

LOCALIZACIÓN Y RECONOCIMIENTO DE ROSTROS EN
IMÁGENES MONOCULARES DE FRENTE CON VARIACIÓN EN
ESCALA

Jorge Servín Pérez

2009

Dedicatoria

A las familias: Servín, Pérez y Servín Pérez. Con mucho cariño.

Agradecimientos

Quiero expresar mi más profundo agradecimiento a todas las personas de la Universidad Autónoma Metropolitana, Unidad Cuajimalpa, que tuvieron a bien participar en este proyecto proporcionando los datos, fotografías y demás facilidades requeridas para el mismo.

Quiero agradecer a la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, por la oportunidad que me brindó de estudiar el posgrado en sus aulas. Así también, agradezco el apoyo y enseñanza por parte de los profesores del posgrado.

Agradecimiento especial al Consejo Nacional de Ciencia y Tecnología por los apoyos económicos prestados en los dos últimos trimestres.

Resumen

En este trabajo se presenta una nueva técnica de localización y reconocimiento de rostros en imágenes monoculares de frente con variación en escala. Se describe la construcción del sistema de adquisición de imágenes para generar una base de datos de fotografías que permitan ser comparadas contra imágenes con cambio de escala. Además, se presenta como crear la base de datos de imágenes de prueba, así como los resultados en la localización de ojos, la extracción de la región elíptica de la cara, la normalización, el suavizado con variación total y la aplicación de la técnica de análisis de componentes principales para la generación del espacio de eigenfaces y obtención del rostro promedio. Finalmente, se presentan resultados comparables en eficiencia de reconocimiento de rostros de técnicas similares.

Abstract

In this work we present a novel technique for face detection and recognition of front monocular images with scale variation. A data acquisition set was carefully assembled in order to complete a standardized face photograph database, for implementing the matching procedure with different scales photographs. Furthermore, we present how to create database test images, eyes localization results, elliptic face feature extraction, normalization, total variation deblurring, and principal components analysis applications for building the eigenface space, and mean face. Finally, we present results at the same level of efficiency of similar techniques for face recognition.

Índice General

Índice General	11
Lista de Figuras	15
Lista de Tablas	18
1 INTRODUCCIÓN	19
1.1. Estado del Arte	19
1.1.1. El rostro visual	19
1.1.2. Los cambios en el rostro	20
1.1.3. Automatización del reconocimiento facial	20
1.2. Descripción de Capítulos	25
2 PROBLEMA DE INVESTIGACIÓN	27
2.1. Planteamiento del Problema	27
2.2. Objetivos	28
2.2.1. Objetivo general	28
2.2.2. Objetivos particulares	28
2.3. Justificación	28
2.4. Metodología	29
3 ANTECEDENTES TEÓRICOS	31
3.1. Conceptos Fundamentales	31
3.1.1. Procesamiento digital de imágenes	31
3.1.2. Reconocimiento de patrones	36
3.1.2.1. Segmentación de un objeto basado en el umbralizado	36

3.1.2.2. Etiquetado de regiones o componentes conectadas	39
3.2. Análisis de Componentes Principales	39
3.2.1. Conjunto de datos	40
3.2.2. Vector promedio	41
3.2.3. Matriz de covarianza	41
3.2.4. Componentes principales	42
3.3. La Técnica <i>Eigenface</i>	43
3.3.1. Resolución de las imágenes	43
3.3.2. Cálculo de eigenfaces	44
3.3.3. Cómo utilizar los eigenfaces para reconocimiento de rostros	45
3.4. Normalización	46
3.5. Variación Total	47
3.5.1. Conceptos preliminares	47
3.5.2. Restauración de la imagen	48
3.5.3. Discretización de la variación total de una imagen	49
3.6. Procesamiento Multiresolución (Pirámide de Imágenes)	52
4 PROTOCOLO DE ADQUISICIÓN DE IMÁGENES	55
4.1. Proceso de adquisición de imágenes	55
4.2. Resultados del proceso de adquisición	57
4.3. Observaciones	58
5 LOCALIZACIÓN Y RECONOCIMIENTO DE ROSTROS	63
5.1. Imágenes de Entrada	63
5.2. Localización Facial	65
5.3. Extracción de Características	65
5.4. Reconocimiento de Rostros	67
5.4.1. Imagen de prueba	67
5.4.2. Normalización	67
5.4.3. Bases de datos	68
5.4.4. Selector de escala	69
5.4.5. Rostro promedio	69
5.4.6. Proyección	69

5.4.7. Prueba de similitud	70
5.4.8. Sistema de Rechazo	70
5.5. Identificación	71
6 PRUEBAS Y RESULTADOS	73
6.1. Inicialización de la Base de Datos	73
6.1.1. Localización y extracción de características	73
6.1.2. Normalización	74
6.1.3. Cálculo de rostro promedio y eigenfaces	74
6.1.4. Generación de la pirámide de imágenes	81
6.2. Pruebas de Reconocimiento de Rostros	82
6.2.1. Pruebas de reconocimiento de la base de datos	87
6.2.2. Pruebas de reconocimiento de imágenes con variación en escala	87
7 CONCLUSIONES Y PERSPECTIVAS	97
7.1. Conclusiones	97
7.2. Perspectivas	98
Bibliografía	99
A Aplicaciones	105
B Hoja de registro	107
C Artículos de Investigación	109
C.1. Modelo experimental que verifica la ley de caída libre usando una cámara de video .	110
C.2. Imagen promedio de un conjunto de rostros de las carreras de Ingeniería en Com- putación y Matemáticas Aplicadas	120
D Códigos de programas en MATLAB	127
D.1. Variación total (programa)	128
D.2. Extracción de la región elíptica del rostro (programa)	131
D.3. Normalización de las regiones elípticas en resolución y suavizado con TV (programa)	140
D.4. Cálculo de eigenfaces (programa)	143
D.5. Generación de la pirámide de imágenes (programa)	148

D.6. Verificar imágenes que conforman las bases de datos (programa) 150

D.7. Reconocimiento de rostros, prueba de verificación de cambios de escala 155

Lista de Figuras

1.1. Problemas de investigación en la automatización del reconocimiento facial.	21
1.2. Configuración de un sistema genérico de reconocimiento facial.	22
1.3. Fotografía. a) Rostro original; b) Mapa de piel; c) Mapa de ojos.	24
1.4. Región de la elipse. a) Figura geométrica; b) Región elíptica en el rostro.	24
1.5. Sistema de reconocimiento facial basado en eigenspace estándar.	25
2.1. Fotografías superpuestas. Rostros con variación en escala (ancho, alto y profundidad). . .	28
2.2. Etapas que componen la metodología de este proyecto.	29
3.1. Vecindades. a) $N_4(p)$ y b) $N_8(p)$, del píxel p con coordenadas (m,n)	32
3.2. En métrica 4, el punto p es adyacente al punto q y no al punto r	32
3.3. Camino de longitud 4 en métrica 8.	33
3.4. Bajo la métrica 8, la región se encuentra completamente conectada. Bajo métrica 4, la región debe ser vista como tres regiones 4-conectadas con 8, 4 y 2 elementos respectivamente.	33
3.5. Regiones conectadas.	34
3.6. Una imagen y su histograma.	34
3.7. Cubo de 24 bits RGB. a) Esquema, normalizado a valores entre $[0,1]$; b) Cubo de colores.	36
3.8. Imagen en color (superior) convertida a escala de grises (media) y binarizada (inferior) con umbralizado por el método de Otsu.	37
3.9. (a) Imagen binaria; (c-d) Secuencia de etiquetados.	40
3.10. Vector \mathbf{x} formado con píxeles de n imágenes.	41
3.11. Resultado de la regularización por TV para una función con ruido (imagen superior) y para una imagen con un mandril (imagen inferior). Las imágenes de la izquierda son las originales, las siguientes muestran los resultados de la regularización para los valores de $\alpha=0.0001, 0.001, 0.01, 0.1$ y 1.0	49

3.12. Estructura de imagen piramidal.	53
3.13. Filtro Gaussiano pasa bajas en el dominio del espacio.	53
4.1. Set fotográfico.	56
4.2. Diagrama esquemático	56
4.3. Registro de datos por persona.	57
4.4. Imágenes faciales en color, adquiridas en el set fotográfico e integrarán la base de datos del sistema de reconocimiento facial.	59
4.5. Imágenes con cambio de escala y las fotos correspondientes al rostro.	60
4.6. Ejemplos de imágenes de prueba. Observese el cambio de escala en el rostro de cada imagen.	60
4.7. Nuevo conjunto de imágenes para integrar la base de datos.	61
5.1. Diagrama a bloques del sistema de reconocimiento automático de rostros con variación en escala.	64
5.2. Ejemplos de imágenes de entrada.	64
5.3. Fases en la extracción de características.	65
5.4. Componentes 8-conectadas.	66
5.5. Ejemplo de normalización y suavizado. a) Resolución 274x198 píxeles; b) Resolución 241x175 píxeles; c) Suavizado con variación total.	68
5.6. Niveles 1-7.	68
5.7. Rostros femeninos normalizados en tamaño, iluminación y con suavizado.	70
5.8. Imagen promedio, rostros femeninos.	70
5.9. Conjunto de eigenfaces para rostros femeninos.	71
5.10. Ejemplo de identificación.	71
6.1. Imágenes faciales que integrarán la base de datos del sistema de reconocimiento de rostros.	75
6.2. Extracción de la región de la elipse (imágenes en color) en el rostro.	76
6.3. Conjunto de imágenes normalizados en tamaño y suavizado con TV.	78
6.4. Rostros masculinos normalizados en tamaño, iluminación y suavizados con TV.	79
6.5. Imagen promedio, rostros masculinos.	79
6.6. Conjunto de Eigenfaces para rostros masculinos.	80
6.7. Rostros femeninos normalizados en tamaño, iluminación y suavizados con TV.	80
6.8. Imagen promedio, rostros femeninos.	80

6.9. Conjunto de eigenfaces para rostros femeninos.	81
6.10. Imagen promedio, para rostros mixtos.	81
6.11. Conjunto de eigenfaces, para rostros mixtos.	82
6.12. Nivel 7.	83
6.13. Nivel 6.	83
6.14. Nivel 5.	84
6.15. Nivel 4.	84
6.16. Nivel 3.	85
6.17. Nivel 2.	85
6.18. Nivel 1.	86
6.19. Niveles 1-7.	86
6.20. Ejemplos de reconocimiento. a) Nivel 8; b) Nivel 7; c) Nivel 6; d) Nivel 5; y e) Nivel 4.	88
6.21. Ejemplos de imágenes de prueba. Observese el cambio de escala en el rostro de cada imagen.	89
6.22. Imágenes con cambio de escala y las fotos correspondientes al rostro.	89
6.23. Elipses reducidas a la resolución de 92×112	90
6.24. Imágenes de prueba con variación en escala reducidas a, elipses con resolución de 92×112 , con variación en escala.	90
6.25. Resultado de la prueba, identificación de las cinco imágenes de prueba con variación en escala en la resolución más baja, cinco aciertos.	92
6.26. Nuevo conjunto de imágenes para integrar la base de datos.	93
6.27. Extracción de la región de la elipse (imágenes en color) del nuevo conjunto de rostros.	94
6.28. Conversión a niveles de grises de la región de la elipse del nuevo conjunto de rostros.	95
6.29. Imágenes normalizadas en resolución y suavizado con TV del nuevo conjunto de rostros.	95
B.1. Hoja de registro.	107
C.1. Artículo presentado en la primera Semana de la Computación y Matemáticas Aplicadas 2008, UAM Cuajimalpa.	111
C.2. Artículo presentado en la primera Semana de la Computación y Matemáticas Aplicadas 2008, UAM Cuajimalpa.	121

Lista de Tablas

1.1. Técnicas de reconocimiento facial	23
1.2. Bases de datos públicas de imágenes faciales.	24
4.1. Lista de materiales.	56
4.2. Adquisición de imágenes.	58
6.1. Parámetros del conjunto de elipses.	77
6.2. Niveles en la pirámide de imágenes.	82
6.3. Resultados de Reconocimiento para 49 imágenes.	87
6.4. Parámetros del nuevo conjunto de elipses.	91

Capítulo 1

INTRODUCCIÓN

En esta sección se presenta una aproximación al estado actual de la investigación en reconocimiento de rostros y se mencionan algunos estudios que formarán parte del desarrollo de este proyecto, localización de ojos y extracción elíptica de la cara, así como reconocimiento de rostros basado en eigenspace, entre otros.

1.1. Estado del Arte

El problema de reconocimiento de rostros [1], no es un mero problema de reconocimiento de formas o patrones o imágenes estáticas, considera tareas computacionales aun más generales relativas a la percepción de objetos tridimensionales proyectados en un plano fotográfico en escenas dinámicas y con variabilidad en la iluminación, escala, ruido, defectos naturales de las personas, envejecimiento, maquillaje, lentes y peinado.

1.1.1. El rostro visual

El rostro juega un papel vital en nuestra vida diaria por su amplia variedad de funciones. Al visualizar la expresión facial podemos determinar, en alguna medida, elementos tales como: el modo de juzgar, el carácter o las intenciones. Otras señales faciales, más explícitas, nos permiten estimar factores como: la edad, género, origen étnico y la identidad. Esta última, como señal visual para reconocimiento, es una de las tareas principales de investigación en el área de reconocimiento de rostros o reconocimiento facial.

1.1.2. Los cambios en el rostro

El proceso de identificación de una persona a partir de su apariencia facial tiene que ser realizada en la presencia de una serie de factores que entran en conflicto, alteran la apariencia y hacen la tarea aun más difícil. El rostro puede cambiar su apariencia debido a factores ya sea *intrínsecos* o *extrínsecos*. Las variaciones intrínsecas toman lugar independientemente de cualquier observador y es debida a la naturaleza física del rostro. Por otro lado, las variaciones extrínsecas aparecen cuando el rostro es observado vía la interacción: luz, rostro y observador

Generalmente la apariencia de un rostro se altera considerablemente dependiendo de las condiciones de iluminación y, en particular, de la proyección de la sombra propia. Las características de la cámara (o lente) usadas para observar el rostro también afectan la calidad de la imagen. Otros objetos en la escena pueden causar oclusión, y sombras, así como alteraciones en la luz incidente. Una de las fuentes de variación más significativas es el cambio en la posición. Es de notar que, la posición de un rostro está determinada por la orientación y posición tridimensional (3D) del observador. La causa principal del cambio en la posición es relativa al *movimiento rígido* entre el observador y el sujeto. Un rostro sufre movimiento rígido cuando este cambia su posición y orientación en el espacio tridimensional relativo al observador. Un rostro también puede sufrir un *movimiento no rígido* cuando su forma tridimensional cambia debido, por ejemplo, al hablar o por su expresión facial. Esto resulta en una variación intrínseca de la apariencia. Aunque estos tipos de movimiento ocurren juntos, en general se conviene en tratarlos de forma separada. En este trabajo se considera el rostro en forma rígida con una expresión neutra.

1.1.3. Automatización del reconocimiento facial

En la automatización del reconocimiento facial, el esfuerzo ha sido multidisciplinario y en esta tarea han participado áreas tan variadas como las ciencias de la computación, la ciencia cognitiva, la matemática, la física, la psicología y la neurobiología. Dentro de las aplicaciones deseables del reconocimiento de rostros basado en computadora, se incluyen interfaces hombre-máquina, multimedia, vigilancia, seguridad, teleconferencia, comunicación, animación, etc. Por consiguiente, durante los años recientes se han hecho grandes esfuerzos de investigación en el estudio y desarrollo de modelos computacionales, algoritmos y sistemas de visión por computadora para el reconocimiento automático de rostros.

Gross *et al.* [2], menciona que han emergido sistemas capaces de alcanzar razones de reconocimien-

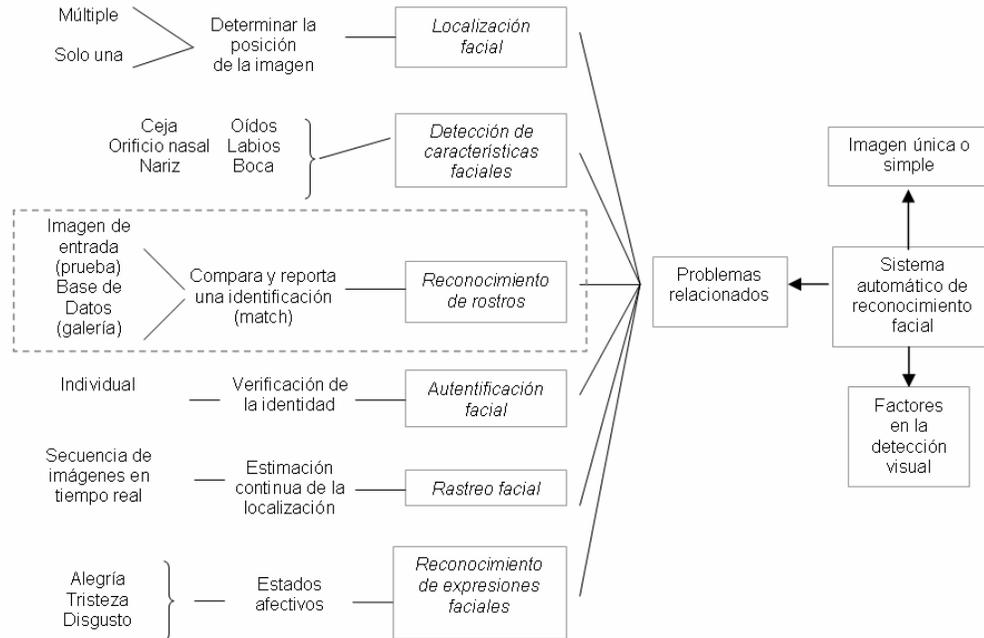


Figura 1.1: Problemas de investigación en la automatización del reconocimiento facial.

to que exceden el 90 % de precisión bajo condiciones controladas. Por otro lado, la variación en la escala de factores tales como la iluminación, la oclusión y la expresión facial entre otras aún no es muy bien entendida.

Yang *et al.* [3], confirma que la detección de rostros desde una imagen es un reto dada la variabilidad en la escala, ubicación, orientación, y posición (Fig. 1). Por otro lado, factores como: cambios en la expresión facial, oclusión, y diferentes condiciones de iluminación, cambian la apariencia de los rostros.

En la Fig. 1.1 se ilustran los problemas relacionados que componen un sistema general de reconocimiento facial, estos se detallan a continuación.

- Localización facial (*face localization*). Su objetivo es determinar la posición de un único rostro; este es un problema de detección que se simplifica asumiendo que una imagen de entrada sólo tiene un rostro [4, 5].
- Detección de características faciales (*facial feature detection*). La meta es detectar la presencia y localización de características tales como: ojos, nariz, boca, etc., asumiendo que una imagen de entrada sólo tiene un rostro [6].

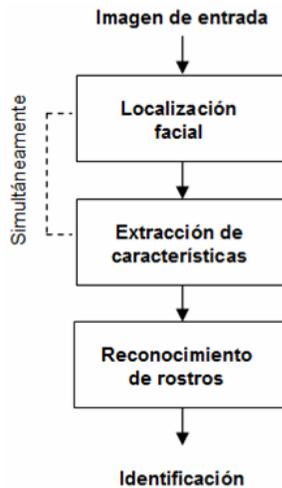


Figura 1.2: Configuración de un sistema genérico de reconocimiento facial.

- Reconocimiento de rostros (*face recognition*). Compara una imagen de entrada (de prueba) contra una base de datos (galería) y reporta una identificación (comparación acertada) si hay alguna [7, 8, 9]. Con respecto al reconocimiento con variación en escala Turk y Pentlad [7] presentan resultados del orden del 60% de aciertos.
- Autenticación facial (*face authentication*). El propósito es verificar el reclamo de un individuo en una imagen de entrada.
- Rastreo facial (*face tracking*). Métodos que estiman de manera continua la ubicación y posible orientación de un rostro en una secuencia de imágenes en tiempo real.
- Reconocimiento de expresiones faciales (*facial expression recognition*). Tiene que ver con la identificación de estados afectivos (alegría, tristeza, disgusto, etc.) en rostros humanos.

Zhao *et al.* [10], mencionan que un sistema automático de reconocimiento facial genérico tiene que considerar las siguientes etapas (Fig. 1.2): a) Imagen de entrada; b) Detección y extracción de características faciales; c) Reconocimiento facial e identificación.

Así también, indican que los avances en técnicas de reconocimiento facial (Tabla 1.1), se pueden clasificar en tres grupos amplios: en métodos holísticos (*holistic methods*), en métodos basados en características (*feature-based methods*) y en métodos híbridos (*hybrid methods*). En el primer caso, las imágenes faciales son representadas como un arreglo bidimensional de píxeles y son comparados contra una o varias plantillas que representan el rostro completo. Este trabajo de tesis emplea la técnica eigenface, ya que aquí se aplican algunos de los modelos matemáticos más utilizados en

Tabla 1.1: Técnicas de reconocimiento facial

Aproximación
Métodos Holísticos
<i>Análisis de componentes principales (PCA)</i>
Eigenfaces
Eigenfaces probabilísticos
Fisherfaces/subespacio LDA
SVM
Evolución
Líneas características
ICA
<i>Otras representaciones</i>
LDA/FLD
PDBNN
Métodos basados en características
Métodos geométricos
Arquitectura de enlace dinámico
Modelos de Markov
Convolución de red neuronal
Métodos híbridos
Eigenfaces modulares
LFA híbridos
Forma normalizada
Basados en componentes

reconocimiento de rostros, en particular el método de análisis de componentes principales (*PCA*, *Principal Component Analysis*) que se emplea para reducir la dimensionalidad alcanzando buenos rendimientos en tiempo y complejidad computacional, este método se describe más adelante, en la sección 3.2. En el segundo caso, se toman como base algunas características geométricas del rostro como la distancia entre diferentes puntos faciales, para su comparación. El tercer caso es una combinación de los dos anteriores.

Hajati *et al.* [11], introducen un método para localización facial y reconocimiento en imágenes de color. El método propuesto emplea la localización de ojos por cómputo y extracción de la región elíptica de la cara (Figs. 1.3 y 1.4).

Ruiz-del-Solar [12], presenta un estudio de reconocimiento de rostros basado en eigenspace, que corresponde a una de las metodologías más utilizadas para el reconocimiento de rostros en imágenes digitales. Mencionan que se han propuesto diferentes aproximaciones basadas en eigenspace: estándar, *differential* o kernel eigenspace. El estudio considera aspectos teóricos así como simulaciones empleando bases de datos públicas “Yale Face Database” y “FERET” (Tabla 1.2). La Fig. 1.5 muestra un ejemplo, diagrama a bloques, de un sistema de reconocimiento basado en eigenspace.

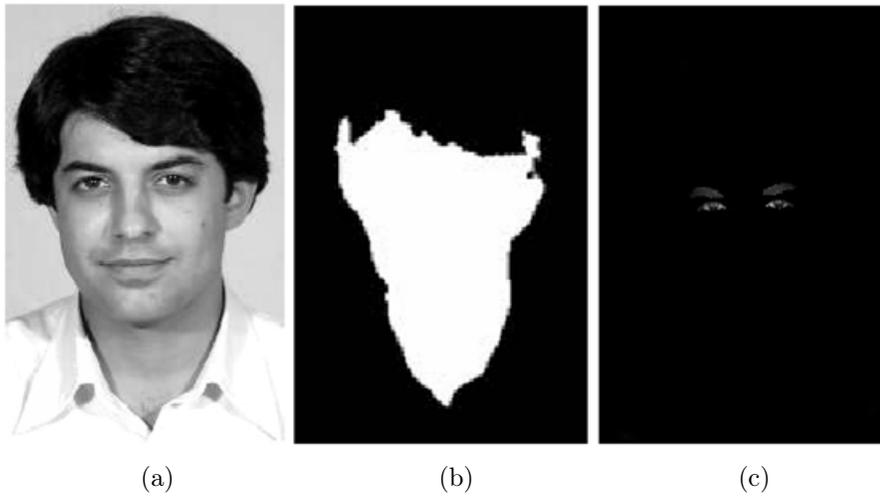


Figura 1.3: Fotografía. a) Rostro original; b) Mapa de piel; c) Mapa de ojos.

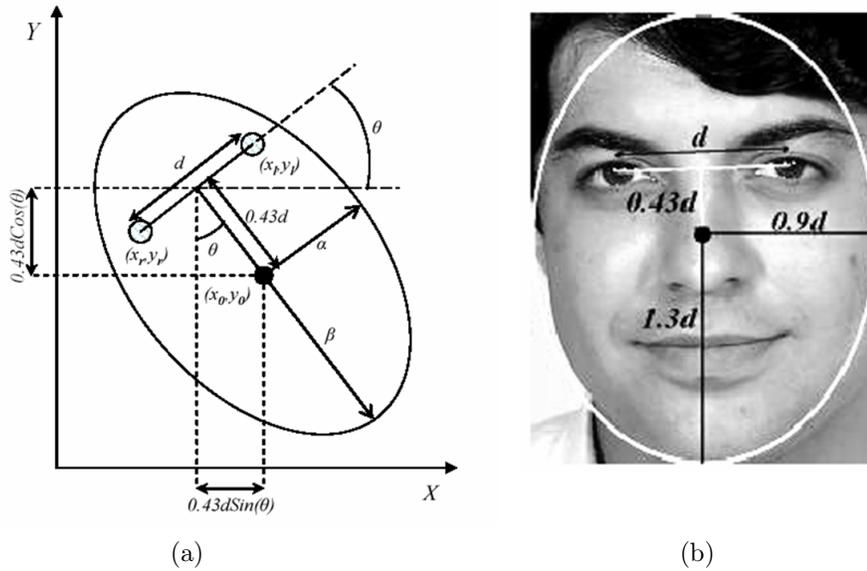


Figura 1.4: Región de la elipse. a) Figura geométrica; b) Región elíptica en el rostro.

Tabla 1.2: Bases de datos públicas de imágenes faciales.

Bases de datos públicas	
MIT	ftp://whitechapel.media.mit.edu/pub/images/
FERET	http://www.nist.gov/humanid/feret
Yale	http://cvc.yale.edu
AT&T	http://www.uk.research.att.com
Harvard	ftp://ftp.hrl.harvard.edu/pub/faces/

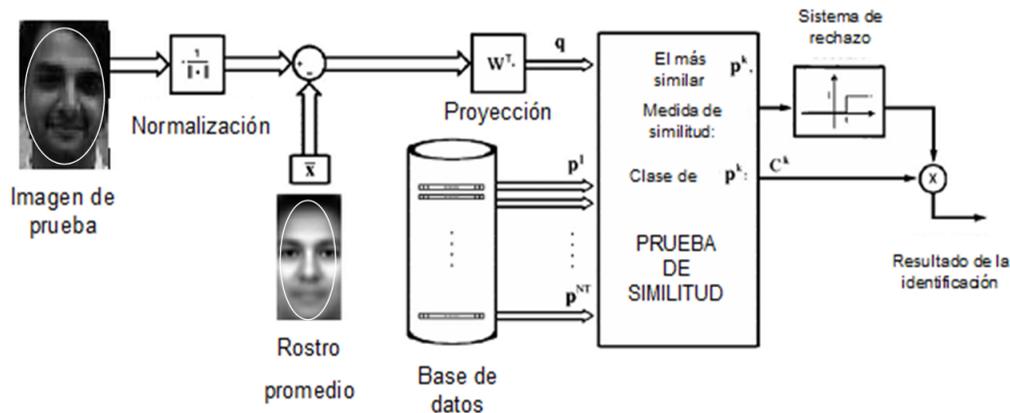


Figura 1.5: Sistema de reconocimiento facial basado en eigenspace estándar.

1.2. Descripción de Capítulos

En el estado del arte, no se ve claro que el reconocimiento de rostros con variación en la escala de las imágenes sea un problema resuelto. Si el ser humano es capaz de hacer este reconocimiento, es de esperar que un sistema automático artificial lo pueda hacer. ¿Cuáles son los factores que dificultan este reconocimiento? ¿Cómo se puede aproximar una mejor solución a este problema? Para dar respuesta a esta y otras interrogantes se presentan en este proyecto las técnicas que permiten hacer el estudio de este problema. A continuación se describen los capítulos siguientes que conforman este trabajo de investigación.

En el capítulo 2, Problema de Investigación, se describen el problema de investigación, objetivos, justificación y metodología.

En el capítulo 3, Antecedentes Teóricos, se presentan los Conceptos Fundamentales relativos a Procesamiento Digital de Imágenes y Reconocimiento de Patronos. También se presentan las técnicas propuestas para el desarrollo de este proyecto de tesis. Análisis de Componentes Principales, se utiliza para reducir la dimensionalidad de los datos (imágenes en este proyecto). La Técnica Eigenface, hace posible el reconocimiento de rostros. La Normalización, mejora la probabilidad de la identificación y consiste en ajustar las imágenes en resolución, suavizado e iluminación. Variación Total, se utiliza para suavizar las imágenes, se incluye, además, una sección que muestra cómo discretizar la variación total de una imagen. Procesamiento Multiresolución, con esta técnica se puede cambiar la escala de las imágenes.

En el capítulo 4, Protocolo de Adquisición de Imágenes, se presentan la construcción del set fotográfico y la adquisición de imágenes, para la creación de la base de datos, así como de prueba.

1. INTRODUCCIÓN

En el capítulo 5, Sistema de Reconocimiento Automático de Rostros, se estudia como se hace: la localización facial, la extracción de características, la normalización, la obtención del rostro promedio y eigenfaces, la generación de la pirámide de imágenes, y el reconocimiento, que forman parte del sistema de reconocimiento automático de rostros.

En el capítulo 6, Pruebas y Resultados, se muestran las pruebas de localización y reconocimiento de rostros así como los resultados de las mismas.

Finalmente, en el capítulo 7, Conclusiones, se presentan las observaciones finales que resultarán en este proyecto, así como, las recomendaciones de trabajo futuro.

Capítulo 2

PROBLEMA DE INVESTIGACIÓN

En esta sección se describe el problema de investigación así como los objetivos del proyecto. Hay que hacer notar que los objetivos reflejan el estado actual de la investigación, así como las sugerencias de los revisores del proyecto. Esto es, los objetivos se han afinado y, en particular, los objetivos específicos presentan la certidumbre natural que resulta después de concluir las etapas que conforman este proyecto.

2.1. Planteamiento del Problema

Para el estudio del problema de variación en escala en el reconocimiento de rostros se propone: aplicar un modelo matemático e implementarlo en un programa computacional para la identificación de personas en fotografías con la característica de variación en escala. Consideremos la imagen de la Fig. 2.1, esta consiste de dos fotografías superpuestas, imágenes monoculares, ambas contienen la imagen de un rostro pero con variación en la escala (ancho, alto y profundidad). Es de interés, que el modelo y programa computacional propuesto sea capaz de analizar ambas imágenes y determinar si corresponden o no a la misma persona. Es decir, se pretende desarrollar un programa computacional que permita de forma automática, por una parte, determinar la presencia de un rostro en la imagen bidimensional y localizar la posición donde se encuentre, a continuación, identificar el rostro detectado. Esto es, comparar el rostro detectado contra una base de datos y determinar si se encuentra dentro de la base de datos .

Este proyecto busca hacer análisis de imágenes bidimensionales (estáticas), como las que corresponden a fotografías, esto es, el proyecto se acota al problema de reconocimiento de rostros, aunque se tomaran en cuenta elementos relativos a los problemas de localización y detección de características.

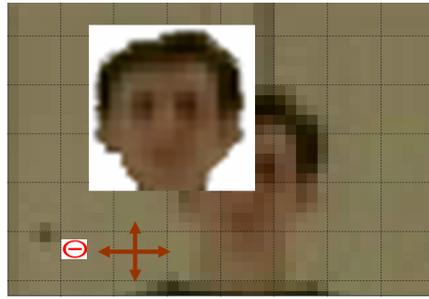


Figura 2.1: Fotografías superpuestas. Rostros con variación en escala (ancho, alto y profundidad).

Están fuera del alcance de este trabajo las áreas de autenticación facial, análisis de imágenes en movimiento, así como de reconocimiento de expresiones faciales.

2.2. Objetivos

2.2.1. Objetivo general

Aplicar un modelo matemático para la localización y reconocimiento de rostros en imágenes monoculares (imágenes bidimensionales adquiridas con un cámara con un solo lente) con variación en escala en fotografías a color que contengan un solo rostro e implementarlo en un programa computacional.

2.2.2. Objetivos particulares

- Modelar el reconocimiento de rostros a través de las técnicas: Análisis de Componentes Principales y *Eigenfaces*.
- Aplicar Variación Total para el suavizado de imágenes.
- Localizar y extraer características, mapa de piel y mapa de ojos, en imágenes en color.
- Aplicar procesamiento multiresolución, para obtener la pirámide de imágenes que permita el manejo de cambios de escala en fotografías.

2.3. Justificación

El reconocimiento de rostros con variación en escala tiene aplicaciones potenciales en la identificación de una persona con propósitos de seguridad, salud o para algún tipo de registro o servicio,

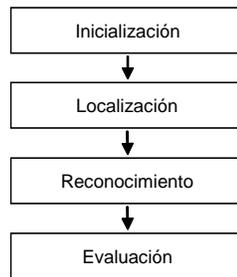


Figura 2.2: Etapas que componen la metodología de este proyecto.

por ejemplo, pasar lista en un salón de clases, así como para aplicaciones de teleconferencia, comunicación, animación, etc.

