



INSTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERIA MECANICA Y
ELECTRICA "UNIDAD CULHUACAN"



SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

VERIFICACIÓN DE IDENTIDAD POR MEDIOS BIOMÉTRICOS

TESIS

Que para obtener el grado de
MAESTRO EN CIENCIAS DE INGENIERIA EN MICROELECTRONICA

presenta

Ing. Jesus Olivares Mercado

Director de Tesis

Dr. Héctor Manuel Pérez Meana

México D.F. Diciembre 2008

A mi familia con todo mi amor.

Agradecimientos

Este trabajo es el final de un periodo mas de esfuerzos, sacrificios y emociones en mi vida, todo esto a sido posible gracias a muchas personas que día tras día han estado conmigo y me apoyaron para poder culminar satisfactoriamente este proyecto.

*En primer lugar quiero agradecer al **Instituto Politécnico Nacional** y en especial a la **Sección de Estudios de Posgrado e Investigación de ESI-ME Culhuacan** por darme la oportunidad de realizar mis estudios de maestría.*

*A mi asesor **Dr. Héctor Manuel Pérez Meana** por aconsejarme, apoyarme y brindarme su confianza en todo momento. A la **Dra. Mariko Nakano Miyatake** y al **Dr. Gabriel Sánchez Pérez** por su ayuda durante este proyecto.*

*A mis padres **Gloria** y **Gabriel** que siempre me han brindado su apoyo, amor y comprensión, y por haber creído en mí y en mis capacidades para alcanzar mis metas.*

*A mis tres hermanas, **Gabriela**, **Xochilt** y **Zoila** que siempre me apoyaron en cada momento y que me han ofrecido su amor incondicionalmente.*

*A **Judith** por ser parte importante de mi vida, darme todo su amor y comprensión y siempre motivarme para seguir adelante.*

*A los profesores **Kazuhiro Hotta** y **Haruhisa Takahashi** asi como a la **Universidad de Electro-Comunicaciones de Tokyo** por haberme aceptado y apoyado durante mi estancia en **Japón**.*

*Al **CONACYT** por brindarme el soporte económico para poder realizar este trabajo de investigación.*

A todas las personas de la SEPI por haberme brindado su amistad y su apoyo durante todo este tiempo.

Resumen

En cuanto a los trabajos realizados en el reconocimiento y la verificación de personas por medio de su rostro, el cual debe ser analizado por distintas técnicas y algoritmos matemáticos para obtener parámetros representativos, además de poder clasificar las características principales de un individuo y así poder usar dichos parámetros para su identificación. Para la correcta identificación o verificación de una persona entre un grupo tipificado de modelos de rostro, es necesario el uso de notaciones que clasifiquen de manera correcta los parámetros de cada una de estas muestras que se desean reconocer y por ende diferenciar una de otra. Para este fin, es necesario construir un sistema de reconocimiento de rostros que debe estar formado por diferentes módulos donde se incluye el análisis de la imagen, la clasificación de parámetros y el reconocimiento para ser utilizado en aplicaciones del mundo real.

Este trabajo de investigación que se presenta generaliza el uso de técnicas para el procesamiento de imágenes bidimensionales, técnicas de clasificación y técnicas de toma de decisiones. Las técnicas para el procesamiento de la imagen son filtros de Gabor, Wavelets (Daubichies) y el método llamado eigenphases, a este último se le aplicaron dos variaciones al método original. Estas técnicas serán evaluadas y comparadas unas con otras mediante la aplicación de sistemas de reconocimiento o verificación de rostros. Las técnicas de clasificación usadas son el Modelo de Mezclas Gaussianas y la Máquina de Vector Soporte. Las diferentes técnicas de toma de decisiones son empleadas en los sistemas y combinadas para obtener algoritmos de identificación y verificación de rostros.

El objetivo es señalado y nombrado de tal forma que pueda ser cuantificado para su posterior evaluación, se pretende llevar los resultados de otros trabajos en el área de reconocimiento o verificación de rostros, para contrastarse con un método o algoritmo propuesto en este trabajo de investigación.

Este trabajo cuenta con un estado del arte, donde se señalarán las características principales de cada uno de los diferentes algoritmos empleados, para atacar dicho problema; obteniendo los resultados que se compararán en el último capítulo de resultados.

Abstract

As for the works carried out in recognition and verification of people by their face, which should be analyzed for different technical and mathematical algorithms to obtain representative parameters, besides being able to classify the main characteristics of an individual and this way to be able to use this parameters for their identification. For the correct identification or verification of a person among a group of face models, it is necessary the use of indications that classify in a correct way the parameters of each one of these samples that are wanted to recognize and therefore to differentiate one of another. For this end, it is necessary to build a system of face recognition that should be formed by different modules where it is included the image analysis, the classification of parameters and the recognition to be used in applications of the real world.

This investigation work generalizes the use of techniques for the processing of the images 2-D, technics of classification and technics of taking decisions. The techniques for the image processing are Gabor filters, Wavelets (Daubichies), and the method called eigenphases, to the last one we apply 2 different variations of the original method. This technics will be evaluated and compared some with other mediating one the application of face recognition or face verification systems. The techniques for classification are Gaussian Mixture Models and Support Vector Machine. The different techniques of taking of decisions are employees in the systems and combined to obtain identification and verification algorithms.

The objective is pointed out and noted in such a way that can be quantified for its later evaluation, it is sought to take the results of other works of recognition or verification area, to be contrasted with a method or algorithm proposed in this investigation work.

This work has a theoretical mark, where the main characteristics will be pointed out of each one of the used different algorithms, to attack this problem; obtaining the results will be compared in the last chapter of results.

Contenido

Dedicatoria	I
Agradecimientos	III
Resumen	V
Abstract	VII
Contenido	IX
Lista de Figuras	XI
Lista de Tablas	XIII
1. Introducción	1
1.1. Justificación	3
1.2. Objetivos	4
1.2.1. Objetivo general	4
1.2.2. Metas	4
1.3. Aportaciones Principales	5
1.4. Organización de la tesis	5
2. Estado del Arte	7
2.1. Sistemas Biométricos	7
2.1.1. Modelo del proceso de identificación personal.	8
2.1.2. Características de un indicador biométrico	8
2.1.3. Características de un sistema biométrico	8
2.1.4. Panorama General de los Sistemas Biométricos	9
2.2. Métodos de Extracción de Características	10
2.2.1. Eigenfaces	10
2.2.2. Discriminantes Lineales de Fisher	14
2.2.3. Transformada Discreta de Walsh	15
2.2.4. Transformada Discreta de Gabor (Filtros de Gabor)	16
2.2.5. Transformada Discreta Wavelet	19
2.2.6. Eigenphases	21
2.3. Conclusiones	23

3. Clasificadores	25
3.1. Redes Neuronales	25
3.1.1. Modelo de una Neurona Artificial	26
3.1.2. Características de una red neuronal	27
3.1.3. Red Neuronal Perceptrón Multicapa	27
3.2. Modelo de Mezclas Gaussianas (GMM)	30
3.2.1. Inicialización del modelo	30
3.2.2. Desarrollo del modelo	33
3.2.3. Evaluación del modelo	35
3.3. Máquina de Vector de Soporte (SVM)	36
3.3.1. Clasificación mediante SVM	38
3.4. Conclusiones	41
4. Sistema Propuesto	43
4.1. Introducción	43
4.2. Diagrama del Sistema	44
4.2.1. Módulo de Entrenamiento	44
4.2.2. Módulo de Prueba	53
4.3. Conclusiones	56
5. Resultados Experimentales	57
5.1. GMM como clasificador y verificador	58
5.2. SVM como clasificador y verificador	63
6. Conclusiones Generales y Trabajos Futuros	67
6.1. Conclusiones Generales	67
6.2. Trabajos Futuros	70
Referencias	71
A. Publicaciones	75

Lista de Figuras

2.1. Sistemas Biométricos.	9
2.2. a)Rostro original. b)Imagen promedio. c)Diferencia del rostro respecto al promedio.	12
2.3. Niveles de descomposición wavelet.	20
2.4. Experimento de Oppenheim.	21
3.1. Modelo de neurona simple.	26
3.2. Red neuronal perceptrón multicapa.	28
3.3. Aproximación de una función de distribución de probabilidad mediante una suma pesada de funciones Gaussianas.	30
3.4. Ilustración del algoritmo de K-Means para M=3.	32
3.5. Modelo de Mezclas Gaussianas, GMM.	34
3.6. Ejemplo de SVM.	38
4.1. Sistema Propuesto.	44
4.2. a) Imagen Original. b) Imagen recortada.	45
4.3. a) Imagen Original. b) Imagen con la malla.	46
4.4. a) Imagen Original. b) Contrastes de Energia. c) Vector caracteristico.	47
4.5. a) Imagen Original. b) Contrastes de Energia. c) Vector caracteristico.	47
4.6. 4 niveles de descomposición wavelet.	48
4.7. Variacion entre vectores característicos.	48
4.8. Esquema para la extracción de características usando PCA.	50
4.9. Esquema general para el entrenamiento del GMM.	51
4.10. Esquema general para el entrenamiento del SVM.	52
4.11. Esquema general para el reconocimiento con GMM.	54
4.12. Esquema general para la verificación con GMM.	54
4.13. Esquema general para el reconocimiento con SVM.	55
4.14. Esquema general para la verificación con SVM.	55
5.1. a) Imágenes de entrenamiento 1. b) Imágenes de entrenamiento 2.	58

Lista de Tablas

5.1. Reconocimiento usando Filtros de Gabor y GMM	60
5.2. Verificacion usando Filtros de Gabor y GMM	60
5.3. Reconocimiento usando DWT y GMM	60
5.4. Verificacion usando DWT y GMM	61
5.5. Reconocimiento usando Eigenphases y GMM	61
5.6. Verificacion usando Eigenphases y GMM	62
5.7. Reconocimiento usando Filtros de Gabor y SVM	63
5.8. Verificacion usando Filtros de Gabor y SVM	64
5.9. Reconocimiento usando DWT y SVM	64
5.10. Verificacion usando DWT y SVM	65
5.11. Reconocimiento usando Eigenphases y SVM	65
5.12. Verificacion usando Eigenphases y SVM	66

Capítulo 1

Introducción

La seguridad es el tema actual, el reconocimiento y la verificación de la identidad de las personas es uno de los aspectos más fundamentales. Los ataques terroristas ocurridos actualmente han demostrado la necesidad de establecer métodos más confiables para verificar la identidad de las personas. Los sistemas biométricos surgen como una solución real a las demandas de verificación. La biometría consiste de un conjunto de métodos automatizados para la verificación de individuos mediante el uso de características físicas o del comportamiento de la persona [1]. Esta tecnología se basa en la premisa de que cada persona es única y posee rasgos distintivos que pueden ser utilizados para identificarla.

Los seres humanos poseen una alta capacidad para reconocer rostros aún en escenarios donde existan altos niveles de variabilidad y ruido. Diseñar sistemas automáticos que emulen esta propiedad natural de los humanos, constituye una tarea compleja y con muchas limitaciones. Probablemente una de las primeras interrogantes sea ¿los rostros son diferenciables como medidas biométricas? Afortunadamente en los últimos años se han realizado una gran cantidad de investigaciones que afirman esta interrogante, en especial el área de la biometría, la biometría busca obtener, clasificar y utilizar la información de estas características, para reconocer o verificar la identidad de las personas, restringir el acceso a sitios no permitidos, controlar horarios, autenticar información, y muchas otras aplicaciones. Para esto se utilizan equipos electrónicos que desarrollan las mediciones biométricas, y algoritmos que permiten digitalizar, clasificar y almacenar la información.

La verificación de rostros es un área de investigación activa con numerosas aplicaciones desde los años 80. Es tal vez, el método biométrico más fácil de comprender ya que para nosotros la cara es la manera más directa de identificar a personas y debido a que la adquisición de datos para este método funciona de forma similar a tomarse una foto, es de mucha aceptación entre los usuarios. Varios algoritmos han sido propuestos para el reconocimiento de rostros, para un estudio detallado ver [2, 3]. El reconocimiento es una actividad muy compleja del cerebro humano sin una explicación concreta. Nosotros podemos reconocer cientos de rostros aprendidos a lo largo de nuestra vida e identificar rostros familiares a primera vista incluso después de varios años de separación.

Los sistemas de reconocimiento de rostros actuales han alcanzado un porcentaje de reconocimiento de hasta el 90 % cuando el rostro en la imagen no está rotado o la rotación es relativamente baja, este porcentaje es bueno si en la aplicación podemos tolerar algunas fallas. Por el contrario, si necesitamos una aplicación donde la seguridad debe ser extrema, no podemos tolerar un alto porcentaje de reconocimiento erróneo. En este trabajo se propone un algoritmo que permita reducir el porcentaje de verificación erróneo por debajo del 5 %, presentando variaciones en las imágenes como son, cambios en la iluminación, oclusión parcial del rostro o algunas expresiones faciales.

Antes de empezar a analizar los procedimientos es necesario dejar claro el concepto de reconocimiento y verificación. En el reconocimiento el sistema no sabe quien es la persona de la cual ha capturado los rasgos característicos (el rostro humano en este caso) por lo cual el sistema tiene que decir a quien pertenecen los datos que acaba de procesar. En la verificación, la persona le informa al sistema cual es su identidad ya sea presentando una tarjeta de identificación o escribiendo alguna clave especial, el sistema captura el rasgo característico de la persona (el rostro humano en este caso), y lo procesa para crear una representación electrónica llamada modelo en vivo (“live template” en inglés). Por último, el sistema compara el modelo en vivo con el modelo de referencia de la persona. Si ambos modelos coinciden la verificación es exitosa. De no serlo, la verificación es no-exitosa.

1.1. Justificación

La necesidad de incrementar la seguridad en algunos lugares exclusivos por medio de un riguroso control de acceso es un campo de activa investigación durante los últimos años. El uso de los patrones característicos de una imagen de un rostro para el reconocimiento o verificación de personas tiene algunas ventajas ante los demás tipos biométricos de identificación de las personas como son: Huellas digitales, patrones de la palma de la mano, patrones del iris, firmas, ya que no se necesita ningún equipo especial para la captura de el rasgo biométrico si no que con una simple cámara uno puede tomar la fotografía para después procesarla además que es un esquema no invasivo. Cualquier sistema de Reconocimiento de rostro es uno de los métodos de identificación de personas, a partir de sus características físicas, con una mayor aceptación entre los potenciales usuarios. Sin embargo, a pesar de que varios métodos han sido propuestos durante estos últimos años, su funcionamiento se debe mejorar aun de manera sustancial con el fin de reducir errores de identificación y verificación aun presentes. El uso de identificación por medio del rostro tiene otra gran ventaja ante las demás características biométricas, ésta es que el reconocimiento de personas puede hacerse a larga distancia usando algún sistema o medio de comunicación tal como una cámara web.

Recientemente los estudios de muchas investigaciones están enfocados en los sistemas de reconocimiento de personas por medio del rostro debido a la gran potencialidad en muchas aplicaciones, tales como: Control de acceso a lugares restringidos o a información confidencial, Apoyo a la ley, etc.

Recientemente se ha propuesto el empleo de métodos probabilísticos para reducir la complejidad computacional. Los resultados obtenidos muestran que estos métodos son altamente eficientes para el entrenamiento de sistemas de reconocimiento o verificación de personas por medio del rostro. Además de los puntos expuestos anteriormente, señalaremos que este tipo de algoritmos aplicados a la biometría son atractivos a los proveedores de seguridad, debido a la comodidad que ofrece el uso de sistemas biometricos. El cliente que acepta dicho sistema de identificación, además de que el cliente mismo no puede olvidar la contraseña, siendo seguro tanto para el cliente como para el sistema de verificación ya que los rasgos faciales no pueden ser robados ni pueden compartirse.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar un algoritmo robusto que permita el reconocimiento y la verificación de la identidad de una persona por medio de su rostro, que sea robusto a cambios de iluminación, de expresiones faciales, y de la oclusión parcial, con un alto grado de acierto.

1.2.2. Metas

- Obtener una Base de Datos con imágenes que tengan variaciones como iluminación, oclusión y expresiones faciales.
- Realizar el análisis para el recorte de la imagen.
- Automatizar el recorte del rostro de cada imagen.
- Analizar los diferentes tipos de extracción de características a usar.
- Automatizar la extracción de características de cada imagen.
- Analizar el uso de las herramientas de clasificación (GMM y SVM).
- Realizar el reconocedor y el verificador usando las características obtenidas y las herramientas de clasificación.
- Evaluar cada sistema por medio de la simulación.
- Obtener los resultados como reconocedores de cada uno de los sistemas de clasificación con cada uno de los tipos de características.
- Obtener los resultados como verificadores de cada uno de los sistemas de clasificación con cada uno de los tipos de características.
- Comparar los resultados obtenidos con el reconocedor y con el verificador y los diferentes métodos de extracción de características.

1.3. Aportaciones Principales

La presente tesis presenta el desarrollo de algunos algoritmos para la extracción de características, así como también el uso de dos clasificadores para poder desarrollar y comparar el desempeño de estos como clasificadores y como verificadores.

Uno de los objetivos es mejorar el porcentaje de reconocimiento y verificación de personas por medio de su rostro.

Algunas de las aportaciones principales de este trabajo se enlistan a continuación:

- Desarrollo de 3 diferentes métodos de extracción de características, las cuales son por medio del uso de filtros de Gabor, Wavelets, Eigenphases
- Aplicación de algunas variantes al método de extracción de Eigenphases, mejorando el porcentaje de aceptación.
- El uso de 2 clasificadores para la tarea de reconocimiento y verificación.
- La comparación de desempeño del sistema como clasificador y como verificador.

1.4. Organización de la tesis

Capítulo 1 Presenta un panorama general del desarrollo de la tesis así como la justificación, los objetivos y metas a alcanzar, y las principales aportaciones de este trabajo.

Capítulo 2 Presenta una introducción acerca de lo que es la biometría y cual es la importancia que ocupa en ella el reconocimiento de rostros, así como también presenta una explicación de lo que es el desarrollo de diferentes métodos de extracción de características incluyendo los utilizados en este trabajo.

Capítulo 3 Presenta la explicación de 3 métodos de clasificación los cuales son: Redes Neuronales, Modelo de Mezclas Gaussianas, y la Máquina de Vector Soporte. De los cuales las redes neuronales son las más utilizadas en el reconocimiento de rostros, por lo que este trabajo se enfoca principalmente en los 2 últimos.

Capítulo 4 Presenta una explicación de lo que es el sistema propuesto usando los 3 métodos de extracción de características y el Modelo de Mezclas Gaussianas como clasificador y como verificador.

Capítulo 5 Presenta una explicación de lo que es el sistema propuesto usando los 3 métodos de extracción de características y la Máquina de Vector Soporte como clasificador y como verificador.

Capítulo 6 Presenta los resultados obtenidos con los dos sistemas propuestos, así como su comparación para poder evaluar el desempeño de cada uno.

Capítulo 7 Presenta las conclusiones que se obtuvieron al desarrollar este trabajo como también lo que sería tentativamente el trabajo futuro.

Capítulo 2

Estado del Arte

2.1. Sistemas Biométricos

Con la evolución de las tecnologías asociadas a la información, nuestra sociedad está cada día más conectada electrónicamente. Labores que eran realizadas por seres humanos son, gracias a las mejoras tecnológicas, realizadas por sistemas automatizados. Dentro de la amplia gama de posibles actividades que pueden automatizarse, aquella relacionada con la capacidad para establecer la identidad de los individuos ha cobrado importancia y como consecuencia directa, la biometría se ha transformado en un área emergente[4].

La biometría es la ciencia que se dedica a la identificación de individuos a partir de una característica anatómica o un rasgo de su comportamiento. Una característica anatómica tiene la cualidad de ser relativamente estable en el tiempo, tal como una huella dactilar, el rostro, patrones de la retina o el iris. Un rasgo del comportamiento es menos estable, pues depende de la disposición psicológica de la persona, por ejemplo la firma. No cualquier característica anatómica puede ser utilizada con éxito por un sistema biométrico. Para que esto así sea debe cumplir con las siguientes características: Universalidad, Unicidad, Permanencia y Cuantificación [5].

2.1.1. Modelo del proceso de identificación personal.

Cualquier proceso de identificación personal puede ser comprendido mediante un modelo simplificado. Este postula la existencia de tres indicadores de identidad que definen el proceso de identificación:

Conocimiento. La persona tiene conocimiento del indicador (por ejemplo: un código).

Posesión. La persona posee un objeto para la identificación (por ejemplo: una tarjeta).

Característica. La persona tiene una característica que puede ser verificada (por ejemplo: su rostro).

2.1.2. Características de un indicador biométrico

Un indicador biométrico es alguna característica con la cual se puede realizar biometría. Cualquiera que sea el indicador, debe cumplir los siguientes requerimientos [5]:

Universalidad. Cualquier persona posee esa característica.

Unicidad. La existencia de dos personas con una característica idéntica tiene una probabilidad muy pequeña.

Permanencia. La característica no cambia en el tiempo.

Cuantificación. La característica puede ser medida en forma cuantitativa.

2.1.3. Características de un sistema biométrico

Las características básicas que un sistema biométrico para identificación personal debe cumplir pueden expresarse mediante las restricciones que deben ser satisfechas. Ellas apuntan, básicamente, a la obtención de un sistema biométrico con utilidad práctica. Las restricciones antes señaladas apuntan a que el sistema considere:

El desempeño. Este se refiere a la exactitud, la rapidez y la robustez alcanzada en la identificación, además de los recursos invertidos y el efecto de factores ambientales y/u operacionales.

La aceptabilidad. La cual indica el grado en que la gente está dispuesta a aceptar un sistema biométrico en su vida diaria. Es claro que el sistema no debe representar peligro alguno para los usuarios y debe inspirar “confianza” a los mismos.

La fiabilidad. La cual refleja cuán difícil es burlar al sistema.

2.1.4. Panorama General de los Sistemas Biométricos

A continuación en la figura 2.1, se muestra una grafica que refleja el porcentaje de uso de cada uno de los sistemas biométricos más comunes.



Figura 2.1: Sistemas Biométricos.

El desarrollo de sistemas para reconocimiento o verificación de rostro ha sido un tópico de investigación activa durante la última década, debido al número de potenciales aplicaciones que estos tienen en muchos campos de la ciencia e ingenierías; los cuales requieren una identificación precisa del usuario o la verificación de la identidad del mismo. Dependiendo de su aplicación particular los sistemas para el rostro se dividen en: sistemas

de reconocimiento y sistemas de verificación, donde el sistema de reconocimiento tiene como función determinar la identidad del individuo dentro de un grupo de posibles personas, mientras que el sistema de verificación tiene como finalidad determinar si el hablante es quien realmente afirma ser.

2.2. Métodos de Extracción de Características

La etapa de extracción de características es una de las etapas más importantes en los sistemas de reconocimiento debido a que en esta etapa se extraen características del rostro en forma y tamaño correcto para dar una buena representación de la información característica de la persona y que servirá para el buen entrenamiento de los modelos de clasificación.

En las siguientes secciones se describirán algunos métodos de extracción de características de la imagen de un rostro.

2.2.1. Eigenfaces

El objetivo del reconocimiento por el método Eigenfaces[6] es extraer la información relevante de una imagen de rostro, codificar esta información tan eficientemente como sea posible y compararla con cada uno de los modelos almacenados de una base de datos. En términos matemáticos, el objetivo es encontrar los componentes principales de la distribución de rostros o los Eigenvectores de la matriz de covarianza de un conjunto de imágenes de rostro [7].

La idea de usar Eigenfaces fue desarrollada por Sirovich y Kirby en 1987 [8] para una representación eficiente de rostros usando análisis de componentes principales. Empezando con un conjunto de imágenes de rostro, calcularon un sistema mejor para la compresión de imágenes argumentando que en principio cualquier colección de imágenes de rostro puede ser aproximadamente reconstruida almacenando una pequeña colección de pesos para cada rostro y un pequeño conjunto de fotografías estándar (las Eigenfotografías). Los pesos que describen a cada rostro son encontrados por proyección de la imagen de rostro sobre cada

Eigenfotografía.

El método de Eigenfaces para el reconocimiento consiste de los siguientes pasos:

1. Adquirir un conjunto de imágenes de entrenamiento.
2. Calcular los Eigenfaces del conjunto de entrenamiento, almacenando solamente las M mejores imágenes con los más altos Eigenvalores. Estas M imágenes definen el “espacio de rostro”.
3. Calcular la correspondiente distribución del espacio de rostro M -dimensional para cada individuo conocido (imagen de entrenamiento), proyectando su imagen de rostro sobre el espacio de rostro.

Teniendo inicializado el sistema, los siguientes pasos son usados para el reconocimiento de nuevas imágenes de rostro:

1. Dada una imagen para ser reconocida, calcular un conjunto de pesos de los M Eigenfaces proyectando este sobre cada uno de los Eigenfaces.
2. Determinar si la imagen es un rostro, para esto se compara si la imagen es lo suficientemente cercana al espacio de rostro.
3. Si este es un rostro, clasificar el peso ya sea con un rostro conocido o uno desconocido.

El cálculo de los Eigenfaces es de la siguiente forma:

Se obtiene una imagen de rostro $\Gamma(x, y)$ de dos dimensiones $N \times N$. Una imagen puede también ser considerada como un vector de dimensión N^2 , de esta forma, una imagen de tamaño típico 256×256 se convierte en un vector de dimensión 65,536, en otras palabras, un punto en un espacio de dimensión 65,536.

La idea principal del análisis de componentes principales es encontrar el vector que mejor describa la distribución de imágenes de rostro sin el espacio de imagen entero. Estos vectores definen el sub-espacio de imágenes de rostro, llamado “espacio de rostro”. Cada vector es de longitud N^2 , lo cual describe una imagen $N \times N$ y es una combinación lineal de las imágenes de rostro originales. Debido a que estos vectores son los Eigenvectores de

la matriz de covarianza correspondiente a las imágenes de rostro originales y debido a que son parecidos a los rostros, se les llama “Eigenfaces”.

Dado un conjunto de imágenes de entrenamiento $\Gamma_1, \Gamma_2, \Gamma_3 \dots \Gamma_M$, el rostro promedio del conjunto se define como:

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (2.1)$$

Cada rostro se diferencia del promedio por el vector:

$$\phi_n = \Gamma_n - \Psi \quad (2.2)$$

La figura 2.2 muestra las imágenes resultantes de esta etapa.

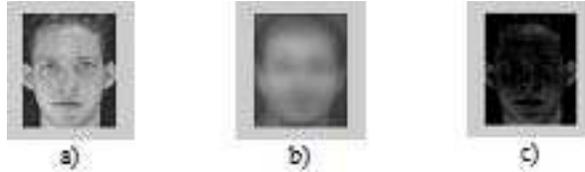


Figura 2.2: a)Rostro original. b)Imagen promedio. c)Diferencia del rostro respecto al promedio.

