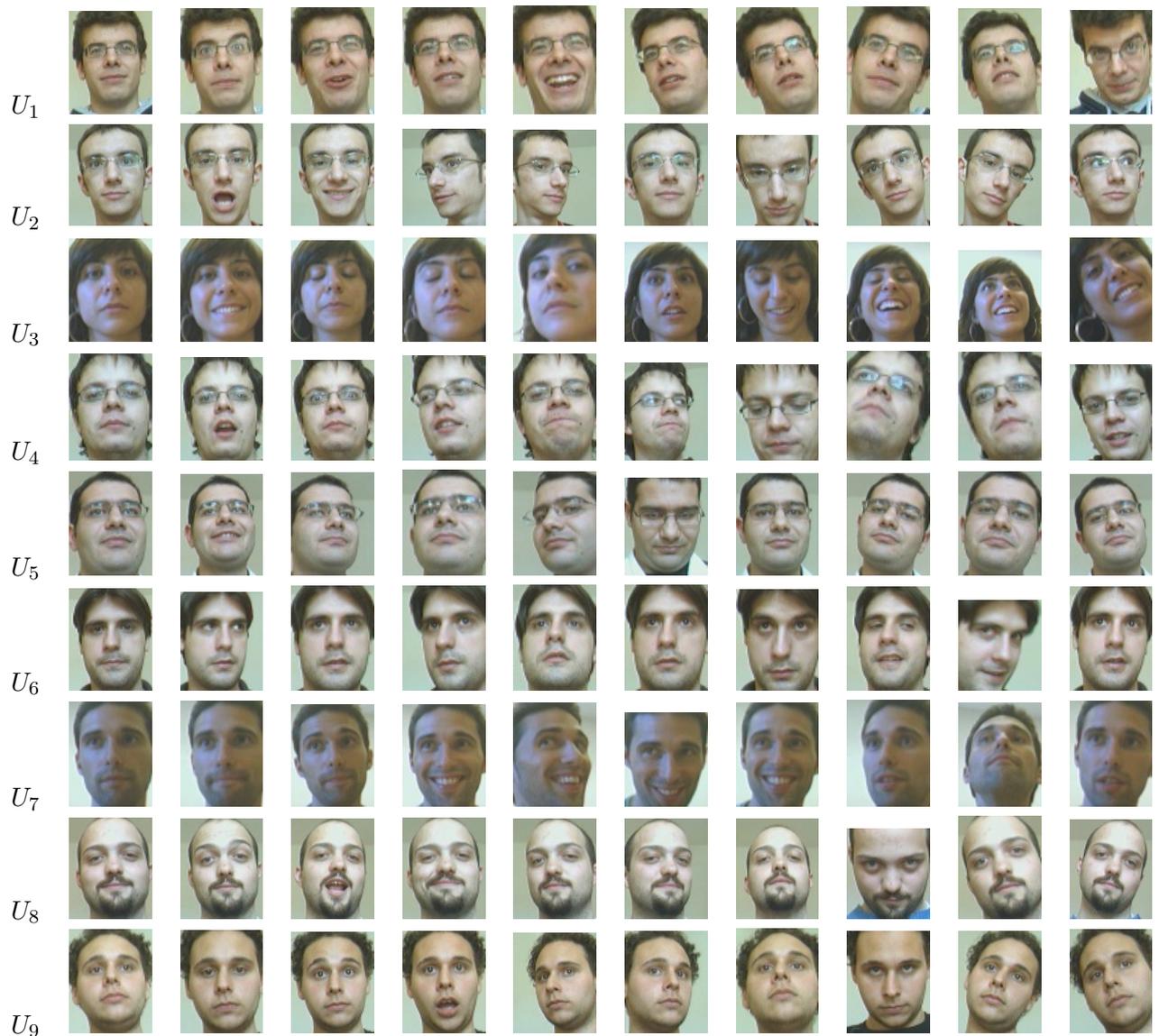


Figura 6.1: Muestras del conjunto de entrenamiento de la base de datos utilizada en condiciones de iluminación interior, sin tratamiento de la iluminación

6.1.1. Tablas

A continuación presentamos los resultados obtenidos con el clasificador a la hora de rechazar a un candidato. Se presentan los resultados de los reconocimientos realizados sobre una secuencia en la que ninguno de los usuarios se encuentra dentro de la base de datos y posteriormente, comparamos con las secuencias del clasificador normal.

Para asignar los umbrales en el clasificador, se han tomado algunas secuencias con candidatos externos a la base para asignar valores a los 2 primeros umbrales (Th_1, Th_2), y posteriormente

ir ajustando los 3 umbrales (Th_1, Th_2, Th_3) de forma más precisa con las secuencias con los que se ha generado la base de datos faciales.

Umbral	Original	Recortada	Ecualización	Máscaras	Retinex
Th_1	0	0	0	0	0
Th_2	0.5	1	1	0.7	0.4
Th_3	0.01	0.005	0.05	0.01	0.005

Tabla 6.1: Umbrales utilizados en el clasificador utilizando distancia euclídea

Umbral	Original	Recortada	Ecualización	Máscaras	Retinex
Th_1	1	1	0.5	1	0.2
Th_2	1.8	2.1	1.2	1	0.4
Th_3	0.03	0.01	0.05	0.01	0.005

Tabla 6.2: Umbrales utilizados en el clasificador utilizando distancia de Mahalanobis

Una vez se han asignado los umbrales en el clasificador, se pasan a probar con las secuencias con candidatos externos e internos a la base de datos. No se han realizado todas las pruebas con los de candidatos internos a la base, puesto que en el resto de casos los resultados del rechazo son los mismos.

	<i>Recortada</i>	<i>Original</i>	<i>Ecualización</i>	<i>Máscaras de iluminación</i>	<i>Retinex</i>
<i>Candidato externo 1</i>	100 %	0 %	0 %	100 %	0 %
<i>Candidato externo 2</i>	100 %	0 %	0 %	100 %	0 %
<i>Candidato externo 3</i>	100 %	0 %	0 %	100 %	0 %
<i>Candidato externo 4</i>	100 %	20 %	0 %	100 %	0 %
<i>Candidato interno 1 (prueba 2 de pruebas de iluminación)</i>	0 %	0 %	0 %	0 %	0 %

Tabla 6.3: Porcentaje de imágenes rechazadas con la distancia euclídea

Como se puede observar en los resultados, los porcentajes de rechazo varían en función de qué tratamiento de la iluminación se haya utilizado, aunque en el caso de las secuencias que contienen candidatos dentro de la base de datos, es independiente, puesto que los umbrales se ajustan para que el número de imágenes aceptadas sea máximo.

	<i>Recortada</i>	<i>Original</i>	<i>Ecualización</i>	<i>Máscaras de iluminación</i>	<i>Retinex</i>
<i>Candidato externo 1</i>	100 %	100 %	100 %	100 %	100 %
<i>Candidato externo 2</i>	100 %	100 %	100 %	100 %	100 %
<i>Candidato externo 3</i>	100 %	100 %	100 %	100 %	5 %
<i>Candidato externo 4</i>	100 %	100 %	100 %	100 %	10 %
<i>Candidato interno 1 (prueba 2 de pruebas de iluminación)</i>	0 %	0 %	0 %	0 %	0 %

Tabla 6.4: Porcentaje de imágenes rechazadas con distancia de Mahalanobis

Se puede ver además cómo la capacidad de rechazo, depende de la medida de distancia utilizada, no siendo tan efectiva la distancia euclídea para estos propósitos como la distancia de Mahalanobis.

Cuando se intentó obtener los umbrales del clasificador para los métodos de ecualización del histograma, Retinex y el algoritmo básico (sin tratamiento de la iluminación), con la distancia euclídea, se comprobó que no convergía al comportamiento deseado, esto es, rechazo de usuarios externos y aceptación de usuarios internos, como se puede ver en la tabla 6.3. En cambio, con el resto de métodos, la convergencia fue mejor. Respecto de la distancia de Mahalanobis, la convergencia a la hora de obtener los umbrales fue mucho más rápida que con la distancia euclídea.

6.1.2. Gráficas

Ahora pasamos a ver las gráficas correspondientes a los resultados expuestos en las tablas 6.3 y 6.4. No se incluyen los 4 casos de rechazo puesto que se comportan de forma muy parecida. En primer lugar comprobaremos gráficamente los umbrales establecidos y dónde quedan las distancias calculadas al centro del universo de decisión y, si pasa esta fase, a los centroides de las constelaciones formadas por las proyecciones de los distintos usuarios de la base de datos.

Recordamos que, si en alguna etapa del clasificador una imagen de las posibles es rechazada, el clasificador no continúa evaluando esa imagen y pasa a tomar la siguiente muestra, por tanto, si un usuario es rechazado en la primera etapa, no se evalúa la segunda, aunque para ver las características de las distancias a los centros de las constelaciones de los usuarios mostraremos las dos gráficas.

Se empieza mostrando algunas de las imágenes utilizadas para el reconocimiento (figura 6.2), a continuación las imágenes empleadas en la base de datos de caras utilizadas, y posteriormente se muestran las figuras que representan las distancias del reconocimiento para un rechazo y una aceptación.

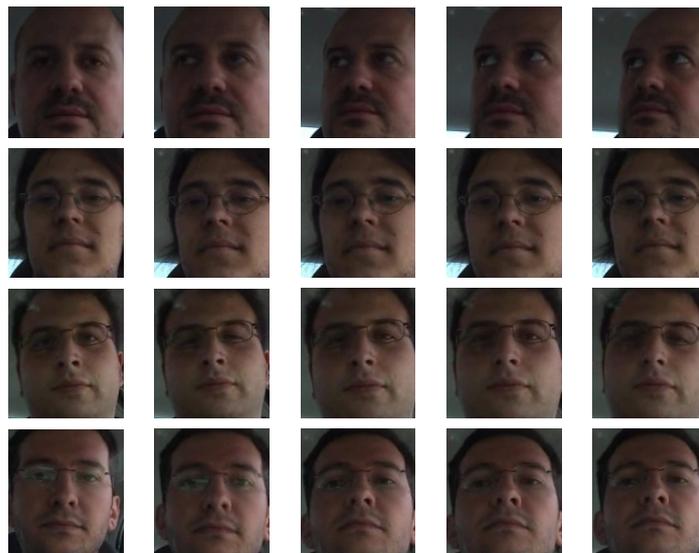


Figura 6.2: Muestras de los candidatos en la secuencia de prueba de rechazo de candidatos

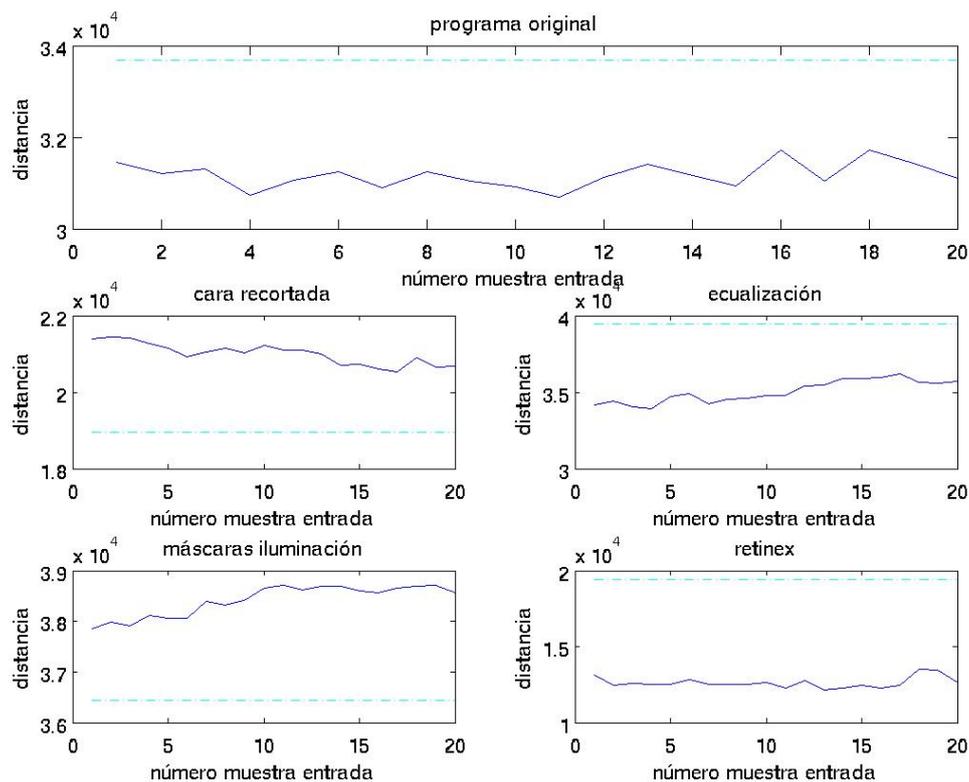
Algunas de las muestras utilizadas para el reconocimiento en la secuencia en la que se acepta el usuario, se encuentran en la figura 6.7.

En la figura 6.2, se puede comprobar que ninguno de los candidatos están incluidos en la base de datos, los cuales se pueden ver en la figura 6.1.

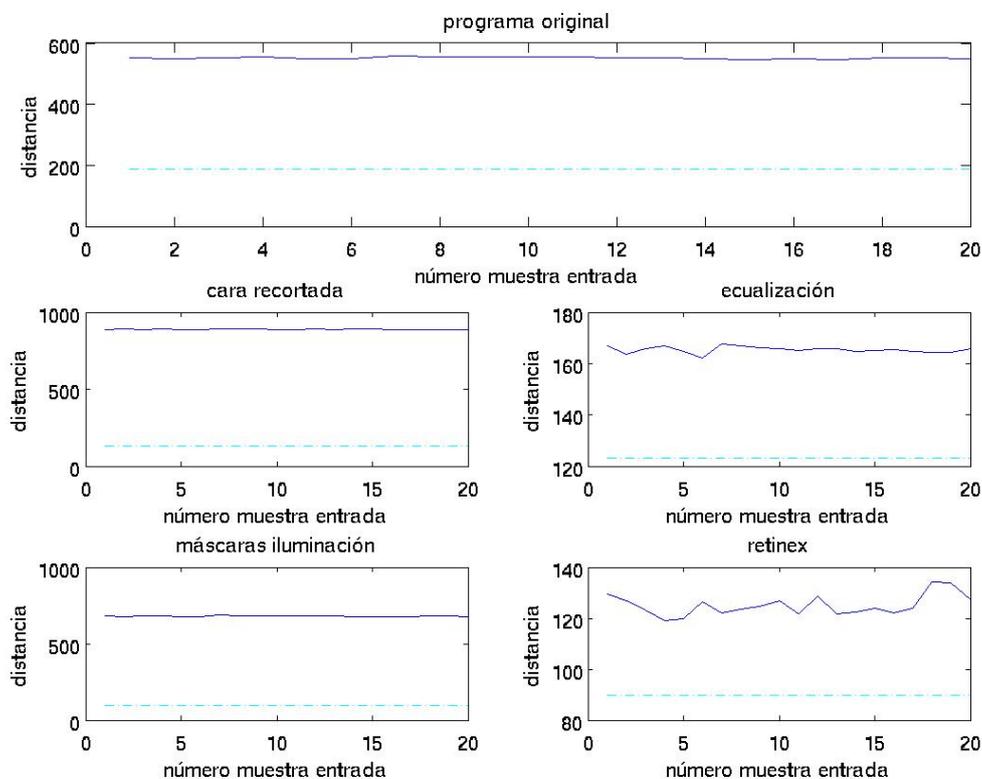
Respecto de las gráficas contenidas en la figura 6.3, se aprecia que la distancia no es capaz de rechazar usuarios externos en todos los casos, mientras que la distancia de Mahalanobis sí lo hace. En estas figuras, el rechazo se aprecia cuando se tiene el umbral por debajo de las distancias obtenidas.

En la mayoría de casos los usuarios quedan rechazados en la primera etapa con la distancia de Mahalanobis, pero los casos (no expuestos en las gráficas) que consiguen pasar, son rechazados en la segunda etapa de clasificación.

En cambio, en las figuras 6.5 y 6.6, vemos cómo se pasan las dos etapas de filtrado al quedar las distancias calculadas por debajo de los umbrales Th_1 y Th_2 , con lo que el usuario pasa a la etapa de reconocimiento.

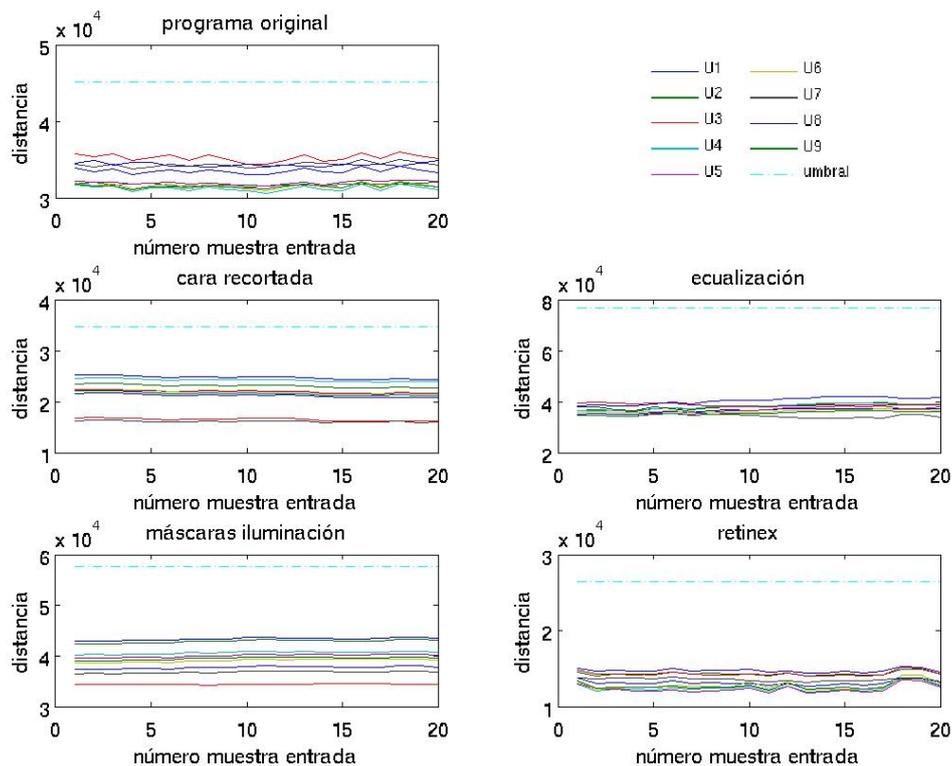


(a) Distancia euclídea

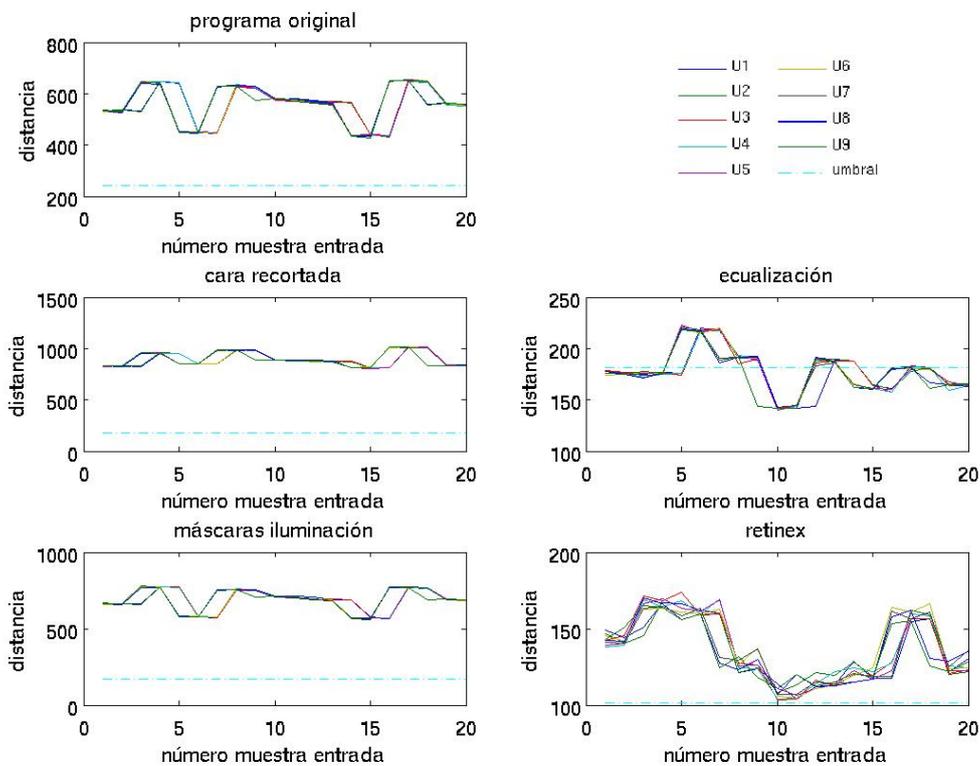


(b) distancia de Mahalanobis

Figura 6.3: Ejemplo de rechazo en la primera etapa del clasificador. Se representa el umbral en línea discontinua y distancia obtenida al centro del universo en línea continua.

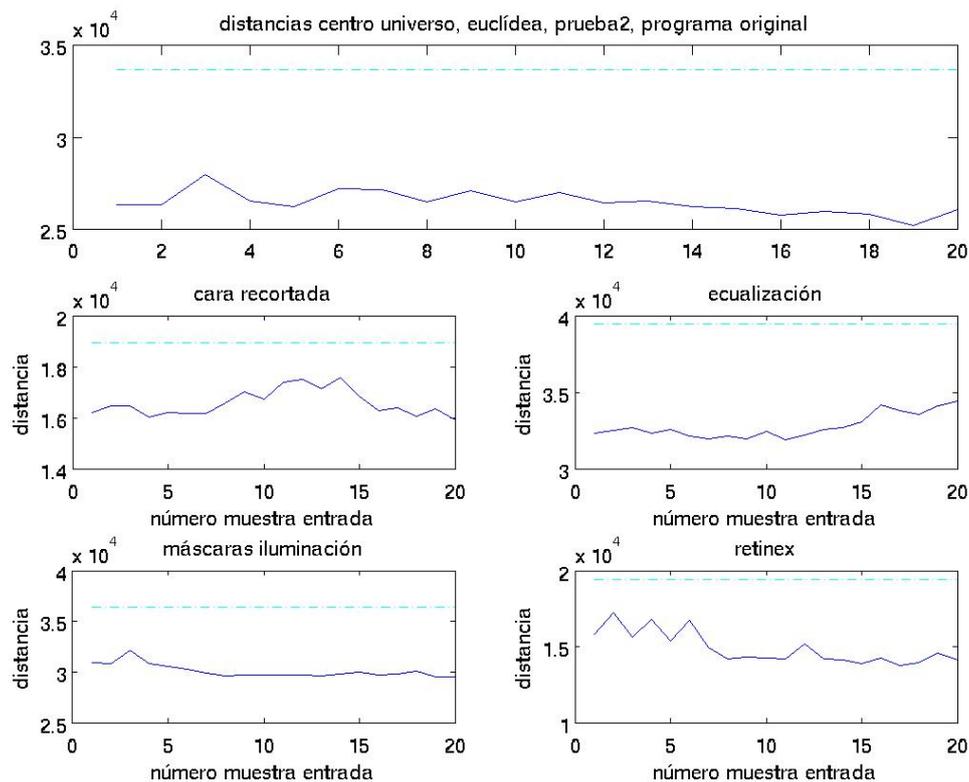


(a) Distancia euclídea

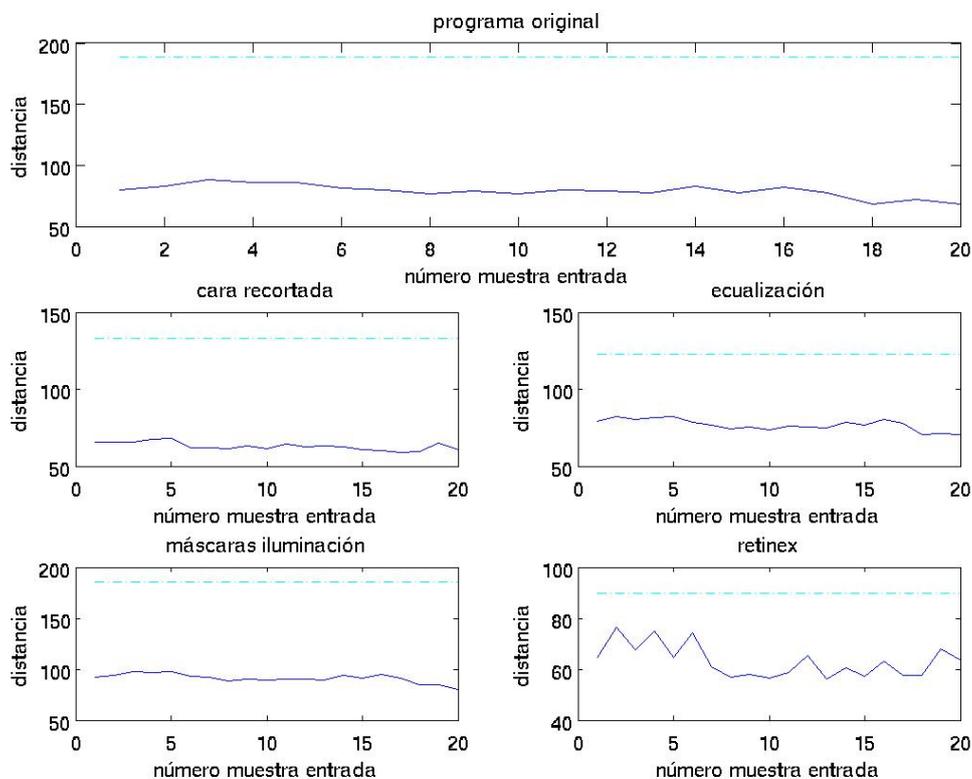


(b) Distancia de Mahalanobis

Figura 6.4: Distancias a los centros de las constelaciones del test, reconocimiento 1.