2.4. Metodología

Para alcanzar los objetivos arriba planteados de manera que culminen en el dispositivo tecnológico de interés, es necesario considerar las siguientes etapas: inicialización, localización, reconocimiento y evaluación (Fig. 2.2).

En la etapa de inicialización se recopilan los elementos materiales para crear un set fotográfico que permita hacer la adquisición de las imágenes de prueba y las imágenes que formarán parte de la base de datos. Las imágenes de prueba serán imágenes que contengan rostros en condición de variación en escala. Para las etapas siguientes se crea un programa computacional que implemente las técnicas más eficientes para localización y reconocimiento facial. En la etapa de localización se determina si existe un rostro en una imagen de prueba dada. En caso de que exista un rostro entonces se reescala para que pase a la etapa de reconocimiento en la cual se verifica si existe una imagen facial en la base de datos que empate con la imagen de prueba. Finalmente, en la etapa de evaluación se verifica la eficiencia del sistema determinando la cantidad de aciertos y errores en el reconocimiento de rostros.

Capítulo 3

ANTECEDENTES TEÓRICOS

A continuación se exponen las bases teóricas sobre las que se sustenta este proyecto. Por su importancia, se enfatizan las técnicas de análisis de componentes principales, la técnica eigenface, normalización, variación total y el procesamiento multiresolución.

3.1. Conceptos Fundamentales

Los algoritmos computacionales de este proyecto se fundamentan en conceptos de las áreas del procesamiento digital de imágenes y reconocimiento de patrones. En esta sección se describen algunos de los conceptos de tratamiento de imágenes mayormente empleados en las técnicas de localización y reconocimiento facial.

3.1.1. Procesamiento digital de imágenes

Una *imagen* se define [13, 14, 15], como una función bidimensional $f(x, y)$, donde x y y son coordenadas espaciales (plano) y la amplitud de f en cualquier par de coordenadas (x, y) es llamado la *intensidad* o *nivel de gris* de la imagen en ese punto.

Si los valores de los niveles de gris son cantidades discretas (enteros) entonces a la imagen se le llama *imagen digital*. Esta imagen digital se compone de elementos llamados *píxeles* (*picture elements*).

El campo del *procesamiento digital de imágenes* se refiere al procesamiento de imágenes digitales por medio de una computadora.

Una *imagen binaria* denotada como $b(x,y)$ es una imagen digital $f(x,y)$ cuantizada a dos niveles de intensidad, 0 y 1.

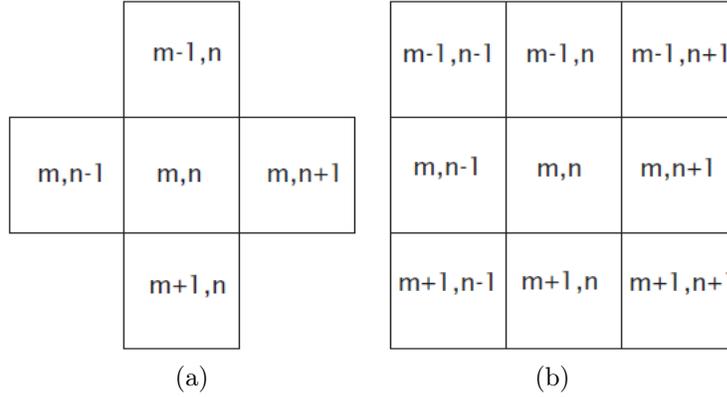


Figura 3.1: Vecindades. a) $N_4(p)$ y b) $N_8(p)$, del píxel p con coordenadas (m,n) .

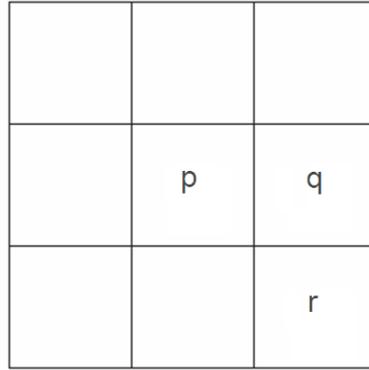


Figura 3.2: En métrica 4, el punto p es adyacente al punto q y no al punto r .

Funciones de distancia o métricas d_4 y d_8 de base k . Sean $p(m,n)$ y $q(u,v)$ con $m,n,u,v \in Z \times Z$ y $k=\{4,8\}$, entonces la distancia euclidiana entre p y q para ambas métricas estarán dadas por:

$$d_4(p,q) = |m-u| + |n-v| = \sqrt{(m-u)^2 + (n-v)^2} \quad (3.1)$$

$$d_8(p,q) = \text{máx}(|m-u|, |n-v|) \quad (3.2)$$

Vecindades $N_4(p)$ y $N_8(p)$. Sea (A, α) un espacio métrico bidimensional α una métrica cualquiera, la vecindad de un píxel p denotada por $N(p)$ es el conjunto de píxeles tales que $0 < \alpha(m,n) \leq 1$. En la Fig. 3.1 se muestran ambos tipos de vecindades $N_4(p)$ y $N_8(p)$.

Puntos conectados. Dos puntos p y q se dicen estar conectados si q está en el conjunto $N_\alpha(p)$.

Puntos adyacentes. Un punto p es adyacente al punto q si ambos están conectados (Fig. 3.2).

Un *camino* entre dos puntos p y q en A es una secuencia de puntos distintos p_1, p_2, \dots, p_n tales que $p_0 = p$ y $p_n = q$, p_i es adyacente a p_{i-1} , $1 \leq i \leq n$ y n es la longitud del camino. Puede haber

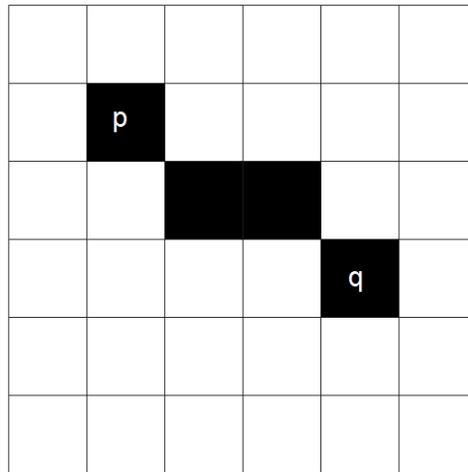


Figura 3.3: Camino de longitud 4 en métrica 8.

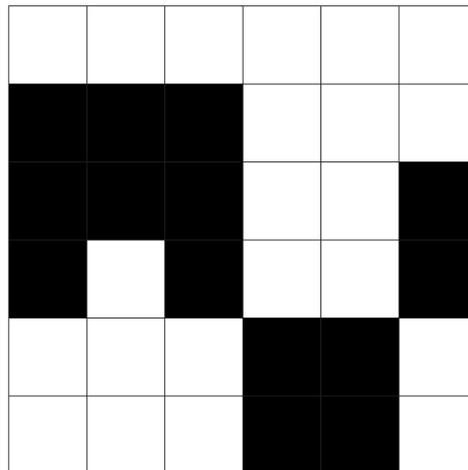


Figura 3.4: Bajo la métrica 8, la región se encuentra completamente conectada. Bajo métrica 4, la región debe ser vista como tres regiones 4-conectadas con 8, 4 y 2 elementos respectivamente.

dos tipos de caminos: 4 y 8 conectados. En la Fig. 3.3 se muestra un camino, en métrica 8, entre los puntos p y q . La longitud del camino es de 4 píxeles.

Una *región conectada* $R \in b(m, n)$ es un conjunto de puntos $p(m, n)$ tales que existe un camino que comienza en p_1 y termina en p_2 para todo p_1 y $p_2 \in R$. Al igual que en el caso de caminos, puede haber regiones 4 y 8 conectadas. En la Fig. 3.4 se muestra una región 8-conectada de 14 elementos. La misma región en métrica 4 debe ser vista como tres regiones 4-conectadas con 8, 4 y 2 elementos respectivamente.

Una región puede ser simplemente conectada (sin hoyos) o múltiplemente conectada (con hoyos). En la Fig. 3.5 se muestra un objeto simplemente conectado, en forma de T, y otro múltiplemente

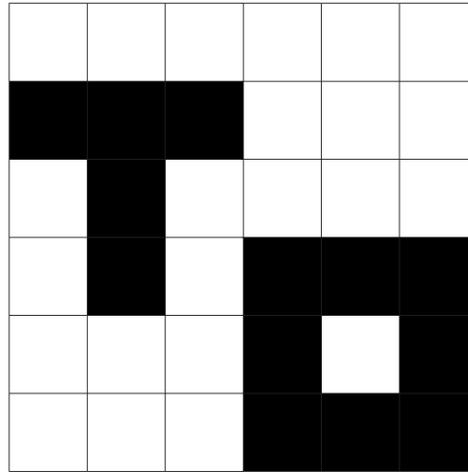


Figura 3.5: Regiones conectadas.

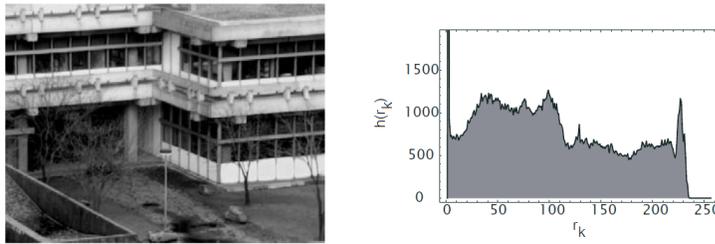


Figura 3.6: Una imagen y su histograma.

conectado con un hoyo.

El *histograma* de una imagen $f(x,y)$ con L niveles de intensidad o de gris en el rango $[0, L-1]$, denotado como $h(r_k)$ es una función discreta:

$$h(r_k) = \frac{n_k}{N} \quad (3.3)$$

donde r_k es el k -ésimo nivel de gris, n_k es el número de píxeles en la imagen con el nivel de intensidad r_k y N el número total de píxeles en la imagen. En la Fig. 3.6 se muestra una imagen y su histograma. La función $h(r_k)$ proporciona un estimado de la probabilidad de ocurrencia de un nivel de gris dado n_k de igual forma, la función $h(r_k)$ también habla de manera global acerca de la apariencia de la imagen. Por ejemplo, un histograma de imagen sesgado a la izquierda nos habla de una imagen con poco contraste y oscura, mientras que un histograma con valores sesgados hacia la derecha nos habla también de una imagen con poco contraste pero con brillo.

Fundamentos del color. Básicamente, los colores que los seres humanos y algunos otros animales

perciben en un objeto está determinada por la naturaleza de la luz reflejada sobre el objeto. La luz visible está compuesta por una estrecha banda de frecuencias en el espectro electromagnético. Un cuerpo que refleja la luz balanceada en todo el espacio visible de longitudes de onda aparece en color blanco para el observador. Si la luz es acromática entonces el color que se percibe va del blanco al negro pasando por niveles de grises. La luz cromática abarca desde los 400 a los 700 nm aproximadamente en el espectro electromagnético. Se emplean tres cantidades para describir la calidad de una fuente de luz cromática: radiancia, luminancia y brillo. La *radiancia* es la cantidad de energía que fluye desde la fuente de luz, y sus unidades de medida son en Watts (W). *Luminancia*, medida en lumens (lm), es una medida de la cantidad de energía que un observador percibe de la fuente de luz. El *brillo* es un descriptor subjetivo imposible de medir e incorpora la noción acromática de intensidad y es un factor clave en la descripción de la sensación de color. Las características que generalmente se emplean para distinguir un color de otro son el brillo, el *matiz* y la *saturación*. El matiz es un atributo asociado con la longitud de onda dominante en una mezcla de longitudes de onda. Representa el color dominante percibido por el observador. La saturación se refiere a la pureza relativa o la cantidad de luz blanca mezclada con un matiz. Al matiz y a la saturación juntas se les llama *cromaticidad*, y, por tanto, un color puede ser caracterizado por su brillo y su cromaticidad.

Modelos de color. El propósito de un modelo de color (también llamado espacio de color o sistema de colores) es facilitar la especificación de colores en alguna forma estándar de aceptación general. En esencia, un modelo de color es una especificación de un sistema coordinado y un subespacio dentro de ese sistema donde cada color se representa por un punto. Los modelos de color que existen son: RGB, NTSC, YCbCr, HSV, CMY, CMYK y HSI (ver [13, 16]). A continuación se describen los espacios de color RGB y YCbCr.

Una imagen de color RGB es una matriz de píxeles de tamaño $M \times N \times 3$, donde cada píxel es una combinación de colores rojo, verde y azul. Por convención, cada matriz se referencia como componente rojo, verde y azul de la imagen RGB. Una imagen RGB es de 8 bits (uint8) si su rango de valores es de [0,255], o de 16 bits (uint16) para [0,65535], o double para [0,1]. El número de bits empleados para representar el píxel de las componentes de la imagen determinan la profundidad de bit (*bit depth*). Por ejemplo, si cada una de las componentes de la imagen es de 8 bits, entonces la imagen RGB correspondiente se dice que es de 24 bits de fondo. El número total de colores en una imagen RGB de 24 bits es de $(2^8)^3 = 16,777,216$. El espacio de color RGB, generalmente se muestra como un cubo de colores como se muestra en la Fig. 3.7. Los vértices del cubo son los colores primarios (rojo, verde y azul) y secundarios (cyan, magenta y amarillo) de la luz.

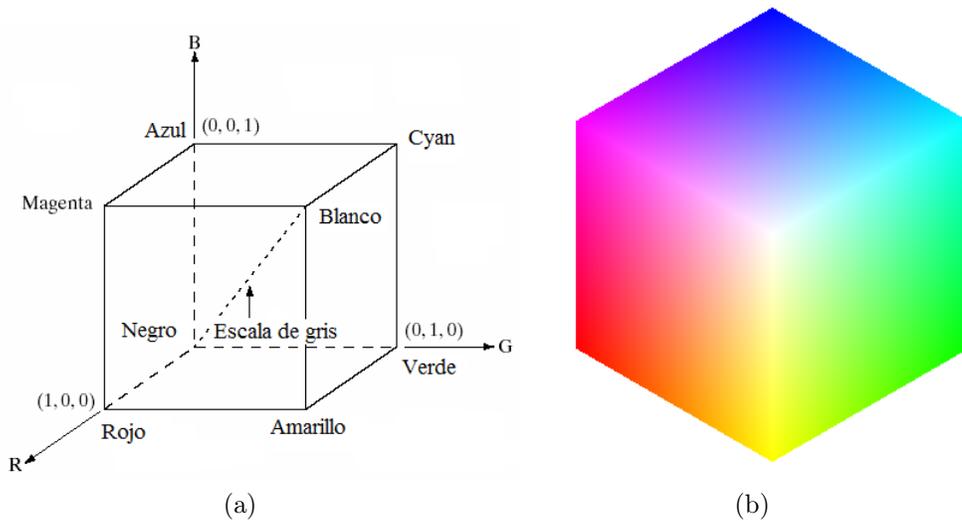


Figura 3.7: Cubo de 24 bits RGB. a) Esquema, normalizado a valores entre [0,1]; b) Cubo de colores.

El termino nivel de gris (*gray level*) se emplea para referirse a la intensidad de las imágenes monocromáticas.

El espacio de color YCbCr. En este espacio, la información de luminancia se representa por una componente única, Y, y la información de color se almacena como dos componentes diferencia de color, Cb y Cr. La componente Cb es la diferencia entre la componente azul y un valor de referencia, y la componente Cr es la diferencia entre la componete roja y un valor de referencia. La transformación empleada para convertir imágenes de RGB a YCbCr es através de la siguiente ecuación:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112.00 \\ 112.000 & -93.786 & -18.214 \end{bmatrix} + \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.4)$$

3.1.2. Reconocimiento de patrones

El reconocimiento de patrones [17], o reconocimiento de formas, se define como la disciplina científica cuya meta es la clasificación de objetos en un número de categorías o clases. Al proceso que permite extraer la región de interés de una imagen se le denomina *segmentación*.

3.1.2.1. Segmentación de un objeto basado en el umbralizado

La segmentación automática de imágenes es, en general, una de las tareas más difíciles de realizar en análisis de imágenes. Tiene una influencia determinante en el correcto desempeño de un sistema

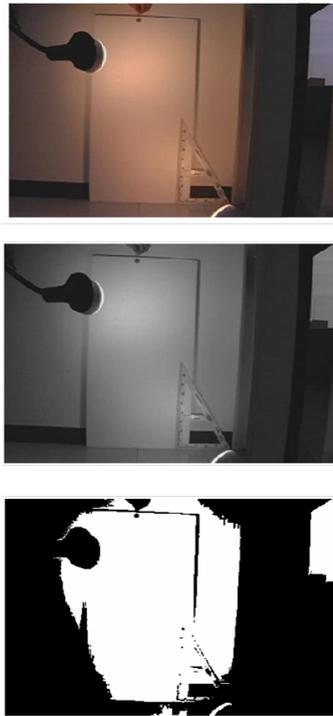


Figura 3.8: Imagen en color (superior) convertida a escala de grises (media) y binarizada (inferior) con umbralizado por el método de Otsu.

de reconocimiento automático de objetos (SRAO). En aplicaciones donde muchas de las condiciones pueden ser controladas, por ejemplo la iluminación, el que los objetos no se toquen o traslapen entre ellos facilita el proceso de segmentación.

La idea básica de las técnicas basadas en el umbralizado [14, Cap. 2], es convertir las imágenes $f(x,y)$ en niveles de gris en una imagen binaria $b(x,y)$, buscando que los objetos de interés de la imagen queden separadas del fondo.

El método de Otsu [18], es un método de umbralizado automático. Para su correcto funcionamiento el método de Otsu supone que los píxeles de la imagen $f(x,y)$ a binarizar pueden ser separados a través de un umbral u (a determinar) en dos clases: C_1 , la clase del objeto u objetos de interés y C_2 , la clase de los píxeles del fondo.

Para el análisis que sigue se hace uso de la siguiente notación:

- r el nivel de gris de un píxel con coordenadas (x,y) .
- L el número de niveles de gris de la imagen $f(x,y)$ con $0 \leq r \leq L - 1$.
- p_r es el número de píxeles con el nivel de gris r .

- $N_T = p_0 + p_1 + \dots + p_r + p_{L-1}$ es el número total de píxeles.
- $p(r) = \frac{p_r}{N_T}$ el histograma normalizado de niveles de gris de $f(x,y)$, tal que $\sum_{r=0}^{L-1} p(r) = 1.0$
- u el umbral a determinar. El objetivo consiste en encontrar el valor óptimo de u , u^* al maximizar o minimizar una función criterio.

El método de Otsu se basa en la técnica del análisis discriminante al maximizar alguna medida que permita separar clases: la de los objetos y las del fondo. Una de estas medidas es la siguiente:

$$J_1(u) = \frac{P_1(u)P_2(u)[\mu_1(u) - \mu_2(u)]^2}{P_1\sigma_1^2(u) + P_2(u)\sigma_2^2(u)} \quad (3.5)$$

donde:

$$P_1(u) = P_r(C_1) = \sum_{r=0}^u p(r)$$

$$P_2(u) = P_r(C_2) = \sum_{r=u+1}^{L-1} 1-p_1(u)$$

$$\mu_1(u) = \sum_{r=0}^u p(r | C_1) = \frac{1}{p_1(u)} \sum_{r=0}^u rp(r)$$

$$\mu_2(u) = \sum_{r=u+1}^{L-1} rP_r(r | C_2) = \frac{1}{p_2(u)} \sum_{r=u+1}^{L-1} rp(r)$$

$$\sigma^2_1(u) = \sum_{r=0}^{L-1} (r\mu_1(u))^2 P_r(r | C_1) = \frac{1}{p_1(u)} \sum_{r=0}^u (r - \mu_1(u))^2 p(r)$$

$$\sigma^2_2(u) = \sum_{r=0}^{L-1} (r\mu_2(u))^2 P_r(r | C_2) = \frac{1}{p_2(u)} \sum_{r=0}^u (r - \mu_2(u))^2 p(r)$$

para poder maximizar el criterio dado por la ecuación (3.5) las medidas de las dos clases deberían estar bastante bien separadas y las varianzas deberían ser lo más pequeñas posibles. Si esto no sucede, el valor del umbral obtenido simplemente no producirá el resultado deseado. Una imagen con un fondo muy grande comparado con el objeto u objetos en la imagen puede también dar lugar a valores de umbral que produzcan resultados indeseados.

El valor óptimo u^* puede encontrarse al buscar en el rango $[0, L-1]$ el valor de u que maximice (3.5). Esto es:

$$u^* = \arg \max_{0 \leq u \leq L-1} J_1(r) \quad (3.6)$$

En la Fig. 3.8 se muestra una imagen binarizada por el método de Otsu.

En el apéndice C.1, se ilustra como ejemplo la segmentación de un objeto en caída libre.

3.1.2.2. Etiquetado de regiones o componentes conectadas

Una vez que una imagen ha sido umbralizada a través de algunos de los métodos, sus componentes conectadas pueden ser etiquetadas. Los píxeles en una región conectada forman una región que puede representar un objeto dado a reconocer. El etiquetado de componentes conectadas (ECC) puede ser visto como una segunda etapa en el proceso de segmentado de una imagen. Se emplea un algoritmo recursivo que consiste de los cuatro pasos siguientes, dada una imagen binaria $b(x,y)$:

1. Barrer $b(x,y)$ hasta encontrar un píxel de tipo objeto (con valor 1 o L-1) aún no etiquetado y asignarle una nueva etiqueta E.
2. Asignar en forma recursiva la etiqueta E a todos sus vecinos, según la métrica elegida.
3. Detener si durante el recorrido no se encuentran más píxeles de tipo objeto.
4. Regresar al paso 1.

En la Fig. 3.9a se muestra una imagen binaria. En las Figs. 3.9b-d se muestra la secuencia de etiquetados hasta obtener la imagen etiquetada final. En la Fig. 3.9a las etiquetas son la unidad. En la Fig. 3.9b se distingue la segunda región con la etiqueta 2 y así sucesivamente para las regiones 3 y 4. Una de las ventajas de este algoritmo es que en una sola pasada se puede obtener el resultado final.

La técnica de umbralizado y la técnica de etiquetado de componentes conectadas pueden ser combinadas para separar las regiones de cada objeto en una imagen. Cada una de las regiones etiquetadas pueden ser utilizadas para entrenar clasificadores para el reconocimiento de objetos.

3.2. Análisis de Componentes Principales

La técnica Análisis de Componentes Principales (*PCA*, *Principal Components Analysis*) [19, 20] es una técnica matemática útil que ha encontrado aplicación en los campos del reconocimiento de rostros y en la compresión de imágenes ya que es una técnica común para encontrar patrones o rasgos principales finitos en problemas de dimensión infinita.

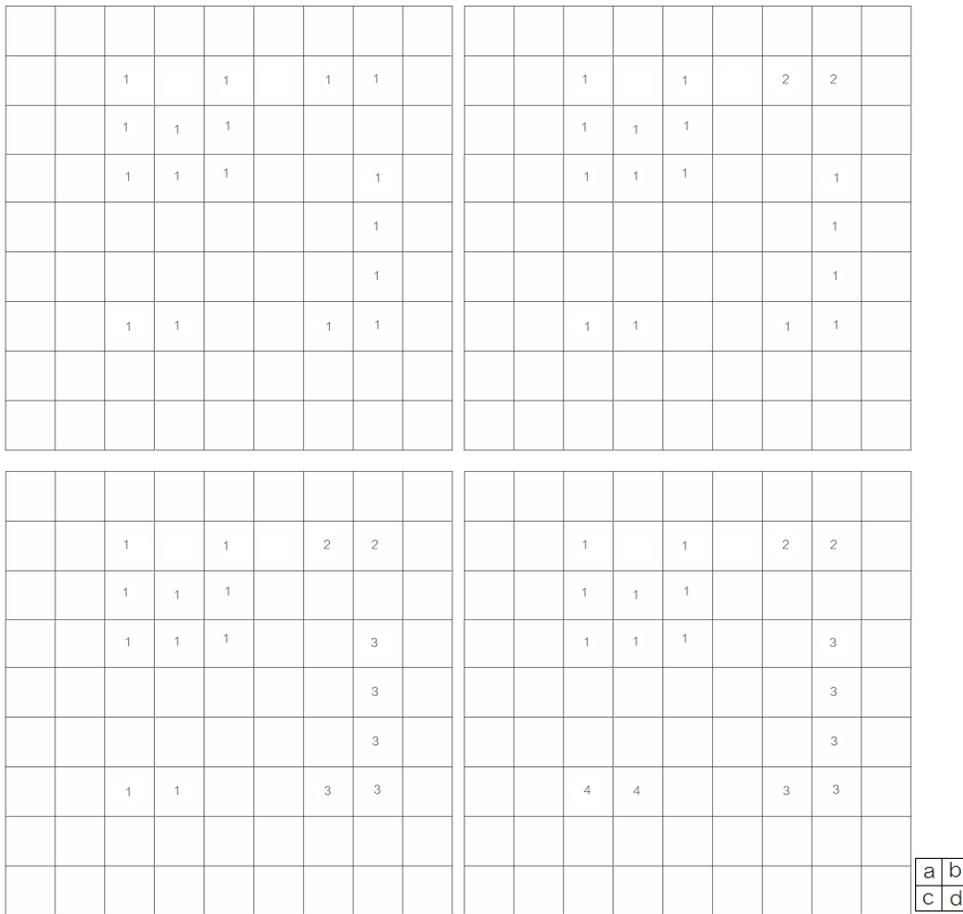


Figura 3.9: (a) Imagen binaria; (c-d) Secuencia de etiquetados.

La técnica consiste de los siguientes pasos: (1) Recopilar el conjunto de datos; (2) Obtener el conjunto promedio de los datos; (3) Calcular la matriz de covarianza; (4) Seleccionar los componentes, formar un vector característico y calcular los eigenvectores y eigenvalores de la matriz de covarianza.

3.2.1. Conjunto de datos

Sea n un conjunto de imágenes de tamaño $M \times N$, donde M es el número de píxeles por renglón y N es el número de píxeles por columna. Entonces, un vector:

$$\mathbf{x} = x_1, x_2, \dots, x_n \tag{3.7}$$

Se compone de n píxeles, donde cada elemento es un píxel localizado en la misma posición de cada imagen (Fig. 3.10). Esto implica que se puede tener una matriz \mathbf{X} de vectores \mathbf{x}_k tales que:

$$\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k \tag{3.8}$$

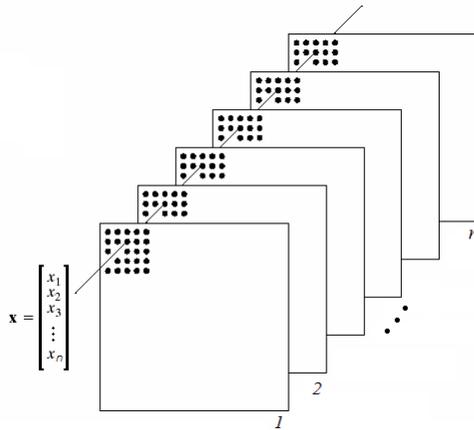


Figura 3.10: Vector \mathbf{x} formado con píxeles de n imágenes.

donde $k=1,2,\dots,MN$ y

$$\mathbf{x}_k = x_1, x_2, \dots, x_n \quad (3.9)$$

Esta matriz, \mathbf{X} , de vectores representa al conjunto de datos.

3.2.2. Vector promedio

El vector promedio del conjunto de datos estará dado por la relación:

$$\mathbf{m}_\mathbf{X} = \frac{1}{MN} \sum_{k=1}^{MN} \mathbf{x}_k \quad (3.10)$$

3.2.3. Matriz de covarianza

La matriz de covarianza $\mathbf{C}_\mathbf{X}$ es una medida de la dispersión entre diferentes variables (más de una dimensión). Para el caso de un vector \mathbf{x}_k y el vector promedio $\mathbf{m}_\mathbf{X}$ tenemos que la matriz de covarianza está dada por:

$$\mathbf{C}_\mathbf{X} = \frac{1}{MN} \sum_{k=1}^{MN} (\mathbf{x}_k - \mathbf{m}_\mathbf{X})(\mathbf{x}_k - \mathbf{m}_\mathbf{X})^T \quad (3.11)$$

Sean \mathbf{e}_i y λ_i , $i=1, 2, \dots, n$, los eigenvectores y eigenvalores correspondientes de la matriz $\mathbf{C}_\mathbf{X}$, ordenados por conveniencia en orden descendente, de modo que $\lambda_j = \lambda_{j+1}$, para $j = 1, 2, \dots, n - 1$. Por definición, los eigenvectores y eigenvalores de una matriz de dimensión $n \times n$ satisfacen la relación:

$$\mathbf{C}_\mathbf{X} \mathbf{e}_i = \lambda_i \mathbf{e}_i \quad (3.12)$$

para $i = 1, 2, \dots, n$.

3.2.4. Componentes principales

La transformación en componentes principales [13] está dado por:

$$\mathbf{Y} = \mathbf{A}(\mathbf{x}_k - \mathbf{m}_\mathbf{X}) \quad (3.13)$$

donde \mathbf{A} representa el vector característico o vector propio (*eigenvector*, del alemán *eigen* propio). Los renglones de \mathbf{A} representan los eigenvectores de $\mathbf{C}_\mathbf{X}$. Los eigenvalores, son los elementos a lo largo de la diagonal de $\mathbf{C}_\mathbf{Y}$, donde:

$$\mathbf{C}_\mathbf{Y} = \mathbf{A}\mathbf{C}_\mathbf{X}\mathbf{A}^T \quad (3.14)$$

Los eigenvectores [21], son vectores unitarios, es decir su longitud es 1 y forman un conjunto ortonormal de vectores en la norma inducida por la matriz de covarianza (son perpendiculares entre ellos respecto a $\mathbf{C}_\mathbf{X}$) y aportan información acerca de los patrones en los datos. Seleccionando los eigenvalores de mayor valor se consigue la reducción de dimensionalidad y la compresión de datos, en el sentido de que con estos vectores se cubre el espacio de los datos. Hay reducción de dimensionalidad en la medida en que la dispersión de los datos se alinean a un eigenvector en particular de manera que se pueden omitir aquellos eigenvectores en los cuales no sea relevante tal alineación, es decir, su eigenvalor sea pequeño. Hay compresión de datos en la medida en que se pueden seleccionar sólo aquellos eigenvalores, que sean más representativos en peso (de mayor valor) de manera que es posible recuperar la información original sin gran pérdida de datos, o sea, con error cuadrático medio pequeño. Esto es, sea:

$$\mathbf{X} = \mathbf{A}^T \mathbf{Y} + \mathbf{m}_\mathbf{X} \quad (3.15)$$

el conjunto de datos original. Entonces, para un conjunto q reducido de eigenvectores tenemos que \mathbf{A} se hace una matriz de tamaño $q \times n$, \mathbf{A}_q , de manera que la reconstrucción estará dada por:

$$\mathbf{X} = \mathbf{A}_q^T \mathbf{Y} + \mathbf{m}_\mathbf{X} \quad (3.16)$$

El error cuadrático medio e_{cm} estará dado por:

$$e_{cm} = \sum_{j=1}^n \lambda_j - \sum_{j=1}^q \lambda_j = \sum_{j=q+1}^n \lambda_j \quad (3.17)$$

Si $q=n$ el error es cero.

3.3. La Técnica *Eigenface*

La técnica eigenface [7, 22], es el PCA aplicado en imágenes faciales (rostros) sobre un espacio representado por un conjunto de rostros grande. Los característicos significativos son conocidos como *eigenfaces* porque son los eigenvectores (componentes principales) del conjunto de rostros; que asemejan, aunque no necesariamente, a características tales como los ojos, oídos o nariz. La operación de proyección caracteriza a un rostro individual por la suma de pesos de los eigenfaces característicos, de manera que para reconocer un rostro particular solo es necesario comparar estos pesos contra aquellos de los rostros conocidos. En términos matemáticos se requiere encontrar los componentes principales de la distribución de rostros, o los eigenvectores de la matriz de covarianza del conjunto de imágenes faciales, tratando una imagen como un punto (vector) en un espacio dimensional infinito. Posteriormente, se ordenan los eigenvectores de acuerdo a una cantidad relativa a la variación entre las imágenes faciales, seleccionando un número finito de estos, los más representativos en peso de acuerdo a su eigenvalor y que contribuyen a recuperar la imagen.

Podemos ver, a cada uno de los eigenvectores, como un pequeño espectro o "fantasma", del rostro original, al cual se le llama *eigenface*. Cada rostro individual se puede representar en términos de una combinación lineal de los eigenfaces. Por lo que, un rostro individual se puede aproximar empleando solo los "mejores" eigenfaces, aquellos que tienen los eigenvalores más largos o pesados, y que más contribuyen en la variación del conjunto de imágenes faciales. Los mejores M eigenfaces se extienden sobre un subespacio M -dimensional, *espacio facial*, de todas las imágenes posibles (se tiene la reducción a M dimensiones del espacio de rostros de dimensión infinita).