Posteriormente se calcula la matriz de covarianza usando la ecuación 2.3:

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T \quad (2.3)$$

donde:

$$A = [\phi_1 \phi_2 \dots \phi_M] \quad (2.4)$$

La matriz C es de dimensión $N^2 \times N^2$ y determinar los N^2 Eigenvectores y Eigenvalores es una tarea intratable para tamaños típicos de imágenes, por esta razón, necesitamos de un método factible para encontrar los Eigenvectores.

Si el número de puntos de datos en el espacio de imagen es menor que la dimensión del espacio total ($M < N^2$), habrá únicamente M-1, en lugar de N^2 Eigenvectores.

Consideremos los eigenvectores v_i de $A^T A$ tal que:

$$A^T A v_i = \mu_i v_i \quad (2.5)$$

Multiplicando ambos lados por A, tenemos:

$$AA^T Av_i = \mu_i Av_i \quad (2.6)$$

Sabemos que $C = AA^T$, entonces:

$$CAv_i = \mu_i Av_i \quad (2.7)$$

Entonces, AA^T y $A^T A$ tienen los mismos eigenvalores y sus eigenvectores están relacionados como sigue:

- AA^T puede tener hasta N^2 eigenvalores y eigenvectores.
- $A^T A$ puede tener hasta M eigenvalores y eigenvectores
- Los M eigenvalores de $A^T A$ corresponden a los M más largos eigenvalores de AA^T .

Por lo tanto,

$$u_l = \sum_{k=1}^M v_{lk} \phi_k = Av_l, \quad l = 1, \dots, M \quad (2.8)$$

Con este análisis los cálculos se reducen bastante, del orden del número de píxeles en las imágenes (N^2) al orden del número de imágenes en el conjunto de entrenamiento (M).

Una vez que los Eigenfaces han sido calculados, la imagen es proyectada en el “espacio de rostro” mediante una simple operación,

$$w_k = u_k^T (\Gamma - \Psi), \quad k = 1, \dots, M' \quad (2.9)$$

Los pesos forman un vector $\Omega^T = [w_1, w_2, \dots, w_M]$ que describe la contribución de cada Eigenfaces en la imagen de rostro de entrada. Por último, el método más simple para determinar cual de los rostros almacenados proporciona la mejor descripción de la imagen de rostro de entrada es mediante la distancia Euclidiana.

$$\varepsilon_k^2 = \|\Omega - \Omega_k\|^2 \quad (2.10)$$

donde Ω_k es un vector que describe la k-ésima clase de rostro.

2.2.2. Discriminantes Lineales de Fisher

Belhumeur [9] propusieron un método basado en los Discriminantes Lineales de Fisher (DLF) conocido como fisherfaces. Para un conjunto con C clases diferentes, DLF proyectan un subespacio lineal óptimo Φ de $C - 1$ dimensiones. El objetivo del método es maximizar el radio de las matrices de dispersión interclase. La matriz de dispersión interclase esta definida como:

$$S_B = \sum_{j=1}^C n^j (\bar{I}^j - \bar{I})(\bar{I}^j - \bar{I})^T \quad (2.11)$$

mientras que la matriz de dispersión intraclase se define como:

$$S_W = \sum_{j=1}^C \sum_{k=1}^{n^j} (I^k - \bar{I}^j)(I^k - \bar{I}^j)^T \quad (2.12)$$

donde $\bar{I} = \frac{1}{N} \sum_{k=1}^n I^k$ es la imagen promedio del conjunto, $\bar{I}^j = \frac{1}{n^j} \sum_{k=1}^{n^j} I^{k,j}$ es la imagen promedio de la j -ésima clase, n^j es el número de ejemplos en la j -ésima clase, $N = \sum_{j=1}^C n^j$ es el número de imágenes en el conjunto y C es el número de clases en el conjunto. El subespacio óptimo E_{optimo} se calcula de la siguiente manera:

$$E_{optimo} = \arg \max_E \frac{|E^T S_B E|}{|E^T S_W E|} = [e_1, e_2, \dots, e_{C-1}] \quad (2.13)$$

donde $[e_1, e_2, \dots, e_{C-1}]$ es el conjunto de los eigenvectores generalizados de S_B y S_W correspondientes a los $C - 1$ eigenvalores generalizados más altos $\lambda_i, i = 1, 2, \dots, C - 1$, esto es:

$$S_B E_i = \lambda_i S_W E_i \quad (2.14)$$

De esta manera, los vectores característicos P para cualquier consulta de imágenes de rostros I en el sentido más discriminante puede ser calculado como:

$$P = E_{optimo}^T I \quad (2.15)$$

Debido a que en la práctica S_W es singular, primero se reduce la dimensionalidad de la imagen de m^2 a K empleando PCA y posteriormente se aplica DLF al subespacio PCA para reducir las dimensiones a $C - 1$. En todos los experimentos reportados por Belhumeur,

fisherfaces presenta una tasa de error menor que eigenfaces. Sin embargo, muchos investigadores han indicado que aunque el método DLF logra el mejor rendimiento con los rostros del conjunto de entrenamiento, no generaliza adecuadamente a nuevos individuos[10].

2.2.3. Transformada Discreta de Walsh

Yoshida[11], propusieron un esquema de reconocimiento de rostros basado en la transformada de Walsh bidimensional, la cual es empleada para reducir las dimensiones de las imágenes de los rostros.

Una función de Walsh admite únicamente dos valores (1 y -1) y puede ser generada mediante el producto de Kronecker (denotado por \otimes) de la matriz de Hadamard H . La matriz de Hadamard H_2 de 2×2 se define como:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad (2.16)$$

Así mismo, la matriz de Hadamard de 4×4 puede ser calculada mediante el producto de Kronecker de dos matrices H_2 como se muestra a continuación:

$$H_4 = H_2 \otimes H_2 = \begin{bmatrix} H_2 & H_2 \\ H_2 & H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (2.17)$$

En general, la matriz de Hadamard de $2^k \times 2^k$ se obtiene como sigue:

$$H_{2^k} = H_2 \otimes H_{2^{k-1}} = H_2 \otimes H_2 \otimes \dots \otimes H_2 \quad (2.18)$$

Las características de frecuencia están expresadas por los cambios de signo en cada fila de la matriz de Hadamard. La función de Walsh se encuentra expresada en cada fila de H_N , donde N es el orden de la matriz de Hadamard. Así, la transformada discreta de

Walsh y su inversa se definen como:

$$V = \frac{1}{N} H_N B \quad (2.19)$$

$$B = H_N V \quad (2.20)$$

donde B es el vector de datos muestreados y V es la transformada discreta de Walsh de $B \cdot V$ se conoce como espectro de Walsh.

Por otra parte, la transformada bidimensional de Walsh y su inversa están definidas como:

$$F = \frac{1}{MN} H_M f H_N \quad (2.21)$$

$$f = H_M F H_N \quad (2.22)$$

donde f es la matriz de datos muestreados y F es la transformada bidimensional de f .

En el método propuesto por Yoshida, la imagen es dividida en un mosaico de $I \times I$ subareas, los valores de cada subarea se obtienen del promedio de los pixeles que la conforman. Posteriormente, el mosaico de subareas es dividido en 16 grupos constituidos por bloques de $J \times J$ subareas. La transformada bidimensional de Walsh es aplicada a cada grupo, consecuentemente, la mayor parte de los componentes tienden a concentrarse en las secuencias más bajas. Finalmente, sólo los componentes de baja frecuencia en el espectro de Walsh bidimensional de cada grupo son extraídos y unidos para formar los vectores característicos.

2.2.4. Transformada Discreta de Gabor (Filtros de Gabor)

Las texturas se utilizan ampliamente por el sistema visual humano para llevar a cabo tareas tales como la segmentación de escenas en distintos objetos y el análisis de la superficie de las geometrías. Problemas similares también deben ser resueltos en el contexto del procesamiento de imágenes. El análisis de texturas es un reto importante sin embargo

debido a la complejidad de la textura de los patrones y de las diferentes condiciones de iluminación que deben tenerse en cuenta.

La segmentación de texturas es un tema que ha recibido gran atención en los últimos años. Bovik[12] influyó en los 90's por la demostración de la fiabilidad de los sistemas de segmentación de texturas sobre la base de un multi-canal de enfoque utilizando funciones 2D de Gabor como filtros localizados espacialmente. También Jain[13] definió un conjunto de filtros de Gabor que cubren todo el espectro de frecuencia y se aproxima a una transformada wavelet de la imagen.

Las funciones de Gabor 2-D quedan determinadas por cuatro parámetros, dos que expresan su localización en el dominio espacial (x, y) y otros dos que expresan la frecuencia espacial de sintonía (F) y orientación (ϕ) [14], así esta se puede expresar como:

$$h(x, y) = g(x', y') \exp(2\pi j F x') \quad (2.23)$$

La señal elemental de Gabor bidimensional espacial, está en función de la respuesta Gaussiana bidimensional $g(x', y')$, la frecuencia espacial (F) y la rotación aplicada (ϕ). La respuesta Gaussiana bidimensional $g(x', y')$, puede expresarse mediante la siguiente ecuación:

$$g(x', y') = \left(\frac{1}{2\pi\lambda\sigma^2} \right) \exp \left[-\frac{(x/\lambda)^2 + y^2}{2\sigma^2} \right] \quad (2.24)$$

Las funciones de Gabor son funciones que operan en el conjunto de los números complejos, cuya parte real es la función de Gabor simétrica y la parte imaginaria es la función de Gabor asimétrica.

$$(x', y') = (x \cos \phi + y \sin \phi, -x \sin \phi + y \cos \phi) \quad (2.25)$$

$$h(x, y) = h_c(x, y) - j h_s(x, y) \quad (2.26)$$

$$h_c(x, y) = g(x', y') \cos(2\pi F x') \quad (2.27)$$

$$h_s(x, y) = g(x', y') \sin(2\pi F x') \quad (2.28)$$

donde $h_c(x, y)$ es la señal elemental de Gabor con componentes reales (simetría par) y $h_s(x, y)$ es la señal elemental de Gabor con componentes imaginarios (simetría impar).

Podemos concebir la información aportada por este par de funciones en cuadratura de fase como un vector bidimensional cuya magnitud informa el contraste de energía en un punto dado y cuya dirección especifica la fase de la energía. El contraste de energía es llamado también amplitud de la señal el cual, al representarlo en niveles de gris, muestra la respuesta de la imagen al par en cuadratura en función de la posición espacial, que es independiente de la fase. Este contraste de energía presenta una gran similitud con el comportamiento de las células complejas y proporciona una medida de la respuesta del canal, que es independiente del cambio de fase local. Al promediar cada uno de estos contrastes de energía resultantes de cada Banco de Filtros de Gabor, obtenemos los vectores características de la respuesta de la imagen en función de la posición y frecuencia espacial.

La información aportada por este par en cuadratura de fase corresponde al contraste de energía en un punto dado. El contraste de energía $M(x, y)$ de un par en cuadratura se obtiene mediante la ecuación:

$$M(x, y) = \sqrt{h_c^2 + h_s^2} \quad (2.29)$$

Esta función $M(x, y)$ presenta gran similitud con el comportamiento de las células complejas y proporciona una medida de la respuesta del canal, que es independiente del cambio de fase local. Al promediar cada una de estas amplitudes de la señal resultante, obtenemos los vectores característicos de la respuesta de la imagen:

$$M = \frac{\sum_{p=1}^B M_p(x, y)}{B} \quad (2.30)$$

Donde B es el número de bancos de filtros de Gabor.

En el método propuesto en [15, 16], se realiza un filtrado por regiones en el dominio espacial. Primero, se divide la imagen en bloques generando una malla de $I \times J$ bloques. Cada bloque de la malla tiene un punto central de localización o célula de campo simple en la posición espacial (x, y) . Las funciones Gaussianas bidimensionales se calculan en cada punto de localización. A cada uno de los campos receptivos se les aplican las frecuencias espaciales y los canales de orientación. Finalmente, mediante la ecuación 2.30 se reduce el número de elementos de salida del filtro para formar los vectores característicos.

2.2.5. Transformada Discreta Wavelet

Wavelets[17, 18] se han utilizado con éxito en el procesamiento de imágenes. Su capacidad para capturar localidades tiempo-frecuencia de información de la imagen que motiva su uso para la extracción de características. La descomposición de los datos en diferentes rangos de frecuencia nos permite aislar los componentes de frecuencia introducido por intrínsecas deformaciones debido a la expresión o factores extrínsecos (como la iluminación) en determinadas sub bandas. Los métodos basados en Wavelet reducen estas sub bandas variables, y se centran en la sub bandas que contienen la información más relevante para representar mejor los datos.

La transformada discreta wavelet mas comúnmente usada es la formulada por la matematica Belga Ingrid Daubechies en 1988[19]. Esta formula esta basada en el uso de relaciones recurrentes para generar progresivamente finos muestreos discretos de una wavelet madre; cada una es la mitad de la escala previa. En su trabajo, Daubechies deriva una familia de wavelets, la primera de las cuales es la wavelet de Haar. El interés en este campo ha explotado desde entonces, y muchas variaciones de la wavelet de Daubechies se han desarrollado.

La DWT de una señal x es estimada pasándola por una serie de filtros pasa bajas y pasa altas. Primero la señal pasa por un filtro pasa bajas con una respuesta impulso $g(n, m)$ obteniendo una convolucion de las dos. La señal es descompuesta simultáneamente usando el filtro pasa altas $h(n, m)$. Los coeficientes de detalle son las salidas de los filtros pasa altas y los coeficientes de aproximación son las salidas de los filtros pasa bajas. Es importante que ambos filtros se apliquen después de cada uno. Por lo tanto, la mitad de las frecuencias que tenemos ahora pueden ser removidas, la mitad de las señales pueden ser descartadas de acuerdo a la regla de Nyquist. Las salidas del filtro son:

$$Y_{low}(n, m) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x(n, m)g(2n - k, 2m - j) \quad (2.31)$$

$$Y_{high}(n, m) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x(n, m)h(2n - k, 2m - j) \quad (2.32)$$

donde g es un filtro pasa altas y h es un filtro pasa bajas.

La aplicación de la Transformada Discreta Wavelet a una imagen produce cuatro matrices de coeficientes wavelet, una de aproximación y tres de detalles (horizontal, vertical y diagonal). La matriz de coeficientes de aproximación contiene los componentes en bandas de baja frecuencia de la imagen original mientras que las matrices de detalles los componentes en bandas de alta frecuencia. En la figura 2.3 se presenta la descomposición de la imagen de un rostro en sus matrices de coeficientes de aproximación (CA), detalles horizontales (CD^h), detalles verticales (CD^v) y detalles diagonales (CD^d).

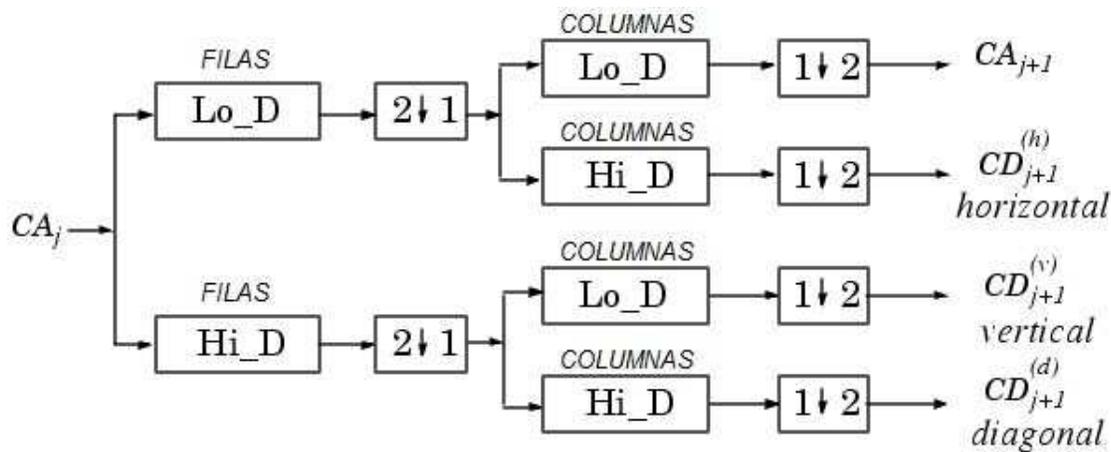


Figura 2.3: Niveles de descomposición wavelet.

Aquí la aproximación de los coeficientes son utilizados para caracterizar la imagen. Esta descomposición es repetida para más adelante incrementar la frecuencia de resolución y la aproximación de los coeficientes descompuestos con filtros pasa-bajas y pasa-altas para finalmente ser muestreada con el método “down-sampled”. Esto es representado como un árbol binario con nodos representando un sub-espacio con diferentes localidades en tiempo-frecuencia. El árbol es conocido como bancos de filtros. A cada nivel en el diagrama mostrado la señal es descompuesta en bajas y altas frecuencias. A pesar del proceso de descomposición la señal de entrada debe ser un múltiplo de $2n$ donde n es el número de niveles. Por ejemplo una señal con 32 muestras, un rango de frecuencias de cero a f_n y tres niveles de descomposición, 4 escalas de salida son producidas.

2.2.6. Eigenphases

Este es un método nuevo propuesto por Savvides[20] y consta de la combinación del Análisis de Componentes Principales y el Espectro de Fase de una imagen.

Oppenheim[21, 22] han demostrado que la información de fase de una imagen conserva la mayor parte de la inteligibilidad de una imagen. Su investigación también muestra que obteniendo sólo el espectro de fase de una imagen, uno puede reconstruir la imagen original hasta un factor de escala, por lo tanto, la información de fase es la más importante en la representación de una señal de 2D en el dominio de Fourier. Esto también es demostrado por un simple experimento, este se muestra en la figura 2.4.

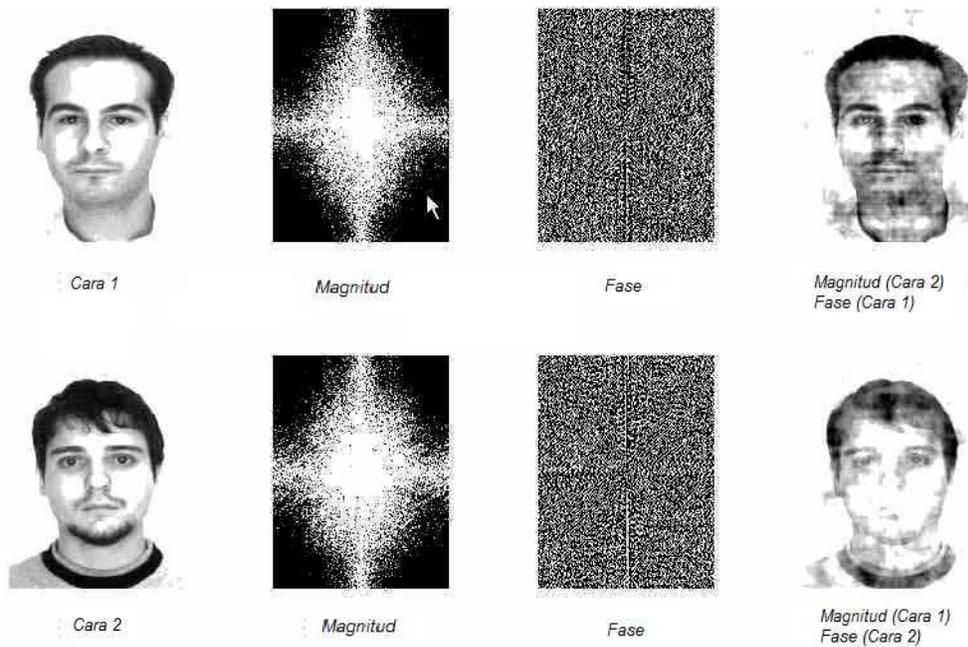


Figura 2.4: Experimento de Oppenheim.

Sin embargo, el desempeño de PCA en el dominio de la frecuencia por sí sola no constituye ningún avance, esto se debe a que los vectores propios obtenidos en el dominio de la frecuencia no son más que la transformada de Fourier de su dominio espacial. Empezamos esta derivación mediante la definición de la norma 2-D de la transformada discreta de Fourier (DFT). Teniendo en cuenta una entrada de una señal discreta 2-D $x[m, n]$ de la talla $M \times N$

denotamos su transformada de Fourier como $X[k, l]$ cuya transformada de Fourier par se define como sigue:

$$x[m, n] \Leftrightarrow X[k, l] \quad (2.33)$$

$$X[k, l] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m, n] \exp \frac{-i2\pi km}{M} \exp \frac{-i2\pi ln}{N} \quad (2.34)$$

$$x[m, n] = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X[k, l] \exp \frac{i2\pi km}{M} \exp \frac{i2\pi ln}{N} \quad (2.35)$$

donde $i = \sqrt{-1}$.

Si tenemos N imágenes de entrenamiento entonces la matriz de covarianza de la transformada de Fourier es obtenida por:

$$\begin{aligned} \widehat{\sum}_f &= \frac{1}{N} \sum_{i=1}^N \{F(x_i - \widehat{\mu})\} \{F(x_i - \widehat{\mu})\}^+ \\ &= F \widehat{\sum}_s F^{-1} \end{aligned} \quad (2.36)$$

como ya se a normalizado con PCA, los eigenvectores w_f de $\widehat{\sum}_f$ son obtenidas por:

$$F \widehat{\sum}_s F^{-1} w_f = \lambda w_f \quad (2.37)$$

multiplicando ambos lados por F^{-1} tenemos:

$$\widehat{\sum}_s F^{-1} w_f = \lambda F^{-1} w_f \quad (2.38)$$

ahora se formula el espacio en el dominio PCA y se nota que el espacio en el dominio de la matriz de covarianza es:

$$C_s = \widehat{\sum}_s \quad (2.39)$$

$$C_s v_s = \lambda v_s \quad (2.40)$$

$$\widehat{\sum}_s v_s = \lambda v_s \quad (2.41)$$

donde v_s es el espacio en el dominio de los eigenvectores.

Comparando la ecuacion 2.38 con la ecuacion 2.41 se puede observar que es una relacion entre el espacio y la frecuencia en el dominio de los eigenvectores (v_s y v_f) relacionados por una transformada de Fourier inversa como sigue:

$$v_s = F^{-1}v_f \quad (2.42)$$

2.3. Conclusiones

En este capitulo se explico en que consiste la biometría así tambien se analizaron diferentes algoritmos para la extracción de características, algunos muy usados como son eigenfaces o el uso de los filtros de Gabor, y otros mas nuevos como son el método de eigenphases. Y se pudo observar que a lo largo del tiempo se han desarrollado diferentes métodos para el reconocimiento de rostro pero ninguno asegura el 100% de reconocimiento ya que existen diferentes factores que impiden este objetivo. Cada uno de los métodos aquí mencionados tienen sus ventajas y desventajas, esto depende de las variaciones que se lleguen a presentar, como son el cambio de iluminación, las expresiones faciales o la oclusión parcial del rostro entre otras.

Capítulo 3

Clasificadores

El objetivo de un clasificador es asignar un nombre a un conjunto de datos correspondientes a un objeto o entidad concreta. Se define a un conjunto de entrenamiento como un conjunto de elementos, estando cada uno de ellos formado por una secuencia de datos correspondientes a un objeto concreto, uno de los cuales es la clase a la que pertenece. Un clasificador es un algoritmo que permite definir un modelo para cada clase, de tal forma que la clase a la que pertenece un elemento se puede calcular a partir de los valores de los datos que definen el elemento. Por lo tanto, de forma más práctica, el objetivo de un clasificador es asignar de forma lo más precisa posible una clase a nuevos elementos previamente no estudiados. Habitualmente se considera también un conjunto de prueba, que nos permite medir la precisión del modelo. La clase de cada elemento del conjunto de prueba es conocida y se utiliza para validar el modelo.

3.1. Redes Neuronales

Las redes neuronales o modelos conexionistas, están inspiradas en el comportamiento del cerebro humano. Esto significa, que cuando se desarrolla un modelo de red neuronal se trata de imitar el comportamiento del cerebro humano.

Una red neuronal está compuesta de un conjunto de elementos que imitan a los componentes neuronales biológicos, tan es así, que dichos elementos son unidades de proce-

samiento llamadas neuronas artificiales, los cuales tienen conexiones de entrada (dentritas) que provienen de otras neuronas y conexiones de salida (axón) hacia otras neuronas. Para poder obtener un buen desempeño, las neuronas de una red neuronal están masivamente interconectados con otras neuronas mediante la serie de conexiones tanto de entrada como de salida, las cuales tienen un cierto peso. Además las redes neuronales artificiales han jugado un papel importante para resolver problemas en distintas aplicaciones como predicción, optimización, etc, pero en particular han mostrado un rendimiento muy convincente en problemas de reconocimiento de patrones.

3.1.1. Modelo de una Neurona Artificial

El primer modelo de una neurona artificial fue la unidad de umbral binario propuesta por McCulloch y Pitts[23], la cual se ilustra en la figura 3.1.

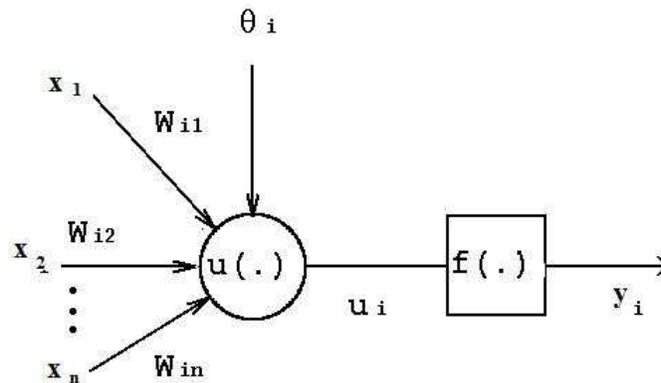


Figura 3.1: Modelo de neurona simple.

Este modelo de neurona se caracteriza por la descripción funcional de la red de conexiones y la activación de neuronas. Cada neurona recibe valores de entrada x_i , los cuales son propagados desde otras neuronas en la red a través de una red de conexiones unidireccionales. Cada neurona tiene asociada un peso sináptico denotado por w_{ij} , el cual determina el efecto de la j -ésima neurona sobre la i -ésima neurona; si el peso es positivo, la conexión se denomina excitatoria; si es negativo se denomina inhibitoria. Las entradas de la i -ésima neurona son acumuladas junto con una entrada externa θ_i (comunmente llamada bias) para obtener el valor neto u_i . Posteriormente, el valor neto u_i es transformado por

medio de una función de activación no lineal f para obtener y_i . Las funciones de activación más comúnmente utilizadas son: la función escalon, rampa, sigmoidea y gaussiana.

3.1.2. Características de una red neuronal

En general, las redes neuronales presentan tres principales características:

Aprendizaje. Una red neuronal tiene la capacidad de almacenar conocimiento mediante un proceso de aprendizaje (entrenamiento). Este conocimiento es almacenado por los pesos de las conexiones entre las neuronas que conforman a la red neuronal.