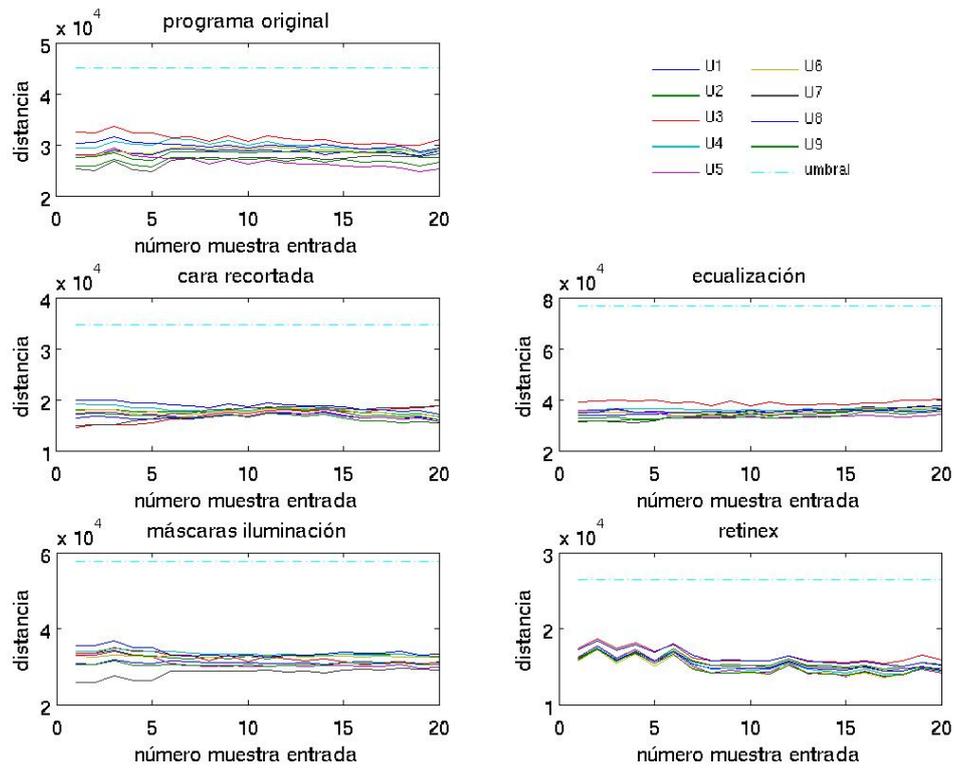


(a) Distancia euclídea

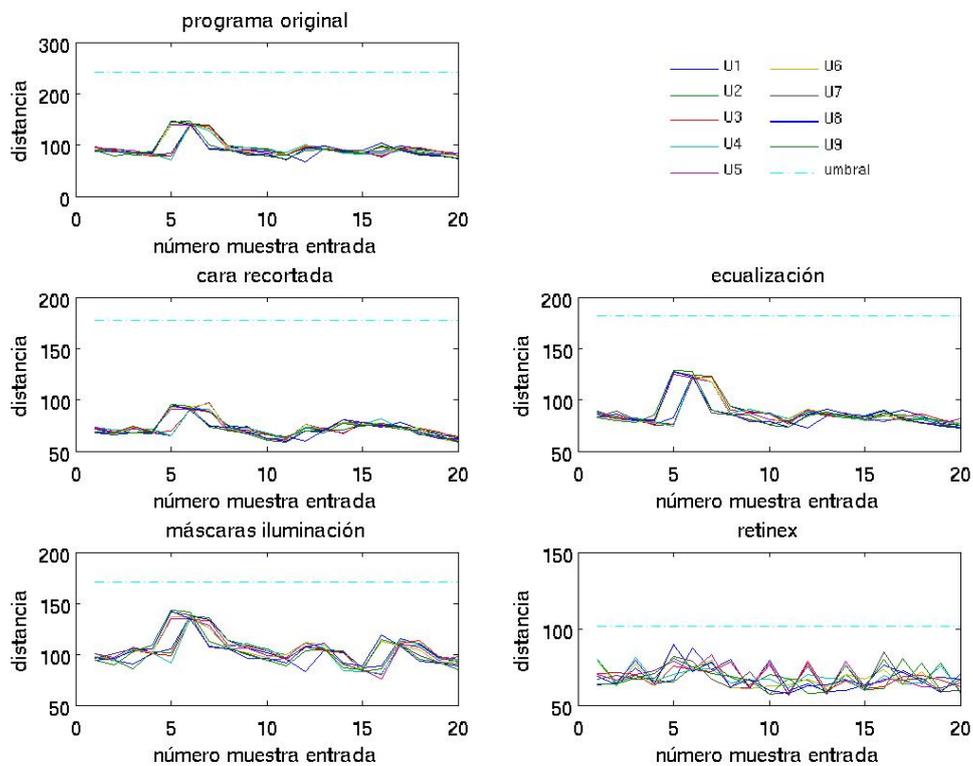


(b) Distancia de Mahalanobis

Figura 6.5: Ejemplo de aceptación por parte de la primera etapa del clasificador. Se representa el umbral en línea discontinua y distancia obtenida al centro del universo en línea continua.



(a) Distancia euclídea



(b) Distancia de Mahalanobis

Figura 6.6: Distancias a los centros de las constelaciones de la prueba

6.1.3. Conclusiones Parciales

De los resultados anteriormente mostrados, se extrae que es conveniente utilizar la medida de distancia de Mahalanobis para el clasificador. Con esto se consigue una separabilidad entre clases superior y por tanto una mayor capacidad de rechazo, si una muestra no pertenece a ninguna de las clases existentes en la base de datos, o si se encuentra entre clases de la base de datos, pero lo suficientemente lejos, esta separación introducida por la distancia de Mahalanobis hará que la muestra sea rechazada. La distancia euclídea puede ser una buena aproximación para realizar el reconocimiento, pero para el cometido que nos ocupa, no ha tenido la capacidad de separabilidad entre clases suficiente, requerida en la función que se necesita en este subsistema.

Se puede observar también que la segunda etapa de clasificación, es necesaria para el descarte de candidatos no incluidos en la base de datos que sean capaces de pasar la primera etapa de clasificación. Además, el hecho de descartar muestras de entrada, en el caso de que se pueda, implica un ahorro de tiempo de cómputo en la etapa final, donde se decide si el usuario es parecido solo a un usuario.

La distancia de Mahalanobis también se ha visto que funciona para todos los tratamientos de iluminación, mientras que la distancia euclídea no lo ha hecho, lo cual nos servirá además para implementar los 2 subsistemas de manera conjunta e incluso que se complementen. Por tanto los resultados que se presenten de aquí en adelante serán los correspondientes a la distancia de Mahalanobis.

6.2. Resultados de los tratamientos de la iluminación

En este apartado del capítulo de resultados se presentan los correspondientes a los diferentes métodos de tratamiento de iluminación probados para paliar este problema.

En primer lugar se verán los resultados de secuencias grabadas en condiciones de iluminación más o menos controladas, fuera de un coche, en el cuarto de trabajo, sobre un fondo regular, como es una pared. La iluminación sobre la cara es aproximadamente uniforme puesto que la iluminación se obtiene a partir de los fluorescentes, salvo las modificaciones que se produzcan debido a la luz que entre por la ventana.

En segundo lugar se prueban los resultados en condiciones reales, esto es, con el coche en el que se va a implantar el prototipo. En estas pruebas las condiciones de iluminación son más variables que en los anteriores casos, como se verá en las muestras de entrada comparándolas con las muestras recogidas en la base de datos. Estos cambios en la iluminación son fruto de, estando en la misma hora del día, variar la orientación del coche, para simular el hecho de cambiar el coche de aparcamiento, puesto que en estos casos la orientación es completamente aleatoria, y la hora de tomar y dejar el vehículo es también algo aleatoria, aunque se suele aparcar en los mismos sitios. Las condiciones ambientales en el exterior son tremendamente variables, puesto que se pueden dar condiciones de clima nublado, despejado, dependiendo de la hora del día, etc, y los impactos sobre la iluminación de la escena son impredecibles.

6.2.1. Secuencias de prueba en condiciones de iluminación controladas

A continuación se presentan tablas con los resultados obtenidos en secuencias tomadas en el laboratorio de trabajo. En este lugar, las condiciones de iluminación son aproximadamente constantes, al estar iluminada la escena por tubos fluorescentes. Estas condiciones de iluminación pueden variar un poco en caso de que se hayan abierto más o menos las ventanas a distintas horas del día.

Las características de la base de datos son las mismas que las que se describieron en los resultados correspondientes al rechazo de candidatos externos, aunque recordaremos aquí.

Para el entrenamiento de la base de datos se grabó a 9 posibles usuarios, y posteriormente, se grabaron 4 secuencias de prueba, para comprobar, pasado cierto tiempo y variando mínimamente las condiciones de iluminación (imágenes tomadas por la mañana y los secuencias tomados por la tarde), cómo varían los resultados de reconocimiento.

Las características de las secuencias se pasan a describir a continuación. La resolución de estos vídeos es de 320x240, grabados con la cámara de pruebas fire-i a 15 fotogramas por segundo (si se querían obtener secuencias al doble de resolución, la tasa de adquisición bajaba a la mitad, lo cual era incómodo para los usuarios puesto que los movimientos que debían de hacer eran bastante más lentos).

Respecto a las características de la base de datos, 30 han sido el número de imágenes tomadas por cada usuario permitido, en color, y el tamaño utilizado para el recorte de las caras ha sido de 60x80. Diez de las imágenes incluidas en la base de datos de cada usuario, sin tratamiento de la iluminación, se encuentran en la figura 6.1. En el caso particular de la base de datos de la cara recortada, la resolución es de 60x35.

6.2.1.1. Tabla de resultados con el conjunto de secuencias de prueba

A continuación mostramos la tabla de resultados para las distintas secuencias de prueba, comprobando primero la bondad de la base de datos, y a continuación, las pruebas con secuencias diferentes a las utilizadas en la base de datos, intentando hacer una aproximación paulatina a las condiciones que nos encontraremos con la realidad.

Para obtener los resultados de estos tratamientos se ha deshabilitado la función del clasificador, para obtener con claridad todas las distancias a los puntos de la base de datos y no descartar resultados por malos que estos sean. En pruebas realizadas con el clasificador habilitado, se ha podido observar cómo algunas de las muestras han sido descartadas en la primera y en la segunda etapa, sobre todo en la última. Las muestras descartadas fueron las correspondientes a la distorsión en la apariencia de la imagen, debido a la iluminación fuerte en algunas zonas y generación de sombra en otras.

Para deshabilitar el clasificador se utilizarán como umbrales para el clasificador $Th_1 = 100$, $Th_2 = 100$ y $Th_3 = 0$, para tener margen de 100 veces sobre la distancia máxima de las que nos limitan para que se entre en la última etapa del clasificador, y Th_3 hacerlo 0, para que el usuario detectado sea siempre el usuario correspondiente con el que ha generado la mínima distancia.

	<i>Recortada</i>	<i>Original</i>	<i>Ecualización</i>	<i>Máscaras de iluminación</i>	<i>Retinex</i>
<i>Usuario 1</i>	90 %	100 %	100 %	100 %	5 %
<i>Usuario 2</i>	100 %	100 %	100 %	100 %	30 %
<i>Usuario 3</i>	100 %	100 %	100 %	100 %	95 %
<i>Usuario 4</i>	100 %	100 %	100 %	100 %	100 %
<i>Usuario 5</i>	100 %	100 %	40 %	100 %	65 %
<i>Usuario 6</i>	100 %	100 %	100 %	100 %	100 %
<i>Usuario 7</i>	100 %	100 %	100 %	100 %	85 %
<i>Usuario 8</i>	100 %	100 %	100 %	100 %	75 %
<i>Usuario 9</i>	0 %	100 %	100 %	100 %	100 %
<i>Prueba 1</i>	0 %	5 %	90 %	0 %	100 %
<i>Prueba 2</i>	100 %	95 %	10 %	95 %	0 %
<i>Prueba 3</i>	95 %	100 %	100 %	100 %	15 %
<i>Prueba 4</i>	0 %	0 %	0 %	0 %	55 %
<i>Media imágenes reconocidas correctamente</i>	76 %	86 %	80 %	84 %	63 %

Tabla 6.6: Resultados obtenidos con distancia de Mahalanobis en los secuencias grabadas en la sala

Como se puede observar en la tabla 6.6, los métodos que mejor resultados dan son el original (sin tratamiento alguno de la iluminación), seguido de cerca por el tratamiento de las máscaras de iluminación aplicadas a la base de datos.

También podemos observar que el peor método para realizar los reconocimientos de forma

independiente, es el de Retinex, aunque en pruebas donde todos los métodos fallan (reconocimientos entre 0 y 50%), Retinex da un porcentaje de seguridad superior al 50%. El problema de esto es que no se ha encontrado una forma de saber a priori cuándo un método u otro está fallando, aunque el clasificador en funcionamiento elimina las muestras con poca calidad o muy deformadas por una iluminación excesiva.

6.2.1.2. Gráficas

Por último, mostramos las gráficas para el reconocimiento de la segunda secuencias de prueba, al ser la más representativa de este conjunto de secuencias de prueba. En las secuencias, como se puede ver en la tabla 6.6, los resultados no son tan buenos como los obtenidos con las secuencias utilizadas para generar la base de datos. Esto es debido al conjunto de imágenes capturadas para el reconocimiento, que es algo diferente del conjunto de entrenamiento. En la secuencia correspondiente a la prueba 2, es el 7^o.

La media es utilizada en el clasificador como medida supervisora del resultado obtenido por el vecino más cercano con criterio de mayoría. En estas gráficas se muestra el valor de media obtenido con cada método y se remarcará el método por el que se ha obtenido la media mínima para el método y usuario correspondiente.

Cotejando los resultados de las tablas, los de las gráficas y las medias, vemos que son coherentes, puesto que allá donde en las tablas no se observan coincidencias significativas, las gráficas de distancias a lo largo del reconocimiento y las de distancias mínimas medias son coherentes.

A continuación se muestran los resultados de la prueba 2.

En primer lugar, se muestran en la figura 6.7, 3 de las 20 muestras utilizadas para el reconocimiento, tratadas con los distintos métodos propuestos. La similitud de estas imágenes no es tan alta como lo podrían ser las tomadas de la propia secuencia, pero son lo suficientemente parecidas para que el sistema no las confunda con otro usuario.

En segundo lugar se muestran en la figura 6.8 los resultados de las distintas distancias obtenidas a lo largo del tiempo, para cada tratamiento de iluminación, para la distancia de Mahalanobis. Aquí podremos ver gráficamente en qué fotogramas se ha acertado o ha fallado el sistema, con quién de todos los usuarios se ha confundido, y si el margen sobre los demás usuarios es suficiente o es tan reducido que se puede confundir en ese caso con facilidad.

Se puede apreciar en la gráfica correspondiente a la cara recortada que el margen sobre el resto de usuarios es bastante grande, mientras que en el caso de no tratar la iluminación o probar con el método de las plantillas de iluminación, el margen se reduce, y siendo nulo éste en los métodos de ecualización de histograma y Retinex.

En cuanto a los resultados obtenidos sobre la media de las distancias mínimas a los distintos usuarios, mostrados en la figura 6.9, vemos que, para el caso de la distancia euclídea, se confirman los resultados obtenidos por el clasificador de vecino más cercano con criterio de mayoría, siendo en este caso el usuario n^o 7 el que aparece en el vídeo de la prueba 2.

A la vista de los resultados obtenidos en las otras 3 secuencias de prueba, se puede deducir por qué falla en ciertas situaciones. En el caso de que las imágenes tomadas en la base de datos sean claras, y sin embargo, luego el vídeo de prueba sea más oscuro, tenemos problemas con el no tratamiento de la iluminación, con la cara recortada y con las máscaras de iluminación. En cambio, si las imágenes de la base de datos son oscuras y las de los vídeos de prueba también, estos últimos métodos son los que aciertan, y los que fallan son los métodos de ecualización de histograma y el de Retinex.

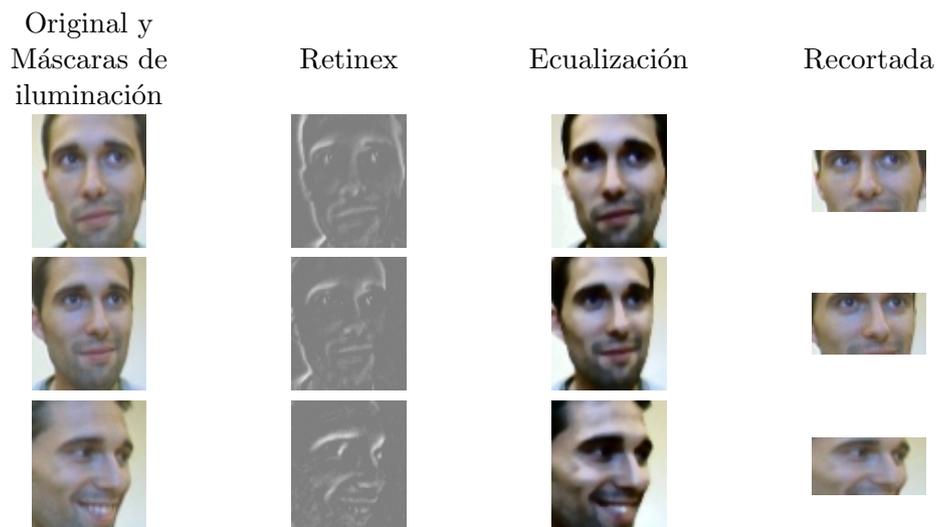


Figura 6.7: Muestras tomadas para reconocimiento del vídeo de prueba 2

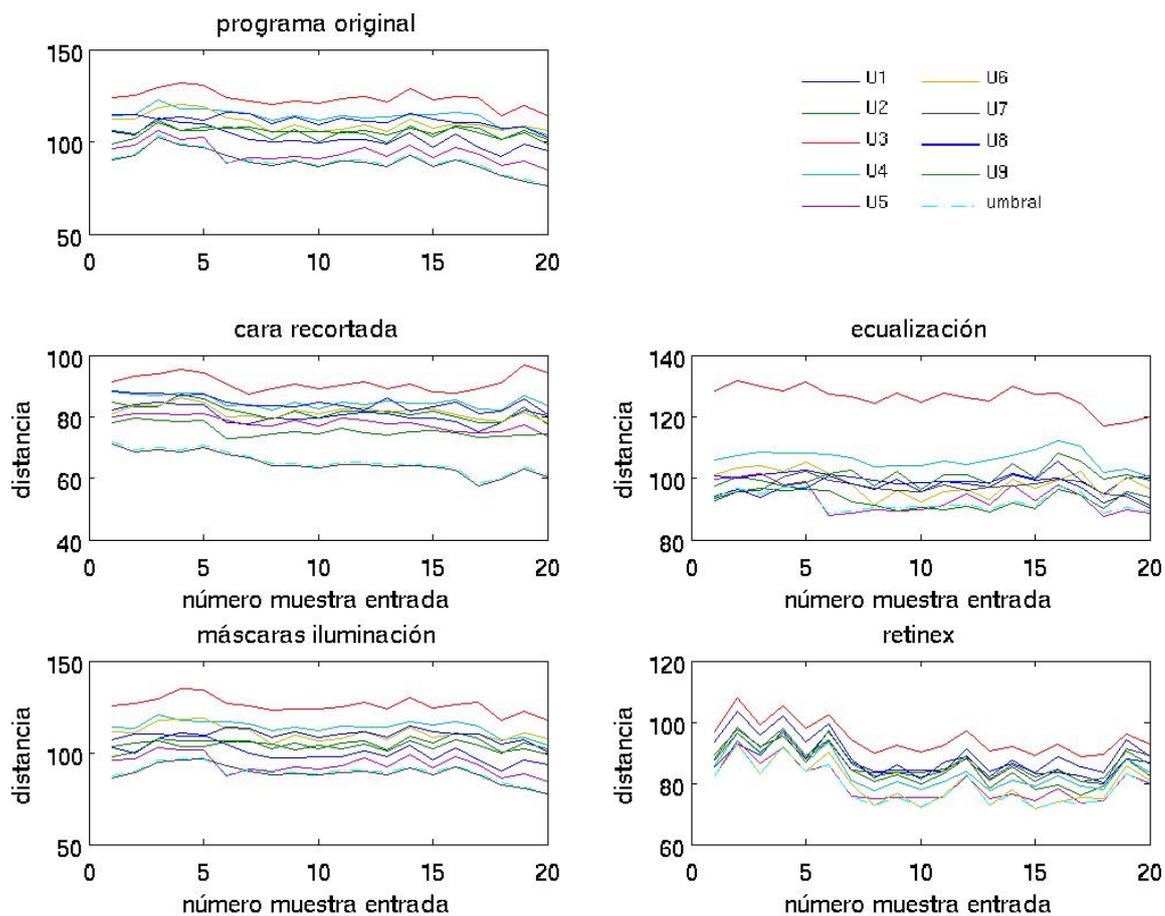


Figura 6.8: Distancias mínimas a usuarios de la BB.DD. en la prueba 2

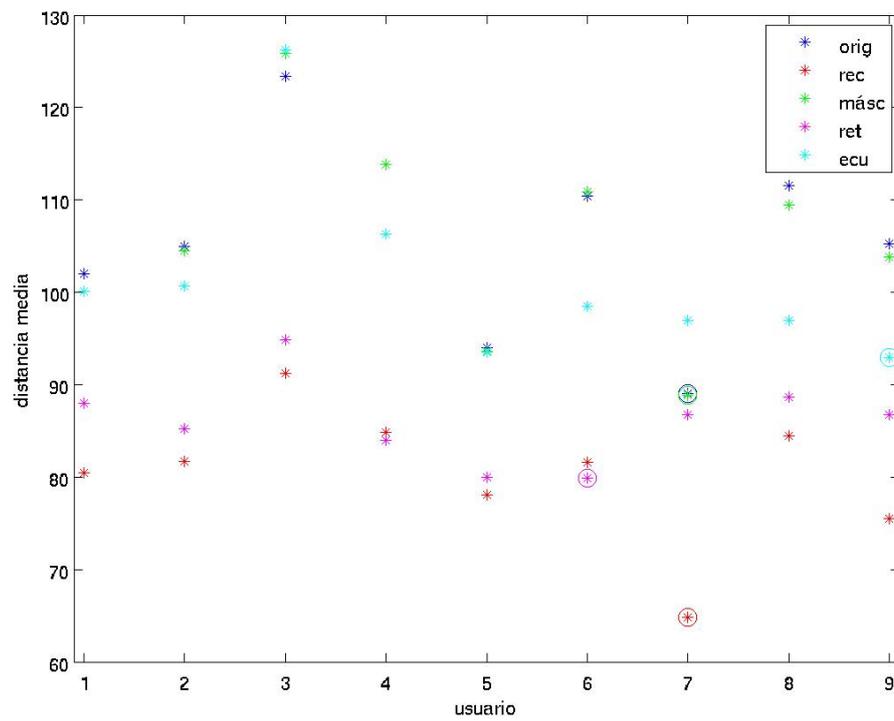


Figura 6.9: Distancias mínimas medias a usuarios de la BB.DD. en la prueba

6.2.2. Vídeos de prueba en condiciones de iluminación exteriores

A diferencia del conjunto de secuencias de prueba anteriores, la variabilidad de las condiciones de iluminación en un entorno real es bastante mayor, puesto que a distintas horas del día, la posición del sol en el cielo es distinta, por tanto la luz entrará por distintas ventanas e irá generando luces y sombras de forma diferente. A la misma hora del día aproximadamente, las condiciones de iluminación también varían si la orientación del coche cambia. Si se tienen estas dos variables en cuenta, la variabilidad en la apariencia de un mismo rostro es muy alta y puede ser importante a la hora de realizar un reconocimiento correcto.

En la figura 6.10 se presentan varias muestras de la base de datos utilizada para realizar los reconocimientos en estas condiciones. Las muestras de cada usuario se han tomado de una sola de las secuencias, sin tener en cuenta el hecho de que cada una tiene en realidad 2 vídeos tomados desde perspectivas diferentes, esto es, la base de datos ha sido construida únicamente a partir de una de las perspectivas.

Las características de las secuencias de entrenamiento y test son las siguientes:

- Cámara: Bastler Scout
- Resolución: 640x480 / 1200x480
- Imágenes por segundo: 15 / 30
- Color: escala de grises

En cuanto a las características de la base de datos son las siguientes:

- Resolución: 80x110
- Número de imágenes por usuario: 30
- Número de usuarios: 5
- Número de autovectores: 35

En cuanto a la figura 6.11 se muestran distintas imágenes de las secuencias del conjunto de prueba. De estos vídeos, excepto del 6, se han tomado las fotografías para realizar la base de datos. Vemos que en algunos casos las caras no están recortadas correctamente o la cara ha detectado como cara un falso positivo y se capta como cara alguna parte del coche. También podemos apreciar las diferencias de iluminación entre los distintos vídeos de una misma secuencia, e incluso, dentro del mismo vídeo cuando se ajusta el valor de ganancia y/o exposición de la cámara.

Las secuencias de vídeo fueron tomados por un sistema estéreo, por tanto, además de las variaciones de iluminación en distintas orientaciones, podemos comprobar las diferencias que se pueden tener al variar la perspectiva desde la que se toman las imágenes, en un principio este aspecto podría no parecer importante, pero así como la variabilidad en los gestos es importante,



Figura 6.10: Muestras del conjunto de entrenamiento de la base de datos utilizada en condiciones de iluminación exterior, sin tratamiento de la iluminación

también lo es en la perspectiva, pues la proyección de las imágenes varía y por tanto la apariencia de la proyección en las imágenes de entrada también. En principio esta comprobación no es necesaria, puesto que este sistema es monocular, pero si se diese el caso de que la cámara se desplazase, podemos comprobar cómo el sistema podría seguir funcionando.

El conjunto de vídeos con el que se han realizado pruebas es más amplio que el utilizado para condiciones de iluminación controlada, con lo que los resultados, estarán estadísticamente más cerca de resultados reales.

6.2.2.1. Tabla de resultados del conjunto de secuencias de prueba

A continuación se presentan las tablas de resultados de todas las secuencias que se grabaron para probar el sistema en condiciones de iluminación nocturnas. Como en el caso anterior, se ha deshabilitado el clasificador, para ver cómo son los resultados, por malos que éstos sean. Además, queremos ver cómo funcionan los sistemas de manera independiente, por tanto esta es la mejor manera.

Las secuencias cuyo segundo número sea 1, corresponden al vídeo de la cámara izquierda (según se mira de frente) del sistema estéreo que se utilizó para grabar, y el número 2 corresponde a la cámara derecha. El tercer dígito corresponde con el número de reconocimiento de la secuencia. Si solo se realiza un reconocimiento, este número se omite, mientras que si hay más de uno, se empieza desde el primero a numerar hasta el número de reconocimientos incluidos en ese vídeo. Sabiendo esto pasamos a describir algunos detalles que podemos encontrar en las tablas.

En las tablas se puede observar cómo, efectivamente el cambio de perspectiva influye en cierta manera en el reconocimiento de la imagen. Por ejemplo en el vídeo 6, en el reconocimiento 1 se ve cómo afecta esa variación al número de imágenes correctamente reconocidas.

Si las condiciones de iluminación son "extrañas", por ejemplo, si se introducen reflejos que

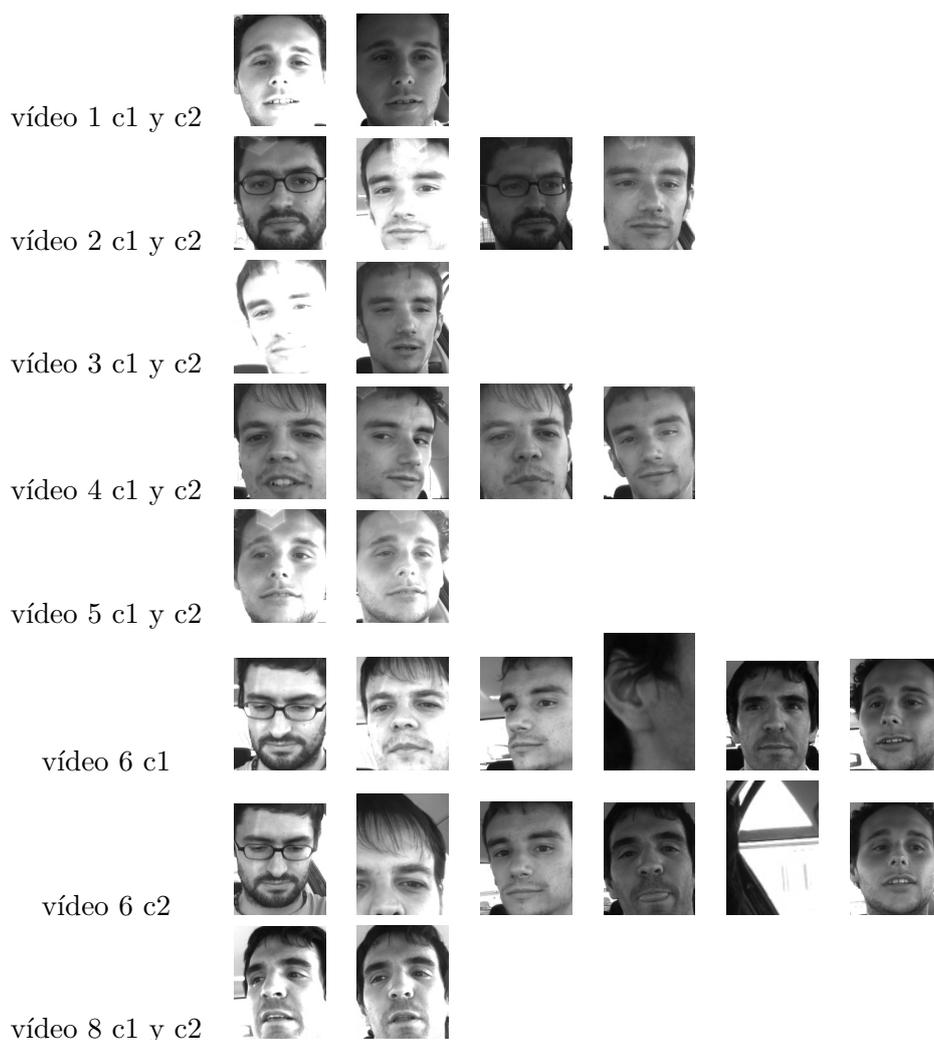


Figura 6.11: Ejemplos de caras contenidas en los distintos vídeos del coche

no se introducen normalmente en las imágenes (figura 6.11, video5 c1, video3 c1 y video 2 c1.2) y por tanto no están incluidos en la base de datos, el resultado no será bueno, puesto que las tasas de reconocimiento serán bajas. Este tipo de anomalías posiblemente pueden filtrarse en el clasificador, el cual tiene el poder de rechazar este tipo de imágenes. Si la iluminación es muy dispar, el sistema tampoco la podrá corregir adecuadamente, con lo cual, las tasas de reconocimiento bajarán. Ejemplos de esto los tenemos en la secuencia 1 y en el vídeo de la secuencia 3 grabado con la cámara 1 entre otros, como se puede ver en las muestras de los vídeos mostradas en la figura 6.11.