3.3.1. Resolución de las imágenes

Sea \mathbf{F} un conjunto de M imágenes o fotografías de tamaño $m \times n$, donde:

$$\mathbf{F} = \mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_M \quad (3.18)$$

Las operaciones de cálculo que se proponen en la sección siguiente, se reducen si cada imagen se reordena de manera que el tamaño o resolución de las mismas se lleven a la resolución $N \times 1$ (todas las columnas de la imagen se reagrupan en una sola) donde $N = m \times n$. De esta forma se obtendrá:

$$\mathbf{\Gamma} = \tau_1, \tau_2, \dots, \tau_M \quad (3.19)$$

que es un conjunto de M imágenes de resolución $N \times 1$.

3.3.2. Cálculo de eigenfaces

El rostro promedio, Ψ , del conjunto de fotografías estará dado por:

$$\Psi = \frac{1}{M} \sum_{k=1}^M \Gamma_k \quad (3.20)$$

Sea:

$$\Phi_i = \Gamma_i - \Psi$$

la diferencia entre la i -ésima imagen o fotografía y el rostro promedio. Entonces la matriz de covarianza estará dada por la relación:

$$\mathbf{C} = \frac{1}{M} \sum_{k=1}^M \Phi_k \Phi_k^T = \mathbf{A} \mathbf{A}^T \quad (3.21)$$

donde

$$\mathbf{A} = [\Phi_1 \Phi_2 \dots \Phi_M]$$

es una matriz de tamaño $N \times M$. A partir de la matriz de covarianza se pueden obtener los eigenfaces (eigenvectores) y los eigenvalores. Aplicando:

$$\mathbf{A} \mathbf{A}^T \mathbf{v}_i = \mu_i \mathbf{v}_i$$

donde \mathbf{v}_i es el vector de eigenfaces y μ_i el conjunto de eigenvalores. Notemos que la resolución de esta operación estará dada por:

$$\mathbf{A} \mathbf{A}^T \Rightarrow (N \times M)(N \times M)^T = (N \times N)$$

Para hacer este cálculo es necesario hacer computacionales las operaciones reduciendo la resolución de las mismas. Aplicando propiedades del álgebra lineal se puede reducir el cómputo de las operaciones a un valor dado por $M \times M$. Esto es:

$$\mathbf{A}^T \mathbf{A} \Rightarrow (N \times M)^T (N \times M) = (M \times M)$$

De esta forma el procedimiento para calcular los eigenvectores y eigenvalores será el siguiente. Consideremos la ecuación:

$$\mathbf{A}^T \mathbf{A} \mathbf{u}_i = \mu_i \mathbf{u}_i$$

De esta ecuación obtener los eigenvectores \mathbf{u}_i y eigenvalores μ_i . A continuación ajustar el eigenvector \mathbf{u}_i obtenido multiplicándolo por \mathbf{A} ya que:

$$\mathbf{A} \mathbf{A}^T (\mathbf{A} \mathbf{u}_i) = \mu_i (\mathbf{A} \mathbf{u}_i) \quad (3.22)$$

donde

$$\mathbf{v}_i = \mathbf{A}\mathbf{u}_i \quad (3.23)$$

nos da el conjunto de eigenfaces buscado.

En el apéndice C.2, se ilustra como ejemplo el cálculo de la imagen promedio así como de los eigenfaces correspondientes para un conjunto de doce fotografías.

3.3.3. Cómo utilizar los eigenfaces para reconocimiento de rostros

Los eigenfaces son adecuados para describir imágenes faciales bajo condiciones controladas. Un valor pequeño M' ($M' < M$ donde M es el número de fotografías y $M' \simeq 0.4M$) es suficiente para realizar la identificación.

Un nuevo rostro ($\mathbf{\Gamma}$) se transforma en sus componentes eigenface (proyectado sobre el “espacio facial”) por la operación:

$$\omega_k = \mathbf{u}_k^T (\mathbf{\Gamma} - \mathbf{\Psi}) \quad (3.24)$$

para $k = 1, \dots, M'$. Estos componentes, pesos, forman un vector

$$\mathbf{\Omega}^T = [\omega_1, \omega_2, \dots, \omega_{M'}] \quad (3.25)$$

que describen la contribución de cada eigenface en la representación del nuevo rostro de entrada. Este vector se puede emplear en un algoritmo de reconocimiento de patrones para determinar a que número de clase de rostro predefinido pertenece, si existe alguno. El método, simple, para determinar este número de clase es encontrar la clase k que minimiza la distancia euclidiana:

$$\epsilon_k^2 = \|(\mathbf{\Omega} - \mathbf{\Omega}_k)\|^2 \quad (3.26)$$

donde $\mathbf{\Omega}_k$ es un vector que describe la k -ésima clase facial. Diferentes fotografías de un persona se clasifican en una clase $\mathbf{\Omega}_i$ promediando los resultados de su representación en eigenfaces. Un rostro se clasifica como perteneciente a la clase k cuando el mínimo ϵ_k está por debajo de un umbral seleccionado θ_ϵ . De otra forma el rostro se clasifica como “desconocido” y opcionalmente se puede emplear para crear una nueva clase facial.

Crear el vector de pesos es equivalente a proyectar la imagen facial original sobre el espacio facial de baja dimensión, las imágenes (muchas de ellas sin parecido a un rostro) se proyectarán sobre un vector patrón. La distancia ϵ entre la imagen y el espacio facial estará dada por

$$\epsilon^2 = \|\Phi - \Phi_f\| \quad (3.27)$$

donde

$$\Phi = \Gamma - \Psi$$

y

$$\Phi_f = \sum_{i=1}^{M'} \omega_i u_i$$

De lo anterior hay cuatro posibilidades para una imagen de entrada y su vector patrón: (1) espacio facial cercano y clase facial cercana, (2) espacio facial cercano pero ninguna clase facial cercana conocida, (3) lejano del espacio facial y clase facial cercana, y (4) lejano del espacio facial y ninguna clase facial cercana conocida. En el primer caso, un individuo es reconocido e identificado. En el segundo caso, un individuo desconocido está presente. Los dos últimos casos indican que la imagen no es un rostro. El caso tres muestra un caso típico de falso positivo que es detectado dada la distancia significativa entre la imagen de entrada y el espacio facial. Es decir, un falso positivo corresponde a una imagen detectada como si fuera un rostro o siendo un rostro como si formara parte de la base de datos. Un falso negativo corresponde a un rostro que no es detectado y que sí forma parte de la base de datos.

3.4. Normalización

La normalización [1, pag. 93][23], es un proceso que permite mejorar el reconocimiento y consiste en alinear tanto como sea posible las imágenes faciales. La alineación puede ser en traslación, orientación y escala.

Para eliminar la influencia del cabello y del ruido de fondo se puede extraer solo la región elíptica de la cara. Para reducir los efectos de los cambios en la iluminación global y direccional, se pueden aplicar sombreado lineal y ecualización a las imágenes faciales. Sin embargo, estas operaciones pueden ser muy costosas, especialmente en escenas dinámicas. Una alternativa, consiste en aplicar una corrección en intensidad a cada píxel aplicando la relación:

$$CI(x_n) = \frac{x_n - \mu}{\sigma} \quad (3.28)$$

donde CI representa la corrección en intensidad, x_n representa al píxel, μ representa la intensidad media y σ representa la desviación estándar de la imagen.

3.5. Variación Total

Variación total (TV, *Total Variation*) es una técnica matemática que permite suavizar una imagen (quitar ruido). En esta sección se describe su formulación, sin restricción y con restricción a ruido, así como su discretización.

3.5.1. Conceptos preliminares

Cantidades variables llamadas *funcionales* [24], juegan un papel importante en diferentes problemas que se presentan en el análisis, mecánica, geometría, etc. Un *funcional* es una correspondencia que asigna un número (real) definido a cada función (o curva) que pertenece a alguna clase. Por tanto, se puede decir que un funcional es una clase de función donde la variable es en sí misma una función (o curva). La rama más desarrollada del “cálculo de funciones” está interesada en la búsqueda de máximos y mínimos de funcionales y se le conoce con el nombre de *cálculo de variaciones*.

Sea $J[y]$ un funcional de la forma

$$J[y] = \int_a^b F(x, y, y') dx \quad (3.29)$$

definida en el conjunto de funciones $y(x)$ la cual tiene primeras derivadas continuas en $[a, b]$ y satisface las condiciones de frontera $y(a)=A$, $y(b)=B$. Entonces una condición necesaria para que $J[y]$ tenga un extremo para una función dada $y(x)$ es que satisfaga la ecuación de Euler:

$$F_y - \frac{d}{dx} F_{y'} = 0 \quad (3.30)$$

Las curvas de la ecuación de Euler se conocen como extremos. La ecuación de Euler es una ecuación diferencial de segundo orden, su solución dependerá, en general, de las condiciones de frontera.

El problema variacional más simple consiste en encontrar un extremo débil de un funcional de la forma (3.29), donde la clase de funciones admisibles consisten en todas las curvas suaves que unen dos puntos.

El *método de diferencias finitas* es de amplio uso para resolver problemas variacionales. Considerando un funcional de la forma (3.29) para encontrar una función relacionada de la forma más corta que considera el análisis clásico procedemos como sigue. Emplear los puntos:

$$a = x_0, x_1, \dots, x_n, x_{n+1} = b$$

se divide el intervalo $[a,b]$ en $n+1$ partes iguales. Entonces se reemplaza la curva $y=y(x)$ por la línea poligonal con vértices

$$(x_0, A), (x_1, y(x_1)), \dots, (x_n, y(x_n)), (x_{n+1}, B),$$

y se aproxima la funcional $J[y]$ por la suma:

$$J(y_1, \dots, y_n) = \sum_{i=1}^{n+1} F(x_i, y_i, \frac{y_i - y_{i-1}}{h})h, \quad (3.31)$$

donde

$$y_i = y(x_i), \quad h = x_i - x_{i-1}$$

$$y_0 = A \quad y_{n+1} = B$$

cada línea poligonal está determinada por la ordenación y_1, \dots, y_n de sus vértices y la suma (3.31) es por tanto, una función de n variables y_1, \dots, y_n .

3.5.2. Restauración de la imagen

Consideremos el problema de restaurar una imagen contaminada con ruido o degradada. Sea $u_0 = Ku_{true} + \eta$ la imagen adquirida donde K es un operador de nivel borroso y η es el nivel de ruido o degradación de la imagen. Encontrar una aproximación $u(\vec{x})$ a la imagen verdadera $u_{true}(\vec{x})$. TV es un método [25] para recuperar la imagen verdadera, lo más exacta posible y encontrar una nueva imagen en la cual la información de interés sea más obvia y más fácil de extraer.

Se puede modificar una imagen disminuyendo la variación total

$$TV(u) \equiv \int |\nabla u(\vec{x})| d\vec{x} \quad (3.32)$$

en la imagen, mientras se preserva algún ajuste a los datos originales u_0 . A la ecuación (3.32) se le conoce como variación total de u . Al resolver la ecuación siguiente, la nitidez o suavizado de la imagen (Fig. 3.11) lo determinan u_0 y α . Esta ecuación se conoce como la formulación no restringida de Tikhonov.

$$\min_u \frac{1}{2} \|u - u_0\|^2 + \alpha TV(u) \quad (3.33)$$

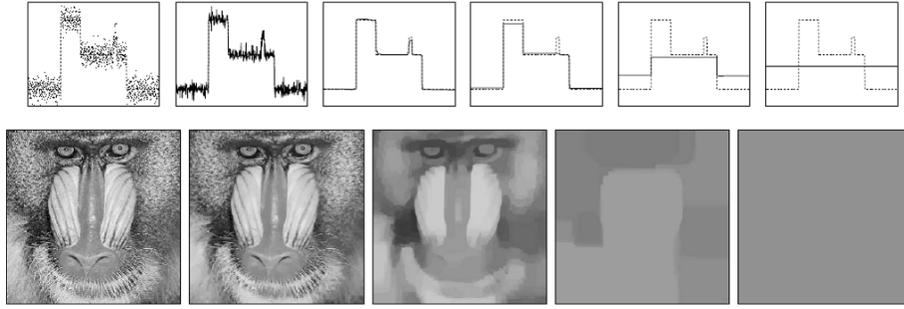


Figura 3.11: Resultado de la regularización por TV para una función con ruido (imagen superior) y para una imagen con un mandril (imagen inferior). Las imágenes de la izquierda son las originales, las siguientes muestran los resultados de la regularización para los valores de $\alpha=0.0001$, 0.001 , 0.01 , 0.1 y 1.0 .

La formulación restringida a ruido está dada por:

$$\min_u TV(u) \text{ sujeta a } \|u - u_0\|^2 = \sigma^2 \quad (3.34)$$

Computacionalmente este modelo generalmente se resuelve a través de su ecuación de Euler-Lagrange [26, pag. 179]. Adoptando el método de pendiente descendente (*steepest descent marching*) con tiempo t artificial, propuesto por Rudin, Osher and Fatemi [27], tenemos que para:

$$E_{tv}[u | u_0] = \int_{\Omega} |\nabla u| dx + \frac{\lambda}{2} \int_{\Omega} (u_0(x) - u(x))^2 dx \quad (3.35)$$

La ecuación de Euler-Lagrange estará dada por

$$u_t(x, t) = -\frac{\partial E_{tv}}{\partial u} = \nabla \cdot \left[\frac{\nabla u}{|\nabla u|} \right] - \frac{\lambda}{2} (u(x, t) - u_0(x)) \quad (3.36)$$

Para procesamiento de imágenes, estos mecanismos de difusión son necesarios y una ventaja en técnicas de suavizado.

3.5.3. Discretización de la variación total de una imagen

Consideremos una imagen u definida en $\{0, \dots, N-1\}^2$, donde N es un entero positivo, extendida en \mathbb{Z}^2 . La forma clásica de discretizar (3.32) es de la siguiente forma [28, 30, 31, 32]:

$$TV_d(u) = \sum_{0 \leq k, l < N} |\nabla u|_d(k, l) \quad (3.37)$$

donde

$$|\nabla u|_d(k, l) = \sqrt{(u(k+1, l) - u(k, l))^2 + (u(k, l+1) - u(k, l))^2} \quad (3.38)$$

Hay otras formas de discretizar la variación total, por medio de diferencias finitas o con valores absolutos, sin embargo la ecuación (3.37) es la forma más simple y tiene la ventaja de ser euclidiana.

Para los casos de las ecuaciones (3.33) y (3.34) la discretización se realiza como sigue[27]. Sea $u_0(x, y)$ una función de intensidad que representa los píxeles de una imagen con ruido donde $x, y \in \Omega$. Sea $u(x, y)$ la imagen libre de ruido, entonces:

$$u_0(x, y) = u(x, y) + n(x, y) \quad (3.39)$$

donde n es el ruido aditivo. Se quiere reconstruir u desde u_0 . El problema de minimización restringida es:

$$\min \int_{\Omega} \sqrt{u_x^2 + u_y^2} dx dy \quad (3.40)$$

sujeta a

$$\int_{\Omega} u dx dy = \int_{\Omega} u_0 dx dy \quad (3.41)$$

La restricción significa que el ruido $n(x, y)$ en (3.39) es de promedio cero y

$$\int_{\Omega} \frac{1}{2} (u - u_0)^2 dx dy = \sigma^2 \quad (3.42)$$

donde $\sigma > 0$. La segunda restricción considera que la desviación estándar del ruido $n(x, y)$ es σ .

Así se tiene una restricción lineal y otra no lineal. Sus ecuaciones de Euler-Lagrange son:

$$0 = \frac{\partial}{\partial x} \left(\frac{u_x}{\sqrt{u_x^2 + u_y^2}} \right) + \frac{\partial}{\partial y} \left(\frac{u_y}{\sqrt{u_x^2 + u_y^2}} \right) - \lambda_1 - \lambda_2 (u - u_0) \quad (3.43)$$

en Ω , con $\frac{\partial u}{\partial n} = 0$ sobre la frontera de $\Omega = \partial\Omega$.

El procedimiento de solución emplea una ecuación parabólica con el tiempo como un parámetro en evolución, o de manera equivalente, se aplica el método de gradiente descendente. Esto significa que resolvemos:

$$u_t = \frac{\partial}{\partial x} \left(\frac{u_x}{\sqrt{u_x^2 + u_y^2}} \right) + \frac{\partial}{\partial y} \left(\frac{u_y}{\sqrt{u_x^2 + u_y^2}} \right) - \lambda(u - u_0) \quad (3.44)$$

para $t > 0$, $x, y \in \Omega$, $u(x, y, 0)$ dado, $\frac{\partial u}{\partial n} = 0$ sobre $\partial\Omega$.

Se ha suprimido la primera restricción porque es automáticamente forzada por el procedimiento en evolución si la media de $u(x, y, 0)$ es la misma de $u_0(x, y)$. Cuando t se incrementa esto se aprovecha para quitar ruido a la imagen. Se debe calcular $\lambda(t)$. Al multiplicar la ecuación (3.44) por $(u - u_0)$ e integrar por partes sobre Ω , una vez que se alcanza el estado permanente, el lado izquierdo de la ecuación (3.44) se desvanece, entonces tenemos que:

$$\lambda = -\frac{1}{2\sigma^2} \int \left[\sqrt{u_x^2 + u_y^2} - \left(\frac{(u_0)_x u_x}{\sqrt{u_x^2 + u_y^2}} + \frac{(u_0)_y u_y}{\sqrt{u_x^2 + u_y^2}} \right) \right] dx dy \quad (3.45)$$

El método numérico en dos dimensiones está dado como sigue, sea:

$$x_i = ih, y_j = jh, i, j = 0, 1, \dots, N, \text{ con } Nh = 1,$$

$$t_n = n\Delta t, n = 0, 1, \dots,$$

$$u_{ij}^n = u(x_i, y_j, t_n),$$

$$u_{ij}^0 = u_0(ih, jh) + \sigma\varphi(ih, jh)$$

Los datos iniciales se eligen de manera que las restricciones se satisfagan inicialmente, es decir, que promedien cero.

$$\begin{aligned} u_{ij}^{n+1} = & u_{ij}^n + \frac{\Delta t}{h} (\Delta_x^- \left(\frac{\Delta_x^+ u_{ij}^n}{((\Delta_x^+ u_{ij}^n)^2 + (m(\Delta_x^+ u_{ij}^n, \Delta_y^- u_{ij}^n))^2)^{1/2}} \right) \\ & + \Delta_y^- \left(\frac{\Delta_y^+ u_{ij}^n}{((\Delta_x^+ u_{ij}^n)^2 + (m(\Delta_x^+ u_{ij}^n, \Delta_x^- u_{ij}^n))^2)^{1/2}} \right)) \\ & - \Delta t \lambda^n (u_{ij}^n - u_0(ih, jh)) \end{aligned}$$

para $i, j = 1, \dots, N$, con condiciones de frontera:

$$u_{0j}^n = u_{1j}^n, u_{Nj}^n = u_{N-1,j}^n, u_{i0}^n = u_{iN}^n = u_{i,N-1}^n$$

donde

$$\Delta_x^\mp u_{ij} = \mp(u_{i\mp 1,j} - u_{ij})$$

$$\Delta_y^\mp u_{ij} = \mp(u_{i,\mp 1,j} - u_{ij})$$

$$m(a, b) = \min_{\text{mod}}(a, b) = \left(\frac{\text{sgn } a + \text{sgn } b}{2} \right) \min(|a|, |b|)$$

Así también para λ^n tenemos:

$$\lambda^n = -\frac{h}{2\sigma^2} \left[\sum_{i,j} \left(\sqrt{(\Delta_+^x u_{ij}^n)^2 + (\Delta_+^y u_{ij}^n)^2} - \frac{(\Delta_+^x u_{ij}^0)(\Delta_+^x u_{ij}^n)}{\sqrt{((\Delta_+^x u_{ij}^n)^2 + (\Delta_+^y u_{ij}^n)^2)}} - \frac{(\Delta_+^y u_{ij}^0)(\Delta_+^y u_{ij}^n)}{\sqrt{((\Delta_+^x u_{ij}^n)^2 + (\Delta_+^y u_{ij}^n)^2)}} \right) \right]$$

Para estabilidad se impone una restricción para el tamaño del paso, dado por:

$$\frac{\Delta t}{h^2} \leq c$$

3.6. Procesamiento Multiresolución (Pirámide de Imágenes)

Una herramienta importante, conceptualmente simple en estructura para representar imágenes a más de una resolución es la pirámide de imágenes [13, pag. 351]. Una pirámide de imágenes es un conjunto de imágenes con resolución decreciente dispuestas en forma de una pirámide. Como se puede observar en la Fig. 3.12a la base de la pirámide tiene una representación de resolución alta de la imagen que se quiere procesar (Fig. 3.12b); la cima contiene una resolución baja. Al subir desde la base de la pirámide tanto el tamaño como la resolución decrecen. Como la base de la pirámide es de tamaño $N \times N$ entonces esta en el nivel J , dada por:

$$J = \log_2 N \tag{3.46}$$

y el siguiente nivel, $(N/2) \times (N/2)$, en $J-1=(\log_2 N)-1$. Por ejemplo, sea $N=512$ entonces el nivel J estará dado por $J = \log_2(512)=9$. Por tanto, $J-1=9-1=8$ donde $N/2=512/2=256=2^{J-1} = 2^8$. Esto quiere decir, que la imagen con resolución de 512×512 píxeles se podrá reducir desde el nivel 9 al nivel 1 . En la Fig. 3.12b se muestra una pirámide Gaussiana con filtrado en el dominio del espacio, empleando un kernel de convolución Gaussiano de 5×5 pasa bajas (Fig. 3.13).

$$h(x) = \sqrt{2\pi}\sigma A e^{-2\pi^2\sigma^2 x^2}$$

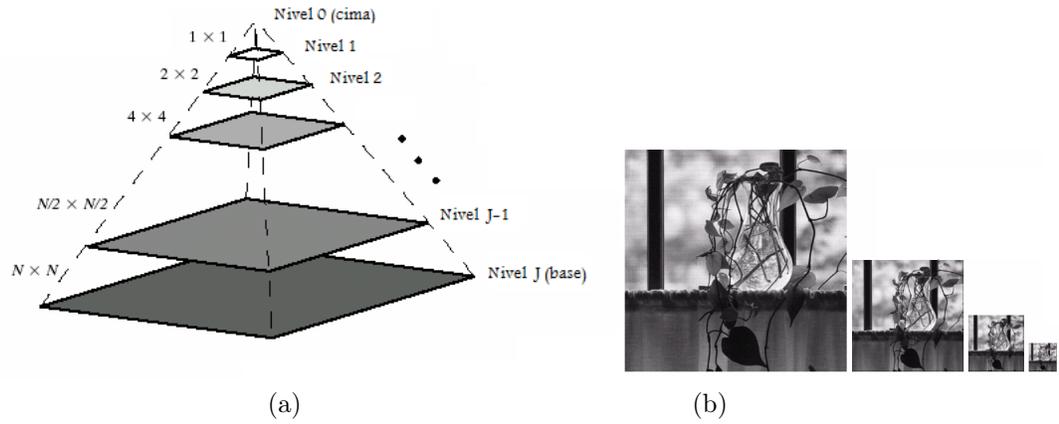


Figura 3.12: Estructura de imagen piramidal.

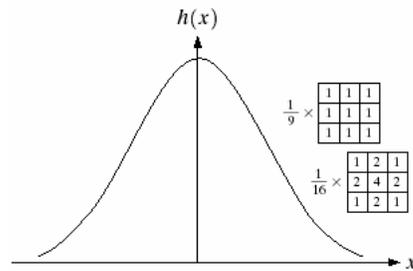


Figura 3.13: Filtro Gaussiano pasa bajas en el dominio del espacio.

Capítulo 4

PROTOCOLO DE ADQUISICIÓN DE IMÁGENES

El protocolo de adquisición de imágenes se compone de dos partes. En la primera parte se crea el set fotográfico en el cual se hará la adquisición de las imágenes que integrarán la base de datos del sistema de reconocimiento de rostros. En este set fotográfico se busca mantener las mejores condiciones de iluminación, el mismo fondo y distancia similar entre el rostro y la cámara fotográfica. De manera complementaria, se obtienen diez fotografías con cambio de escala, fondo e iluminación diferentes, que se integrarán a la base de datos con objeto de que permitan el reconocimiento facial con cambios de escala. En la segunda parte, se hace la adquisición de las imágenes que se utilizarán como imágenes de prueba con cambios de escala, en un set alternativo (salón de clase) donde cambian las condiciones de iluminación y escala de los rostros.

4.1. Proceso de adquisición de imágenes

El proceso de adquisición consiste en: (1) Construcción del set fotográfico; (2) Promover la colaboración de la comunidad académica en el proyecto; (3) Registrar al participante; y finalmente (4) Hacer la adquisición del rostro de frente en fotografías a color y con alta resolución (con fondo azul para crear la base de datos).

Para poder hacer la adquisición de las imágenes para crear la base de datos es necesario configurar el set fotográfico que se ilustra en la Fig. 4.1. El set fotográfico consta de la lista de materiales que se indica en la Tabla 4.1.

El diagrama esquemático de la Fig. 4.2 ilustra la disposición de los materiales en el set fotográfico.

4. PROTOCOLO DE ADQUISICIÓN DE IMÁGENES

Tabla 4.1: Lista de materiales.

1.	Lámparas reguladoras 500 w (2 unidades)
2.	Paneles reflectores (2 unidades)
3.	Banquillo (1 unidad)
4.	Fondo azul (1 unidad)
5.	Cámara profesional Nikon d80 (1 unidad)
6.	Tripie profesional (1 unidad)



Figura 4.1: Set fotográfico.

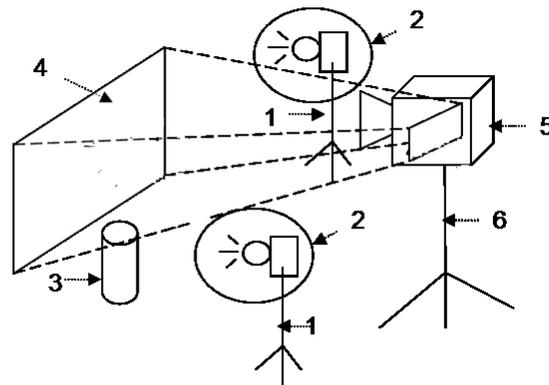


Figura 4.2: Diagrama esquemático

Dos lámparas reguladoras (1) iluminan a la persona a través de su panel reflector (2). El panel reflector evita que la intensidad de la luz moleste la retina de los ojos. El banquillo (3) estabiliza la posición de la persona. El fondo azul (4) facilitará el proceso de segmentación. La cámara profesional (5) permite capturas de imágenes en alta resolución 10.2 Mpíxeles (3872x2592 resolución máxima y 1936x1296 resolución mínima). La alta resolución permitirá cambiar la escala de las imágenes.



Figura 4.3: Registro de datos por persona.

La colaboración fue favorable no solo de la comunidad academica sino de las autoridades, quienes nos dieron la autorización así como todas las facilidades para el desarrollo de este protocolo de adquisición. Para este proceso colaboraron más de 50 personas entre alumnos y personal de la Universidad Autónoma Metropolitana, Unidad Cuajimalpa.

Durante el proceso de adquisición, cada persona registró sus datos personales (Fig. 4.3), en particular: nombre, edad, fecha y lugar de nacimiento, color de pelo, entre otros. Además, autorizó el uso de su imagen con propósitos de investigación como se ilustra en la hoja de registro del apéndice B.

Las fotografías en color son imágenes monoculares de frente, con fondo azul. El proyecto se acota a este tipo de imágenes y no a imágenes de perfil debido a la complejidad de estas últimas. El fondo azul permite mejorar la segmentación de las imágenes para la creación de la base de datos del sistema. La alta resolución es una característica que facilita los cambios de escala (reducción) de las imágenes sin gran pérdida de datos.

4.2. Resultados del proceso de adquisición

Como resultado del proceso de adquisición, en una primera etapa se obtuvieron las imágenes que se ilustran en la Fig. 4.4. Sin embargo, no fueron suficientes y se requirió una segunda etapa de adquisición de la cual se obtuvieron 488 fotografías (Figs. 4.7 y 4.5). Se integrarán 400 imágenes en la base de datos del sistema. Se utilizarán 48 fotografías como imágenes de prueba con variación en escala (Fig. 4.6).

Tabla 4.2: Adquisición de imágenes.

Base de Datos	Imágenes		Resolución (p)	B	Tipo
	Propósito	Total			
UAM-C	BD	440	W 323-241 H 231-175	24	JPEG → BMP
	P	48	W 3872 H 2592		JPEG → BMP
P Imágenes de prueba. BD Imágenes para crear la base de datos. B Bits			p Píxeles W Ancho H Alto		

En la Tabla 4.2 se muestra la resolución de las imágenes. La resolución de las imágenes para crear la base de datos varía entre 323 y 241 píxeles para el ancho (W) y de 231 a 175 para el alto (H). La profundidad es de 24 bits y son convertidas del formato JPEG de la cámara al tipo BMP (esta conversión mejora la presentación de los resultados). La variación en la resolución es resultado del proceso de recorte para lograr encuadrar el rostro en la fotografía. Las resolución de las imágenes de prueba es de 3872 píxeles de ancho y 2592 píxeles de alto. La profundidad es de 24 bits y son de tipo JPEG y también se convierten al tipo BMP.

4.3. Observaciones

Hay que hacer notar las diferencias entre las Figs. 4.4 y 4.7. Se requirió una segunda etapa de adquisición de imágenes, esto en función de los resultados experimentales. En la primera etapa, se adquirieron las 50 imágenes que se muestran en la Fig. 4.4. Sin embargo, faltaron diez fotos adicionales por cada persona necesarias en el sistema de reconocimiento de rostros. Esto nos llevo a una segunda etapa de adquisición. No todas las personas que participaron en la primera etapa estuvieron disponibles para la segunda, por diferentes razones. Finalmente, se logró hacer la adquisición que resultó en un nuevo conjunto de imágenes para integrar la base de datos (Fig. 4.7).

Por otra parte, también hay una diferencia entre las Figs. 4.5 y 4.6. Las fotos que corresponden al rostro en la Fig. 4.5 mantienen el mismo tamaño o resolución, porque es un requisito para la técnica eigenface que se explicó en el capítulo anterior. Estas fotos integran la base de datos del sistema.

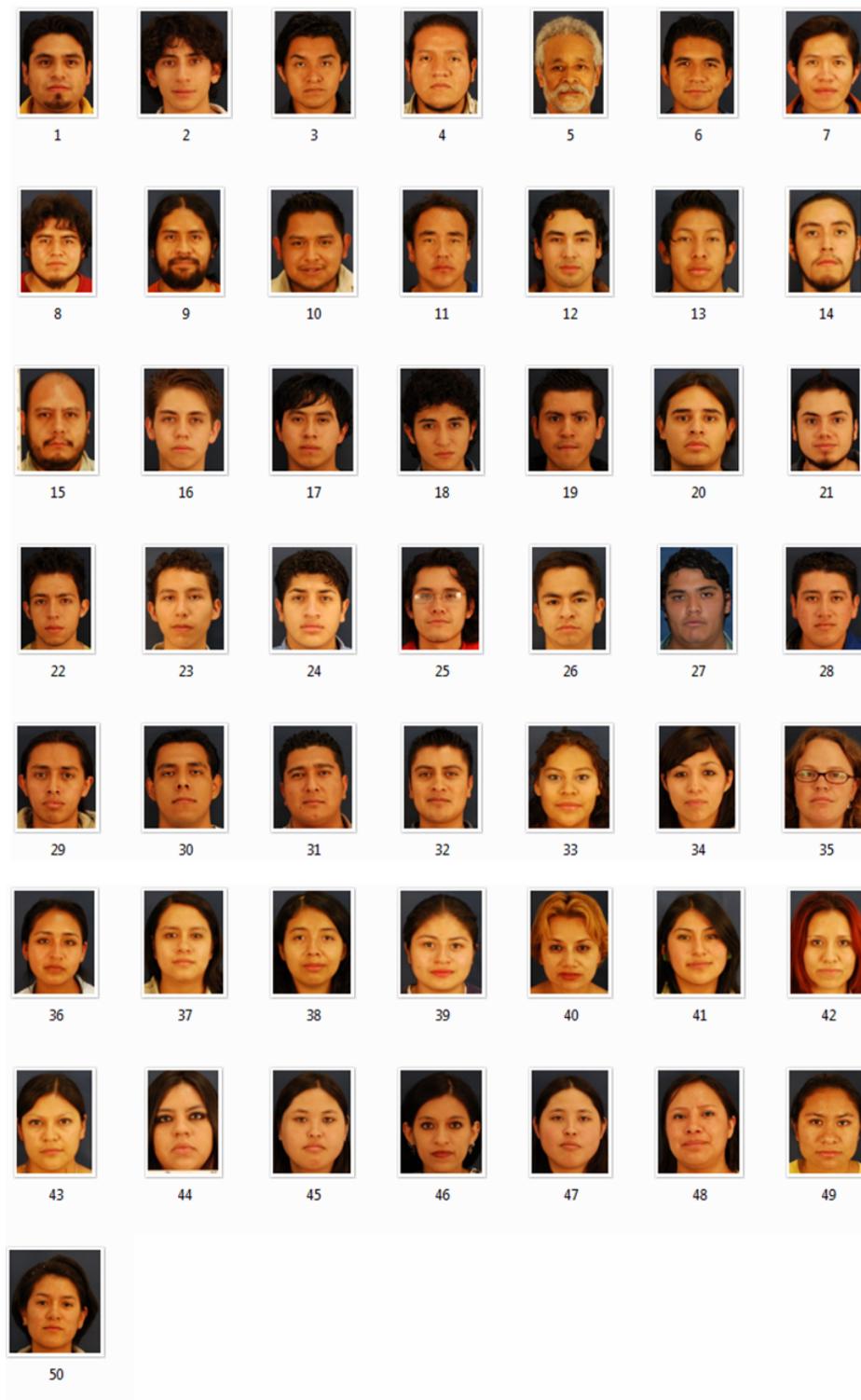


Figura 4.4: Imágenes faciales en color, adquiridas en el set fotográfico e integrarán la base de datos del sistema de reconocimiento facial.

