

Diseño de Sistemas de Reconocimiento de Rostros

**Correa, Martín Sebastián
Chichizola, Franco**

Tabla de Contenidos

TABLA DE CONTENIDOS	1
PARTE I – INTRODUCCION	5
Presentación.....	7
Definición del Problema.....	8
Motivaciones y Expectativas.....	12
PARTE II – MARCO TEORICO – CONCEPTOS FUNDAMENTALES	13
Capítulo 1 – Procesamiento digital de Imágenes.....	15
1.1. Introducción al procesamiento digital de imágenes.....	15
1.1.1. Representación digital de imágenes.....	15
1.1.2. Etapas fundamentales del procesamiento de imágenes.....	15
1.2. Fundamentos de la imagen digital.....	16
1.2.1. Un modelo de imagen simple.....	16
1.2.2. Muestreo y cuantificación.....	17
1.2.3. Algunas relaciones básicas entre pixels.....	18
1.3. Mejora de la imagen.....	20
1.3.1. Fundamentos.....	21
1.3.2. Mejoras por procesamiento de punto.....	22
1.4. Filtrado espacial.....	26
1.4.1. Fundamentos.....	26
1.4.2. Filtros suavizantes.....	26
1.4.3. Filtros realzantes.....	28
1.5. Mejora en el dominio de la frecuencia.....	29
1.5.1. Filtrado pasa bajo.....	30
1.5.1.1. Filtro ideal.....	30
1.5.1.2. Filtro de Butterworth.....	30
1.5.2. Filtrado pasa alto.....	31
1.5.2.1. Filtro ideal.....	31
1.5.2.2. Filtro de Butterworth.....	31
1.5.3. Filtrado homomórfico.....	31
1.6. Procesamiento de imágenes en color.....	32
1.6.1. Fundamentos del color.....	33
1.6.2. Modelos de color.....	34
1.6.2.1. El modelo de color RGB.....	34
1.6.2.2. El modelo de color CMY.....	35
1.6.2.3. El modelo de color YIQ.....	35
1.6.2.4. El modelo de color HSI.....	35
1.6.3. Procesamiento de imágenes en falso color.....	36
1.6.3.1. División de la intensidad.....	37
1.6.3.2. Transformaciones de color del nivel de gris.....	37
1.6.3.3. Un método de filtrado.....	37
1.6.4. Procesamiento de imágenes en color real.....	38
1.7. Segmentación de imágenes.....	38
1.7.1. Detección de discontinuidades.....	39
1.7.2. Enlazado de bordes y detección de límites.....	41
1.7.3. Umbralización.....	41
1.7.4. Segmentación orientada a regiones.....	42

1.8. Representación y Descripción	43
1.8.1. Esquema de representación	43
1.8.1.1. Código de cadena	44
1.8.1.2. Aproximación poligonal	44
1.8.1.3. Firmas	45
1.8.1.4. Lados del contorno	45
1.8.1.5. El esqueleto de una región	46
1.8.2. Descriptores de contorno	46
1.8.2.1. Algunos descriptores simples	46
1.8.3. Descriptores de región	47
1.8.3.1. Algunos descriptores simples	47
1.8.3.2. Descriptores topológicos	47
1.8.3.3. Textura	47
1.8.4. Morfología	48
1.8.5. Descriptores relacionales	48
1.9. Reconocimiento e Interpretación	48
1.9.1. Elementos del análisis de imágenes	48
1.9.2. Patrones y clases de patrones	49
1.9.3. Métodos de decisión teórica	50
1.9.4. Métodos estructurales	52
1.9.4.1. Correspondencia de números de forma	52
1.9.4.2. Correspondencia de cadena	53
1.9.4.3. Métodos sintácticos	53
1.9.5. Interpretación	53
1.9.5.1. Fundamentos	53
1.9.5.2. Tipos de conocimiento	53
Capítulo 2 – Sistemas Biométricos.....	55
2.1. Modelo del proceso de identificación personal	55
2.2. Características de un indicador biométrico	55
2.3. Características de un sistema biométrico para identificación personal	56
2.4. Arquitectura de un sistema biométrico para identificación personal	56
2.5. Fase operacional de un sistema de identificación personal	58
2.6. Exactitud en la identificación: medidas de desempeño	59
2.7. Sistemas biométricos actuales	61
Capítulo 3 – Patrones de Reconocimiento.....	65
3.1. Introducción	65
3.2. Metodologías para el reconocimiento de patrones	65
3.2.1. Heurísticas	65
3.2.2. Matemáticas	65
3.2.3. Lingüísticas (sintácticas)	65
3.3. Etapas para el reconocimiento de patrones	66
3.3.1. Adquisición de datos	66
3.3.2. Preprocesamiento	66
3.3.3. Extracción de características	67
3.3.4. Reconocimiento	67
Capítulo 4 – Introducción a las Redes Neuronales.....	68
Capítulo 5 – Sistemas de Reconocimiento de Rostros.....	69
5.1. Introducción	69
5.2. Investigaciones sobre reconocimiento de rostros	71
5.2.1. Historia del reconocimiento de rostros	71
5.2.2. Actuales	72
5.3. Ventajas del Reconocimiento de Rostros	73

5.4. Aplicaciones	74
5.5. Tecnologías	75
5.5.1. Análisis de Características Locales (Local Feature Analysis, LFA)	77
5.5.2. Redes Neuronales	78
5.5.3. Modelo Markov (Hidden Markov Model, HMM)	80
5.5.4. Elastic Bunch Graph Matching	81
5.5.5. Eigenfaces	83
PARTE III – ASPECTOS DE IMPLEMENTACION.....	87
Capítulo 6 – Estudio Preliminar.....	89
6.1. Análisis del sistema de reconocimiento a implementar	89
6.2. Herramientas de Desarrollo	89
6.3. Características de la Entrada	90
6.3.1. Obtención de la muestra	90
6.3.2. Estudio de diversos formatos gráficos	91
Capítulo 7 – Desarrollo de la Aplicación.....	92
Capítulo 8 – Optimizaciones y Pruebas.....	97
CONCLUSIONES.....	109
ANEXO A – Base de datos de rostros ORL.....	113
ANEXO B – Manual del Usuario del sistema de reconocimiento de rostros.....	115
BIBLIOGRAFIA.....	119

PARTE I

INTRODUCCION

1. Presentación

El objetivo principal del proyecto es el de llevar a cabo una investigación básica y aplicada en el campo del procesamiento digital de imágenes. Dentro del procesamiento de imágenes, encontramos sistemas de reconocimiento y clasificación que tratan con áreas como seguridad y autenticación, automatización industrial, análisis de imágenes médicas, etc.

En particular, nos abocamos a observar técnicas para la detección e identificación de objetos complejos, mediante el reconocimiento basado en patrones ya conocidos, con lo cual se persigue simular la facilidad del ser humano para clasificar/seleccionar/descartar. El inconveniente de esto es que los sistemas solo proveen soluciones a problemas particulares, dependiendo del conocimiento previo con el cual se nutre en primer instancia.

Dentro de este marco, nos encontramos con el estudio de los sistemas de reconocimiento de rostros, los cuales pretenden simular (como se mencionó anteriormente en general), el proceso que realiza un ser humano al observar a una persona en la vida real, tal vez reconociéndola, resultándole familiar o simplemente determinando que no la conoce.

Se realiza un estudio de diferentes modelos para la detección y/o reconocimiento de rostros, tratando de no realizar una comparación, sino más bien una presentación teórica de estas técnicas. Se realiza un estudio de mayor énfasis de la técnica que más promete resultados tendientes a lo óptimo, de tal manera de reunir el conocimiento necesario, para poder realizar una implementación en un lenguaje de programación, y poder comprobar la eficiencia y eficacia de la misma, realizando un análisis de índole estadístico.

Asimismo, y como resultado del análisis exhaustivo de las distintas técnicas, se obtiene como valor agregado, el estudio de un patrón de diseño de los sistemas de reconocimiento de rostros en general, capturando tanto puntos en común a nivel microsistema (es decir, distintas técnicas utilizan mismas características), como a nivel macro (es decir, distintas técnicas utilizan características distintas pero persiguiendo estas objetivos en común).

2. Definición del Problema

2.1. Estado del Arte

El potencial esperable en la tecnología de reconocimiento de rostros es que gracias a ella, podremos dejar de usar llaves para entrar a edificios, llevar tarjetas de identificación, recordar contraseñas, entre otras cosas, tal como cita Ray Kurzweil en su libro *The Age of Spiritual Machines* (pag. 77,172,276), “realizando identificaciones seguras de personas usando técnicas de reconocimiento por patrones, aplicando en particular patrones de reconocimiento visuales y de voz, lograremos remplazar cerraduras y llaves en muchos casos”.

En 1997 se comenzó a utilizar el primer sistema de reconocimiento de rostros, por las personas en general, en máquinas de pago/depósito de cheques”. La empresa Miros introdujo esta tecnología bajo el nombre de TrueFace. Esta tecnología se implementó como subsistema de seguridad dentro del sistema llamado Mr. Payroll (un sistema compuesto por máquinas al mejor estilo cajero automático, abierto las 24-horas), verificando la identidad del usuario (reconociendo su rostro) que quería realizar una transacción dentro de Mr. Payroll, y aceptando dicha transacción si el usuario era quien decía ser.

2.2. Características del funcionamiento en general

El reconocimiento de rostros es una tarea relativamente fácil para los humanos, y un ejemplo claro de esto es el caso de los niños quienes rápidamente aprenden a reconocer el rostro de sus padres. Esto podría hacer pensar que enseñarle a una computadora como reconocer diferentes rostros es una tarea sencilla, pero desafortunadamente esto no es así.

Este problema que fue considerado desde las primeras etapas de visión por computadora, y ha sido estudiado más a fondo en los últimos años, gracias a los avances del poder computacional que han permitido implementar algoritmos más complejos utilizando diferentes técnicas.

En líneas generales, se puede ver como una aplicación de la Inteligencia Artificial, que trata con la habilidad de las computadoras para reconocer una imagen de un rostro dentro de una base de rostros ya conocidas.

Pese a que es un campo de reciente investigación, se ha sido desarrollado en profundidad, y tiene como resultado varias aplicaciones con interesante futuro, existiendo en la actualidad muchos modelos disponibles.

El uso de un sistema de reconocimiento de rostros, es muy interesante para una gran variedad de tareas. Puede ser utilizado, en lugares de trabajo o el hogar para seguridad, advirtiendo cuando se acercan personas sospechosas o desconocidas. Las empresas pueden encontrar útil este tipo de sistemas para determinar si una persona pertenece a la misma o es

un extraño. Puede ser muy útil, en identificación criminal, determinando rasgos del individuo a investigar. También se puede usar en las nuevas tecnologías emergentes como el video-mail. (Montgomery Blair)

El reconocimiento de rostros, se asocia al campo de la Inteligencia Artificial, debido a que intenta resolver un tipo problema en el que los seres humanos son más rápidos y precisos. También es un proceso que debe ser enseñado. La computadora debe ser capaz de tomar el conocimiento que tiene de como comparar un rostro con otro y generalizarlo para todos los rostros. Este es un proceso extremadamente dificultoso para aprender, y en el cual la teoría de reconocimiento de patrones juega un papel decisivo.