Pasamos a comentar los resultados de la tabla siguiente. Se aprecia cómo, en una gran parte de los casos, el método de tratamiento de iluminación de ecualización de histograma, a diferencia de en el caso de iluminación controlada, es el mejor método en la mayoría de casos. Peores resultados dan los métodos de cara recortada, sin tratamiento de la iluminación y el de máscaras de iluminación.

Se puede considerar que un método funciona perfectamente si en la mayoría de casos se sobrepasa el 80 % de aciertos, esto es, 16/20 reconocimientos correctos. En el caso de la ecualización del histograma, llegamos a un reconocimiento medio aproximado del 80 %, lo cual está dentro de los objetivos que se han fijado. El resto de métodos no presentan una mejora significativa salvo

Retinex, cuya mejoría es apreciable, pero no suficiente, al equipararse solo al resto de métodos.

	<i>Recortada</i>	<i>Original</i>	<i>Ecuilización</i>	<i>Máscaras de iluminación</i>	<i>Retinex</i>
<i>Video1 c1</i>	0 %	10 %	90 %	10 %	0 %
<i>Video1 c2</i>	95 %	5 %	15 %	5 %	5 %
<i>Video2 c1 r1</i>	0 %	100 %	95 %	100 %	0 %
<i>Video2 c1 r2</i>	0 %	0 %	20 %	45 %	5 %
<i>Video2 c2</i>	100 %	100 %	100 %	100 %	35 %
<i>Video3 c1</i>	0 %	0 %	100 %	0 %	0 %
<i>Video3 c2 r1</i>	100 %	100 %	100 %	100 %	35 %
<i>Video3 c2 r2</i>	100 %	95 %	100 %	95 %	100 %
<i>Video4 c1 r1</i>	100 %	35 %	45 %	30 %	100 %
<i>Video4 c1 r2</i>	100 %	100 %	100 %	100 %	100 %
<i>Video4 c2 r1</i>	100 %	95 %	100 %	100 %	100 %
<i>Video4 c2 r2</i>	100 %	100 %	100 %	100 %	100 %
<i>Video5 c1</i>	20 %	0 %	45 %	30 %	20 %
<i>Video5 c2</i>	0 %	0 %	15 %	0 %	0 %
<i>Video6 c1 r1</i>	5 %	15 %	90 %	10 %	5 %
<i>Video6 c1 r2</i>	100 %	100 %	100 %	100 %	100 %
<i>Video6 c1 r3</i>	100 %	100 %	100 %	100 %	100 %
<i>Video6 c1 r4</i>	40 %	0 %	80 %	0 %	20 %
<i>Video6 c1 r5</i>	100 %	100 %	100 %	100 %	100 %
<i>Video6 c2 r1</i>	65 %	85 %	90 %	100 %	0 %
<i>Video6 c2 r2</i>	0 %	100 %	30 %	85 %	40 %
<i>Video6 c2 r3</i>	100 %	55 %	90 %	60 %	100 %
<i>Video6 c2 r4</i>	0 %	0 %	80 %	0 %	10 %
<i>Video6 c2 r5</i>	0 %	0 %	100 %	0 %	0 %
<i>Video8 c1</i>	75 %	90 %	100 %	80 %	30 %
<i>Video8 c2</i>	0 %	100 %	100 %	95 %	35 %
<i>Media imágenes reconocidas correctamente</i>	54 %	57 %	80 %	59 %	46 %

Tabla 6.8: Resultados obtenidos mediante distancia de Mahalanobis, vídeos coche

6.2.2.2. Gráficas

A continuación pasamos a ver algunos de los resultados de las secuencias. En primer lugar, en la figura 6.12, vemos algunas muestras de el vídeo 6 grabado con la cámara 1 y en el reconocimiento 1, en el cual, basándonos en la tabla 6.8, el método que mejor funciona de todos los aplicados es el de ecualización, seguido con unos porcentajes bastante más bajos de los métodos de máscaras de iluminación y en el que no se trata de ninguna forma la iluminación. Los peores métodos posibles son los de cara recortada y Retinex.

En la figura 6.13, vemos cómo los márgenes más amplios entre el usuario detectado y los demás, se dan en 3 métodos de tratamiento de la iluminación: sin tratamiento, con ecualización de histograma y con máscaras de iluminación.

Respecto a la figura 6.14, vemos cómo se confirman a través de las distancias medias las medidas obtenidas.

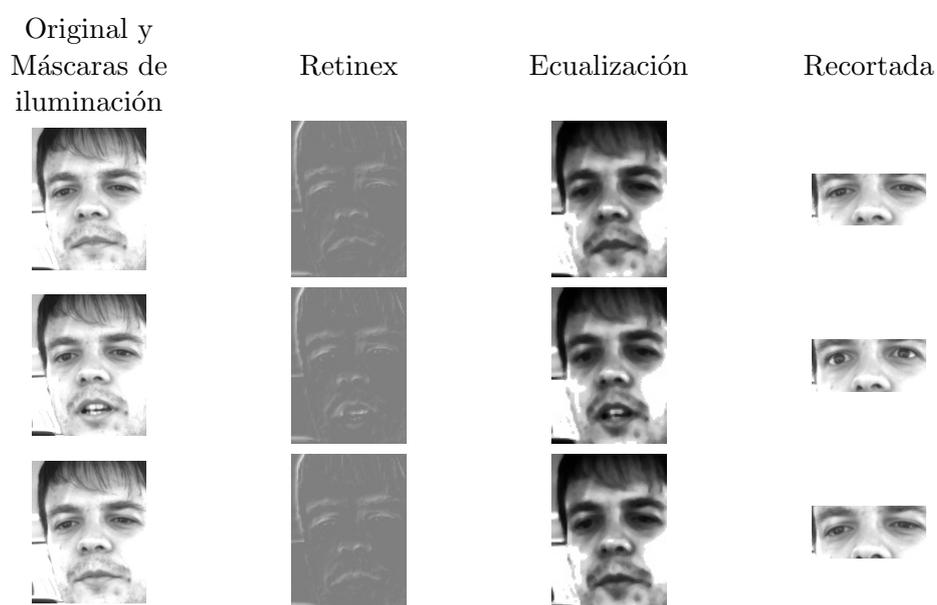


Figura 6.12: Muestras tomadas para el reconocimiento 2, vídeo 6, cámara izquierda

Si cambiamos la perspectiva, la iluminación captada por la cámara contraria es diferente. Por tanto, el reconocimiento se verá afectado, empeorando, como se puede apreciar en la tabla 6.8. En ella, podemos ver los resultados de la cámara 1 son mejores que en la cámara 2, puesto que las muestras de entrenamiento se tomaron de la cámara 1. En esta situación de cambio de perspectiva, los resultados arrojados por el tratamiento de ecualización de histograma son los mejores (en comparación con los demás métodos), en la mayoría de los casos.

Si variamos las características de iluminación de la cara, en la mayoría de casos el porcentaje de aciertos baja drásticamente, como puede observarse en las secuencias 1, 2 y 5.

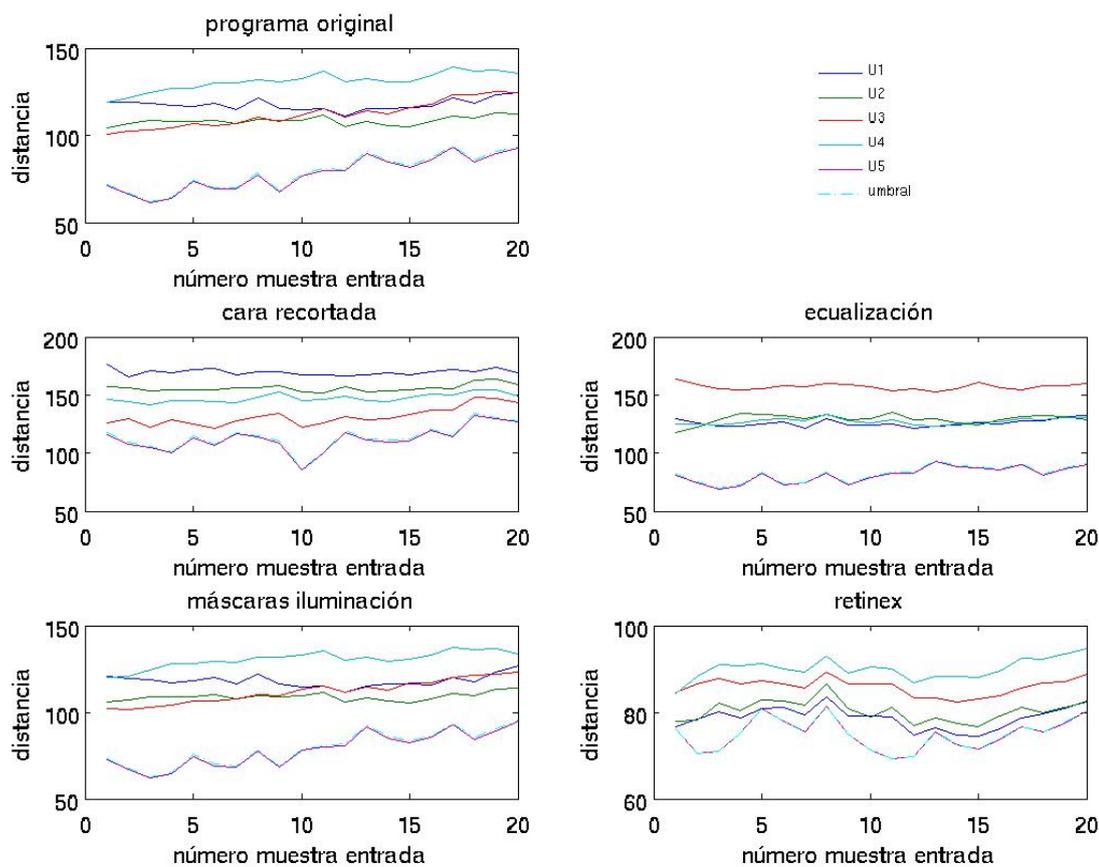


Figura 6.13: Distancias generadas por el reconocimiento con distintos tratamientos de la iluminación, Vídeo de coche 6 cámara 1 reconocimiento 2

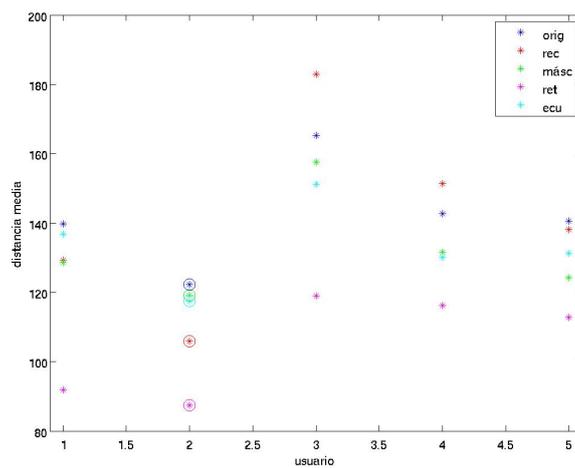


Figura 6.14: Medias de distancia mínima media a los usuarios, con los tratamientos de iluminación analizados, para el vídeo 6, cámara 1 y reconocimiento 2

6.2.3. Tiempos de ejecución

A continuación mostramos una tabla con los tiempos de ejecución medios obtenidos para los distintos casos, en el peor caso que hemos podido probar, esto es, en el caso de que el clasificador no rechace ninguna muestra y en el caso de que el número de usuarios sea el mayor posible, en nuestro caso, 9.

<i>Distancia</i>	<i>Recortada</i>	<i>Original</i>	<i>Ecuilización</i>	<i>Máscaras</i>	<i>Retinex</i>
<i>Euclídea</i> <i>tiempo por frame</i> <i>clasificador</i>	11,3 - 12,3 ms	17.6 - 22 ms	18-19.6 ms	71.6-89 ms	19.5 - 25 ms
	11.7 ms	17.9 ms	18.2 ms	93.68 ms	29.6 ms
<i>Mahalanobis</i>	368.2 ms	862.3 ms	899.2 ms	6126.8 ms	926.7 ms

Tabla 6.9: Tiempos de ejecución del sistema para distintos tratamientos de iluminación

Debido a la implementación del sistema del sistema de medida de euclídea, tenemos 2 etapas, una en la que se calculan las distancias para cada imagen de entrada, y otra posterior en la que, se pone en funcionamiento el clasificador. En cambio, en la implementación de la distancia de Mahalanobis, al necesitar los todos los frames del reconocimiento para calcular la matriz de covarianza, el cálculo de distancias y la etapa de clasificador se hace todo de forma seguida y consume más tiempo, por lo que se podrían perder frames, pero hay que tener en cuenta que ya se ha recopilado toda la información necesaria para el reconocimiento, por tanto, ya no importa que se pierdan unos frames, sino que el resultado del reconocimiento sea correcto.

Evidentemente, estos tiempos de ejecución son proporcionales al número de usuarios almacenados en la base de datos, por lo que si este es menor, el tiempo se reducirá, si, como en el caso de enriquecimiento masivo de las máscaras de iluminación, el tiempo se dispara, al crecer proporcionalmente el número de comparaciones a realizar.

Entre la distancia euclídea y la de Mahalanobis, se aprecian ciertas diferencias en el tiempo de cálculo. Si dividimos el tiempo total entre el número de muestras adquiridas, nos queda la siguiente tabla

<i>Recortada</i>	<i>Original</i>	<i>Ecuilización</i>	<i>Máscaras</i>	<i>Retinex</i>
18,41	43,12	44,96	306,34	46,33

Tabla 6.10: Tiempos de proceso equivalentes por frame con distancia de Mahalanobis

En ésta podemos apreciar claramente las diferencias de tiempo de cómputo. También podemos comprobar que los tratamientos más básicos de iluminación aplicados introducen una carga extra entre 1 y 3 ms. Los tiempos de cómputo son altos comparados con los dados por la distancia euclídea, pero hay que tener en cuenta el tamaño de las imágenes utilizadas para el reconocimiento y además, el hecho de que hay que calcular matrices de covarianza, invertirlas y multiplicarlas por las proyecciones de entrada.

Si dividimos el tiempo de cómputo del método de las máscaras de iluminación entre el número de máscaras utilizadas + 1 (siendo este término adicional el correspondiente a las imágenes originales), vemos que se obtiene un tiempo de 43 ms aproximadamente, el cual idéntico al tiempo

de cómputo obtenido al aplicar el programa sin tratamiento alguno, lo cual es de suponer, puesto que no se aplica ningún tratamiento intermedio a la imagen en vivo, sino que el tratamiento ha sido realizado en la base de datos.

6.2.4. Conclusiones parciales

En las secuencias con condiciones controladas de iluminación se ve cómo el método que no trata la iluminación y los métodos de tratamiento de la iluminación de las máscaras de iluminación para enriquecer la base de datos y el de la cara recortada son los que mejores resultados dan. De tener que elegir uno para este tipo de condiciones, se optaría el que no se aplica el tratamiento de la iluminación, puesto que es el que menos sobrecarga computacional va a introducir.

En cuanto a las secuencias tomadas en condiciones no controladas de iluminación, podemos concluir que el mejor método, con diferencia sobre los demás, para obtener porcentajes de confianza altos, es el de ecualización de histograma, y también el mejor método de medida de distancia ha sido el de la distancia de Mahalanobis, la cual aumenta en la mayoría de casos el margen del usuario detectado sobre el resto.

El sistema se supone que va a funcionar en entornos exteriores más que en interiores, donde va a tener la iluminación ambiente controlada, por lo tanto, la elección final para el tratamiento de la iluminación es la ecualización de histograma.

En un futuro se debería de ser capaz de cuándo tenemos condiciones uniformes de iluminación, y cuando condiciones medianamente aleatorias de iluminación y condiciones muy malas de iluminación en la cara, para poder conmutar de un algoritmo a otro.

Por último decir que la mejoría obtenida con el método de máscaras de iluminación no compensa respecto del tiempo de cómputo que supone, con lo cual queda descartado.

6.3. Resultados del detector de gafas de sol

Este subsistema se ha probado con 7 secuencias, de los cuales se pasan a comentar las gráficas de resultados de algunos de ellos, puesto que en algunos los resultados son repetitivos. Se han probado secuencias en los que no se llevan puestas gafas de sol y en las que sí se llevan puestas, además de llevar distintos tipos de gafas de sol.

6.3.1. Tablas

Una vez se ha visto gráficamente los resultados del detector de gafas, se pasa a mostrar la siguiente tabla (6.11) que contiene las estadísticas para estas secuencias de los resultados arrojados por este detector.

	<i>Vídeo1</i>	<i>Vídeo2</i>	<i>Vídeo3</i>	<i>Vídeo4</i>	<i>Vídeo5</i>	<i>Vídeo6</i>	<i>Vídeo7</i>
Lleva gafas	NO	SI	NO	SI	NO	SI	SI
<i>Imágenes totales</i>	289	97	183	99	198	32	52
<i>Imágenes evaluadas</i>	175	50	14	99	10	32	40
<i>Tasa de evaluación de imágenes</i>	60,5 %	51,5 %	7,6 %	100 %	5 %	100 %	76,9 %
<i>Bordes evaluados</i>	364	151	29	246	21	90	86
<i>Bordes que superan el umbral</i>	8	43	7	37	3	34	40
<i>Imágenes con detección de gafas</i>	6	35	7	36	3	25	40
<i>Tasa de falsos positivos (sin contar con tasa de evaluación)</i>	3 %	-	50 %	-	30 %	-	-
<i>Tasa de acierto (sin contar con tasa de evaluación)</i>	-	70 %	-	36 %	-	78 %	100 %
<i>Tasa de falsos positivos antes de filtrado</i>	2 %	-	4 %	-	2 %	-	-
<i>Tasa de acierto antes de filtrado</i>	-	36 %	-	36 %	-	78 %	77 %
<i>Tasa final de acierto</i>	100 %	100 %	100 %	100 %	100 %	100 %	100 %

Tabla 6.11: Estadísticas del detector de gafas

Para empezar a evaluar las estadísticas, contamos el número total de fotogramas de la secuencia y número de imágenes evaluadas en el mismo, para poder obtener un porcentaje del tiempo que va a tener que funcionar el detector.

A continuación comprobamos cuántos son los bordes evaluados (no solo los detectados) en total, y de éstos, cuántos superan el umbral. Además, no todos los bordes que superan el umbral suponen que sean igual al número de imágenes en los que se detectan gafas, sino que en una misma imagen, varios bordes pueden estar por encima del umbral, y por tanto el número de imágenes siempre será menor que el número de bordes que superan el umbral.

A la vista de las últimas filas, se aprecia que en realidad se tiene una tasa de falsos positivos muy baja (sin tener en cuenta el filtrado posterior de envejecimiento pero teniendo en cuenta el porcentaje de imágenes en los que se ha evaluado la función fitness). Esta tasa de falsos positivos es invariante a las condiciones de iluminación, las cuales han variado en las distintas secuencias, como se ha indicado anteriormente.

Sin embargo, respecto de la tasa de acierto previa a la detección, se ve, como se preveía, que es fuertemente dependiente de la iluminación ambiente (los primeros vídeos tienen más reflejos y tienen un nivel de iluminación más alto que los últimos), lo cual sugiere que este bloque se debería introducir posteriormente al bloque de tratamiento de la iluminación, para intentar eliminar esta dependencia, si este tratamiento de la iluminación no genera bordes que puedan dificultar la labor de este detector.

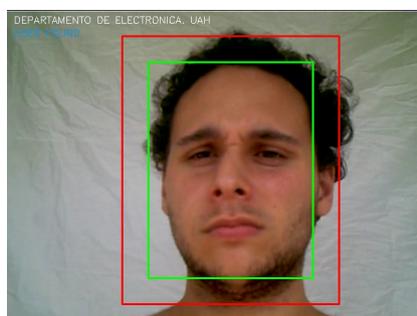
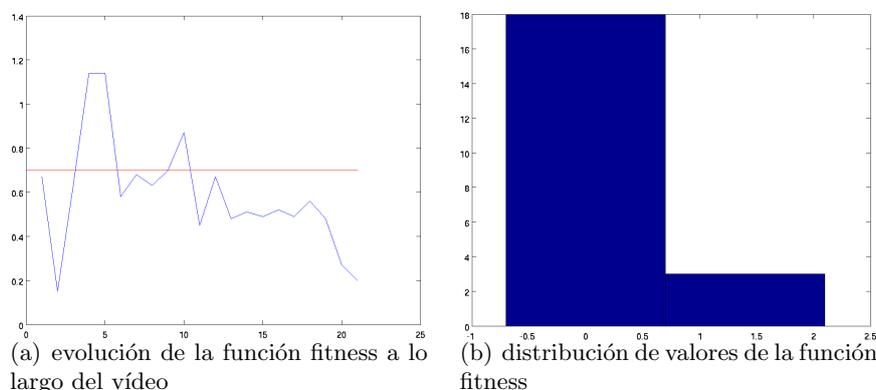
Ahora bien, estas tasas de acierto, con el algoritmo de pesos y envejecimiento aplicado en la máquina de estados principal del sistema, se ven filtradas y corregidas, consiguiendo un porcentaje de acierto en las secuencias de entrenamiento del 100 %.

6.3.2. Gráficas

Pasamos a presentar las gráficas de los resultados obtenidos de 3 de las secuencias de prueba grabadas en distintas condiciones. Estas secuencias se han tomado en condiciones de iluminación diurna, con ligeras variaciones de iluminación. Estos resultados quedan representados por 3 gráficas que darán una idea global del funcionamiento de este detector. Por un lado, en las subfiguras (a) de las siguientes figuras, se muestra la evolución temporal de la función fitness evaluada a lo largo de las distintas secuencias. El hecho de que se haga una representación temporal de la función fitness es para ver la frecuencia con la que se sobrepasa el umbral y poder ver cómo se comportaría el detector teniendo en cuenta que envejece los resultados a lo largo del tiempo, con lo cual es capaz de filtrar picos aislados. Por otra parte, en las subfiguras (b), se muestra un histograma que representa el número de evaluaciones de la función fitness a lo largo de toda la secuencia que sobrepasan el umbral y las que son inferiores al mismo, siendo establecido este umbral empíricamente a 0,7. En las subfiguras (c), se muestra una imagen de la secuencia a la que corresponden los resultados analizados.

En la figura 6.15, se muestra un ejemplo el programa funcionando en condiciones parecidas variando la iluminación y el fondo. Este cambio podría llegar a hacer que se generen sombras que pueden ser perjudiciales para el reconocimiento, puesto que el umbralizado de Otsu puede fijar como umbral óptimo, uno en el que incluya sombras en la cuenca de los ojos, lo que puede interpretarse como gafas de sol. Como se puede ver en las subfiguras 6.15.a y 6.15.b, el número de bordes evaluados es muy reducido, esto significa que en la mayoría de las imágenes de entrada se ha encontrado un único borde, correspondiente al contorno completo de la imagen, por lo que no se evalúa. Además, para los bordes evaluados, el número de veces que la función fitness que supera el umbral no es alto ni frecuente, y con el envejecimiento de las medidas que se realiza en

el programa principal, se eliminarán los efectos de estas evaluaciones positivas poco frecuentes y se rechazan estos falsos positivos, que si no se envejeciese, darían lugar a una detección incorrecta de gafas.

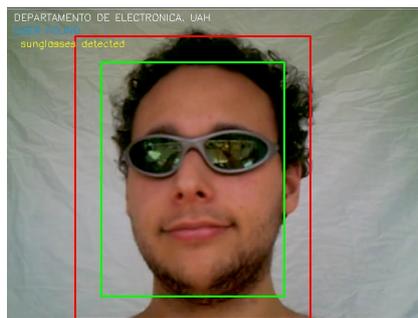
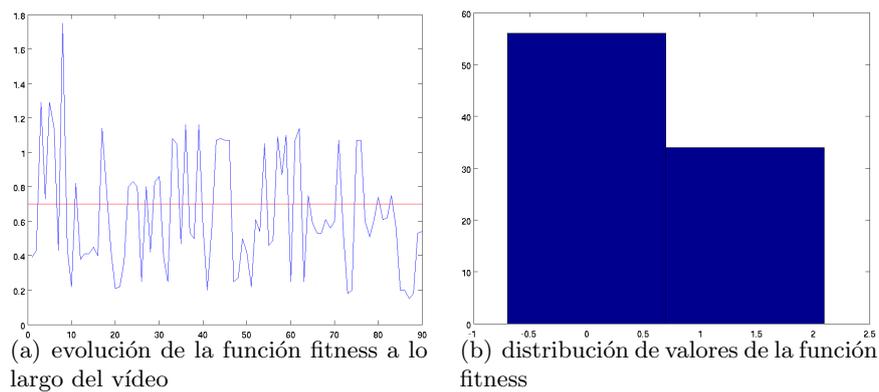


(c) muestra de entrada

Figura 6.15: Secuencia 5 de prueba del detector de gafas

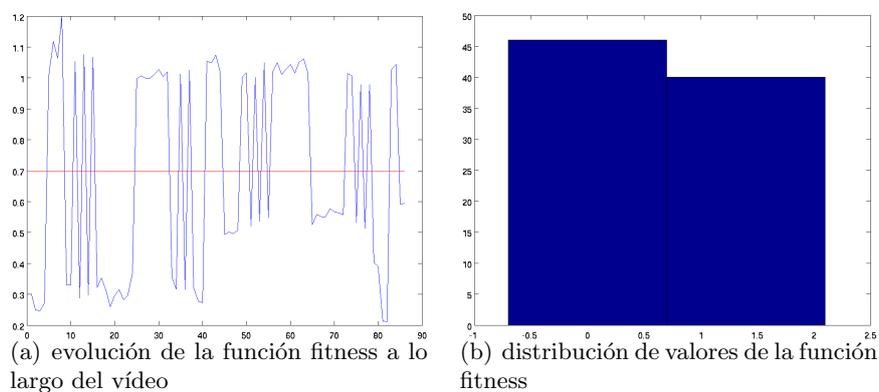
En cuanto a la figura 6.16, mostramos un ejemplo de una escena un poco menos iluminada que en el secuencia 2, también con gafas de sol puestas. En 6.16.a, se observa que la gráfica oscila con cierta frecuencia alrededor del umbral, e incluso se podría decir que la media de la función de evaluación está en torno al umbral fijado experimentalmente. En el histograma de la subfigura 6.16.b, se aprecia cómo prácticamente se iguala el número de veces en los que la evaluación de la función fitness está por encima y por debajo del umbral.