Generalización. Significa que se pueden obtener salidas razonables cuando se usan entradas diferentes a las utilizadas durante el proceso de entrenamiento.

Adaptabilidad. Significa que una red neuronal puede ser re-entrenada para funcionar adecuadamente ante cambios en su medio ambiente.

Hay diferentes arquitecturas de redes neuronales, entre las cuales podemos mencionar: redes alimentadas hacia delante de capa simple, redes alimentadas hacia delante multicapas, y redes recurrentes. Para más información acerca de la historia, la teoría y los diferentes tipos de redes neuronales artificiales favor de consultar [24, 25, 26].

Una de las arquitecturas más utilizadas para el reconocimiento de rostros es la red de perceptrón multicapas (MLP). Este tipo de arquitectura de red consta de tres componentes principales: una capa de entrada, una o más capas ocultas, y una capa de salida.

3.1.3. Red Neuronal Perceptrón Multicapa

El perceptrón multicapa es una red neuronal artificial (RNA) formada por múltiples capas como se muestra en la figura 3.2, esto le permite resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón (también llamado perceptrón simple). El perceptrón multicapa puede ser totalmente o localmente conectado. En el primer caso cada salida de una neurona de la capa i es entrada de todas las neuronas de la capa $i + 1$, mientras que el segundo, cada neurona de la capa i es entrada de una serie de neuronas (región) de la capa $i + 1$.

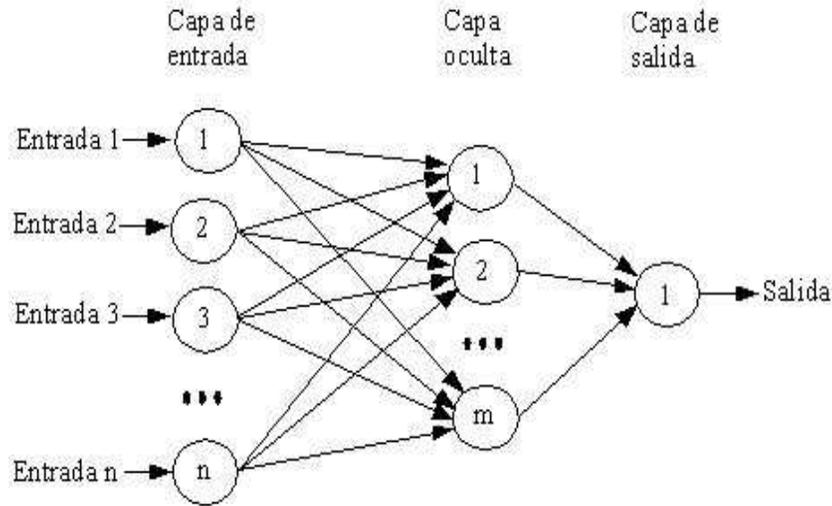


Figura 3.2: Red neuronal perceptrón multicapa.

Este tipo de redes comúnmente son entrenadas utilizando el algoritmo de retropropagación del error, el cual consta de dos fases. La primera, es una fase hacia delante, en la cual la información colocada en las neuronas de la capa de entrada (patrón) es propagada hacia delante a través de las capas ocultas y hasta la capa de salida, la cual genera en las neuronas que la conforman, la respuesta a la información dada como entrada. Durante esta fase, todos los pesos de las conexiones entre neuronas son utilizados. La segunda fase es conocida como fase hacia atrás, la cual consiste en modificar los valores de los pesos de acuerdo al error generado por las neuronas de la capa de salida.

Las neuronas de la capa oculta usan como regla de propagación, la suma ponderada de las entradas con los pesos sinápticos w_{ij} y sobre esa suma ponderada se aplica una función de transferencia de tipo sigmoide, que es acotada en respuesta. Similarmente en la retropropagación, el error en los nodos de las capas ocultas es proporcional a la sumatoria de los gradientes de los nodos de la capa siguiente pesados por los pesos de conexión.

La aparición de una función de activación del tipo sigmoide es debido a restricciones analíticas en los algoritmos de entrenamiento. Una función de activación de este tipo es:

$$y(p) = \frac{1}{1 + \exp(-p)} \quad (3.1)$$

donde p es la sumatoria de las entradas al nodo pesadas por los pesos de conexión y $y(p)$ la salida del nodo correspondiente:

$$p = \sum_{i=1}^n x_i w_i \quad (3.2)$$

Algoritmo de retropropagación

1. Inicializar los pesos aleatoriamente y con valores menores que 1.
2. Escoger un patrón del conjunto de entrenamiento, calcular la salida (para esto se avanza neurona por neurona, capa por capa, hasta llegar a la salida) y el error asociado en cada nodo de la capa de salida.
3. Adaptar los pesos: empezando con la capa de salida, y “modificando hacia atrás”, de la siguiente manera:

$$w_{ij}(k+1) = w_{ij}(k) + \alpha \delta_{ij} Y_{ij} \quad (3.3)$$

Para nodos de la capa de salida:

$$\delta_{pj} = Y_{pj}(1 - Y_{pj})(d_{pj} - Y_{pj}) \quad (3.4)$$

Para unidades en las capas escondidas:

$$\delta_{pj} = Y_{pj}(1 - Y_{pj}) \left(\sum_k \delta_{pk} w_{pk} \right) \quad (3.5)$$

4. Volver al paso 2. y repetir el proceso hasta que los pesos converjan (o sea el error entre las salidas deseadas y las obtenidas sea menor que un valor deseado).

Generalmente estos pasos (de 1 a 4) se realizan un número determinado de veces, llamadas épocas de entrenamiento y luego se observa si la red aprendió.

No existe un límite para fijar la cantidad de capas de un perceptron multiple, pero se ha demostrado que con una capa oculta y con el número suficiente de nodos, es capaz de solucionar casi cualquier problema. Si se agrega una capa oculta más, soluciona cualquier tipo de problemas y en forma más eficiente que con una sola.

3.2. Modelo de Mezclas Gaussianas (GMM)

Los Modelos de Mezclas Gaussianas pueden emplearse para representar funciones de densidad de probabilidad complejas, a partir de la marginalización de distribuciones conjuntas entre variables observadas y variables ocultas. El modelo de mezclas Gaussianas se basa en el hecho de que un número importante de distribuciones de probabilidad pueden ser aproximadas por medio de una suma pesada de funciones Gaussianas como se muestra en la figura 3.3. El uso de este clasificador ha destacado en el campo de reconocimiento de hablantes[27] obteniendo muy buenos resultados.

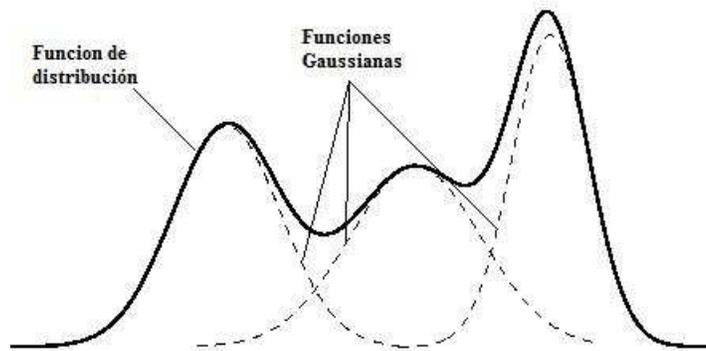


Figura 3.3: Aproximación de una función de distribución de probabilidad mediante una suma pesada de funciones Gaussianas.

Par a poder llevar acabo el desarrollo del modelo de mezclas gaussianas se deben tener en cuenta 3 puntos muy importantes:

1. Inicialización del modelo.
2. Desarrollo del modelo.
3. Evaluación del modelo

3.2.1. Inicializacion del modelo

Igualmente, los modelos de mezcla permiten el agrupamiento de datos. El algoritmo de “K-means” es un algoritmo que corresponde a un límite no probabilístico particular de

la estimación por máxima verosimilitud aplicado a mezclas de distribuciones gaussianas.

El problema consiste en identificar grupos de datos en un espacio multidimensional. Se supone un conjunto x_1, \dots, x_N de una variable aleatoria de D dimensiones en un espacio Euclidiano. Un grupo se puede pensar como un conjunto de datos cuya distancia entre sí es pequeña comparada con la distancia a los puntos fuera del grupo.

Se introducen un conjunto de vectores de D dimensiones μ_k , con $k = 1, 2, \dots, K$ en el que μ_k es el prototipo asociado al grupo k -ésimo. El objetivo consiste en encontrar una asignación de los datos observados a los grupos, así como un conjunto de vectores μ_k tal que se minimice la suma de los cuadrados de las distancias entre cada punto a su vector más cercano μ_k .

Por ejemplo: Inicialmente seleccione los M primeros vectores característicos como los centros iniciales, como se muestra en la figura 3.4, esto es

$$\mu_i = X_i \quad (3.6)$$

Seguidamente se agrega un vector mas y se obtiene la distancia entre el nuevo vector y los M centros, determinándose que el nuevo vector pertenece al centro con el cual su distancia se la menor. Posteriormente se calcula el nuevo centro promediando los elementos que pertenecen a ese centro. Así denotando por $X_{k,j}$ los vectores característicos pertenecientes al centro μ_k el nuevo centro estará dado por:

$$\mu_k = \frac{1}{N} \sum_{j=1}^N X_{k,j} \quad (3.7)$$

Este proceso se repite hasta que la distancia entre el k -ésimo centro en la iteración n y la $n + 1$ es menor a una constante dada de antemano.

Considerando la figura 3.4 se observa que, los primeros tres vectores se emplean como los centros iniciales. Seguidamente el se inserta se analiza el cuarto vector el cual tiene la menor distancia con respecto al centro x . Seguidamente se calcula el nuevo promediando los dos vectores pertenecientes al centro X . Seguidamente se analiza el vector 5, el cual tiene una distancia menor con respecto al centro O , el cual se modifica usando los vectores 1 y 5, como se muestra en la iteración segunda de la figura 3.4. Seguidamente se analiza el

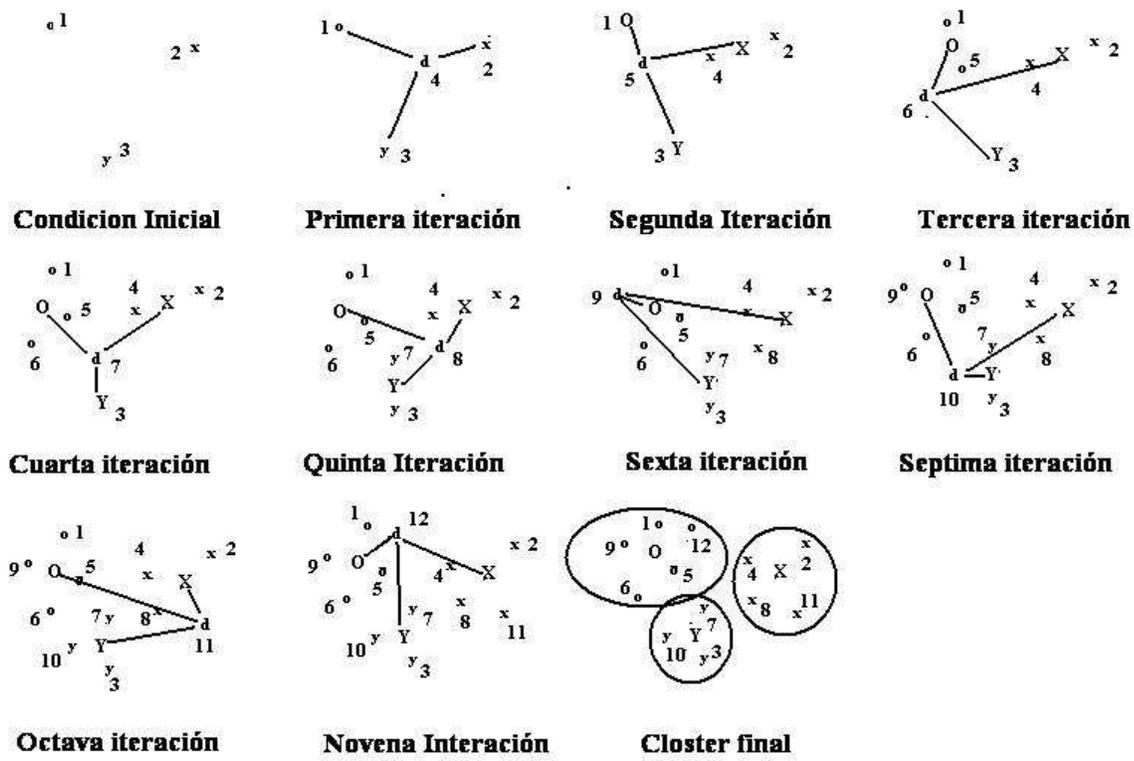


Figura 3.4: Ilustración del algoritmo de K-Means para M=3.

vector 6, el cual tiene una distancia mínima con respecto a centro O . Aquí el nuevo centro O se modifica usando los vectores 1,5 y 6. El proceso continua hasta la iteración novena, donde el centro O se calcula usando los vectores 1,5,6,9,12, el centro X se calcula usando los vectores 2,4,8,11, mientras que el Y se obtiene a partir de los vectores 3,7,10.

Una vez obtenidos los centros, la varianza de cada centro se obtiene usando la relación:

$$\sigma_k = \sum_{j=1}^{N_k} (\mu_k - X_{k,j})^2 \quad (3.8)$$

3.2.2. Desarrollo del modelo

Una densidad de mezclas Gaussianas es una suma pesada de M componentes de densidad, esto se muestra en la figura 3.5 y se obtiene por la siguiente ecuación [28]:

$$p(x|\lambda) = \sum_{i=1}^M p_i b_i(x) \quad (3.9)$$

Donde es un vector N -dimensional, $b_i(\vec{x}), i = 1, 2, \dots, M$, son los componentes de densidad y $p_i, i = 1, 2, \dots, M$, son los pesos de las mezclas. Cada componente de densidad es una D -variación de la función Gaussiana de la forma:

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\sigma_i|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (x - \mu_i)' \sigma_i^{-1} (x - \mu_i) \right\} \quad (3.10)$$

Donde $()'$ denota el vector transpuesto, μ_i denota el valor medio de N dimensiones y σ_i la matriz de covarianza la cual se supone diagonal; y p_i los pesos de la distribución los cuales satisfacen la relación:

$$\sum_{i=1}^M p_i = 1 \quad (3.11)$$

Así el modelo de la distribución se determina mediante el vector medio, la matriz de covarianza y los pesos de la distribución con lo cual el modelo se representa como:

$$\lambda = p_i, \mu_i, \sigma_i, \quad i = 1, 2, \dots, M \quad (3.12)$$

La estimación de los parámetros del sistema usando el algoritmo ML (Maximum Likelihood) busca encontrar los parámetros del sistema que aproximen de la mejor manera

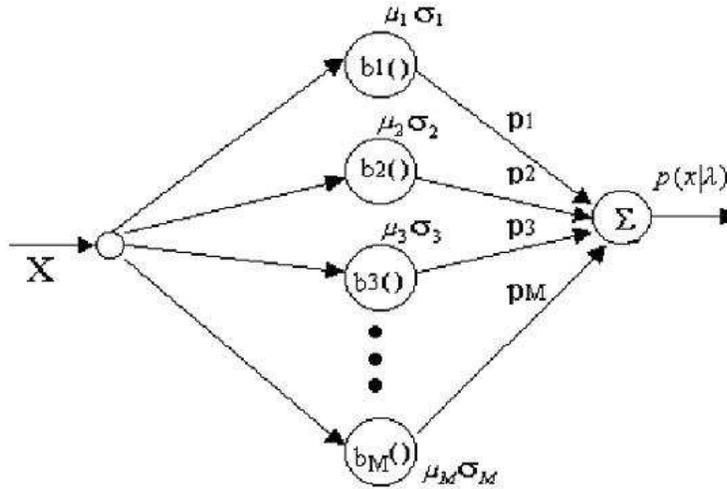


Figura 3.5: Modelo de Mezclas Gaussianas, GMM.

posible la distribución de las características del rostro bajo análisis, así se buscará encontrar los parámetros de λ que maximicen la distribución. Para una secuencia de T vectores de entrenamiento $X = x_1, \dots, x_T$, el GMM likelihood puede ser escrito como:

$$p(X|\lambda) = \prod_{t=1}^T p(X_t|\lambda) \quad (3.13)$$

Desafortunadamente la ecuación 3.13 es no lineal en relación a los parámetros de λ , por lo que no es posible realizar la maximización directamente; por lo que se debe utilizar un algoritmo iterativo llamado Baum-Welch. El algoritmo Baum-Welch es el algoritmo utilizado por HMM para estimar sus parámetros y tiene el mismo principio básico del algoritmo de Maximización de la Expectación (Expectation-Maximization EM), en el cual se parte de un conjunto inicial de parámetros $\lambda(r-1)$ y se estima un nuevo modelo $\lambda(r)$, donde r es denota la r -ésima iteración, de manera que:

$$p(X|\lambda(r)) \geq p(X|\lambda(r-1)) \quad (3.14)$$

Así, este nuevo modelo ($\vec{\lambda}$), se convierte en el modelo inicial para la siguiente iteración.

Cada T elementos se deben de actualizar los parámetros del modelo como sigue:

Pesos de la mezcla

$$p_i = \frac{1}{T} \sum_{t=1}^T p(i|X_{t+k}, \lambda) \quad (3.15)$$

Media

$$\mu_i = \frac{\sum_{t=1}^T p(i|X_{t+k}, \lambda) X_{t+k}}{\sum_{t=1}^T p(i|X_{t+k}, \lambda)} \quad (3.16)$$

Covarianza

$$\sigma_i = \frac{\sum_{t=1}^T p(i|X_{t+k}, \lambda) (X_{t+k} - \sigma_i)^2}{\sum_{t=1}^T p(i|X_{t+k}, \lambda)} \quad (3.17)$$

Para calcular la probabilidad a posteriori es obtenida por:

$$p(i|X_{t+k}, \lambda) = \frac{p_i b_i(X_{t+k})}{\sum_{j=1}^M p_j b_j(X_{t+k})} \quad (3.18)$$

3.2.3. Evaluación del modelo

Para llevar a cabo la evaluación del modelo considere que el sistema será empleado para identificación de R personas, los cuales están representados por los modelos $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_R$. El objetivo es entonces encontrar el modelo con la máxima a probabilidad a posteriori para una secuencia de observación dada. Formalmente, la persona identificada será aquella que satisfaga la relación:

$$\hat{R} = \arg \max Pr(\lambda_k|X), \quad k = 1, 2, \dots, R \quad (3.19)$$

Usando el teorema de Bayes, la ecuación 3.19 se puede expresar como:

$$\hat{R} = \arg \max \frac{p(X|\lambda_k)Pr(\lambda_k)}{p(X)}, \quad k = 1, 2, \dots, R \quad (3.20)$$

Suponiendo que la probabilidad de cada persona es igualmente probable, esto es $Pr(\lambda_k) = \frac{1}{R}$ y tomando en cuenta que $P(X)$ es igual para todos los modelos de hablantes, la ecuación 3.20 se simplifica a:

$$\hat{R} = \arg \max p(X|\lambda_k), \quad k = 1, 2, \dots, R \quad (3.21)$$

Sustituyendo $p(X|\lambda_k)$,

$$p(X|\lambda) = \prod_{t=1}^T p(X_t|\lambda_k) \quad (3.22)$$

en la ecuación 3.21 se obtiene:

$$\hat{R} = \arg \max \prod_{t=1}^T p(X_t|\lambda_k), \quad k = 1, 2, \dots, R \quad (3.23)$$

Finalmente usando logaritmos tenemos:

$$\hat{R} = \arg \max \sum_{t=1}^T \log_{10}(p(X_t|\lambda_k)), \quad k = 1, 2, \dots, R \quad (3.24)$$

donde $p(X_t|\lambda_k)$ esta dada por la ecuación 3.9, o sea por la salida del sistema mostrado en la figura 3.5.

3.3. Maquina de Vector de Soporte (SVM)

Las maquinas de vectores de soporte (SVM por sus siglas en ingles, "Support Vector Machine"), fueron desarrolladas por Vapnik (1995) [29], para el problema de clasificación pero la forma actual de SVM para regresión fue desarrollada en los laboratorios de AT&T por Vapnik [30]. SVM esta ganando gran popularidad como herramienta para la identificación de sistemas no lineales, esto debido principalmente a que SVM esta basado en el principio de minimización del riesgo estructural (SRM por sus siglas en ingles, "Structural Risk Minimization"), principio originado de la teoría de aprendizaje estadístico desarrollada por Vapnik en [29], el cual ha demostrado ser superior al principio de minimización del riesgo empírico (ERM por sus siglas en ingles "Empirical Risk Minimization"), utilizado por las redes neuronales convencionales. Algunas de las razones por las que este método ha tenido éxito es que no padece de mínimos locales y el modelo solo depende de los datos con mas información llamados vectores de soporte (SV por sus siglas en ingles, "Support Vectors"). Las grandes ventajas que tiene la SVM son:

- Una excelente capacidad de generalización, debido a la minimización del riesgo estructurado.

- Existen pocos parámetros a ajustar; el modelo solo depende de los datos con mayor información.
- La estimación de los parámetros se realiza a través de la optimización de una función de costo convexa, lo cual evita la existencia de un mínimo local.
- La solución de SVM es sparse, esto es que la mayoría de las variables son cero en la solución de SVM, esto quiere decir que el modelo final puede ser escrito como una combinación de un número muy pequeño de vectores de entrada, llamados vectores de soporte.

La aplicación más sencilla de esta técnica es el problema de clasificación binaria (solo hay definidas dos clases). La idea subyacente consiste en encontrar una hipótesis H que minimice la probabilidad de error empírico (la probabilidad de que H tenga un error en un conjunto de prueba seleccionado aleatoriamente).

En Vapnik [31] se prueba que minimizar el error empírico es equivalente a encontrar el hiperplano (figura 3.6) que se encuentra a la distancia máxima de las muestras de entrenamiento más cercanas para las dos clases. La clave del procedimiento consiste en establecer una correspondencia entre las muestras en el espacio de entrada y otro conjunto de vectores transformados en un espacio de dimensión mayor o igual, el llamado espacio de características. Para realizar esta transformación se utiliza una correspondencia previamente definida llamada kernel. Es en este espacio de características en el que se construye el hiperplano óptimo que separa las dos clases. Por lo tanto, el procedimiento termina con una función de decisión lineal en el espacio de características, donde los pesos se calculan como la solución de un problema cuadrático con restricciones (ver por ejemplo [32]).

En la figura 3.6 se muestra un ejemplo gráfico de aplicación de SVM. Se han considerado dos clases, etiquetadas como $A+$ y $A-$. Las clases definen dos grupos que por simplicidad se han considerado separables. Los vectores de soporte (marcados como a , b y c) son los que se utilizan para calcular el hiperplano que separa las dos clases. En el dibujo se indica también el margen, distancia del hiperplano a los Vectores de Soporte.

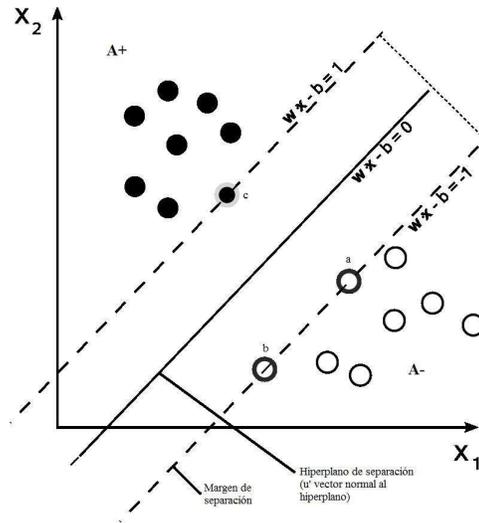


Figura 3.6: Ejemplo de SVM.

Estos algoritmos son en la actualidad el centro de atención dentro de el campo de reconocimiento de patrones, y se han aplicado en multitud de casos: reconocimiento de escritura a mano [33]; reconocimiento de objetos [34] o categorización de texto [35].

3.3.1. Clasificación mediante SVM

Supóngase el caso más sencillo de reconocimiento de patrones: dos clases linealmente separables. Sea un conjunto de un número m de ejemplos etiquetados $D = (x_i, y_i)_{i=1}^m$ donde se puede suponer que la etiqueta va a tomar uno de dos valores posibles (solo disponemos de dos clases) que, por simplificar, designaremos -1 y 1 , es decir, $y_i \in \{-1, 1\}$. Para este conjunto se quiere determinar, entre todos los clasificadores lineales que separan los datos, aquel que tiene el menor error de generalización. El error de generalización mide la capacidad de la red de dar una respuesta correcta ante patrones que no han sido empleados en su entrenamiento. Intuitivamente, una buena elección consiste en el hiperplano que obtiene el mayor margen entre las dos clases, donde el margen entre las dos clases, donde el margen se define como la suma de los cuadrados de las distancias del hiperplano a los puntos más cercanos de las dos clases.

Si las dos clases no son separables linealmente todavía se puede encontrar un hi-

perplano que maximice el margen y que minimice una cantidad proporcional al número de errores de clasificación. La compensación entre el margen y el error de clasificación se controla por una constante C positiva, seleccionada previamente. En este caso como se puede mostrar que la solución a este problema es un clasificador lineal:

$$f(x) = \text{signo}\left(\sum_{i=1}^m \lambda_i y_i x^T x_i + b\right) \quad (3.25)$$

cuyos coeficientes λ son la solución del siguiente problema, formulado en notación matricial:

Minimizar,

$$W(\Lambda) = -\Lambda^T \mathbf{1} + \frac{1}{2} \Lambda^T D \Lambda \quad (3.26)$$

Sujeto a,

$$\Lambda^T y = 0 \quad (3.27)$$

$$\Lambda - C \mathbf{1} \leq 0 \quad (3.28)$$

$$-\Lambda \leq 0 \quad (3.29)$$

Donde $(\Lambda)_i = \lambda_i$, $(\mathbf{1})_i = 1$ y $D_{ij} = y_i y_j x^T x_j$. Se observa que un número pequeño de coeficientes lambda son no nulos y como cada coeficiente se corresponde con uno de los ejemplos, significa que la solución está determinada por los ejemplos asociados con los coeficientes no nulos. Estos ejemplos se llaman Vectores Soporte, y son los únicos relevantes para la solución del problema: el resto de datos ejemplo pueden ser eliminados del conjunto de entrenamiento, no alterándose por ello la solución. Intuitivamente, los Vectores Soporte corresponden a los datos del conjunto de ejemplos que están en la frontera entre las dos clases. Su número es habitualmente bajo y Vapnik [31] demostró que es proporcional al error de generalización del clasificador.