Debido a la dificultad de este proceso de aprendizaje, existen muchas limitaciones que se deben aplicar a los sistemas de este tipo. Primero, hay que tener en cuenta que las bases de rostros son (y deberían serlo para abarcar un gran número de rostros aprendidos) lo suficientemente grandes, por lo cual ocupan mucha memoria (no tan importante es la memoria secundaria, sobre todo en estas épocas, sino cuanta memoria principal ocupa ante el proceso de reconocimiento). Esto causa que a veces se relegate la precisión para ganar velocidad de cálculo.

Los rostros son en general muy similares entre sí. Cada uno tiene nariz, boca, ojos, etc. La computadora debe ser capaz de tomar la menor variación posible para que pueda distinguir dos rostros. Otras limitaciones incluyen la variación, tanto de poses como de expresiones en los rostros. Variando los ángulos desde donde se toma la fotografía (futuro rostro digitalizado), se puede llegar a hacer confundir a la computadora, cuando trata de revisar una imagen contra la misma imagen desde un ángulo diferente. Diferentes expresiones causan el mismo problema. Por ejemplo, un rostro feliz frente al mismo rostro triste, tienen muchas diferencias (boca hacia arriba vs. boca hacia abajo) pese a ser de la misma persona.

Otras limitaciones que encontramos son la iluminación y el fondo del rostro. La mayoría de los modelos necesitan imperiosamente que tanto la iluminación como el fondo de dos imágenes de rostros a comparar sean similares, para facilitar nuevamente cálculos, y obtener un rostro libre de “ruido” externo, pese a que esto no representa las situaciones del mundo real.

Como síntesis, imaginemos el funcionamiento de un sistema en la vida real, donde existe una computadora que debe reconocer rostros, para permitir o no el acceso a una oficina de trabajo.

Cuando una persona quiere acceder a la oficina, se registra ante la computadora como un usuario del sistema (cabe mencionar que lo podría hacer mediante voz). De una cámara de vídeo digital, se genera una imagen, en donde el sistema encuentra el rostro, y utilizando alguna técnica, se logran definir características propias del rostro.

Recordar que cada individuo al que se le permite el acceso posee una imagen previamente almacenada en una base de datos (de hecho cuando se agrega una nueva persona al sistema, deberían ser tomadas varias imágenes de la misma, para poder capturar con mayor precisión las características de su rostro).

Inmediatamente el sistema deberá comparar la imagen tomada de la cámara con la imagen asociada al usuario que quiere ingresar. Debería reconocer un rostro aun cuando el ángulo de visión de la cámara sea diferente, o si la persona tiene un nuevo corte de pelo, anteojos o diferentes expresiones en el rostro. Tampoco debería ser engañado con una foto o algún otro tipo de artilugio. Cuando la comparación se completa, el sistema debería tomar la acción apropiada, admitiendo a las personas autorizadas (abriendo las puertas, por ejemplo) y rechazando las personas no-autorizadas.

2.3. Problema a Resolver

La tarea de reconocimiento de rostros la podemos encontrar en lugares donde se necesite tener un acceso restringido a una cierta área, o en sistemas de identificación de personas como los que son utilizados por la policía.

De manera general, los sistemas de reconocimiento de rostros se pueden clasificar en dos categorías:

- 1.- Cuando no se necesita hacer un reconocimiento en tiempo real.
- 2.- Cuando se quiere identificar a una persona en tiempo real (por ejemplo, en un sistema de monitoreo de seguridad), o cuando se desea permitir el acceso únicamente a un grupo de personas.

En este proyecto nos enfocamos en los sistemas de la primera categoría, es decir, los sistemas que no se ejecutan en tiempo real.

El proyecto lo podemos describir de manera muy general como sigue: dada como entrada la imagen del rostro de una persona, el sistema será capaz de identificar el rostro, dando como resultado la identidad de dicha persona.

2.4. Objetivos a Cumplir

Objetivo General

El objetivo general de este proyecto, es el desarrollo de un sistema que sea capaz de reconocer rostros, utilizando como conjunto de entrenamiento y prueba de la eficiencia del sistema, distintas bases de datos de rostros (En especial contamos con la base de datos de rostros de ORL - Olivetti Research Lab -, la cual contiene 10 imágenes diferentes de 40 sujetos distintos).

Objetivos Específicos

1. Analizar la base de datos y determinar la necesidad de desarrollar una etapa de preprocesamiento sobre las imágenes.
2. Realizar la extracción de características de los rostros, para reducir la imagen en un número más pequeño de componentes, los cuales contengan la información discriminante del rostro.
3. Estudiar varios modelos de sistemas de reconocimiento de rostros, y escoger el más apropiado para el desarrollo del sistema.
4. Desarrollar la etapa de entrenamiento del sistema, usando una parte de la base de datos de rostros ORL como datos de entrenamiento.
5. Realizar el reconocimiento utilizando los rostros de la base de datos que no fueron utilizados durante el entrenamiento y algunos de los rostros que también se usaron

2.5 Sistemas Conocidos de Reconocimiento de Rostros

Varios investigadores a nivel mundial, han desarrollado varias técnicas de reconocimiento de rostros.

Un grupo de ellos (con referente en el Dr. Robert Frischholz [22]), aduce que el esfuerzo en el reconocimiento de rostros radica en un paso previo y fundamental al mismo, que consiste en la detección de uno o varios rostros humanos, dentro de escenarios arbitrarios. De hecho aseguran que es el proceso más importante y complejo, ya que cuando los rostros se han localizado de manera exacta dentro de las imágenes, la tarea de reconocimiento se ha simplificado lo suficiente, como para llegar a ser algo prácticamente trivial.

No obstante, hay que destacar que los investigadores que profundizan sobre el tema del reconocimiento de rostros en sí, deben lidiar con los problemas propios de la complejidad que posee un rostro. Un mismo rostro puede variar en una inmensurable cantidad de expresiones, así como puede sufrir cambios por factores externos, como puede ser luminosidad, clima, etc.,

más aun considerando que el rostro termina siendo una imagen digitalizada mediante un medio que suele ser una cámara fotográfica o una videocámara.

En una primera línea de investigación, podemos citar a un grupo de personas muy interrelacionadas con especialistas en psicología, que lo que intentan reproducir es el comportamiento del cerebro humano en cuanto a como va conociendo y posteriormente reconociendo otras personas. Es por ello que sus trabajos se fundamentan muy fuertemente en la realización de sistemas utilizando redes neuronales **[26]** (se destacan H.Adbi, O'Toole).

También tenemos personas dedicadas al estudio del reconocimiento a nivel computacional, es decir, utilizar las herramientas matemáticas/informáticas con los cuales se cuenta y analizar la imagen digitalizada de un rostro de esta manera..

Cabe destacar el aporte de personas vinculadas al ambiente neuro/psicológico **[27]**, que generan modelos abstractos de un posible comportamiento de un reconocimiento de rostros sin ahondar en detalles (que no son de su incumbencia) computacionales (entre los más destacados encontramos a Burton y Bruce). El resultado de sus investigaciones muchas veces es tenido en cuenta como punto de partida para el estudio a nivel computacional del reconocimiento de rostros.

Motivaciones y Expectativas

El desarrollo de esta aplicación trae aparejada una serie de aspectos altamente positivos en el área de la investigación.

Es necesario realizar un estudio profundo del procesamiento de imágenes y de técnicas de reconocimiento para particularizarlos en el problema a resolver, logrando la interiorización del propósito e implementación de la mayor parte de las metodologías y algoritmos relacionados con estos tipos de procesamiento dentro del mundo de la informática.

Se pretende que este estudio no se limite a una mera investigación teórica, sino por el contrario, que el resultado del esfuerzo en la investigación, se vea reflejado en un producto (software) terminado y que lleve a niveles satisfactorios en cuanto a resultados esperados.

Por otro lado, la solución del problema no resulta sencilla a priori, debiéndose sortear diversos inconvenientes no predeterminados.

Además, se puede decir que este tipo de desarrollo obliga a la implementación y prueba de variados algoritmos, que aunque no sean de utilidad en definitiva para la solución, aportan conocimientos y experiencias extras a las que se obtengan del producto final.

PARTE II

MARCO TEORICO

Conceptos Fundamentales

Capítulo 1 – PROCESAMIENTO DIGITAL DE IMÁGENES

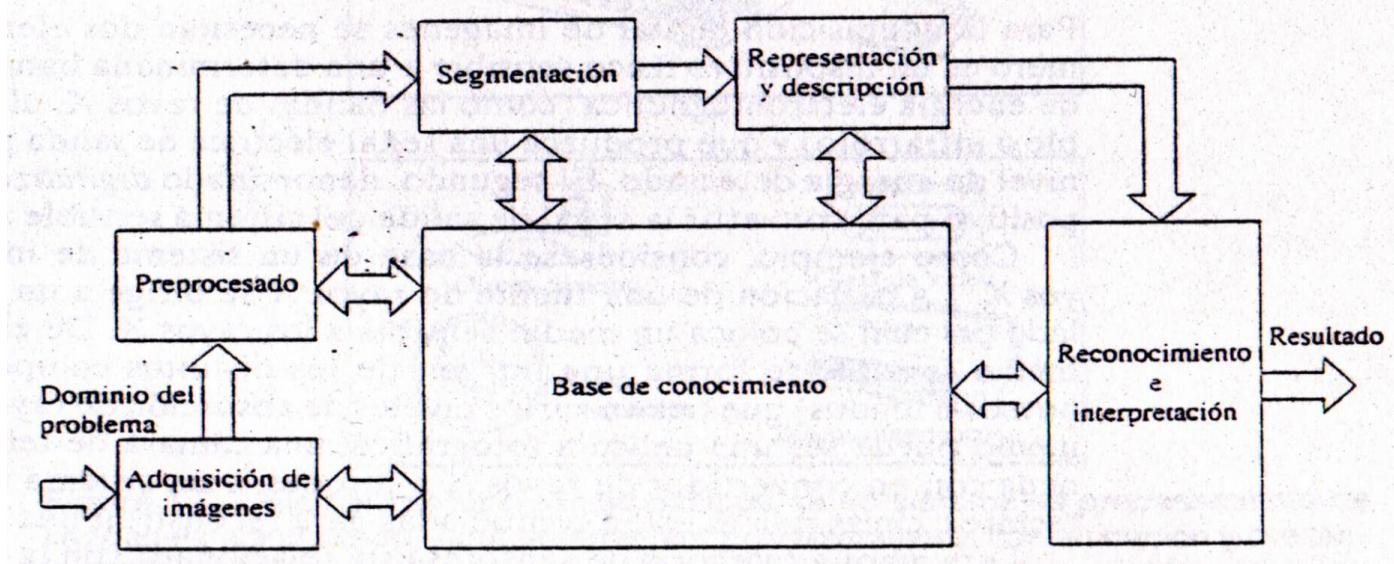
1.1. Introducción al procesamiento digital de imágenes

1.1.1. Representación digital de imágenes

El término *imagen monocroma* o simplemente *imagen*, se refiere a una función bidimensional de intensidad de luz $f(x,y)$ donde x e y representan las coordenadas espaciales, y el valor de f en un punto (x,y) es proporcional al brillo (o *nivel de gris*) de la imagen en ese punto.

Una *imagen digital* es una imagen $f(x,y)$ que se ha discretizado tanto en las coordenadas espaciales como en el brillo. Una imagen digital puede considerarse como una matriz cuyos índices de fila y columna identifican un punto de la imagen, y el valor del correspondiente elemento de la matriz indica el nivel de gris en ese punto. Los elementos de una distribución digital de este tipo se denomina *elemento de la imagen*, o más comúnmente *pixels* o *pels*.