Como último ejemplo ponemos el de la figura 6.17, en la cual, hemos cambiado el tipo de gafa utilizada, para comprobar, como se pretendía, que el sistema fuese capaz de detectar distintos tipos de gafas y por tanto ser flexible en este aspecto. Vemos que las gráficas obtenidas son parecidas a las de la figura 6.16, aunque quizá en este caso, estas diferencias van a un mejor funcionamiento del detector, puesto que los resultados de la evaluación de la función fitness son mucho más claros. En cuanto al número de muestras evaluadas, vemos que no es muy grande comparado con la primera de las secuencias de prueba. Esto es debido a que, al detectar la presencia de gafas en la cara, se reinicia la máquina de estados completamente, con lo que vuelve a la fase de detección y tracking de la cara, lo cual lleva cierto tiempo y hace que disminuya el número de frames en los que se ha aplicado el detector. Estos máximos ofrecidos en el caso de la figura 6.17, son mucho más seguidos que en el resto de figuras.



(c) muestra de entrada

Figura 6.16: Secuencia 6 de prueba del detector de gafas



(c) muestra de entrada

Figura 6.17: Secuencia 7 de prueba del detector de gafas

6.3.3. Tiempos de cómputo

La sobrecarga que esta función de detección va a incluir en el sistema va a depender del número de bordes que se tengan que evaluar una vez pasado el primer filtrado de bordes. Si la cantidad de bordes que pasan es alta, esto es, existe una probabilidad alta de que se pueda detectar gafas, el algoritmo sobrecargará al sistema en mayor medida, mientras que si no, apenas se notará el efecto de la inclusión de este sistema en lo que a tiempo de cómputo se refiere.

Los tiempos de cómputo medios obtenidos son los siguientes:

Detección negativa	Detección positiva
0.95 ms	2.51 ms

Tabla 6.12: Sobrecarga introducida en el sistema debida a la inclusión del subsistema de detección de gafas

Vemos que la carga que introduce no es demasiado alta en ninguno de los 2 casos, y sin embargo, los beneficios que podemos obtener de él son considerables, puesto que las gafas de sol modifican bastante la apariencia del usuario y por tanto, podrían hacer que el sistema fallase.

6.3.4. Conclusiones parciales

Este sistema no introduce apenas sobrecarga en el sistema y sin embargo, el sistema es lo suficientemente flexible como para poder rechazar todo tipo de gafas en principio, siempre que estas generen bordes en la imagen umbralizada, lo cual hace de este subsistema un candidato perfecto para una implementación real.

6.4. Resultados de reconocimiento diurno y nocturno

Por último se presentan los resultados del sistema de detección diurno/nocturno y los reconocimientos realizados en las distintas condiciones.

A continuación se muestran algunas de las muestras tomadas para generar la base de datos. Éstas se han incluido en las figuras 6.18 y 6.19. Como podemos observar en las mismas, las características de las imágenes nocturnas son muy diferentes de las diurnas. Esto ha supuesto el tener que realizar un cambio sobre la estructura inicial de la base de datos. Se ha optado por dividir la base de datos en 2 bases independientes, nocturna y diurna, y, en función de las condiciones iniciales tomadas de las imágenes de entrada, cargar la correspondiente y, si estamos en el caso de iluminación diurna encender los iluminadores de infrarrojos.

Las características de las secuencias de entrenamiento y test son las siguientes:

- Cámara: Sony Handy-cam
- Resolución: 640x480
- Imágenes por segundo: 30
- Color: RGB

En cuanto a las características de la base de datos son las siguientes:

- Resolución: 80x110
- Número de imágenes por usuario:30
- Número de usuarios: 7
- Número de autovectores: 35

A la vista de las imágenes nocturnas, se intuye que el tratamiento de la iluminación en estos casos no va a ser necesario, puesto que las condiciones de iluminación ambientales, al ser prácticamente nulas, van a tener un efecto mínimo sobre la cara, y la iluminación predominante es la que se va a imponer con los iluminadores de infrarrojos, la cual está fija en posición. Podría afectar de manera considerable al reconocimiento las luces de otros coches que pasen cerca o la iluminación proveniente de las farolas, con lo cual habrá que investigar posteriormente este tipo de condiciones.

6.4.1. Tabla de resultados de reconocimiento diurno y nocturno

Pasamos a comentar las tablas obtenidas resultado de los reconocimientos realizados con el reconocimiento diurno y nocturno.

Estas secuencias no han sido tomados con la cámara de vídeo y el conjunto de iluminadores del proyecto, sino que, se han capturado las imágenes mediante una cámara que ya lleva incorporado un sistema de visión nocturna mediante unos diodos iluminadores de infrarrojos, por tanto, para el caso del decisor borroso, al no tener posibilidad de acceder a los datos de exposición en cada frame, se ha tenido que fijar manualmente éste al máximo posible de esta variable de entrada.

Por tanto, se podría haber reducido en este caso a un simple decisor por umbral a partir de la media de nivel de gris de la imagen, con un valor de umbral de 55.

A continuación se pasan a mostrar tablas que reflejan el porcentaje de reconocimientos válidos sobre las secuencias de prueba, con los que se han generado la base de datos y con las de prueba, donde el usuario entra y sale del coche, y durante ese tiempo realiza el reconocimiento. Como en anteriores ocasiones, se ha deshabilitado el clasificador, para poder obtener todos los resultados posibles, por muy malos que éstos sean. Las condiciones de iluminación son mejores que en otras secuencias, puesto que el coche se encontraba a la sombra con una iluminación uniforme, y no pasó mucho tiempo desde que se grabaron los vídeos de generación de la base de datos a que se grabasen los de prueba, lo cual mejora los porcentajes de reconocimiento.

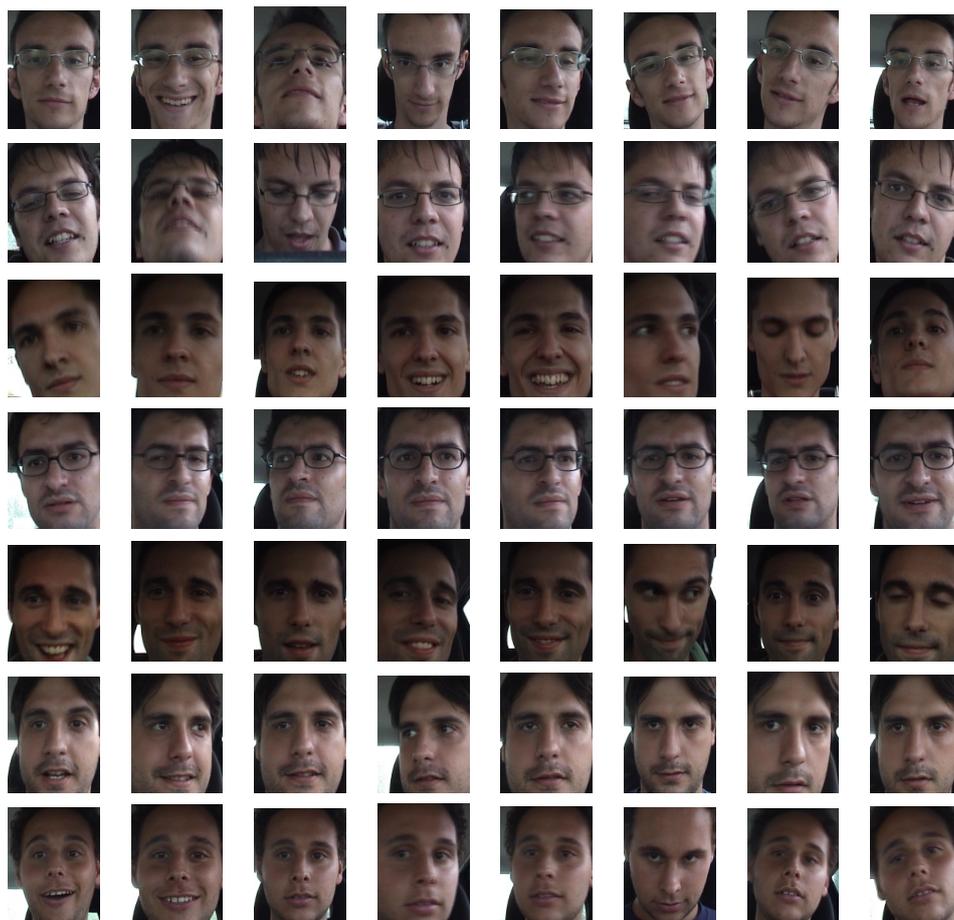


Figura 6.18: Muestras del conjunto de entrenamiento de la base de datos utilizada en condiciones de iluminación interior, sin tratamiento de la iluminación

En los resultados obtenidos con la distancia de Mahalanobis, mostrados en la figura 6.13, se ve cómo en la mayoría de los reconocimientos, los porcentajes de reconocimiento son del 100%. Experimentos que dieron resultados nefastos con distancia euclídea se han visto corregidos con la distancia de Mahalanobis hasta llegar a un 100% de seguridad acerca del usuario que se encuentra en frente de la cámara.

Podemos observar en los dos casos cómo el porcentaje de acierto en las imágenes nocturnas supera el de las imágenes diurnas, lo cual no nos sorprende, puesto que los grados de libertad que

en condiciones diurnas tiene la iluminación para cambiar la apariencia del rostro, en nocturno la iluminación es fija y lo único que se varía son los gestos de la cara, los cuales se han introducido en la base previamente.

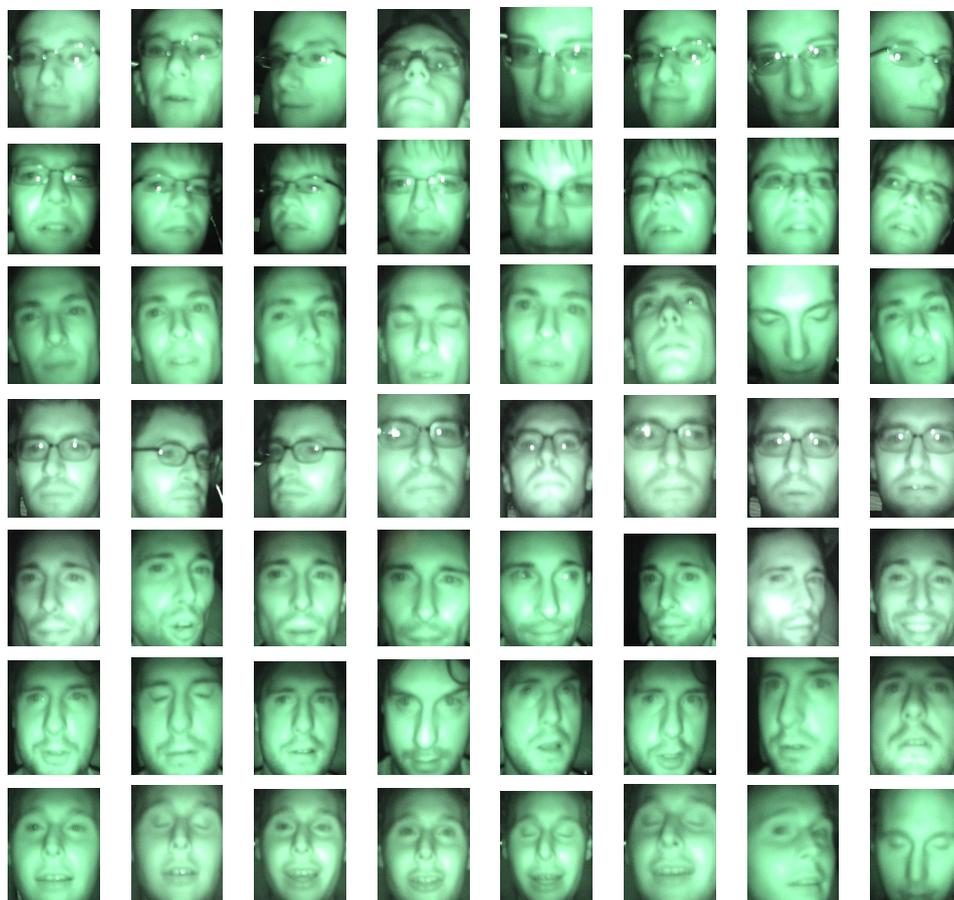


Figura 6.19: Muestras del conjunto de entrenamiento de la base de datos utilizada en condiciones de iluminación interior, sin tratamiento de la iluminación

Se ve en la tabla 6.13 que la tasa de acierto es cercana al 100%, puesto que los vídeos de generación de base de datos y los de test son muy parecidos. En condiciones reales, con pruebas realizadas sustituyendo las fotos de alguno de los usuarios de la base de datos por fotos de bases anteriores, se ve cómo este porcentaje baja considerablemente, hasta tasas similares a los resultados obtenidos en las pruebas de tratamiento de la iluminación. En algunos casos, los usuarios han sido rechazados al ser las características de las imágenes de entrada y las de la base de datos muy diferentes, y en caso de desactivar el clasificador, el candidato estaba en las primeras opciones del sistema, no siendo siempre la primera, aunque con el tratamiento de ecualización de histograma, los resultados medios de acierto superan el 70%. Las pruebas realizadas en condiciones diurnas, en cambio, no se ha dispuesto de un conjunto de secuencias más amplio para poder comprobar estadísticamente mejor los resultados, con lo que es de esperar, que esta tasa de acierto caiga alrededor del 80%.

	<i>Diurno</i>	<i>Nocturno</i>
<i>Usuario 1</i>	100 %	100 %
<i>Usuario 2</i>	100 %	95 %
<i>Usuario 3</i>	100 %	100 %
<i>Usuario 4</i>	90 %	75 %
<i>Usuario 5</i>	100 %	100 %
<i>Usuario 6</i>	100 %	100 %
<i>Usuario 7</i>	100 %	100 %
<i>Reconocimiento 1</i>	100 %	100 %
<i>Reconocimiento 2</i>	100 %	100 %
<i>Reconocimiento 3</i>	100 %	95 %
<i>Reconocimiento 4</i>	100 %	100 %
<i>Reconocimiento 5</i>	100 %	-
<i>Reconocimiento 6</i>	90 %	-
<i>Reconocimiento 7</i>	100 %	-
<i>Tasa media de reconocimiento</i>	98,5 %	96,8 %

Tabla 6.13: Resultados del reconocimiento para imágenes diurnas y nocturnas con distancia euclídea

6.4.2. Gráficas

Después de ver los resultados obtenidos a nivel general con las tablas de resultados, se muestra uno de los resultados más representativos. También se mostrarán resultados con imágenes prácticas acerca de los distintos valores de nivel medio de gris, para ver la diferencia que existe entre imágenes nocturnas y diurnas.

En primer lugar se muestra en la gráfica 6.20 los niveles medios de gris para las imágenes nocturnas y diurnas.

Los valores correspondientes a las barras situadas más a la derecha son las de las imágenes diurnas, con un alto nivel medio de gris, y en cambio, las barras situadas más a la izquierda se corresponden con los niveles medios obtenidos en las imágenes nocturnas.

Con las imágenes reales, se ve claramente el espacio existente entre las 2 clases (nivel de gris medio en imágenes diurnas y nocturnas), por lo que en principio, el sistema borroso de decisión se puede sustituir por un umbral en medio de las 2 clases. El sistema de decisión fue desarrollado antes de poseer esta información y se pretendía que fuese flexible (fácil de adaptar) y robusto, al tener en cuenta varias medidas.

Ahora pasamos a mostrar los resultados, de las pruebas con imágenes diurnas sin tratamiento de la iluminación, y posteriormente, las nocturnas. Basándonos en la tabla 6.13, se observa que existen dos patrones que siguen los resultados obtenidos, por un lado, están los reconocimientos que en distancia euclídea consiguen un porcentaje alto de resultados y en la distancia de Mahalanobis, se mantiene aproximadamente igual o mejora. Por otro lado existen los reconocimientos pésimos en distancia euclídea, que se transforman en reconocimientos perfectos cuando cambiamos a distancia de Mahalanobis. Este último caso se da en los reconocimientos 1 y 6. En primer lugar se muestra el último caso, al ser más raro y a continuación el caso normal.

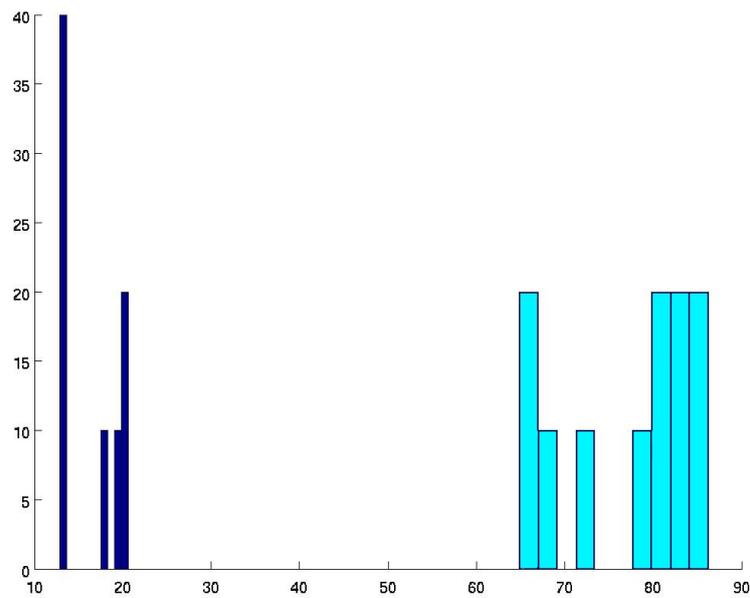


Figura 6.20: medias de nivel de gris en las imágenes iniciales, tanto diurnas como nocturnas



Figura 6.21: Muestras del primer y segundo reconocimientos en condiciones diurnas

Vemos en las 2 figuras anteriores (6.22 y 6.23) que el detector de media y el de vecino más cercano con criterio de mayoría coinciden en el resultado arrojado, por tanto, no se contradecirían y el reconocimiento es válido. Además vemos en la gráfica de la figura 6.22 cómo el margen que hay sobre el resto de usuarios es más que suficiente para que no se descarte ninguna muestra y para tener una seguridad del 100% en el reconocimiento.

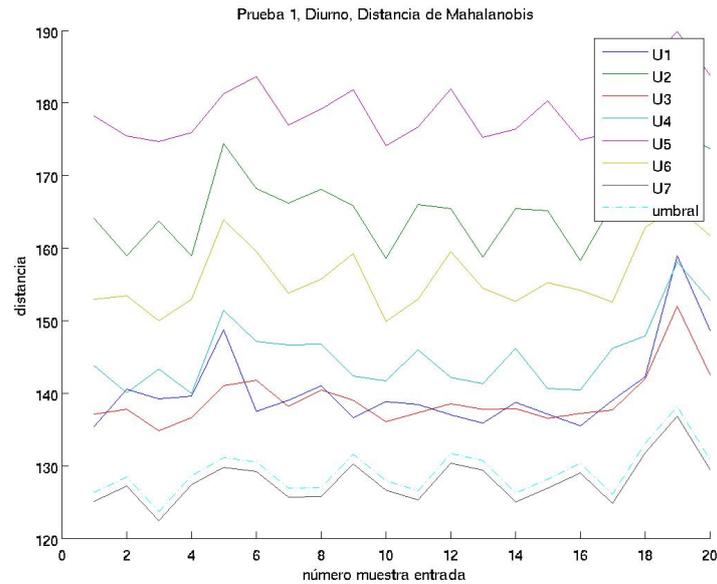


Figura 6.22: Experimento primero de reconocimiento en condiciones diurnas

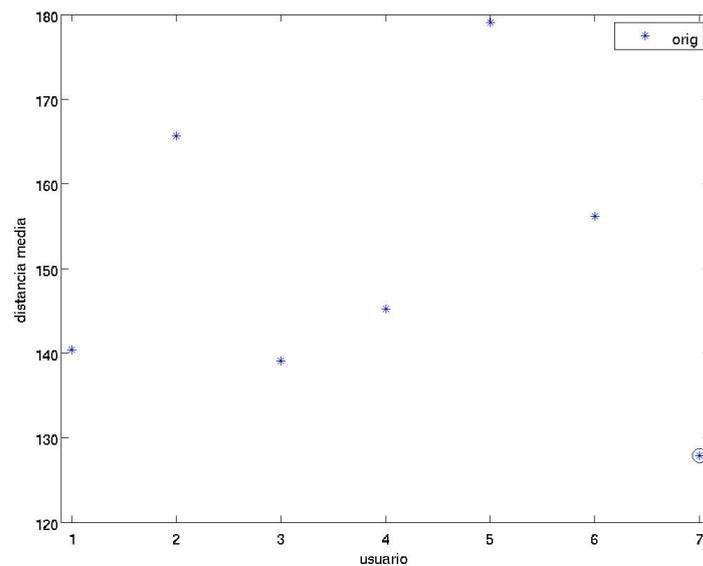


Figura 6.23: Distancias medias obtenidas en el experimento primero de reconocimiento en condiciones diurnas

Una vez hemos visto los ejemplos en reconocimiento diurno, pasamos a mostrar algunas gráficas del conocimiento nocturno.

A la vista de las tablas, en todos los casos los reconocimientos realizados superan el 85%, por tanto, uno de los casos cualquiera será suficiente para ilustrar el comportamiento del sistema ante este tipo de imágenes.

Comenzamos mostrando ejemplos de muestras tomadas para el reconocimiento, no las que

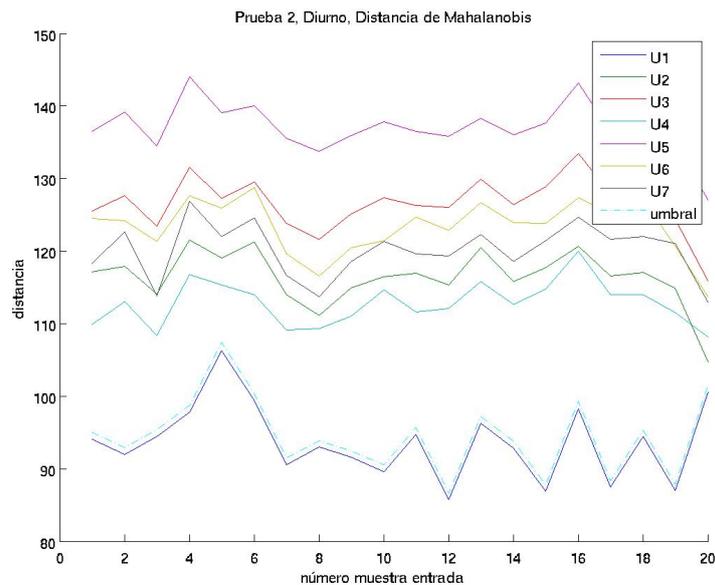


Figura 6.24: Experimento segundo de reconocimiento en condiciones diurna

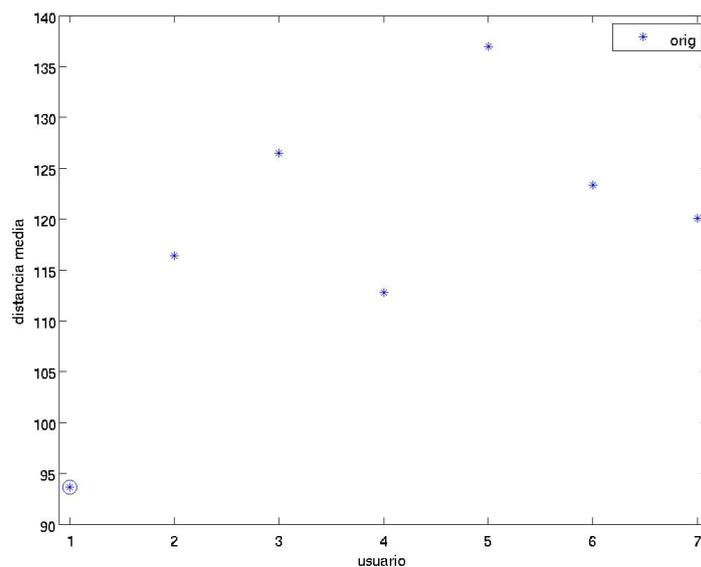


Figura 6.25: Distancias medias obtenidas en el experimento sexto de reconocimiento en condiciones diurnas

están en la base de datos, por muy similares que puedan parecer.

Como se puede ver en la última muestra de la figura 6.26, el trozo de cara detectada no se corresponde realmente con la cara, sino que solo se centra en una parte. Este problema, como en veces anteriores es debido al sistema de tracking y al de detección, no estrictamente es problema del algoritmo o de la iluminación, que son los que se intentan paliar. De todos modos, este



Figura 6.26: Muestras del tercer reconocimiento en condiciones nocturnas

problema de recorte se ha solucionado incluyendo muestras de esta situación particular en la base de datos. Las muestras incluidas no han sido las utilizadas en el reconocimiento, sino otras similares en otros instantes de tiempo, que no coinciden exactamente en apariencia, cosa que si hubiese sido así, se habría alcanzado un valor de distancia cercano a 0 y este no es el caso.

En la gráfica 6.27.a vemos cómo sólo los usuarios que han tenido un problema similar con el recorte de caras, son los candidatos más probables con poco margen entre sí, al ser una zona reducida de la cara, estando separados el resto de manera considerable. Aún estando con muy poco margen en la distancia euclídea, podemos ver en la gráfica 6.28, correspondiente a la media, cómo el resultado del vecino más cercano con criterio de mayoría es confirmado.

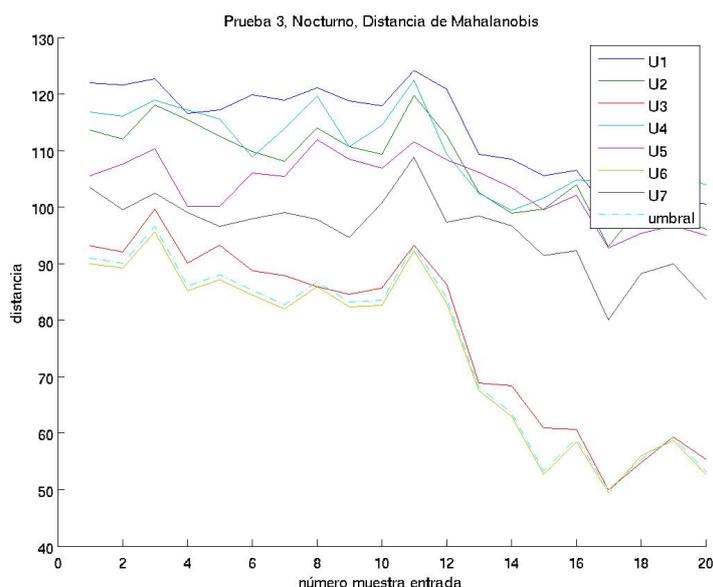


Figura 6.27: Experimento tercero de reconocimiento en condiciones nocturnas

6.4.3. Tiempo de cómputo

En este caso, el sistema funcionará aproximadamente con los tiempos comentados en la sección de independización de la iluminación, si la resolución hubiese sido la misma (subsección

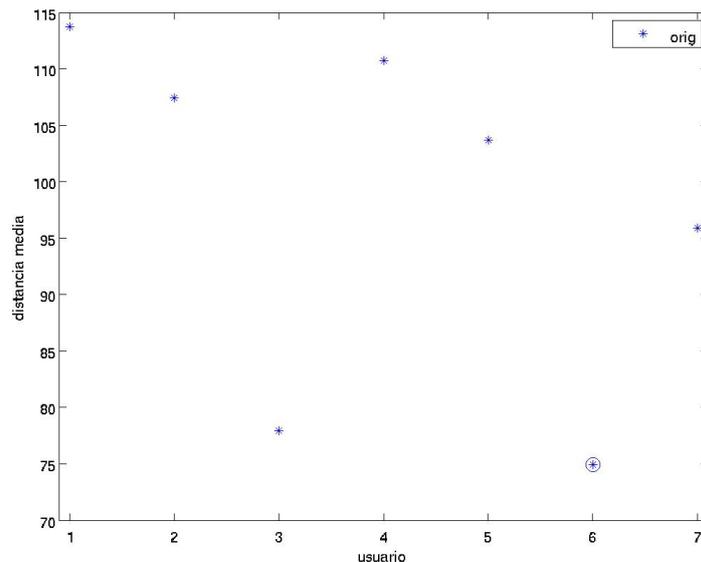


Figura 6.28: Distancias medias obtenidas en el experimento tercero de reconocimiento en condiciones nocturnas

6.2.3). De todos modos, se muestran a continuación tiempos de cómputo medios obtenidos utilizando la distancia euclídea y la de Mahalanobis.