Como no es muy frecuente que los problemas reales sean resolubles mediante clasificadores lineales, la técnica debe extenderse para permitir superficies no lineales. Esto se realiza fácilmente al proyectar el conjunto inicial de variables x en un espacio de dimensión mayor (llamado espacio de características) dado por:

$$x \in \mathfrak{R}^n \Rightarrow z(x) \equiv (\phi_1(x), \dots, \phi_n(x)) \in \mathfrak{R}^n \quad (3.30)$$

Y reformular el problema del clasificador lineal en el espacio de características. La solución tendrá la forma:

$$f(x) = \text{signo} \left(\sum_{i=1}^m \lambda_i y_i z^T(x) z(x_i) + b \right) \quad (3.31)$$

Y por lo tanto será no lineal en el espacio de las variables iniciales. Llegados a este punto surgen dos problemas:

1. La elección de las características $\phi_i(x)$, que debe realizarse de forma tal que lleve a una clase de superficies de decisión “rica”
2. El calculo del producto escalar $z^T(x)z(x_i)$ lo cual puede ser computacionalmente prohibitivo si el numero n de características es muy grande.

Una posible solución a este problema consiste en permitir que n crezca hasta el infinito y realizar la siguiente seleccion de las funciones, $\phi_i(x) : z(x) \equiv (\sqrt{\alpha_1}\psi_1(x), \dots, \alpha_i\psi_i(x), \dots)$. Donde α_i y ψ_i son los autovalores y las autofunciones de un operador cuyo nucleo o kernel $K(x, y)$ es una función simetrica y definida positiva. Con esta elección el producto escalar en el espacio de características se presenta de forma sencilla porque se obtiene:

$$z^T(x)z(y) = \sum_{i=1}^{\infty} \alpha_i \psi_i(x)\psi_i(y) = K(x, y) \quad (3.32)$$

donde la última igualdad se obtiene del teorema Mercer-Hilbert-Schmidt[36] para funciones definidas positivas. El problema que acontinuacion se debe resolver es el mismo que el planteado en el caso anterior, pero ahora la matriz D tiene por elementos $D_{ij} = y_i y_j K(x_i, x_j)$. Como consecuencia de esta elección, el clasificador SVM tendrá la forma:

$$f(x) = \text{signo} \left(\sum_{i=1}^m \lambda_i y_i K(x, x_i) + b \right) \quad (3.33)$$

Las elecciones de kernel mas habituales son:

- Kernel lineal

$$k(x, y) = x^T y \quad (3.34)$$

- Kernel polinomial

$$k(x, y) = (x^T y + c)^q \quad (3.35)$$

donde c es una constante y q es un entero positivo.

- Kernel Gaussiano

$$k(x, y) = \exp\left(\frac{-\|x - y\|^2}{2\sigma^2}\right) \quad (3.36)$$

Los resultados teóricos sugieren que la eficacia de las máquinas de vector de soporte se debe a su capacidad de encontrar reglas que clasifiquen objetos con un grado de confianza alto y a su capacidad para evitar el sobre-entrenamiento. En efecto, resulta frecuente que algunos algoritmos encuentren reglas complejas que clasifican perfectamente los datos del conjunto de entrenamiento; pero estas reglas pueden ser completamente inútiles para clasificar correctamente nuevas muestras, ya que están tan relacionadas con el conjunto de entrenamiento que no tienen capacidad de generalización (este efecto se conoce como "sobre-entrenamiento"). Las redes neuronales son capaces de clasificar correctamente casi cualquier conjunto de entrenamiento, pero pueden caer en un sobre-entrenamiento en el que no puedan generalizar correctamente la clasificación.

Las máquinas de vector de soporte han sido utilizadas para la clasificación de varios tipos de patrones; también en verificación facial. Por ejemplo, G. Guo et al. Cambridge[37] utilizan los SVM para clasificar los sujetos de la base de datos de Olivetti Cambridge. Procesan las imágenes con un análisis de componentes principales y se obtiene un porcentaje de error del 3%, considerando una estructura en forma de búsqueda binaria para clasificar los 40 individuos de la base de datos. En [38] se utiliza una base de datos de 295 sujetos, con un resultado de reconocimiento correcto del 98% (en este caso, los cambios de iluminación del conjunto de prueba también están presentes en el conjunto de entrenamiento).

3.4. Conclusiones

Se estudiaron 3 clasificadores, las Redes Neuronales, Modelo de Mezclas Gaussianas, y la Máquina de Vector Soporte, de las cuales la que destaca más a lo largo de la historia son las redes neuronales ya que se pueden utilizar para cualquier clase de clasificación o

verificación, en contraste el Modelo de Mezclas Gaussianas es un método probabilístico que trata de aproximar una señal a través de la suma pesada de Gaussianas, la ventaja de este clasificador con respecto a los otros dos es que el modelo para cada patrón es independiente de los demás, y por último la Máquina de Vector Soporte que es un discriminante lineal que trata de separar las clases por medio de los vectores soporte logrando así la clasificación de los patrones.

El desempeño de el Modelo de Mezclas Gaussianas y la Máquina de Vector Soporte se podrá observar en los próximos capítulos cuando se utilicen como clasificadores y verificadores usando diferentes métodos de extracción de características.

Capítulo 4

Sistema Propuesto

4.1. Introducción

En el presente capítulo se describen las técnicas desarrolladas para el reconocimiento y la verificación de rostros, tomando como datos de entrada una de las técnicas de extracción de características (Filtros de Gabor, Transformada Discreta Wavelet, o el método de Eigenphases) y tomando como clasificadores al Modelo de Mezclas Gaussianas y la Máquina de Vector de Soporte. El sistema propuesto se divide en dos módulos: el módulo de entrenamiento y el módulo de reconocimiento o verificación, según sea el caso.

La primera etapa se divide en 3 fases:

1. Pre procesamiento: consiste en la localización del rostro así como el recorte del mismo.
2. Extracción de características: se emplea uno de los métodos ya mencionados para obtener los vectores con los cuales se entrenara el GMM.
3. Clasificador: se utilizan los vectores obtenidos para calcular el modelo de la persona a través del GMM o SVM.

El segundo módulo se divide también en 3 fases, donde las 2 primeras son las mismas que en la etapa de entrenamiento, solo que ahora en la tercera fase en lugar de entrenar al GMM o al SVM para la obtención del modelo, estos se utiliza como clasificadores y como verificadores.

A continuación se describen una por una las fases del sistema propuesto.

4.2. Diagrama del Sistema

En esta sección se hablara mas a detalle de cada una de las etapas del sistema propuesto, la figura 4.1 muestra un diagrama a bloques del sistema de reconocimiento o verificación.

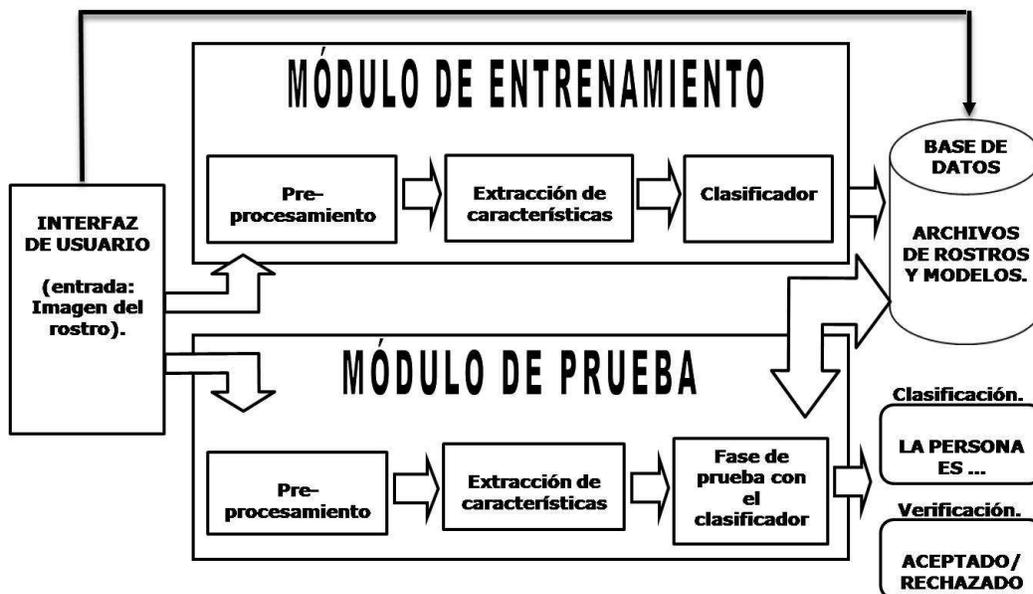


Figura 4.1: Sistema Propuesto.

4.2.1. Módulo de Entrenamiento

Pre-procesamiento

En esta etapa básicamente se busca el rostro en la imagen y se recorta para evitar así tener información inútil en la imagen a procesar, todas las imágenes deben de quedar de un tamaño específico, en este caso se toma una imagen de 768×576 y se recorta lo mas posible la parte blanca para después redimensionarla obteniendo una imagen de 192×288 . La figura 4.2 muestra una imagen original y una ya recortada.

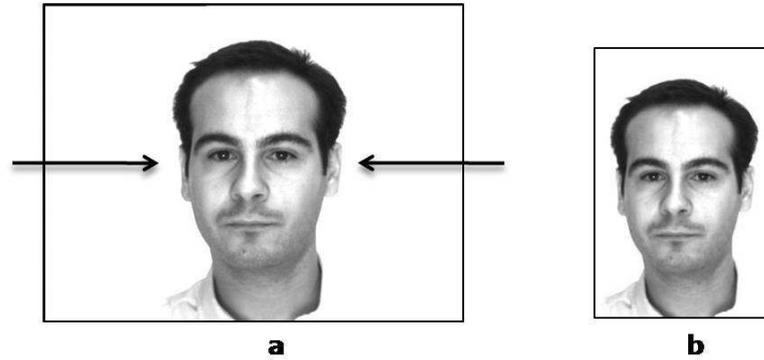


Figura 4.2: a) Imagen Original. b) Imagen recortada.

La base de datos utilizada es “AR Face Database” [39], la cual cuenta con 120 personas y con 78 imágenes de cada una con variaciones de iluminación, expresiones faciales y oclusión parcial del rostro, a todas las imágenes se les aplica el recorte y el redimensionamiento para así obtener las imágenes a las cuales se les aplicaran los métodos de extracción de características.

Extracción de características.

En esta sección se explican los tres métodos de extracción de características utilizados en este trabajo los cuales son los Filtros de Gabor, la Transformada Discreta Wavelet y el método de Eigenphases, los cuales son mencionados en el Capítulo 2, por lo que aquí solo se mostrara el procedimiento que se llevo a cabo para la implementación de estos métodos.

Filtros de Gabor

Para la aplicación de este método de extracción de características del cual se hizo mención en el capítulo 2 apartado 2.2.4 se realizo un filtrado por regiones en el dominio espacial. La malla empleada en cada una de las imagenes fue de 12×8 . Cada recuadro de la malla contara con un punto central de localización al cual se le aplicaran los filtros de Gabor, estos puntos centrales de localización siempre se encontraran en la posición espacial (x, y) , donde $x = 0$ y $y = 0$, como se muestra en la figura 4.3.

Las fases de modulación sinusoidales (ϕ) que se utilizaron son los siguientes: $0, \pi/9,$

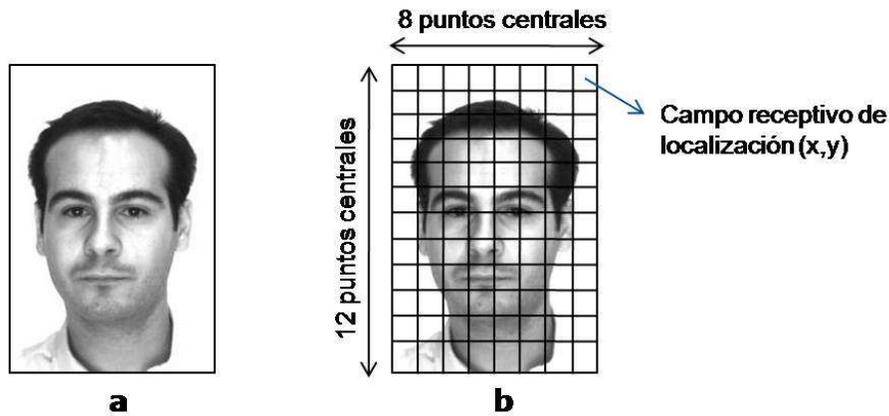


Figura 4.3: a) Imagen Original. b) Imagen con la malla.

$2\pi/9$, $\pi/3$, $4\pi/9$, $5\pi/9$, $2\pi/3$, $7\pi/9$, y $8\pi/9$. Se seleccionaron seis canales de frecuencia espacial: $\pi/2$, $\pi/4$, $\pi/8$, $\pi/16$, $\pi/32$, $\pi/64$. Una vez obtenidos los puntos de localización, se calculan las funciones Gaussianas bidimensionales en cada uno. A cada uno de los campos receptivos donde ya fueron calculadas las funciones Gaussianas, se les aplica las frecuencias espaciales y los canales de orientación, dando como resultado 54 filtros de Gabor en cada una de las posiciones espaciales (5184 Filtros de Gabor en toda la imagen).

Con la matriz resultante de cada posición de 12×8 elementos formamos un vector horizontal de 96 elementos, para cada canal de frecuencia y orientación espacial. El siguiente paso es agrupar estos vectores en una matriz general de 54×96 , la cual es la representación bidimensional resultante de los 54 bancos de filtros bidimensionales de Gabor. La matriz de 54×96 elementos al ser graficada en tonos de gris representa a los contrastes de energía en cada uno de los campos receptivos de localización donde fue aplicado el filtro bidimensional de Gabor.

Por último mediante la ecuación 2.30 reducimos el número de elementos de salida del filtro de 5184 a 96 elementos, estos 96 elementos representan los valores de los vectores característicos, los cuales son similares para el mismo individuo como se muestra en la figura 4.4. Y al comparar con la figura 4.5 muestra que son diferentes entre individuos.

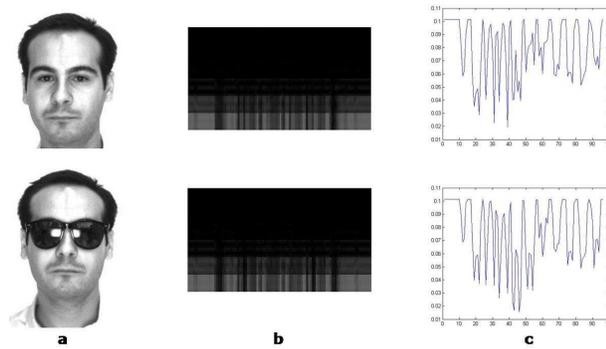


Figura 4.4: a) Imagen Original. b) Contrastes de Energia. c) Vector caracteristico.

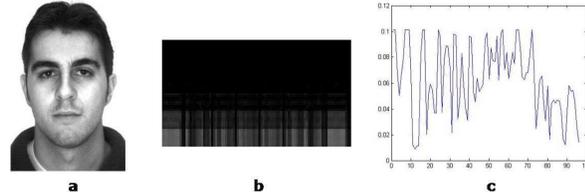


Figura 4.5: a) Imagen Original. b) Contrastes de Energia. c) Vector caracteristico.

Transformada Discreta Wavelet

Para la aplicación de este método de extracción de características del cual se hizo mención en el capítulo 2 apartado 2.2.5 se aplicó la Transformada Discreta Wavelet de Daubechies la cual fue propuesta por Ingrid Daubechies, una de las más importantes precursoras de las wavelets, ya que desarrolló lo que se conoce como “wavelets ortonormales de soporte compacto”, con lo cual hizo viable el análisis discreto de las wavelets.

Los componentes en bandas de baja frecuencia contribuyen a la descripción global de la imagen, la cual se cree contiene la información más relevante para la diferenciación entre clases. Por tal motivo, la extracción de características por medio de este método se lleva a cabo mediante la descomposición recursiva del rostro, de donde los coeficientes en las bandas de más baja frecuencia son extraídos para formar los vectores característicos.

En la figura 4.6 se muestra la descomposición en 4 niveles de un rostro, el vector característico es derivado de los componentes de baja frecuencia del cuarto nivel de descomposición. La descomposición recursiva de la imagen de un rostro conduce a la reducción

de las dimensiones en un cuarto de su tamaño por cada nivel de descomposición.

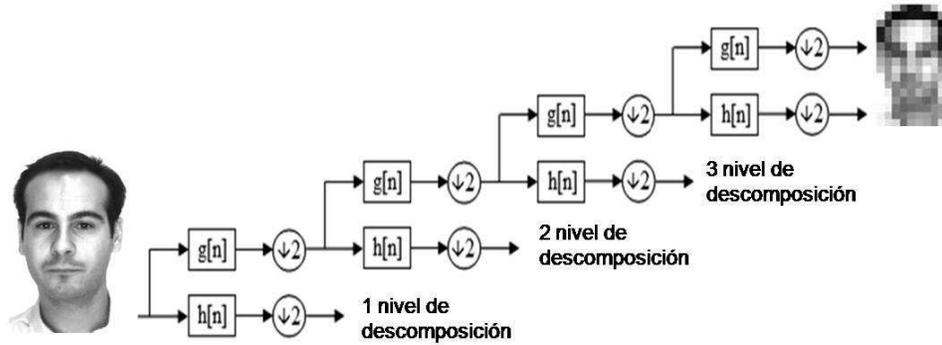


Figura 4.6: 4 niveles de descomposición wavelet.

La comparación entre los vectores característicos de 3 imágenes de un mismo rostro con variaciones de iluminación, expresión facial y oclusión parcial del rostro se ilustra en la figura 4.7. Se observa la estabilidad de los vectores característicos derivados y la enorme reducción de las dimensiones en la representación del rostro. De esta manera, se comprueba la robustez de la representación wavelet contra diferentes tipos de variación.

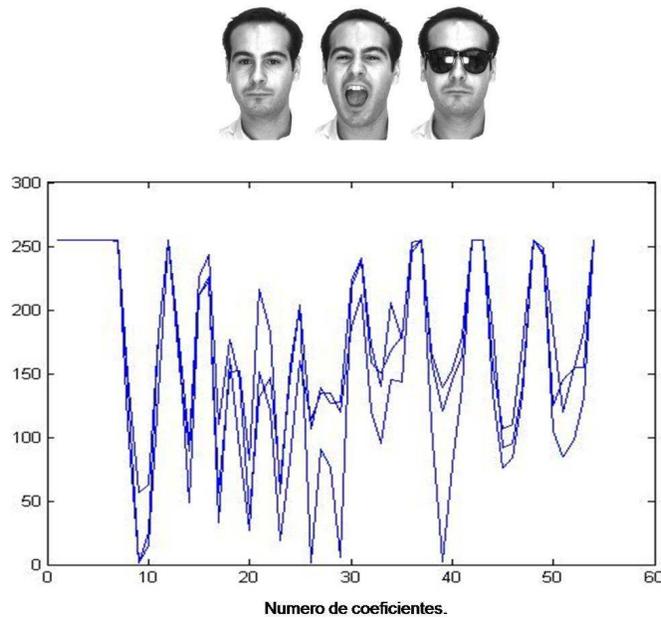


Figura 4.7: Variación entre vectores característicos.

Eigenphases

Para la aplicación de este método de extracción de características del cual se hizo mención en el capítulo 2 apartado 2.2.6 se tendrá que redimensionar la imagen de 192×288 a 48×36 esto se hace ya que se usará el método de Análisis de Componentes Principales (PCA) para la extracción de características, cabe mencionar que al método de Eigenphases original[20] se le aplicaron algunas variantes en este trabajo.

Después de redimensionar la imagen se obtendrá el espectro de fase de toda la imagen, ya que como se menciona en el capítulo 2 este retiene más información de la imagen original, en esta parte es donde se aplicaron las diferentes variantes las cuales son:

- Normalizar la imagen completa y después extraer el espectro de fase de toda la imagen.
- Normalizar la imagen localmente con ventanas de 3×3 y después extraer el espectro de fase de la imagen completa.
- Normalizar la imagen localmente con ventanas de 6×6 y después extraer el espectro de fase de la imagen completa.
- Normalizar la imagen y extraer el espectro de fase localmente con ventanas de 3×3 de cada imagen.
- Normalizar la imagen y extraer el espectro de fase localmente con ventanas de 6×6 de cada imagen.

Para obtener la normalización local de la imagen, la norma de las regiones locales es normalizada a 1. Se toma una región local $I(x)$ donde x es la posición central de la región local, la norma local está dada por:

$$I(x)' = \frac{I(x)}{\|I(x)\|} \quad (4.1)$$

donde $\|I(x)\|$ es la norma de la región local. Todas las imágenes deben de ser normalizadas antes de entrenar.

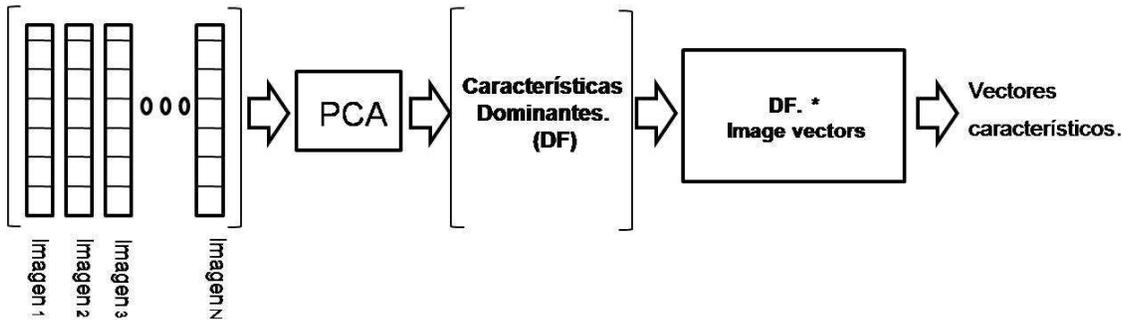


Figura 4.8: Esquema para la extracción de características usando PCA.

Posteriormente se aplicará un procedimiento como el que se muestra en la figura 4.8, en el cual los vectores obtenidos después de aplicar cualquiera de las variantes antes mencionadas con

En el cual la matriz obtenida después de aplicar cualquiera de las variantes antes mencionadas contando el método original de Eigenphases, se convertirán en un vector columna, para posteriormente construir una matriz con los vectores columna de todos los rostros que se utilizarán para el entrenamiento del clasificador.

Cuando ya tenemos la matriz con todos los vectores columna de entrenamiento aplicaremos el método de Análisis de Componentes Principales (PCA), los pasos a seguir son los siguientes:

1. Obtener alguna información.
2. Calcular la media.
3. Calcular la matriz de covarianza.
4. Calcular los vectores propios y los valores propios de la matriz de covarianza.
5. Escoger el número de componentes y formar un vector de características.
6. Calcular los vectores característicos.

Para más información acerca del Análisis de Componentes Principales consultar [40].

Una vez obtenida la matriz principal obtenida por PCA se calcula el producto de esta con cada uno de los vectores columna de los rostros obteniendo como resultado el vector característico que se utilizara para el entrenamiento y para la prueba de los clasificadores.

clasificadores

En esta sección se explican los dos clasificadores utilizados en este trabajo los cuales son el Modelo de Mezclas Gaussianas (GMM) y la Maquina de Vector de Soporte (SVM), los cuales son mencionados en el Capitulo 3, por lo que aquí solo se mostrara el procedimiento que se llevo a cabo para la implementación de estos clasificadores.

Entrenamiento del GMM

Para llevar a cabo el entrenamiento de este clasificador del cual se hizo mención en el capitulo 3 apartado 3.2 se utilizo el toolbox llamado stprtool (statistical Pattern Recognition Toolbox) la cual contiene el Modelo de Mezclas Gaussianas.

La figura 4.9 muestra un esquema general de como se entrena este clasificador.

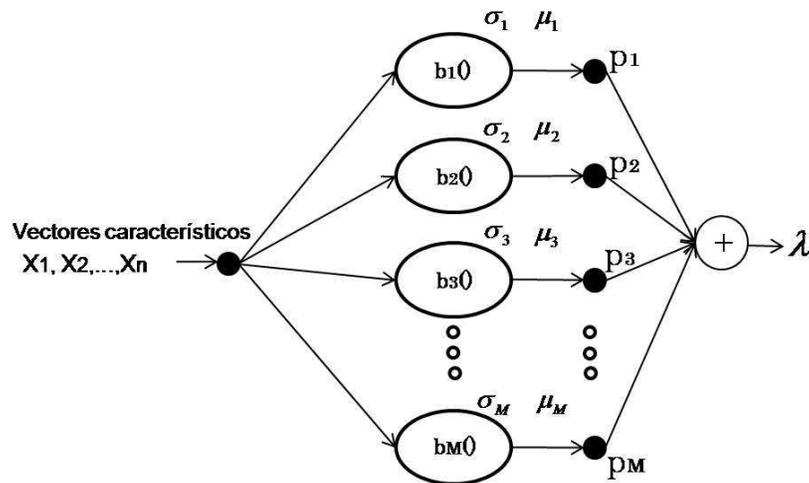


Figura 4.9: Esquema general para el entrenamiento del GMM.

donde x_1, x_2, \dots, x_n son los vectores característicos de una persona que se utilizaran para el entrenamiento del GMM, n es el numero de muestras a utilizar por persona, $\sigma_1, \sigma_2, \dots, \sigma_M$

son las matrices de covarianza del modelo y $\mu_1, \mu_2, \dots, \mu_M$ son los valores medios del modelo y p_1, p_2, \dots, p_M son las probabilidades de cada mezcla, estos tres elementos son los que constituyen el modelo final, M es el numero de mezclas Gaussianas que se utilizaran para generar el modelo, y por ultimo λ es el modelo final de la persona.

Básicamente el algoritmo de mezclas Gaussianas es iterativo hasta que cumpla con la condición de la ecuacion 3.14. Hasta entonces el toolbox regresara el modelo para las características del rostro con el cual se entreno, este modelo como ya se menciona esta compuesto de una matriz de valores medios de cada Gaussiana, un conjunto de matrices de covarianza igual al numero de Gaussianas y del tamaño del vector de cada Gaussiana, y la mezcla de pesos de cada una de las Gaussianas, todo esto se representa por λ, μ, p , este será el modelo que se utilizara para la tarea de reconocimiento o verificación que se vera mas adelante en el modulo de prueba.

Entrenamiento de la SVM

Para llevar a cabo el entrenamiento de este clasificador del cual se hizo mención en el capitulo 3 apartado 3.3 se utilizo la herramienta llamada svm-mat-2.84-1.

La figura 4.10 muestra un esquema general de como se entrena este clasificador.