1.1.2. Etapas fundamentales del procesamiento de imágenes



Las etapas fundamentales que se deben realizar para el procesamiento de imágenes son las siguientes:

Adquisición de la imagen: es la primer etapa del proceso, y se refiere a la adquisición de una imagen digital. Para lo cual se necesita un sensor de imágenes y la posibilidad de digitalizar la señal producida por el sensor. Si la salida del sensor (por ejemplo una cámara) no es digital, puede emplearse un convertidor analógico-digital para digitalizarla.

Preprocesamiento: es la etapa que le sigue a la adquisición, y su función básica es la de mejorar la imagen de forma que se aumenten las posibilidades de éxito en los procesos posteriores.

Segmentación: consiste en partir una imagen de entrada en sus partes u objetos constituyentes. En general, la segmentación autónoma es una de las labores más difíciles del tratamiento digital de imágenes.

Representación y descripción: de la etapa anterior, se obtienen los datos de pixel en bruto, por lo cual, en esta nueva etapa, se deben convertir los datos de alguna manera adecuada para su procesamiento por computadora. Como primer paso, se debe decidir la *representación* de los datos: como un contorno, como una región completa, o bien una combinación de ambas.

El segundo paso para lograr dicha transformación, es especificar un método para describir los datos de forma que se resalten los rasgos de interés. La *descripción*, también llamada *criterio de selección*, consiste en extraer rasgos con alguna información cuantitativa de interés o que sean fundamentales para diferenciar una clase de objetos de otra.

Reconocimiento e interpretación: esta es la última etapa. El *reconocimiento* es el proceso que asigna una etiqueta a un objeto basándose en la información proporcionada por sus descriptores. La *interpretación* implica asignar significado a un conjunto de objetos reconocidos.

Se debe tener en cuenta que siempre se tiene algún tipo de información sobre las imágenes que se deben tratar. Estos datos, se encuentran reunidos en la llamada *base de conocimiento*, cuya complejidad y cantidad de información varía según la aplicación. Dicha base, no solo interviene en cada una de las etapas a realizar, sino que también se utiliza en la interacción entre estas.

No siempre se utilizan todas estas etapas, al igual que no siempre se tiene una base de conocimiento. Por ejemplo, el mejorar una imagen para interpretaciones visuales humanas rara vez pasa más allá de la etapa de preprocesamiento.

1.2. Fundamentos de la imagen digital

1.2.1. Un modelo de imagen simple

El término *imagen* se refiere a una función bidimensional de la luz y la intensidad, a la que indicamos por $f(x,y)$, donde el valor o amplitud de f en las coordenadas espaciales (x,y) da la intensidad (iluminación) de la imagen en este punto. Puesto que la luz es una forma de energía, $f(x,y)$ debe ser estrictamente mayor que cero, y finito, es decir que: $0 < f(x,y) < \infty$

Las imágenes cotidianas provienen, normalmente, de la luz reflejada por los objetos. Por lo tanto, la naturaleza básica de $f(x,y)$ puede caracterizarse por dos componentes:

1. La cantidad de luz incidente sobre la escena contemplada (componente de *iluminación*). Se indica por $i(x,y)$, y su rango de valores es: $0 < i(x,y) < \infty$.
2. La cantidad de luz reflejada por los objetos de la escena (componente de *reflectancia*). Se indica por $r(x,y)$, y su rango de valores: $0 \leq r(x,y) \leq 1$, donde el 0 es la absorción total, y uno la reflexión total.

Por lo que $f(x,y)$ puede verse como el producto de ambas componentes:

$$f(x,y) = i(x,y) \cdot r(x,y)$$

La intensidad en una imagen monocromática, es denominada *nivel de gris (l)*, y tendrá un límite inferior mayor que cero (L_{max}), y un límite superior finito (L_{min}). Al intervalo entre estos dos límites se lo llama *escala de grises*. Una práctica habitual consiste en desplazar este intervalo hasta que el límite inferior sea cero ($L_{min} = 0$), quedando el intervalo $[0,L]$, donde el valor 0 se considera como negro, y L como blanco. Todos los valores intermedios corresponden a tonos de gris que van del negro al blanco.

1.2.2. Muestreo y cuantificación

A) Muestreo uniforme y cuantificación

Para ser utilizado para el procesamiento por medio de computadora, una función de imagen $f(x,y)$ debe ser digitalizada tanto espacialmente como en su amplitud. La digitalización de las coordenadas espaciales (x,y) se denomina *muestreo de la imagen*, y la digitalización de la amplitud se conoce como *cuantificación del nivel de gris*.

Una imagen continua $f(x,y)$ se describe de forma aproximada por una serie de muestras igualmente espaciadas, organizadas de forma de una matriz de $N \times M$, representando una imagen digital, donde cada elemento es una cantidad discreta y se lo conoce como *elemento de la imagen, pixel o pel*.

El proceso de digitalización implica tomar ciertas decisiones sobre el valor de M, N, y la cantidad de niveles de gris permitidos (G). Generalmente se le dan valores que sean potencia de dos. Es decir:

- $N = 2^n$
- $M = 2^k$
- $G = 2^m$

Por lo tanto, el número de bits necesarios para almacenar una imagen digitalizada es:

$$b = M \times N \times m$$

Cuanto mayor sean los 3 parámetros (M, N, G), mayor será la resolución o grado de detalle discernible, y por lo tanto, más se aproximarán la imagen digitalizada a la original. Pero también será mayor el almacenamiento necesario y la necesidad de procesamiento.

B) Muestreo no uniforme y cuantificación

Para un valor fijo de la resolución espacial, la apariencia de una imagen puede mejorar en muchos casos empleando un esquema adaptativo en el que el proceso de muestreo dependa de las características de la imagen. Por ejemplo, se puede necesitar un muestreo fino en las proximidades de las transiciones bruscas en los niveles de gris, mientras que se puede aplicar un muestreo tosco en las regiones relativamente suaves.

Una técnica alternativa para distribuir los niveles de gris consiste en calcular la frecuencia de aparición de todos los niveles permitidos. Si, en un cierto rango, algunos niveles de gris aparecen frecuentemente, mientras que otros aparecen raramente, los niveles de cuantificación deben estar finamente espaciadas en este rango y ampliamente espaciados fuera de él. Este método se denomina a menudo *cuantificación en banda*.

1.2.3. Algunas relaciones básicas entre pixels

A) Vecinos de un pixel

Un pixel p con coordenadas (x,y) tiene:

- 4 vecinos verticales y horizontales, cuyas coordenadas son
 $(x,y-1); (x,y+1); (x-1,y); (x+1,y)$
Son llamados 4-vecinos de p , y se representan por $N_4(p)$.
- 4 vecinos diagonales, cuyas coordenadas son
 $(x-1,y-1); (x-1,y+1); (x+1,y-1); (x+1,y+1)$
Son representados por $N_D(p)$

Juntando ambos grupos de pixels, obtenemos los 8-vecinos de p , representados por $N_8(p)$.

B) Conectividad

Para que dos pixels estén conectados, deben cumplir con:

- Ser adyacentes en algún sentido (4 u 8-vecinos).
- Sus niveles de gris cumplen un criterio de similitud especificado (como ser iguales, o pertenecer a un determinado rango).

Supongamos que tenemos un conjunto V de valores de nivel de gris, que determina el criterio de similitud (si p y q tienen valores dentro de V cumplen dicho criterio). Se consideran tres tipos de conectividad:

- I) 4-conectividad: p y q con valores dentro de V , y $q \in N_4(p)$.
- II) 8-conectividad: p y q con valores dentro de V , y $q \in N_8(p)$.
- III) m-conectividad (conectividad mixta): p y q con valores dentro de V . Y $(q \in N_4(p))$, o bien $(q \in N_D(p)$ y además $N_4(p) \cap N_4(q)$ con valores dentro de V es vacío).

Es una modificación de la II, que se utiliza para eliminar múltiples caminos de conexión.

Un pixel p es *adyacente* a q si están conectados. Se puede definir 4-, 8-, m-adyacencia, dependiendo de la conectividad usada. Dos subconjuntos de la imagen S_1 , y S_2 , son adyacentes si algún pixel de S_1 es adyacente a algún pixel de S_2 .

Un *camino* desde un pixel p hasta uno q es una sucesión de distintos pixels p_0, \dots, p_n , donde $p_0 = p$ y $p_n = q$, de tal manera que p_i y p_{i+1} son adyacentes (para $i=0..n-1$), siendo n la longitud del camino. Pudiendo definir 4-, 8-, o m-camino, dependiendo del tipo de adyacencia usado.

Si p y q pertenecen a S (subconjunto de la imagen), se dice que p está *conectado* con q dentro de S si existe un camino entre p y q donde todos los pixels pertenecen a S . Para cualquier p dentro de S , el conjunto de pixels de S conectados a p se denomina *componente conexa de S* .

Por medio del etiquetado de cada una de estas componentes conexas, se divide a S en subconjuntos disjuntos, donde cada uno es una componente conexa distinta.

C) **Etiquetado de componentes conexos**

Es un algoritmo que consiste en darle a cada componente conexa de una imagen una etiqueta (un valor asociado) diferente.

D) **Medidas de distancia**

Para los pixel p,q y z de coordenadas (x,y), (s,t) y (u,v) respectivamente, D es una función de distancia o una métrica si:

- A) $D(p,q) \geq 0$ $\{D(p,q) = 0 \text{ si y solo si } p = q\}$.
- B) $D(p,q) = D(q,p)$.
- C) $D(p,z) \leq D(p,q) + D(q,z)$.

- La distancia *euclídea* entre p y q está definido por:

$$D_e(p,q) = [(x-s)^2 + (y-t)^2]^{1/2}$$

Los pixels que están a una distancia menor o igual que un algún valor r de (x,y), son los puntos contenidos en un círculo de radio r y origen (x,y).

- La distancia D_4 (denominada distancia *city-block*) entre dos puntos p y q, se define como:

$$D_4(p,q) = |x-s| + |y-t|$$

Los puntos que están a una distancia de (x,y) menor o igual a r, forman un rombo centrado en (x,y).

- La distancia D_8 (llamada distancia de *tablero de ajedrez*) entre p y q se define como:

$$D_8(p,q) = \max(|x-s|, |y-t|)$$

Los pixels a una distancia de (x,y) menor o igual a r forman un cuadrado centrado en (x,y).

1.3. Mejora de la imagen

El principal objetivo de las técnicas de mejora es procesar una imagen de forma que resulte más adecuada que la original para una aplicación específica.

Estas técnicas pueden ser de tres categorías básicas:

- Métodos en el dominio espacial (se refiere al propio plano de la imagen, por lo que estas técnicas se basan en la manipulación directa de los pixels de la imagen).
- Métodos en el dominio de la frecuencia (se basan en la modificación de la transformada de Fourier de la imagen).
- Combinación de métodos de ambas categorías.

En cualquiera de los casos, no existe un método que sea mejor que otro, debido a que depende de la aplicación específica para la cual se necesite. En cada situación, el mejor método será el que de los mejores resultados para esa aplicación específica.