	Diurno	Nocturno
Euclídea	35 ms/frame +22 ms clasif	37 ms/frame + 25ms clasif
Mahalanobis	855 ms	842 ms

Tabla 6.14: Tiempos de ejecución del reconocimiento para imágenes nocturnas y diurnas sin tratamiento de iluminación

Se observa que los tiempos de cómputo son algo mayores. Esto es debido a que se ha incrementado ligeramente la resolución de las muestras utilizadas para el reconocimiento en un intento por mejorar el margen de confianza.

En total, a lo largo de los primeros 10 frames, el tiempo de cómputo obtenido en el subsistema borroso que decide si se está en condiciones diurnas y nocturnas está entre 178 y 189 ms en el caso de una detección diurna y entre 176 y 179 ms en el caso de una detección nocturna.

Hay que tener en cuenta que este tiempo es el que lleva incluido el tiempo de adquisición de las imágenes y del cálculo del nivel de gris, necesario tanto para un sistema de umbralizado como para el sistema borroso propuesto.

La carga inicial en este caso no afecta al comportamiento del sistema final en cuanto a comportamiento en el tiempo se refiere a la hora de realizar el reconocimiento, puesto que solo se realiza al principio, siendo el resto del sistema independiente de este detector inicial.

6.4.4. Conclusiones parciales

Vemos otra vez cómo la distancia de Mahalanobis, en general, mejora los resultados de reconocimiento, sobre todo a la vista de las pruebas de reconocimiento diurnas 1 y 6, en las que pasamos de equivocarnos completamente a reconocer perfectamente al usuario correcto. También hay que tener en cuenta de que en este caso no se ha aplicado tratamiento de iluminación alguno. En las imágenes nocturnas la distancia de Mahalanobis también es beneficiosa, puesto que amplía los márgenes cuando se necesitan.

El subsistema que detecta si una imagen es nocturna o diurna para cargar la base de datos adecuada, actualmente es sustituible por un sencillo umbral a partir del nivel medio de iluminación de la escena. Se creó este sistema al no tener información a priori de cómo iban a ser las imágenes diurnas y nocturnas.

Como en resultados anteriores, fallos en el detector de caras y en el sistema de tracking son críticos a la hora de realizar el reconocimiento, con lo que habrá que seguir en esta línea antes de implementar este prototipo en un sistema comercial.

Las imágenes nocturnas no parece que vaya a necesitar de un tratamiento de la iluminación, o si se ha de utilizar, se utilizará algo parecido a lo visto anteriormente, para condiciones en las que la iluminación se encuentra controlada. En los casos expuestos se ha visto cómo la iluminación es, para todos los casos la misma. El foco utilizado en la cámara es lo suficientemente potente para poder despreciar buena parte del efecto del resto de las luces en la cara.

Las características de las imágenes nocturnas varían mucho si se toman de un modelo de cámara u otro. Los parámetros de sensibilidad son críticos en estos casos, así como los filtros utilizados, con lo que una variación en la implementación, haría que se tuviese que grabar siempre con la misma cámara, tanto para grabar los vídeos de generación de la base de datos como la cámara que irá a bordo del vehículo.

Las características de las secuencias diurnas son mucho más variables, en lo que a iluminación se refiere que a las nocturnas.

Capítulo 7

Conclusiones y Trabajos futuros

7.1. Conclusiones finales

Las imágenes tomadas para el reconocimiento, al ser seguidas si el sistema de tracking o el de detección funciona adecuadamente, son muy parecidas. Dependiendo de la tasa de fotogramas por segundo, se empleará más o menos tiempo para el reconocimiento. Si la imagen es estable, y la tasa de fotogramas de fotogramas es de 7.5, en menos de 3 segundos se ha realizado el reconocimiento, si no estamos aplicando un tratamiento de la iluminación basado en el método de las máscaras de iluminación en el cual tardará bastante más al ser el número de imágenes en la base de datos muy superior.

A la vista de los resultados, se deduce que la distancia de Mahalanobis es mejor que la distancia euclídea para los propósitos finales, sobre todo para el rechazo de usuarios externos a la base de datos, aunque también para corrección y mejora del margen de confianza del reconocimiento (siendo definido este como el número de imágenes correctamente reconocidas sobre el total de muestras tomadas).

Se ha visto que el algoritmo Retinex es el que peor funciona de todos si se quiere realizar un reconocimiento de manera independiente.

Para el tratamiento de la iluminación en condiciones que tienden a lo uniforme, con ligeras variaciones a lo largo del día, se prefiere incluso no tratar la iluminación, puesto que se han conseguido resultados parecidos a tratándola a través del método presentado de las máscaras de iluminación, a diferencia de que éste es más costoso computacionalmente. Respecto del otro método de reconocimiento para condiciones en las que la iluminación varíe poco, el de la cara recortada, vemos que el tiempo que se puede emplear sea menor, pero tampoco mucho más, y el porcentaje de reconocimiento no varía demasiado, con lo que quedamos con el sistema original, y así además no tenemos que hacer grandes modificaciones iniciales a la base de datos.

Para casos en los que la iluminación sufra modificaciones importantes, como saturación de la cara, iluminación parcial con sobre-iluminación en una parte de la cara y falta de la misma en la otra, el método que mejores resultados ha dado ha sido el de la ecualización de histograma.

A la vista de las conclusiones anteriores, se podría pensar en que la implementación final del sistema podría ser, basada completamente en la distancia de Mahalanobis y con 2 bases de datos diferenciadas, una la original tratada con la ecualización de histograma y la segunda base, la de condiciones de iluminación nocturna. Con estas 2 bases, se cubre un amplio espectro de posibilidades en lo que a iluminación se refiere.

Por supuesto, antes de inicializar el tracking, se incluirá el detector de gafas como se ha indicado anteriormente, para evitar fallos en el reconocimiento. Este sistema, puede dar lugar a falsos positivos, pero esta circunstancia es preferible frente a dejar pasar a imágenes en las que se encuentren gafas de sol.

Para asegurar un correcto funcionamiento del sistema, se ha de trabajar en la mejora del tracking y del detector de Viola & Jones modificado implementado en OpenCV, puesto que, en caso de que se introduzcan imágenes correspondientes a falsos positivos, pueden arruinar un reconocimiento que podría pasar como válido cuando en realidad no lo es.

Por último, pasamos a describir ciertos trabajos futuros que serán necesarios si se quiere llevar a cabo de manera comercial, puesto que le quedan algunos ajustes que realizar, que pasamos a describir a continuación en distintas secciones

7.2. Trabajos futuros en iluminación

En el sistema actual no se ha conseguido una mejora significativa frente a condiciones de iluminación adversas, por lo que habría que trabajar en una corrección por partes de la imagen o una corrección del nivel gamma por zonas, o probar otra serie de métodos no probados, que se encuentran en la bibliografía.

Se debería además, adelantar unas etapas el procesamiento de la iluminación, para beneficiar al detector de Viola & Jones, puesto que en unas condiciones adversas de iluminación no funciona correctamente, mientras que en condiciones de iluminación uniformes, o por lo menos no tan bruscas, el detector funciona correctamente.

En cuanto a iluminación nocturna, como se ha dicho en el apartado de resultados, se puede dar el caso de que un foco esté iluminando parte de la cara, alterando el patrón normal de iluminación en caso de no tener otras fuentes de iluminación cercanas. Esto podría ser un problema si este patrón llegase a alterar suficientemente la apariencia de la cara y por tanto, ya sería necesario aplicar tratamientos de iluminación.

También existe un umbral no definido entre condiciones nocturnas y diurnas en el que el sistema puede interpretar una imagen como nocturna aún teniendo suficiente iluminación como para ser diurna, por tanto habría que analizar este caso en especial, puesto que no se ha dispuesto de secuencias para analizar este caso concreto. Estas condiciones se pueden dar al amanecer o al anochecer, durante cierto tiempo no despreciable, cuando el sol está en el horizonte.

Además, a la vista de los resultados, se debería ser capaz de detectar el tipo de condiciones de iluminación de la iluminación en las que nos encontramos, pudiendo seleccionar un tratamiento de iluminación u otro. Si se consigue esto, se debería ampliar la base de datos, en un modo parecido al que se ha propuesto al dividir la base en nocturna y diurna, para poder tener precalculadas las proyecciones y las bases de los espacios con las imágenes con los distintos tratamientos ya aplicados. Yendo más allá incluso, si las condiciones variasen durante el reconocimiento, se podría incluso seleccionar el método adecuado a cada tipo de imagen según sus características, y fusionar los resultados del reconocimiento posteriormente, por lo que habría que pensar también en una normalización de las distancias, para poder integrar los resultados de los distintos métodos adecuadamente.

7.3. Trabajos futuros en tracking

Respecto del tracking no funciona bien con todas las relaciones de aspecto posibles en una cara (no todas las caras tienen la misma relación de aspecto, puesto que se puede dar el caso de tener una cara alargada o una cara más redondeada, lo que hace que la relación de aspecto detectada tienda a ser $\gg 1$ o $\simeq 1$ respectivamente), e incluso el ajuste varía dependiendo de la distancia con la cámara. Por esto se sugiere una normalización de la relación de aspecto, o incluso, la utilización de algún algoritmo de localización de puntos característicos. Existen implementaciones mejoradas para tracking de caras en desarrollo por el grupo de investigación en el que se ha desarrollado este proyecto.

Para evitar pérdidas una vez localizada una cara y para seguirla con una mejor estabilidad, además de para mejorar el ajuste de la cara, se debería implementar un filtro de Kalman (normal o extendido, dependiendo de la planta que tengamos, si el ruido que se vaya a introducir es gaussiano o no).

Se ve el detector de Viola & Jones, en las pruebas experimentales, es dependiente de la iluminación de la cara. Se sugiere un tratamiento previo para eliminar este problema.

7.4. Trabajos futuros en inmunización frente a intentos de engaño del sistema

Un problema fundamental que tenemos en este sistema, es el ataque de gemelo. Una persona muy parecida a una que tengamos en la base de datos, puede pasar como la que tenemos en esa base, y por tanto, ahí se tendrá un error de autenticación. Este tipo de errores se tendrán que suplir con sistemas híbridos, que utilicen como entradas de autenticación no solo la apariencia, sino también alguna otra huella biométrica o algún código o tarjeta.

Además del problema comentado anteriormente, la implementación en este sistema de reconocimiento de un sistema híbrido nos dará otras ventajas para aumentar el porcentaje de éxito en el reconocimiento.

Intentar aplicar técnicas LDA en el sistema para mejorar el porcentaje de acierto y rechazo del sistema (debido a las propiedades de separación de clases que lleva implementadas el algoritmo), ya sea por sustitución de 2DPCA por LDA o por concatenación de ambos métodos, haciendo primero la proyección en 2DPCA y posteriormente en LDA para crear la separación de clases necesaria.

Otra debilidad detectada en el sistema es que, en el subsistema que decide si la imagen es nocturna o diurna, si se apunta con un foco directamente a la cámara, puede hacer que decida que el entorno es diurno y por tanto, la base de datos que se cargue sea la diurna, y, cuando se encuentre un candidato en el asiento del coche, que el reconocimiento no sea correcto.

7.5. Trabajos futuros respecto de la apariencia de las imágenes de entrada

Además, el sistema aún no es capaz de reconocer a usuarios si las características de su cara cambian sustancialmente, esto es, el usuario se deja barba, perilla o bigote. Respecto al corte de pelo, aunque se intenta filtrar con en el recorte de la cara que excluye el pelo, si éste llega a

tapar la cara, por ejemplo con un flequillo, el sistema no sería capaz de reconocer correctamente al usuario.

Un problema al que puede estar expuesto el sistema es que, al estar basado en apariencia, si se enfrenta al sistema una foto de un usuario autorizado, se podría acceder al sistema. En realidad lo que tenemos que hacer es comprobar si lo que hay enfrente de la cámara es en realidad una foto y la persona real. La primera idea posible para comprobar que sea una fotografía o no, es comprobar si lo que estamos viendo tiene tercera dimensión o no, si no tiene tercera dimensión esa zona, podríamos rechazar automáticamente al usuario con la certeza de se ha puesto una foto en frente de la cámara. Otra solución posible y más barata, aunque puede no funcionar siempre, es aprovechar el hecho de que una fotografía va a generar una discontinuidad con el entorno natural o de fondo. Si se utiliza esta circunstancia, se pueden detectar bordes en la imagen y por tanto con esos bordes y con una transformada de Hough o por mínimos cuadrados, se puede encontrar bordes en la imagen de fondo que correspondan a un rectángulo, relajando la restricción a que la cumplan 3 de los 4 bordes, pues en uno de los 4 bordes posibles el sujeto que intente hacer este tipo de ataque la tiene que sujetar con uno de los dedos o de otra forma.

Se podría realizar la implementación de un sistema de selección de muestras adaptativo y también una aproximación a IPCA. Este tipo de sistemas adaptativos, harán que, con el paso del tiempo, se seleccionen las muestras más utilizadas, aumentando la tasa de reconocimiento.

Para mejorar el reconocimiento, antes de ajustar la imagen de características, se podría eliminar el fondo de la imagen utilizando una máscara elíptica predeterminada, como se ha hecho en otros papers de la bibliografía.

Otra idea basada en IPCA pero variando un poco este método es, al principio sobre todo, ejecutar el programa en modo supervisado, y realizar algunos reconocimientos, sean fallidos estos o acertados. Estos reconocimientos se deberían realizar en distintas condiciones de iluminación, a distintas horas del día, para que el aspecto de las imágenes sea distinto, aunque el usuario sea el mismo. A partir de éstos, extraer estadísticas de cuáles son los más utilizados, y los que no lo sean, se sustituirán por algunas de las imágenes de entrada de los reconocimientos fallidos y se recalculará la base del espacio.

7.6. Trabajos futuros en el clasificador

Implementar un algoritmo de clasificación para evitar confusiones entre usuarios aceptados por el sistema una vez pasadas las 2 primeras etapas de filtrado, mediante un algoritmo más complejo de clasificación, K-medias borrosas, SVM, redes neuronales o similar. La necesidad de probar con este tipo de algoritmos viene dada puesto que PCA/2DPCA no genera una separación muy clara entre clases, sino más bien que en algunos casos, estas clases se entremezclan en el espacio, por lo que se puede tener problemas.

7.7. Trabajos futuros en la base de datos

En la selección de muestras se ha implementado un muestreo equi-espaciado en el tiempo y también se ha hecho mediante selección manual, seleccionando las muestras que al operador le han parecido más representativas (este último método aumenta la variedad de las muestras, pero puede que algunas de ellas no se ajusten al perfil normal que se va a incluir). Para la selección automática de muestras que no sea a través del muestreo periódico, se pretende implementar un sistema adaptativo en el cual se vaya generando un núcleo PCA (o 2DPCA) con la secuencia de

vídeo de entrada, y si la imagen siguiente se diferencia lo suficiente de la anterior, esta imagen se incluirá en el núcleo PCA.

Parte III

Pliego de condiciones

Capítulo 8

Pliego de condiciones

8.1. Condiciones generales

A continuación pasamos a enumerar cuales serán los materiales y programas básicos necesarios para la implantación de este sistema en un coche, lo que al prototipo se refiere, no al sistema final, el cual está todavía por determinar.

8.1.1. Equipos

En esta sección se indicarán los elementos físicos (hardware) que se utilizarán para implantar este prototipo.

8.1.1.1. Cámara

La cámara que se utilizará para la adquisición de imágenes del sistema es una Basler Scout, aunque para las primeras pruebas se utilizó una Unibrain Fire-i.



Figura 8.1: Cámaras utilizadas

Las 2 primeras cámaras son profesionales y con una velocidad de transmisión elevada, lo que las hace ideales para transmitir a una alta tasa de imágenes por segundo con imágenes de gran resolución, lo cual es interesante, junto con un equipo potente, para realizar el reconocimiento lo más rápido posible, y también la resolución que sea lo suficientemente grande para poder realizar el reconocimiento con la mayor seguridad posible.

Respecto de la óptica de la cámara, se utilizará una óptica que no contenga un filtro de infrarrojos que disminuya mucho la sensibilidad de la cámara en esta banda, para poder realizar un correcto reconocimiento nocturno. Se preferirá para ajustar el obturador una óptica que tenga esta propiedad.

8.1.1.2. Tarjeta de adquisición

Se prefiere el uso de este tipo de tarjetas si se embarca un PC de sobremesa con un SAI y un transformador de corriente continua a alterna para su alimentación.

La velocidad de las tarjetas de adquisición Firewire serán las mismas que las máximas de la cámara utilizada, 400 u 800, con buffer suficiente para soportar la captura correctamente. El programa de captura desarrollado por el departamento utiliza, además de un buffer incluido en la memoria RAM, los buffers internos de la tarjeta para poder realizar una captura sin perder una sola imagen.



Figura 8.2: Tarjetas adaptadoras para bus Firewire a distintos buses de PC o Mac, de la firma Unibrain

8.1.1.3. Cable de conexión

Bus Firewire (IEEE 1394) 400 u 800, cable tipo a o b, pero preferiblemente tipo b, o lo que es lo mismo, con conectores de 9 a 6 pines, puesto que se requiere no tener que introducir alimentación externa y las tarjetas de adquisición tienen conectores de 6 pines como se verá más adelante.

Según la cámara y la tarjeta de adquisición que utilicemos se utilizará cable específico de 400 o 800Mbps.



Figura 8.3: Cable Firewire

8.1.1.4. Iluminador de infrarrojos

El iluminador de infrarrojos está ya diseñado por el departamento de electrónica para el proyecto Cabintec. Está compuesto de por 10 diodos de infrarrojos modelo OPE5685 divididos en 2 ramas, para que los 10 diodos se puedan alimentar con tensiones cercanas a los 8 - 12 V y con corrientes suficientes para hacerlos conducir e iluminar correctamente la cara a la que está enfocando la cámara.

Se necesitará un iluminador al utilizar solo una cámara, y además, puesto que el sistema completo ha sido desarrollado, también se necesitará la tarjeta de sincronización, la cual se utilizará para controlar mediante el puerto serie la cantidad de iluminación que se proyecta al sujeto.

8.1.1.5. PC

Para poder ejecutar el programa, se ha de utilizar un PC o Mac. Si se opta por un Mac, éste es portátil y tiene Firewire se podrá utilizar directamente los cables de 9 pines a 6 pines. En el caso de que el portátil utilizado sea basado en PC y tiene una entrada mini-Firewire (4 pines), se habrá de utilizar un adaptador externo, y además utilizar una toma de alimentación externa, si la cámara dispone de ella, puesto que las entradas mini-Firewire no tienen capacidad de proporcionar alimentación a los dispositivos.

Como mínimo este ordenador ha de contar con un procesador a 2.66 GHz, pero quizá este sistema no sea suficiente si se quiere ejecutar en tiempo real, lo cual hace recomendar un micro-procesador de doble núcleo, con una velocidad aproximada por cada uno de unos 2 GHz, para repartirse la carga.

Para la ejecución del sistema operativo, el programa, la carga de base de datos y el trabajo con imágenes, se recomienda que el PC tenga 1 y 2GB de memoria RAM.

Para procesar los vídeos de entrenamiento, extraer fotografías, instalar el sistema operativo y otra serie de programas, se recomienda utilizar un disco duro de 80 GB mínimo.

Si el ordenador no contiene entradas de 6 pines, o de 4 pines + adaptador, se recomienda, para la tarjeta de adquisición que se ha de introducir, que la placa base tenga buses del tipo PCI (PCI, PCI express si existen tarjetas adecuadas para adaptar desde éste a Firewire, y PCMCIA para portátiles).

Se recomienda también que la placa base lleve un puerto serie para poder controlar la tarjeta de sincronización del iluminador de infrarrojos.

8.1.2. Software

Una vez se ha descrito por encima el hardware necesario para el control, se describirá el software necesario que se ha de instalar para que el sistema completo funcione.

8.1.2.1. S.O. Linux

Se ha seleccionado este sistema operativo para la ejecución del programa por varios motivos.

En primer lugar, porque consume menos recursos para el sistema operativo que windows, aprovechándolos más eficientemente, y teniendo muchos más recursos disponibles para la ejecución por parte del software desarrollado.

Otra de las razones por las que se ha seleccionado este sistema operativo es porque las librerías de visión computacional (OpenCV) están desarrolladas tanto para windows como para Linux y son de libre distribución, fácilmente descargables e instalables para su funcionamiento. Además, Linux, con sus repositorios de paquetes, proporciona una manera fácil para la actualización no solo de las librerías asociadas a OpenCV sino de las librerías para poder leer distintos formatos de imágenes(png, tiff, jpg, etc) y vídeos(mpeg y avi).

También se seleccionó este sistema porque los drivers y las librerías para poder acceder a los buses Firewire, si no se utilizan las funciones implementadas en las librerías de visión, son fácilmente accesibles mediante código fuente en lenguaje c.

Por último, la posible transición del programa a un dispositivo empotrado puede ser más sencilla desde un Linux normal a un Linux embebido, que desde windows a Linux embebido.

Dentro de Linux existen multitud de familias, siendo la empleada en nuestro caso, distribuciones basadas en debian como son ubuntu/kubuntu u otras familias como knoppix.

Se utilizarán las versiones de escritorio, desktop o de similar denominación en lugar de la versión de servidores, server o similar, puesto que la configuración del kernel es algo distinta al ser las necesidades de proceso de éstos sistemas distintas entre sí.

En cuanto a la versión del kernel, para tener soporte implantado en kernel del bus Firewire, y por otras razones, se aconseja el uso de versiones con un núcleo versión 2.6.xx, el cuál tienen la inmensa mayoría de distribuciones de Linux actuales.

8.1.2.2. Generación de base de datos de caras

Una vez se han instalado con éxito las librerías anteriormente mencionadas, se procederá a la compilación del software de generación de la base de datos para poder generarla antes de proceder a reconocimiento.

Este software generará los datos adecuados para el programa de reconocimiento a partir de las imágenes de muestra de los usuarios.

Para obtener las imágenes de muestra, se puede realizar de manera manual, o mediante muestreo periódico, arrancando el programa de reconocimiento con unas opciones particulares que se pasarán a comentar posteriormente.

8.1.2.3. Reconocimiento de usuarios y extracción de muestras

Con todo lo anterior instalado correctamente y la base de datos generada correctamente, se puede pasar a realizar el reconocimiento de usuarios o de extraer muestras de datos de entrada, mediante el programa desarrollado en este proyecto. Lo único que hay que hacer para que éste funcione correctamente es compilar los archivos para que generen los archivos ejecutables.

8.2. Condiciones de ejecución

La instalación del equipo en el coche se ha de realizar teniendo en cuenta las limitaciones existentes en el mismo de espacio o limitaciones eléctricas, si se opta por la ejecución del mismo en un PC y se necesita de un SAI y de un convertidor de corriente continua a alterna, puesto que la pantalla, el PC, las cámaras, los iluminadores, la tarjeta de sincronización y demás periféricos, suponen una gran carga para el SAI, y puede que alguno de los fusibles del coche se fundan. Por tanto, se recomienda ser especialmente cuidadosos a la hora de la instalación del hardware en el coche.

Respecto del software, el método de instalación del sistema operativo en el ordenador que se ejecute, depende de la distribución, aunque en líneas generales, basta con seleccionar las opciones predeterminadas. Hay que tener especial cuidado a la hora del particionado, seleccionando el espacio que sea oportuno del disco duro, preferiblemente, si no se va a instalar windows para otras cosas, se formateará toda la partición en ext3 excepto 2 GB para crear una partición de intercambio o swap.

La instalación del resto de paquetes y programas necesarios para la ejecución del programa se encuentra detallada en el apéndice A.

8.3. Especificaciones técnicas

Una vez se ha descrito de manera general qué se necesita para implantar el sistema, y después de haber dado unas nociones generales acerca de cómo hacerlo funcionar se pasan a especificar las características que han de tener los equipos necesarios para que todo funcione.

8.3.1. Cámara de vídeo

A continuación se pasa a enumerar cuales son las características de la cámara de vídeo utilizada, Bastler Scout o similar.

En primer lugar se pasa a describir las características del sensor utilizado en la cámara de vídeo del sistema final y después su tabla de características.

Se puede ver que tiene una alta respuesta en la zona del espectro visible, aunque la respuesta no es del todo despreciable a la longitud de onda de trabajo (0,3) en el modo de trabajo en

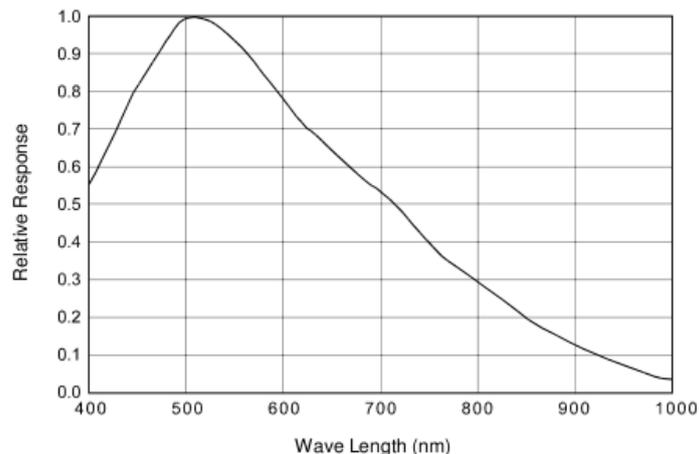


Figura 8.4: Características en monocromo de la cámara utilizada, Bastler Scout

Specification	scA640-70fm/fc	scA640-74fm/fc	scA780-54fm/fc	scA1000-30fm/fc	scA1400-17fm/fc
Sensor Size (H x V pixels)	fm: 659 x 494 fc: 658 x 492	fm: 659 x 494 fc: 658 x 492	fm: 782 x 582 fc: 780 x 580	fm: 1034 x 779 fc: 1032 x 778	fm: 1392 x 1040 fc: 1390 x 1038
Sensor Type	Sony ICX424 AL/AQ	Sony ICX414 AL/AQ	Sony ICX415 AL/AQ	Sony ICX204 AL/AK	Sony ICX285 AL/AQ
	Progressive scan CCD				
Optical Size	1/3"	1/2"	1/2"	1/3"	2/3"
Pixel Size (H x V μm)	7.4 x 7.4	9.9 x 9.9	8.3 x 8.3	4.65 x 4.65	6.45 x 6.45
Max. Frame Rate (at full resolution with 8 bit output)	71 fps	74 fps	54 fps	30 fps	17 fps
Mono/Color	All models available in mono or color				
Image Data Output Type	IEEE 1394b				
Image Data Output Formats	Mono Models: Mono 8, Mono 12 Packed, Mono 16, YUV 4:2:2, YUV 4:2:2 (YUYV) Color Models: Raw 8, Raw 12 Packed, Raw 16, YUV 4:2:2, YUV 4:2:2 (YUYV), Mono 8				
ADC Bit Depth	12 bits				
Synchronization	Via external trigger signal, via the IEEE 1394 bus, or free run				
Exposure Control	Programmable via the IEEE 1394 bus				
Power Requirements	+8 to +36 VDC supplied via the IEEE 1394 cable, < 1% ripple				
Typical Power Consumption	2.5 W @ 12 V	2.5 W @ 12 V	2.5 W @ 12 V	2.5 W @ 12 V	3.0 W @ 12 V
I/O Ports	2 opto-isolated input port 4 opto-isolated output ports				
Lens Adapter	C-mount (CS-mount optional)				
Size (L x W x H)	73.7 mm x 44 mm x 29 mm (without lens adapter or connectors) 85.5 mm x 44 mm x 29 mm (with lens adapter and connectors)				
Weight	110 g (typical)				120 g (typical)
Conformity	CE, FCC, DCAM				

Tabla 8.1: Tabla de características de la cámara utilizada

blanco y negro, lo cual nos es útil a la hora de trabajar con radiación infrarroja, en el caso de que no se utilicen filtros para esta radiación, los cuales no vienen incluidos de serie en la cámara.