4. PROTOCOLO DE ADQUISICIÓN DE IMÁGENES

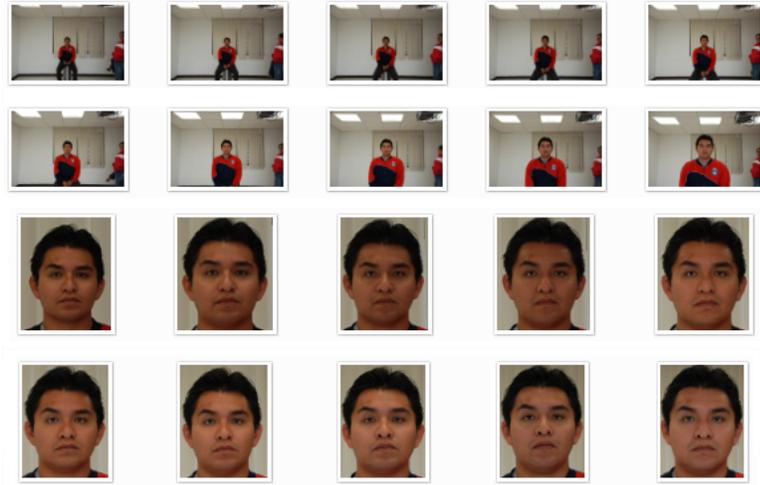


Figura 4.5: Imágenes con cambio de escala y las fotos correspondientes al rostro.

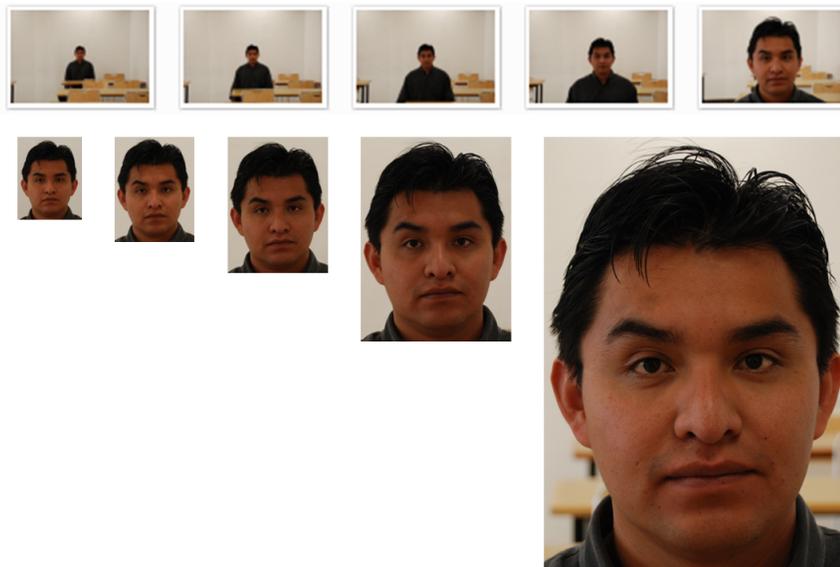


Figura 4.6: Ejemplos de imágenes de prueba. Observese el cambio de escala en el rostro de cada imagen.

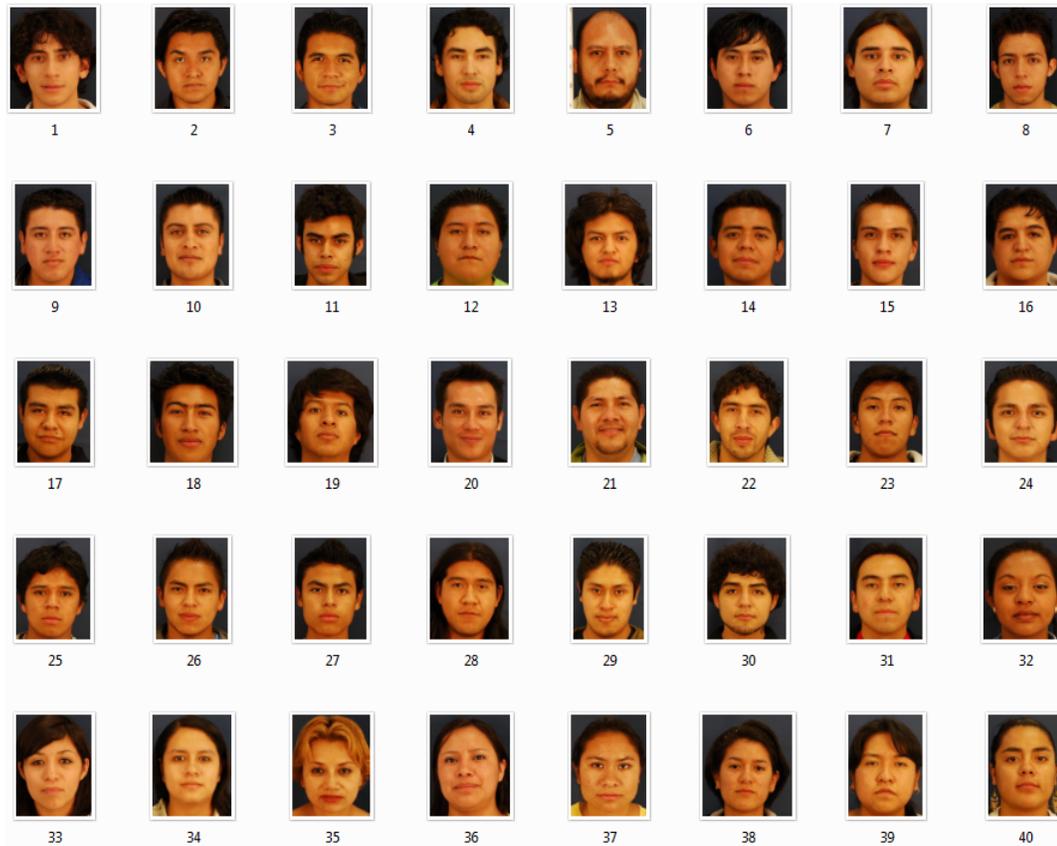


Figura 4.7: Nuevo conjunto de imágenes para integrar la base de datos.

Capítulo 5

LOCALIZACIÓN Y RECONOCIMIENTO DE ROSTROS

El sistema de reconocimiento automático de rostros propuesto en este proyecto es el resultado de una combinación entre el sistema de reconocimiento de rostros genérico (Fig. 1.2) y el sistema de reconocimiento automático de rostros basado en eigenspace (Fig. 1.5) con algunas variaciones. El sistema propuesto incluye las etapas: imagen de entrada, localización facial, extracción de características, imagen de prueba, normalización, imagen promedio del nivel n al nivel $n-k$, donde $k=8,7,\dots,1$; selector de escala, proyección, bases de datos del nivel n al nivel $n-k$, prueba de similitud, sistema de rechazo e identificación. A continuación se describe como opera cada una de estas etapas (Fig. 5.1), que permiten hacer el reconocimiento de rostros con variación en escala.

5.1. Imágenes de Entrada

Para esta etapa se consideran como imágenes de entrada las imágenes que resultarán del protocolo de adquisición descrito en la sección (4.2) del capítulo anterior. Son de dos tipos: imágenes para crear la base de datos (Fig. 5.2a) e imágenes de prueba (Fig. 5.2b). Notemos que las imágenes de prueba tienen la característica del cambio de escala. Ambos tipos de imágenes, deben considerarse como imágenes de entrada. Las primeras, permiten verificar que la propia base de datos es reconocida por el sistema, por tanto, se espera que en la fase de pruebas se obtenga un 100% de aciertos en el reconocimiento para este tipo de imágenes. En el caso de las segundas, sería ideal que el sistema las reconociera al 100%, sin embargo, esto no siempre es así como se observará en la sección de resultados del capítulo siguiente.

5. LOCALIZACIÓN Y RECONOCIMIENTO DE ROSTROS

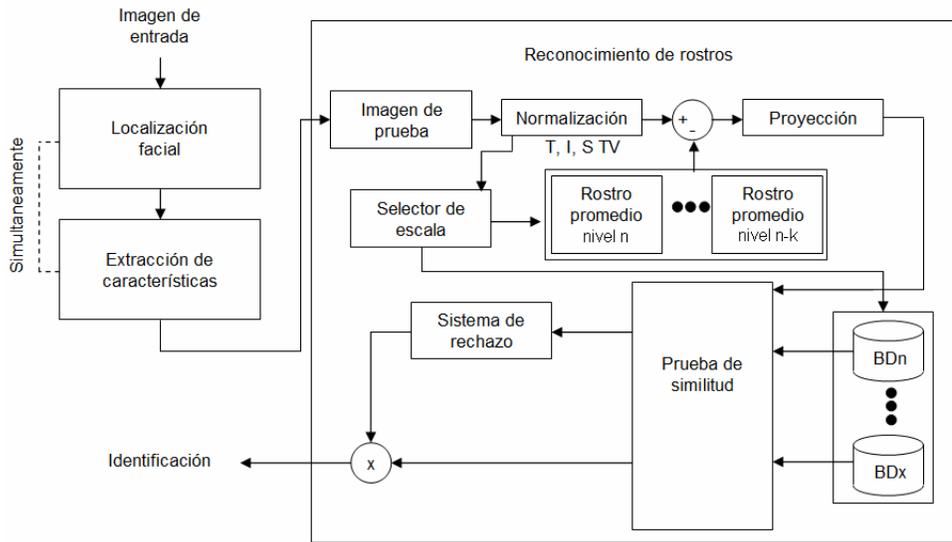


Figura 5.1: Diagrama a bloques del sistema de reconocimiento automático de rostros con variación en escala.



(a)



(b)

Figura 5.2: Ejemplos de imágenes de entrada.

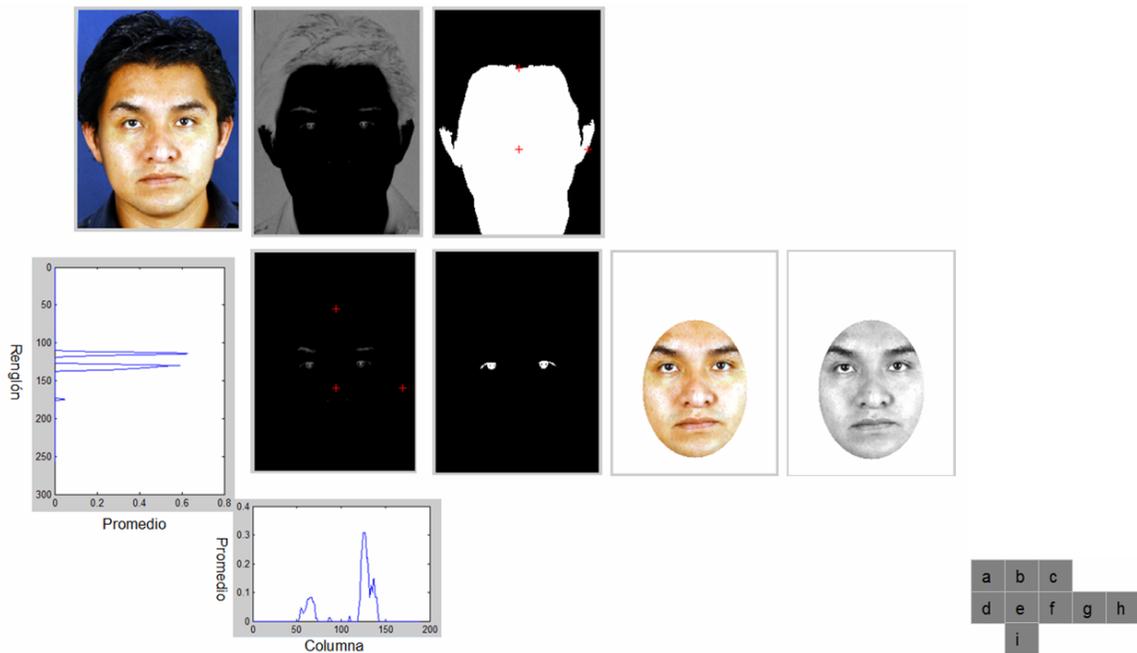


Figura 5.3: Fases en la extracción de características.

5.2. Localización Facial

Dada una imagen de entrada cómo saber si existe, o no, un rostro en esa imagen. Esto se determina a través de la localización del mapa de piel en la imagen de interés.

En la Fig. 5.3a tenemos la fotografía original en color. Para obtener las regiones que caracterizan a los ojos, se hace la conversión del espacio RGB (rojo, verde y azul) de la fotografía al espacio YCbCr (luminancia y crominacia). A continuación se hace la diferencia matricial entre la componente Cb y la componente R para obtener la Fig. 5.3b.

Para obtener el mapa de piel que corresponde al rostro, se hace la diferencia matricial entre la componente Cb y las componentes R y Y (Fig. 5.3c). A continuación se binariza la imagen, se etiquetan las componentes 8-conectadas, y se eliminan las componentes etiquetadas dentro de la región de la cara para obtener la Fig. 5.3c.

5.3. Extracción de Características

Una vez detectado el mapa de piel del rostro se puede proceder a extraer del mismo las características de interés (cejas, ojos, nariz, o boca). En este proyecto es de interés extraer la región que corresponde a los ojos ya que es a través de estos que se puede segmentar la región elíptica del

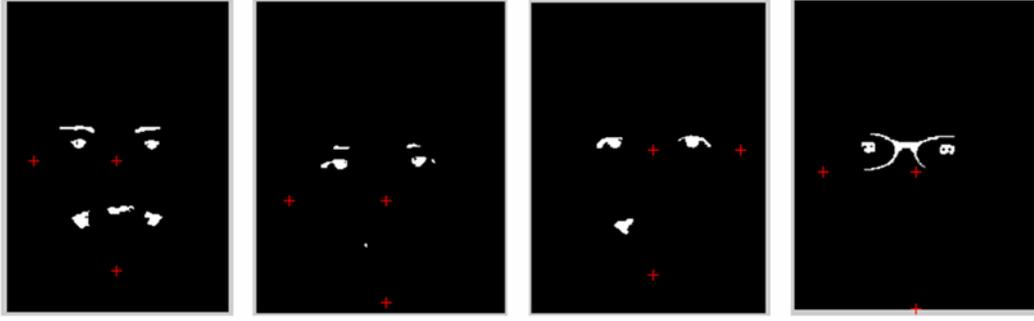


Figura 5.4: Componentes 8-conectadas.

rostro. El proceso de localización del mapa de ojos se realiza de la siguiente forma. De la Fig. 5.3b se obtienen los píxeles que estén dentro de la región blanca de la Fig. 5.3c y se obtiene la Fig. 5.3e. Se binariza y se etiqueta la imagen para poder eliminar las componentes 8-conectadas que resulten y sean menores a la desviación estandar del conjunto de componentes así como las que estén en la vertical entre los ojos (Fig. 5.3), a continuación se obtienen las gráficas de relieve [29], que se refieren a los promedios de las intensidades de los píxeles correspondientes a los renglones (Fig. 5.3d) y a las columnas (Fig. 5.3i) de la imagen, esto para poder obtener los puntos (x, y) que corresponden a cada uno de los ojos. Lo anterior, es necesario para poder discriminar adecuadamente de entre las componentes 8-conectadas que no corresponden a los ojos (Fig. 5.4).

La distancia d , en píxeles, entre los ojos, permite aplicar las ecuaciones (5.1-5.6) que se describen en [11], para extraer la región elíptica del rostro (Fig. 5.3g, en color y Fig. 5.3h, en escala de grises) utilizando la imagen en color de entrada.

$$\theta = \arctan((y_l - y_r) / (x_l - x_r)) \quad (5.1)$$

$$x_0 = (x_l - x_r) / 2 + 0.43 \times d \times \cos(\theta) \quad (5.2)$$

$$y_0 = (y_l - y_r) / 2 - 0.43 \times d \times \cos(\theta) \quad (5.3)$$

$$\beta = 1.3 \times d \quad (5.4)$$

$$\alpha = 0.9 \times d \quad (5.5)$$

$$d = \sqrt{(x_l - x_r)^2 + (y_l - y_r)^2} \quad (5.6)$$

donde:

(x_0, y_0) corresponde al centro de la elipse.

(x_l, y_l) y (x_r, y_r) indican las coordenadas del ojo izquierdo y el ojo derecho respectivamente.

d es la distancia entre los ojos.

α es el eje menor de la elipse.

β es el eje mayor de la elipse.

θ es la orientación de la elipse.

5.4. Reconocimiento de Rostros

En esta etapa se compara la imagen de prueba contra las imágenes que forman la base de datos del sistema. Esto, a través del proceso de normalización, proyección de la imagen de prueba en el espacio de eigenfaces y comparación de similitud.

5.4.1. Imagen de prueba

La imagen de prueba pasa a ser la región elíptica del rostro que se obtuvo en la sección, anterior, de extracción de características (Fig. 5.3h).

5.4.2. Normalización

La normalización es un proceso de ajuste de cada una de las imágenes a nivel de iluminación, resolución, alineación de ojos, nariz y boca [23]. En la (Fig. 5.5) se ilustra un ejemplo de cambio de resolución y el suavizado de la imagen. La imagen pasa de una resolución de 274x198 píxeles (Fig. 5.5a) a 241x175 píxeles (Fig. 5.5b). El suavizado con TV (Fig. 5.5c) elimina ruido en la imagen y aumenta la probabilidad de mejorar la coincidencia en el reconocimiento de rostros. Los ajustes a nivel de alineación no son necesarios ya que en la adquisición de imágenes cada uno de los rostros pasó por este protocolo de alineación (ver rostro promedio más adelante).

El proceso de normalización en iluminación se realiza aplicando la ecuación (3.28) a la imagen, junto con el cálculo de los eigenfaces como se describe más adelante.

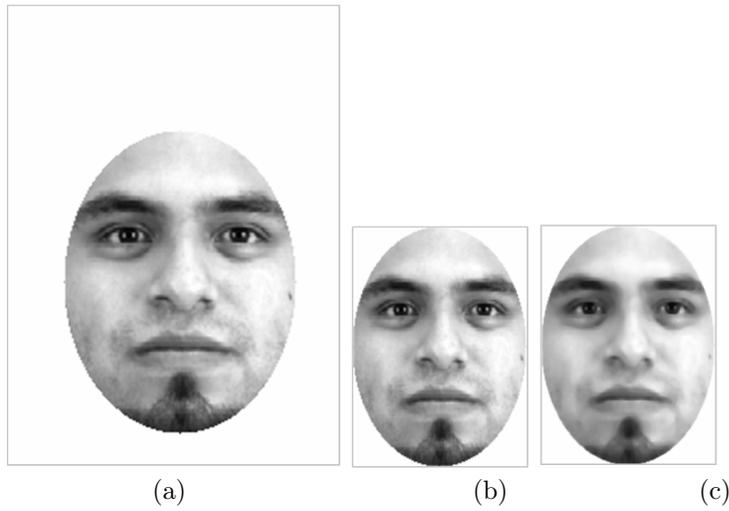


Figura 5.5: Ejemplo de normalización y suavizado. a) Resolución 274x198 píxeles; b) Resolución 241x175 píxeles; c) Suavizado con variación total.

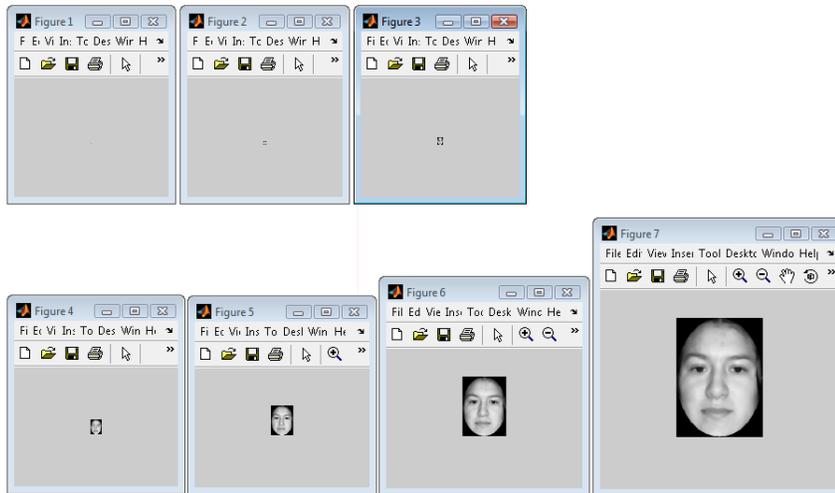


Figura 5.6: Niveles 1-7.

5.4.3. Bases de datos

Para hacer reconocimiento con variación en escala es necesario crear diferentes niveles, pirámide, de bases de datos con objeto de que las imágenes de entrada con diferente resolución se comparen con la resolución más cercana, hacia abajo de la pirámide.

La resolución de las imágenes es de 241x175 píxeles para el nivel 8. Está resolución se puede considerar como la base de la pirámide. Los niveles 7, 6, 5, 4, 3, 2 y 1 se muestran en la Fig. 5.6.

5.4.4. Selector de escala

En esta etapa se selecciona la base de datos más adecuada para compararse con la imagen de prueba. Es decir, si la resolución de la imagen de prueba está por debajo de un nivel de la pirámide, entonces se debe escoger el siguiente nivel, hacia abajo de la misma y la imagen de entrada se debe normalizar en resolución al nivel seleccionado ya que para hacer la comparación se requiere que las resoluciones sean iguales. No se debe normalizar la imagen de prueba hacia arriba de la pirámide ya que se pierden detalles de la imagen en ese proceso, que afectarán el reconocimiento. La pérdida de detalles también se da hacia abajo de la pirámide pero hasta los niveles más bajos, entre el 4 y el 1, que es cuando empieza a desvanecerse la forma del rostro hasta convertirse en un píxel.

5.4.5. Rostro promedio

El cálculo del rostro promedio se realiza aplicando la ecuación (3.20). Para el caso de rostros femeninos (Fig. 5.7), tenemos 17 fotografías. Podemos observar que la imagen promedio caracteriza al rostro femenino (esto es menos robusto en las cejas, nariz y boca, comparado con el rostro masculino). También, se alcanza a observar la forma de los lentes, esto es así debido al número, menor, de fotografías comparado con el conjunto de rostros masculinos.

5.4.6. Proyección

Es necesario que la imagen de prueba se proyecte en el espacio de eigenfaces aplicando la ecuación (3.24) con objeto de que se pueda reducir la dimensionalidad de la imagen y determinar los pesos que le corresponden en este espacio y permitir la comparación de similitud en la siguiente etapa de reconocimiento, para determinar si se encuentra o no esa imagen en el espacio de eigenfaces o base de datos del sistema. En la Fig. 5.9 se muestran el espacio de eigenfaces correspondientes al conjunto anterior de rostros femeninos.

Se debe crear un conjunto de bases de datos que contengan los espacios de eigenfaces para cada nivel en la pirámide. El espacio de eigenfaces se crea aplicando las ecuaciones (3.18-3.23).

La pirámide de bases de datos se crea reduciendo las imágenes que se asignaron como integrantes de la base de datos (4.7) aplicando procesamiento multiresolución con la ecuación (3.46) para cada nivel en la pirámide y a continuación la aplicación del conjunto de ecuaciones que corresponden a la técnica eigenface a cada nivel. De esta forma se tendrán n espacios de eigenfaces.



Figura 5.7: Rostros femeninos normalizados en tamaño, iluminación y con suavizado.



Figura 5.8: Imagen promedio, rostros femeninos.

5.4.7. Prueba de similitud

En esta prueba se compara la imagen proyectada en el espacio de eigenfaces aplicando la ecuación (3.26). Como se calcula el valor mínimo de la diferencia de la imagen proyectada contra las imágenes en el espacio de eigenfaces, el resultado de la prueba puede caer en una clase que no corresponda en realidad al rostro que le debe corresponder. Esto nos producirá diferentes resultados. Si la imagen forma parte de la base de datos pero el resultado indica que no, entonces se tendrá un falso negativo. Si la imagen no forma parte de la base de datos pero el resultado indica que si, entonces se tendrá un falso positivo.

5.4.8. Sistema de Rechazo

Para evitar que la imagen produzca resultados falsos, se requiere una etapa que rechace estos resultados aplicando la ecuación (3.27). Aquí se comparan el peso de la imagen de prueba contra el peso de la imagen de la base de datos si la base de datos se compone de una imagen por clase. La comparación se hace verificando la diferencia entre ambas imágenes, si son iguales entonces la



Figura 5.9: Conjunto de eigenfaces para rostros femeninos.

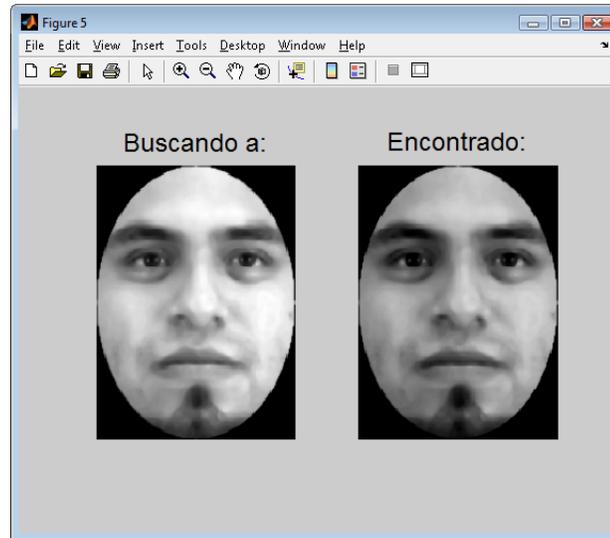


Figura 5.10: Ejemplo de identificación.

diferencia será cero. Si es un grupo de imágenes las que forman una clase, entonces la verificación consiste en que la diferencia entre los pesos de la imagen de prueba y el promedio de los pesos de las imágenes que forman la clase no rebase un umbral mínimo y máximo, que se determina a partir del grupo de imágenes que forman la clase determinando cuál es el peso mínimo y máximo del grupo de imágenes en esa clase.

5.5. Identificación

La identificación es el resultado final de la prueba de similitud y del sistema de rechazo. Si forma parte de la base de datos entonces la identificación es positiva. El caso ideal es que la identificación sea siempre positiva para las imágenes que se prueban en el sistema y que formen parte de la base de datos.

Capítulo 6

PRUEBAS Y RESULTADOS

Una vez especificado los procedimientos que muestran como hacer la localización y reconocimiento de rostros, descritos en el capítulo anterior, continuamos con la sección de pruebas y resultados. Aquí, se aplican pruebas que verifiquen resultados esperados, que nos den indicios de ajustes para aspectos no contemplados en las etapas iniciales del proyecto y, finalmente, que nos orienten en la dirección de los algoritmos más adecuados que nos acerquen a la comprensión del reconocimiento de rostros con variación en escala.

6.1. Inicialización de la Base de Datos

Para hacer reconocimiento de rostros se requiere de la bases de datos que permita la verificación contra la imagen de prueba. En esta sección se describen las pruebas necesarias para generar las bases de datos del sistema de reconocimiento de rostros. Cómo es reconocimiento de rostros con cambios de escala, es necesario generar varias bases de datos, esto de acuerdo a la pirámide de imágenes que va del nivel 8 al 1. Para esto es necesario realizar los siguientes procesos: (1) Localizar el mapa de ojos y extraer la región elíptica del rostro; (2) Normalizar las regiones elípticas en: tamaño o resolución, iluminación y suavizado; (3) Generar el conjunto de eigenfaces; (4) Generar pirámide de bases de datos. A continuación se describen estos procesos.

6.1.1. Localización y extracción de características

Para poder hacer la extracción de la región en forma de elipse que corresponde a la cara es necesario localizar la región que corresponde al mapa de piel y posteriormente localizar el mapa de ojos sobre este mapa de piel. A continuación, para segmentar y extraer la región de interés, utilizar

la posición de los ojos para aplicar las ecuaciones de la elipse vistas en la sección (5.3) a las imágenes de la Fig. 6.1. El resultado de este proceso se indica en la Fig. 6.2. Observemos que, de las cincuenta imágenes hay una que presenta error, la imagen ec48 (elipse en color #48).

Junto con la segmentación de las elipses en color, se obtienen los datos que se muestran en la Tabla 6.1, en la que se reporta lo siguiente: el número de fotografía (ec#.bmp), la distancia (d) entre los ojos, el centro de la elipse (x_0, y_0), su eje menor y eje mayor, y la resolución de cada imagen. Así también, en la parte inferior de la tabla se indica una fila con los valores mínimos de cada columna. Estos datos serán de utilidad para el proceso de normalización de la sección que sigue.

En el apéndice D.2 se muestra el código en MATLAB que realiza la localización y extracción de características.

6.1.2. Normalización

Para normalizar el conjunto de fotografías dado en la Fig. 6.2, se utilizan los datos de la Tabla 6.1. Los valores mínimos de la resolución son los que corresponderán a la normalización en resolución. Esto es, todas las fotografías se convertirán a la resolución 241x175 píxeles, que denominaremos resolución global. El número de matrices por imagen cambiará a una, ya que también se convertirán a escala de grises. El centro (x_0, y_0) de cada elipse se lleva al centro de la resolución global de cada imagen. Esto permitirá que todas las imágenes aparezcan centradas en la resolución global. En la Fig. 6.3, se muestra el resultado del proceso de normalización, así como de suavizado con TV. Se tienen cincuenta imágenes normalizadas, en1-en50, donde las letras “en” significan **e**lipse **n**ormalizada. En los apéndices D.3 y D.1 se muestran los códigos en MATLAB que realizan el proceso de normalización y de suavizado (con TV), respectivamente.

El proceso de normalización en iluminación se realiza junto con el cálculo de los eigenfaces como se describe más adelante.

6.1.3. Cálculo de rostro promedio y eigenfaces

A continuación, se muestran ejemplos de cálculo de rostro promedio y eigenfaces para: rostros masculinos, rostros femeninos y rostros mixtos. Ver código en MATLAB en el apéndice D.4.

En la Fig. 6.4, tenemos un conjunto de 24 fotografías con rostros masculinos. La imagen promedio se muestra en la Fig. 6.5, observemos que esta imagen tiene la forma de un sólo rostro, esto es, frente, cejas, nariz y boca definidas.

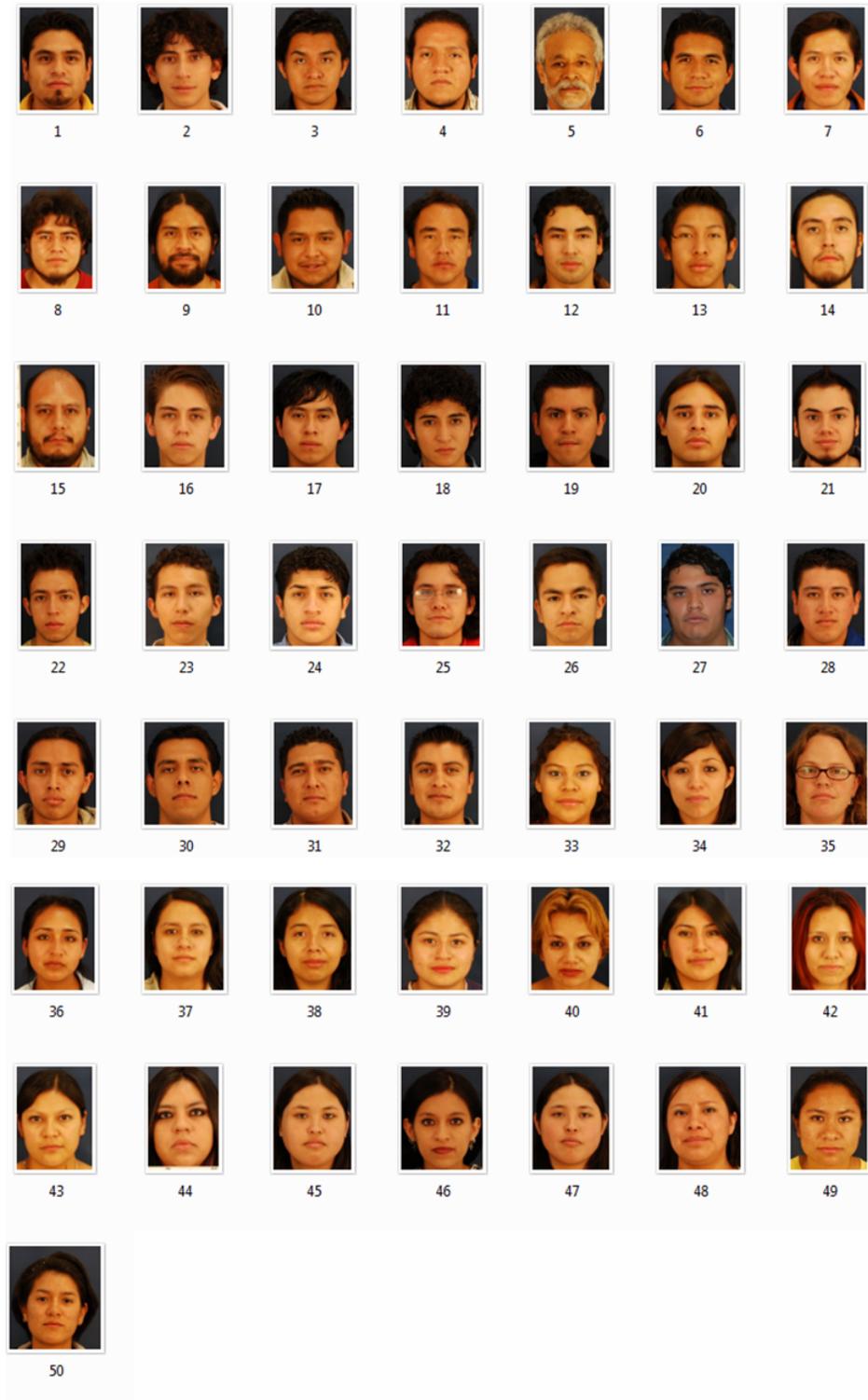


Figura 6.1: Imágenes faciales que integrarán la base de datos del sistema de reconocimiento de rostros.