1.3.1. Fundamentos

1.3.1.1. Métodos en el dominio espacial

Son procedimientos que operan directamente sobre los pixels de la imagen. Las funciones de procesamiento de la imagen en el dominio espacial pueden expresarse como: $g(x,y) = T[f(x,y)]$, donde $f(x,y)$ es la imagen de entrada, $g(x,y)$ es la imagen procesada, y T es un operador que actúa sobre f , definido en algún entorno de (x,y) .

En general, para definir un entorno alrededor de (x,y) , se emplea un área de subimagen cuadrada o rectangular centrada en (x,y) . El centro de la subimagen se mueve pixel a pixel aplicando el operador T en cada posición (x,y) para obtener g . Se pueden usar otros tipos de entornos, como aproximaciones a círculos, pero los cuadrados o rectángulos son los más usados por ser más fáciles de implementar.

La forma más simple de T corresponde a un entorno de 1×1 . En este caso, g depende solo del valor de f en el punto (x,y) , y T se convierte en una función de transformada del nivel de gris. Como la mejora de la imagen en cada punto depende sólo del nivel de gris en ese punto, las técnicas de esta categoría se conocen como *procesamiento de punto*.

Independientemente de la aplicación específica, la idea general consiste en determinar g en un punto (x,y) a partir de los valores de f en un entorno predefinido de (x,y) . En general, se utilizan las llamadas *máscaras* (o ventanas o filtros). Una máscara es una pequeña distribución bidimensional (de por ejemplo 3×3), en la que los valores de los coeficientes determinan la naturaleza del proceso. Estas técnicas se conocen como *procesamiento por máscara* o *filtrado*.

1.3.1.2. Métodos en el dominio de la frecuencia

La base de las técnicas en el dominio de la frecuencia es el *teorema de convolución*. Sea $g(x,y)$ una imagen formada por la convolución de una imagen $f(x,y)$ y un *operador lineal invariante de posición* $h(x,y)$ (es aquel cuyo resultado depende sólo del valor de $f(x,y)$ en un punto de la imagen, y no de la posición del punto), es decir: $g(x,y) = h(x,y) * f(x,y)$.

Entonces, por el teorema de convolución: $G(u,v) = H(u,v) \cdot F(u,v)$, donde G, H , y F son las transformadas de Fourier de g, h , y f respectivamente. En una aplicación típica de mejora de la imagen, $f(x,y)$ es conocida, y el objetivo, después de calcular $F(u,v)$, es seleccionar $H(u,v)$ de forma que la imagen deseada $g(x,y)$ (transformada de Fourier inversa de $H(u,v) \cdot F(u,v)$) presente resaltada alguna características de $f(x,y)$, como por ejemplo los bordes.

1.3.2. Mejoras por procesamiento de punto

Son las técnicas de mejora de imagen que se basan sólo en la intensidad de pixels individuales. Se indicará con r y s a la intensidad de los pixels antes y después del procesamiento, respectivamente.

1.3.2.1. Algunas transformaciones de intensidad simples

1.3.2.1.1. Negativos de imágenes: El negativo de una imagen digital se obtiene empleando la función de transformación $s = T(r)$, donde $T(r) = (-1)r + (L-1)$, siendo L el número de niveles de gris.

Lo que se trata de hacer es invertir el orden de blanco a negro, de forma que la intensidad de la imagen de salida disminuye conforme la intensidad de entrada aumente.

1.3.2.1.2 Aumento de contraste: La idea en estas técnicas es incrementar el rango dinámico de los niveles de gris de la imagen. Para esto se utiliza una función de transformación que varía de acuerdo a lo que se quiera hacer. Por ejemplo, $s = T(r)$:

- Si $T(r) = r$, entonces no se producen cambios en los niveles de gris.
- Si $T(r) = 0$ (para $r < r_1$) y $T(r) = L-1$ (para $r > r_1$), la transformación se convierte en una función umbral que crea una imagen binaria, teniendo un problema cuando $r = r_1$, ya que tomaría todos los valores, desde 0 hasta $L-1$.
- En general $T(r)$ es monótona creciente.

1.3.2.1.3. Compresión del rango dinámico: A veces el rango dinámico de una imagen es muy grande como para ser utilizado, por ejemplo, por exceder la capacidad del dispositivo de presentación, en cuyo caso sólo las partes más brillantes de la imagen aparecerán en la pantalla. Por lo tanto, conviene achicar dicho rango.

1.3.2.1.4. Fraccionamiento del nivel de gris: A menudo se desea destacar un rango específico del nivel de gris de una imagen. Hay varias formas de hacer el fraccionamiento, pero las dos básicas son las siguientes:

- Adjudicar un valor alto a todos aquellos niveles de gris del rango de interés, y un valor bajo al resto. Obteniendo así un color claro para los niveles de gris de interés, y oscuro para el resto.
- Aumentar la intensidad para los niveles de gris deseados, preservando al mismo tiempo el fondo y las tonalidades de gris de la imagen. Es decir, que aclara aquellas partes que tienen un nivel de gris dentro del rango deseado, conservando igual al resto.

1.3.2.1.5. Fraccionamiento de los planos de bits: En lugar de destacar cierto rango de intensidad, puede desearse destacar la contribución que realiza a la imagen ciertos bits específicos. Suponiendo que tenemos una imagen cuyos pixels están representados por un byte cada uno($b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0$), la imagen se puede ver como 8 planos, donde el plano i está formado por el bit b_i de cada pixel, para $i = 0, 1, \dots, 7$. Cuanto mayor sea i , el plano tendrá datos de mayor significativo visual, mientras que los planos con i menor, contribuyen a los detalles más finos de la imagen. Entrando entre los primeros los planos de 3 a 7, y entre los segundos los planos de 0 a 2.

1.3.2.2. Procesamiento de histogramas

El histograma de una imagen digital con niveles de gris en el rango $[0, L-1]$ es una función discreta

$$p(r_k) = n_k / n$$

donde r_k es el k -esimo nivel de gris, n_k es el número de pixels de la imagen con el ese nivel de gris, y n es el número total de pixels de la imagen, y $k = 0, 1, \dots, L-1$.

Se podría representar gráficamente los $p(r_k)$ para todos los valores de $k = 0, 1, \dots, L-1$, dando así una descripción global de la apariencia de la imagen.

Hay métodos para manipular los histogramas de una forma coherente y significativa. Ellos son:

1.3.2.2.1. Ecuación de histogramas: Sea r una variable que representa los niveles de gris de la imagen a mejorar. Supongamos que los pixels son cantidades continuas en un intervalo $[0, 1]$, con $r = 0$ representando al negro, y $r = 1$ representando al blanco.

Para cada r del intervalo $[0, 1]$, obtenemos un nivel s , por medio de la siguiente transformación: $s = T(r)$, donde se verifica:

- $T(r)$ es de valor único y monótonamente creciente en el intervalo $[0, 1]$.
- $T(r)$ toma valores en el intervalo $[0, 1]$, para r perteneciente al mismo intervalo.

La función de transformación inversa de s a r se indica por: $r = T^{-1}(s)$ para s en $[0, 1]$, donde $T^{-1}(s)$ también satisface las condiciones (a) y (b) con respecto a la variable s .

Siendo $p_r(r)$ la función de densidad de probabilidad de que aparezca el nivel de gris r en la imagen, obtenemos que:

$$s = T(r) = \int_0^r p_r(w) dw \quad 0 \leq r \leq 1$$

siendo el término de la derecha la *función de distribución acumulativa* de r (CDF). Cuando se emplea una función de transformación igual a la CDF, se produce una imagen cuyos niveles de gris tienen una densidad uniforme, lo cual producirá un incremento del rango dinámico de los pixels.

Para poder ser aplicado en el procesamiento digital de imágenes, se debe expresar en forma discreta. Para los valores de gris que constituyen los valores discretos, se tienen las probabilidades:

$$p(r_k) = n_k / n \quad \text{donde} \quad 0 \leq r_k \leq 1 \quad y \quad k = 0, 1, \dots, L-1.$$

Una representación gráfica de $p_r(r_k)$ en función de r_k se denomina un *histograma*, y la técnica empleada para obtener un histograma uniforme se conoce como *ecualización del histograma* o *linealización del histograma*.

La forma discreta de $T(r)$ está dada por:

$$s_k = T(r_k) = \sum_{j=0}^k (n_j / n) = \sum_{j=0}^k p_r(r_j) \quad 0 \leq r_k \leq 1 \quad y \quad k = 0, 1, \dots, L-1.$$

La transformada inversa se indica por: $r_k = T^{-1}(s_k)$ para s_k en $[0, 1]$, donde $T(r_k)$ y $T^{-1}(s_k)$ cumplen las condiciones (a) y (b).

1.3.2.2 Especificación del histograma: Algunas veces es deseable tener la posibilidad de especificar determinados perfiles del histograma, capaces de destacar determinados rangos de gris de la imagen. Supongamos que $p_r(r)$ y $p_z(z)$ son las funciones de probabilidad original y deseada, respectivamente. Supongamos que la ecualización del histograma se acaba de aplicar a la imagen original, obteniendo:

$$s = T(r) = \int_0^r p_r(w) \, dw$$

Si la imagen deseada estuviese disponible, sus niveles de gris también se podrían ecualizar empleando la función de transformación:

$$v = G(z) = \int_0^z p_z(w) \, dw$$

El proceso inverso $z = G^{-1}(v)$, proporcionaría los niveles z de la imagen deseada, que es lo que debemos hallar. Sin embargo, $p_s(s)$ y $p_v(v)$ corresponden a densidades uniformes idénticas. Así, utilizaremos los niveles uniformes s obtenidos de la imagen original en vez de los v : $z = G^{-1}(s)$. Admitiendo que $G^{-1}(s)$ sea de valor único, el procedimiento se puede resumir en tres pasos:

1. Ecuallizar los niveles de la imagen original.
2. Especificar la función de densidad deseada y obtener la función de transformación $G(z)$.
3. Aplicar la función de transformación inversa $z = G^{-1}(s)$, a los niveles obtenidos en el paso 1.

El mayor problema en el método de especificación de histograma es el de construir un histograma significativo. Existen 2 posibles soluciones:

1. Especificar una función de probabilidad particular (como una densidad gaussiana) y luego formar un histograma digitalizando la función dada.
2. Especificar la forma del histograma por medio de un dispositivo gráfico, cuya salida se introduzca en el procesador ejecutando el algoritmo de especificación del histograma.

1.3.2.2.3. Mejora local: Los dos métodos anteriores actúan sobre toda la imagen. Pero hay veces que se desea mejorar sólo una parte de ella, y que al emplear los métodos anteriores no se produce mejora sobre esa parte por ser, por ejemplo, muy pequeña. La solución consiste en planear funciones de transformación en la vecindad de cada pixel de la imagen.

El procesamiento consiste en definir un entorno rectangular o cuadrangular, y mover su centro pixel a pixel. Para cada ubicación, se calcula el histograma de los puntos del entorno y se obtiene la ecualización o bien la función de transformación de la especificación del histograma. Finalmente se emplea esta función para dar el nivel de gris al pixel del centro del entorno. Luego se mueve el centro del entorno a un pixel adyacente y se repite el proceso. Como solo cambia una fila o una columna del entorno durante la translación de pixel a pixel, se puede actualizar el histograma obtenido para la posición anterior sin necesidad de volverlo a hacer en su totalidad, con lo cual reduce el tiempo de procesamiento.