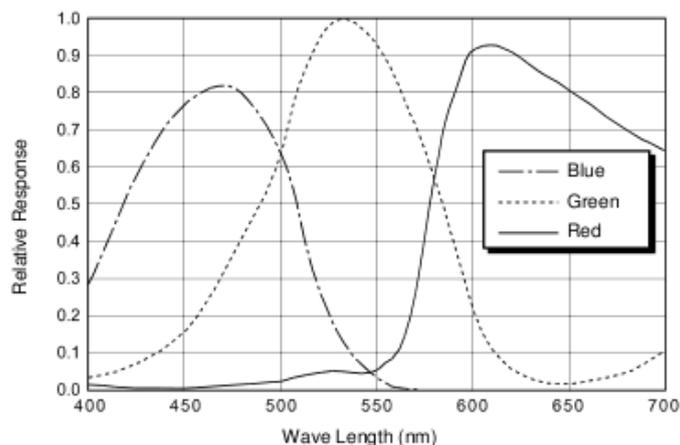


Figura 8.5: Características del modo de color de la cámara

8.3.2. Diodos de infrarrojos

El modelo de diodo infrarrojo utilizado para la corona es el OPE5685, el cual tiene, como características fundamentales una apertura de 20° aproximadamente a 3 dB de potencia, una alta velocidad de conmutación (para aplicaciones posteriores), 850 nm de longitud de onda de emisión (infrarrojo cercano), puede llegar a emitir una gran potencia.

8.3.3. Tarjeta capturadora (PCI)

A continuación se enumeran las especificaciones técnicas de la tarjeta capturadora que se incluiría en el PC (Unibrain firewirePCI400 o similar):

- 3 puertos de conexión a 100, 200 y 400 Mbps
- Alimentación desde el cable, y monitorización de la misma
- Toma de alimentación para los dispositivos conectados de $\pm 12V$.
- Compatible con PCI 2.1
- Conector de corriente interno
- Provee 3 memorias FIFO
- Soporte de DMA para PCI
- Soporte de Plug and Play (PnP)
- Capacidad de autoarranque en bus PCA para que no sea necesaria la intervención de la CPU
- Temperatura de $0^\circ C$ a $45^\circ C$.
- Humedad del 10 % al 90 % sin condensación.

8.3.4. PC

El ordenador que se utilice para ejecutar el software de reconocimiento no tiene unas especificaciones técnicas fijas, sino que tiene unos requisitos mínimos, en ciertos aspectos como son la velocidad de micro, la memoria RAM, los buses que ha de tener, etc.

- Microprocesador

Basado en x86, mononúcleo con frecuencia de reloj superior a 3.5 GHz o un sistema de doble núcleo con frecuencia de reloj superior a 2 GHz.

- Placa base

Soporte para bus PCI 2.1 o superior, si no se tiene conector Firewire de 6 pines integrado, como en Mac.

Puerto serie integrado o en tarjeta PCI para poder acceder a tarjeta de sincronización.

- RAM

1GB o superior, frecuencia de bus frontal a partir de 533 MHz, DDR2 o 3, (recomendado DDR3 833MHz)

- Disco duro

80 GB o superior, a ser posible de 7200 rpm, para la reducción del tiempo de escritura.

Parte IV

Presupuesto

Capítulo 9

Presupuesto

9.1. Cantidades necesarias, precios unitarios y presupuestos parciales

En este apartado se mostrará cuales han sido los materiales necesarios, tanto físicos como lógicos y de tiempo de desarrollo, para la implementación del sistema que se ha expuesto en este proyecto.

9.1.1. Elementos Hardware

A continuación se especifican los componentes y aparatos necesarios para el caso de una instalación de un PC en el coche para la instalación del prototipo basado en un PC, con una tarjeta capturadora firewire, 1 cámara con su correspondiente iluminador infrarrojo y un soporte hecho con ventosas y plástico, la tarjeta de sincronización para regular el nivel de iluminación infrarroja, y un conjunto conversor y estabilizador para extraer la energía eléctrica necesaria del coche.

Las características del PC añadido al presupuesto son las siguientes:

- Caja Semitorre Tacens Prior negra S/F
- Fuente de alimentación 420W Tacens radix II
- Intel Core2 Duo E7200 2.53GHz
- Refrigeración gelus pro II
- Placa base Asus P5KPL-VM SK755 DDR2 PCX M-ATX
- Memoria 2GB kingston DDR2 800MHz
- Gráfica PointOfView 8400 GS 512MB PCX DDR2
- Disco duro Maxtor 160 GB 7200rpm SATA2
- Regrabadora LG DVD-R/+RW GH22LP 22x LS negra
- Frontal multifunción lector de tarjetas Tacens

- Monitor Acer AL1511S
- Teclado
- Ratón

En el caso de utilizar un portátil, se debería utilizar una tarjeta adaptadora distinta que tenga un conector de 6 pines, a menos que se utilice un mac para la extracción, que suelen traer este tipo de conectores de serie. El empleo de otro tipo de equipos hará que el precio final varíe, puesto que para un ordenador portátil no es necesario el conversor DC/AC, así como el SAI. En cambio, se necesitará un adaptador de USB a serie para poder controlar la tarjeta de sincronización.

Descripción	Cantidad	Precio unitario (€)	Subtotal (€)
PC con monitor, ratón y teclado	1	528,51	528,51
Tarjeta capturadora firewire Fireboard-Blue 1394a PCI adapter	1	29	29
Cable firewire 1394a 9p-6p	1	18,95	18,95
Cámara Basler Scout sca1390-17fm	1	1799	1799
Óptica de alta resolución Fujinon 1:1.4/9 mm, HF9HA-1B	1	210	210
Diodos de infrarrojos OPE5685	10	0,68	6,8
Placa para iluminador de infrarrojos, proyecto Cabintec	1	7,95	7,95
Tarjeta de sincronización, proyecto Cabintec	1	33,59	33,59
Cable de conexión serie	1	9,95	9,95
Soporte de la cámara	1	42,99	42,99
Conversor DC/AC (12 VDC-220VAC 800W SP80012)	1	125	125
SAI APC SMART UPS 1500VA	1	438,41	438,41
		Total Hardware	3250,15

Tabla 9.2: Tabla de precios de elementos hardware

9.1.2. Elementos de software comercial

Respecto a esta sección, decir que el sistema operativo utilizado para el desarrollo e implementación de este sistema: linux ubuntu/kubuntu/edubuntu. Este sistema es de libre distribución

y gratuito. Las librerías asociadas para la lectura de los distintos formatos de imágenes y vídeos posibles están incluidas dentro de la distribución del sistema operativo, y también son libres y gratuitas. Los programas de edición de imágenes, como gimp también están incluidos en el sistema operativo, por lo que se aplica lo mismo que en los anteriores.

Además del sistema operativo también es necesario la instalación de las librerías de visión artificial OpenCV. Estas librerías son también de libre distribución y gratuitas.

El único software que se ha utilizado para el desarrollo de este programa y para la extracción de resultados es Matlab, el cual es software comercial y hay que pagar las correspondientes licencias, tanto por el programa base como para cada uno de los módulos adicionales que se utilicen.

Concepto	Cantidad	Precio unitario (€)	Período de amortización (años)	Subtotal (€)
Matlab base for linux	1	1950,00	3	650,00
Image Processing Toolbox	1	1000,00	3	333,33
			Total Software	983,33

Tabla 9.3: Tabla de precios de elementos de software comercial

9.1.3. Ingeniería y montaje

Por último se especifica la parte del precio del proyecto correspondiente a los gastos de ingeniería y montaje. Se considera empleados 4 meses en el desarrollo del proyecto tal y como está en este momento, a tiempo completo. Además del desarrollo se ha de tener en cuenta el tiempo de montaje de los equipos en el coche, lo cual también se incluye en este apartado.

Concepto	Cantidad	Precio unitario (€/h)	Subtotal (€)
Horas de ingeniería invertidas en desarrollo de software	672	30,00	20160,00
Horas invertidas en montaje del prototipo	4	20,00	80,00
Total Ingeniería			20240,00

Tabla 9.4: Tabla de horas de ingeniería invertidas para el desarrollo del proyecto

9.2. Presupuesto general

Por último sumamos las cantidades que se han obtenido en cada uno de los apartados anteriores y sumaremos el porcentaje establecido por el contratista para su beneficio acerca de este proyecto.

Parte	Precio
Hardware	3250,15
Software comercial	983,33
Ingeniería y montaje	20240,00
<i>Total proyecto</i>	24473,48
<i>IVA (16%)</i>	3915,76
<i>Total con impuestos incluidos</i>	28389,24
Beneficio del contratista (10%)	2838,92
Total	31228,16

Tabla 9.5: Presupuesto final

Parte V

Apéndice

Apéndice A

Manual de puesta a punto e instalación

A.1. Instalación de Linux

Para el desarrollo e implementación del sistema se ha utilizado la distribución de Linux ubuntu / kubuntu. Está basada en debian y se ha elegido por la facilidad de instalación (basado en el instalador de debian) y de configuración posterior. La versión desde la que se empezó con este proyecto a funcionar fue la 6.06.

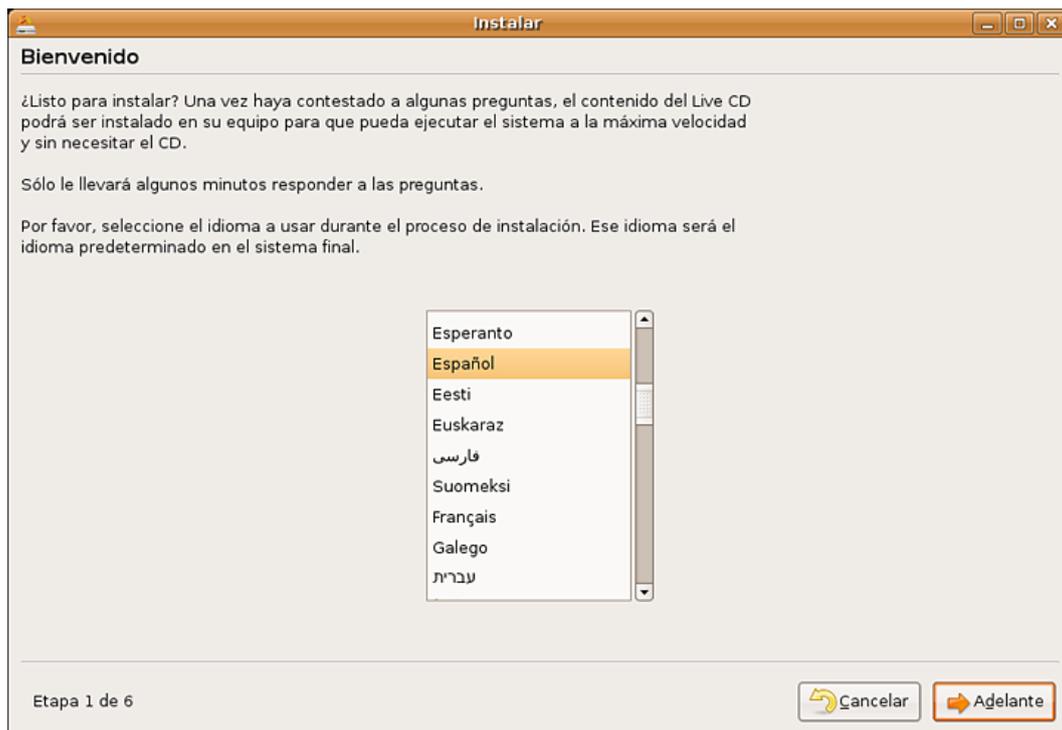
Si la versión de Linux que se va a instalar es distinta de las anteriores, las instrucciones de instalación que se indican a continuación pueden variar ligeramente, puesto que cada instalación tiene su propio instalador. Algunas distribuciones permiten la selección de paquetes que se van a instalar antes de comenzar la copia de archivos. Otras distribuciones permiten elegir el entorno gráfico con el que queremos trabajar.

Para poder instalarlo necesitamos disponer del CD live, en el cual está incluida la instalación. Debemos arrancarlo al inicio del ordenador, antes de que arranque el sistema operativo del disco duro, por lo que si no está activada la opción de arranque, habrá que activarla en la BIOS.

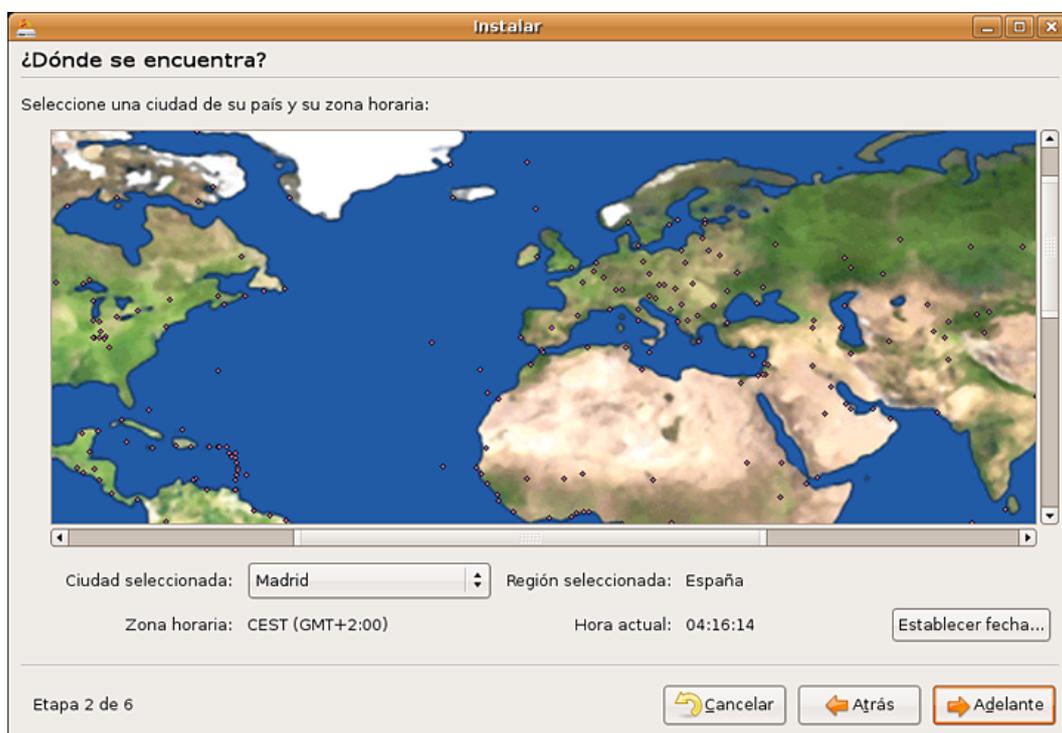
Una vez ha arrancado, tenemos dos opciones, instalarlo en modo gráfico reducido (tipo MSDOS), modo texto o arrancar el sistema operativo en modo live, probar el sistema y si nos convence instalarlo con el icono install que aparecerá en el escritorio. Los dos modos de instalación son muy parecidos, por lo que, para ilustrarlo mejor, se pasará a explicar el paso a paso del modo gráfico.



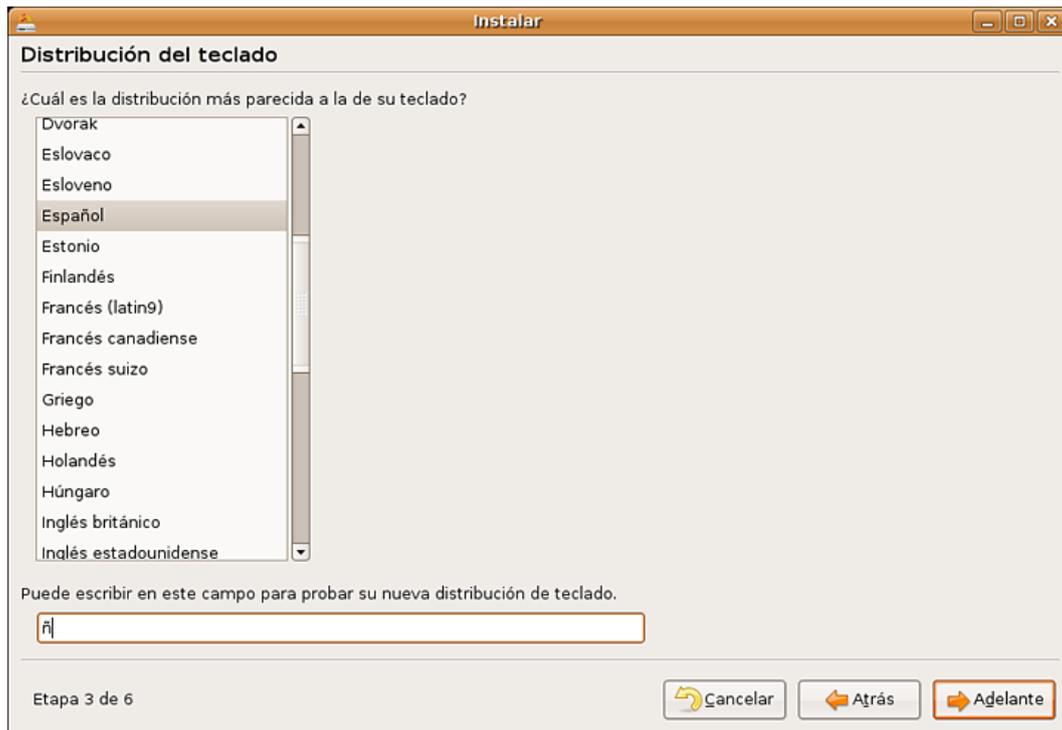
Una vez hemos dado click al icono de install, comenzará la instalación. En la primera pantalla, se pedirá el idioma que se utilizará en la instalación



Una vez se ha seleccionado el idioma, se preguntará por el país donde estamos. Si la fecha es incorrecta también se puede cambiar en esta pantalla



A continuación se preguntará sobre el tipo de teclado que tenemos instalado en el ordenador



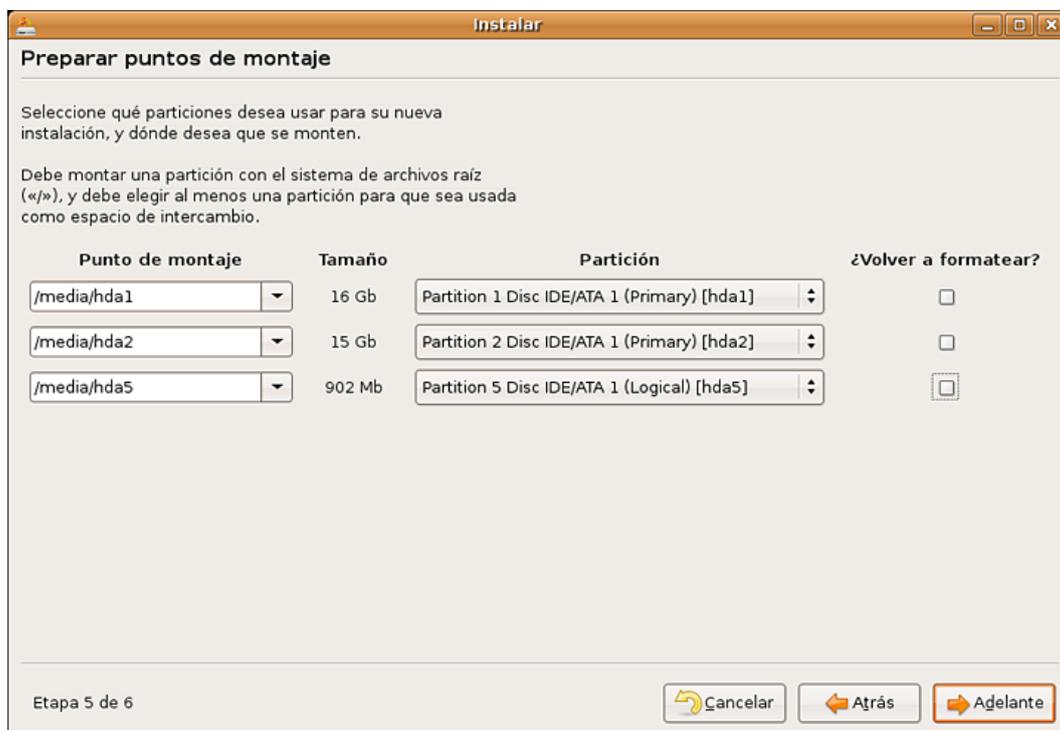
A continuación se preguntará acerca de información personal o de la compañía y del futuro sistema (el nombre del ordenador)



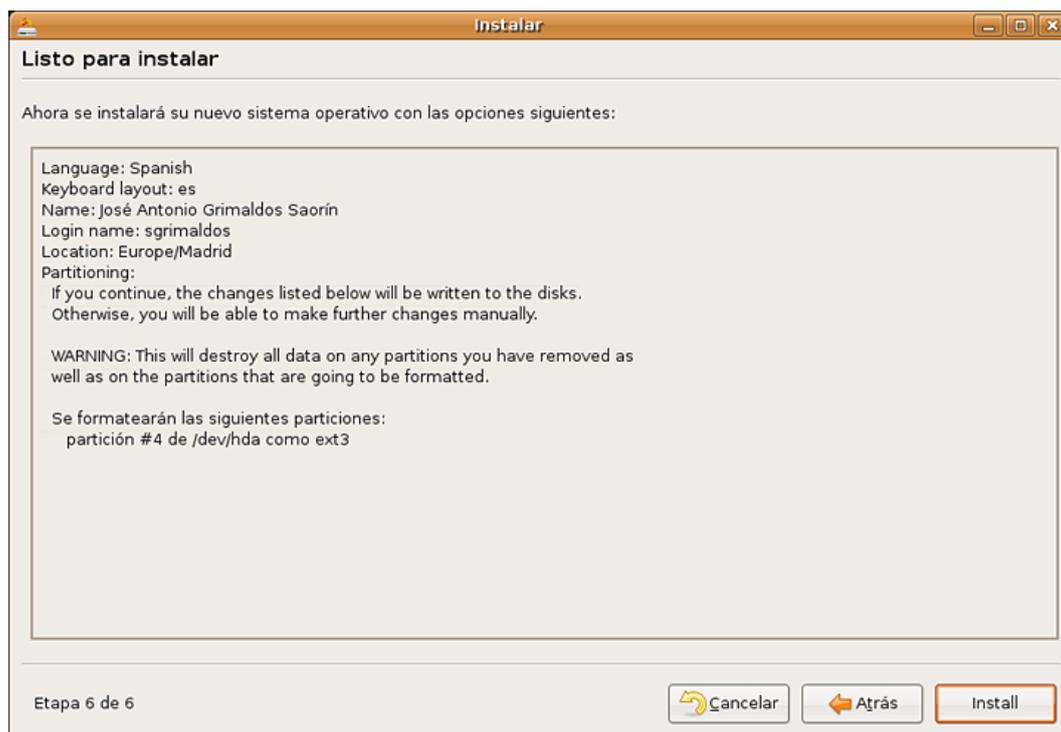
Después de rellenar estos datos, se procederá a seleccionar la partición de disco donde queremos instalar el sistema, o utilizar espacio libre o editar la tabla de particiones.



Si se opta por la configuración manual de la tabla de particiones, nos encontraremos algo como esto



Dentro de esta opción, podemos eliminar y crear y formatear particiones con distintos formatos. Una vez hayamos terminado de configurar el particionado, nos saldrá lo siguiente



A continuación, se instalará todo el sistema incluyendo las operaciones de particionado, y al final nos pedirá reiniciar. Entonces el sistema arrancará por sí solo con el cargador de arranque predeterminado (Grub)

A.2. Configuración de Linux

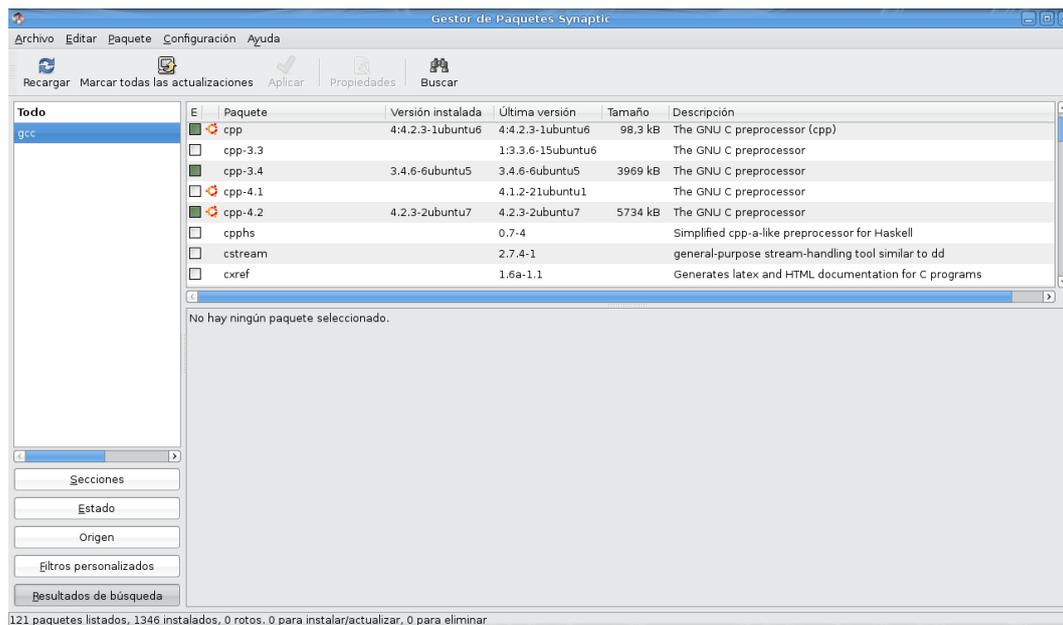
Sobre la instalación básica de este sistema operativo empezamos a instalar los paquetes necesarios para la lectura de los distintos formatos de imágenes y vídeos, codecs, librerías, compiladores, drivers y archivos de cabecera necesarios.

A.2.1. Compiladores

Para empezar, instalamos los compiladores pues es lo más básico con lo que vamos a trabajar. De todos los posibles instalamos los que más se utilizan, que son gcc y g++, en su versión 3.4 (por defecto, la distribución de Linux instala la 4.0 ó 4.2).

La instalación de la versión 3.4 de gcc y g++, puede realizarse de diversas formas:

- apt-get/aptitude install gcc-3.4 y g++-3.4 (y todas sus dependencias)
- el gestor gráfico de paquetes synaptic (opción elegida).
- paquetes precompilados, otros administradores de paquetes con repositorios, etc.



Esta versión del compilador es necesaria para la correcta compilación del paquete `ffmpeg` que necesitaremos posteriormente. Si teníamos instalada otra versión, para hacer que funcione la que necesitamos, tenemos que introducir los siguientes comandos:

1. eliminamos los links actuales

```
sudo rm /usr/bin/gcc
sudo rm /usr/bin/g++
```

2. redirigimos a la versión que necesitamos

```
sudo ln -s /usr/bin/gcc-3.4 /usr/bin/gcc
sudo ln -s /usr/bin/g++-3.4 /usr/bin/g++
```

Una vez hemos hecho esto, ya podemos pasar a instalar las librerías necesarias.

A.2.2. Firewire (IEEE 1394)

A partir de las versiones 2.14.8 del kernel de Linux, el control de estos dispositivos viene integrado en el núcleo, pero para poder utilizarlo correctamente no solo nos basta con esto, sino que tenemos que introducir algunas librerías específicas para poder tratar sus datos correctamente.

Estas librerías son: `raw1394`, `video1394`, `ohci1394` e `ieee1394`.

Las que no estén instaladas y activas, las instalamos con `synaptic` y las activamos con `modprobe` y la correspondiente librería si es que todavía no está funcionando después de instalar y reiniciar la máquina. Para este dispositivo tenemos la particularidad de que `OpenCV` puede no conectar directamente con el nombre de dispositivo adecuado, con lo que tendremos que hacer ciertos arreglos para que funcione correctamente.