Figura 6.2: Extracción de la región de la elipse (imágenes en color) en el rostro.

Tabla 6.1: Parámetros del conjunto de elipses.

Foto ec#.bmp (#)	d (p)	x ₀ (p)	y ₀	Eje menor (p)	Eje mayor (p)	Resolución		
						Renglones (p)	Columnas (p)	Matrices
1	69	103	166	69	90	274	198	3
2	61	100	142	61	80	241	204	3
3	61	96	159	61	80	257	190	3
4	72	92	152	72	94	258	185	3
5	58	90	165	58	76	255	175	3
6	63	94	163	63	82	257	185	3
7	64	99	153	64	84	254	192	3
8	54	91	153	54	71	258	180	3
9	58	93	145	58	76	260	185	3
10	65	99	158	65	85	251	200	3
11	60	97	152	60	78	253	185	3
12	66	101	152	66	86	254	200	3
13	70	110	155	70	91	257	210	3
14	67	96	137	67	88	257	185	3
*15	54	106	138	54	71	256	193	3
16	63	98	156	63	82	258	200	3
17	61	94	153	61	80	253	200	3
18	62	107	167	62	81	258	202	3
19	54	95	142	54	71	243	189	3
20	64	97	152	64	84	274	202	3
21	70	88	166	70	91	271	181	3
22	65	90	171	65	85	276	189	3
23	65	99	160	65	85	270	207	3
24	64	99	152	64	84	253	195	3
25	59	95	151	59	77	261	198	3
26	86	107	205	86	112	320	229	3
27	66	94	160	66	86	285	202	3
28	64	103	162	64	84	268	202	3
29	65	92	148	65	85	252	193	3
30	61	91	138	61	80	245	190	3
31	62	98	143	62	81	256	196	3
32	83	107	185	83	108	323	231	3
33	69	98	154	69	90	253	198	3
34	67	88	142	67	88	254	188	3
35	62	91	151	62	81	253	200	3
36	58	107	151	58	76	256	200	3
37	61	100	136	61	80	256	196	3
38	69	99	146	69	90	253	190	3
39	65	105	160	65	85	254	200	3
40	64	98	170	64	84	256	194	3
41	63	100	124	63	82	253	204	3
42	63	101	155	63	82	256	185	3
43	73	84	157	73	95	257	188	3
44	77	83	150	77	101	255	175	3
45	64	102	146	64	84	253	190	3
46	67	102	161	67	88	254	200	3
47	63	98	147	63	82	258	190	3
*48	45	101	173	45	59	258	205	3
49	65	102	157	65	85	258	190	3
50	70	98	126	70	91	258	200	3
mínimos	54	83	124	54	71	241	175	3

Notas:
1.- (p) = píxeles.
2.- Los valores mínimos no toman en cuenta las imágenes 15 y 48.



Figura 6.3: Conjunto de imágenes normalizados en tamaño y suavizado con TV.

El conjunto de eigenfaces generado se muestra en la Fig. 6.6). Notemos que en este conjunto, en algunas imágenes se destacan la forma ovalada del rostro, la cejas, los ojos, la barba, el bigote y la boca. Estas son características finitas del espacio infinito que representan el conjunto de datos o imágenes.

Para el caso de rostros femeninos (Fig. 6.7), tenemos 17 imágenes. Podemos observar que la imagen promedio caracteriza al rostro femenino (este es menos robusto en las cejas, nariz y boca, comparado con el rostro masculino). También, se alcanza a observar la forma de los lentes, esto es así debido al número, menor, de imágenes comparado con el conjunto de rostros masculinos. En la (Fig. 6.9) se muestran el conjunto de eigenfaces correspondientes.



Figura 6.4: Rostros masculinos normalizados en tamaño, iluminación y suavizados con TV.



Figura 6.5: Imagen promedio, rostros masculinos.

Al considerar todas las imágenes, de rostros masculinos y femeninos, obtenemos la imagen promedio y el espacio de eigenfaces que se muestran en las (Figs. 6.10 y 6.11) respectivamente.

En general, es más fácil localizar un rostro masculino que un rostro femenino en una fotografía, ya que son las mujeres las que tienden a embellecer su rostro con productos que alteran las características naturales del rostro. Por ejemplo, el maquillaje introduce colores que agrandan o reducen el espacio de zonas como los ojos, esto afecta la localización del mapa de ojos. Este mapa de ojos es el proceso de extracción de características más importante, ya que con estos se obtiene el parámetro d , que es la distancia entre los ojos, el cual permite determinar la región de la elipse en el rostro. La elipse es

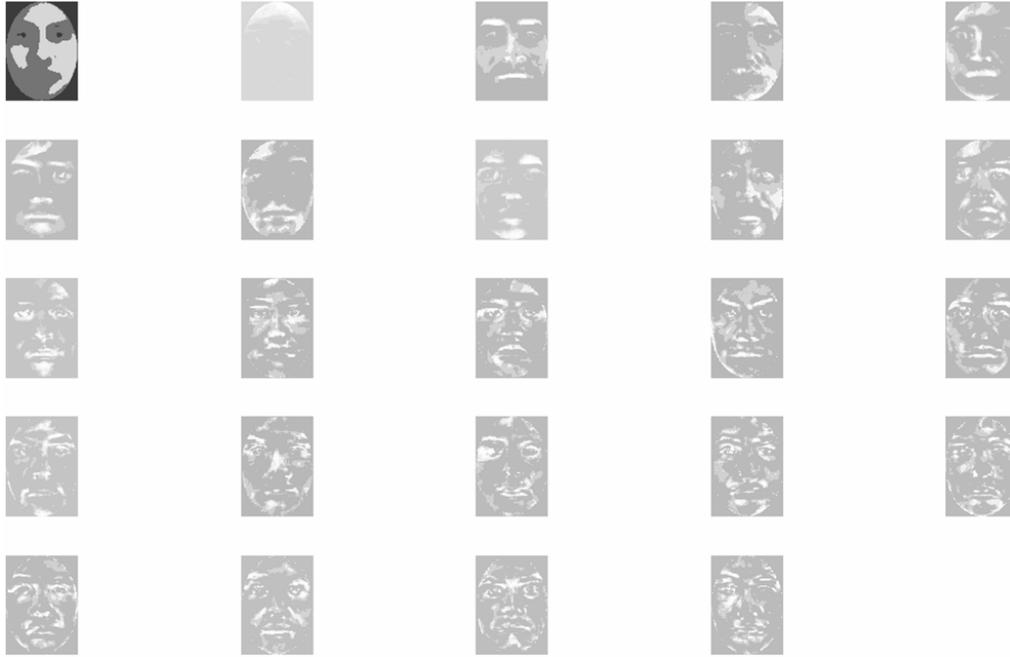


Figura 6.6: Conjunto de Eigenfaces para rostros masculinos.



Figura 6.7: Rostros femeninos normalizados en tamaño, iluminación y suavizados con TV.



Figura 6.8: Imagen promedio, rostros femeninos.



Figura 6.9: Conjunto de eigenfaces para rostros femeninos.



Figura 6.10: Imagen promedio, para rostros mixtos.

importante porque, como se mencionó en la sección 3.4, elimina la influencia del ruido por el fondo y el cabello. Se puede decir lo mismo en el caso del cabello, en las mujeres es más problemático eliminar su influencia ya que las mujeres tienden a colorearlo y a darle formas que dificultan el proceso de localización de los ojos, al generar demasiadas componentes 8-conectadas.

6.1.4. Generación de la pirámide de imágenes

Para hacer reconocimiento con variación en escala es necesario crear diferentes niveles de bases de datos con objeto de que las imágenes con diferente resolución se comparen con la resolución más cercana, hacia abajo de la pirámide.

A continuación utilizaremos la base de datos de imágenes con las regiones del rostro en forma de elipse, dada por la (Fig. 6.4) para generar la pirámide de imágenes que se indica en la tabla (6.2). La resolución actual de estas imágenes es de 241x175 píxeles. Esta resolución se puede considerar como la base de la pirámide, nivel 8. Los niveles 7, 6, 5, 4, 3, 2 y 1 se muestran en las Figs. 6.12 - 6.18 respectivamente.



Figura 6.11: Conjunto de eigenfaces, para rostros mixtos.

Tabla 6.2: Niveles en la pirámide de imágenes.

J	2^J	Resolución
8	2^8	256
	$\approx 2^8$	241
7	2^7	128
6	2^6	64
5	2^5	32
4	2^4	16
3	2^3	8
2	2^2	4
1	2^1	2
0	2^0	1

Ver código en MATLAB en el apéndice D.5, para generar la pirámide de imágenes.

Podemos observar que se empiezan a perder características de la imagen a partir del nivel 3 (Fig. 6.19). Por tanto, podemos decir que los niveles más apropiados para reconocimiento van del nivel 8 al nivel 4.

6.2. Pruebas de Reconocimiento de Rostros

En esta sección se muestran las pruebas de reconocimiento de rostros para las imágenes de prueba que forman la base de datos, así como de aquellas que no se integran a la base de datos pero que si cambian de escala.



Figura 6.12: Nivel 7.



Figura 6.13: Nivel 6.

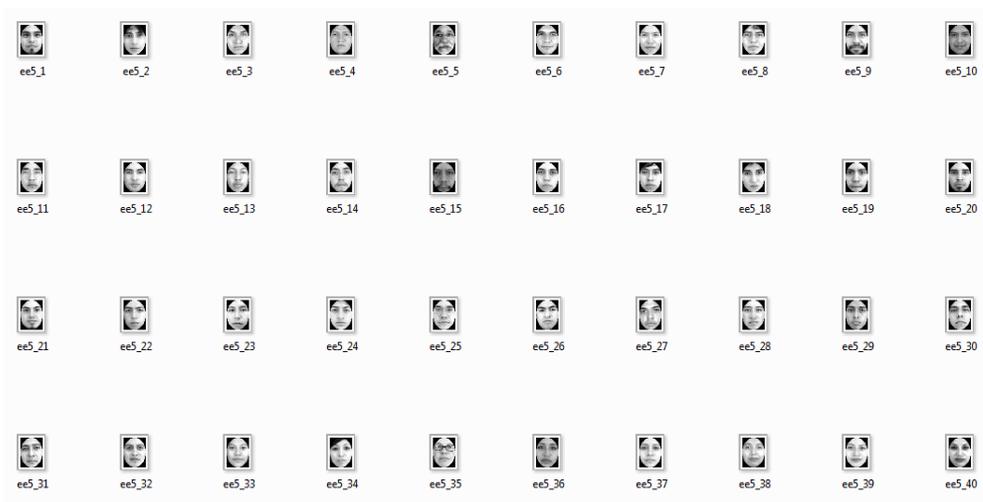


Figura 6.14: Nivel 5.

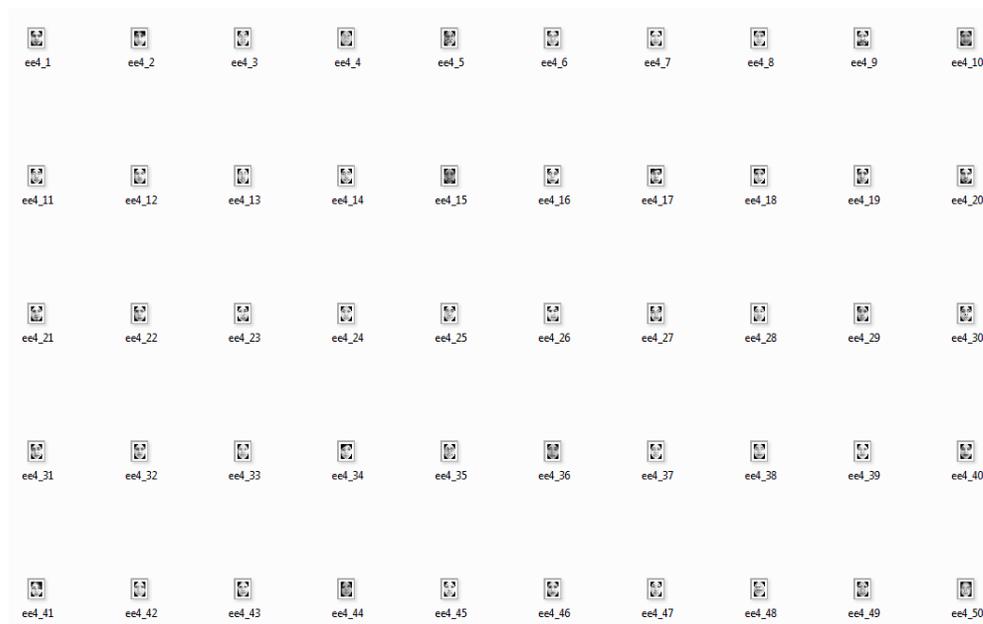


Figura 6.15: Nivel 4.

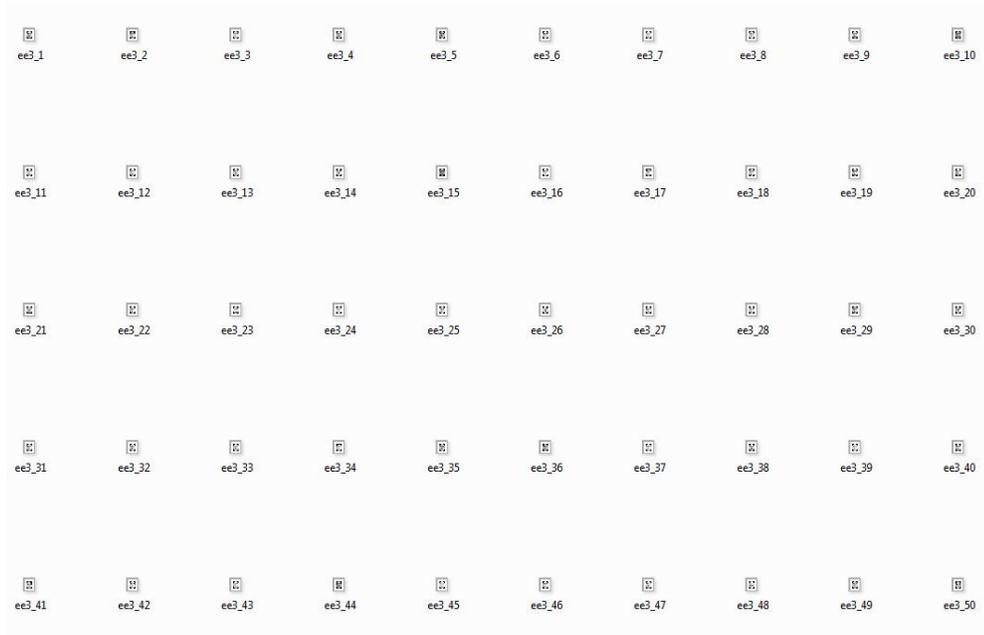


Figura 6.16: Nivel 3.



Figura 6.17: Nivel 2.

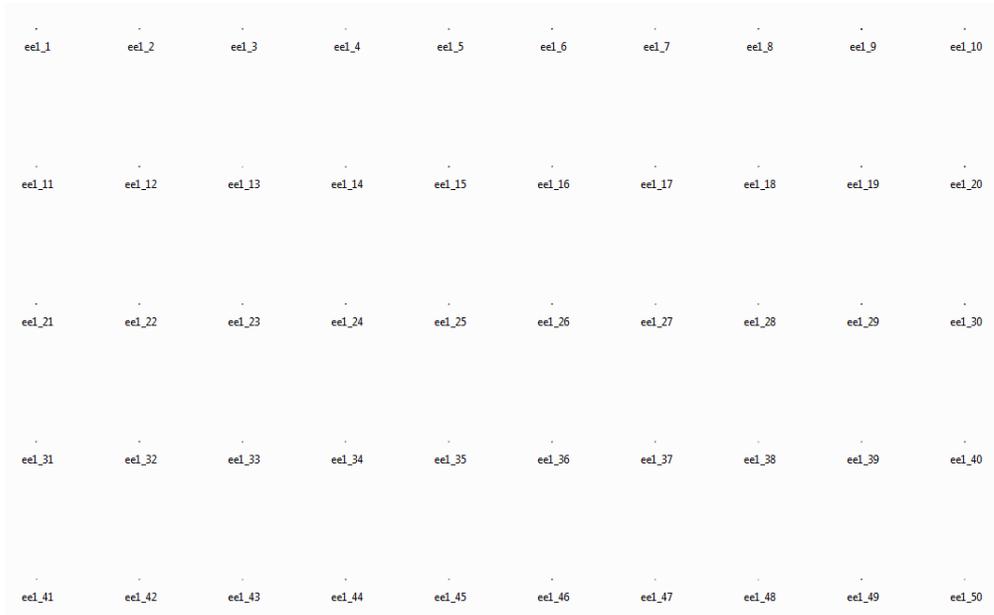


Figura 6.18: Nivel 1.

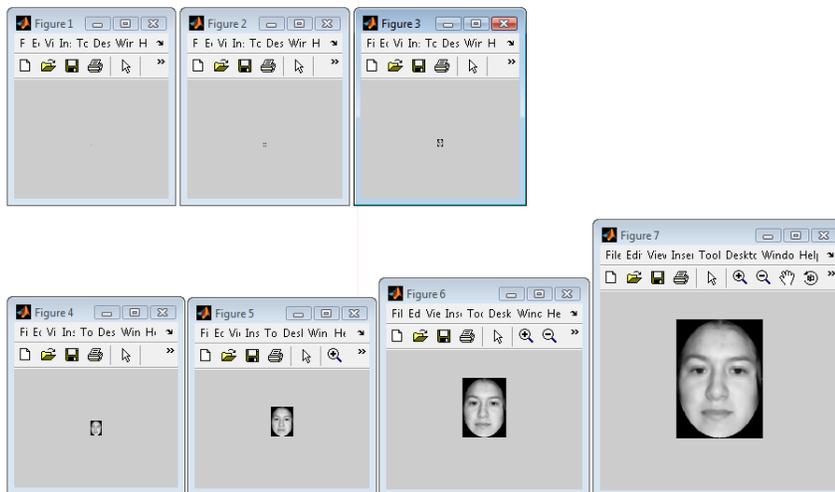


Figura 6.19: Niveles 1-7.

Tabla 6.3: Resultados de Reconocimiento para 49 imágenes.

Nivel	Aciertos	Errores
8	100 %	0 %
7	100 %	0 %
5	100 %	0 %
5	100 %	0 %
4	100 %	0 %

6.2.1. Pruebas de reconocimiento de la base de datos

A continuación se obtiene el espacio de eigenfaces para cada nivel y se hacen pruebas para determinar los resultados de reconocimiento tomando como imágenes de prueba las imágenes que conforman las bases de datos (Fig. 6.1), imágenes que no se integran a la base de datos pero que si deben ser reconocidas por el sistema. Los resultados de estas pruebas se indican en la Tabla 6.3 y en la Fig. 6.20 se muestra un ejemplo de reconocimiento para cada nivel. Notemos por los gránulos de la imagen como se va perdiendo la forma del rostro con la disminución de la resolución en cada nivel. Los niveles que se prueban van del 8 al 4. Si se omite la imagen que presenta error (en48.bmp), entonces tenemos un 100 % de reconocimiento para 49 imágenes. Este era un valor esperado y las pruebas lo confirman. Es decir, si sólo se integran a la base de datos las imágenes que no presentarán error en la extracción de características, entonces el reconocimiento será del cien por ciento, como debe de ser.

Ver código en MATLAB en el apéndice D.6, que verifica los niveles 8 al 4 de las bases de datos.

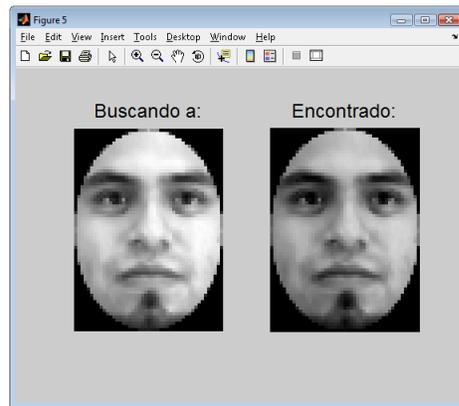
6.2.2. Pruebas de reconocimiento de imágenes con variación en escala

El conjunto de imágenes de la sección anterior (Fig. 6.1) no fueron suficientes para asegurar el reconocimiento de imágenes con variación en escala como los que se indican en la Fig. 6.21.

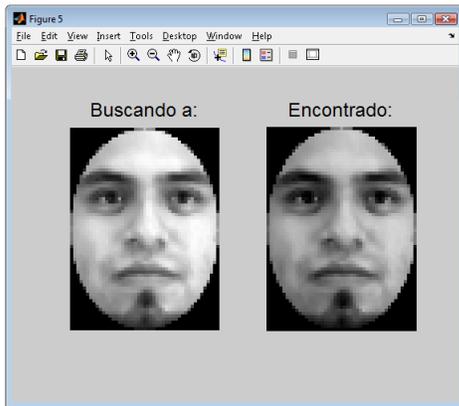
Después de diagnosticar el problema, se concluyó que era necesario agregar a ese grupo de imágenes otro conjunto de nueve más para formar un grupo de diez imágenes por clase como se muestra en la prueba de la Fig. 6.25. La decisión de considerar un nuevo grupo de diez imágenes se basó en el algoritmo de reconocimiento de rostros propuesto en [33]. La prueba consistió de lo siguiente. Las cinco imágenes de prueba con variación en escala se redujeron a la resolución de 92×112 . Las otras diez imágenes, correspondientes a las elipses reducidas (Fig. 6.23) con la misma resolución de 92×112 , se integraron a la base de datos del sistema de reconocimiento de rostros que se indica en el apéndice D.7. Lo anterior haciendo recortes manuales de las elipses con software de edición de imágenes. Era de esperar que las resoluciones de las imágenes de prueba al reducirlas a



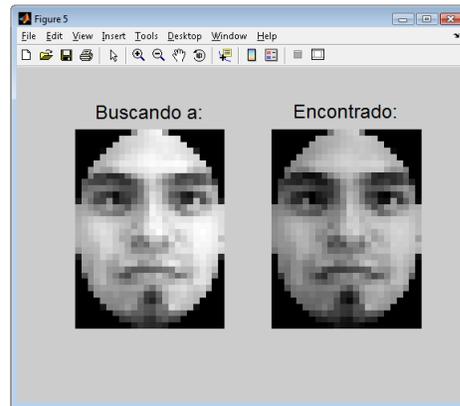
(a)



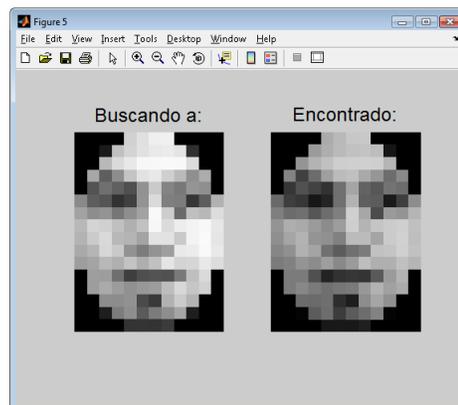
(b)



(c)



(d)



(e)

Figura 6.20: Ejemplos de reconocimiento. a) Nivel 8; b) Nivel 7; c) Nivel 6; d) Nivel 5; y e) Nivel 4.

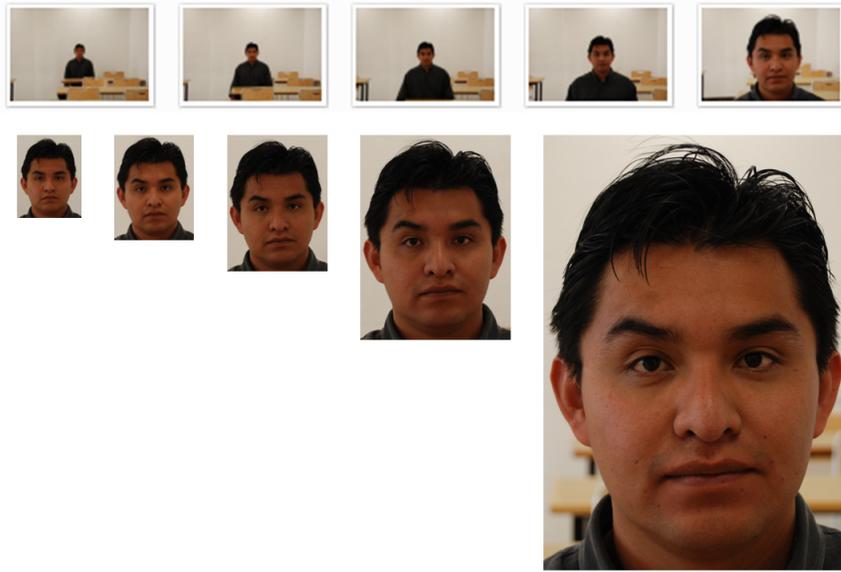


Figura 6.21: Ejemplos de imágenes de prueba. Obsérvese el cambio de escala en el rostro de cada imagen.

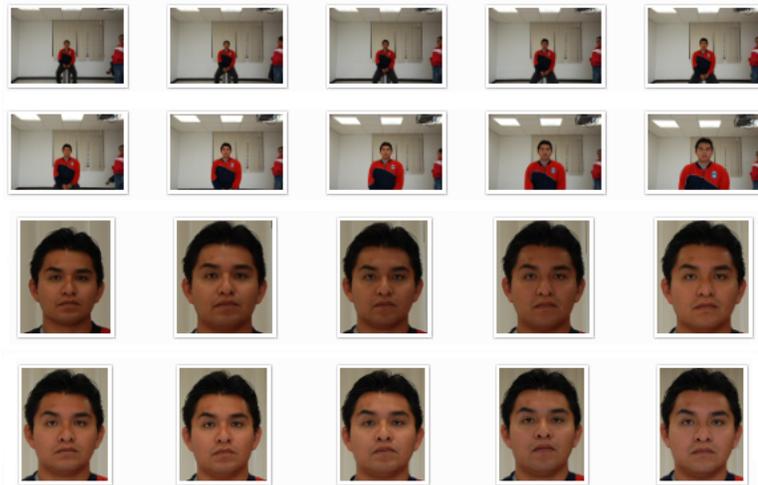


Figura 6.22: Imágenes con cambio de escala y las fotos correspondientes al rostro.

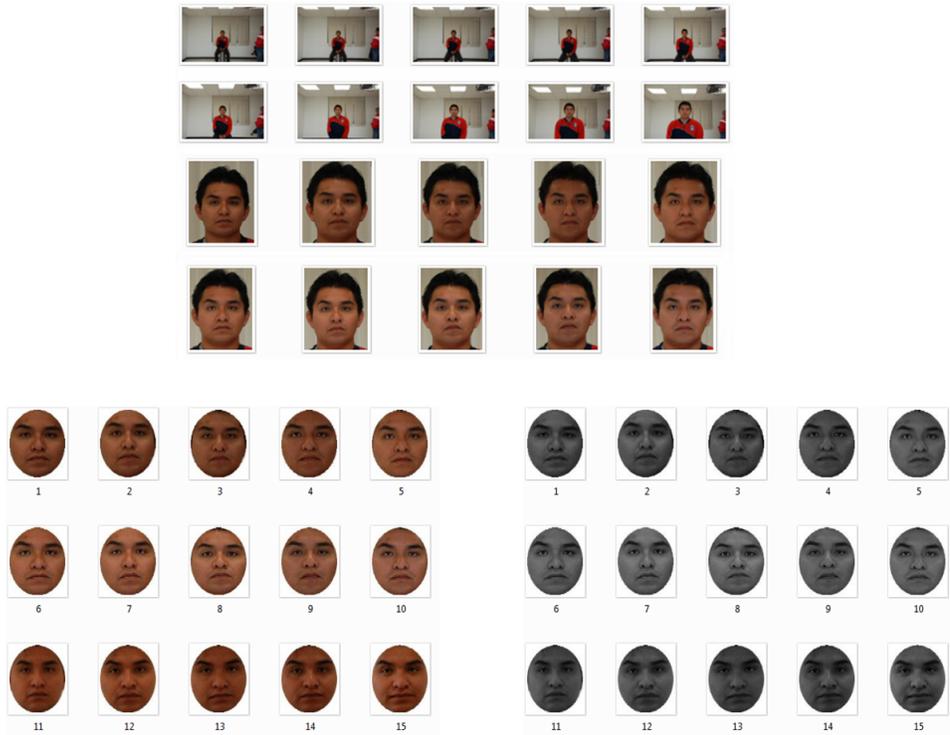


Figura 6.23: Elipses reducidas a la resolución de 92×112 .



Figura 6.24: Imágenes de prueba con variación en escala reducidas a, elipses con resolución de 92×112 , con variación en escala.

Tabla 6.4: Parámetros del nuevo conjunto de elipses.

Foto ec#.bmp (#)	d (p)	x_0 (p)	y_0 (p)	Eje menor (p)	Eje mayor (p)	Resolución		
						Renglones (p)	Columnas (p)	Matrices
1	61	100	142	61	80	241	204	3
2	61	96	159	61	80	257	190	3
3	63	94	163	63	82	257	185	3
4	66	101	152	66	86	254	200	3
5	54	106	138	54	71	256	193	3
6	61	94	153	61	80	253	200	3
7	70	108	158	70	91	264	217	3
8	65	90	171	65	85	276	189	3
9	64	103	162	64	84	268	202	3
10	83	107	185	83	108	323	231	3
11	62	88	165	62	81	264	186	3
12	63	99	147	63	82	251	193	3
13	58	104	151	58	76	238	203	3
14	74	107	168	74	97	269	206	3
15	67	89	166	67	88	269	184	3
16	64	89	158	64	84	258	192	3
17	70	98	176	70	91	285	204	3
18	63	97	150	63	82	236	193	3
19	61	100	151	61	80	247	209	3
20	66	109	169	66	86	274	207	3
21	60	102	139	60	78	260	193	3
22	61	87	153	61	80	255	185	3
23	65	91	145	65	85	255	182	3
24	67	93	154	67	88	243	188	3
25	65	98	176	65	85	270	196	3
26	71	96	172	71	93	274	195	3
27	68	92	166	68	89	263	187	3
28	66	108	158	66	86	271	212	3
29	66	101	182	66	86	289	201	3
30	61	96	165	61	80	258	199	3
31	69	93	152	69	90	261	189	3
32	55	103	138	55	72	233	190	3
33	67	88	142	67	88	254	188	3
34	61	100	136	61	80	256	196	3
35	64	98	170	64	84	256	194	3
36	45	101	173	45	59	258	205	3
37	65	102	157	65	85	258	190	3
38	54	96	140	54	71	221	195	3
39	72	97	155	72	94	255	187	3
40	64	97	136	64	84	253	183	3
Mínimos	54	87	136	54	71	221	182	3
Notas: 1.- (p) Píxeles. 2.- Los valores mínimos no consideran la imagen con error, 36.								