En lugar de emplear histogramas, la mejora local puede basarse en otras propiedades de las intensidades de los pixels de un entorno. El valor medio y la varianza de la intensidad son dos propiedades utilizadas para esto.

Una transformación local basada en estos conceptos, aplica la intensidad de una imagen de entrada $f(x,y)$ en una nueva imagen $g(x,y)$ realizando la siguiente transformación para cada pixel (x,y) :

$$G(x,y) = A(x,y) \cdot [f(x,y) - m(x,y)] + m(x,y)$$

donde $A(x,y) = K \cdot (M / \sigma(x,y))$ $0 < K < 1$, $m(x,y)$ y $\sigma(x,y)$ representan la media y la varianza de los niveles de gris calculadas en un entorno centrado en (x,y) . M es la media global de $f(x,y)$ y K es una constante.

1.3.2.3. Substracción de imágenes

La diferencia entre dos imágenes $f(x,y)$ y $g(x,y)$, expresada de la forma:

$$g(x,y) = f(x,y) - h(x,y)$$

se obtiene calculando la diferencia entre todos los pares de pixels correspondientes de f y h . La substracción de imágenes tiene numerosas e importantes aplicaciones en la segmentación y en la mejora.

1.3.2.4. Promediado de la imagen

Consideremos una imagen con ruido $g(x,y)$ formado por la adición de una función de ruido $\eta(x,y)$ a una función original $f(x,y)$; es decir:

$$g(x,y) = f(x,y) + \eta(x,y)$$

donde se ha realizado la hipótesis de que en cada par de coordenadas (x,y) el ruido es una función sin correlación y tiene un valor medio cero. El objetivo es reducir los efectos del ruido a base de sumar un conjunto de imágenes $\{g_i(x,y)\}$.

Si el ruido verifica esas restricciones, entonces:

$$g(x,y) = (1/M) \sum_{i=1}^M g_i(x,y)$$

es el promedio de M imágenes diferentes con ruido. Conforme aumente la cantidad de imágenes promediadas (M), más se asemeja el promedio a $f(x,y)$.

1.4. Filtrado espacial

1.4.1. Fundamentos

El empleo de máscaras espaciales se llama *filtrado espacial*, y a las máscaras se las denomina *filtros espaciales*.

Los filtros denominados de *pasa bajo* atenúan o eliminan las componentes de alta frecuencia en el dominio de Fourier, a la vez que dejan inalteradas las bajas frecuencias. Las componentes de alta frecuencia caracterizan los bordes y los restantes detalles muy marcados de la imagen, por lo que los filtros de pasa bajo hacen más borrosa la imagen.

Los filtros de *pasa alto* atenúan o eliminan las componentes de baja frecuencia, que son los responsables de las características lentamente variables de la imagen, por lo que este tipo de filtro agudiza los bordes y los restantes detalles finos.

Un tercer tipo de filtrado, el *filtrado de pasa banda*, elimina una región elegida de frecuencias intermedias. Estos son más utilizados para la restauración de imágenes que para la mejora de ellas.

Estos tres tipos de filtros son filtros espaciales lineales, y su funcionamiento es el siguiente:

- La máscara tendrá distintos coeficientes en cada una de sus posiciones.
- Al ubicar la máscara en una zona de la imagen, el valor del pixel central de esa zona estará dado por la suma de cada uno de los productos del coeficiente de la máscara por la intensidad del pixel correspondiente en dicha zona.

Por ejemplo, si la máscara de 3x3 es:

W	w	w
1	2	3
W	w	w
4	5	6
W	w	w
7	8	9

Llamando a los pixel bajo la máscara z_1, \dots, z_9 , entonces el valor de z_5 quedará:

$$R = \sum_{i=0}^9 w_i z_i$$

- Luego se corre la máscara hacia el siguiente pixel y se repite la operación hasta haber recorrido toda la imagen.

Los nuevos valores se pueden ir guardando en la misma imagen, o se pueden poner en otra imagen, para evitar así que los nuevos valores afecten el procesamiento de los pixels siguientes.

Los filtros espaciales no lineales, se trabajan de una manera similar a la anterior, pero sin usar coeficientes en la máscara, sino que operan sólo sobre los valores de los pixels, como por ejemplo, calculando la mediana, obteniendo el máximo o el mínimo, u otros métodos.

1.4.2. Filtros suavizantes

Los filtros suavizantes se usan para que la imagen aparezca borrosa y para reducir el ruido. Lo primero es útil en algunas etapas de preprocessamiento, como la eliminación de los pequeños detalles de una imagen, y el relleno de pequeños espacios entre líneas o curvas. La reducción del ruido se puede realizar mediante el aumento de la borrosidad con un filtro

lineal, o también con un filtro no lineal.

1.4.2.1. Filtrado espacial de pasa baja: El filtro debe tener todos sus coeficientes positivos, y la construcción más simple sería aquella en la que todos los coeficientes fueran 1. Lo cual llevaría a que la respuesta R fuera la suma de los niveles de gris de todos los pixels bajo la máscara, lo que puede provocar que R quede fuera del rango válido de niveles de gris. Para evitar esto, se divide R por la cantidad de pixels de la máscara, o bien dividir cada coeficiente de la máscara por esa cantidad. De esta manera, R será simplemente el promedio de todos los pixels en el área bajo la máscara, por lo que a menudo al uso de este tipo de máscara se las llama *promediado en el entorno*.

1.4.2.2. Filtrado por la mediana: Cuando el objetivo es más la reducción del ruido que el difuminado, el empleo de los filtros de mediana es más útil que el anterior. Lo que se hace es obtener para cada pixel la mediana de los niveles de gris en un entorno de este pixel, y reemplazando su valor por esa mediana. Estos filtros son no lineales.

1.4.3. Filtros realzantes

El objetivo principal del realce es el de destacar los detalles finos de una imagen o intensificar detalles que han sido difuminados.

1.4.3.1. Filtrado espacial de pasa alto básico: El filtro o máscara debe tener coeficientes positivos cerca de su centro y negativos en la periferia. Por ejemplo, para una máscara de 3×3 se podrían tener como coeficientes a (-1) para los ocho puntos periféricos, y (8) para el punto central, multiplicado todo por (1/9). En algunos casos, podrían dar resultados negativos, por lo cual se debería realizar un desplazamiento para que todos los valores entren en el rango [0, L-1].

1.4.3.2. Filtrado high-boost: Una imagen filtrada con pasa alto puede ser calculada como la diferencia entre la imagen original y una versión de ella pasada por un filtro de pasa bajo, es decir: Pasa Alto = Original - Pasa Bajo.

Para un filtro high-boost o de énfasis de las frecuencias altas se hace:

$$\text{High-boost} = (A) (\text{Original}) - \text{Pasa Bajo} = (A-1)(\text{Original}) + \text{Pasa Alto}$$

donde $A \geq 1$ es un *factor de amplificación*.

Si $A = 1$ entonces da el resultado de un filtro pasa alto normal. Cuando $A > 1$, parte de la imagen original se añade al resultado del filtro pasa alto. Por lo que el resultado se parece a la imagen original con un grado relativo de mejora de los bordes, dependiendo del valor de A .

Para la implementación se usa una máscara donde el término central será $W = cA-1$, donde c será la cantidad de pixels de la máscara, y el resto de los coeficientes será -1, todo multiplicado por $(1/c)$ (*factor de normalización*).

1.4.3.3. Filtros diferenciales: Así como la integración o promediado tiende a difuminar la imagen, es de esperar que la diferenciación haga lo contrario, es decir aumente la nitidez de la imagen. El método más común de diferenciación es el *Gradiente*.

Para una función $f(x,y)$, el gradiente de f en el punto (x,y) se define como el vector:

$$\nabla f = \begin{vmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{vmatrix}$$

El módulo de este vector:

$$\nabla f = \text{mag}(\nabla f) = [(\partial f / \partial x)^2 + (\partial f / \partial y)^2]^{1/2}$$

es la base de las varias aproximaciones a la diferenciación de la imagen.

Considerando una imagen de 3×3 con valores z_1, \dots, z_9 , hay diferentes formas de aproximar ∇f alrededor del punto z_5 :

- La más simple consiste en emplear la diferencia $(z_5 - z_8)$ en la dirección x , y la $(z_5 - z_6)$ en la dirección y , logrando así: $\nabla f \approx [(z_5 - z_8)^2 + (z_5 - z_6)^2]^{1/2}$, o emplear el valor absoluto: $\nabla f \approx [|z_5 - z_8| + |z_5 - z_6|]$.
- Otra consiste en usar las diferencias cruzadas: $\nabla f \approx [(z_5 - z_9)^2 + (z_6 - z_8)^2]^{1/2}$, o con los valores absolutos: $\nabla f \approx [|z_5 - z_9| + |z_6 - z_8|]$. Para esto se pueden usar máscaras de 2×2 llamadas *operadores de Roberts* de gradiente cuyos coeficientes son $(1,0,0,-1)$ para uno, y $(0,1,-1,0)$ para el otro; se toma el valor absoluto del resultado de cada uno de ellos, y luego se suman.
- También se puede aproximar empleando entornos mayores, como por ejemplo: $\nabla f \approx |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| + |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)|$ Para esta ecuación, se usan los *operadores de Prewitt*, máscaras de 3×3 cuyos coeficientes son: $(-1,-1,-1,0,0,1,1,1)$ y $(-1,0,1,-1,0,1,-1,0,1)$.
- También se puede aproximar por medio de los operadores de Sobel, cuyos coeficientes son: $(-1,-2,-1,0,0,0,1,2,1)$ y $(-1,0,1,-2,0,2,-1,0,1)$.

1.5. Mejora en el dominio de la frecuencia

Los principios de mejora en el dominio de la frecuencia son claros. Simplemente se trata de calcular la transformada de Fourier de la imagen a intensificar, multiplicar el resultado por la

función de transferencia de un filtro y, finalmente, tomar la transformada de Fourier inversa para llegar a una imagen mejorada.

1.5.1. Filtrado de pasa bajo

Los bordes y otras transiciones bruscas en los niveles de gris de una imagen contribuyen significativamente al contenido en altas frecuencias de su transformada de Fourier. Por lo tanto el difuminado o suavizado se consigue, en el dominio de la frecuencia, a base de atenuar un rango específico de componentes de alta frecuencia en la transformada de una imagen dada.

Debemos obtener $G(u,v) = F(u,v) \cdot H(u,v)$, y la transformada inversa de $G(u,v)$ dará la imagen $g(x,y)$ atenuada como deseábamos ($F(u,v)$ es la transformada de Fourier de la imagen original). El principal problema consiste en encontrar una función de transferencia del filtro ($H(u,v)$) que de $G(u,v)$ atenuando los componentes de alta frecuencia de $F(u,v)$.

1.5.1.1. Filtro ideal: Un filtro pasa bajo bidimensional ideal, es aquel cuya función de transferencia verifica la relación:

$$\begin{aligned} H(u,v) &= 1 && \text{si } D(u,v) \leq D_0 \\ &= 0 && \text{si } D(u,v) > D_0 \end{aligned}$$

donde D_0 es una cantidad especificada no negativa, y $D(u,v)$ es la distancia desde el punto (u,v) hasta el origen, es decir : $D(u,v) = [u^2 + v^2]^{1/2}$.

El nombre de filtro ideal indica que todas las frecuencias dentro de un círculo de radio D_0 pasan sin atenuación, mientras el resto se atenúa completamente. A D_0 se le denomina *frecuencia de corte*, y es el que determina las características del filtro.