Para comprobar que funciona todo correctamente, instalamos el programa `Coriander` y comprobamos que las capturas sean correctas.

A.2.3. Librerías

Necesitamos varias librerías para hacer funcionar correctamente todo lo que necesitamos en OpenCV y ffmpeg (los paquetes con el nombre de la librería correspondiente y los dev, que contienen el código fuente necesario para alguna que otra compilación o como mínimo las cabeceras)

- gtk2+
- glade
- libpng
- zlib
- libjpeg
- libtiff
- libjasper

Además de las 4 mencionadas antes para Firewire (raw, video, ohci e ieee)

A.2.4. Ffmpeg

Estas librerías proporcionan los codecs para tratar video comprimido con otros programas como pueden ser reproductores de video, plataformas de procesamiento de vídeo, etc.

Nosotros lo utilizaremos para que OpenCV posea sus librerías de procesamiento de vídeo y además para generar vídeos de prueba o de resultados a partir de otros vídeos o de imágenes ya procesadas.

Descargamos este programa en su versión 0.4.9-pre1.

La instalación de este programa se realiza como se indica en los txt de instalación:

```
./configure --enable-shared
```

En este punto podemos tener algún problema de compilación, con lo que tendremos que modificar el archivo config.mak que nos genera el script de configure.

en OPTFLAGS sustituimos los flags que teníamos antes, por los siguientes:

```
OPTFLAGS= -O3 -g -funit-at-a-time -ffast-math -fomit-frame-pointer -mmmx -msse -msse2 -msse3
```

y a continuación, compilamos y si se compila correctamente instalamos

```
make  
sudo make install
```

Después de la instalación, tenemos que añadir un par de carpetas para que la compilación de OpenCV sea correcta.

En el archivo `/etc/ld.so.conf` hay que añadir lo siguiente

```
/usr/local/lib
```

y en el archivo `.profile` (que está en la carpeta del correspondiente usuario), añadir:

```
$export LD_LIBRARY_PATH=/usr/local/lib
$export LD_LIBRARY_PATH=/usr/local/lib/opencv
```

Con estos 2 tendremos las carpetas donde están las librerías disponibles para que luego, en la compilación de OpenCV sean fácilmente accesibles y no nos den errores de compilación.

A.2.5. OpenCV

Estas son unas librerías básicas, de código abierto, que proporcionan una buena plataforma para el procesamiento de imagen y vídeo y posterior aplicación de visión artificial.

Estas librerías las descargamos desde sourceforge en su versión 1.0.

En caso de que no tengamos la cámara en la ubicación adecuada (`/dev/video1394/0`) tenemos 2 opciones para redirigirla: editar el archivo `cvcap_dc1394.cpp` y allí donde pone:

```
static char * videodev[4] = {
"/dev/video1394/0",
"/dev/video1394/1",
"/dev/video1394/2",
"/dev/video1394/3",
};
```

cambiamos los nombres de ese array por los nombres de dispositivo de vídeo que nos haya creado el sistema de forma predeterminada y recompilar OpenCV con esta modificación.

La segunda opción, que es por la que hemos optado, es crear un script con los siguientes comandos:

```
sudo mkdir /dev/video1394
sudo ln -s /dev/video1394-0 /dev/video1394/0
```

Una vez hecho esto enlazamos el script que contiene estas 2 líneas al inicio del sistema, creando un enlace simbólico en la carpeta en la que se guardan los scripts de arranque del sistema en el modo adecuado. En nuestro caso utilizamos el 5

```
sudo ln -s nombrescript /etc/rc5.d/S99video1394
```

Ahora estamos en condiciones de instalar OpenCV:

```
./configure
```

Con esta orden, comprobamos qué es lo que tiene y no instalado el sistema, para ver si tenemos todo en orden. Si entre los resultados sale que ffmpeg no está, hay que revisar la configuración de alguno de los pasos que hemos realizado, para hacer que funcione, pues es una parte básica.

Una vez nos han salido los resultados correctos, tecleamos:

```
make
```

Y si la compilación es correcta:

```
sudo make install
```

Una vez hemos comprobado que la instalación se ha realizado correctamente y hemos añadido las rutas indicadas anteriormente a los archivos `/etc/ld.so.conf` y en `.profile` de la carpeta del usuario, ejecutamos el siguiente comando para que se carguen todos los módulos que hemos instalado:

```
sudo ldconfig -v
```

Con esto cargaremos todas las librerías que acabamos de instalar y que estén en los directorios que hayamos dejado indicados en los archivos indicados anteriormente. Con la opción `-v` podemos ver por pantalla que módulos se están cargando y que módulos o no.

Con esto ya hemos terminado la configuración y podemos empezar a trabajar.

A.3. Preparación de las aplicaciones

Una vez se haya instalado todo lo que se ha visto anteriormente, en los distintos directorios, lo que quedará por hacer será ejecutar en la consola los comandos `make clean`, para borrar todos los archivos antiguos, y `make` para generar los ejecutables.

Apéndice B

Adquisición de imágenes desde cámara de vídeo

B.1. Introducción

En el sistema base, el origen de los datos se pensó que fuesen vídeos pregrabados, lo cual hacía que no se pudiese introducir una cámara directamente en el sistema. La necesidad más inmediata para poder ejecutar este sistema es la inclusión de un sistema de adquisición a partir de cámara de vídeo.

Para la adquisición mediante cámaras de vídeo, se puede elegir entre varias opciones, pero en el caso del control de las mismas mediante el programa, nos centraremos en una diferencia fundamental entre las 2: el bus que utilizan. Los posibles buses para conectar cámaras de vídeo son USB y Firewire.

USB se utiliza en webcams, o cámaras de calidad similar debido a su velocidad de transmisión en las versiones 1.0 y 1.1. En versiones posteriores de este protocolo se ha mejorado este aspecto y cámaras de mejor calidad han ido incorporando USB 2.0 como bus de transmisión de datos.

Firewire se utiliza en las cámaras de cierta calidad con velocidades de transferencia más altas. Al ser éste el bus utilizado en nuestro sistema, se describirá con más detalle más adelante.

B.2. Firewire (IEEE1394)

Este es un protocolo desarrollado por Apple Computer a mediados de los 90, y se adoptó como estándar multiplataforma bajo el nombre de IEEE1394. Actualmente la firma Sony también lo usa, bajo el nombre de i-link. Debido a su alta velocidad de transferencia y sobre todo por aquellos años, se empezó a implantar en dispositivos que necesitaban estas velocidades de transferencia y ya se está adoptando como un estándar de facto como también pasó con USB.

B.2.1. Características

La inmensa mayoría de cámaras de gama media o cámaras profesionales utilizan el bus Firewire. Este bus, desde su nacimiento tiene una alta velocidad de transferencia, de 100 a 400Mbps. En versiones posteriores se incrementó esta velocidad de transferencia a 800Mbps y por último hasta llegar a una versión de 3.2 Gbps.

Dada su alta velocidad de transferencia y sus cortos tiempos de negociación (la existencia de estos tiempos es debida a que es un bus compartido), es muy útil para dispositivos que requieran de estas características como cámaras de vídeo con gran resolución, discos duros, impresoras, grabadoras, etc. También puede emplearse para montar redes de área local, pudiendo sustituir al protocolo más extendido para este cometido, Ethernet 10/100 (IEEE 802.3). Para esta última aplicación se ha desarrollado la pila de protocolos para que funcione sobre este bus, o sea, protocolos IP y superiores, para montar aplicaciones de red y conexión de múltiples ordenadores.

Comparando alguna de las características de Firewire respecto de USB, se obtiene la siguiente tabla

	<i>Firewire</i>	<i>USB 1.1</i>
<i>Máximo de dispositivos</i>	62	127
<i>Conexión en caliente</i>	Sí	Sí
<i>Longitud máxima del cable (m)</i>	4.5	5
<i>Velocidad de transferencia básica (Mbps)</i>	200 Mbps	12 Mbps
<i>Ancho de banda, modos (Mbps)</i>	100, 200 y 400Mbps	-
<i>Plataforma</i>	Mac	PC
<i>Tipos de dispositivos conectables</i>	Videocámaras DV, cámaras de alta resolución, unidades de alta velocidad, etc	Cámaras de baja resolución, unidades lentas, módems, teclados, ratones, etc

Tabla B.1: Comparativa entre USB y Firewire

Solo con esta tabla es más que suficiente para ver cómo este bus es superior en características a USB 1.1, no así en todos los parámetros en su versión 2.0.

Tiene la capacidad de alimentar a dispositivos externos, como USB, con la diferencia de que este bus es capaz de dar más potencia que USB. USB 2.0 puede dar un máximo de 2.5 w mientras que Firewire puede entregar hasta 45w, potencia más que suficiente para alimentar dispositivos de gama alta.

Existen varios sistemas para adaptar entre los 2 formatos, para que, ordenadores que solo tengan uno de estos 2 interfaces, puedan utilizar los dispositivos del otro.

B.2.2. Funciones de acceso y control en Linux

Para ver cuáles son las funciones que nos dan acceso para el control del bus y la recogida de los datos que vienen del mismo, pasamos a describir la pila de módulos de los que está compuesto el controlador de Linux.

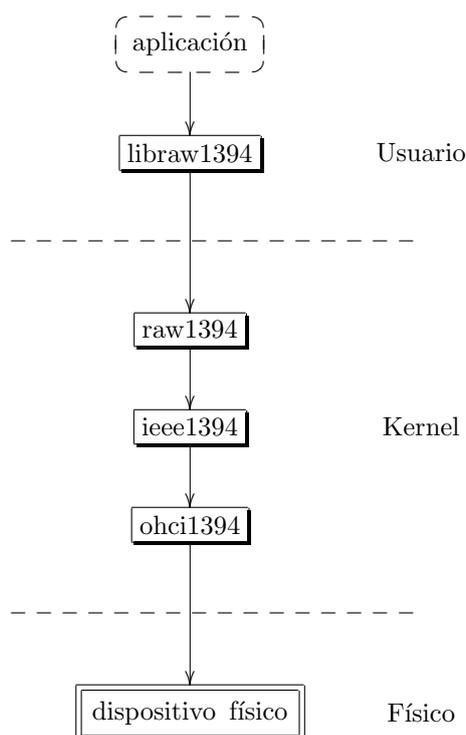


Figura B.1: Esquema de implementación del driver IEEE1394 en las distintas capas del S.O.

Todos los módulos pueden estar o no arrancados en la máquina a la vez sin que ocurran interacciones perjudiciales entre ellos.

Ohci es el módulo de más bajo nivel que interactúa con el hardware de la tarjeta. Todos los módulos de bajo nivel pueden controlar a la vez hasta 4 tarjetas diferentes.

Respecto a `ieee1394`, es un driver intermedio entre las capas bajas del mismo a las capas altas, las cuales son la interacción con la parte de usuario, que viene dado por `raw1394`.

El módulo `raw1394` es la interfaz entre los módulos de más bajo nivel, hasta el nivel de usuario. Este driver resulta ser un tanto complicado para el manejo directo por una aplicación de usuario, además que el espacio de memoria y los permisos de ejecución son diferentes, por tanto, se necesita de un driver intermedio, ya en espacio de usuario, que nos permita el acceso a funciones del driver.

Este último módulo que funciona en el plano de usuario es `libraw1394` y las librerías que incluyen las funciones necesarias, `libdc1394`. Con estos módulos se tienen las herramientas básicas para poder asignar parámetros a direcciones de memoria, adquirir y enviar datos a los dispositivos.

A partir de ahora, se describe cómo se ha de tratar con estos dispositivos y funciones para utilizar con ellos

Se necesita un manejador de dispositivo, una estructura que contenga los parámetros de la cámara y contenga la información de captura y un control del puerto al que se va a realizar la captura.

```
raw1394handle_t handle;
dc1394_cameracapture camera;
struct raw1394_portinfo ports[MAX_PORTS];
```

```
dc1394_camerainfo info;
```

Estas estructuras de datos hay que inicializarlas correctamente:

```
handle = raw1394_new_handle  
numPorts = raw1394_get_port_info
```

Para liberar el manejador (cuando terminemos)

```
raw1394_destroy_handle
```

Para adquirir el número de cámaras conectadas

```
camera_nodes = dc1394_get_camera_nodes
```

Para adquirir la información del dispositivo

```
dc1394_get_camera_info
```

Para adquirir otras informaciones desde la cámara

```
dc1394_get_camera_feature_set  
feature_set.feature[i].value;
```

Según se necesite un modo de captura u otro, se ha de llamar a una función distinta de configuración. Estos modos de configuración se verán más adelante por las particularidades que tienen. Después de la configuración dada, se procede a iniciar la transmisión de datos.

```
dc1394_start_iso_transmission
```

Para capturar una imagen de pantalla desde la transmisión

```
dc1394_single_capture
```

Para terminar la captura, se ha de llamar a la función

```
dc1394_stop_iso_transmission
```

B.2.3. Modos de adquisición

Existen 2 modos de adquisición fundamentales para la obtención de datos desde los dispositivos IEEE1394, utilizando DMA o sin ella.

Si utilizamos DMA el flujo de información está gestionado por este controlador de memoria, el cual va sirviendo imágenes a medida que vayamos consumiendo del buffer, mientras que sin

DMA el dispositivo nos dará la información en el instante que se le solicite. Esto puede tener ventajas e inconvenientes.

Si se es capaz de ejecutar el algoritmo requerido en el sistema en tiempo real, nos podemos permitir el lujo de utilizar DMA, puesto que estaremos seguros de que el buffer no se llenará con la suficiente rapidez y por tanto, el DMA no dará un error.

Sin embargo, si el algoritmo que se ejecuta por cada imagen es más pesado, el DMA se llenará con rapidez y dependiendo del tamaño de la memoria durará más o menos tiempo sin dar un error, pero al poco tiempo lo dará, por lo que se prefiere ir solicitando imágenes bajo demanda, por lo que se utilizará el primero de los tres comandos que se verán a continuación.

Las funciones que provee el driver para tratar con las distintas capturas son las siguientes:

```
dc1394_setup_capture  
dc1394_dma_setup_capture  
dc1394_dma_setup_format7_capture
```

Estas funciones configuran parámetros de la captura como la resolución, la tasa de imágenes por segundo, el modo de vídeo, etc.

Una vez hemos visto los distintos modos, pasamos a ver cómo se ha incluido esta parte en el sistema.

B.3. Inclusión en el sistema

Al probar directamente las funciones de la librería de visión artificial para realizar la captura, se observó que el programa funcionaba bien durante los primeros frames capturados, y que a partir de entonces, el programa daba un error de captura debido a que ésta se realizaba mediante DMA y el buffer que se empleaba se llenaba con rapidez en cuanto se aplicaba un tratamiento mayor que el simple hecho de llevar las imágenes obtenidas para mostrarlas en pantalla. Por esta razón se decidió obtener las imágenes bajo demanda, saltando el buffer que incluye DMA. Puesto que las imágenes se obtienen bajo demanda, existe la posibilidad de perder alguna de las imágenes, aunque en este sistema, el hecho de perder algún frame cada varios segundos no supone un problema grave.

Debido a que el sistema de adquisición implementado de forma predeterminada en OpenCV, no existe una forma sencilla de poder cambiar el modo de adquisición de las imágenes, se ha optado por implementar una clase en C++ con los métodos básicos de inicialización del dispositivo, comienzo de captura, parada de dispositivo y de captura. Además, como se necesitará posteriormente, se han incluido funciones que leen o modifican ciertos parámetros de la cámara.

Además de implementar la clase, se utilizará, en lugar de una imagen nueva con la estructura típica de OpenCV, se creará la cabecera la vez inicial y los datos de la imagen se redigirán a un array en el que se irá refrescando continuamente la información proveniente de la cámara, y cambiando el orden de los canales, puesto que normalmente se trabaja en RGB, pero OpenCV trabaja en BGR, por lo que hay que intercambiar el azul y el rojo.

Apéndice C

Manual de uso

C.1. Captador de muestras

Para poder captar las muestras desde un flujo de entrada, ya sea este una cámara de vídeo (el que se encuentra en `ident_camara`) o un vídeo pregrabado (`ident_coche`) este programa, se abrirá una ventana de terminal

En esta ventana, nos iremos al directorio donde se tenga instalado el programa a utilizar con el comando de sistema `cd`, por ejemplo:

```
cd /home/usuario/proyecto/aplicacion/
```

Una vez en el directorio, comprobamos que estén los ejecutables `ident` o alguna de sus variantes.

Todas las versiones del programa de reconocimiento incluye en su implementación el sistema de adquisición de muestras periódico. Para ejecutar las versiones que adquieren directamente de la cámara, se ha de ejecutar el programa de la siguiente forma

```
./ident -b nombreusuario
```

Si se ha grabado un vídeo para hacer este procesamiento posteriormente, se ha de ejecutar el programa con la siguiente inclusión de parámetros

```
./ident -f video.ext -b nombreusuario
```

Las extensiones posibles que se pueden introducir en el programa son múltiples (`avi`, `mpg` y otras), puesto que están instaladas las librerías de descompresión de vídeo y `OpenCV` es capaz de utilizarlas.

La ejecución de cualquiera de los dos comandos anteriores, generará una carpeta cuyo nombre es el nombre de usuario y en la que estarán incluidas las imágenes de muestra.

C.2. Base de datos faciales

C.2.1. Procedimiento para la obtención de vídeos de entrenamiento

Para una correcta obtención de los vídeos de entrenamiento, se ha elaborado un procedimiento, que se ha seguido aproximadamente en todas las fases que han requerido entrenamiento de la base de datos.

Para entrenar adecuadamente esta base de datos y hacer que al usuario se le reconozca en una cierta variedad de situaciones, se ha de tener incluida en la base de datos una variedad de gestos, posturas de la cara, giros, aunque se pretende que todos estos sean naturales, para estar más cerca de lo que en el día a día se vaya a encontrar el sistema.

En primer lugar, se sentará al usuario frente a la cámara a unos 40 cm de la misma aproximadamente, esto es, la distancia media que se va a encontrar entre el cuadro de mandos del coche y la posición natural de la cara en el habitáculo. El entorno ha de tener una iluminación uniforme y con un nivel medio suficiente, esto es, ha de estar suficientemente iluminado, en condiciones diurnas.

A continuación, se empezará a grabar obteniendo imágenes frontales del usuario. En primer lugar se pedirá al candidato que tenga un gesto normal y a continuación realice una serie de gestos habituales, como puede ser que diga algunas palabras, que sonría, esté triste, enfadado, etc. Después de grabados unos 20 segundos con este tipo gestos, pasamos a la siguiente etapa.



Figura C.1: Ejemplos de adquisición de gestos variados para la base de datos

Pasamos a pedirle al usuario que realice giros laterales con la cabeza, de izquierda a derecha o viceversa y después de arriba hacia abajo. Por último se pedirá que se realicen giros oblicuos en la cabeza para situaciones raras que se puedan dar. Para esta etapa se pedirá que se haga también de manera lenta para poder seleccionar los mejores fotogramas posteriormente, aproximadamente 1 minuto en total para la recogida de fotogramas.



Figura C.2: Ejemplos de adquisición de posibles movimientos de la cara

Con esta serie de gestos, se puede tener una base de datos suficientemente completa para realizar un reconocimiento adecuado con la iluminación uniforme. Se repetirá este procedimiento para la toma de imágenes nocturnas.

C.2.2. Estructura de la base de datos faciales

La base de datos de la que partimos tiene distintos tipos de archivos. Por un lado, tenemos las fotos de muestra para cada usuario, los archivos de proyección en el espacio de cada usuario y archivos de texto que contienen el nombre de cada usuario. Por otro lado tenemos la base de autovectores del espacio 2DPCA, y el archivo de configuración de la base de datos, en el cual incluimos información acerca de la composición de la base de datos.

El archivo de configuración de la base de datos (BaseCFG.txt) contiene los siguientes campos:

- anchura y altura de las imágenes de la base de datos (a la hora de generar la base de autovectores)
- número de autovectores utilizados
- número de usuarios en la base de datos
- número de muestras por cada usuario
- nombre del directorio base de la base de datos.

Al incluir la información de reconocimiento nocturno, la estructura de la base de datos se ha tenido que modificar para adaptarnos a esta circunstancia. El archivo de configuración de la base de datos no se ha visto modificado.

Se partía de la siguiente estructura de la base de datos:

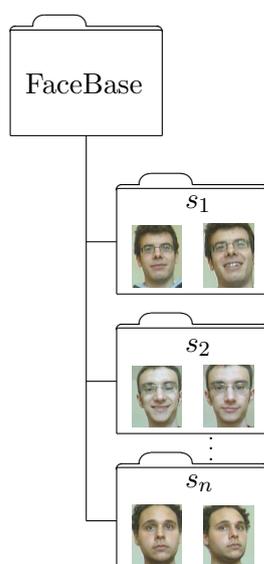


Figura C.3: Estructura de directorios en la base de datos básica

En cada carpeta de usuario se han guardado las imágenes de muestra, unos archivos con el nombre del usuario que está contenido en la carpeta, y además, los archivos que contienen la proyección de las imágenes en el espacio 2DPCA. En la carpeta FaceBase se guarda la base de autovectores seleccionados que es capaz de proyectar todas las imágenes al espacio transformado.

Para soportar el hecho de tener 2 tipos de imágenes por cada usuario, se ha decidido modificar la estructura de la siguiente manera:

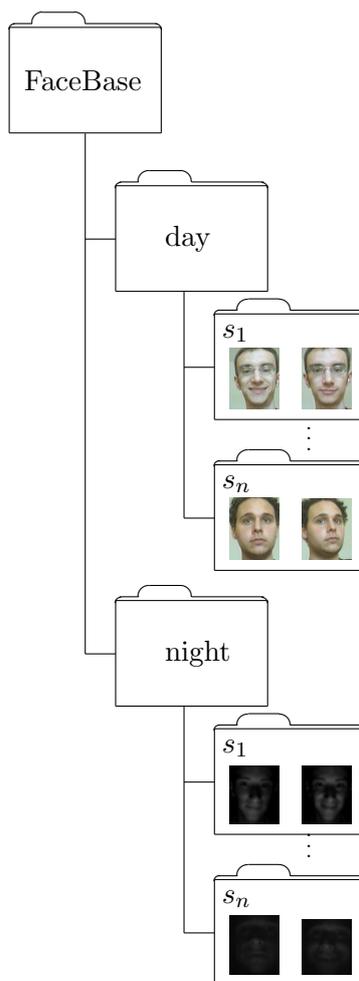


Figura C.4: Estructura de directorios en la base de datos modificada

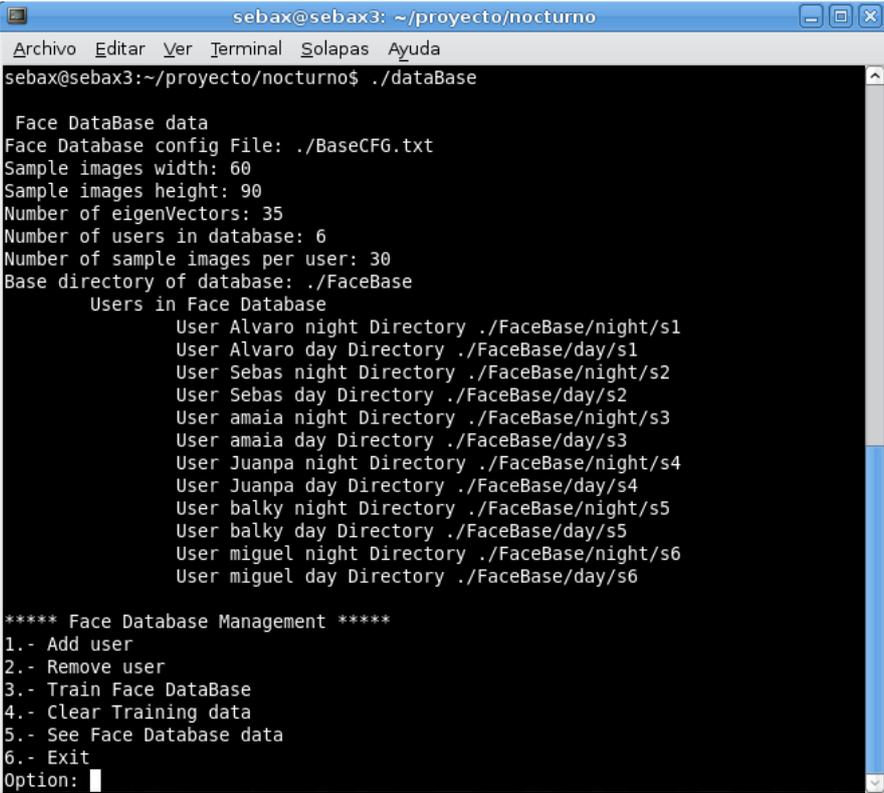
De esta manera lo único que tenemos que hacer es, cuando el programa detecta que es de día o de noche, sencillamente leer la base de datos de día o de noche. Las carpetas day y night tienen la misma estructura interna que la carpeta FaceBase anterior, con la diferencia de dónde se encuentra el archivo de información, que se encuentra en la carpeta FaceBase en lugar de en cada una de las carpetas de la base de datos, porque los usuarios se encuentran en las 2 bases para poder ser incluidos en el sistema.

C.2.3. Programa de administración

Con las carpetas de imágenes generadas por los comandos anteriores, y el archivo de configuración de la base de datos completado podemos ejecutar el programa a través del comando

```
./dataBase
```

En este programa se mostrará un menú con una serie de opciones, con las que se podrán administrar la base de datos faciales.



```
sebox@sebox3: ~/proyecto/nocturno
Archivo Editar Ver Terminal Solapas Ayuda
sebox@sebox3:~/proyecto/nocturno$ ./dataBase
Face DataBase data
Face Database config File: ./BaseCFG.txt
Sample images width: 60
Sample images height: 90
Number of eigenVectors: 35
Number of users in database: 6
Number of sample images per user: 30
Base directory of database: ./FaceBase
  Users in Face Database
    User Alvaro night Directory ./FaceBase/night/s1
    User Alvaro day Directory ./FaceBase/day/s1
    User Sebas night Directory ./FaceBase/night/s2
    User Sebas day Directory ./FaceBase/day/s2
    User amaia night Directory ./FaceBase/night/s3
    User amaia day Directory ./FaceBase/day/s3
    User Juanpa night Directory ./FaceBase/night/s4
    User Juanpa day Directory ./FaceBase/day/s4
    User balky night Directory ./FaceBase/night/s5
    User balky day Directory ./FaceBase/day/s5
    User miguel night Directory ./FaceBase/night/s6
    User miguel day Directory ./FaceBase/day/s6
**** Face Database Management ****
1.- Add user
2.- Remove user
3.- Train Face DataBase
4.- Clear Training data
5.- See Face Database data
6.- Exit
Option: █
```

Figura C.5: menú principal del programa gestor de la base de datos

En este ejemplo de ejecución existe ya una base creada, pero el funcionamiento de la misma, con el menú proporcionado es claro. A continuación vamos a especificar algunas de las acciones posibles.

C.2.4. Añadir usuario

Para poder añadir a un usuario a la base de datos, seleccionaremos la opción 1 del menú. Para que se añada correctamente, se ha de modificar la estructura de directorios generada automáticamente por el programa de toma de muestras.

Las imágenes diurnas, han de estar, dentro del directorio del usuario, en un subdirectorio llamado *day*, mientras que las imágenes nocturnas quedarán en el subdirectorio *night*.

Con esta estructura, si introducimos un usuario con la opción uno, dando el nombre del usuario que queremos introducir, o lo que es lo mismo, el nombre del directorio que contiene las muestras diurnas y nocturnas del usuario. Este directorio de usuario ha de estar en la misma carpeta que el programa *dataBase*.

C.2.5. Eliminar usuario

Para quitar a un usuario de la base de datos permanentemente, seleccionamos la opción 2, y nos preguntará por el nombre de usuario que deseamos eliminar. Se borrará la parte nocturna como la diurna, el archivo de información con el nombre del usuario y se modificará el archivo de configuración de la base de datos al que restaremos 1 al número de usuarios.