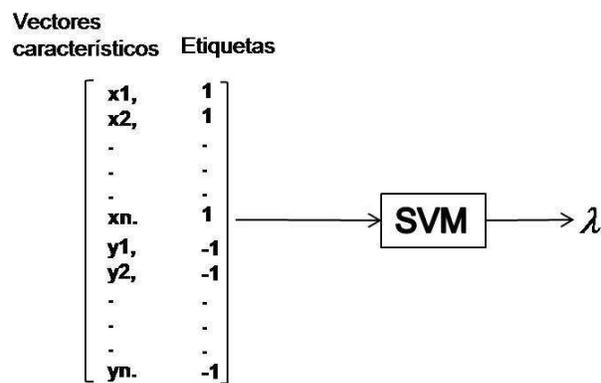


Figura 4.10: Esquema general para el entrenamiento del SVM.

donde $x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n$ son los vectores característicos de una persona que se utilizaran para el entrenamiento del SVM, n es el numero de muestras a utilizar por persona,

y las etiquetas son para diferenciar una clase de otra en este caso son dos clases ya que se utiliza un modelo de SVM una contra todas (one vs all) en donde 1 es el rostro correcto y -1 es un rostro intruso, posteriormente la SVM es entrenada y obtendremos λ que será el modelo final que utilizaremos para la tarea de reconocimiento o verificación.

Este clasificador nos proporcionara un modelo que esta compuesto del número de clases (en este caso 2), el número de vectores de soporte y sus valores, con esto será suficiente para poder hacer satisfactoriamente la tarea de clasificación y verificación.

4.2.2. Modulo de Prueba

En este modulo se siguen los mismos pasos para la etapa de pre-procesamiento y extracción de características, lo único que cambia es la parte del clasificador ya que ahora solo se hará la tarea de reconocimiento y verificación con GMM y con SVM, donde hay que tener en cuenta que el sistema de reconocimiento tiene como función determinar la identidad del individuo dentro de un grupo de posibles personas, para lo cual se tiene que comparar el rostro de cada persona con todos los modelos que existen en la base de datos, para poder así determinar a que modelo pertenece la imagen del rostro y saber cual es la identidad de esa persona, mientras que el sistema de verificación tiene como finalidad determinar si el hablante es quien realmente afirma ser, para lo cual se tiene que usar un método alternativo de reconocimiento, ya sea una tarjeta inteligente, un password, etc., el cual le dirá al sistema de quien se trata para que después de tomar la muestra del rostro se determine si realmente pertenece a la persona que dice ser. Ambos metodos se explican a continuación.

Clasificacion y Verificacion con GMM

La figura 4.11 muestra un esquema para la tarea de reconocimiento, en el cual como entrada tendremos un vector característico de un rostro el cual será la entrada de los M modelos Gaussianos obtenidos anteriormente (1 para cada persona), como resultado de esto tendremos un vector de probabilidad de cada uno de los modelos y a través del teorema de Bayes se tomara la decisión para asignar la imagen a un individuo.

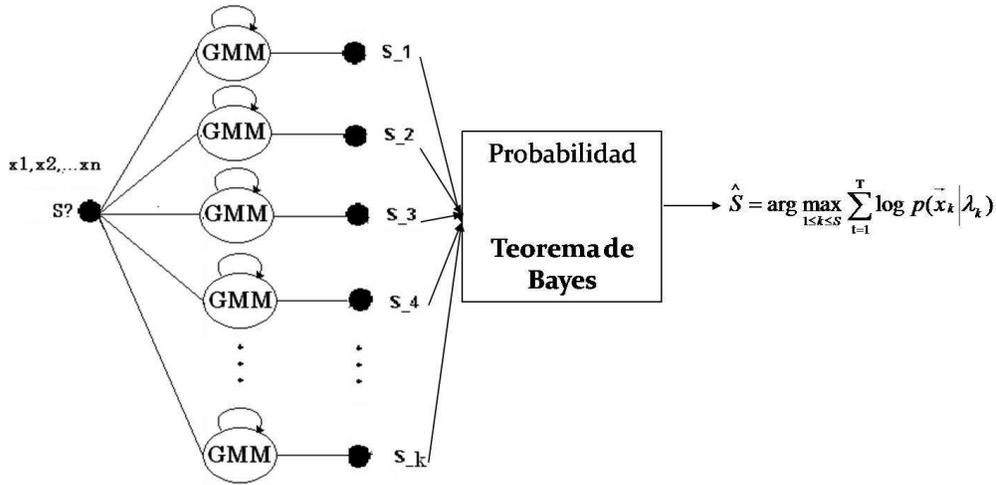


Figura 4.11: Esquema general para el reconocimiento con GMM.

Para la parte de verificación, como ya se mencionó solo se tomará en cuenta el modelo que se pretende verificar, la figura 4.12 muestra el esquema general para la verificación usando GMM.

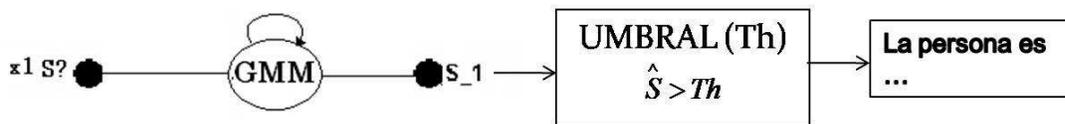


Figura 4.12: Esquema general para la verificación con GMM.

Donde se puede observar que ahora solo se introduce la imagen del rostro en cuestión al modelo de mezclas al que según corresponde, al igual que en el reconocimiento después de pasar el vector característico por el modelo obtendremos como resultado un vector de probabilidades, el cual si es mayor que un umbral previamente establecido se considera que la imagen si pertenece a ese modelo, de lo contrario este sería rechazado.

Los resultados obtenidos usando como método de reconocimiento al GMM se podrán observar en el siguiente capítulo.

Clasificación y Verificación con SVM

La figura 4.13 muestra un esquema para la tarea de reconocimiento usando la SVM, se puede observar que es el mismo procedimiento que se usó para el reconocimiento usando el GMM solo que ahora los modelos usados para reconocer son los obtenidos por la Máquina de Vector de Soporte y los resultados obtenidos pasan directamente al teorema de Máxima Verosimilitud para discriminar a que modelo pertenece ese rostro.

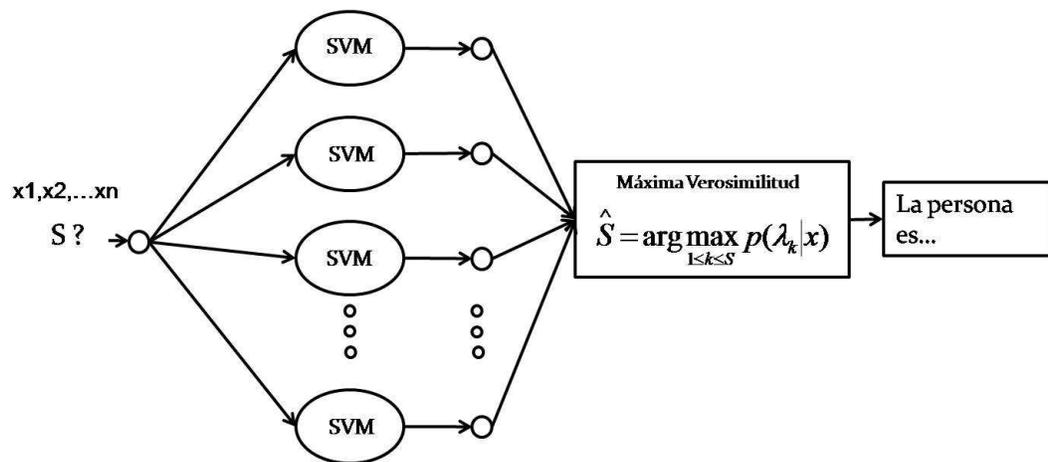


Figura 4.13: Esquema general para el reconocimiento con SVM.

Pasa lo mismo con la parte de verificación, es el mismo procedimiento que en la verificación con GMM lo único que cambia es el uso de los modelos. Esto se muestra en la figura 4.14.

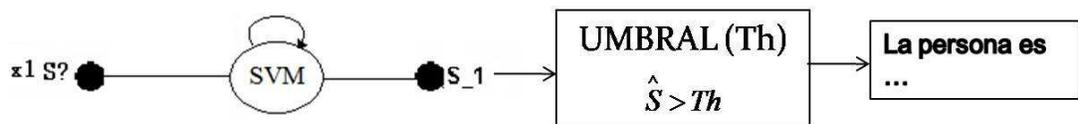


Figura 4.14: Esquema general para la verificación con SVM.

Cabe destacar que el umbral establecido en este método es igual para todos los individuos a diferencia de la verificación con GMM donde el umbral puede variar de un individuo a otro.

4.3. Conclusiones

En este capítulo se observó como es la aplicación de los métodos de extracción de características propuestos en este trabajo, por lo que se notó que es relativamente sencillo la aplicación de cada uno de estos además que no requieren de un gran costo computacional, así como también se habló de como es el procedimiento para el entrenamiento de los dos clasificadores usando dos herramientas distintas las cuales son sencillas de manejar.

Por otra parte se pudo notar que existen diferentes formas para extraer las características de la imagen de un rostro, algunos de estos son robustos a diferentes variaciones como la iluminación o la oclusión.

El combinar cada uno de estos clasificadores con las diferentes técnicas de extracción de características propuestos en este trabajo haciendo un enfoque en el método de Eigenfaces ya que es un método algo novedoso al cual se le propusieron algunas variantes, todo esto nos ayudara a hacer una comparación y sacar algunas conclusiones de esto tanto entre los métodos de extracción de características como de los clasificadores y ambos.

Capítulo 5

Resultados Experimentales

En esta tesis, se presenta un estudio sobre el uso de tres métodos diferentes de extracción de características y dos clasificadores. En este capítulo se reportan todas las pruebas realizadas así como los resultados obtenidos en estas.

Este capítulo se divide en 2 partes la primera es usando el GMM como clasificador y verificador, la segunda es usando el SVM como clasificador y como verificador. A su vez cada una de estas etapas se divide en 3 partes, la primera es usando el método de Filtros de Gabor como, la segunda es usando la Transformada Discreta Wavelet y por último el uso de Eigenphases.

Es importante mencionar que para cada una de las pruebas se utilizaron dos tipos de conjuntos de imágenes de entrenamiento las cuales son nombradas imágenes de entrenamiento 1 e imágenes de entrenamiento 2, en la que en el primer conjunto de imágenes se utilizan solamente imágenes con cambios de iluminación y con expresiones faciales, tanto que en el segundo conjunto de imágenes se utilizan imágenes con cambios de iluminación, con expresiones faciales y ahora se incluyen para el entrenamiento las imágenes con oclusión parcial del rostro que en la base de datos que se utiliza consta de imágenes con lentes de sol y con bufanda, un ejemplo de estos conjuntos de imágenes se muestra en la figura 5.1.

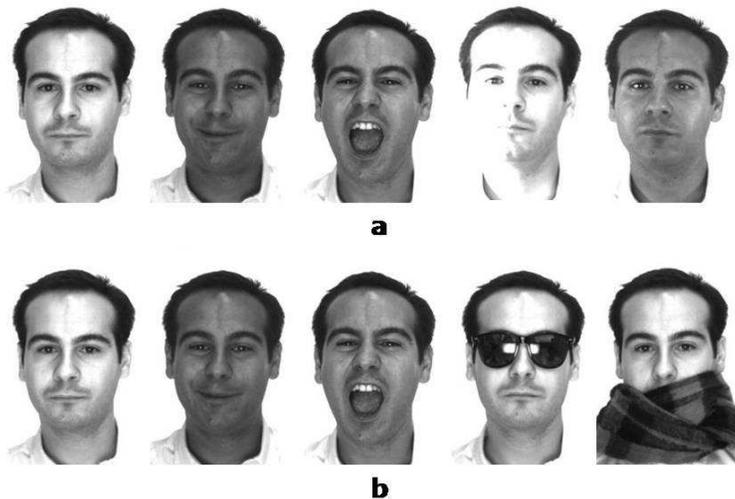


Figura 5.1: a) Imágenes de entrenamiento 1. b) Imágenes de entrenamiento 2.

5.1. GMM como clasificador y verificador

Para el entrenamiento se usaron 2 conjuntos de imágenes, uno solo utilizando imágenes con cambios de iluminación y expresión y la otra usando imágenes con estas variantes además de imágenes con oclusión parcial del rostro como son el uso de lentes de sol o bufanda, para el entrenamiento del GMM se utilizaron 23 imágenes de entrenamiento cuando se usaron las imágenes de entrenamiento 1 y 40 cuando se usaron las imágenes de entrenamiento 2, esto es de las 78 existentes por cada una de las 120 personas. Para la parte del reconocimiento se utilizó toda la base de datos de rostros llamada “AR Face Database” la cual cuenta con 120 personas y 78 imágenes de cada una que nos da un total de 9360 imágenes de prueba incluyendo las imágenes de entrenamiento, esto significa que el sistema intentará reconocer alrededor de 55 o 38 imágenes de cada persona que nunca ha visto.

Por otra parte, después de varias pruebas hechas para la parte de verificación con GMM se notó que este es muy sensible a los cambios bruscos de información como ocurre en las imágenes donde la persona trae puesta una bufanda ya que se obtuvieron resultados con un alto porcentaje de falso rechazo, por lo que se tuvieron que descartar estas imágenes dejando solo las que traen lentes de sol como oclusión parcial del rostro, por lo cual ahora

se utilizan las mismas 26 imágenes sin oclusión para entrenar pero ahora solo se toman 60 imágenes de cada persona para la fase de prueba (descartando 18 imágenes con bufanda), esto es cuando se utilizan las imágenes de entrenamiento 1, para las imágenes de entrenamiento 2 ahora se utilizan 30 imágenes en vez de 40 para el entrenamiento (solo para la fase de verificación con GMM) y las 60 imágenes de cada persona para la fase de prueba. Es necesario mencionar que dentro de esas 60 imágenes para la fase de prueba están incluidas las imágenes con las cuales se entreno el sistema, por lo que el sistema tratara de verificar alrededor de 34 y 30 imágenes de cada persona que nunca ha visto.

También hay que destacar que el total de imágenes de la base de datos pasaría de 9360 a 7200 imágenes en total, esto es por que se descuentan 18 imágenes con bufanda de cada una de las 120 personas en la base de datos.

Cabe destacar que el Modelo de Mezclas Gaussianas calcula un modelo independiente para cada una de las personas de la base de datos ya que trata de aproximar a través de gaussianas la señal creada con las matrices características de las imágenes por lo que solo se necesita en la entrada información de la persona para la cual se desea calcular su modelo, esto a su vez significa que necesita mucha mas información que un clasificador discriminante como la Maquina de Vector de Soporte.

Filtros de Gabor

La primera prueba realizada con GMM fue la combinación de este clasificador con el método de Filtros de Gabor tanto para reconocer como para verificar, una vez obtenidos los vectores característicos mediante este método de extracción de características se realizo el entrenamiento del GMM.

Como se explico en el capitulo 4 para el entrenamiento de este clasificador solo se necesita la información de la persona en cuestión por lo que el modelo es independiente de los demás, esta puede ser considerada como una ventaja con respecto a otros clasificadores ya que al ingresar una persona a la base de datos no es necesario entrenar de nuevo con todas las personas en ella.

Los resultados que se obtuvieron se muestran en la tabla 5.1.

Tabla 5.1: Reconocimiento usando Filtros de Gabor y GMM

	Porcentaje de reconocimiento
Imágenes de entrenamiento 1	71.4316
Imágenes de entrenamiento 2	91.5278

Posteriormente se utilizó el GMM como verificador obteniendo los resultados de la tabla 5.2.

Tabla 5.2: Verificación usando Filtros de Gabor y GMM

	Promedio de	Porcentaje
Imágenes de entrenamiento 1	Falsa Aceptación	4.7386
	Falso Rechazo	7.2639
	Error Global	4.7596
Imágenes de entrenamiento 2	Falsa Aceptación	1.9833
	Falso Rechazo	4.125
	Error Global	2.0012

En las tablas 5.1 y 5.2 muestran que los resultados obtenidos usando el GMM son aceptables tomando en cuenta que mientras se le proporcione más información al GMM su desempeño aumentará.

Transformada Discreta Wavelet

La siguiente prueba que se realizó con el GMM fue usando el método de extracción de características basado en la Transformada Discreta Wavelet (DWT) tanto para reconocer como para verificar.

Los resultados de la tabla 5.3 corresponden a las pruebas de reconocimiento.

Tabla 5.3: Reconocimiento usando DWT y GMM

	Porcentaje de reconocimiento
Imágenes de entrenamiento 1	71.3034
Imágenes de entrenamiento 2	92.5107

A continuación los resultados obtenidos fueron cuando se usó el mismo método de extracción de características pero ahora el Modelo de Mezclas Gaussianas es usado como

verificador. Estos resultados se muestran en la tabla 5.4.

Tabla 5.4: Verificacion usando DWT y GMM

	Promedio de	Porcentaje
Imágenes de entrenamiento 1	Falsa Aceptación	6.6922
	Falso Rechazo	6.6389
	Error Global	6.6918
Imágenes de entrenamiento 2	Falsa Aceptación	1.9201
	Falso Rechazo	5.25
	Error Global	1.9478

Como se espero los resultados obtenidos con la Transformada Discreta Wavelet son muy aceptables tanto en reconocimiento como en verificación, hay que recordar que en la parte de verificación, el error que hay que cuidar que no sea muy alto es el de falsa aceptación, ya que es cuando se acepta una persona con la identidad de otra.

Eigenphases

Por ultimo, la prueba realizada con GMM fue la combinación de este clasificador con el método de Eigenphases tanto para reconocer como para verificar, una vez obtenidos los vectores característicos mediante este método con todas las variantes propuestas en este trabajo se realizo el entrenamiento del Modelo de Mezclas Gaussianas.

Después de obtener los modelos de cada una de las personas de la base de datos se realizaron las pruebas de reconocimiento, los resultados obtenidos se muestran en la tabla 5.5.

Tabla 5.5: Reconocimiento usando Eigenphases y GMM

	Porcentaje de reconocimiento					
	Norm. completa	Local 3	Local 6	Local fourier 3	Local fourier 6	Eigen-phases
Imágenes de entrenamiento 1	60.4487	65.7158	61.8483	63.1261	65.1282	60.6293
Imágenes de entrenamiento 2	87.1902	92.6609	89.3805	87.9551	91.4103	87.2412

donde el significado de cada campo es:

Norm. completa Consiste en normalizar la imagen completa y después extraer el espectro de fase de toda la imagen.

Local 3 Consiste en normalizar la imagen localmente con ventanas de 3×3 y después extraer el espectro de fase de la imagen completa.

Local 6 Consiste en normalizar la imagen localmente con ventanas de 6×6 y después extraer el espectro de fase de la imagen completa.

Local fourier 3 Consiste en normalizar la imagen y extraer el espectro de fase localmente con ventanas de 3×3 de cada imagen.

Local fourier 6 Consiste en normalizar la imagen y extraer el espectro de fase localmente con ventanas de 6×6 de cada imagen.

Eigenphases Se usa el método original.

Estos significados hay que tomarlos en cuenta también para la parte de verificación.

Los resultados de esta etapa se muestran en la tabla 5.6.

Tabla 5.6: Verificacion usando Eigenphases y GMM

		% Promedio de		
		Falsa Aceptación	Falso Rechazo	Error Global
Imágenes de entrenamiento 1	Norm completa	37.7096	14.8333	37.519
	Local 3	24.2568	14.5139	24.1756
	Local 6	25.0183	18.3611	24.9628
	Local fourier 3	27.9608	19.0417	27.8865
	Local fourier 6	18.8384	15.6111	18.8115
	Eigenphases	37.7096	14.8333	37.519
Imágenes de entrenamiento 2	Norm completa	21.3925	21.4583	21.3931
	Local 3	17.6977	14.1528	17.6682
	Local 6	19.3148	17.9722	19.3036
	Local fourier 3	23.8016	19.5972	23.7666
	Local fourier 6	15.6389	15.6389	15.6652
	Eigenphases	21.3925	21.4583	21.3931

5.2. SVM como clasificador y verificador

Como ya se menciona para el entrenamiento se usaron 2 conjuntos de imágenes, uno solo utilizando imágenes con cambios de iluminación y expresión y la otra usando imágenes con estas variantes además de imágenes con oclusión parcial del rostro como son el uso de lentes de sol o bufanda, para el entrenamiento de la SVM se utilizaron 10 imágenes de cada una de las 120 personas que nos da un total de 1200 imágenes de entrenamiento. Para la parte del reconocimiento o la verificación se utilizo toda la base de datos de rostros llamada “AR Face Database” la cual cuenta con 120 personas y 78 imágenes de cada una que nos da un total de 9360 imágenes de prueba incluyendo las 1200 imágenes de entrenamiento, esto significa que el sistema intentara reconocer a 8160 imágenes de cada persona que nunca ha visto.

Como se explico en el capitulo 4 para el individuo que se quiere reconocer se le agregara una etiqueta con el número 1 y las otras 1190 imágenes serán etiquetadas con un -1 por lo cual solo tendremos 2 clases.

Filtros de Gabor

La primera prueba realizada con SVM fue la combinación de este clasificador con el método de extracción de características de Filtros de Gabor tanto para reconocer como para verificar, una vez obtenidos los vectores característicos mediante este método, se realizo el entrenamiento de la Maquina de Vector de Soporte.

Los primeros resultados que se obtuvieron con este clasificador y este método de extracción de características son los de reconocimiento que se muestran en la tabla 5.7.

Tabla 5.7: Reconocimiento usando Filtros de Gabor y SVM

	Porcentaje de reconocimiento
Imágenes de entrenamiento 1	75.9829
Imágenes de entrenamiento 2	94.9038

A continuación los resultados obtenidos fueron cuando se uso el mismo método de extracción de características pero ahora la Maquina de Vector de Soporte es usado como

verificador. Estos resultados se muestran en la tabla 5.8.

Tabla 5.8: Verificación usando Filtros de Gabor y SVM

	Promedio de	Porcentaje
Imágenes de entrenamiento 1	Falsa Aceptación	0.4282
	Falso Rechazo	22.6496
	Error Global	0.6134
Imágenes de entrenamiento 2	Falsa Aceptación	0.1185
	Falso Rechazo	8.3761
	Error Global	0.1873

En la tabla 5.7 y 5.8 se puede notar que se obtienen mejores resultados cuando se utiliza el conjunto de imágenes de entrenamiento 2, esto se debe a que le das mucho más información acerca de los rostros a diferencia del conjunto de imágenes de entrenamiento 1.

Transformada Discreta Wavelet

La segunda prueba realizada con SVM fue la combinación de este clasificador con el método de extracción de características de la Transformada Discreta Wavelet (DWT) tanto para reconocer como para verificar, una vez obtenidos los vectores característicos mediante este método, se realizó el entrenamiento de la Máquina de Vector de Soporte.

Los primeros resultados que se obtuvieron con este clasificador y este método de extracción de características son los de reconocimiento que se muestran en la tabla 5.9.

Tabla 5.9: Reconocimiento usando DWT y SVM

	Porcentaje de reconocimiento
Imágenes de entrenamiento 1	79.5406
Imágenes de entrenamiento 2	97.2863

A continuación los resultados obtenidos fueron cuando se usó el mismo método de extracción de características pero ahora la Máquina de Vector de Soporte es usado como verificador. Estos resultados se muestran en la tabla 5.10.

Al igual que en los Filtros de Gabor en las tablas 5.9 y 5.10 se puede observar que si le damos información de todas las imágenes (con cambios de iluminación, con expresiones

Tabla 5.10: Verificación usando DWT y SVM

	Promedio de	Porcentaje
Imágenes de entrenamiento 1	Falsa Aceptación	0.1661
	Falso Rechazo	22.2650
	Error Global	0.3502
Imágenes de entrenamiento 2	Falsa Aceptación	0.0425
	Falso Rechazo	4.6368
	Error Global	0.0808

faciales y oclusión parcial del rostro) aunque sea poca información pero que sea de las diferentes variaciones el porcentaje tanto de reconocimiento como de verificación aumenta. También se puede observar que los porcentajes obtenidos con la DWT son un poco mejor que los obtenidos usando los Filtros de Gabor.

Eigenphases

Por último, la prueba realizada con SVM fue la combinación de este clasificador con el método de Eigenphases tanto para reconocer como para verificar, una vez obtenidos los vectores característicos mediante este método con todas las variantes propuestas en este trabajo, se realizó el entrenamiento de la Máquina de Vector de Soporte.

Como se hizo mención en el capítulo 4 sección 4.2.1 en el apartado de Eigenphases se enlistaron las 5 variantes que se proponen al método original de Eigenphases. Los resultados que se muestran en la tabla 5.11 son los obtenidos de la prueba de reconocimiento con todas las variantes.

Tabla 5.11: Reconocimiento usando Eigenphases y SVM

	Porcentaje de reconocimiento					
	Norm. completa	Local 3	Local 6	Local fourier 3	Local fourier 6	Eigenphases
Imágenes de entrenamiento 1	80.6838	84.4444	81.0577	85.9188	85.5876	80.6338
Imágenes de entrenamiento 2	96.2821	97.6709	97.2863	97.9167	97.8526	96.2821

Los significados de cada uno de los campos son los mismos expuestos en la parte de Eigenphases con GMM.

Los resultados de la etapa de verificación se muestran en la tabla 5.12.

Tabla 5.12: Verificación usando Eigenphases y SVM

		% Promedio de		
		Falsa Aceptación	Falso Rechazo	Error Global
Imágenes de entrenamiento 1	Norm completa	0.0019	34.7415	0.3080
	Local 3	0.0179	27.4466	0.2464
	Local 6	0.0317	29.9893	0.2813
	Local fourier 3	0.0238	22.6603	0.2124
	Local fourier 3	0.1353	18.5791	0.2890
	Eigenphases	0.0019	34.7415	0.3080
Imágenes de entrenamiento 2	Norm completa	0.0023	16.0256	0.1360
	Local 3	0.0306	6.5812	0.0852
	Local 6	0.0408	8.5363	0.1116
	Local fourier 3	0.0347	6.7842	0.0910
	Local fourier 3	0.1484	3.0021	0.1722
	Eigenphases	0.0023	16.0470	0.1360

Se puede observar que los resultados obtenidos utilizando las variantes propuestas son mejores que las obtenidas con el método de Eigenphases original, esto es tanto en reconocimiento como en verificación.

Capítulo 6

Conclusiones Generales y Trabajos Futuros

6.1. Conclusiones Generales

La motivación general que ha guiado esta tesis ha sido contribuir al estudio y al desarrollo de técnicas que permitan realizar de forma adecuada el proceso de verificación facial tomando en cuenta algunas variaciones como son la iluminación, las expresiones faciales y la oclusión parcial del rostro. En este trabajo se han estudiado técnicas basadas en imágenes bidimensionales como son los Filtros de Gabor, la Transformada Discreta Wavelet y el método de Eigenphases, además que se han utilizado dos clasificadores: el Modelo de Mezclas Gaussianas y la Máquina de Vector de Soporte.