Cuanto mayor sea el D_0 , menor será la pérdida de nitidez.

1.5.1.2. Filtro de Butterworth: La función de transferencia espacial de Butterworth de pasa bajo (BLPF) de orden n , y con emplazamiento de la frecuencia de corte a una distancia D_0 del origen, está definido por la relación: $H(u,v) = 1 / [1 + (D(u,v) / D_0)^{2n}]$, donde $D(u,v)$ es la distancia antes mencionada.

Al contrario de lo ocurrido con el filtro ideal de pasa bajo, la función de transferencia del filtro de Butterworth de pasa bajo, carece de una discontinuidad brusca que establezca un corte claro entre las frecuencias trasmisidas y las filtradas. Por lo tanto, es habitual definir la frecuencia de corte a partir del lugar de los puntos donde la función $H(u,v)$ corresponde a una determinada fracción de su valor máximo (por ejemplo $\max/2$, es decir, que D_0 va a ser igual al $D(u,v)$ donde $H(u,v) = \max/2$).

1.5.2. **Filtrado de pasa alto**

Debido a que los bordes y demás cambios bruscos de los niveles de gris están relacionados con las componentes de alta frecuencia, puede lograrse el realce de la imagen en el dominio de la frecuencia mediante un proceso de filtrado de pasa alto, que atenúa las componentes de baja frecuencia, sin modificar las de alta frecuencia.

1.5.2.1. Filtro ideal: Un filtro de pasa alto bidimensional ideal viene caracterizado por una función de transferencia que verifica la relación:

$$\begin{aligned} H(u,v) &= 0 \quad \text{si } D(u,v) \leq D_0 \\ &= 1 \quad \text{si } D(u,v) > D_0 \end{aligned}$$

donde D_0 es la distancia de corte medida desde el origen, y $D(u,v)$ es la distancia mencionada en los puntos anteriores.

Este filtro es el opuesto al filtro ideal de pasa bajo, ya que atenúa todas las frecuencias que quedan dentro del círculo de radio D_0 , a la vez que deja pasar sin atenuación alguna el resto de las frecuencias.

Al igual que el filtro ideal de pasa bajo, no puede ser físicamente realizado.

1.5.2.2. Filtro de Butterworth: La función de transferencia del filtro de Butterworth de pasa alto (BHPF) de orden n , y con frecuencia de corte situada a una distancia D_0 del origen, está definido por la relación: $H(u,v) = 1 / [1 + (D_0 / D(u,v))^{2n}]$, donde $D(u,v)$ es la distancia antes mencionada.

Este tipo de filtro produce que zonas de distintos niveles de gris (pero con variaciones suaves) aparezcan como si fueran iguales. Para aliviar este problema, se añade una constante a la función de transferencia, ya que así se conservan las componentes de baja frecuencia, y amplificando aún más las componentes de alta frecuencia. Esta técnica se denomina *énfasis de las frecuencias bajas*.

Se suele usar también después del énfasis de frecuencias altas un postfiltrado para redistribuir los niveles de gris, para lo cual se puede usar la ecualización del histograma.

1.5.3. **Filtrado homomórfico**

Una imagen puede $f(x,y)$ expresarse en término de sus componentes de iluminación y reflectancia: $f(x,y) = i(x,y).r(x,y)$. Por lo tanto sería bueno poder separar ambas componentes en una imagen.

Como la transformada de Fourier de un producto de dos funciones no es separable, debemos definir la función: $z(x,y) = \ln(f(x,y)) = \ln[i(x,y)] + \ln[r(x,y)]$, y luego obtener: $Z(u,v) =$

$I(u,v) + R(u,v)$, donde I y R son la transformada de Fourier de $\ln[i(x,y)]$ y $\ln[r(x,y)]$ respectivamente.

Si se procesa $Z(u,v)$ por medio de un filtro cuya función sea $H(u,v)$, entonces: $S(u,v) = H(u,v).Z(u,v) = H(u,v).I(u,v) + H(u,v).R(u,v)$, donde $S(u,v)$ es la transformada de Fourier del resultado.

En el dominio espacial: $s(x,y) = i'(x,y) . r'(x,y)$, donde $i'(x,y)$ y $r'(x,y)$ son las transformadas de Fourier inversa de $\{H(u,v) I(u,v)\}$ y $\{H(u,v) R(u,v)\}$ respectivamente.

Como $z(x,y)$ es el \ln de $f(x,y)$, entonces para obtener $g(x,y)$ (la imagen mejorada) debemos aplicarle la función inversa a $s(x,y)$:

$$g(x,y) = \exp[s(x,y)] = \exp[i'(x,y)].\exp[r'(x,y)] = i_0(x,y).r_0(x,y)$$

siendo i_0 el componente de iluminación de la imagen de salida, y r_0 el componente de reflectancia.

En resumen, los pasos son:

1. Teniendo f , le aplico el \ln , obteniendo z ;
2. A z le saco la transformada de Fourier, y obtengo Z ;
3. Multiplico a Z por H , y obtengo S ;
4. A S le aplico la transformada de Fourier inversa, y obtengo s ;
5. Aplico la función \exp a s y obtengo el resultado final g .

La componente de iluminación se puede asociar con las componentes de baja frecuencia de la transformada de Fourier del algoritmo de una imagen; mientras que la componente de reflectancia se puede asociar con las de alta frecuencia. Aunque estas son aproximaciones no muy exactas, pueden emplearse para ayudar a la mejora de la imagen.

Usando una función de filtro homomórfico $H(u,v)$, se puede tener control sobre ambas componentes, ya que $H(u,v)$ afectará de forma distinta a las componentes de alta y baja frecuencia.

1.6. Procesamiento de imágenes en color

El empleo del color en las imágenes es un potente descriptor que a menudo simplifica la identificación de un objeto y su extracción de la escena.

El procesamiento de imágenes en color se divide en dos áreas fundamentales: el procesamiento en color real (o todo color), y en falso color. En la primera categoría las imágenes en cuestión se adquieren mediante un sensor de color, como una cámara de televisión o un escáner de color. En la segunda, el problema consiste en asignar un nivel de color a una determinada intensidad o rango de intensidad monocromático (darle color a una imagen monocromática o a ciertos elementos de ella).

1.6.1. Fundamentos del color

Cuando un haz de rayos solares atraviesa un prisma de vidrio, el haz emergente no es blanco, sino que consiste en un espectro continuo de colores que van desde el violeta hasta el rojo, pasando por azul, verde, amarillo y naranja.

Básicamente, los colores que los seres humanos percibimos en un objeto están determinados por la naturaleza de la luz reflejada por el objeto. La luz visible está formada por una banda de frecuencias relativamente estrecha del espectro electromagnético. Un cuerpo que refleje luz que está relativamente equilibrada en todas las longitudes de onda aparece como blanco para el observador. Sin embargo, un cuerpo que tiene una mayor reflectancia en una determinada banda del espectro visible aparece como coloreado.

Si la luz es acromática (sin color), su único atributo es la intensidad (o cantidad de luz). Así, el término *nivel de gris* se refiere a una medida escalar de la intensidad que va del negro al blanco, pasando por los grises.

Para describir las características de una fuente cromática de luz, se emplean tres magnitudes básicas:

1. La **radiancia**: La cantidad total de energía que sale de la fuente luminosa. Se mide en watos (w).
2. La **luminancia**: Se mide en lúmines (Ln). Proporciona una medida de la cantidad de energía que un observador percibe procedente de una fuente luminosa (difiere de la radiancia porque si está fuera del rango que el hombre percibe, esta es nula).
3. El **brillo**: Es un descriptor subjetivo que resulta difícil de medir. Incluye la nación acromática de la intensidad, y es uno de los factores fundamentales para describir las sensaciones del color.

Debido a la estructura del ojo humano, todos los colores se ven como combinaciones variables de los denominados tres colores primarios: rojo (R), verde (G), y azul (B).

Los colores primarios se pueden sumar para obtener los colores secundarios de luz: magenta, cián, y amarillo. Mezclando los tres colores primarios, o uno secundario con su color primario opuesto, en las proporciones adecuadas se obtiene luz blanca.

Es importante la distinción entre *colores primarios de la luz* y *colores primarios de pigmentos o colorantes*. Para los últimos, un color primario se define como algo que absorbe o sustrae un color primario de luz y refleja o trasmite los otros dos. Por lo tanto, los colores primarios de pigmentos son: magenta, cián y amarillo. Y los secundarios son rojo, verde y azul. La combinación adecuada de los tres pigmentos primarios o de uno secundario con su primario opuesto, produce el negro.

Las características generalmente empleadas para distinguir un color de otro son:

1. **Brillo**: está relacionado con la noción cromática de intensidad.
2. **Tono**: representa el color dominante en una mezcla (por ejemplo, si decimos que un objeto es rojo, amarillo o naranja, estamos especificando su tono).
3. **Saturación**: se refiere a la pureza relativa o cantidad de luz blanca mezclada con su tono. Los colores puros del espectro están completamente saturados, colores como el rosa (rojo y blanco) están menos saturados, siendo su grado de saturación inversamente proporcional a la cantidad de luz blanca añadida.

El tono y la saturación considerados conjuntamente constituyen la *cromaticidad*, por lo tanto, un color se puede caracterizar por su brillo y su cromaticidad.

La cantidad de rojo, verde y azul necesarios para formar un color particular se denominan los valores *triestímulo* y se indican por X, Y, y Z respectivamente. Así, un color queda especificado por sus *coeficientes tricromáticos*, definidos como:

$$x = X / (X+Y+Z) \quad y = Y / (X+Y+Z) \quad z = Z / (X+Y+Z)$$

Siendo $x + y + z = 1$.

1.6.2. Modelos de color

Un modelo de color es la especificación de un sistema de coordenadas tridimensional y de un subespacio de este sistema en el que cada color quede representado por un único punto.

La mayoría de los modelos están orientados bien hacia el hardware, o bien hacia aplicaciones donde se pretende manipular el color. Entre los primeros se encuentran los modelos:

- (a) **RGB** (rojo, verde, azul).
- (b) **CMY** (cián, magenta, amarillo).
- (c) **YIQ** (Y= luminicencia, I y Q son dos componentes cromáticos llamados fase y cuadratura).

Entre los segundos, los modelos son:

- (d) **HSI** (tono, saturación, intensidad).
- (e) **HSV** (tono, saturación, valor):

1.6.2.1. El modelo de color RGB: En este modelo, cada color aparece con sus componentes espectrales primarias de rojo, verde y azul. Este modelo está basado en un sistema de coordenadas cartesianas. El subespacio de color de interés es el cubo de ancho, alto, y largo igual a 1, desde el origen, en el cual los colores RGB están en tres vértices (1,0,0) (0,1,0) (0,0,1), respectivamente; el cián, magenta, y amarillo en otros tres vértices (0,1,1) (1,0,1)

(1,1,0) , respectivamente; el negro en el origen (0,0,0); y el blanco en el vértice opuesto (1,1,1).

La escala de grises se extiende del negro al blanco a lo largo de una diagonal del cubo, y los colores son puntos del cubo o de su interior, definidos por vectores que se extienden desde el origen. Por conveniencia, se supone que todos los valores de color han sido normalizados, de forma que el subespacio de interés es el cubo unidad, es decir que todos los valores de R, G, B están en el intervalo [0,1].