C.2.6. Entrenar base de datos

Una vez hemos añadido todos los usuarios permitidos a la base, se pasa a realizar el entrenamiento seleccionando la opción 3. En esta opción se calcula la base 2DPCA a partir de los, en este caso, 35 primeros autovectores, y éstos serán guardados en el archivo *Eigenvectors* que se encontrará en los raíces de las bases nocturnas y diurnas.

A partir de esta base se calcularán las proyecciones de las imágenes respecto del espacio calculado. Estas proyecciones se guardarán en archivos *.fti*.

C.2.7. Eliminar información de entrenamiento

Si se ha añadido o eliminado a un usuario, teniendo datos en la base, se han de borrar los datos anteriores, para poder calcular el espacio de proyección correctamente. Este comando borrará el archivo *Eigenvectors* de cada una de las bases de datos (nocturna y diurna), y todos los archivos *.fti*.

C.3. Aplicación de reconocimiento

A continuación se detallará cómo se han de ejecutar los programas para que funcione correctamente, las características y dónde está cada uno de ellos.

C.3.1. Aplicación básica y variaciones con el clasificador

Esta aplicación se encuentra en la carpeta *ident_coche*, y existen varios programas a ejecutar. *ident* es el programa básico, *identumb* es el identificador que implementa el clasificador comentado anteriormente con distancia euclídea y *identumbmah* es el clasificador que utiliza la distancia de Mahalanobis. El modo de uso para el primero es ligeramente distinto que para el resto.

```
./ident -f archivo.ext
./identumb -f archivo.ext u3 u1 u2
./identumbmah -f archivo.ext u3 u1 u2
```

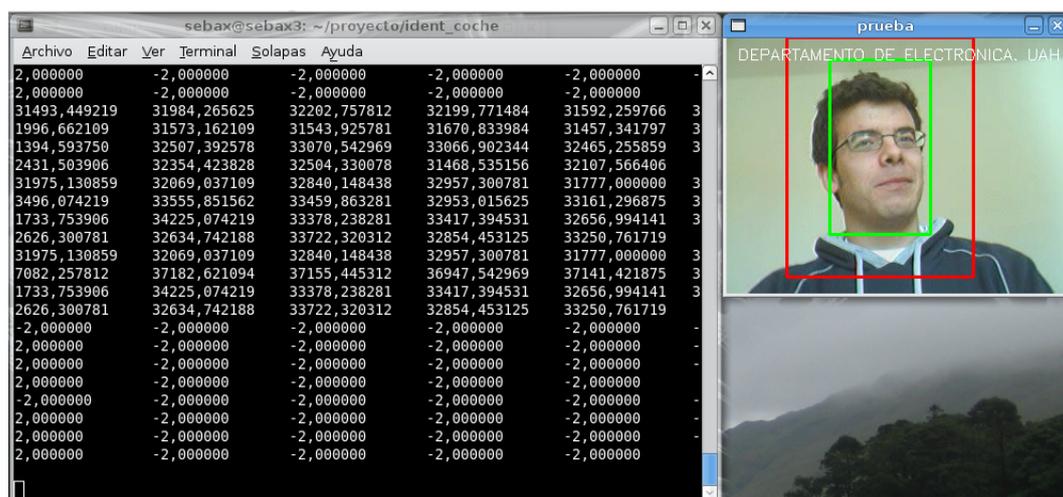


Figura C.6: Muestra de pantalla de la ejecución del programa base

Los umbrales para el clasificador son opcionales, el sistema carga unos por defecto, aunque si se quieren recoger todos los datos, conviene asignar los umbrales 0 100 100 o superiores para U_1 y U_2 .

C.3.2. Aplicación con captura desde cámara de vídeo

Este programa está en la carpeta `ident_camara`. No implementa el clasificador modificado, e implementa la parte de captura desde la cámara Firewire antes comentada.

Para ejecutarlo se ha de proceder de la siguiente manera:

```
./ident_camara
```

No son necesarios más argumentos, puesto que no hay archivos de entrada, y tampoco está implementado el clasificador.

C.3.3. Aplicaciones de tratamiento de iluminación

Cada una de estas necesita su propia carpeta y también tienen sus propios nombres dependientes del método, por tanto pasamos a especificar cada uno a continuación.

C.3.3.1. Zona de interés reducida

En este caso tenemos solo la versión cuya entrada se realiza mediante vídeo. Las versiones implementadas han sido con el clasificador y con distancia euclídea y de Mahalanobis. Estas versiones están en la carpeta `ident_coche_rec`

```
./identr -f archivo.ext u3 u1 u2
./identr2 -f archivo.ext u3 u1 u2
```

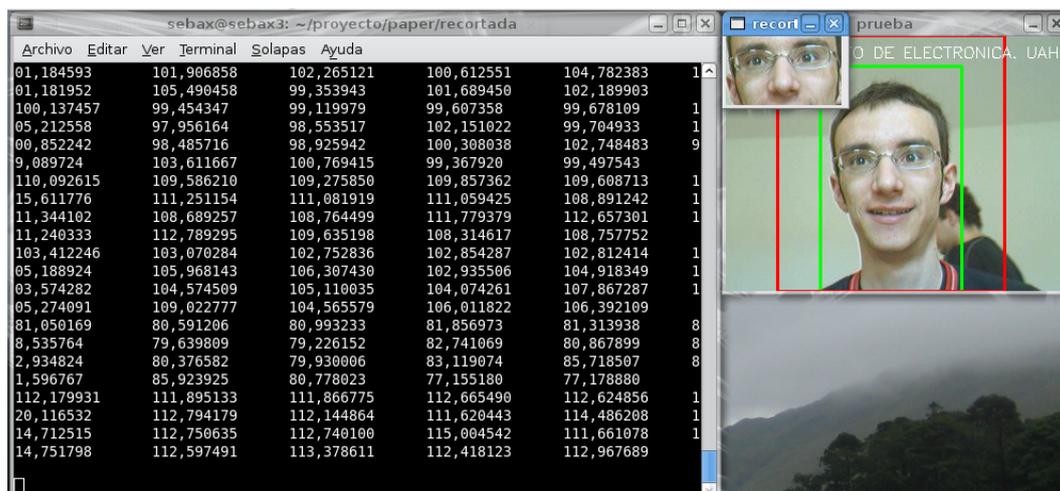


Figura C.7: muestra de pantalla del programa de cara recortada

Para adaptar las imágenes de entrada a la base de datos se ha utilizado el programa recbase, incluido en la misma carpeta, este programa lo único que hará será recortar las fotos de las dimensiones originales para adaptarlas, como se ha visto anteriormente.

C.3.3.2. Ecuación

En este caso se tienen aplicaciones tanto para origen de datos en el vídeo como en la cámara, tanto para el programa original, como para el clasificador con distancia euclídea y Mahalanobis.

```
./main
./mainoff -f archivo.ext
./main2 u3 u1 u2
./main2off -f archivo.ext u3 u1 u2
./main3 u3 u1 u2
./main3off -f archivo.ext u3 u1 u2
```

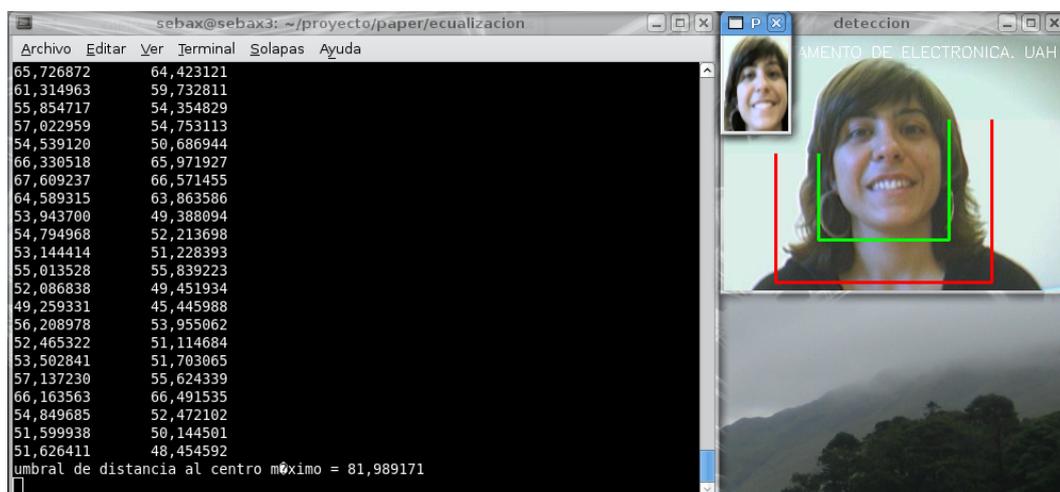


Figura C.8: Muestra de pantalla del programa de tratamiento con ecuación de histograma

Además se tiene un programa para el tratamiento previo de las imágenes, sin parámetros extra de ningún tipo.

```
./normbase
```

C.3.3.3. Máscaras de iluminación

En este caso se tienen aplicaciones para vídeo como origen de datos, para el programa original y para las 2 versiones del clasificador.

```
./ident -f archivo.ext
./identumb -f archivo.ext u3 u1 u2
./identumbmah -f archivo.ext u3 u1 u2
```

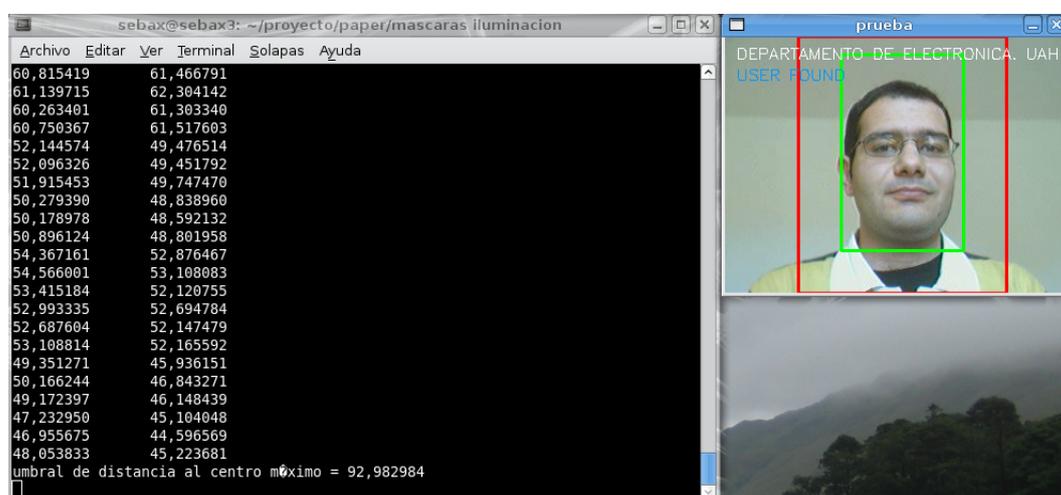


Figura C.9: Muestra de pantalla del programa de máscaras de iluminación

La aplicación que trata la base de datos, dentro de la misma, ha de tener una carpeta llamada illumtemplates y en ella han de estar las plantillas de iluminación creadas.

```
./fx2 offset min max
```

Después de varias pruebas y haber analizado cómo quedan las imágenes de la base con distintos offsets, mínimos y máximos, se ha de modificar el archivo de configuración de la base de datos, para adecuar el número de muestras al actual, siendo el generado por la aplicación $nsamples*(nplantillas+1)$.

C.3.3.4. Retinex

Como en el caso anterior, se tienen aplicaciones para el origen de datos desde vídeo o cámara, con clasificador de distancia euclídea y de Mahalanobis.

```

./Retinex
./Retinex_off -f archivo.ext
./Retinex_umb u3 u1 u2
./Retinex_off_umb -f archivo.ext u3 u1 u2
./Retinex_umb_mah u3 u1 u2
./Retinex_off_umb_mah -f archivo.ext u3 u1 u2

```

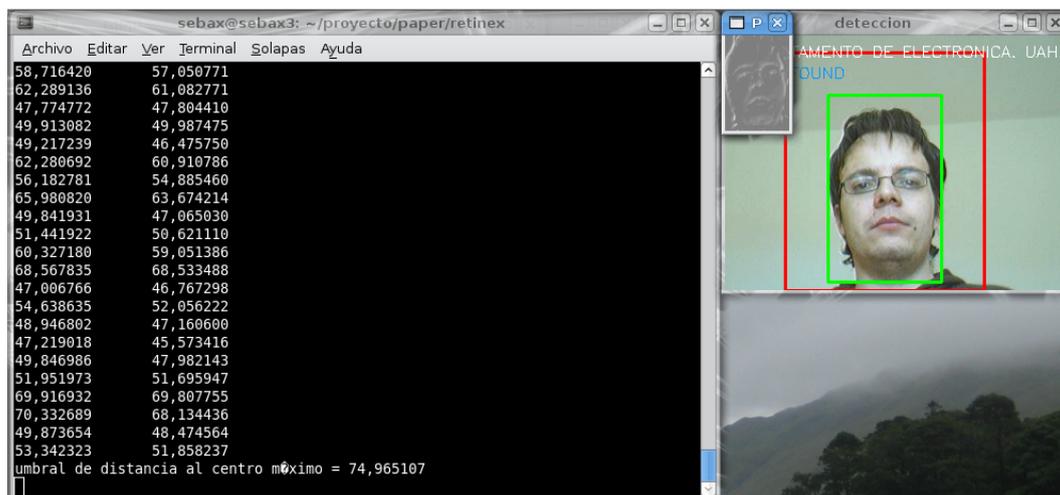


Figura C.10: Muestra de programa con tratamiento del algoritmo Retinex

En este caso, también existe programa de adaptación de imágenes de las recogidas por el programa a imágenes para procesar en la base de datos.

```
./normbase
```

C.3.4. Aplicación de detección de gafas

Este software se encuentra en la carpeta detectar gafas. Como en el caso de la cara recortada, los ejecutables a utilizar son `identr` (con distancia euclídea) e `identr2` (con distancia de Mahalanobis). Aunque el detector de gafas es independiente del clasificador, por similitud con el resto de carpetas se ha mantenido este criterio. Se ha eliminado la funcionalidad de reconocimiento de usuarios para este caso, puesto que no es necesario para comprobar el funcionamiento de la detección de gafas. La forma de ejecutarlo es igual que las anteriores, pudiendo omitir, como en los casos anteriores, los valores de los umbrales, puesto que no son utilizados.

C.3.5. Aplicación de reconocimiento nocturno y diurno

Este software se encuentra en la carpeta nocturno. En ella podemos encontrar, como en el caso base, 2 ejecutables, `identumb` (para distancia euclídea) e `identumbmah` (para distancia de Mahalanobis). en este caso los programas tienen como entrada los vídeos de prueba. Para ejecutar los programas se procederá del mismo modo que las aplicaciones anteriores.

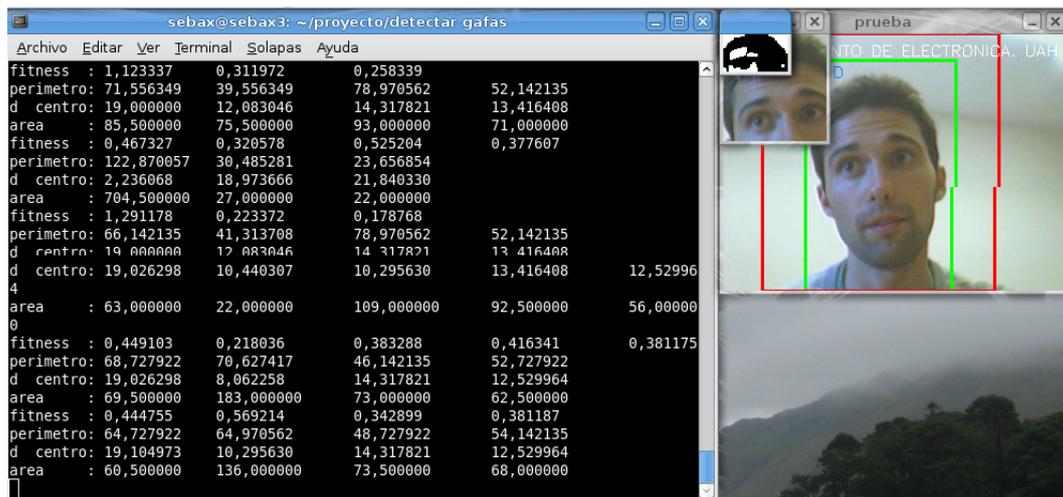


Figura C.11: Muestra de pantalla del programa de detección de gafas

```
./identumb -f archivo.ext u3 u1 u2
./identumbmah -f archivo.ext u3 u1 u2
```

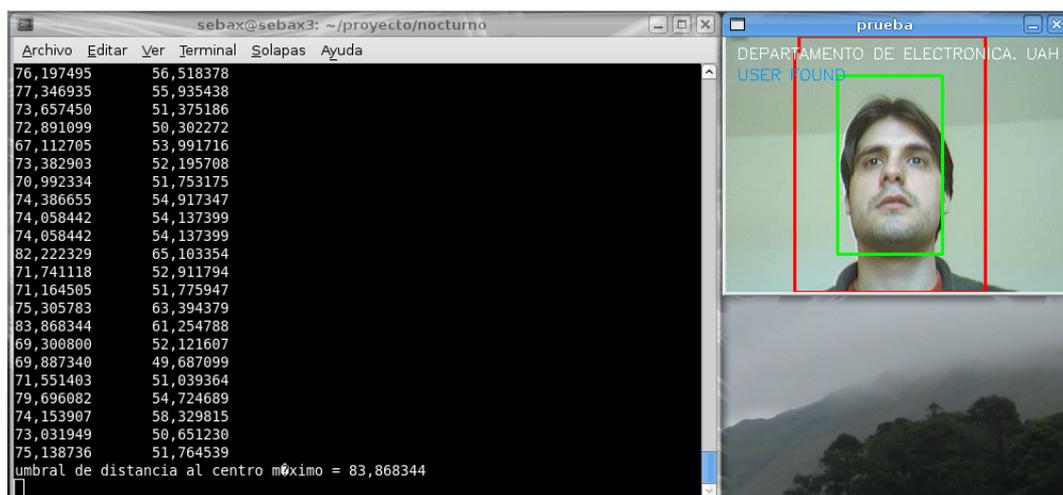


Figura C.12: Muestra de pantalla del programa de detección diurna y nocturna con imágenes diurnas

En este caso tenemos un modo de ejecución especial, para realizar un filtrado de mediana a las imágenes de entrada en caso de estar ante un vídeo diurno. para que funcione, habrá que añadir la opción `-s`

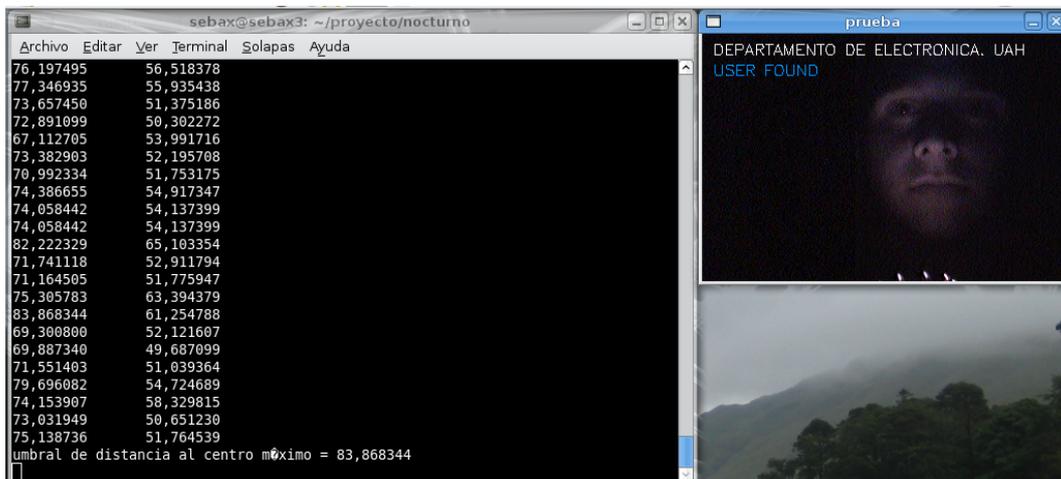


Figura C.13: Muestra de pantalla de imágenes nocturnas y diurnas con imágenes nocturnas

Apéndice D

Contenido del DVD

Bibliografía

- [1] B. Abidi, S. Huq, and M. Abidi. Fusion of visual, thermal, and range as a solution to illumination and pose restrictions in face recognition. In *Proc. 38th Annual 2004 International Carnahan Conference on Security Technology*, pages 325–330, 11–14 Oct. 2004.
- [2] A. Adler. Automatic face recognition system architecture for collaborative virtual environments. In *Proc. IEEE International Workshop 2002 HAVE Haptic Virtual Environments and Their Applications*, pages 1–6, 17–18 Nov. 2002.
- [3] A. Amira and P. Farrell. An automatic face recognition system based on wavelet transforms. In *Proc. IEEE International Symposium on Circuits and Systems ISCAS 2005*, pages 6252–6255, 23–26 May 2005.
- [4] Luis M. Bergasa, Jesús Nuevo, Miguel A. Sotelo, Rafael Barea, and Elena Lopez. Visual monitoring of driver inattention. 2007.
- [5] Luis Miguel Bergasa and Manuel Mazo. *Apuntes de Visión Artificial del master de Sistemas Electrónicos Inteligentes*. Universidad de Alcalá, Departamento de Electrónica, 2005.
- [6] David H. Brainard and Brian A. Wandell. Analysis of the retinex theory of color vision. *Journal Optical Society of America*, 3(10):11, October 1986.
- [7] Sebastián Bronte, Luis M. Bergasa, Jesús Nuevo, and Rafael Barea. Sistema de reconocimiento de conductores. *Workshop on Physical Agents, WAF2008*, pages 163–169, 2008.
- [8] Xin Chen. *PCA-Based Face Recognition in Infrared Imagery: Baseline and Comparative Studies*. PhD thesis, Univerity of Notre Dame, 2003.
- [9] Autor Desconocido. Biometría facial en el aeropuerto de barajas. *Revista Auditoría y Seguridad*, 9:94–97, 2007.
- [10] Bo Du, Shiguang Shan, Laiyun Qing, and Wen Gao. Empirical comparisons of several preprocessing methods for illumination insensitive face recognition. *Acoustics, Speech, and Signal Processing*, 2005.
- [11] Shan Du and R. Ward. Wavelet-based illumination normalization for face recognition. In *Proc. IEEE International Conference on Image Processing ICIP 2005*, volume 2, pages II–954–7, 11–14 Sept. 2005.
- [12] E.G.Ercoreca. El robo de coches cuesta a las aseguradoras 80 millones. *www.cincodias.com*, 2006.
- [13] E.R.Daives. *Machine Vision: Theory, Algorithms, Practicalities*. Morgan Kauffman, 2005.

- [14] Forsyth and Ponce. *Computer Vision, A modern Approach*. Prentice Hall, 2003, 2003.
- [15] Wen Gao, Shiguang Shan, Xiujuan Chai, and Xiaowei Fu. Virtual face image generation for illumination and pose insensitive face recognition. *International Conference On Multimedia and Expo*, 2003.
- [16] Marius Herscovitz, Evgeny Artyomov, and Orly Yadid-Pecht. Improving the global impression of brightness of the multi scale retinex algorithm for wide dynamic range pictures. *The VLSI Systems Center, Ben-Gurion University*.
- [17] Glenn Hines, Zia ur Rahman, Daniel Jobson, and Glenn Woodell. Single scale retinex using digital signal processors.
- [18] Kyong I.Chang, Kevin W.Bowyer, and Patrick J. Flynn. Face recognition using 2d and 3d facial data. *Workshop in multimodal User Authentication*, pages 25–32, December 2003.
- [19] J.M.Buenaposada. *Análisis de expresiones faciales mediante visión por computador*. PhD thesis, Universidad politécnica, 2005.
- [20] J.M.Buenaposada, E.Muñoz, and L.Baumela. Efficiently estimating facial expression and illumination in appearance-based tracking. *Proceedings of BMVC, Edimburgh, UK*, September 2006.
- [21] J.R.Parker. *Algorithms for Image Processing and Computer Vision*. Wiley, 1996.
- [22] Bernd Jähne and Horst Haußecker. *Computer Vision and Applications, A Guide for Students and Practitioners*. Academic Press, 2000.
- [23] Seong G. Kong, Jingu Heo, Bisma R. Abidi, Joonki Paik, and Mongi A. Abidi. Recent advances in visual and infrared face recognition, a review. *Computer vision and image understanding*, 97:103–135, 2005.
- [24] Martin D. Levine, Maulin R. Gandhi, and Jisnu Bhattacharyya. Image normalization for illumination compensation in facial images. *Department of Electrical & Computer Engineering & Center for Intelligent Machines. McGill University, Montreal, Canada*, August 2004.
- [25] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. *IEEE ICIP*, 1:900–903, September 2002.
- [26] L.M.Bergasa. *Apuntes de la asignatura de control neuronal y borroso*. Universidad de Alcalá, Departamento de Electrónica.
- [27] L.M.Bergasa, J.A. Hernández, R. Barea, E. López, and M. S. Escudero. Sira, a robotic system for elderly assistance. *35th International Symposium on Robotics*, March 2004.
- [28] Yasuyuki Matsushita, Ko Nishino, Katsushi Ikeuchi, and Masao Sakauchi. Illumination normalization with time-dependent intrinsic images for video surveillance. *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.
- [29] Manuel Mazo and Daniel Pizarro. *Apuntes de visión de la asignatura de Sistemas Electrónicos de Control*. Universidad de Alcalá, Departamento de Electrónica, 2005.
- [30] Ajmal S. Mian, Mohammed Bennamoun, and Robyn Owens. An efficient multimodal 2d-3d hybrid approach to automatic face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11), November 2007.

- [31] Tim Morris. *Computer Vision and Image Processing*. Palgrave Macmillan, 2004.
- [32] Nissan. Concept car to showcase anti drunk-driving technology. *www.nissan-global.com*, 2007.
- [33] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, mar 1979. minimize inter class variance.
- [34] Laiyun Qing, Shiguang Shan, and Wen Gao. Eigen-harmonics faces: Face recognition under generic lighting. *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.
- [35] Shiguang Shan, Wen Gao, Bo Cao, and Debin Zhao. Illumination normalization for robust face recognition against varying lighting conditions. *IEEE International Workshop on Analysis and modeling of faces and gestures*, 2003.
- [36] Mihoko Shimano, Kenji Nagao, Takahiro Okabe, Imari Sato, and Yoichi Sato. Face recognition under varying illumination based on map estimation incorporating correlation between surface points. *Lecture notes in computer science*, 2006.
- [37] Diego A. Socolinsky and Andrea Selingerz. A comparative analysis of face recognition performance with visible and thermal infrared imagery. *International Conference On Pattern Recognition*, 2002.
- [38] Anuj Srivastava and Xiuwen Liu. Statistical hypothesis pruning for identifying faces from infrared images. *Image and Vision Computing*, 20:100–109, 2002.
- [39] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *CVPR*, 2001.
- [40] W.Zhao, R.Chellappa, P.J.Philips, and A.Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, December 2003.
- [41] Wendy S. Yambor, Bruce A. Draper, and J.Ross Beveridge. Analyzing pca-based face recognition algorithms: Eigenvector selection and distance measures. Technical report, Computer Science Department, Colorado State University, 2000.
- [42] Jian Yang, David Zhang, Alejandro F. Frangi, and Jing yu Yang. Two-dimensional pca: A new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 2004.
- [43] Lei Zhang and Sen Wang. Face synthesis and recognition from a single image under arbitrary unknown lighting using a spherical harmonic basis morphable mode. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:209–216, 2005.
- [44] Miguel Ángel Sotelo Vázquez. Transparencias de tratamiento de imagen aplicado a sistemas inteligentes de transporte.