En este capítulo se comparan las distintas técnicas utilizadas y se da un resumen de las principales aportaciones realizadas en esta tesis además de una perspectiva de lo que puede ser el trabajo a futuro.

En la tesis se han considerado 3 métodos de extracción de características y dos clasificadores para su uso en sistemas de reconocimiento y verificación facial. Las imágenes utilizadas provienen de la base de datos llamada “AR Face Database” que consta de 120 personas con 78 imágenes de cada una, además de tener variaciones en lo que es la iluminación, la expresión facial y la oclusión parcial del rostro.

Las pruebas realizadas con GMM como clasificador muestra que el porcentaje de reconocimiento es aceptable tomando en cuenta que mientras mas información se use para el entrenamiento del GMM su desempeño aumentara. Los mejores resultados obtenidos cuando se entrena con 40 imágenes (imágenes de entrenamiento 2) es cuando se utiliza el método de Eigenphases utilizando la variante de normalizar localmente con una ventana de tamaño 3×3 donde se obtiene un porcentaje de reconocimiento de 92.6609% muy cercano también al porcentaje que se obtiene utilizando wavelets y en el peor de los casos se obtiene un porcentaje de reconocimiento de 94.9038% esto es utilizando el método de eigenphases con su variación de normalizar la imagen completa, obteniendo un porcentaje de reconocimiento de 87.1902%.

En contraste cuando se entrena al GMM con solo 26 imágenes sin incluir imágenes con oclusión parcial (imágenes de entrenamiento 1) el rendimiento del sistema decrece considerablemente obteniendo en el mejor de los casos un porcentaje de reconocimiento del 71.4316% cuando se utilizan los Filtros de Gabor y un porcentaje de 60.4487% en el peor de los casos cuando se utiliza el método de Eigenphases con su variación de normalizar la imagen completa.

Tanto que las pruebas realizadas con la Maquina de Vector de Soporte como clasificador muestra que el porcentaje de reconocimiento obtenido es bueno siempre y cuando sea entrenado con información de todas las variaciones existentes en la base de datos utilizada, los mejores resultados son los obtenidos con el método de Eigenphases utilizando la variante de normalizar y obtener el espectro de fase localmente con una ventana de tamaño 3×3 donde se obtiene un porcentaje de reconocimiento de 97.9167% y en el peor de los casos se obtiene un porcentaje de reconocimiento de 94.9038% esto es utilizando los Filtros de Gabor considerando que este porcentaje no esta del todo mal.

Por el contrario cuando se entrena al SVM solo con algo de información (utilizando las Imágenes de entrenamiento 1) se obtiene en el mejor de los casos un porcentaje de reconocimiento del 85.9188% utilizando también la variante de normalizar y obtener el espectro de fase localmente con una ventana de tamaño 3×3 y en el peor de los casos se obtiene el 75.9829% de reconocimiento también utilizando los Filtros de Gabor.

En la parte de verificación de los dos clasificadores se debe tomar en cuenta que el porcentaje de la falsa aceptación es mas importante que otra cosa ya que el sistema no debe de permitir que sea aceptada una persona con la identidad de otra, por lo que en esta parte lo que se busca es que el sistema cometa un porcentaje mínimo de error en la falsa aceptación.

Las pruebas realizadas con el GMM como verificador muestran resultados que varían entre ellos y que puede que sean un poco bajos, esto es, como se menciono en el capitulo de Resultados, el GMM calcula un modelo para cada una de las personas y este es individual para todos, por lo que el sistema no tiene conocimiento de nada mas que del rostro del cual se obtiene el modelo, por lo que el GMM necesita mas información de entrenamiento que un clasificador supervisado como una red neuronal o la Maquina de Vector de Soporte.

Otro aspecto es el umbral a considerar para tomar la decisión de verificación ya que depende de este para que el porcentaje de falsa aceptación no sea tan grande.

Las pruebas realizadas con la Maquina de Vector de Soporte como verificador muestra que al igual que en el reconocimiento el porcentaje de falso rechazo disminuye cuando se entrena con información de todas las variaciones existentes en la base de datos utilizada, el mas bajo porcentaje de falsa aceptación es el que se obtuvo usando el método de Eigenphases obteniendo un porcentaje de 0.0023 % de falsa aceptación, mientras que el mas alto fue de 0.1484 % usando la variante de normalizar y obtener el espectro de fase localmente con una ventana de tamaño 6×6 . Ahora el otro aspecto a considerar es el falso rechazo en el cual el mas bajo porcentaje fue obtenido utilizando la variante de normalizar y obtener el espectro de fase localmente con una ventana de tamaño 6×6 con un porcentaje de 3.0021 % mientras el mas alto fue usando el método de Eigenphases obteniendo un porcentaje de 16.0470 % de falso rechazo. Esto significa que se puede tener un bajo porcentaje de falsa aceptación pero a su vez se tiene un alto porcentaje de falso rechazo y viceversa, por lo que se debe de buscar un balance entre estos dos porcentajes.

Ahora, cuando la SVM es entrenada solo con las imágenes de entrenamiento 1 el mas bajo porcentaje obtenido para falsa aceptación es de 0.0019 % y sigue perteneciendo al método de Eigenphases mientras que el mas alto ahora le pertenece a los Filtros de Gabor

con un porcentaje de 0.4282 %, al igual el falso rechazo tiene un porcentaje de 18.5791 % en el mejor de los casos y pertenece a la variante de normalizar y obtener el espectro de fase localmente con una ventana de tamaño 6×6 tanto que en el peor de los casos se tiene un porcentaje de 34.7415 % de falso rechazo.

Todos estos resultados obtenidos en conjunto con los que se muestran en las tablas del capítulo anterior muestran que las variantes propuestas en este trabajo para el método de Eigenphases arrojan buenos resultados en comparación con los otros métodos de extracción de características expuestos en este trabajo, esto es tanto en reconocimiento como en verificación.

6.2. Trabajos Futuros

Como se menciona en los primeros capítulos de esta tesis, es difícil encontrar un sistema biométrico capaz de reconocer o verificar sin ninguna falla, lo único que se puede hacer es el disminuir el error del sistema lo más que se pueda.

En base a las pruebas realizadas en este trabajo surgen algunas líneas de continuación como son:

1. La combinación de los vectores de características obtenidos con los métodos de extracción expuestos en este trabajo así como incluir algunos otros.
2. Usar imágenes a color.
3. La combinación de este sistema biométrico con algún otro, por ejemplo huella, palma de la mano, voz, etc. Para hacer más robusto el sistema.
4. Usar imágenes en tercera dimensión.
5. Probar el desempeño del sistema con imágenes con oclusión artificial (generada por el programador).
6. Usar una base de datos con otras características como rotación, tamaño, inclinación.
7. Buscar una forma de entrenar el GMM de forma supervisada.

Referencias

- [1] P. Reid. *BIOMETRICS for Networks Security*. Prentice Hall, 2004.
- [2] S. Sirohey R. Chellappa, C. Wilson. Human and machine recognition of faces: A survey. *Proc. IEEE*, 83(5):705–740, 1995.
- [3] A. Shashua. *Geometry and Photometry in 3D Visual Recognition*. PhD thesis, Massachusetts Institute of Technology, 1992.
- [4] B. Miller. Vital signs of identity. *IEEE Spectrum*, 31(2):22–30, 1994.
- [5] A. Jain L. Hong. Integrating faces and fingerprints for personal identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1295–1307, 1998.
- [6] A. Pentland M. Turk. Eigenfaces for recognition. *J. Cognitive Neuroscience.*, 3(1):71–86, 1991.
- [7] I. Smith. *A tutorial on Principal Components Analysis*, February 2002.
- [8] M. Kirby L. Sirovich. Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. Am.A.*, 4(3):519–524, 1987.
- [9] D. Kriegman P. Belhumeur, J. Hespanha. Eigenfaces vs fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.
- [10] J. Hager P. Ekman T. Sejnowski G. Donato, M. Bartlett. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:974–989, 1999.

-
- [11] H. Asai M. Yoshida, T. Kamio. Face image recognition by 2-dimensional discrete walsh transform and multi-layer neural network. *IEICE TRANSACTIONS on fundamentals of Electronics, Communications and Computer Sciences*, 86(10):2623–2627, 2003.
- [12] Clark M. Geisler W.S. Bovik, A. C. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, 1990.
- [13] Farrokhnia F. Jain, A.K. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
- [14] Higgins W. E. Dunn, D. Optimal gabor filters for texture segmentation. *IEEE Trans. Image Proc.*, 4(7), 1995.
- [15] H. M. Pérez Meana J. C. Oropeza Ortega, M. Nakano Miyatake. Reconocimiento de rostros empleando filtros de gabor y redes neuronales de retropropagación. *Avances de Ciencias de la Computacion.*, pages 112–216, 2003.
- [16] M. Nakano Miyatake H. M Pérez Meana G. Aguilar Torres, G. Sanchez Pérez. Face recognition algorithm using the discrete gabor transform. *Proceedings of the 17th International Conference on Electronics, Communications and Computers.*, page 35, 2007.
- [17] C. K. Chui. *An Introduction to Wavelets*. Boston Academic Press, 1992.
- [18] Burcu Kepenekci. *Face Recognition using Gabor Wavelet Transform*. 2001.
- [19] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 41:906–966, 1988.
- [20] P.K. Khosla Marios Savvides, B.V.K. Vijaya Kumar. Eigenphases vs. eigen-faces. *Proceedings of the 17th International Conference on Pattern Recognition*, 3, 2004.
- [21] A. V. Oppenheim M. H. Hayes, J. S. Lim. Signal reconstruction from phase or magnitude. *IEEE Trans. Acoust., Signal Processing*, 28:672–680, 1980.

-
- [22] J. S. Lim A. V. Oppenheim. The importance of phase in signals. *Proc. IEEE*, 69(5):529–541, 1981.
- [23] W. Pitts W. McCulloch. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
- [24] Kevin Gurney. *An Introduction to Neural Networks*. CRC Press, 2003.
- [25] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 2005.
- [26] Laurene V. Fausett. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice Hall, 1994.
- [27] Richard C. Rose Douglas A. Reynolds. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING*, 3(1), 1995.
- [28] Seung Y. Na. Jin Y. Kim, Dae Y. Ko. Implementation and enhancement of gmm face recognition systems using flatness measure. *IEEE Robot and Human Interactive Communication*, 2004.
- [29] V. Vapnik. *The nature of statistical learning theory*. Wiley, 1998.
- [30] A. Smola V. Vapnik, S. Golowich. Support vector method for function approximation, regression, estimation and signal processing. *Advances in Neural Information Processing Systems*, 9:281–287, 1997.
- [31] V. Vapnik. *Statistical learning theory*. Wiley, 1998.
- [32] M. Wright P. E. Gill, W. Murray. *Practical Optimization*. Academic Press, 1981.
- [33] V. Vapnik C. Cortes. Support vector networks. *Machine Learning*, 20:273–297, 1995.
- [34] H. Bülthoff C. Burges V. Vapnik T. Vetter V. Blanz, B. Schölkopf. Comparison of view-based object recognition algorithms using realistic 3d models. *Artificial Neural Networks*, 1112:251–256, 1996.

-
- [35] T. Joachims. Text categorization with support vector machines. *Technical report*, 8, 1997.
- [36] B. S. Nagy F. Riesz. *Functional analysis*. Ungar, 1955.
- [37] K. Chan G. Guo, S. Z. Li. Face recognition by support vector machines. *Proc. fourth IEEE International Conference on Face and Gesture Recognition*, pages 196–201, 2000.
- [38] J. Kittler Y. P. Li K. Jonsson, J. Matas. Learning support vectors for face verification and recognition. *Proc. fourth IEEE International Conference on Face and Gesture Recognition*, pages 205–210, 2000.
- [39] R. Benavente A.M.Martinez. The ar face database. CVC Technical Report 24. Junio 1998.
- [40] Lindsay I Smith. *A tutorial on Principal Components Analysis*. Febrero 2002.

Apéndice A

Publicaciones

“Feature Extraction and Face Verification Using Gabor and Gaussian Mixture Models”

Jesús Olivares Mercado, Gabriel Sánchez Pérez, Mariko Nakano Miyatake, Héctor Pérez Meana

Publicado en MICAI, Springer Lecture Note in Artificial Intelligence, 2007

“Feature Extraction using Gabor and DWT for GMM based Face Verification Algorithms”

Jesús Olivares Mercado, Gabriel Sánchez Pérez, Mariko Nakano Miyatake, Héctor Pérez Meana

Publicado en NNAM, Journal of Computing Research, 2007

“Extracción de Características Usando DGT y DWT para el Desarrollo de Algoritmos de Verificación Basados en el Modelo de Mezclas Gausiana”

Jesús Olivares Mercado, Gabriel Sánchez Pérez, Mariko Nakano Miyatake, Héctor Pérez Meana

Publicado en CIDETEC, Memoria de CIDETEC, 2007

“Face Recognition Based on the Phase Spectrum of Local Normalized Image”

Jesús Olivares Mercado, Kazuhiro Hotta, Haruhisa Takahashi, Héctor Pérez Meana, Gabriel Sánchez Pérez

Publicado en MICAI, Proceedings of MICAI, 2008

“Eigenphase of Local Normalized Image”

Jesus Olivares Mercado, Kazuhiro Hotta, Haruhisa Takahashi, Hector Perez Meana

Publicado en FIT, Proceedings of FIT

“Face Recognition Based on Normalization and the Phase Spectrum of the Local Part of an Image”

Jesus Olivares Mercado, Kazuhiro Hotta, Haruhisa Takahashi, Hector Perez Meana, Mariko Nakano Miyatake, Gabriel Sanchez Perez

Publicado en ISVC, Springer Lecture Note in Computer Science 2008

Feature Extraction and Face Verification Using Gabor and Gaussian Mixture Models

Jesus Olivares-Mercado, Gabriel Sanchez-Perez, Mariko Nakano-Miyatake,
and Hector Perez-Meana

ESIME Culhuacan, Instituto Politécnico Nacional, Av. Santa Ana No. 1000, Col. San Francisco
Culhuacan, 04430 Mexico D.F. Mexico
mariko@calmecac.esimecu.ipn.mx

Abstract. This paper proposes a faces verification in which the feature extraction is carried out using the discrete Gabor function (DGF), while the Gaussian Mixture Model (GMM) is used in the face verification stage. Evaluation results using standard data bases with different parameters, such as the number of mixtures and the number of face used for training show that proposed system provides better results than other proposed systems with a correct verification rate larger than 95%. Although, as happens in most face recognition systems, the verification rate decreases when the target faces present some rotation degrees.

Keywords: Gabor functions, Gaussian Mixture Model, Face verification.

1 Introduction

The development of security systems based on biometric features has been a topic of active research during the last three decades, because the verification of the people identity to access control and to enforce security in restricted areas, etc. are fundamental aspects in these days. The terrorist attacks happened during the last decade have demonstrated that it is indispensable to have reliable security systems in offices, banks, companies, trades, etc.; increasing in such way the necessity to develop more reliable methods to verify the people identity. The identity verification systems using biometric methods appear to be good alternatives for the development of such security systems.

The biometrics systems consist of a group of automated methods for recognition or verification of people identity using physical characteristics or personal behavior of the person under analysis [1]. This technology is based on the premise that each person is unique and possesses distinctive features that can be used to identify him. Following these ideas several biometric based security systems have been developed using fingerprints, iris, voice, hand and face features. Among them, the face verification systems appear to be a desirable alternative because it is non-invasive and its computational complexity is relatively low.

The face verification has been a topic of active research during the last three decades; because it is perhaps, the biometric method easier of understanding because for us the face is the most direct way to identify the people. In addition the data

770 J. Olivares-Mercado et al.

acquisition of this method consists in taking a picture, doing it one of the biometric methods with larger acceptance among the users.

The recognition is a very complex activity of the human brain. For example, we can recognize hundred of faces learned throughout our life and to identify familiar faces at the first sight, even after several years of separation, with relative easy. However for a computer it is not a simple task. For instance, recently proposed face recognition systems, achieve a recognition rate of about 91% when the face image is not rotated or the rotation is relatively low. However although, this recognition rate is good enough for several practical applications, it may be not large enough for applications where the security should be extreme; such that we cannot tolerate a high erroneous recognition rate. This paper proposes a face recognition algorithm that is able of achieving an erroneous verification rate below 9%.

Several methods have been proposed for face recognition [2], [3], such as the methods of the statistical correlation of the face geometry [4]; the face form which uses the distances among the position of the eyes, mouth, nose, etc. as well as those using the neuronal networks technology that trait to imitate the operation of the human brain. Many of these systems can recognize a person even when they present some physical changes, such as the growth of the beard or mustache, changes in the color or hair style, the use of glasses, etc. Although in general they are sensitive to rotations of the face images.

Before beginning to analyze the procedures used for face recognition, it is necessary to point out the verification concept. In face verification, the person informs to the system about his/her identity, presenting an identification card or writing a special password, etc. The system captures the person's features (for example the persons' face in this case), and the proceeds to determine if the people is whom his/her claims to be.

2 Proposed System

This section provides a detailed description of the proposed face verification algorithm which consists of three stages. Firstly a feature extraction of the face is

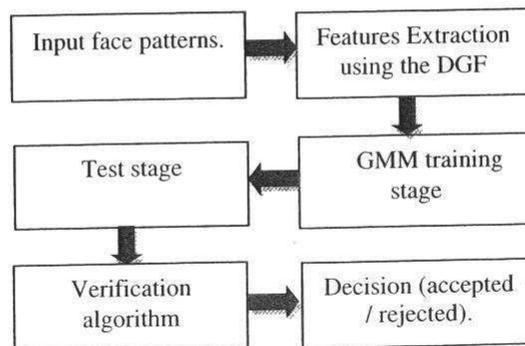


Fig. 1. Proposed face verification algorithm

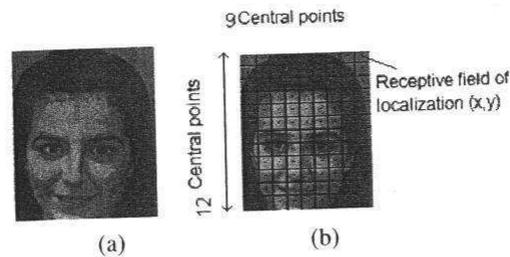


Fig. 2. a) Original face. b) Face image divided in 108 receptive fields and 108 central points (x, y) .

carried out using the Gabor discrete transform (DGF). Next using these feature vectors, a model for each face is obtained using a Gaussian Mixture Model (GMM). Finally during the verification process, the GMM output is used to take the final decision. Figure 1 shows the block diagram of proposed algorithm.

2.1 Feature Extraction Stage

The feature extraction plays a very important role in the any pattern recognition system. To this end, the proposed algorithm uses the DGT which has some relation with the human visual system (HVS). The two dimensional discrete Gabor functions (2D-DGF) depend on four parameters, two that express their localization in the space (x, y) and other two that express the spatial frequency, f_m , and the orientation ϕ_n , with $m=1,2,\dots,N_f$ and $n=1,2,\dots,N_\phi$ [5]. Thus to estimate the features vector, firstly the captured image $(N \times M)$ is divided in $M_x M_y$ receptive fields each one of size $(2N_x+1) \times (2N_y+1)$ (Fig. 2), where $N_x=(N-M_x)/2M_x$, $N_y=(M-M_y)/2M_y$. This fact allows that the features vector size be independent of the captured image size. Next, the central point of each receptive field whose coordinates are given by (c_i, d_k) , where $i=1,2,\dots,N_x$; $k=1,2,3,\dots,N_y$, are estimated. Subsequently it is estimated the first point of the cross-correlation between each receptive field and the $N_f N_\phi$ Gabor functions which are given by

$$h_{m,\phi}(x, y) = g(x', y') \exp(j2\pi f_m(x'+y')) \quad (1)$$

where

$$(x', y') = ((x \cos \phi_n + y \sin \phi_n), (-x \sin \phi_n + y \cos \phi_n)) \quad (2)$$

The DGF, which are complex valued functions, can be represented as

$$h_{m,\phi}(x, y) = h_{m,\phi}^c(x, y) - j h_{m,\phi}^s(x, y) \quad (3)$$

where

$$h_{m,\phi}^c(x, y) = g(x', y') \cos(2\pi f_m(x'+y')) \quad (4)$$

$$h_{m,\phi}^s(x, y) = g(x', y') \sin(2\pi f_m(x'+y')) \quad (5)$$

772 J. Olivares-Mercado et al.

Equations (4) and (5) denote the real and imaginary DGF components. Next using eqs.(3)-(5) we can estimate the first component of the cross correlation function of each DGF with each receptive field as follows

$$\psi(u,v) = \sum_{x=-N_x}^{N_x} \sum_{y=-N_y}^{N_y} I(x-c_i, y-d_k) (h_{m,\phi}^c(x-c_i, y-d_k) - h_{m,\phi}^s(x-c_i, y-d_k)) \quad (6)$$

where $u=M_x(i-1)+k$ and $v=N_f(m-1)+n$. Next, to avoid complex valued data in the features vector we can use the fact that the magnitude of $\psi(u,v)$ presents a great similarity with the behavior of the complex cells of the human visual system. Thus the magnitude of $\psi(u,v)$ could be used. However, as shown in eq.(6) the number of elements in the features vector is still so large even for small values of M_x, M_y, N_ϕ y N_f . Thus to reduce the number of elements in the features vector, we can use the first point of the total cross correlation between each receptive field and the set of DGF, which can be obtained by taking the average of $\psi(u,v)$ with respect to v . Thus the features vector of proposed algorithm, $M(u)$, becomes

$$M(u) = \frac{1}{N_v} \sum_{v=1}^{N_v} |\psi(u,v)| \quad (7)$$

Where $N_v=N_fN_\phi$. Figure 3 illustrate this procedure.

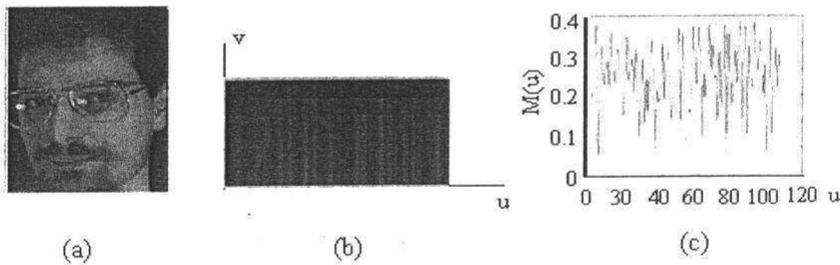


Fig. 3. a) Original image. b) Matrix containing the first point of cross correlation between DGF and receptive fields. c) Features vector of proposed algorithm.

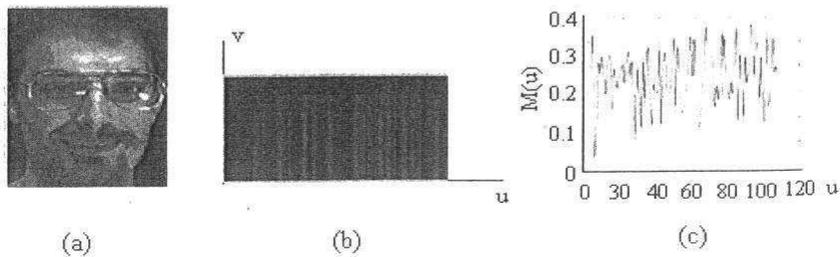


Fig. 4. a) Original image. b) Original image. b) Matrix containing the first point of cross correlation between DGF and receptive fields. c) Features vector of proposed algorithm.

2.2 Face Verification Stage

To perform the face verification task a GMM will be used because, the GMM, which consists of a sum of M weighted Gaussian density functions is able to approximate any probability distribution if the number of Gaussian components is large enough. Consider the GMM shown in Fig. 5 which is described by the following equation [6]:

$$p(\hat{x} | \lambda) = \sum_{i=1}^M p_i b_i(\hat{x}) \quad (8)$$

where \hat{x} is a N -dimensional vector, $b_i(\hat{x}), i=1,2,\dots,M$, are the density components and $p_i, i=1,2,\dots,M$, are the mixture weights. Each density component is a D -dimensional Gaussian function given as:

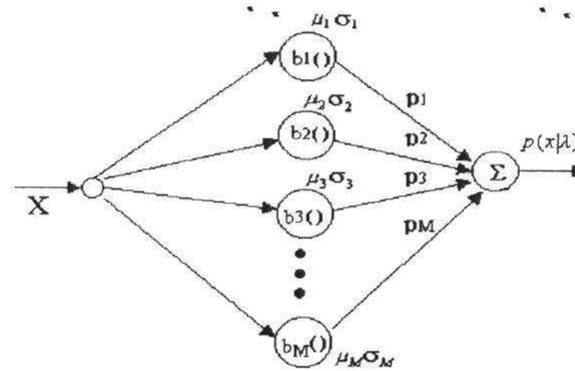


Fig. 5. Gaussian Mixture Model

$$b_i(\hat{x}) = \frac{1}{(2\pi)^{D/2} |\sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\hat{x} - \mu_i)' \sigma_i^{-1} (\hat{x} - \mu_i)\right\} \quad (9)$$

where $(\cdot)'$ denotes the transpose vector, μ_i is the mean vector and σ_i the covariance matrix which is assumed to be diagonal; and p_i denotes the mixture weights which satisfy that:

$$\sum_{i=1}^M p_i = 1 \quad (10)$$

The distribution model is determined by the mean vector, the covariance matrix and the distribution weights such that that the model of the face under analysis is given by

$$\lambda = \{p_i, \mu_i, \sigma_i\} \quad i = 1, 2, \dots, M \quad (11)$$

774 J. Olivares-Mercado et al.

The optimal parameter estimation is a non-linear problem, such that we need an iterative algorithm to estimate the optimal parameters of the face verification algorithm. Thus we can use the ML algorithm (Maximum Likelihood) to search for the parameters of the system providing the best approach to the model of the face under analysis. That is the target is to find the parameters of λ that maximize the a posteriori probability distribution. For a sequence of vectors T of training $X = \{x_1, \dots, x_T\}$, GMM likelihood can be written as:

$$p(X/\lambda) = \prod_{t=1}^T p(X_t/\lambda) \quad (12)$$

Unfortunately the equation (12) is non-linear in relation with the parameters λ . However, these can be estimated in an iterative way using the EM algorithm (Expectation-Maximization), in which starting from an initial set of parameters $\lambda(r-1)$ a new model is estimated $\lambda(r)$, where r denotes the r -th iteration, such that:

$$p(X/\lambda(r)) \geq p(X/\lambda(r-1)) \quad (13)$$

To accomplish this task, each T elements the GMM parameters of are updated as follows

Mixture weights:

$$p_i = \frac{1}{T} \sum_{t=1}^T p(i/X_{t+k}, \lambda) \quad (14)$$

Mean:

$$\mu_i = \frac{\sum_{t=1}^T p(i/X_{t+k}, \lambda) X_{t+k}}{\sum_{t=1}^T p(i/X_{t+k}, \lambda)} \quad (15)$$

Covariance:

$$\sigma_i = \frac{\sum_{t=1}^T p(i/X_{t+k}, \lambda) (X_{t+k} - \mu_i)^2}{\sum_{t=1}^T p(i/X_{t+k}, \lambda)} \quad (16)$$

Finally the probability to posteriori it is obtained for:

$$p(i/X_{t+k}, \lambda) = \frac{p_i b_i(X_{t+k})}{\sum_{j=1}^M p_j b_j(X_{t+k})} \quad (17)$$

During the testing phase we need to estimate the probability that the face under analysis corresponds to a given model, that is $P_r(\lambda|X)$. To this end consider the Bayes theorem which is given by

$$\hat{R} = \Pr(\lambda|X) = \frac{p(X|\lambda)\Pr(\lambda)}{p(X)} \quad (18)$$

where $P(X|\lambda)$ is the given by eq. (12), $P_r(\lambda)$ is the probability distribution of λ model, and $P(X)$ is the probability distribution the face under analysis. Assuming that all faces are equally probable then $P_r(\lambda)=1/R$. Next taking in account that $P(X)$ is same for all the face models, and substituting eq. (12) into eq. (18) it follows that

$$\hat{R} = p(X|\lambda) = \prod_{t=1}^T p(X_t|\lambda) \quad (19)$$

Finally taking the logarithm of eq. (19) we get

$$\hat{R} = \sum_{t=1}^T \log_{10}(p(X_t|\lambda)) \quad (20)$$

where $p(X|\lambda)$, which is given by (12), is the output of GMM system shown in the Fig 5.