Las imágenes del modelo de color RGB consiste en tres planos de imagen independientes, uno por cada color primario, que al combinarlos forman una imagen en color compuesta.

1.6.2.2. El modelo de color CMY: Como ya se vio, C, M,Y son los colores secundarios de luz, o bien los colores primarios de los pigmentos. Por ejemplo, si se ilumina con luz blanca una superficie recubierta de un pigmento cián, no se refleja luz roja de ella (el cián absorbe la luz roja de la luz blanca).

Este modelo es igual al anterior, pero invirtiendo las posiciones de los colores. Para pasar del modelo RGB al CMY se hace la siguiente operación:

$$(C, M, Y) = (1, 1, 1) - (R, G, B)$$

Para la conversión inversa (de CMY a RGB) se hace:

$$(R, G, B) = (1, 1, 1) - (C, M, Y)$$

En ambos casos todos los valores de color están normalizados.

1.6.2.3. El modelo de color YIQ: Básicamente, el YIQ es una recodificación del RGB. La componente Y de este sistema proporciona toda la información de vídeo que necesita un sistema de televisión monocromática. La conversión de RGB a YIQ se define como:

$$\begin{vmatrix} Y \\ I \\ Q \end{vmatrix} = \begin{vmatrix} 0,299 & 0,587 & 0,114 \\ 0,596 & -0,275 & -0,321 \\ 0,212 & -0,523 & 0,311 \end{vmatrix} \begin{vmatrix} R \\ G \\ B \end{vmatrix}$$

Para la conversión inversa (de YIQ a RGB), se lleva a cabo la operación matricial inversa.

La principal ventaja del modelo YIQ en el procesamiento de imágenes es que la luminancia (Y) y la información del color (I y Q) están desacopladas.

1.6.2.4. El modelo de color HSI: El modelo HSI debe su utilidad a dos hechos básicos:

- 1) La componente de la intensidad (I) está desacoplada de la información cromática contenida en la imagen.
- 2) Las componentes de tono (H) y saturación (S) están íntimamente relacionadas con la forma en que los seres humanos percibimos el color.

Este modelo se basa en el tono (atributo cromático que describe un color puro), la saturación (medida del grado en que un color puro está diluido en luz blanca), y la intensidad.

Conversión de RGB a HSI: Para esta conversión se debe realizar las siguientes operaciones:

- $I = (1/3) (R + G + B)$
- $S = 1 - [3/(R + G + B)].[\min(R,G,B)]$
- $H = \cos^{-1} \{0,5 . [(R-G) + (R-B)] / [(R-G)^2 + (R-B).(G-B)]^{1/2}\}$

donde $H = 360^\circ - H$, si $(B/I) > (G/I)$. Para poder normalizar el tono en el intervalo $[0,1]$, se hace $H = H/360^\circ$. Si $S = 0$, el tono no está definido. Si $I = 0$, entonces la saturación no está definida.

Conversión de HSI a RGB: Para esta conversión se debe distinas operaciones, dependiendo del valor de H:

1. Para H en $(0^\circ, 120^\circ]$ tenemos:

- $r = (1/3) \{1 + [S . \cos H / \cos (60^\circ-H)]\}$
- $b = (1/3) (1 - S)$
- $g = 1 - (r + b)$

2. Para H en $(120^\circ, 240^\circ]$ tenemos:

- $H = H - 120^\circ$
- $r = (1/3) (1 - S)$
- $b = 1 - (r + g)$
- $g = (1/3) \{1 + [S . \cos H / \cos (60^\circ-H)]\}$

3. Para H en $(240^\circ, 360^\circ]$ tenemos:

- $H = H - 240^\circ$
- $r = 1 - (g + b) (1/3) (1 - S)$
- $b = (1/3) \{1 + [S . \cos H / \cos (60^\circ-H)]\}$
- $g = (1/3) (1 - S)$

Para obtener los valores de RGB hago:

- $R = 3 I r$
- $G = 3 I g$
- $B = 3 I b$

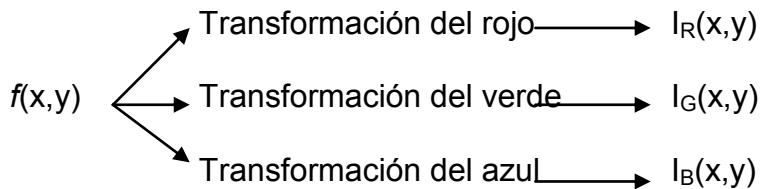
1.6.3. Procesamiento de imágenes en falso color

Hay métodos para asignar color a imágenes monocromáticas basándose en algunas propiedades de su contenido de niveles de gris. Algunos de los métodos son los siguientes:

1.6.3.1. División de la intensidad: El intervalo de niveles de gris $[0, L]$, donde 0 es el negro, y L es el blanco, se divide en M regiones o subintervalos disjuntos, asignándole a cada uno de estos un color distinto. La imagen monocromática es mapeada, y a cada pixel se le asigna el color correspondiente de acuerdo a la región en que se encuentre su nivel de gris.

Es decir que : $f(x,y) = c_k \quad \text{si } f(x,y) \in R_k$, donde c_k es el color correspondiente a la región k -esima del intervalo, con $0 \leq k \leq M$.

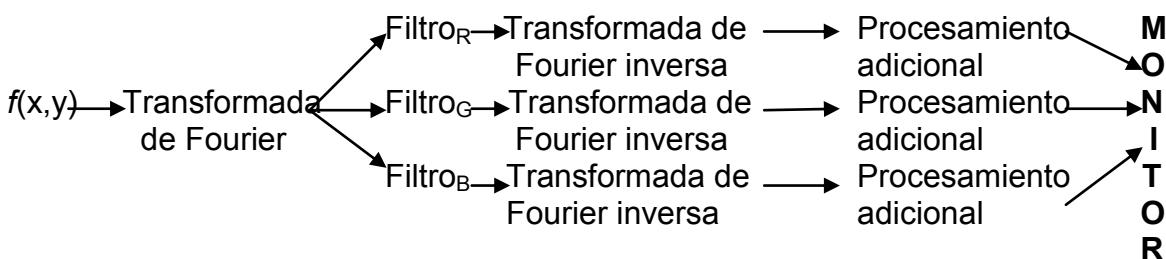
1.6.3.2. Transformaciones de color del nivel de gris: La idea consiste en llevar a cabo tres transformaciones independientes del nivel de gris de cualquier pixel de entrada. A continuación, los tres resultados alimentan separadamente los cañones rojo, verde y azul de un monitor de TV en color. Este método produce una imagen compuesta cuyo contenido de color está modulado por la naturaleza de las funciones de transformación (son transformaciones del nivel de gris de una imagen, y no de la posición de las funciones).



1.6.3.3. Un método de filtrado: La idea es semejante a la de filtrado básico, pero aquí la transformada de Fourier de la imagen está modificada de forma independiente por cada una de las tres funciones de filtro, para generar tres imágenes que puedan alimentar las entradas rojo, verde y azul de un monitor color. Para cada uno de los colores, los pasos son los siguientes:

1. La transformada de Fourier de la imagen de entrada se altera al utilizar una determinada función de filtro.
2. Al resultado se le hace la transformada de Fourier inversa para obtener una imagen.
3. Puede llevarse a cabo algún procesamiento adicional (como una ecualización de histograma) sobre la imagen obtenida en 2, antes de que alimente a la entrada correspondiente (R, G, o B) del monitor color.

El objetivo es codificar el color teniendo en cuenta el contenido de la frecuencia.



1.6.4. Procesamiento de imágenes en color real

En el modelo HSI tenemos dos ventajas:

1. La intensidad e información del color de este modelo están desacopladas.
2. El tono y la saturación están íntimamente relacionados con la forma en que los seres humanos describen la percepción del color.

1.6.4.1. Componentes HSI a partir de una imagen RGB: La terminología usada para referirse a una imagen de color real normalmente proporciona el número de bits utilizados en la representación del color. Así, una imagen de color de 24 bits indica que se usan 24 bits para representarla. En el modelo RGB, generalmente, los bits están distribuidos por igual entre los tres componentes de color de la imagen. Para normalizar el valor (que pertenezca al rango $[0,1]$), se debe multiplicar el valor de cada componente por $\{1 / (2^b - 1)\}$, siendo b la cantidad de bits de cada componente.

En el modelo HSI la situación de los componentes es algo diferente. La intensidad, se puede manejar igual que cualquiera de los componentes del modelo RGB, al igual que la saturación, que toma valores desde 0 (sin saturación) hasta 1 (saturación total), con incrementos de $[1 / (2^b - 1)]$. Para el tono, la situación cambia, ya que este se mide en grados pertenecientes al intervalo $[0^\circ, 360^\circ]$, por lo que el valor representado se debe multiplicar por $[360^\circ / (2^b - 1)]$.

1.6.4.2. Mejora utilizando el modelo HSI: Este modelo es ideal para el mejoramiento de la imagen, ya que permite usar cualquiera de las técnicas de mejora monocroma sin realizar ningún cambio, aplicándolas al componente de intensidad, sin tocar la información relacionada con el color.

Para esto, primero se debe convertir la imagen en el modelo RGB al modelo HSI, luego se aplica la técnica correspondiente al componente de intensidad, y luego se transforma el resultado al modelo RGB para poder verlo. De esta manera, el contenido de color de la imagen no resulta afectada.

Si aplicara las técnicas de mejora directamente sobre los componentes RGB, podrían afectar los colores de la imagen.

1.7. Segmentación de imágenes

La parte del procesamiento donde se extrae información de la imagen se conoce comúnmente como *análisis de imágenes*. El primer paso en este es la segmentación de la imagen, que consiste en subdividirla en sus partes constituyentes u objetos. El nivel al que se lleva a cabo la segmentación depende del problema, es decir, la segmentación se debe detener cuando los objetos de interés hayan sido aislados.

Los algoritmos de segmentación de imágenes monocromáticas generalmente se basan en una de las dos propiedades básicas de los valores del nivel de gris: discontinuidad y similitud. En la primera categoría, se divide una imagen basándose en los cambios bruscos de nivel de gris. En la segunda categoría, los principales métodos se basan en la umbralización, crecimiento de región, y división y fusión de regiones.

1.7.1. Detección de discontinuidades

Los tres tipos básicos de discontinuidad de una imagen digital que se pueden detectar son: puntos, líneas, y bordes.

En la práctica, la forma más común de ver las discontinuidades es pasar una máscara (filtro o ventana) a través de la imagen. En una máscara el valor resultante de cada pixel depende de su valor y del de los vecinos. Por ejemplo, si tenemos una máscara de 3x3 con coeficientes $w_1 \dots w_9$ (de izquierda a derecha, comenzando en la primera fila), entonces el resultado será:

$$R = \sum_{i=0}^9 z_i w_i$$

siendo z_i el nivel de gris del pixel asociado con el coeficiente w_i . Habitualmente, la respuesta de la máscara está definida con respecto a la posición de su centro, R sería el valor resultante para el pixel asociado con el coeficiente central de la máscara. Cuando la máscara está centrada en un pixel del límite, la respuesta se calcula utilizando el entorno parcial apropiado.

1.7.1.1. Detección de puntos

Para la detección de puntos aislados de una imagen se utiliza la siguiente máscara:

-1	-1	-1
-1	8	-1
-1	-1	-1

Se sabe que se ha detectado un punto en la posición en la que está centrada la máscara si $|R| > T$ donde T es un umbral no negativo.