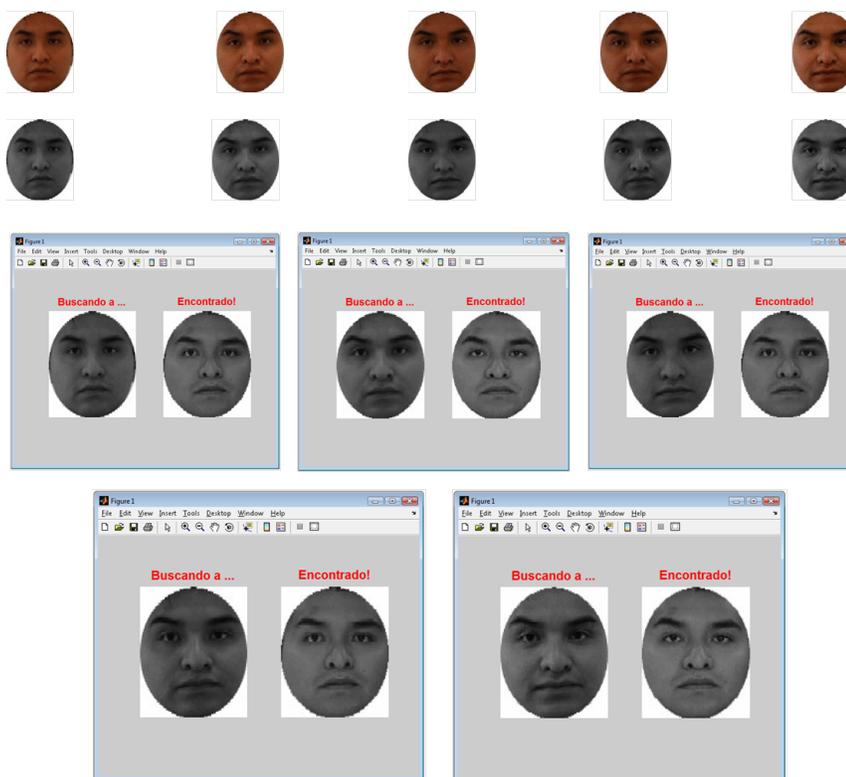


Figura 6.25: Resultado de la prueba, identificación de las cinco imágenes de prueba con variación en escala en la resolución más baja, cinco aciertos.

una resolución más baja logran identificarse, bajo la condición de aumentar el número de fotografías a diez casos que permitieron aumentar los límites, mínimo y máximo de reconocimiento de la clase.

Esto nos indica que se pueden hacer ajustes de manera que, el grupo de diez imágenes se conforme con una imagen de la Fig. 6.26 y con nueve de la Fig. 6.22. La primera se considera como la imagen de referencia o la que aparecerá como resultado de la identificación y las otras nueve complementarán el espacio de eigenfaces de esta clase. La función de este conjunto de diez imágenes es aumentar el umbral mínimo y máximo de reconocimiento por parte de la clase.

Con el nuevo conjunto de rostros, se realizan las mismas pruebas de localización y extracción de características (Fig. 6.27). Los parámetros de las elipses extraídas se reportan en la Tabla 6.4. Enseguida se cambian a escala de grises y se normalizan (Figs. 6.28 y 6.29).

El proceso complementario, agregar el grupo de 9 imágenes al sistema (recortando las 400 fotografías, 10 por persona del grupo de 40), así como la implementación de algunas etapas del diagrama a bloques del sistema automático de reconocimiento de rostros propuesto en el capítulo anterior (Fig. 5.1), implica una serie de ajustes necesarios en la adquisición de imágenes (seleccionar el mejor

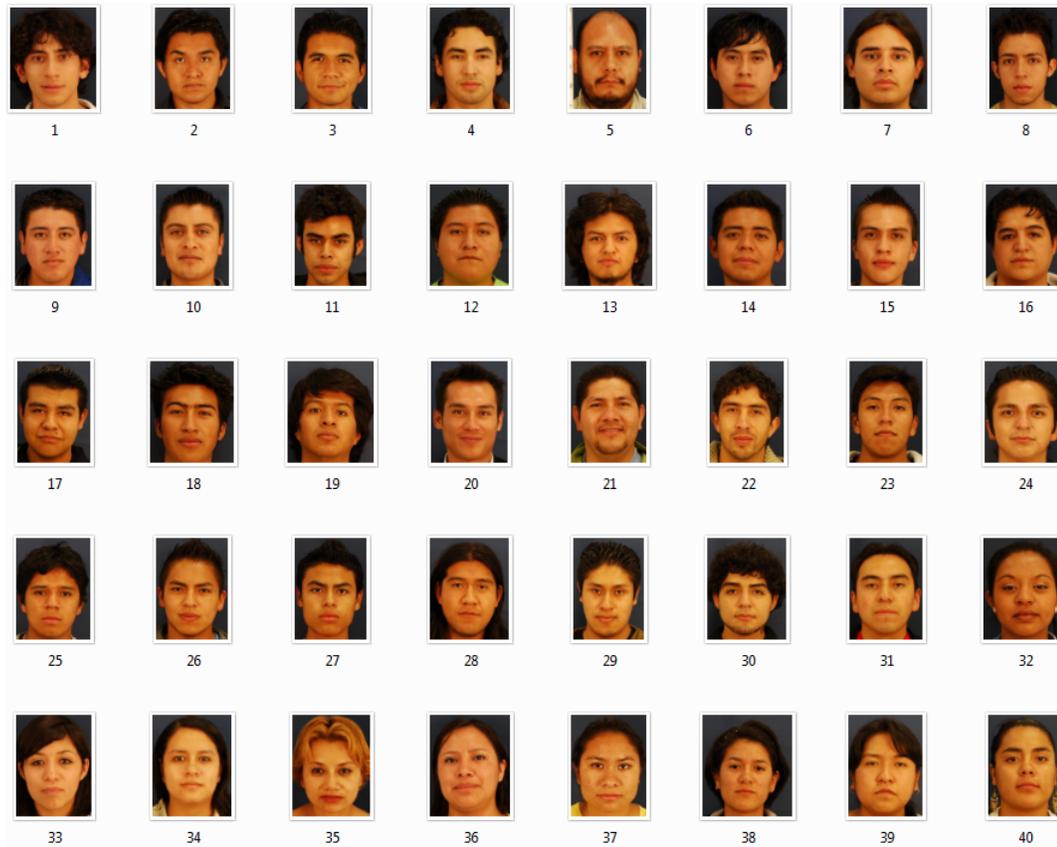


Figura 6.26: Nuevo conjunto de imágenes para integrar la base de datos.

grupo que sea función de la distancia cambios de escala o de pequeños cambios en la rotación de la cara) y en particular en el de la extracción de la elipse.

Finalmente, con base en los experimentos realizados, se puede afirmar que la ruta de solución está planteada y que es tiempo y disposición de la comunidad académica (que faciliten la adquisición de imágenes diversas), lo que permitirá mejorar el sistema.



Figura 6.27: Extracción de la región de la elipse (imágenes en color) del nuevo conjunto de rostros.



Figura 6.28: Conversión a niveles de grises de la región de la elipse del nuevo conjunto de rostros.



Figura 6.29: Imágenes normalizadas en resolución y suavizado con TV del nuevo conjunto de rostros.

Capítulo 7

CONCLUSIONES Y PERSPECTIVAS

7.1. Conclusiones

El reconocimiento de rostros con variación en escala consiste en la comparación de una imagen de prueba, con cambios en su resolución, contra una galería o base de datos de imágenes previamente definida en el sistema computacional de reconocimiento. Si el reconocimiento es positivo se tiene una identificación. Esto significa que la imagen de prueba si corresponde a la persona que forma la base de datos.

La creación de la base de datos es un proceso que consiste en: la segmentación de la región en forma de elipse del conjunto de fotografías que la conformarán; su normalización en resolución, iluminación y suavizado con TV; cálculo de eigenfaces y generación de la pirámide de imágenes. Para la segmentación es necesario localizar un rostro en la imagen y aplicar extracción de características. Es decir, localizar el mapa de piel o zona que corresponde a la cara en la fotografía, a continuación extraer la región de los ojos en ese mapa de piel, localizando el mapa de ojos. Utilizar estas regiones para aplicar la fórmula correspondiente a la elipse como lo indica la geometría analítica indicada en el artículo correspondiente. Una vez hecha la extracción, cada fotografía pasa por el proceso de normalización. A continuación se aplica la técnica de análisis de componentes principales al conjunto de imágenes para obtener el espacio de eigenfaces los cuales reducen la dimensionalidad del conjunto de datos que representan el total de fotografías de manera que se genera un espacio de rostros contra el cual se compararan las imágenes de prueba en la fase de reconocimiento. Para que la fase de reconocimiento sea capaz de identificar rostros con variación en escala es necesario generar la pirámide de imágenes que consiste en reducir la resolución de la base de datos inicial o de base como lo indica el proceso multiresolución para los niveles 8 al 4. Los resultados muestran que

las fases de reconocimiento son satisfactorios en las pruebas que corresponden a la identificación de las imágenes que forman parte de la base de datos. Para el caso del reconocimiento de las imágenes de prueba con variación en escala y que no se integraron a la base de datos pero que se refieren a personas que si se integraron en la misma, se resolvió el problema que no permitía ese reconocimiento y finalmente las pruebas demostraron cuál es el camino para mejorar el sistema.

7.2. Perspectivas

Es recomendable complementar este trabajo con las variantes del rostro en diferentes rotaciones sobre su vertical y sobre su horizontal de manera que también se puedan detectar diferentes cambios de perfil.

Adicionalmente, se puede incluir la variante de cambios de iluminación como se indica en [36], donde aparece media cara casi oscura y la otra parte bastante bien iluminada. Esta es una variante que requeriría normalización en iluminación con variación total. En un salón de clase donde la luz incide por uno de los costados, puede generar ejemplos de este tipo de imágenes.

Es posible que el color, en vez de la escala de grises, sea un factor que ayude en el reconocimiento cuando las resoluciones son muy bajas [35]. La alta resolución en la adquisición de las imágenes puede aumentar el número de niveles en la pirámide de las bases de datos.

Lo anterior aumentaría el alcance del sistema de reconocimiento de rostros. Implicaría hacer la adquisición de otro conjunto de 400 imágenes para rotaciones verticales, más otro igual para rotaciones horizontales, más otro para diferentes cambios de iluminación. Estas se integrarían a la base de datos del sistema. También se requeriría otro grupo adicional de imágenes de prueba, con mayores variaciones en su escala, adquiridas con cámaras con mayor capacidad de resolución.

Bibliografía

- [1] Shaogang Gong, Stephen J. McKenna y Alexandra Psarrou. *Dynamic Vision: From Images to Face Recognition*. London, Imperial College Press. 2001.
- [2] Ralph Gross, Jiambo Shi, y Jeff Cohn. *Quo Vadis Face Recognition? - The current state of the art in Face Recognition*. Technical Report, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 25 pages. (2001).
- [3] Ming-Hsuan Yang, David J. Kriegman, y Narendra Ahuja (2002). *Detecting Faces in Images: A Survey*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 1, January.
- [4] K. Lam and H. Yan. *Fast Algorithm for Locating Head Boundaries*. J. Electronic Imaging, vol. 3, no. 4, pp. 351-359, 1994.
- [5] B. Moghaddam and A. Pentland. *Probabilistic Visual Learning for Object Recognition*. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 696-710, July 1997.
- [6] I. Craw, D. Tock, and A. Bennett. *Finding Face Features*. Proc. Second European Conf. Computer Vision, pp. 92-96, 1992.
- [7] M. Turk and A. Pentland. *Eigenfaces for Recognition*. J. Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991.
- [8] A. Samal and P.A. Iyengar. *Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey*. Pattern Recognition, vol. 25, no. 1, pp. 65-77, 1992.
- [9] R. Chellappa, C.L. Wilson, and S. Sirohey. *Human and Machine Recognition of Faces: A Survey*. Proc. IEEE, vol. 83, no. 5, pp. 705-740, 1995.

- [10] W. Zhao, R. Chellapa, P. J. Philips y A. Rosenfeld (2003). *Face Recognition: A Literature Survey*. ACM Computing Survey, vol. 35, no. 4. December 2003, pp. 399-458.
- [11] Farshid Hajati, Karim Faez and Saeed Khoshfetrat Pakazad. *An Efficient Method for Face Localization and Recognition in Color Images*. In the Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Oct. 2006.
- [12] Ruiz-del-Solar, J. Navarrete, P. *Eigenspace-based face recognition: a comparative study of different approaches*. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, Publication Date: Aug. 2005, Volume: 35, Issue: 3, On page(s): 315- 325.
- [13] Gonzalez, Rafael C. & Richard E. Woods. *Digital Image Processing*. USA, Prentice Hall. 2002 second edition.
- [14] Sossa Azuela, Juan Humberto. *Rasgos descriptores para el reconocimiento de objetos*. Instituto Politécnico Nacional. 2006.
- [15] Jähne, Bernd. *Digital Image Processing*. Springer. 2005. 6th edition.
- [16] González, Rafael C., Richar E. Woods & Steven L. Eddins. *Digital Image Processing using MATLAB*. Pearson Education. 2004.
- [17] Theodoris, Segios y Konstantinos Koutrombas. *Pattern recognition*. Elsevier. 3rd edition. 2006.
- [18] Otsu, N. *A thresholding selection method from gray-level histograms*. IEEE Transactions on System, Man and Cybernetics, 9(1): 62-66. 1979.
- [19] L. I. Smith. *A Tutorial on Principal Components Analysis*. 2002.
- [20] J. Shlens. *A Tutorial on Principal Components Analysis*. 2005.
- [21] D. M. Etter. *Engineering Problem Solving with MATLAB*. Prentice Hall. 1993.
- [22] [http://www.pages.drexel.edu/sis26/Eigenface Tutorial.htm](http://www.pages.drexel.edu/sis26/Eigenface%20Tutorial.htm). En esta página Web se encuentra un procedimiento detallado para obtener el espacio de eigenfaces.
- [23] Ganhua Li *et al.* *An Efficient Face Normalization Algorithm Based on Eyes Detection*. Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems October 9 - 15, 2006, Beijing, China.
- [24] Gelfand, I. M. y S. V. Fomin. *Calculus of variations*. Dover publications. 2000.

-
- [25] Strong M., David, Jean-Francois Aujol, y Tony F. Chan. *Scale recognition, regularization parameter selection, and Meyer's G norm in total variation regularization*. Society for Industrial and Applied Mathematics. SIAM Journal on Computing. MMS Volume 5, Issue 1, pp. 273-303. (2004).
- [26] Chan, Tony F. y Jianhong Shen. *Image processing and analysis: variational, PDE, wavelet and stochastic methods*. SIAM. 2005
- [27] L. Rudin, S. Osher, and E. Fatemi. *Nonlinear total variation based noise removal algorithms*. *Phys. D*, 60:259-268, 1992.
- [28] Lionel Moisan. *How to discretize the total variation of an image?* Proceedings of ICIAM'07.
- [29] Sobottka, Karin y Ioannis Pitas. *Extraction of Facial Regions and Features using Color and Shape Information*. Pattern Recognition. 25-29 Aug 1996. Volume: 3, On page(s): 421-425 vol.3.
- [30] Nakamura, Shoichiro. *Análisis numérico y visualización gráfica con MATLAB*. Pearson Educación. 1997.
- [31] White, Robert E. *Computational Mathematics: Methods, and Analysis with MATLAB and MPI*. Chapman & Hall/CRC. 2003.
- [32] Burden, Richard L. y J. Douglas Faires. *Numerical Analysis*. 8th edition. 2005.
- [33] Ferdinando Samaria, Andy Harter. *Parameterisation of a Stochastic Model for Human Face Identification*. Proceedings of 2nd IEEE Workshop on Applications of Computer Vision, Sarasota FL, December 1994. Este artículo presenta un algoritmo de reconocimiento de rostros que utiliza la base de datos de rostros de AT&T. Ver página Web: <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>
- [34] Getreuer, Pascal. TV denoise algorithm. 2007. En la página Web que sigue se encuentra el algoritmo de variación total. <http://www.mathworks.com/matlabcentral/fileexchange/16236>.
- [35] Sinha, Pawen, *et al.* *Face recognition by humans: nineteen results all computer vision researches should know about*. Proceeding of the IEEE vol. 94, No. 11, November 2006.
- [36] Terrence Chen, Wotao Yin, Xiang Sean Zhou, Dorin Comaniciu, and Thomas S. Huang. *Illumination Normalization for Face Recognition and Uneven Background Correction Using Total Variation Based Image Models*. Computer Vision and Pattern Recognition, 2005. CVPR 2005.

IEEE Computer Society Conference on Volume 2, Issue , 20-25 June 2005 Page(s): 532 - 539
vol. 2

Apéndices

Apéndice A

Aplicaciones

Los resultados obtenidos en esta investigación tienen aplicaciones potenciales en: las áreas de registro o identificación de personal para empresas públicas o privadas; reconocimiento de asistencia en instituciones de educación públicas y privadas; entre otras.

Identificación de personal

Cómo identificación de personal, se puede utilizar para registrar el acceso en las entradas a las unidades, departamentos u oficinas de la entidad pública o privada, utilizando únicamente el rostro de la persona, evitando con esto, por ejemplo, colas y pérdida de tiempo en el registro de sus datos, hora de entrada, firma y entrega de gafete contra credencial de identificación. En entidades donde la cantidad de personal es significativa un sistema de esta naturaleza es altamente recomendable.

Reconocimiento de asistencia

En instituciones de educación, por ejemplo de bachillerato o licenciatura, también se puede aplicar este sistema no solo para registrar el acceso, sino también, para registrar de manera automática las asistencias a las clases de las unidades de enseñanza del alumno, las asistencias a biblioteca o las asistencias a algún otro departamento de interés. Esto permitiría a los profesores evitar perder tiempo en este tipo de actividades y para las autoridades escolares representaría una fuente de datos altamente significativa sobre el comportamiento de sus alumnos en estos rubros, facilitando la toma de decisiones.

Apéndice B

Hoja de registro

PROYECTO DE INVESTIGACIÓN	
Localización y reconocimiento de rostros en imágenes monoculares de frente con variación en escala	
Jorge Servín Pérez. Asesores: Dr. Carlos Barrón Romero y Dr. Felipe Monroy Pérez	
México D. F. a ___ de _____ de 2008	
A quien corresponda:	
Por medio de la presente quiero expresar mi intención de participar en el proyecto "Localización y reconocimiento de rostros en imágenes monoculares de frente con variación en escala", cediendo en uso, los datos e imágenes que voluntariamente proporcione, o se me tomen, para los fines académicos y de investigación del mismo.	
Nombre: _____	
Carrera: 1) Ingeniería en Computación 2) Matemáticas Aplicadas	[]
Preparatoria: 1) Pública 2) Privada	[]
Fecha de nacimiento (dd/mm/aaaa): _____	
Lugar de nacimiento (Municipio, Estado): _____	
Colonia: _____	
Sexo: M) Masculino F) Femenino	[]
Color de ojos: O) Oscuro C) Claros	[]
Color de pelo: N) Negro C) Castaño R) Rubio P) Pelirrojo	[]
Altura: _____	
Peso: _____	
Número de hermanos: _____	
Sin más por el momento, me despido de ustedes, quedando a sus órdenes para cualquier aclaración.	
Atentamente	
_____ Nombre y firma	

Figura B.1: Hoja de registro.

Apéndice C

Artículos de Investigación

Artículos presentados en la Primera Semana de la Computación y Matemáticas Aplicadas (SC-MA'08) en la Universidad Autónoma Metropolitana, Unidad Cuajimalpa, del 14-19 de julio de 2008.

C.1. Modelo experimental que verifica la ley de caída libre usando una cámara de video.

C.2. Imagen promedio de un conjunto de rostros de las carreras de Ingeniería en Computación y Matemáticas Aplicadas.

C.1. Modelo experimental que verifica la ley de caída libre usando una cámara de video



Departamento de Matemáticas Aplicadas y Sistemas
Departamento de Tecnologías de la Información

Otorgan el presente

Reconocimiento

A: **Ing. Jorge Servín Pérez**

Por: haber presentado el trabajo “Modelo experimental que verifica la ley de caída libre usando una cámara de video”

(sometido para su publicación en las memorias del evento)
en la 1era Semana de Computación y Matemáticas Aplicadas,
celebrada del 14 al 18 de julio de 2008

Dr. Pedro Pablo González Pérez

Comité Organizador

Figura C.1: Artículo presentado en la primera Semana de la Computación y Matemáticas Aplicadas 2008, UAM Cuajimalpa.

Modelo experimental que verifica la ley de caída libre usando una cámara de video

A. R. Sagaceta-Mejía^a, J. Servín-Pérez^b, C. Barrón-Romero^b y F. Monroy-Pérez^c

^aMatemáticas Aplicadas, UAM Cuajimalpa.

^bDepartamento de Matemáticas Aplicadas y Sistemas, UAM Cuajimalpa.

^cDepartamento de Ciencias Básicas e Ingeniería, UAM Azcapotzalco

En este trabajo se desarrolló un modelado matemático considerando la ley de gravitación como patrón para verificarlo y validarlo experimentalmente. En la primera etapa se capturaron una colección de fotografías mediante una cámara de video profesional para obtener los parámetros del sistema dinámico de caída libre: distancias y tiempos. Se arregló de forma estándar la disposición de los objetos que se arrojaron respecto al plano de la cámara para que con una escuadra de dimensiones conocidas dentro de las fotografías, estas proporcionen longitudes a escala. Este arreglo es conocido como calibración para obtener distancias a escala. Por otro lado la cámara de video profesional con su frecuencia de disparo define el intervalo fijo de la captura de imágenes y esto permite determinar los tiempos del movimiento de caída libre. Como segunda etapa, se escribieron procedimientos en Matlab para filtrar las imágenes y estimar los parámetros dinámicos del sistema. Los resultados obtenidos verifican aceptablemente el conocido modelo Newtoniano de caída libre, es decir, hay concordancias entre los valores experimentales con los valores predichos por el modelo de la velocidad uniformemente acelerada por la constante de gravitación. Consideramos que este trabajo brinda una experiencia educativa completa en donde la conjunción del modelado, el uso de una cámara de video y los programas de Matlab se pueden aplicar para propósitos didácticos en la enseñanza de la Física.

1. INTRODUCCIÓN

La constante g es la aceleración de la gravedad, la cual afecta a todos los cuerpos en el planeta. A continuación buscamos determinar dicha constante experimentalmente a partir de la caída libre de un cuerpo, capturada por medio de una cámara de video, además de comprobar la hipótesis de la independencia de su valor con respecto a la masa del objeto al que afecta. Esta comprobación se realiza en forma manual utilizando Matlab para verificar la posición del objeto en cada fotografía, del conjunto que componen el video que captura la cámara de video del cuerpo experimental en caída libre. Posteriormente se verifican de manera automática, con programación en Matlab, estos datos para confirmar los resultados manuales. Para realizar la verificación automática se aplican técnicas de procesamiento de imágenes y de reconocimiento

de patrones. Una vez que se tienen las posiciones del objeto en cada fotografía se procede a calcular: la distancia, velocidad y la aceleración de la gravedad. Finalmente se realiza una comparación de los datos con la teoría.

2. LA CAÍDA LIBRE DE LOS CUERPOS

El ejemplo más común del movimiento con aceleración (casi) constante es el de un cuerpo que cae hacia la Tierra. Si no hay resistencia del aire, se observa que todos los cuerpos, cualesquiera que sean su tamaño, su peso o su composición, caen con la misma aceleración en la misma región vecina a la superficie terrestre y, si la distancia recorrida no es demasiado grande, la aceleración permanece constante durante la caída. El movimiento ideal en el que se desprecia tanto la resistencia del aire como el pequeño cambio de la aceleración con la altura, se llama

caída libre [1].

La aceleración de un cuerpo que cae libremente se llama aceleración debida a la gravedad y se le denota con el símbolo \mathbf{g} . Cerca de la superficie de la tierra su magnitud es de aproximadamente de 9.78 m/s^2 para la Ciudad de México, y está dirigida hacia el centro de la tierra.

Escojamos un referencial que esté rígidamente fijo a la tierra y tomemos la parte positiva del eje y vertical hacia arriba. La aceleración \mathbf{g} debida a la gravedad será, entonces un vector que apunta hacia abajo (hacia el centro de la tierra), o sea en la dirección negativa del eje y . En este caso se pueden aplicar las ecuaciones de movimiento con aceleración constante:

$$v_y = v_{y_0} + a_y t \quad (1)$$

$$y = v_{y_0} t + \frac{1}{2} a_y t^2 \quad (2)$$

A menudo se escoge el origen de manera tal que $y_0 = 0$ y $a_y = -g$. En este trabajo se asume que g es positivo, porque es un valor que se quiere determinar. No se hacen operaciones con vectores. Por tanto, no es de utilidad el signo negativo para g .

3. SISTEMA DE ADQUISICIÓN DE IMÁGENES

En general una simple fotografía no nos sirve para tomar mediciones, ya que hace falta alguna referencia o medida patrón en la misma, que relacione o asocie a un grupo de píxeles con una unidad de distancia (cm) para poder realizar la medición. Esto se resuelve agregando ese patrón de medida, regla, en la imagen a capturar.

3.1. Modelo del sistema de adquisición

En la (Fig. 1) se muestra el modelo de una cámara fotográfica y el objeto, un triángulo junto con una regla o medida patrón, a capturar en una imagen.

Se observa en la figura anterior una transformación en perspectiva cuyo objeto se encuentra lejos y paralelo al plano de la fotografía. Hay que notar que, no se introduce distorsión significativa ya que se preserva la geometría (esto es, círculos en el plano del objeto dan círculos en el plano fo-

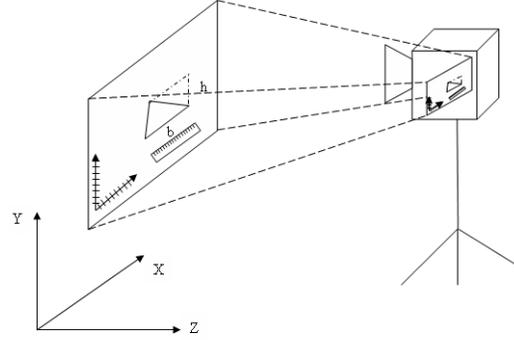


Figura 1. Modelo del sistema de adquisición de imágenes.

tográfico). Es decir, la proyección $P : \mathbf{R}^3 \rightarrow \mathbf{R}^2$ se encuentra dada por la relación:

$$P(x, y, z) = \left(\frac{fx}{d+z}, \frac{fy}{d+z} \right) = (Fx, Fy)$$

Demostración, sea:

$$x^2 + y^2 = r$$

donde

$$\begin{aligned} x^2 &= r \cos \theta \\ y^2 &= r \sen \theta \end{aligned}$$

entonces

$$(r \cos \theta, r \sen \theta) = (fr \cos \theta, fr \sen \theta)$$

Haciendo $fr = R$, tenemos que:

$$(fr \cos \theta, fr \sen \theta) = (R \cos \theta, R \sen \theta)$$

Por lo tanto, permanece la geometría. También se agrega la referencia o medida patrón, regla, la cual permitirá hacer la correspondencia entre píxeles y distancia para observar los cambios de escala correspondiente a la fotografía y el objeto a medir.

3.2. Características de la cámara de video

Frame. Es un fotograma o cuadro, una imagen particular dentro de una sucesión de imágenes que componen una animación. La continua sucesión de estos fotogramas producen a la vista la sensación de movimiento, fenómeno dado por las pequeñas diferencias que hay entre cada uno de ellos.

Shutter. Indica la velocidad de obturación, la velocidad a la que se abre y cierra el obturador de un dispositivo fotográfico y, por tanto, al periodo de tiempo durante el cual está abierto éste. Normalmente se encuentran configuradas a 30 frames/s, es decir, obtendremos 30 fotografías por cada segundo que pase.

Velocidades rápidas. Superiores a 1/60 segundos; el obturador permanece abierto muy poco tiempo dejando pasar menos luz hacia el elemento fotosensible. Con ellas se consigue congelar el movimiento y resaltar el dinamismo de los objetos en movimiento.

Velocidad lentas. Inferiores a 1/60 segundos; el obturador permanece abierto más tiempo dejando pasar más luz. Con ellas se consiguen imágenes borrosas. En la (Fig. 2a.) la flecha indica el objeto (canica). En la (Fig. 2b.) la flecha apunta a una barra borrosa que representa al objeto en caída, a su lado izquierdo se muestra la sombra que genera la iluminación. En estas velocidades es recomendable usar un trípode para evitar que la fotografía salga borrosa.

El fenómeno de caída libre sucede en cuestión de fracción de segundo, es decir, casi imperceptible al ojo humano. Para una cámara profesional es necesario configurarla con la mayor velocidad del shutter, a 1/2000. Hay que mantener una buena iluminación ya que por la gran velocidad, el shutter deja pasar menos luz al elemento fotosensible. También, hay que evitar que la iluminación genere sombras que introduzcan ruido en el objeto que cae.

4. PROCESAMIENTO AUTOMÁTICO DE IMÁGENES

Para poder obtener los parámetros físicos, distancia y tiempo, de imágenes en video que captura el movimiento de un objeto en caída li-

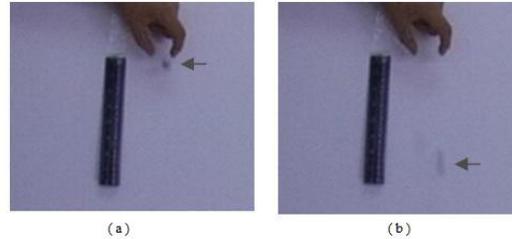


Figura 2. (a) Objeto; (b) Objeto borroso.

bre se requieren cuatro etapas de procesamiento: adquisición de datos, filtrado, segmentación y evaluación.

El procesamiento digital de imágenes se realiza en las primeras dos etapas, adquisición de datos y filtrado. En la tercera etapa, se lleva a cabo el reconocimiento de patrones a través de la segmentación. En la última etapa se evalúa la distancia y el tiempo a la que cae el objeto.

4.1. Procesamiento digital de imágenes

Una *imagen* se define [2] como una función bidimensional $f(x, y)$, donde x y y son coordenadas espaciales (plano) y la amplitud de f en cualquier par de coordenadas (x, y) es llamado la *intensidad* o *nivel de gris* de la imagen en ese punto.

Si los valores de los niveles de gris son cantidades discretas (enteros) entonces a la imagen se le llama *imagen digital*. Esta imagen digital se compone de elementos llamados *píxeles* (*picture elements*).

El campo del *procesamiento digital de imágenes* se refiere al procesamiento de imágenes digitales por medio de una computadora.

En la etapa de *adquisición* de datos se emplea una cámara de video profesional la cual captura el movimiento en caída libre del objeto en una secuencia de *frames* o cuadros (fotografías). Esta cámara genera un archivo tipo videoclip, por ejemplo "Caída libre - clip 001.avi". Para poder obtener los parámetros físicos de interés es nece-

sario descomponer el video en frames o fotografías individuales, proceso que se realiza a través de software profesional para creación de videos. Las etapas de filtrado, segmentación y evaluación se procesan con programación en Matlab.

Cada fotografía que se obtiene en la etapa anterior, es una imagen digital en color la cual se compone de tres matrices de tamaño $m \times n$ para los colores rojo, verde y azul respectivamente. Esta imagen en color se puede convertir en una imagen en escala de grises (una sola matriz) [3] donde la intensidad de cada píxel va del negro (0) al blanco (255). El tratamiento de las imágenes se facilita en la escala de grises (0-255).

Una vez convertida la imagen a escala de grises, estas tienen que pasar por una etapa de filtrado (mascaras de 5×5 píxeles) a fin de que se elimine el ruido introducido por fenómenos aleatorios asociados a las características propias de la cámara e iluminación de la escena, entre otros.

4.2. Reconocimiento de patrones

El reconocimiento de patrones, o reconocimiento de formas, se define [4] como la disciplina científica cuya meta es la clasificación de objetos en un número de categorías o clases.

En una fotografía encontramos una gran cantidad de objetos. Para el problema que nos ocupa el objeto de interés es aquel que corresponde a la clase: objeto que cae. Al proceso que permite extraer la región u objeto de interés en una imagen se le denomina *segmentación*. El parámetro físico de distancia se obtiene a través de la posición en la cual se encuentra el objeto que cae. La segmentación permite discriminar el objeto que cae de entre los otros objetos.

Para segmentar el objeto que cae es necesario aplicar la técnica de *diferencia matricial* entre pares de imágenes consecutivas y posteriormente la técnica de *binarización*. La primera técnica elimina todos los objetos que no cambian de una imagen a la otra, dejando un color negro como fondo y colores en tonos de grises para el objeto que cae. La segunda técnica consiste en convertir la imagen que resulta de la diferencia matricial a solo dos colores blanco y negro, donde el color blanco corresponde al conjunto de píxeles que conforman el objeto que cae y el negro al

fondo de la imagen. Esto se hace para aislar completamente el objeto que cae del fondo de la imagen. La aplicación de filtros de suavizado mejoran la segmentación.

Finalmente la etapa de evaluación consiste en determinar el *centro de masa* que corresponde al conjunto de píxeles en color blanco. Este centro de masa corresponde a la posición (x, y) del objeto en la fotografía. A partir de esta posición se puede calcular el parámetro físico de distancia. El parámetro físico de tiempo lo da el intervalo entre pares de imágenes consecutivas, parámetro asociado al tiempo en que se captura cada frame por la cámara de video ($1/30=0.033\dots$).

A continuación se ilustra un algoritmo para determinar los parámetros físicos de un objeto en caída libre.