3 Evaluation Results

The evaluation of proposed system was carried out by computer simulations using the database created by Olivetti Research Laboratory in Cambridge, UK (ORL), which consists of images of 30 people with 10 images of each one which differs on illumination, face rotation, different inclination, different hairstyle, wardrobe changes, etc. The images size is 128 x 128 pixels.

To do the proposed method robust against changes of sizes and translation; the algorithm firstly assumes that the gray level of picture background is constant. Next the algorithm estimates the position and size of the image by analyzing the gray label variation on the image. Once the image size and position have been estimated, the image is divided in 12x9 receptive fields, as shown in Fig. 2, whose central point will be always located in the space position (x, y) , where $x=0$ and $y=0$. After the image was divided in 108 receptive, the features vector is estimated using eqs. (1)-(6) with 9 phases $0, \pi/9, 2\pi/9, \pi/3, 4\pi/9, 5\pi/9, 2\pi/3, 7\pi/9$ and $8\pi/9$; and six normalized frequencies $\pi/2, \pi/4, \pi/8, \pi/16, \pi/32, \pi/64$. This produces a matrix with 5832 elements that are subsequently reduced to 108 using eq. (7), which are different and unique for each person and they are similar for the same person like it is shown in the figure 3 and 4.

Next the feature vector is it is applied to a GMM to obtain the model of the face under analysis. This is achieved substituting the feature vectors obtained in the Gaussian mixture model (GMM) to estimate the weights, the mean and variance as described by eq. (9)-(17). To training the GMM we assume that the 108 elements

776 J. Olivares-Mercado et al.

estimated in the feature extraction stage are divided in features vector of L elements as follows

$$S = \{X_0, X_1, X_2, \dots, X_T, X_{T+1}, X_{T+2}, X_{T+3}, \dots, X_{L-1}\} \quad (21)$$

Subsequently, using these vectors a group of vectors in L segments with T features vectors, X_t , each one, is formed in the following way:

$$S_0 = \{X_0, X_1, X_2, X_t, X_{t+1}, \dots, X_T\} \quad (22)$$

*

$$S_k = \{X_k, X_{k+1}, X_{k+2}, \dots, X_t, X_{t+1}, \dots, X_{T+k}\} \quad (23)$$

In this work we take $T=12$ in order that each 12 feature vectors the GMM parameters be updated.

Table 1. False rejection error

# of faces of training	# Gaussian Mixtures	Low threshold %	Half threshold %	high threshold %
1 face	9	12	8	5.1
	16	18.5	12.9	8.7
	32	12.4	9.2	6.8
3 faces	9	12	7.7	4.7
	16	13.25	9.1	5.9
	32	8.66	6.48	4.6
5 faces	9	9.3	5.9	4
	16	10.9	7	4.4
	32	5.9	4.6	3.9

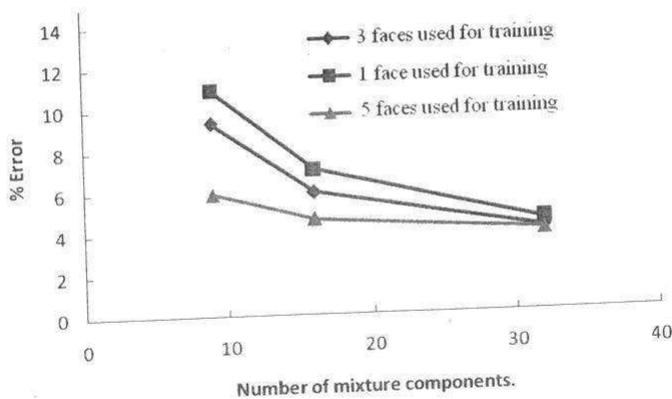


Fig. 6. Rejection error with different number of mixtures and 1, 3 and 5 training faces of each person

In the verification stage a threshold is used which depends of the face under analysis. To improve the verification performance this threshold will be divided in three categories: low, half and high thresholds. Two additional variants were also introduced for evaluation: one is the numbers of faces used for the training and the second one is the numbers of Gaussians mixtures to be used. Here the numbers of faces used are 1, 3, and 5; each one with 9, 16 and 32 Gaussian mixtures. Table I shows the results obtained using the database of 300 faces. These results correspond to the average error obtained during the test phase. Figure 6 shows that better results can be obtained when the number of mixtures and images used for training increase. Here is evident that in this situation the error decreases considerably using any of the three proposed thresholds.

Table 2. False acceptance error

# of faces of training	# Gaussian Mixtures	Low threshold %	Half threshold %	high threshold %
1 face	9	11.57	6.71	3.14
	16	18.5	12.35	7.7
	32	11.8	8.34	5.62
3 face	9	11.4	6.5	3.13
	16	12.5	8	4.5
	32	7.42	5	2.9
5 face	9	8.7	4.5	2.2
	16	10.2	5.8	2.9
	32	4.4	2.8	2

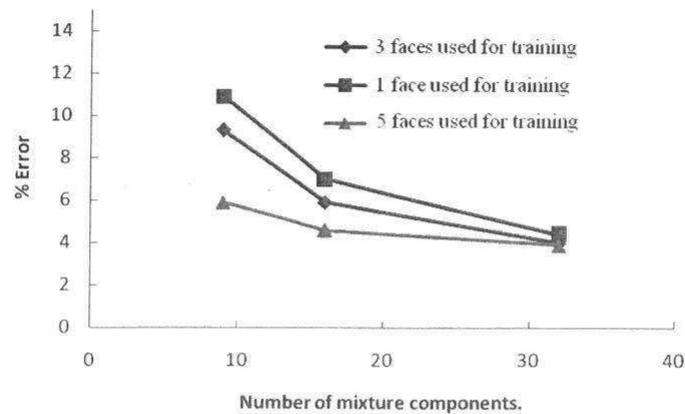


Fig. 7. False acceptance error with different number of mixtures and 1, 3 and 5 training faces of each person

Simulation results show that proposed algorithm performs fairly well in comparison with other previously proposed methods [1], [2], [6], even with faces that present an appreciable rotation, as happens in the ORL database.

778 J. Olivares-Mercado et al.

4 Conclusions

This paper proposed a face verification algorithm in which the Gabor functions are used for feature extraction and the GMM to perform the verification task. Evaluation results shows that the system achieves a false rejection error between 18.5% in the worst case cases to 3.9% cases at best, when a different number of images for training as well as different number mixtures in the GMM are used; while the false acceptance error obtained is between 18.5 and 2; depending on the number of mixtures and number of faces used for training. These results can be considered to be fairly good if we consider that the database used is composed by 300 different faces. This quantity of face is very similar to any database in a real application. Evaluation results show that the system performance increase when more faces are used for training the GMM and it has a larger number of mixtures. This is valid for false acceptance error as well as for false rejection error.

Acknowledgements

We thank the National Science and Technology Council and to the National Polytechnic Institute for the financial support during the realization of this research.

References

- [1] Reid, P.: *BIOMETRICS for Networks Security*, pp. 3–7. Prentice Hall, New Jersey (2004)
- [2] Chellappa, R., Wilson, C., Sirohey, S.: *Human and Machine Recognition of Faces: A Survey*. Proc. IEEE 83(5), 705–740 (1995)
- [3] Sashua, A.: *Geometry and Photometry in 3D Visual Recognition*. PhD thesis, Massachusetts Institute of Technology (1992)
- [4] Baron, R.J.: *Mechanisms of human facial recognition*. International Journal of Man-Machine Studies, 137–178 (1981)
- [5] Dunn, D., Higgins, W.E.: *Optimal Gabor Filters for Texture Segmentation*. In: IEEE Trans. Image Proc 4(7) (July 1995)
- [6] Kim, J.Y., Ko, D.Y., Na, S.Y.: *Implementation and Enhancement of GMM Face Recognition Systems Using Flatness Measure*. IEEE Robot and Human Interactive Communication (September 2004)
- [7] Reynolds, D.A., Rose, R.C.: *Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models*. IEEE Trans. Speech and audio Proc 3(1) (January 1995)

Feature Extraction using Gabor and DWT for GMM based Face Verification Algorithms

Jesus Olivares-Mercado, Gabriel Sanchez-Perez, Mariko Nakano-Miyatake, Hector Perez-Meana

Postgraduate Section of Mechanical Electrical Engineering School, Instituto Politécnico Nacional, Av. Santa Ana no. 1000 Col. San Francisco Culhuacan, Mexico D.F. Mexico
hmpm@prodigy.net.mx

(Paper received on July 01, 2007, accepted on September 1, 2007)

Abstract. This paper presents two feature extraction methods for developing face verification algorithms based on the Gaussian Mixtures Model (GMM). The first one using the discrete wavelet transform, (DWT) while the second is based on the discrete Gabor transform (DGT). In both cases, firstly the feature extraction is carried out using either the DGF or the DWT. Next the Gaussian Mixture Model (GMM) is used to perform the face verification task. Evaluation results using the standard data bases with different parameters, such as the mixtures number, the number of faces used for training as well as the transform used for feature extraction show that proposed system provides better results than other previously proposed systems with a correctly detections larger than 95%, using any of these transforms. Although, as happens in most face recognition systems, the verification rate decreases when the target faces present some rotation degrees.

1 Introduction

The development of security systems based on biometric features is currently a topic of active research, because it has a great importance in the development of the identity verification systems for access control, to enforce the security in restricted areas, and several other security applications. The terrorist attacks that have happened during the last decade have done evident the necessity of developing more reliable security systems, in offices, banks, companies, trades, etc. Among them the identity verification based on biometric methods appear to be a good alternative for the development of such security systems.

The biometrics systems consist of a group of automated methods for recognition or verification of people identity using physical characteristics or personal behavior of the person under analysis [1]. This technology is based on the fact that each person is unique and possesses distinctive features that can be used to identify her/him. Following these ideas several biometric based security systems have been developed using fingerprints, iris, voice hand and face features. Among them, the face verification systems appear to be a desirable alternative because in is non-invasive and its computational complexity is low, it is the biometric method easier of understanding

106 J. Olivares Mercado et al.

since for us the face is the most direct way to identify people; besides that the data acquisition of this method consists on taking a picture, doing it one of the biometric methods with larger acceptance among the users.

The recognition is a very complex activity of the human brain. For example, we can recognize hundred of faces learned throughout our life and to identify familiar faces at the first sight even after several years of separation with relative easy. However for a computer it is not a simple task. For instance, recently proposed face recognition systems, achieve a recognition rate of about 91% when the face image is not rotated or the rotation is relatively low. However although, this recognition rate is good enough for several practical applications, it may be so large for applications where the security should be extreme; such that we cannot tolerate a high erroneous recognition average. This paper proposes a face recognition algorithm that is able of achieving an erroneous verification rate below 9%. Several methods have been proposed for face recognition [2], [3] such as the methods based on statistical correlation of the geometry [4]; the face form which uses the distances among the position of the eyes, mouth, nose, etc. as well as those using the neuronal networks technology that trait to imitate the operation of the human brain [2]. Many of these systems can recognize a person even when they present some physical changes, such as the growth of the beard or mustache, changes in the color or the style of the hair, the use of glasses, etc. Although in general are sensitive to rotations of the face images.

Before starting the proposed methods analysis used for face recognition, it is necessary to point out the verification concept. In face verification, the person says to the system about his/her identity, presenting an identification card or writing a special password. The system holds the person's features (for example the persons face in this case), and then proceeds to solve if the person is who (his/her) claims to be.

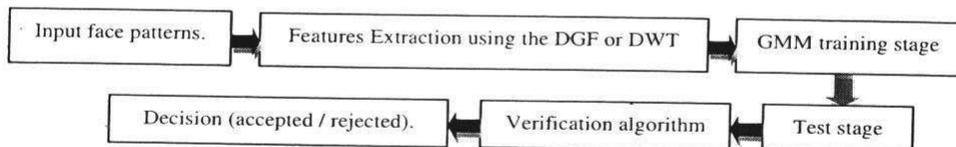


Fig. 1. Proposed face verification algorithm.

2 Proposed System

This section provides a detailed description of the proposed face verification algorithm which consists of three stages. Firstly a feature extraction of the face is carried out, using either the Gabor discrete transform (DGF) or discrete wavelet transform (DWT). Next using these features vectors, a model for each face is obtaining using a Gaussian Mixtures Model (GMM). Finally during the verification process, the GMM output is used to take the final decision. Figure 1 shows the block diagram of proposed algorithm.

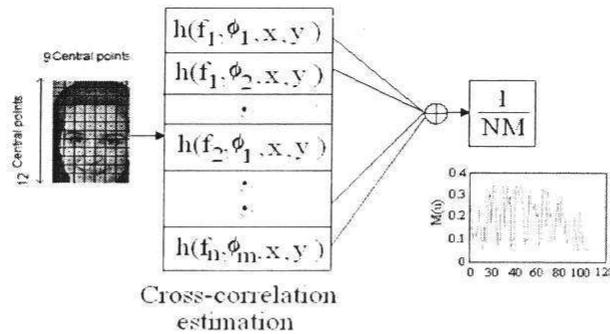


Fig. 2. DGT model.

2.1 Feature extraction with DGT

The feature extraction stage is one of the most important steps in any pattern recognition system. To this end, the proposed algorithm uses the DGT which has some relation with the human visual system (HVS). The two dimensional discrete Gabor functions (2D-DGF) depends on four parameters, two of them express their localization in the space (x, y) while the other two express the spatial frequency, f_m , and the orientation ϕ_n , where $m=1,2,\dots,N_f$ and $n=1,2,\dots,N_\phi$ [5]. Thus to estimate the features vector, firstly the captured image $(N \times M)$ is divided in $M_x M_y$ receptive fields each one of size $(2N_x + 1) \times (2N_y + 1)$ (Fig. 2), where $N_x = (N - M_x) / 2M_x$, $N_y = (M - M_y) / 2M_y$. This fact allows that the number of elements in the features vector be independent of the captured image size. Next, the central point of each receptive field whose coordinates are given by (c_i, d_k) , where $i=1,2,\dots,N_x$; $k=1,2,3,\dots,N_y$, are estimated. Subsequently the first point of the cross-correlation $\psi(u, v)$ between each receptive field and the $N_f N_\phi$ Gabor functions $h_{m, \phi}(x, y)$ is estimated, where

$$h_{f, \phi}(x, y) = g(x', y') \exp(2\pi i f_m (x' + y')) \quad (1)$$

where denotes the Gabor function, and

$$(x', y') = ((x \cos \phi_n + y \sin \phi_n), (-x \sin \phi_n + y \cos \phi_n)) \quad (2)$$

As shown in Fig. 1, $N_f N_\phi$ correlations are estimated for each receptive field, leading to an extremely large features vector. Thus to reduce the elements in the features vector, the first point of the total cross correlation between each receptive field and the set of DGF is used, which can be obtained taking the average of $\psi(u, v)$ with respect to v . Therefore the proposed algorithm features vector $M(u)$, is given by

$$M(u) = \frac{1}{N_v} \sum_{v=1}^{N_v} |\psi(u, v)| \quad (3)$$

108 J. Olivares-Mercado et al.

where $N_v = N_f N_\phi$. To do the proposed method robust against changes of sizes and translation; the algorithm firstly assumes that the gray level of picture background is constant. Next the algorithm estimates the position and size of the image by analyzing the gray label variation on the image. Once the image size and position have been estimated, the image is divided in 12×9 sections, as shown in Fig. 2, whose central point will be always located in the space position (x, y) , where $x=0$ and $y=0$. After the image was divided in 108 sections, the features vector was estimated with 9 phases and 6 normalized frequencies as mentioned before. This produces a matrix with 5832 elements that are subsequently reduced to 108 using eq. (3).

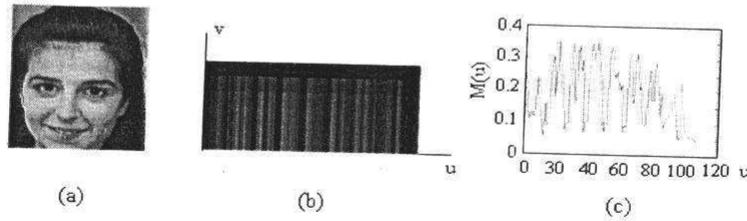


Fig. 3. a) Original image. b) Cross correlation matrix between DGF and receptive fields. c) Features vector obtained of proposed algorithm.

2.2 Feature extraction with DWT

The most commonly used set of discrete wavelet transforms was formulated by the Belgian mathematician Ingrid Daubechies in 1988. This formulation is based on the use of recurrence relations to generate progressively finer discrete samplings of an implicit mother wavelet function; each resolution is a half of that of the previous scale.

The DWT of a given signal x is estimated by passing it through a series of low pass and high pass filters (Fig. 4). First the samples are passed through a low pass filter with impulse response $g(n, m)$ resulting in a convolution of the two. The signal is also decomposed simultaneously using a high-pass filter $h(n, m)$. The detail coefficients are the high-pass filter outputs and the approximation coefficients are the low-pass ones. It is important that the two filters, related to each other, are known as a quadrature mirror filter. However, since half the frequencies of the signal have now been removed, half the samples can be discarded according to Nyquist's rule. The filter outputs are:

$$Y_{LOW}(n, m) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x(n, m) g(2n - k, 2m - j) \quad (4)$$

$$Y_{HIGH}(n, m) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x(n, m) h(2n - k, 2m - j) \quad (5)$$

This decomposition reduces the spatial resolution since only a quarter of each filter output allows characterizing the face image. However, because each output has band width equal to a quarter of the original one, the output image can be decimated to reduce the image size.

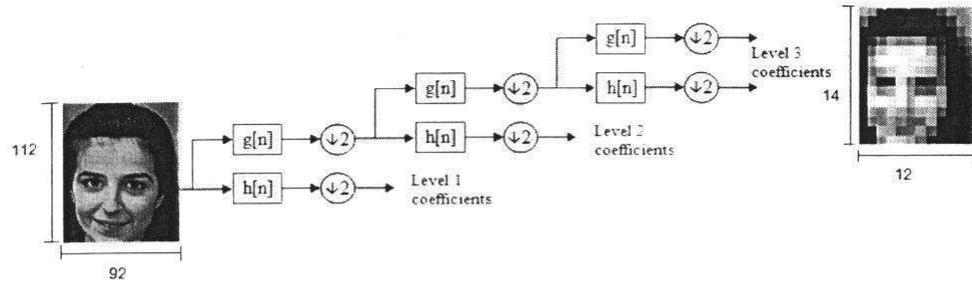


Fig 4. 3 level wavelet decomposition

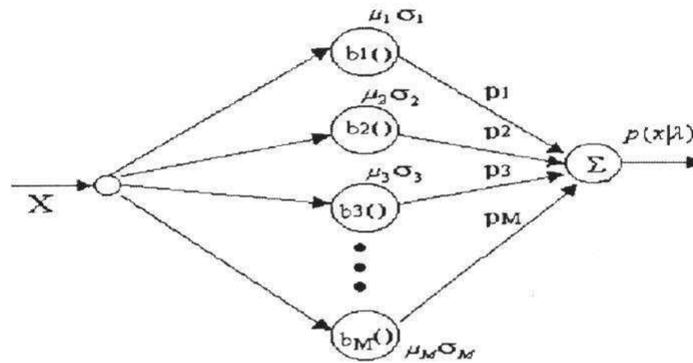


Fig. 5. Gaussian Mixture Model

Here only the approximation coefficients are used to characterize the face image. This decomposition is repeated to further increase the frequency resolution and the approximation coefficients decomposed with high and low pass filters and then down-sampled. This is represented as a binary tree with nodes representing a sub-space with different time-frequency localization. The tree is known as a filter bank.

At each level in the above diagram the signal is decomposed into low and high frequencies. Due to the decomposition process the input signal must be a multiple of 2^n where n is the number of levels. For example a signal with 32 samples, frequency range 0 to f_n and 3 levels of decomposition, 4 output scales are produced:

2.3 Face verification stage

To perform the face verification task a GMM will be used because, the GMM, which consists of a sum of M weighted Gaussian density functions, is able to approximate any probability distribution if the number of Gaussian components is large enough. Consider the GMM shown in Fig. 5 which is described by the following equation [6]:

110 J. Olivares-Mercado et al.

$$p(\bar{x} / \lambda) = \sum_{i=1}^M p_i b_i(\bar{x}) \quad (6)$$

where \bar{x} is a N-dimensional vector, $b_i(\bar{x}), i=1,2,\dots,M$, are the density components and $p_i, i=1,2,\dots,M$, are the mixture weights. Each density component is a D-dimensional Gaussian function given as:

$$b_i(\bar{x}) = \frac{1}{(2\pi)^{D/2} |\sigma_i|^{1/2}} \exp \left\{ -\frac{1}{2} (\bar{x} - \bar{\mu}_i)' \sigma_i^{-1} (\bar{x} - \bar{\mu}_i) \right\} \quad (7)$$

Where ()' denotes the transpose vector, μ_i denotes the mean vector and σ_i the covariance matrix which is assumed to be diagonal; and p_i are the mixture weights which satisfies that $\sum_{i=1}^M p_i = 1$. The model of the distribution is determined by the mean vector, the covariance matrix and the distribution weights for a face model is given by $\lambda = \{p_i, \mu_i, \sigma_i\} \quad i=1,2,\dots,M$. The estimation of the optimal parameter is a non-linear problem, such that we need an iterative algorithm to estimate the optimal parameters of the face verification algorithm, and the Maximum Likelihood algorithm (ML) is used to search the optimal parameters of the system, providing the best approach to the face model under analysis. So then the goal is to find the best parameters of λ that maximize the a posteriori probability distribution. For a sequence of vectors T of training $X = \{x_1, \dots, x_T\}$, GMM likelihood can be written as [7]:

$$p(X / \lambda) = \prod_{i=1}^T p(X_i / \lambda) \quad (8)$$

Equation (8) is not lineal in relation of the parameters λ , then is necessary to carry out the estimation in an iterative way using the Expectation-Maximization algorithm (EM), in which starting from an initial set of parameters $\lambda(r-1)$ and a new model is estimated $\lambda(r)$, where r denotes the r-th iteration, so that:

$$p(X / \lambda(r)) \geq p(X / \lambda(r-1)) \quad (9)$$

To achieve this goal, each T partial feature vectors X_i , of the GMM parameters are updated [7]. During the testing phase we need to estimate the probability that one face under analysis corresponds to a given model, that is $P_r(\lambda/X)$. To achieve this, the Bayes theorem is used, then obtain:

$$\hat{R} = \sum_{i=1}^T \log_{10} (p(X_i / \lambda)) \quad (10)$$

where $p(X_i/\lambda)$ is the conditional probability of the face X given by face model λ , this is the GMM system response shown in the Fig. 6.

3 Evaluation Results

The evaluation of proposed system was carried out by computer simulations using the database created by Olivetti Research Laboratory in Cambridge, UK (ORL), which consists of images of 30 people with 10 images of each one which differs on, face rotation, different inclination, etc. The images have a size of 92 x 112 pixels.

After the feature vector is estimated using the DGT or the DWT it is applied to a GMM for obtaining the model of the face under analysis. This is achieved introducing the estimated vectors to the Gaussian mixture model (GMM) to obtain the weights, the mean and variance as described in [6], [7], where for training we assume that the 108 elements obtained in the feature extraction stage are divided in partial features vector of L elements as follows

$$S = \{X_0, X_1, X_2, \dots, X_T, X_{T+1}, X_{T+2}, X_{T+3}, \dots, X_{L-1}\} \quad (11)$$

Subsequently, form a group of vectors in L segments with T features vectors, X_t , each one in the following way:

$$S_0 = \{X_0, X_1, X_2, X_t, X_{t+1}, \dots, X_T\} \quad (12)$$

$$S_k = \{X_k, X_{k+1}, X_{k+2}, \dots, X_t, X_{t+1}, \dots, X_{T+k}\} \quad (13)$$

In this work the system uses $T=12$ in order that each 12 feature vectors of the GMM parameters being updated. In the table 1 and 3 the face rejection results using the DGT DWT respectively are shown, the Fig. 2 and 4 show the false acceptance rate using DGT and DWT respectively.

Table 1. False rejection error using DGT.

# of faces of training	# Gaussian Mixtures	Low threshold %	Half threshold %	high threshold %
1 face	9	12	8	5.1
	16	18.5	12.9	8.7
	32	12.4	9.2	6.8
3 faces	9	12	7.7	4.7
	16	13.25	9.1	5.9
	32	8.66	6.48	4.6
5 faces	9	9.3	5.9	4
	16	10.9	7	4.4
	32	5.9	4.6	3.9

112 *J. Olivares-Mercado et al.*

Table 2. False acceptance error using DGT.

# of faces of training	# Gaussian Mixtures	Low threshold %	Half threshold %	high threshold %
1 face	9	11.57	6.71	3.14
	16	18.5	12.35	7.7
	32	11.8	8.34	5.62
3 face	9	11.4	6.5	3.13
	16	12.5	8	4.5
	32	7.42	5	2.9
5 face	9	8.7	4.5	2.2
	16	10.2	5.8	2.9
	32	4.4	2.8	2

Table 3. False rejection error using DWT.