1. Leer par de imágenes consecutivas.
2. Convertir imágenes a escala de grises.
3. Filtrar las imágenes.
4. Segmentar, obtener la diferencia matricial entre pares de imágenes consecutivas, binarizar la diferencia y filtrar.
5. En la imagen diferencia calcular el centro de masa (x, y) del objeto.
6. Repetir los pasos 1 a 5 para el resto de las imágenes.

5. DESCRIPCIÓN EXPERIMENTAL

1. Instalar la cámara de video paralela al plano donde se realizara el experimento, con una configuración de $1/30$ frames y $1/2000$ para la velocidad del shutter, o la mayor posible.
2. Verificar que el plano se encuentre totalmente blanco (Fig. 3.), si no es así, adecuarlo para evitar errores de visión en la medición. Instalar la iluminación, tratando de evitar que se generen sombras e introduzcan ruido en la imagen. Ubicar la referencia, perpendicular al piso, para garantizar que se encuentra totalmente vertical, es decir, sin ningún ángulo de inclinación.

3. Seleccionar el objeto, el cual no deberá cambiar, ya que se realizarán varios experimentos para asegurar la validez de los resultados.
4. Grabar el fenómeno mínimo 3 veces, para garantizar los resultados bajo las mismas condiciones.
5. Una vez obtenido el video correspondiente utilizar software editor de video, para descomponer el video en fotografías por segundo.
6. A continuación seleccionar las fotografías donde se encuentra el fenómeno en caída libre, se omiten aquellas donde no hubo movimiento, así como, de las que corresponden al rebote del objeto contra el piso.
7. A través de software matemático como Matlab, hacer la medición manual de cada una de las posiciones del objeto en las fotografías. Confirmar estas mediciones de manera automática aplicando procesamiento digital de imágenes y reconocimiento de patrones a través de programación en Matlab.
8. Determinar la correspondencia pixel por cm (escala), utilizando matlab.
9. Determinar los tiempos del experimento a partir del número de frames asociados a un segundo ($1/30=0.033$ segundos/frame).
10. Determinar la distancia a partir de la posición del objeto en el frame:

$$d = \sqrt{x^2 + y^2}$$

11. Determinar la velocidad, para dos frames consecutivos, a partir de la derivada de la distancia con respecto al tiempo:

$$v = \frac{dd}{dt} = \frac{(d_2 - d_1)}{(t_2 - t_1)}$$

12. Calcular la aceleración a través de la derivada de la velocidad con respecto al



Figura 3. Set experimental.

tiempo:

$$a = \frac{dv}{dt} = \frac{d^2d}{dt^2} = \frac{(dd_2/dt - dd_1/dt)}{(t_2 - t_1)}$$

13. Obtener el promedio de las segundas derivadas para determinar la aproximación a la gravedad
14. Finalmente, con los datos obtenidos verificar si los objetos caen con la misma aceleración.

6. RESULTADOS

En la (Fig. 3.) se muestra el set empleado para la adquisición de imágenes correspondientes a los objetos sujetos a experimentación.

Los experimentos se realizaron para dos objetos distintos, una moneda de dos pesos mexicanos y una canica semitransparente.

En la Tabla 1 se muestran los resultados para la caída libre de la moneda. Por ejemplo, para el frame 24 (primera columna) tenemos que el objeto se encontraba en la posición (128, 24) (segunda y tercera columna), el tiempo transcurrido del frame 23 al 24 es de 0.03 segundos (columna 4), la diferencia entre el frame 24 y el frame 23

Tabla 1
Resultados experimentales y cálculo de la gravedad (p=píxeles).

Frame	x (p)	y (p)	t (s)	x (p)	y (p)	d (p)	$v = dd/dt$ (p)	$a = d^2d/dt^2$ (p/s ²)	cm/s ²	m/s ²
23	128.44	18.64	0.00	0.00	0.00	0.00	161.26	2377.96	1185.59	11.86
24	128.00	24.00	0.03	-0.44	5.36	5.38	240.53	1789.71	892.31	8.92
25	127.52	32.01	0.07	-0.92	13.36	13.39	300.18	1707.04	851.08	8.51
26	127.03	42.00	0.10	-1.41	23.36	23.40	357.09	2459.97	1226.48	12.26
27	126.40	53.89	0.13	-2.03	35.24	35.30	439.08	1746.88	870.95	8.71
28	125.49	68.50	0.17	-2.94	49.85	49.94	497.31	2156.54	1075.20	10.75
29	125.02	85.07	0.20	-3.42	66.43	66.52	569.20	1933.50	964.00	9.64
30	124.03	104.02	0.23	-4.40	85.38	85.49	633.65	1429.88	712.90	7.13
31	123.44	125.14	0.27	-4.99	106.49	106.61	681.31	1769.89	882.42	8.82
32	122.65	147.84	0.30	-5.78	129.19	129.32	740.31	1224.10	610.31	6.10
33	121.86	172.50	0.33	-6.57	153.86	154.00	781.11			
34	121.01	198.53	0.37	-7.43	179.88	180.03				
							Promedio	1930.15	962.33	9.62

es de $128-128.44=-0.44$ (columna 5) para x y de $24-18.64=5.36$ para y (columna 6), la distancia del frame 23 al 24 es de

$$\sqrt{x^2 + y^2} = \sqrt{(-0.44)^2 + (5.36)^2} = 5.38$$

(columna 7), la velocidad será de

$$\frac{13.3929-5.37539}{0.0667-0.0333} = 240.5271$$

(columna 8), la aceleración será de

$$240.5271-161.2619=1789.7136 \approx 1789.71$$

(columna 9), la conversión de píxeles a centímetros (Tabla 2) será de

$$1789.71357/2.0057=892.30602$$

(columna 10), este último valor convertido a m/s^2 será de

$$892.3061(cm/s^2)[(m)/100(cm)] = 8.92(m/s^2)$$

(columna 11). Finalmente, al promediar todos los valores de la columna 11 obtendremos el valor de la gravedad buscado, $9.62 m/s^2$, que es muy

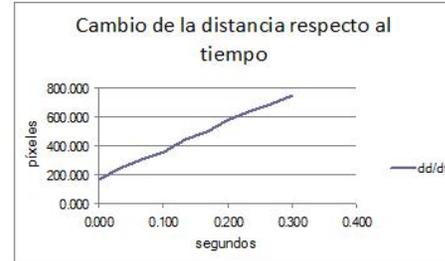


Figura 4. Gráfica distancia vs tiempo.

aproximado al valor para la Ciudad de México, $9.78 m/s^2$.

En la gráfica de la (Fig. 4) podemos observar que al graficar la distancia d con respecto al tiempo t , obtenemos una línea casi recta que corresponde a $v = dd/dt = (d^2d/dt)t = gt$, donde g es la pendiente.

En la Tabla 3 se muestran los resultados obtenidos para g en seis experimentos. Los tres primeros corresponden al objeto canica y los 3 siguientes corresponden al objeto moneda.

En la Tabla 4 se muestran las diferencias en-

Tabla 2

Conversión de datos, píxeles por cm.

pixel/cm
2.005717221

Tabla 3

Valores experimentales de la gravedad para objetos en caída libre.

Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6
9.68	9.59	9.48	9.27	9.75	9.48

tre los datos manuales y los datos que arroja de manera automática el programa en Matlab. Con base en estos resultados se puede afirmar que, los datos obtenidos automáticamente confirman los cálculos manuales con un error aproximado de 1 pixel.

En la (Fig. 5.), podemos ver las etapas de procesamiento para obtener de forma automática el parámetro físico de distancia para cada posición del objeto en el frame correspondiente. Dos frames consecutivos, en color (Fig. 5a, b.), son convertidos a escala de grises (Fig. 5c, d.), estas imágenes pasan por un filtro para atenuar el ruido, a continuación se hace la diferencia matricial entre la imagen siguiente menos la imagen anterior para segmentar el objeto, esta "imagen diferencia" produce una imagen con fondo negro y con el objeto en nivel de gris, ubicado en la posición que se indica en la imagen anterior (Fig. 5c.). A continuación se binariza esta diferencia, dejando solo píxeles blancos y negros. Con otro proceso de filtrado se elimina cualquier ruido adicional de manera que solo queden en la imagen los píxeles que corresponden al objeto que se quiere segmentar. En la (Fig. 5f.) se muestra el negativo, de la (Fig. 5e.) sin ruido alguno. Finalmente, para obtener la posición del objeto en la imagen se calcula el centro de masa de ese conjunto de píxeles en color negro (Fig. 5f.), promediando las posiciones x e y de cada pixel.

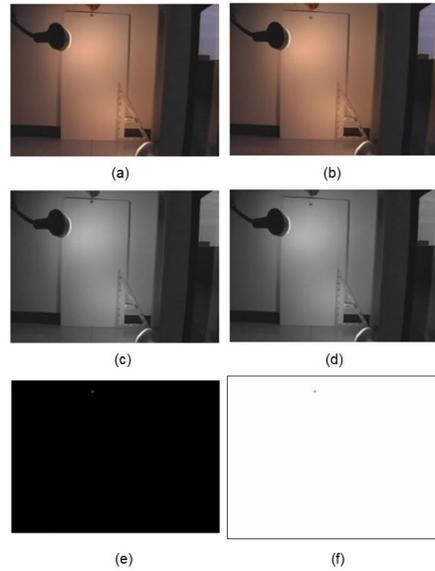


Figura 5. (a), (b) Frames consecutivos en color; (c), (d) Frames en escala de grises; (e) Diferencia (d-c); (f) Negativo de (e).

Tabla 4
Comparación de resultados experimental vs automático.

Frame	Medición manual		Medición automática		Diferencia	
	x	y	x	y	x	y
22	128.511	14.8448	128.500	15.500	0.011	-0.655
23	128.435	18.6449	128.400	18.800	0.035	-0.155
24	127.998	24.0025	127.714	24.000	0.284	0.003
25	127.519	32.0065	127.500	32.000	0.019	0.003
26	127.027	42.0016	127.000	41.000	0.027	1.002
27	126.401	53.8882	126.400	54.200	0.001	-0.312
28	125.492	68.4962	125.500	68.000	-0.008	0.496
29	125.020	85.0724	124.833	85.000	0.187	0.072
30	124.033	104.020	124.286	104.000	-0.253	0.020
31	123.441	125.138	123.182	124.818	-0.259	0.320
32	122.651	147.836	122.000	147.000	0.651	0.836
33	121.862	172.502	120.800	171.800	1.062	0.702
34	121.005	198.526	119.952	197.810	1.053	0.716

7. CONCLUSIONES

Aceptamos las leyes de la Física y conocemos los modelos matemáticos pero nuestro propósito es que a partir de la obtención de medidas, verificamos esa teoría con la realidad. Es posible apreciar a través de las Tablas 1 y 2, los valores tan cercanos a la gravedad, es cierto que no es muy exacto, ya que diferimos en decimales, pero dadas las condiciones donde se realizó el experimento, nuestro error es muy aceptable. El cálculo automático de los datos ayudó a disminuir considerablemente los tiempos, además de que siempre es sistemático el programa y elimina ciertos errores acumulados en la obtención manual de los datos. Además es necesario tomar en cuenta que la experimentación introduce errores y no es posible tener datos exactos (siempre realizamos pequeñas aproximaciones), las cuales afectan nuestros resultados. Las mediciones son muy importantes y a pesar de los errores mencionados la comparación de resultados fueron congruentes entre la teoría y la realidad. Luego Galileo no se equivocó.

REFERENCIAS

1. Resnick, Robert y David Halliday. *Física: Parte 1*. CECSA.
2. González, Rafael C. y Richard E. Woods. *Digital image processing*. Prentice hall. Segunda edición. 2002.
3. Ver ayuda en Matlab (R2006b): "Converting an 8-Bit RGB Image to Grayscale".
4. Theodoridis, Sergios y Konstantinos Koutrombas. *Pattern recognition*. Elsevier. Third edition. 2006.

**C.2. Imagen promedio de un conjunto de rostros de las carreras de
Ingeniería en Computación y Matemáticas Aplicadas**

UNIVERSIDAD AUTÓNOMA METROPOLITANA
UNIDAD CUAJIMALPA

SCMA'2008

Departamento de Matemáticas Aplicadas y Sistemas
Departamento de Tecnologías de la Información

Otorgan el presente

Reconocimiento

A: Ing. Jorge Servín Pérez

Por: haber presentado el trabajo “Imagen promedio de un conjunto de rostros de las carreras de Ingeniería en Computación y Matemáticas Aplicadas”
(sometido para su publicación en las memorias del evento)
en la 1era Semana de Computación y Matemáticas Aplicadas,
celebrada del 14 al 18 de julio de 2008

Dr. Pedro Pablo González Pérez
Comité Organizador

Figura C.2: Artículo presentado en la primera Semana de la Computación y Matemáticas Aplicadas 2008, UAM Cuajimalpa.

Imagen promedio de un conjunto de rostros de las carreras de Ingeniería en Computación y Matemáticas Aplicadas

A. Soto-Girón^a, V. García-Martínez^a, J. Servín-Pérez^b, C. Barrón-Romero^b y F. Monroy-Pérez^c

^aIngeniería en Computación, UAM Cuajimalpa.

^bDepartamento de Matemáticas Aplicadas y Sistemas, UAM Cuajimalpa.

^cDepartamento de Ciencias Básicas e Ingeniería, UAM Azcapotzalco.

En este trabajo se aplica la técnica Análisis de Componentes Principales para obtener la imagen promedio y los rasgos principales de un conjunto de rostros de alumnos de las carreras de Ingeniería en Computación y Matemáticas Aplicadas de la UAM Cuajimalpa. Este método se puede aplicar en la identificación de una persona con propósitos de seguridad, salud o para algún tipo de registro o servicio. Para realizar este trabajo se construyó un set de fotografía como sistema de adquisición de imágenes. Para el tratamiento de las fotografías se aplicaron técnicas de procesamiento de imágenes como filtrado, para mejorar la calidad de las mismas disminuyendo el ruido asociado, así como la técnica Análisis de Componentes Principales que permite extraer los rasgos de un conjunto de datos (imágenes faciales) al reducir la multidimensionalidad de estos a una dimensión más baja que represente los rasgos más importantes (nariz aguileña u otro) del conjunto de datos. La motivación de este trabajo no sólo es apoyar a un trabajo de mayor alcance, como es el reconocimiento de rostros con variación en escala, sino el de promover el interés de los alumnos de la carrera de Ingeniería en Computación por las aplicaciones típicas de las áreas de Procesamiento Digital de Imágenes, Reconocimiento de Patrones y Visión por Computadora.

1. INTRODUCCIÓN

En el presente se describe la técnica matemática de Análisis de Componentes Principales y cómo se aplica al cálculo del rostro promedio, así como, de los rasgos principales de estos (eigenfaces) correspondientes a un conjunto de estudiantes de las carreras de Ingeniería en Computación y Matemáticas Aplicadas de la UAM Cuajimalpa. Además, se describe el procedimiento computacional para obtener estos resultados.

2. ANÁLISIS DE COMPONENTES PRINCIPALES

La técnica Análisis de Componentes Principales (PCA, *Principal Components Analysis*) [1, 2] es una técnica matemática útil que ha encontrado aplicación en los campos del reconocimiento de rostros y en la compresión de imágenes ya que es una técnica común para encontrar patrones

o rasgos principales finitos en problemas de dimensión infinita.

La técnica consiste de los siguientes pasos: 1. Recopilar el conjunto de datos; 2. Obtener el conjunto promedio de los datos; 3. Calcular la matriz de covarianza; 4. Seleccionar los componentes, formar un vector característico y calcular los eigenvectores y eigenvalores de la matriz de covarianza.

2.1. Conjunto de datos

Sea n un conjunto de imágenes de tamaño $M \times N$, donde M es el número de píxeles por renglón y N es el número de píxeles por columna. Entonces, un vector:

$$\mathbf{x} = x_1, x_2, \dots, x_n \quad (1)$$

Se compone de n píxeles, donde cada elemento es un píxel localizado en la misma posición de cada imagen. Esto implica que se puede tener una matriz \mathbf{X} de vectores \mathbf{x}_k tales que:

$$\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k \quad (2)$$

donde $k=1,2,\dots,MN$ y

$$\mathbf{x}_k = x_1, x_2, \dots, x_n \quad (3)$$

Esta matriz de vectores representa al conjunto de datos.

2.2. Vector promedio

El vector promedio del conjunto de datos estará dado por la relación:

$$\mathbf{m}_X = \frac{1}{MN} \sum_{k=1}^{MN} \mathbf{x}_k \quad (4)$$

2.3. Matriz de covarianza

La matriz de covarianza \mathbf{C}_X es una medida de la dispersión entre diferentes variables (más de una dimensión). Para el caso de un vector \mathbf{x}_k y el vector promedio \mathbf{m}_X tenemos que la matriz de covarianza está dada por:

$$\mathbf{C}_X = \frac{1}{MN} \sum_{k=1}^{MN} (\mathbf{x}_k - \mathbf{m}_X)(\mathbf{x}_k - \mathbf{m}_X)^T \quad (5)$$

2.4. Componentes principales

La transformación en componentes principales [3] está dado por:

$$\mathbf{Y} = \mathbf{A}(\mathbf{x}_k - \mathbf{m}_X) \quad (6)$$

donde \mathbf{A} representa el vector característico o vector propio (*eigenvector*, del alemán *eigen* propio). Los renglones de \mathbf{A} representan los eigenvectores de \mathbf{C}_X . Los eigenvalores son los elementos a lo largo de la diagonal de \mathbf{C}_Y , donde:

$$\mathbf{C}_Y = \mathbf{A}\mathbf{C}_X\mathbf{A}^T \quad (7)$$

Los eigenvectores [4] son vectores unitarios, es decir su longitud es 1 y forman un conjunto ortonormal de vectores en la norma inducida por la matriz de covarianza (son perpendiculares entre ellos respecto a \mathbf{C}_X) y proveen información acerca de los patrones en los datos. Seleccionando los eigenvalores de mayor valor se consigue la reducción de dimensionalidad y la compresión de datos, en el sentido de que con estos vectores se cubre el espacio de los datos. Hay reducción de dimensionalidad en la medida en que la dispersión

de los datos se alinean a un eigenvector en particular de manera que se pueden omitir aquellos eigenvectores en los cuales no sea relevante tal alineación, es decir, su eigenvalor sea pequeño. Hay compresión de datos en la medida en que se puede seleccionar sólo aquellos eigenvalores, que sean más representativos en peso (de mayor valor) de manera que es posible recuperar la información original sin gran pérdida de datos, o sea, con error cuadrático medio pequeño. Esto es, sea:

$$\mathbf{X} = \mathbf{A}^T \mathbf{Y} + \mathbf{m}_X \quad (8)$$

el conjunto de datos original. Entonces, para un conjunto q reducido de eigenvectores tenemos que \mathbf{A} se hace una matriz $q \times n$, \mathbf{A}_q , de manera que la reconstrucción estará dada por:

$$\mathbf{X} = \mathbf{A}_q^T \mathbf{Y} + \mathbf{m}_X \quad (9)$$

El error cuadrático medio estará dado por:

$$e_{cm} = \sum_{j=1}^n \lambda_j - \sum_{j=1}^q \lambda_j = \sum_{j=q+1}^n \lambda_j \quad (10)$$

Si $q=n$ el error es cero.

3. LA TÉCNICA EIGENFACE

La técnica eigenface [5,6] es el PCA aplicado en imágenes faciales (rostros) sobre un espacio representado por un conjunto de rostros grande. Los característicos significativos son conocidos como *eigenfaces* porque son los eigenvectores (componentes principales) del conjunto de rostros; que asemejan, aunque no necesariamente, a características tales como los ojos, oídos o nariz. La operación de proyección caracteriza a un rostro individual por la suma de pesos de los eigenfaces característicos, de manera que para reconocer un rostro particular solo es necesario comparar estos pesos contra aquellos de los rostros conocidos. En términos matemáticos se requiere encontrar los componentes principales de la distribución de rostros, o los eigenvectores de la matriz de covarianza del conjunto de imágenes faciales, tratando una imagen como un punto (vector) en un espacio dimensional infinito. Posteriormente, se ordenan los eigenvectores de acuerdo

a una cantidad relativa a la variación entre las imágenes faciales, seleccionando un número finito de estos.

Estos eigenvectores se pueden considerar como un conjunto de características que juntos caracterizan la variación entre las imágenes faciales. Podemos ver, a cada uno de los eigenvectores, como un pequeño espectro o "fantasma", del rostro original, al cual se le llama *eigenface*. Cada rostro individual se puede representar en términos de una combinación lineal de los eigenfaces. Por lo que un rostro individual se puede aproximar empleando solo los "mejores" eigenfaces, aquellos que tienen los eigenvalores más largos o pesados, y que más contribuyen en la variación del conjunto de imágenes faciales. Los mejores M eigenfaces se extienden sobre un subespacio M -dimensional, *espacio facial*, de todas las imágenes posibles (se tiene la reducción a M dimensiones del espacio de rostros de dimensión infinita).

3.1. Adquisición de imágenes

Sea F un conjunto de M imágenes o fotografías de tamaño $m \times n$ donde:

$$\mathbf{F} = \mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_M \quad (11)$$

Las operaciones de cálculo se reducen si cada imagen se reordena de manera que el tamaño o resolución de las mismas se lleven a la resolución $(m \times n) \times 1 = N \times 1$. De esta forma se obtendrá:

$$\mathbf{\Gamma} = \tau_1, \tau_2, \dots, \tau_M \quad (12)$$

que es un conjunto de M imágenes de resolución $N \times 1$.

3.2. Cálculo de los eigenfaces

El rostro promedio del conjunto de fotografías estará dado por:

$$\mathbf{\Psi} = \frac{1}{M} \sum_{k=1}^M \mathbf{\Gamma}_k \quad (13)$$

y sea

$$\mathbf{\Phi}_i = \mathbf{\Gamma}_i - \mathbf{\Psi} \quad (14)$$

la diferencia entre la i -ésima imagen o fotografía y el rostro promedio. Entonces la matriz de co-

varianza estará dada por la relación:

$$\mathbf{C} = \frac{1}{M} \sum_{k=1}^M \mathbf{\Phi}_k \mathbf{\Phi}_k^T = \mathbf{A} \mathbf{A}^T \quad (15)$$

donde

$$\mathbf{A} = [\mathbf{\Phi}_1 \mathbf{\Phi}_2 \dots \mathbf{\Phi}_M] \quad (16)$$

es un vector de tamaño $N \times M$. A partir de la matriz de covarianza se pueden obtener los eigenfaces (eigenvectores) y los eigenvalores. Aplicando:

$$\mathbf{A} \mathbf{A}^T \mathbf{v}_i = \mu_i \mathbf{v}_i \quad (17)$$

donde \mathbf{v}_i es el vector de eigenfaces y μ_i el conjunto de eigenvalores. Notemos que la resolución de esta operación estará dada por:

$$\mathbf{A} \mathbf{A}^T \Rightarrow (N \times M)(N \times M)^T = (N \times N) \quad (18)$$

Para hacer este cálculo es necesario hacer computacionales las operaciones reduciendo la resolución de las mismas. Esto es, si las imágenes son de tamaño $m \times n$ entonces para una imagen con $m=376$ y $n=292$ píxeles tendremos que la resolución será de $m \times n = 376 \times 292 = 109,792$ píxeles. Como $N = m \times n$, entonces $N \times N = 12,054,283,264$ que es un valor muy alto para calcular. Aplicando propiedades del álgebra lineal se puede reducir el computo de las operaciones a un valor dado por $M \times M$. Si M corresponde a seis fotografías entonces $M \times M = 6 \times 6 = 36$. Esto es:

$$\mathbf{A}^T \mathbf{A} \Rightarrow (N \times M)^T (N \times M) = (M \times M) \quad (19)$$

De esta forma el procedimiento para calcular los eigenvectores y eigenvalores será el siguiente. Consideremos la ecuación:

$$\mathbf{A}^T \mathbf{A} \mathbf{u}_i = \mu_i \mathbf{u}_i \quad (20)$$

De esta ecuación obtener los eigenvectores \mathbf{u}_i y eigenvalores μ_i . A continuación ajustar el eigenvector \mathbf{u}_i obtenido multiplicándolo por \mathbf{A} ya que:

$$\mathbf{A} \mathbf{A}^T (\mathbf{A} \mathbf{u}_i) = \mu_i (\mathbf{A} \mathbf{u}_i) \quad (21)$$

donde

$$\mathbf{v}_i = \mathbf{A} \mathbf{u}_i \quad (22)$$

nos da el conjunto de eigenfaces buscado.



Figura 1. Sistema de adquisición de imágenes.



Figura 2. Fotografías de los alumnos de la carrera de Ingeniería en Computación.

4. RESULTADOS

En la figura 1 se muestra el set fotográfico empleado para la adquisición de imágenes. Este set fotográfico se compone de los siguientes elementos: dos lámparas reguladores de 500 Watts, dos paneles reflectores, un banquillo, un fondo azul, una cámara profesional NIKON D80, y un trípode profesional. Este set fotográfico se empleó para adquirir el conjunto de fotografías correspondientes a los alumnos de Ingeniería en Computación (Fig. 2) y Matemáticas Aplicadas (Fig. 5). Se empleó MATLAB R2006b para implementar las ecuaciones descritas en este artículo y obtener los resultados gráficos correspondientes. Los resultados muestran un conjunto de $M=6$ fotografías (con resolución de 376×292 píxeles) por carrera. Aplicando la ecuación (13) a este conjunto de fotografías, de manera separada, se obtiene el rostro promedio correspondiente. En la figura 3 se muestra el rostro promedio de los alumnos de Ingeniería en Computación y en la figura 6 se ilustra el rostro promedio de los alumnos de Matemáticas Aplicadas. Finalmente, aplicando la ecuación (22) se obtienen los eigenfaces correspondientes (fig. 4 y Fig. 7), para todos sus eigenvalores.



Figura 3. Rostro promedio de los alumnos de la carrera de Ingeniería en Computación.



Figura 4. Eigenfaces correspondientes a los alumnos de la carrera de Ingeniería en Computación.

5. CONCLUSIONES

Los resultados obtenidos muestran que el rostro promedio de los alumnos de la carrera de Matemáticas Aplicadas tienden hacia un rostro femenino, con cara redonda, labios gruesos y ojos



Figura 5. Fotografías de los alumnos de la carrera de Matemáticas Aplicadas.



Figura 6. Rostro promedio de los alumnos de la carrera de Matemáticas Aplicadas.



Figura 7. Eigenfaces correspondientes a los alumnos de la carrera de Matemáticas Aplicadas

casi unidos (Fig. 5 y Fig. 7). Para el caso de los alumnos de Ingeniería en Computación, los rasgos son más finos, se mantiene el rostro robusto y aparece la región que corresponde a la barba (Fig. 4 y Fig. 6). Finalmente, se puede afirmar que esta herramienta se puede utilizar para obtener rasgos dominantes de un rostro y determinar a que carrera pertenece, o cuál es el origen (estado, país, etc.) del cual proviene un estudiante o persona.

REFERENCIAS

1. L. I. Smith. *A Tutorial on Principal Components Analysis*. 2002.
2. J. Shlens. *A tutorial on Principal Components Analysis*. 2005.
3. R. C. Gonzalez y R. E. Woods. *Digital Image Processing*. Prentice Hall. 2nd ed. 2002.
4. D. M. Etter. *Engineering Problem Solving with MATLAB*. Prentice Hall. 1993.
5. M. Turk and A. Pentland, *Eigenfaces for Recognition*, *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.
6. <http://www.pages.drexel.edu/sis26/EigenfaceTutorial.htm>.

Apéndice D

Códigos de programas en MATLAB

- D.1. Variación total (programa)
- D.2. Extracción de la región elíptica del rostro (programa)
- D.3. Normalización de las regiones elípticas en resolución y suavizado con TV (programa)
- D.4. Cálculo de eigenfaces (programa)
- D.5. Generación de la pirámide de imágenes (programa)
- D.6. Verificar imágenes que conforman las bases de datos (programa)
- D.7. Reconocimiento de rostros, prueba de verificación de cambios de escala

D.1. Variación total (programa)

```

1 function u = tvdenoise(f,lambda,NumSteps,u)
2 %TVDENOISE Total variation image denoising
3 % u = TVDENOISE(f,lambda,NumSteps) denoises the input image f over
4 % NumSteps number of iterations. The larger the parameter lambda, the
5 % stronger the denoising. The output u approximately minimizes the
6 % Rudin-Osher-Fatemi (ROF) denoising model
7 % || f - u ||^2_L^2 + lambda*TV(u)
8 % where TV(u) is the total variation of u.
9 %
10 % TVDENOISE(...,u0) specifies the initial image u0. By default, u0 = f.
11 %
12 % Example: Run TVDENOISE without any inputs for a demo
13 % >> tvdenoise
14
15 % Pascal Getreuer 2007
16 clc
17 if nargin < 4
18     if nargin == 0
19         %%% Demo %%%
20
21         % Generate image
22         [x,y] = meshgrid(linspace(-1,1,180),linspace(-1,1,180));
23         [th,r] = cart2pol(x,y);
24         f = (sqrt(2)-r).^2.*(sin(12*r + 4*th) > 0 | r < 0.05)*255;
25         f = conv2(f,ones(5)/25,'same');
26         f = f(3:5:end,3:5:end);
27         f = f + randn(size(f))*30; % Add noise
28
29         % Denoise the image with lambda=400 and 25 iterations
30         v = tvdenoise(f,400,25);
31
32         % Plots
33         subplot(1,2,1);
34         image(f);
35         title('Noisy Image');
36         axis image; axis off; colormap(gray(256));
37         subplot(1,2,2);
38         image(v);
39         title('Denoised Image');
40         axis image; axis off; colormap(gray(256));
41         shg;
42
43         return;
44     end
45     u = f;
46 elseif any(size(f) ~= size(u))
47     error('u0 must have the same size as f.');
```

```
57 il = [1,1:N2-1];
58 ir = [2:N2,N2];
59 iu = [1,1:N1-1];
60 id = [2:N1,N1];
61
62 for k = 1:NumSteps
63     ul = u(:,il);
64     ur = u(:,ir);
65     uu = u(iu,:);
66     ud = u(id,:);
67     c = a./sqrt(EpsSqr + (ur - u).^2 + (ud - u).^2);
68     cl = c(:,il);
69     cu = c(iu,:);
70     u = (u + dt*f + c.*(ur + ud) + cl.*ul + cu.*uu) ...
71         ./ (1 + dt + 2*c + cl + cu);
72 end
73
74
75
```

D.2. Extracción de la región elíptica del rostro (programa)

```

1 %Localización y Reconocimiento de rostros en imágenes monoculares con
2 %variación en escala
3 %Programa hecho en MATLAB 7.3.0.267 (R2006b) (c)
4
5 %Programa: Experimento, detección de ojos en 50 fotografías
6 %resultados: aciertos 48 (96%), Errores:2(4%)
7
8 %limpiar espacio de trabajo
9 clear all
10 close all
11 clc
12
13
14 D=[];
15
16 for ii=1:1:50 %Iniciar cuenta
17
18 %cargar imagen empezando en ii=1, es decir, desde '1.bmp' a '50.bmp'.
19     archivo=strcat(num2str(ii),'.bmp'); %concatenar nombre de archivo
20     [x,map] = imread(archivo);           %cargar imagen
21     x = imadjust(x,stretchlim(x),[0 1]);%autoajustar la intensidad
22
23     %Visualizar la imagen original 'x' ajustada en intensidad
24     figure, imshow(x),title('Imagen original ajustada');
25
26     fotooriginal=x; %conservar imagen original
27
28 %convertir el espacio RGB de la imagen a escala de grises
29     y=rgb2gray(x);
30     figure, imshow(y),title('y'); %Visualizar la imagen
31     fotoengris=y; %Conservar la imagen en escala de grises
32     yy=y;
33
34 %convertir el espacio RGB de la imagen al espacio YCbCr
35     YCBCR = rgb2ycbcr(x);
36     Y=double(YCBCR(:,:,1)./255); %normalizar 'Y' luminancia
37     figure, imshow(Y),title('Y'); %visualizar la imagen
38     cb=double(YCBCR(:,:,2)); %Extraer la componente 'Cb'
39     cr=double(YCBCR(:,:,3)); %Extraer la componente 'Cr'
40     cb2=(cb.^2)./255; %Eleva al cuadrado la compoene 'Cb' y normalizar
41     cr2=((255-cr).^2)./255; %Eleva al cuadrado la compoene 'Cr' y normalizar
42     %Hacer la diferencia entre la componente Cb de 'YCBCR'
43     %y la componente R de 'x'
44     E=uint8(cb)-x(:,:,1);
45     figure, imshow(E),title('Imagen Cb-x(:,:,1)'); %visualizar la imagen
46
47 %Obtener el mapa de piel
48     %Hacer la diferencia entre Cb de 'YCBCR' y la componente R de 'x'
49     %y la componente 'Y' de 'YCBR'
50     Z=double(cb)-double(x(:,:,1))-Y;
51     level=graythresh(Z); %determinar automaticamente umbral de binarización
52     Z=im2bw(Z,level); %binarizar imagen
53     Z=~Z; %complementar la imagen
54     [L componentes]=bwlabeled(Z); %Etiquetar componentes 8-conectadas
55     %Determinar el tamaño de cada componente
56     s=[];