# of faces of training	# Gaussian Mixtures	Low threshold %	Half threshold %	high threshold %
1 face	12	4.55	4	3.41
	16	5.53	4.2	3.3
	32	9.8	7.9	6.1
3 faces	12	1.5	2.18	2.26
	16	8.8	7.32	6.08
	32	16.7	14.8	13.3
5 faces	12	6.7	5.9	5.3
	16	3.6	3.2	2.9
	32	3.4	3	2.9

Table 4. False acceptance error using DWT.

# of faces of training	# Gaussian Mixtures	Low threshold %	Half threshold %	high threshold %
1 face	12	3.82	2.63	1.74
	16	4.54	3	1.9
	32	9.2	7.11	5.1
3 face	12	0.66	0.56	0.43
	16	8	6.45	5.11
	32	16.3	14.3	12.7
5 face	12	5.8	4.3	3.34
	16	1.31	0.9	0.44
	32	1.45	1	0.8

In the verification stage a threshold is used which depends of the face under analysis. Top improve the verification performance this threshold may be divided in three categories: low, half and high thresholds. Two variants more were introduced for evaluation: one is the numbers of faces used for the training and the second one is the numbers of Gaussians mixtures to be used and these variants are applied in the two used techniques

Simulation results show that proposed algorithm performs fairly well in comparison with other previously proposed methods [1], [2], [6], even with faces that present an appreciable rotation, as happens in the ORL database. We can also see that there is not a much difference between using DWT and DGT.

4 Conclusions

This paper presented two face verification algorithms in which the DGT or DWT are used for feature extraction and the GMM to perform the verification task. Evaluation results obtained were very different with each one of the variants proposals, from 18.5% in the worst case, until 3.9% in the best case, using DGT, and 16.7% in the worst case, until 2.9% in the best case, using DWT. These results are very satisfactory if we consider that the database used is composed by 30 persons. This quantity of face is very similar to any database in a real application. In the case of accepting a person with a false identity we have a percentage of error of 18.5% at worst case and a 2% in the best one using the DGT; while using the DWT we have a 16.3% cases at worst case; and a 0.44% in the best one using DWT. In summary, can observe that the system performance becomes better when more faces are used for training the GMM and it has a larger number of mixtures. This is valid for false acceptance error as well as for false rejection error.

Acknowledgements

We thanks the National Science and Technology Council and to the National Polytechnic Institute for the financial support during the realization of this research.

References

1. P. Reid: BIOMETRICS for Networks Security, New Jersey: Prentice Hall, 2004, pp 3-7.
2. R. Chellappa, C. Wilson, and S. Sirohey: Human and Machine Recognition of Faces: A Survey, Proc. IEEE, vol. 83, no. 5, pp. 705-740, 1995.
3. A. Shashua: Geometry and Photometry in 3D Visual Recognition, PhD thesis, Massachusetts Institute of Technology, 1992.
4. Robert J. Baron: Mechanisms of human facial recognition. International Journal of Man-Machine Studies, pp: 137-178, 1981.
5. Dunn, D., Higgins, W. E.: Optimal Gabor Filters for Texture Segmentation, IEEE Trans. Image Proc., Vol. 4, No. 7, Jul. 1995.
6. Jin Y. Kim, Dae Y. Ko, Seung Y. Na.: Implementation and Enhancement of GMM Face Recognition Systems Using Flatness Measure, IEEE Robot and Human Interactive Communication, September 2004.
7. Douglas A. Reynolds, Richard C. Rose: Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models, IEEE Trans. Speech and audio Proc., Vol. 3, No. 1, Jan. 1995.

114 *J. Olivares-Mercado et al.*

8. Busso C., Narayanan S. S.: Interrelation Between Speech and Facial Gestures in Emotional Utterances: A Single Subject Study, *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. PP 2007.
9. Yan, S., Xu, D., Tang, X.: Face Verification With Balanced Thresholds, *Transactions on Image Processing*, IEEE, Vol. 16, No. 1, Pag. 262- 268, Jan 2007.
10. Kar, Soumitra, Hiremath, Swati, Joshi, Dilip G.: Chadda, Vinod.K.; Bajpai, Apurva, "A Multi-Algorithmic Face Recognition System", *International Conference on Advanced Computing and Communications*, Pag. 321 – 326, December 2006.

Face Recognition Based on Normalization and the Phase Spectrum of the Local Part of an Image

Jesus Olivares-Mercado^{1,2}, Kazuhiro Hotta², Haruhisa Takahashi²,
Hector Perez-Meana¹, Mariko Nakano Miyatake¹, and Gabriel Sanchez-Perez¹

¹ ESIME Culhuacan, Instituto Politecnico Nacional, Av. Santa Ana No. 1000,
Col. San Francisco Culhuacan, 04430 Mexico D.F. Mexico
`hmpm@calmecac.esimecu.ipn.mx`

² The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu,
Tokyo 182-8585, Japan
`hotta@ice.uec.ac.jp`, `takahasi@ice.uec.ac.jp`

Abstract. This paper proposes a robust faces recognition method based on the *Normalization and the Phase Spectrum of the Local Part of an Image*. The *Principal Components Analysis (PCA)* and the *Support Vector Machine (SVM)* are used in the classification stage. We evaluate how the proposed method is robust to illumination, occlusion and expressions using “AR Face Database”, which includes the face images of 109 subjects (60 males and 49 females) under illumination changes, expression changes and partial occlusion. The proposed method provides results with a correct recognition rate more than 96.7%.

1 Introduction

The development of security systems based on biometric features has been a topic of active research during the last three decades, because the recognition of the people identity to access control is a fundamental aspects in these days. The terrorist attacks happened during the last decade have demonstrated that it is indispensable to have reliable security systems in offices, banks, airports, etc.; increasing in such way the necessity to develop more reliable methods to people recognition. The biometrics systems consist of a group of automated methods for recognition or verification of people identity using physical characteristics or personal behavior of the person under analysis [1]. In particular the face recognition has been a topic of active research because the face is the most direct way to recognize the people. In addition, the data acquisition of this method consists in taking a picture, doing it one of the biometric methods with larger acceptance among the users. Various kinds of face recognition methods have been proposed [2,3]. In recently years, faces are recognized with high accuracy recognition. However, partial occlusion, illumination variations and expression decrease the accuracy drastically.

The recognition is a very complex activity of the human brain. For example, we can recognize hundred of faces learned throughout our life and to identify familiar

faces at the first sight, even after several years of separation, with relative ease. However it is not a simple task for a computer. For instance, recently proposed face recognition systems, achieve a recognition rate of about 90% when the face in the image is not rotated or the rotation is relatively low [4]. Although this recognition rate is good enough for several practical applications, it may be not large enough for applications where the security should be extreme; such that we cannot tolerate a high erroneous recognition rate. In particular, illumination, occlusion, facial expressions are big obstacles in the practical environment. This paper proposes a face recognition algorithm that is able of achieving an erroneous recognition rate below 5% with this characteristics.

In recent years, the robust face recognition method using phase spectrum has been proposed [5]. However, they extracted phase information from global image. Global features are influenced easily by illumination, occlusion and expression [6,7]. Therefore, we use the device for emphasizing the local features of a face image before extracting phase spectrum. Concretely, the norm of local region is normalized in advance. This process is effective for face recognition. When the part of a face is occluded by something such as sunglasses or scarf, global similarity between images is much influenced. In local normalized image, the similarities of some local regions are influenced by occlusion but the similarities of almost local regions are not influenced. Therefore, the local normalized image is robust to partial variations. After normalizing the norm of local region, we extract the phase information in the frequency domain representation. This method is very useful because the phase spectrum of an image retains the most of the intelligibility of this. Furthermore, we can increase the recognition accuracy using the PCA to obtain the main characteristics of the training faces. The features obtained by PCA are fed into one-vs-all SVM.

The proposed method is evaluated using “AR Face Database” [8]. The data base includes the face images with partial occlusion, illumination variations and expression. The phase spectrum of local normalized image outperforms conventional eigenphase method [5]. In particular the robustness to partial occlusion is also improved. This shows the effectiveness of the proposed approach.

In section 2, the details of proposed system are explained. Experimental results are shown in section 3. Finally, section 4 describes conclusions.

2 Proposed System

This section provides a detailed description of the proposed face verification algorithm which consists in two modules of four stages each one. Figure 1 shows the block diagram of proposed algorithm.

Firstly we normalize the norm of the local region of the image. The phase spectrum of the same local region is extracted, after that, the Principal Components Analysis (PCA) [9,10] is applied to the phase spectrum to obtain a dominant feature of the faces. Next, the features in principal components space are fed into classifier based on Support Vector Machine (SVM) [11,12]. In the final step of training, one-vs-all SVM is used to classify multi-classes. In the final step of

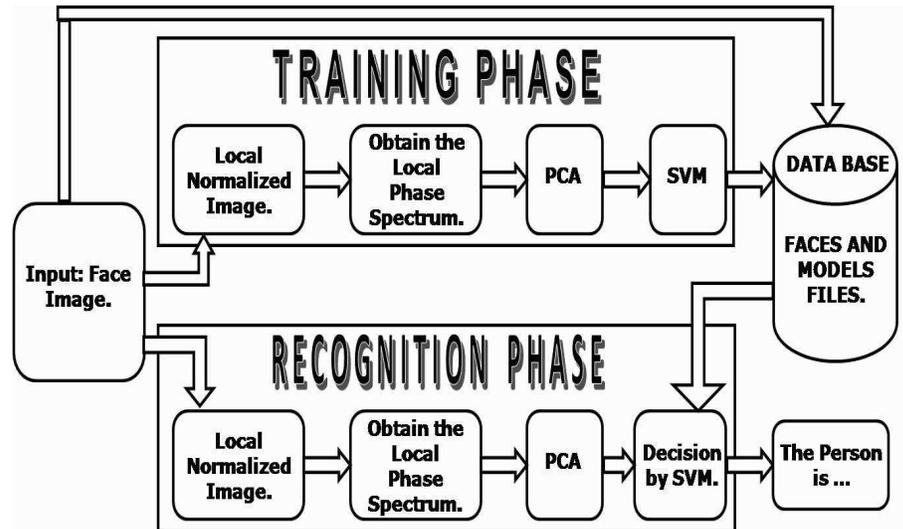


Fig. 1. Proposed face recognition algorithm

test, PCA features of phase spectrum of a test image are fed into all SVMs and the test image is classified to the class given maximum likelihood.

2.1 Feature Extraction Stage

The feature extraction plays a very important role in the any pattern recognition system. To this end, the proposed algorithm is based in the Normalization and the Phase Spectrum of local regions of an Image. To obtain local normalized image, the norm of local regions of $M \times M$ pixels where $M = 3, M = 6$ and $M = all$ are normalized to 1 and then the phase spectrum is computed. Figure 2 shows this process. By normalizing the norm and obtain the phase spectrum of local regions, it becomes robust to partial variation such as illumination changes or occlusion. We note local region as $I(x)$ where x is the center position of local region, the local norm is normalized as:

$$I(x)' = \frac{I(x)}{\|I(x)\|} \quad (1)$$

where $\|I(x)\|$ is the norm of the local region. All the training images must be normalized before training. In the test, we need to apply the same process before extracting the phase spectrum. In this paper we consider the normalization and phase spectrum extraction of local region of an image to improve the accurate classification and robustness to partial variations.

After normalizing the norm of local region, we extract the phase spectrum of the same local region. This can be computed through of a Fourier Transform

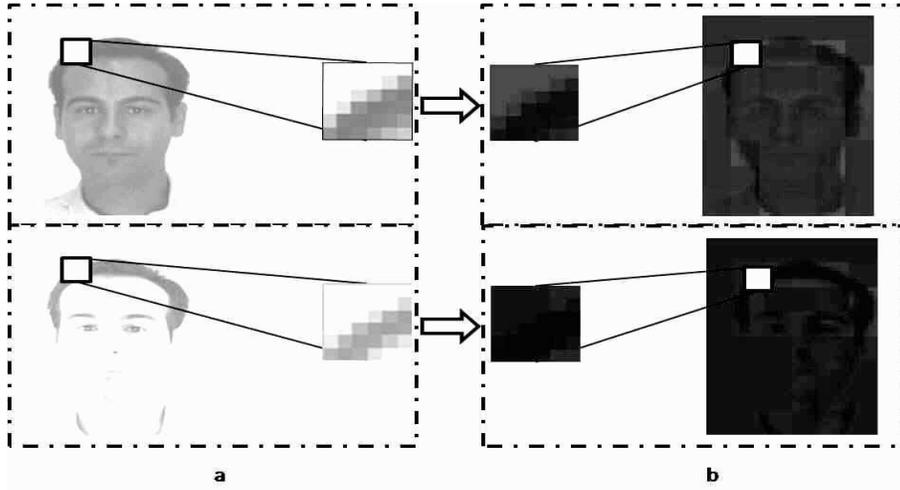


Fig. 2. a) Original image. b) Image after apply the normalization.

which are given by:

$$F(u) = |F(u) \exp^{j\phi(u)}| \quad (2)$$

where

$$|F(u)| = [R^2(u) + I^2(u)]^{\frac{1}{2}} \quad (3)$$

is the magnitude, and

$$\phi(u) = \arctan \left[\frac{I(u)}{R(u)} \right] \quad (4)$$

is the phase, is needed extract the phase spectrum of every images of both training and testing. Oppenheim et. al [13,14] have shown that phase information of an image retains the most of the intelligibility of an image. This is also demonstrated by Oppenheim's experiment shown in Figure 3.

Those two face images are obtained from the "AR Face Database" [10]. We apply the Fourier Transform to these two images and obtain magnitude and the phase. We combine the phase of the image 1 with the magnitude of the image 2 and the phase of the image 2 with the magnitude of the image 1. Thus we see that the component that provides more information about the image has been the phase. This shows that phase information is more effective than magnitude.

2.2 Classification Stage

To perform the face classification task, a PCA is used to obtain the main characteristics of the faces training, Figure 4 shows the process:

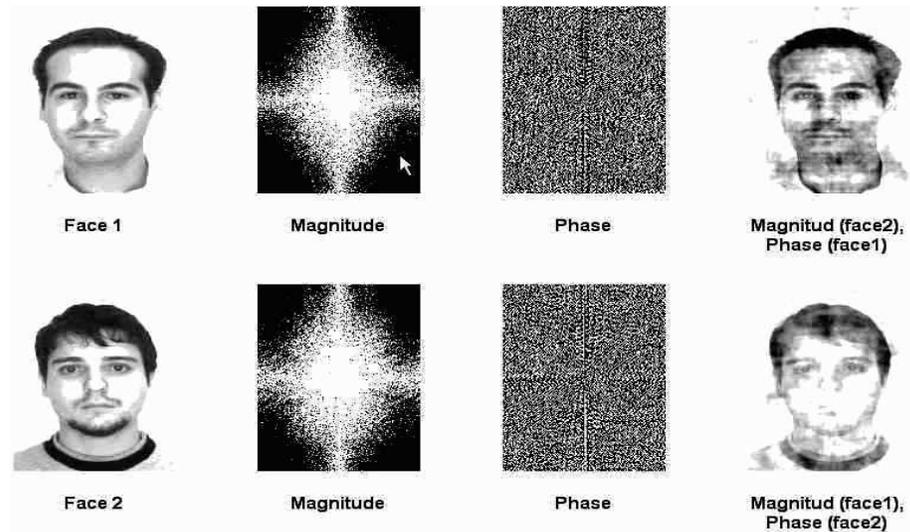


Fig. 3. Oppenheim's experiment

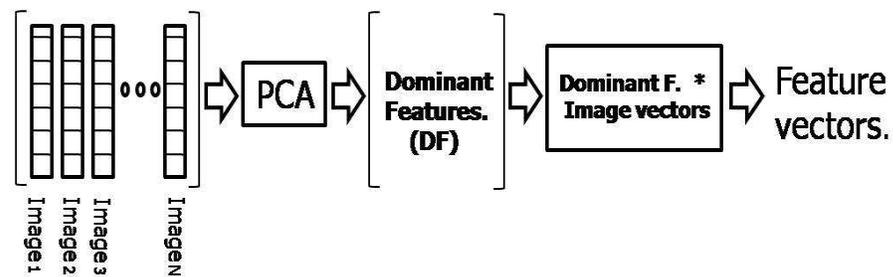


Fig. 4. Scheme of feature extraction by PCA

Image 1, Image 2...Image N in Figure 4 are the phase spectrum of the training faces. In training phase, basis vectors are obtained by PCA. In the testing phase, the basis vectors obtained in training phase are used to extract the features for SVM.

After extracting the features by PCA, SVM is used to classify a test face image. Figure 5 shows classification phase. S is the feature vector of the person to recognize. S is applied to all one-vs-all SVMs.

The class given the Maximum Likelihood is used as the person's identity, the equation to obtain the Maximum Likelihood is as follows:

$$\hat{S} = \arg \max_{1 \leq k \leq S} P(\lambda_k | x) \quad (5)$$

where \hat{S} is the winner and thus reveals the person's identity to whom this picture was assigned, x is the column vector of the image to analyze and λ_k is the SVM model of the person k .

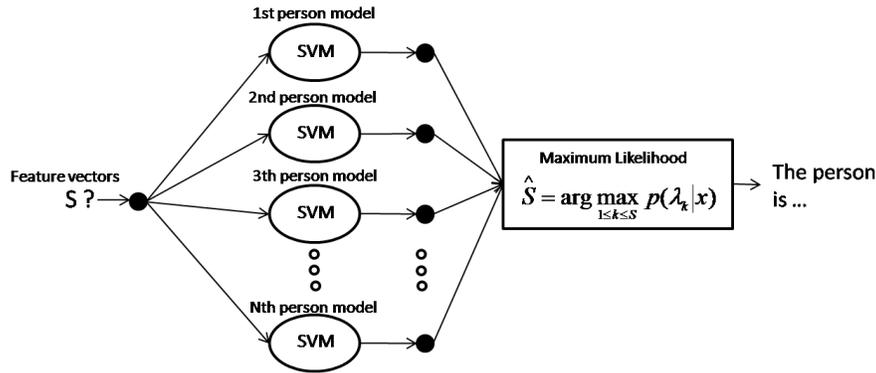


Fig. 5. Classification phase by SVM

3 Evaluation Results

The evaluation of proposed system was carried out by computer simulations using “AR Face Database”, created by Purdue University, USA. The database includes the face images of 109 subjects (60 males and 49 females) under illumination changes, expression changes and partial occlusion. The images size is 288×384 pixels. In this paper, we resize the image of a 48×36 pixels for processing. Each subject has 39 images, in the followings experiments, 6 images per subject are used in training and the remaining 33 images per subject are used in test.

The algorithm assumes that the gray level of picture background is constant. First the algorithm estimates the normalized and extract the phase spectrum of the image. Once the normalized image and phase spectrum have been estimated, the image is converted in a column vector and so on with all the training images to implement the process to obtain the features vectors as shown in Fig. 4.

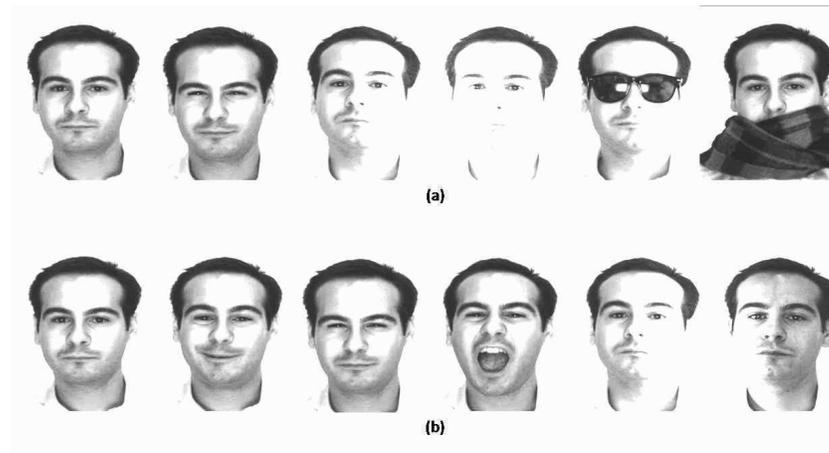
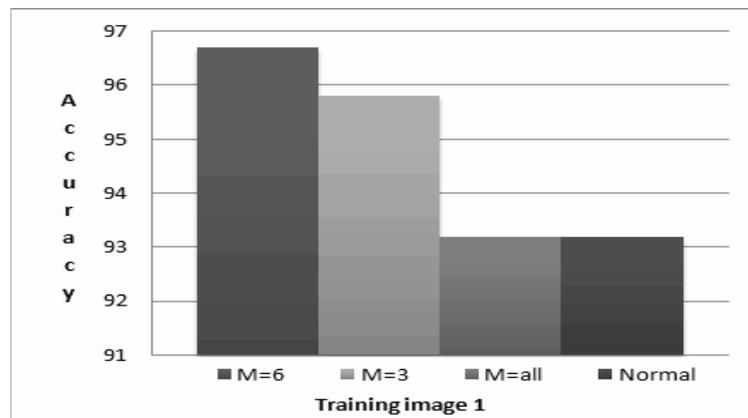
Next, the feature vectors of training images are applied to a SVM to obtain the model of the each class, these models are used in the classification stage, where the input of each one is the feature vector of the face to classify and the output is a probability, and the class to which it belongs is obtained by selecting maximum likelihood as shown in Fig. 5.

In this work we take 3 different variants in the window size to make the local normalized image, and comparing these with the method that only uses the phase spectrum of standard image. We try $M = 3$, $M = 6$ and $M = all$. $M = all$ means that the window is equal to the size of the image. The local norm is normalized with non-overlap manner. We take another option which is “Normal” as shown in Table 1, “Normal” means that the image will be processed without being normalized. Namely, this is nearly same as eigenphase approach [5].

The table 1 shows the results with 2 sets of images of training, in the first set of images (training images 1) takes 6 images in which illumination variations, expressions, sunglasses and scarf are included. The Figure 6(a) shows the

Table 1. Accuracy on face recognition

	M=6	M=3	M=All	Normal
Training images 1	96.7	95.8	93.2	93.2
Training images 2	83.2	82.5	75	75

**Fig. 6.** a) Training images 1. b) Training images 2.**Fig. 7.** Accuracy using the Training images 1

example of training images 1. The recognition accuracy of $M = 6$ achieves 96.7%, our method outperforms the eigenphase method. The improved rate is 3.5%.

The second set of images (training images 2) takes 6 images which only illumination variations and expressions are included. Therefore the accuracy to the second set shows the robustness to partial occlusion. Figure 6(b) shows the

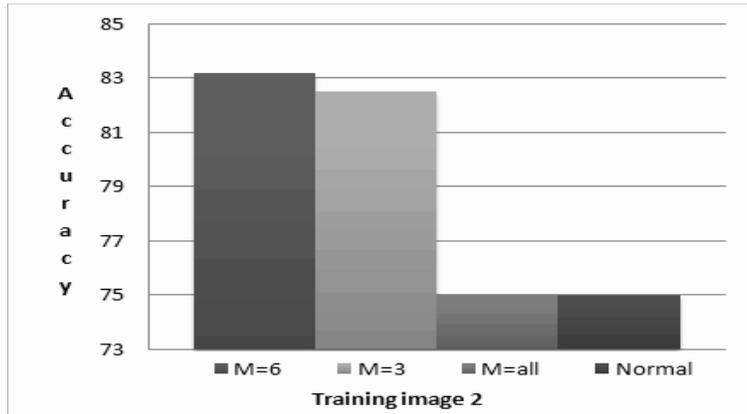


Fig. 8. Accuracy using the Training images 2

example of the second training set. Table 1 and figure 8 shows accuracy to training image 2. The recognition accuracy of $M = 6$ achieves 83.2%. The improved rate is 8.2%. The difference from eigenphase method becomes larger than that in the training images 1. This shows that the local normalizes image has the robustness to partial variations.

Simulation results show that proposed algorithm performs fairly well in comparison with other previously proposed methods [4,5], even with faces that present different illumination, expression and partial occlusion. Although the part of a face is not visible because of the accessories such as sunglasses and scarf, the proposed method achieves high accuracy.

4 Conclusions

This paper proposed a face recognition algorithm based on the *Normalization and the Phase Spectrum of the Local Part of an Image*. PCA is used for dominant feature extraction and the SVM is used to perform the recognition task. Evaluation results shows that the proposed system achieves a accuracy between 93.2% in the worst case and 96.7% at best when we use the training image 1. We obtain a accuracy between 75% in the worst case and 83.2% at best when we use the training image 2 in which face images with partial occlusion are not included.

Hence, we can see that the proposed system improved about 3% to 8% by taking the normalization and the Phase Spectrum of the Local Part of an Image. Although the database includes the face images of 109 subjects with various variations such as illumination changes, facial expressions and partial occlusion, high accuracy is obtained. This is the main contribution in this work.

Acknowledgments. We thank the National Science and Technology Council, the National Polytechnic Institute of Mexico and the University of Electro-Communications for the financial support during the realization of this research.

References

1. Reid, P.: BIOMETRICS for Networks Security, pp. 3–7. Prentice Hall, New Jersey (2004)
2. Chellappa, R., Wilson, C.L., Sirohey, S.: Human and machine recognition of faces: A survey. *Proceedings of the IEEE* 83(5), 705–740 (1995)
3. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. *ACM Computing Surveys* 35(4), 399–458 (2003)
4. Aguilar-Torres, G., Sanchez-Perez, G., Nakano-Miyatake, M., Prez-Meana, H.: Face Recognition Algorithm Using the Discrete Gabor Transform. In: *Conielectcomp*, p. 35 (2007)
5. Savvides, M., Vijaya Kumar, B.V.K., Khosla, P.K.: Eigenphases vs. Eigenfaces. In: *Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004)*, August 2004, vol. 3 (2004)
6. Hotta, K.: Robust face recognition under partial occlusion based on support vector machine with local Gaussian summation kernel. *Image and Vision Computing* (in press, 2008)
7. Martinez, A.M.: Recognizing imprecisely localized partially occluded, an expression variant face from a single sample per class. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24(6), 748–763 (2002)
8. Martinez, A.M., Benavente, R.: The AR face database, CVC Technical Report # 24 (June 1998)
9. Smith, L.I.: A tutorial on Principal Components Analysis, February 26 (2002)
10. Bishop, C.M.: *Pattern recognition and machine learning*. Springer, Heidelberg (2006)
11. Vapnik, V.N.: *Statistical Learning Theory*. John Wiley & sons (1998)
12. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), Software available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
13. Hayes, M.H., Lim, J.S., Oppenheim, A.V.: Signal Reconstruction from Phase or Magnitude. *IEEE Trans. Acoust., Signal Processing ASSP-28*, 672–680 (1980)
14. Oppenheim, A.V., Lim, J.S.: The importance of phase in signals. *Proc. IEEE* 69(5), 529–541 (1981)