```

```

57     for k=1:componentes
58         s=[s; size(find(L==k),1)];
59     end
60
61     %extraer la region
62     [maximo indice]=max(s);      %
63     [rows cols]=find(L==indice);
64     BW=zeros(size(L));
65     for k=1:maximo
66         BW(rows(k,1),cols(k,1))=L(rows(k,1),cols(k,1));
67     end
68     BW2 = imfill(BW,'holes');
69     [rows cols]=find(L==indice);
70     figure, imshow(BW2),title('Region');
71
72
73     %extraer la elipse máxima dentro de la región
74     py=uint8(mean(rows))%
75     px=uint8(mean(cols))% estos puntos casi corresponden a un punto en la nariz
76     BW3 = imregionalmax(BW2,8); %la region de la cara es 0
77     figure, imshow(BW3),title('Minimos'),hold on, plot(px,py,'r'),
78
79
80     %crecer una elipse
81     [p q]=size(BW3)
82     ax1=double(px);
83     ax2=double(px);
84     by1=double(py);
85     by2=double(py);
86
87     while BW3(by1,px)==1 && by1>1
88         by1=by1-1;
89     end
90
91     while BW3(by2,px)==1 && by2<p
92         by2=by2+1;
93     end
94
95     while BW3(py,ax1)==1 && ax1>1
96         ax1=ax1-1;
97     end
98
99     while BW3(py,ax2)==1 && ax2<q
100         ax2=ax2+1;
101     end
102
103
104     if py-by1 <= by2-py
105         bymin=by1;
106         ejemayor=py-by1;
107     else
108         bymin=by2;
109         ejemayor=by2-py;
110     end
111
112     if px-ax1 <= ax2-px

```

```

113
114     axmin=ax1;
115     ejemenor=px-ax1;
116     else
117         axmin=ax2;
118         ejemenor=ax2-px;
119     end
120
121
122     plot(px,bymin,'+r');
123     plot(axmin,py,'+r');
124
125     %reducir eje menor a 3/4 partes para quitar el ruido cercano a los extremos
126     %exteriores de los ojos
127
128     ejemenor=0.75*ejemenor;
129     ejemayor=0.75*ejemayor;
130
131     %Calculo de la ellipse, considerar sistema de coordenadas de MATLAB
132     h=double(px)
133     k=double(py)
134     a=double(ejemenor)
135     b=double(ejemayor)
136
137
138     %region=255*uint8(ones(size(E)));
139     region=255*zeros(size(E));
140
141     for j=(k-b):1:(k+b)
142         for i=(h-a):1:(h+a)
143             x=i;
144
145             if(j<k)
146                 y=(-1)*sqrt(((b^2) * (1-(((x-h)^2)/(a^2)))))) + k ;
147
148                 if(j>=y)
149                     region(j,i) = E(j,i);
150                 else
151                     region(j,i) = 0;
152
153                 end
154             else
155                 y=sqrt(((b^2) * (1-(((x-h)^2)/(a^2)))))) + k ;
156                 if(j<=y)
157                     region(j,i) = E(j,i);
158                 else
159                     region(j,i) = 0;
160                 end
161             end
162         end
163     end
164
165     figure, imshow(uint8(region)),
166     title('Región de la ellipse en niveles de gris');hold on;
167     plot(px,bymin,'+r');
168     plot(axmin,py,'+r');

```

```

169     plot(px,py, '+r');
170
171     %binarizar imagen
172     level=graythresh(region);
173     BW4=im2bw(region,level);
174     [BW4 componentes]=bwlabel(BW4,8);
175     %componentes
176     s=[];
177     for k=1:componentes
178         s=[s; size(find(BW4==k),1)];
179     end
180     % s
181     %media=mean(s)
182     desviacionstd=std(s,1);
183
184     %eliminar componentes menores a la desviación estandar
185     t=[];
186     for k=1:componentes
187         if componentes>6
188             if s(k,1)<desviacionstd
189                 [rows cols]= find(BW4==k);
190                 for l=1:size(rows(:,1))
191                     region(rows(l,1),cols(l,1))=0;
192                 end
193             end
194         end
195     end
196
197     [BW4 componentes]=bwlabel(region,8);
198     %componentes
199     s=[];
200     for k=1:componentes
201         s=[s; size(find(BW4==k),1)];
202     end
203
204     %Eliminar componentes a lo largo del eje vertical
205     [p q]=size(BW4);
206     by=uint16(py);
207
208     while by>1
209
210         k=BW4(by,px);
211         if k~=0
212             [rows cols]= find(BW4==k);
213             for l=1:size(rows(:,1))
214                 BW4(rows(l,1),cols(l,1))=0;
215             end
216         end
217         by=by-1;
218     end
219
220     by=uint16(py);
221     while by<p
222
223         k=BW4(by,px);
224         if k~=0

```

```

225         [rows cols]= find(BW4==k);
226         for l=1:size(rows(:,1))
227             BW4(rows(l,1),cols(l,1))=0;
228         end
229     end
230     by=by+1;
231 end
232
233 figure, imshow(BW4), title('componentes');hold on;
234 plot(px,bymin,'+r');
235 plot(axmin,py,'+r');
236 plot(px,py,'+r');
237
238 %Extraer posición y1 y y2 de los ojos
239 [BW4 componentes]=bwlabel(BW4,8);
240 componentes
241 cm=[];
242     for k=1:componentes
243         [rows cols]= find(BW4==k);
244         cm=[cm; k uint16(mean(rows)) uint16(mean(cols))];
245     end
246
247 cm
248 cm=sortrows(cm,2)
249
250 cmy=mean(cm(:,2))
251 cmx=mean(cm(:,3))
252 plot(cmx,cmy,'or')
253
254 %Extraer posición x1 y x2 de los ojos
255 MX=mean(BW4);
256 MY=mean(BW4');
257 figure,plot(MX)
258 figure,plot(MY)
259 [p q]=size(MX);
260 [valor x1]=max(MX(1,1:px));
261 MX=[zeros([1 px]) MX(1,px:q)];
262 [valor x2]=max(MX(1,:));
263
264 x1
265 x2
266
267 %subplot(1,2,1),imshow(BW4),subplot(1,2,2),plot(MY)
268 %Buscar los picos máximos de cada componente y su ancho
269 [p q]=size(MY)
270 maximos=[]; %ancho del pico y posición en el eje y
271 maximo=0;
272 ancho=0;
273 ancho1=0;
274 anchos=[]; %ancho entre picos y no picos
275 r=0;
276 for k=p:1:q
277
278     if MY(1,k)~=0
279         if r==0;
280             anchos=[anchos;ancho];

```

```

281         ancho=0;
282     end
283     r=1;
284     if MY(1,k)>maximo
285         maximo=MY(1,k);
286         indice=k;
287     end
288     ancho1=ancho1+1;
289 else
290     if r==1
291         maximos=[maximos; ancho1 indice];
292         indice=0;
293         r=0;
294         maximo=0;
295         anchos=[anchos; ancho1];
296         ancho1=0;
297     end
298     ancho=ancho+1;
299 end
300
301 end
302 anchos=[anchos; ancho];
303
304 maximos
305 anchos
306
307 %Eliminar componentes adicionales
308
309 [npicos nancho]=size(maximos);
310 switch npicos
311     case 0
312         y1=0;
313         y2=y1;
314         x1=0;
315         x2=x1;
316     case 1
317         y1=maximos(1,2);
318         y2=y1;
319     case 2
320         ancho1=maximos(1,1);
321         ancho2=anchos(3,1);
322         if ancho2>(2*ancho1)
323             y1=maximos(1,2);
324             y2=y1;
325         else
326             y1=maximos(2,2);
327             y2=y1;
328         end
329     case 3
330         y1=anchos(1,1)+anchos(2,1)+anchos(3,1)+(anchos(4,1)/2);
331         y2=y1;
332     otherwise
333 end
334
335
336 BW4 = bwselect(BW4, [x1;x2], [y1;y2], 8);

```

```

337 figure,imshow(BW4),title('Mapa de ojos');
338
339 disp(sprintf('x1=%d, x2=%d, y1=%d y2=%d',x1,x2,y1,y2));
340
341
342 %cálculo de la distancia d entre los ojos
343 d=sqrt((x2-x1)^2+(y2-y1)^2)
344
345 %Calculo de la ellipse, considerar sistema de coordenadas de MATLAB
346 tetha=atan((y2-y1)/(x2-x1))
347 x0=((x1+x2)/2)+0.43*d*sin(tetha)
348 y0=((y1+y2)/2)+0.43*d*cos(tetha)
349 ejemenor = 1*d
350 ejemayor = 1.3*d
351
352
353 %extraer la región de la ellipse
354 h=ceil(x0)
355 k=ceil(y0)
356 a=ceil(ejemenor)
357 b=ceil(ejemayor)
358
359 %[fotom foton]=size(fotoengris);
360 [fotom foton]=size(fotoengris);
361 region=0*uint8(ones([fotom foton]));
362 %region=255*uint8(ones([ejemayor ejemenor]));
363 region2=0*uint8(ones([fotom foton 3]));
364 for j=(k-b):1:(k+b)
365     for i=(h-a):1:(h+a)
366         x=i;
367
368         if(j<k)
369             y=(-1)*sqrt(((b^2) * (1-(((x-h)^2)/(a^2)))))) + k ;
370
371             if(j>=y)
372                 region(j,i) = fotoengris(j,i);
373                 region2(j,i,1) = fotooriginal(j,i,1);
374                 region2(j,i,2) = fotooriginal(j,i,2);
375                 region2(j,i,3) = fotooriginal(j,i,3);
376             else
377                 region(j,i) = 0;
378                 region2(j,i,1) = 0;
379                 region2(j,i,2) = 0;
380                 region2(j,i,3) = 0;
381
382             end
383         else
384             y=sqrt(((b^2) * (1-(((x-h)^2)/(a^2)))))) + k ;
385             if(j<=y)
386                 region(j,i) = fotoengris(j,i);
387                 region2(j,i,1) = fotooriginal(j,i,1);
388                 region2(j,i,2) = fotooriginal(j,i,2);
389                 region2(j,i,3) = fotooriginal(j,i,3);
390             else
391                 region(j,i) = 0;
392                 region2(j,i,1) = 0;

```

```
393         region2(j,i,2) = 0;
394         region2(j,i,3) = 0;
395     end
396     end
397 end
398 end
399
400 figure, imshow(region), title('Región de la ellipse en niveles de gris');
401 figure, imshow(region2), title('Región de la ellipse en color');
402
403 nombreadchivo1=char(strcat('eg',num2str(ii),'.bmp'));
404 imwrite(region,nombreadchivo1,'bmp');
405 nombreadchivo2=char(strcat('ec',num2str(ii),'.bmp'));
406 imwrite(region2,nombreadchivo2,'bmp');
407
408 D=[D;d h k a b size(fotooriginal)];
409
410 ii
411 disp('cualquier tecla para continuar')
412 pause
413 close all
414 end
415 D
416 mean(D)
417
418
419
```

D.3. Normalización de las regiones elípticas en resolución y suavizado con TV (programa)

La normalización en resolución se realiza a partir de la línea 60 en el programa. El suavizado con TV se realiza en la línea 65.

Para el primer conjunto de 50 imágenes (Fig. 6.2), modificar las líneas 14 y 18 con:

```
N=50;
```

```
errores=[48];
```

Para el segundo conjunto de 40 imágenes (Fig. 6.27), modificar las líneas 14 y 18 con:

```
N=40;
```

```
errores=[36];
```

```

1 %Programa hecho en MATLAB 7.3.0.267 (R2006b) (c)
2
3 %Programa: normalizar las imágenes al mismo tamaño y
4 %           suavizarlas con variación total.
5
6 function normalizacion
7
8 %Limpiar espacio de trabajo
9 clear all
10 close all
11 clc
12
13 %Definir el número de imágenes a normalizar
14 N=40;
15
16 %Definir los errores que no se deben considerar para obtener los parámetros
17 %mínimos
18 errores=[36];
19
20 %Cargar archivo de parámetros
21 D=load('d.txt');
22
23 %Eliminar errores y obtener mínimos
24 disp(sprintf('Programa: Normalización de la base de datos.'))
25 disp(sprintf('\nEliminando errores para calcular parámetros mínimos...'))
26 E=[];
27 [m n]=size(D);
28
29 for i=1:1:N %Conjunto de imágenes
30     if find(errores==i) %No considerar los errores
31     else
32         E=[E;D(i,2:n)];
33     end
34 end
35 minimos=min(E);
36
37 %Recortar imagen y guardar
38 disp(sprintf('\nRecortando imágenes aplicando parámetros mínimos...'))
39 E1=[]; %tabla con valores
40 for i=1:1:N
41     archivo=strcat('eg',num2str(i),'.bmp');
42     [x,map] = imread(archivo);
43     disp(sprintf('\nRecortando archivo: %s ...',archivo))
44     %figure,imshow(x); pause
45     region=centrar(x,D(i,:));
46     E1=[E1; i D(i,2) D(i,5) D(i,6) size(region)] ;
47     nombreamchivo1=char(strcat('en',num2str(i),'.bmp'));
48     disp(sprintf('\nGuardando archivo recortado: %s ...',nombreamchivo1))
49     imwrite(region,nombreamchivo1,'bmp');
50     close all
51 end
52 E1; %Mostrar la tabla de datos
53
54 %Reducir tamaño al mínimo y guardar
55 E2=[];
56 for i=1:1:N

```

```

57     archivo=strcat('en',num2str(i),'.bmp');
58     disp(sprintf('\nNormalizando en resolución y suavizado con TV:, archivo: %s ...',
archivo))
59     [x,map] = imread(archivo);
60     region=imresize(x,[minimos(1,6) minimos(1,7)],'bilinear');
61     %size(region)
62     E2=[E2; i size(region)];
63     figure, imshow(region);
64     %suavizar con variación total
65     region =uint8(tv(double(region),300,10,double(region)));
66     figure, imshow(uint8(region))
67     nombreadchivo1=char(strcat('en',num2str(i),'.bmp'));
68     disp(sprintf('\nGuardando archivo normalizado: %s ...',nombreadchivo1))
69     imwrite(region,nombreadchivo1,'bmp');
70     close all
71 end
72
73 E2;
74
75 %Subfunción: centrar la imagen 'x' de acuerdo a los parámetros dados en 'e'
76 function region=centrar(x,e)
77 y=zeros(size(x));
78 x1=e(1,3)-e(1,5);
79 y1=e(1,4)-e(1,6);
80 x2=e(1,3)+e(1,5);
81 y2=e(1,4)+e(1,6);
82 region=x(y1:y2,x1:x2);
83
84

```

D.4. Cálculo de eigenfaces (programa)

```

1 %Localización y Reconocimiento de rostros en imágenes monoculares con
2 %variación en escala
3 %Programa hecho en MATLAB 7.3.0.267 (R2006b) (c)
4
5 %Programa: Experimento, Cálculo de Eigenfaces
6
7 %limpiar espacio de trabajo
8 clear all
9 close all
10 clc
11 % Consederar los errores de las elipses de entrada
12 errores=[15 48];
13
14 %Número de imagenes
15 M=25;
16
17 %Constantes para normalizar en iluminación con valores para std y mean.
18 %Valores cercanos muchas imagens
19 um=100;
20 ustd=80;
21
22 %Leer las imágenes en formato .bmp
23 S=[]; %matriz de imágenes
24 figure(1);
25 for i=1:M
26     str=strcat('en',int2str(i),'.bmp'); %concatenar el nombre de archivo para elipses
27     eval('img=imread(str);');
28
29     subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
30     imshow(img)
31     if i==3
32         title('Imágenes de entrada','fontsize',16)
33     end
34     drawnow;
35     [irow icol]=size(img);
36     temp=reshape(img',irow*icol,1); %crear la matriz N=nxm
37
38     if find(errores==i) %No considerar los archivos con errores
39     else
40         S=[S temp]; %crear la matriz NxM
41     end
42 end
43
44 %Normalizar las imagenes para reducir los errores por iluminación.
45 for i=1:size(S,2)
46     temp=double(S(:,i));
47     me=mean(temp);
48     st=std(temp);
49     S(:,i)=(temp-me)*ustd/st+um;
50 end
51
52 %Mostrar las imágenes normalizadas
53 figure(2);
54 for i=1:size(S,2)
55     str=strcat(int2str(i),'.bmp');
56     img=reshape(S(:,i),icol,irow);

```

```

57     img=img';
58     subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
59     imshow(img)
60     drawnow;
61     if i==3
62         title('Imágenes normalizadas en iluminación','fontsize',16)
63     end
64 end
65
66 %Rostro promedio
67 m=mean(S,2);
68 img=reshape(uint8(m),icol,irow);
69 figure(3),imshow(img'),title('Imagen promedio','fontsize',16)
70
71 A=double(S');
72 L=A*A';
73 [vv dd]=eig(L);
74
75 % Ordenar y eliminar los eigenvalores que sean cero
76 v=[];
77 d=[];
78 for i=1:size(vv,2)
79     if(dd(i,i)>1e-4)
80         v=[v vv(:,i)];
81         d=[d dd(i,i)];
82     end
83 end
84 %ordenar en forma ascendente
85 [B index]=sort(d);
86 ind=zeros(size(index));
87 dtemp=zeros(size(index));
88 vtemp=zeros(size(v));
89 len=length(index);
90 for i=1:len
91     dtemp(i)=B(len+1-i);
92     ind(i)=len+1-index(i);
93     vtemp(:,ind(i))=v(:,i);
94 end
95 d=dtemp;
96 v=vtemp;
97
98
99 %Normalización de eigenvectores
100 for i=1:size(v,2)
101     v(:,i)=v(:,i)./sqrt(sum(v(:,i).^2));
102 end
103
104 %Eigenvectores de la matriz de covarianza C
105 u=[];
106 for i=1:size(v,2)
107     u=[u (double(S)*v(:,i))./sqrt(d(i))];
108 end
109
110 %Normalizacion de eigenvectores
111 for i=1:size(u,2)
112     u(:,i)=u(:,i)./sqrt(sum(u(:,i).^2));

```

```

113 end
114
115 %Visualizar eigenfaces
116 figure(4);
117 for ii=1:size(u,2)
118     img=reshape(u(:,ii),icol,irow);
119     img=histeq(img',255);
120     subplot(ceil(sqrt(M)),ceil(sqrt(M)),ii)
121     imshow(img)
122     drawnow;
123     if ii==3
124         title('Eigenfaces','fontsize',16)
125     end
126 end
127
128 % Determinar el peso de cada imagen
129 omega = [];
130 for h=1:size(S,2)
131     WW=[];
132     for ii=1:size(u,2)
133         WeightOfImage = dot(u(:,ii)',double(S(:,h))');
134         WW = [WW; WeightOfImage];
135     end
136     omega = [omega WW];
137 end
138
139 ff=[];
140 for h=1:size(S,2)
141     InImWeight = [];
142     for ii=1:size(u,2)
143         WeightOfInputImage = dot(u(:,ii)',double(S(:,h))');
144         InImWeight = [InImWeight; WeightOfInputImage];
145     end
146     f=[];
147     for ii=1:size(omega,2)
148         q = omega(:,ii);
149         DiffWeight = InImWeight-q;
150         mag = norm(DiffWeight);
151         f = [f; mag];
152     end
153     ff=[ff sort(f)];
154 end
155
156
157 disp('Cualquier tecla para continuar...')
158 pause
159
160 %Fase de de reconocimiento
161 for i=1:1:50
162     if finderrores==i)
163         else
164
165     filename=strcat('en',int2str(i),'.bmp');
166     InputImage=filename;
167     eval('InputImage=imread(InputImage);');
168     figure(5)

```

```

169 subplot(1,2,1)
170 imshow(InputImage); colormap('gray');title('Buscando a...', 'fontsize',18)
171 InImage=reshape(InputImage',irow*icol,1);
172 temp=double(InImage);
173 me=mean(temp);
174 st=std(temp);
175 temp=(temp-me)*ustd/st+um;
176
177 InImWeight = [];
178 for ii=1:size(u,2)
179     WeightOfInputImage = dot(u(:,ii)',temp');
180     InImWeight = [InImWeight; WeightOfInputImage];
181 end
182
183 % Encontrar la distancia euclidiana entre cada uno de los eigenfaces
184 e=[];
185 for ii=1:size(omega,2)
186     q = omega(:,ii);
187     DiffWeight = InImWeight-q;
188     mag = norm(DiffWeight);
189     e = [e; mag];
190 end
191 valormaximo=max(e);
192 [valorminimo, indice ]=min(e);
193
194 %disp('valores maximo/minimo de imagen de entrada:')
195
196 subplot(1,2,2)
197 imagen=reshape(S(:,indice),icol,irow);
198 imshow(imagen'); colormap('gray');
199 %if( (valorminimo>=min(f)) & (valorminimo<=max(f)) )
200 %title(';Encontrado!', 'fontsize',16)
201 %disp(sprintf('i=%d, valorminimo=%d, indice=%d, encontrado',i,valorminimo,indice))
202 %else
203 %title('*No Encontrado*', 'fontsize',16)
204 disp(sprintf('i=%d, valorminimo=%d, indice=%d, no encontrado',i,valorminimo,indice))
205 %end
206 %disp('Cualquier tecla para continuar ...')
207 pause
208 end
209 end

```

D.5. Generación de la pirámide de imágenes (programa)

```

1 %Programa hecho en MATLAB 7.3.0.267 (R2006b) (c)
2
3 %Programa: Generar pirámide de imágenes
4
5 %Limpiar espacio de trabajo
6
7 clear all
8 close all
9 clc
10
11
12 for ii=1:1:50 %UAM-C
13     for J=7:-1:1 % Reducir a partir del nivel 7 hasta el nivel 1
14         %Leer archivo
15         archivo=strcat('en',num2str(ii),'.bmp');
16         [x,map] = imread(archivo);
17         [m n]=size(x);
18         mrenglones= 2^J; %Proporción para el # de renglones
19         ncolumnas=n*((2^J)*1)/m; %Proporción para el # de columnas
20         %generar reducir el tamaño de la imagen, usar 'nearest' vecino más
21         %cercano para que se consideren los tamaños más pequeños.
22         archivoescalado=imresize(x,[mrenglones ncolumnas],'nearest');
23         %Mostrar en pantalla las imágenes escaladas
24         figure, imshow(x),
25         figure, imshow(archivoescalado)
26         %Nombrar el archivo con la referencia siguiente:
27         %ee ellipse escalada, J nivel, # fotografía: aeJ_#.bmp
28         nombreakhivo=char(strcat('ee',num2str(J),'_',num2str(ii),'.bmp'));
29         disp(nombreakhivo)
30         %Guardar el archivo
31         imwrite(archivoescalado,nombreakhivo,'bmp');
32         close all %Limpiar el espacio de figuras (figure)
33     end
34 end
35
36
37
38

```

D.6. Verificar imágenes que conforman las bases de datos (programa)

```

1 %Programa hecho en MATLAB (R2006b) (c)
2 %Programa: Cálculo de Eigenfaces y reconocimiento de rostros para la pirámide
3 %de la base de datos. Desde el nivel 8 al nivel 4.
4 %Este programa verifica todas las imagenes de cada nivel en la base de datos
5
6 %Limpiar espacio de trabajo
7 clear all
8 close all
9 clc
10
11 %Iniciar programa
12
13 for nivel=8:-1:4 %Seleccionar nivel de la pirámide
14 disp(sprintf('Calculando espacio de eigenfaces: nivel %d ...',nivel))
15
16 inicio=1; %Iniciar a partir de la imagen No. 1.
17 M=49; %Considerar hasta imagen No. 49.
18 %*****%
19
20 %Constantes para normalizar en iluminación
21 %Valores para std y mean cercanos a muchas imágenes
22 um=100;
23 ustd=80;
24
25 %Leer las imágenes en formato .bmp
26 S=[]; %Matriz de imágenes
27 figure(1);
28     for i=inicio:M
29         %Concatenar el nombre de archivo para elipses escaladas
30         str=strcat('ee',int2str(nivel),'_',int2str(i),'.bmp');
31         eval('img=imread(str);');
32         %Visualizar el conjunto de imágenes de entrada
33         subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
34         imshow(img)
35         if i==3
36             title('Imágenes de entrada','fontsize',16)
37         end
38         drawnow;
39         [irow icol]=size(img);
40         temp=reshape(img,irow*icol,1); %Crear la matriz N=nxm
41         S=[S temp]; %Crear la matriz NxM
42     end
43
44
45 %Normalizar las imágenes para reducir los errores por iluminación
46 for i=1:size(S,2)
47     temp=double(S(:,i));
48     me=mean(temp);
49     st=std(temp);
50     S(:,i)=(temp-me)*ustd/st+um;
51 end
52
53 %Mostrar las imágenes normalizadas
54 figure(2);
55 for i=1:size(S,2)
56     str=strcat(int2str(i),'.bmp');

```

```

57     img=reshape(S(:,i),icol,irow);
58     img=img';
59     subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
60     imshow(img)
61     drawnow;
62     if i==3
63         title('Imágenes normalizadas en iluminación','fontsize',16)
64     end
65 end
66
67 %Calcular rostro promedio
68 m=mean(S,2);
69 img=reshape(uint8(m),icol,irow);
70 figure(3),imshow(img'),title('Imagen promedio','fontsize',16)
71
72 A=double(S');
73 L=A*A';
74 [vv dd]=eig(L);
75
76 %Ordenar y eliminar los eigenvalores que sean cero
77 v=[];
78 d=[];
79 for i=1:size(vv,2)
80     if(dd(i,i)>1e-4)
81         v=[v vv(:,i)];
82         d=[d dd(i,i)];
83     end
84 end
85
86 %ordenar en forma ascendente
87 [B index]=sort(d);
88 ind=zeros(size(index));
89 dtemp=zeros(size(index));
90 vtemp=zeros(size(v));
91 len=length(index);
92 for i=1:len
93     dtemp(i)=B(len+1-i);
94     ind(i)=len+1-index(i);
95     vtemp(:,ind(i))=v(:,i);
96 end
97 d=dtemp;
98 v=vtemp;
99
100 %Normalización de eigenvectores
101 for i=1:size(v,2)
102     v(:,i)=v(:,i)./sqrt(sum(v(:,i).^2));
103 end
104
105 %Eigenvectores de la matriz de covarianza C
106 u=[];
107 for i=1:size(v,2)
108     u=[u (double(S)*v(:,i))./sqrt(d(i))];
109 end
110
111 %Normalizacion de eigenvectores
112 for i=1:size(u,2)

```

```

113     u(:,i)=u(:,i)./sqrt(sum(u(:,i).^2));
114 end
115
116 %Visualizar eigenfaces
117 figure(4);
118 for ii=1:size(u,2)
119     img=reshape(u(:,ii),icol,irow);
120     img=histeq(img',255);
121     subplot(ceil(sqrt(M)),ceil(sqrt(M)),ii)
122     imshow(img)
123     drawnow;
124     if ii==3
125         title('Eigenfaces','fontsize',16)
126     end
127 end
128
129 %Determinar el peso de cada imagen
130 omega = [];
131 for h=1:size(S,2)
132     WW=[];
133     for ii=1:size(u,2)
134         WeightOfImage = dot(u(:,ii)',double(S(:,h)'));
135         WW = [WW; WeightOfImage];
136     end
137     omega = [omega WW];
138 end
139
140 %Fase de de reconocimiento
141 disp(sprintf('Reconocimiento de rostros de la base de datos: nivel %d ...',nivel))
142
143 for i=1:1:M
144     filename=strcat('ee',int2str(nivel),'_',int2str(i),'.bmp');
145     InputImage=filename;
146     eval(['[InputImage,map]=imread(InputImage)']);
147     figure, subplot(1,2,1)
148     imshow(InputImage); colormap('gray');title('Buscando a:','fontsize',18)
149     InImage=reshape(InputImage',irow*icol,1);
150     temp=double(InImage);
151     me=mean(temp);
152     st=std(temp);
153     temp=(temp-me)*ustd/st+um;
154     InImWeight = [];
155     for ii=1:size(u,2)
156         WeightOfInputImage = dot(u(:,ii)',temp');
157         InImWeight = [InImWeight; WeightOfInputImage];
158     end
159
160 %Encontrar la distancia euclidiana entre cada uno de los eigenfaces
161 e=[];
162 for ii=1:size(omega,2)
163     q = omega(:,ii);
164     DiffWeight = InImWeight-q;
165     mag = norm(DiffWeight);
166     e = [e; mag];
167 end
168 valormaximo=max(e);

```

```
169     [valorminimo, indice ]=min(e);
170     subplot(1,2,2)
171     imagen=reshape(S(:,indice),icol,irow);
172     imshow(imagen'); colormap('gray');title('Encontrado:', 'fontsize',18)
173     %disp(sprintf('i=%d, valorminimo=%d, indice=%d',i,valorminimo,indice))
174     pause
175     end %Fin de reconocimiento de rostros del nivel en prueba
176     close all %Limpiar espacio de figuras del nivel en prueba
177
178 end %Fin de nivel
179
```

D.7. Reconocimiento de rostros, prueba de verificación de cambios de escala

```

1 %% Face recognition
2 % This algorithm uses the eigenface system (based on principal component
3 % analysis - PCA) to recognize faces. For more information on this method
4 % refer to http://cnx.org/content/ml2531/latest/
5
6 %% Download the face database
7 % You can find the database at the following link,
8 % http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html The
9 % database contains 400 pictures of 40 subjects. Download the zipped
10 % database and unzip it in the same directory as this file.
11
12 %% Loading the database into matrix v
13
14 clear all
15 close all
16 clc
17 w=load_database();
18
19 %% Initializations
20 % We randomly pick an image from our database and use the rest of the
21 % images for training. Training is done on 399 pictures. We later
22 % use the randomly selected picture to test the algorithm.
23
24 %ri=round(400*rand(1,1));           % Randomly pick an index.
25
26 %ri=round(1);
27 %r=w(:,ri);                       % r contains the image we later on will use to
test the algorithm
28 %v=w(:,[1:ri-1 ri+1:end]);         % v contains the rest of the 399 images.
29 %v=w(:,[1:ri-1 ri+1:end])
30 v=v;
31 N=20;                               % Number of signatures used for each image.
32 %% Subtracting the mean from v
33 O=uint8(ones(1,size(v,2)));
34 m=uint8(mean(v,2));                 % m is the mean of all images.
35 vzm=v-uint8(single(m)*single(O)); % vzm is v with the mean removed.
36
37 %% Calculating eigenvectors of the correlation matrix
38 % We are picking N of the 400 eigenfaces.
39 L=single(vzm)'*single(vzm);
40 [V,D]=eig(L);
41 V=single(vzm)*V;
42 V=V(:,end:-1:end-(N-1));           % Pick the eigenvectors corresponding to the 10
largest eigenvalues.
43
44
45 %% Calculating the signature for each image
46 cv=zeros(size(v,2),N);
47 for i=1:size(v,2);
48     cv(i,:)=single(vzm(:,i))'*V;   % Each row in cv is the signature for one image.
49 end
50
51 %%
52 %Seleccionar la serie de imágenes de prueba
53
54 for ii=1:5

```

```

55 cd(strcat('C:\Users\jorge\Desktop\Reconocimiento\att\att_faces\orl_faces\s',num2str(
(1)));
56
57     a=imread(strcat(num2str(10+ii),'.pgm'));
58     r=reshape(a,size(a,1)*size(a,2),1);
59 cd ..
60
61
62 %% Recognition
63 % Now, we run the algorithm and see if we can correctly recognize the face.
64 subplot(121);
65 imshow(reshape(r,112,92));title('Buscando a ...','FontWeight','bold','FontSize',
16,'color','red');
66
67 subplot(122);
68 p=r-m;                                % Subtract the mean
69 s=single(p)'*V
70 z=[];
71 size(v,2)
72 for i=1:size(v,2)
73
74     z=[z,norm(cv(i,:)-s,2)];
75     %rem(i,20)
76     if(rem(i,20)==0),imshow(reshape(v(:,i),112,92)),end;
77     drawnow;
78
79 end
80 i
81 [a,i]=min(z);
82 subplot(122);
83 imshow(reshape(v(:,i),112,92));title('Encontrado!','FontWeight','bold','FontSize',
16,'color','red');
84
85 pause
86 close all
87 end
88
89
90 function out= load_database()
91 % We load the database the first time we run the program.
92
93 persistent loaded;
94 persistent w;
95 if(isempty(loaded))
96     v=zeros(10304,400);
97     for i=1:40
98         cd(strcat('C:\Users\jorge\Desktop\Reconocimiento\att\att_faces\orl_faces\s',
num2str(i)));
99
100         for j=1:10
101             a=imread(strcat(num2str(j),'.pgm'));
102             v(:,(i-1)*10+j)=reshape(a,size(a,1)*size(a,2),1);
103
104         end
105
106     cd ..

```

```
107     end
108     w=uint8(v); % Convert to unsigned 8 bit numbers to save memory.
109 end
110 loaded=1; % Set 'loaded' to avoid loading the database again.
111 out=w;
```