Lo que se trata de hacer es medir la diferencia en los niveles de gris entre el punto central y sus vecinos, partiendo de la idea de que el nivel de gris de un punto aislado será bastante diferente del de sus vecinos.

1.7.1.2. Detección de líneas

Para este caso, se deben usar cuatro máscaras:

M1		
-1	-1	-1
2	2	2
-1	-1	-1

M2		
-1	-1	2
-1	2	-1
2	-1	-1

M3		
-1	2	-1
-1	2	-1
-1	2	-1

M4		
2	-1	-1
-1	2	-1
-1	-1	2

Si la primera máscara se traslada por una imagen, podría responder con más fuerza a líneas (de un pixel de grosor) orientadas horizontalmente. Con un fondo constante, la respuesta máxima resultará cuando la línea pase por la fila central de la máscara. La segunda máscara es para líneas en 45° , la tercera para líneas verticales, y la cuarta para líneas en -45° .

Sean R_1 , R_2 , R_3 , y R_4 las respuestas en un punto de las máscaras M_1 , M_2 , M_3 , y M_4 respectivamente, si $|R_i| > |R_j|$ para todo $j \neq i$, con $i,j=1,\dots,4$, este punto tendrá mayor probabilidad de estar asociado con un a línea en la dirección dada por M_i .

1.7.1.3. Detección de bordes

La detección de bordes es el método más común para detectar discontinuidades significativas en el nivel de gris.

1.7.1.3.1. Formulación básica: Un borde es la frontera entre dos regiones con propiedades de nivel de gris relativamente distintas.

Básicamente la idea que subyace en la mayor parte de las técnicas de detección de bordes es el cálculo de un operador local de derivación.

El modulo de la derivada primera se puede usar para detectar la presencia de un borde en una imagen, y el signo de la derivada segunda se puede usar para determinar si un pixel borde está situado en el lado oscuro o claro del mismo. A su vez, la derivada segunda tiene un paso por cero en el punto medio de una transición de nivel de gris.

Para la derivada primera se puede usar el modulo del *gradiente*, y para la derivada segunda se puede utilizar el *Laplaciano*.

1.7.1.4. Detección combinada

Al usar una formulación multimáscara, se hace posible desarrollar un método para determinar si un pixel tiene más posibilidad de ser un punto aislado, o parte de una línea o de un borde.

1.7.2. Enlazado de bordes y detección de límites

En general, las técnicas antes mencionadas detectan las discontinuidades de intensidad, raramente los pixels encontrados por ellas caracterizan completamente el límite, debido a diferentes problemas que introducen discontinuidades en los límites encontrados. Por lo cual se deben emplear procedimientos de enlazado y de detección de límites diseñados para reunir pixels del borde en límites que tengan algún sentido.

1.7.3. Umbralización

La umbralización es uno de los métodos más importantes de la segmentación de imágenes.

1.7.3.1. Fundamentos

La umbralización se puede contemplar como una operación que implica realizar comprobaciones frente a una función T de la forma: $T = T[x, y, f(x,y), p(x,y)]$ donde $f(x,y)$ es el nivel de gris del punto (x,y) , y $p(x,y)$ representa alguna propiedad local de este punto (por ejemplo, la media del nivel de gris de una vecindad centrada en (x,y)). Una imagen umbralizada $g(x,y)$ se define como:

$$g(x,y) = \begin{cases} 1 & \text{si } f(x,y) > T \\ 0 & \text{si } f(x,y) \leq T \end{cases}$$

De este modo, los pixels marcados con 1 (o cualquier otro nivel de intensidad conveniente) corresponde a objetos, mientras que los otros corresponden al fondo; o viceversa, dependiendo si el fondo es oscuro y el objeto luminoso, o al revés.

Cuando T depende solamente de $f(x,y)$, el umbral se denomina *global*. Si T depende a su vez de $p(x,y)$ se denomina *local*. Si además depende de las coordenadas espaciales x e y , se denomina *dinámico*.

Si la imagen se quiere segmentar en más de dos partes, se llama *umbralización multinivel*, y se debe usar más de un umbral. Por ejemplo, para dos objetos (el objeto 1 más luminoso que el objeto 2) y el fondo (oscuro), se deben tener en cuenta dos umbrales T_1 y T_2 donde:

$$g(x,y) = \begin{cases} 2 & \text{si } f(x,y) > T_2 \text{ \{objeto 1\}} \\ 1 & \text{si } T_2 \geq f(x,y) \geq T_1 \text{ \{objeto 2\}} \\ 0 & \text{si } f(x,y) < T_1 \text{ \{fondo\}} \end{cases}$$

1.7.3.2. Umbralización global simple

La más sencilla de todas las técnicas de umbralización es la partición del histograma usando un único umbral, T . La segmentación se completa escaneando la imagen pixel a pixel y

etiquetando a cada uno de ellos como perteneciente al objeto o al fondo, dependiendo de que el nivel de gris de ese pixel sea mayor o menor que el valor T. El éxito del método depende de como se pueda dividir el histograma.

En general, para que este método tenga éxito, se debe tener un entorno muy bien controlado.

1.7.3.3. Umbralización óptima

Supongamos que una imagen tiene sólo dos regiones de brillo principales. El histograma de una imagen de este tipo se puede considerar como una estimación de la función densidad de probabilidad de brillo, $p(z)$. Esta función de densidad total es la suma o mezcla de dos densidades unimodales, una para las regiones claras y la otra para las oscuras. Además, los parámetros de la mezcla son proporcionales a la superficie de cada región de brillo. Si se conoce o se supone la forma de la densidad, se puede determinar un umbral óptimo (en término del error mínimo) por segmentación de la imagen en las dos regiones de brillo.

Teniendo la fórmula de $p(z)$, se determina el error de tipo 1 y 2 con respecto a T, sumando ambos, se forma el error total. Luego, para encontrar el valor umbral para el que el error total es mínimo se necesita derivar dicho error, igualarlo a cero, y después determinar el T que cumple dicha ecuación.

En caso de que todo esto se pueda hacer, se obtiene el umbral óptimo T.

1.7.4. Segmentación orientada a regiones

El objetivo de la segmentación es dividir una imagen en regiones. Hay técnicas de segmentación basadas en encontrar directamente las regiones.

1.7.4.1. Formulación básica

Sea R la representación de la región completa de una imagen. Se puede contemplar la segmentación como un proceso que divide a R en n subregiones, R_1, \dots, R_n , de forma que:

$$a) \bigcup_{i=1}^n R_i = R.$$

- b) R_i es una región conexa, para $i = 1, \dots, n$.
- c) $R_i \cap R_j = \emptyset$, para $\forall i, j$, con $i \neq j$.
- d) $P(R_i) = \text{verdadero}$, para $i = 1, \dots, n$.
- e) $P(R_i \cup R_j) = \text{falso}$, para $i \neq j$.

Donde $P(R_i)$ es un predicado lógico sobre los puntos del conjunto R_i (son propiedades que deben cumplir los pixels de una región, como por ejemplo tener la misma intensidad).

1.7.4.2. Crecimiento de regiones por agrupación de pixels

Este es un procedimiento que agrupa pixels o subregiones dentro de otras regiones más grandes. Se comienza con un conjunto de *puntos generadores*, a partir de los cuales van creciendo las regiones al agregar a cada uno de estos puntos los pixels próximos que tienen propiedades similares (como nivel de gris, textura, color).

1.7.4.3. División y fusión de regiones

Se divide la imagen en un conjunto de regiones arbitrarias disjuntas, y después se fusionan y/o dividen las regiones intentando satisfacer las condiciones vistas antes (en 4.1). Hay distintos algoritmos, pero en general lo que se hace es lo siguiente:

- a) Supongamos que la imagen se representa como una región R_0 .
- b) Dividir en 4 cuadrantes disjuntos cualquier región R_i donde $P(R_i) = \text{falso}$.
- c) Fusionar las regiones adyacentes R_i y R_j para las que $P(R_i \cup R_j) = \text{verdadero}$.
- d) Parar cuando no se puedan hacer más divisiones o fusiones.

El concepto de *segmentación de textura* está basado en el empleo de medidas de textura para los predicados $P(R_i)$.

1.8. Representación y Descripción

Después de segmentar una imagen, el conjunto resultante de pixels segmentados se representa y describe normalmente en una forma adecuada para su posterior procesado por computadora.

Representar una región implica dos posibilidades:

1. Hacerlo en términos de sus características externas (contorno).
2. En término de sus características internas (los pixels que comprenden la región).

El siguiente paso consiste en *describir* la zona en la representación elegida. Las características seleccionadas como descriptores deberían ser tan insensibles como fuera posible a variaciones como cambios de tamaño, traslación, y rotación.

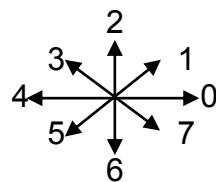
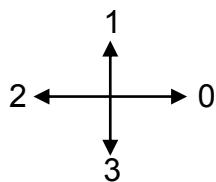
1.8.1. Esquema de representación

Las técnicas de segmentación devuelven una serie de pixels (del contorno o de la región), los cuales se pueden usar para obtener descriptores, pero normalmente se utilizan esquemas que compacten los datos en representaciones que son considerablemente más útiles en el cálculo de descriptores.

Hay diversos métodos de representación, entre ellos se encuentran los siguientes:

1.8.1.1. Código de cadena

Los códigos de cadena se usan para representar un contorno por medio de una sucesión conexa de segmentos de longitud y dirección especificadas. Normalmente esta representación se basa en segmentos de conectividad 4 u 8. La dirección de cada segmento se codifica usando uno de los siguientes esquemas:



Se podría generar un código de cadena de una imagen siguiendo el contorno de ella en el sentido de las agujas del reloj, asignando una dirección a los segmentos que unen cada par de pixels conexos. Este método tiene dos razones principales por la cual no es usado:

1. La cadena de códigos es muy larga.
2. Cualquier perturbación pequeña en el contorno origina cambios en el código.

Una solución consiste en volver a muestrear el contorno seleccionando un espaciado de cuadrícula mayor; para esto, se recorre el contorno, y cada pixel de este, es asignado al punto más próximo en la nueva cuadrícula. Luego se obtiene su código cadena.

El código de cadena de un contorno depende del *punto de partida*. Para solucionar esto, se puede normalizar el código cadena de la siguiente manera: se toma la cadena como una secuencia circular de números, y se redefine el punto de partida de forma que la cadena forme un entero de módulo mínimo.

Se puede conseguir la normalización de tamaño alterando el tamaño del cuadriculado remuestreado.

El código varia de acuerdo a la rotación y al cambio de escala.

1.8.1.2. Aproximación poligonal

Este método trata de representar un contorno a través de un polígono, tratando de captar la esencia de la forma del contorno con un polígono del menor número de lados. Hay distintos métodos para esto:

1. Primero un método para *encontrar polígonos de perímetro mínimo*. Se encierra el contorno en células concatenadas de un tamaño igual a $(d \times d)$. Luego el contorno se encoge formando un polígono de perímetro mínimo que esté completamente dentro de la secuencia de celdas.

2. La *técnica de fusión* consiste en ir uniendo puntos del contorno hasta que el ajuste de la curva de error mínimo cuadrado de los puntos fusionados hasta el momento traspase un umbral preestablecido. Cuando se llega a esto, se comienza con otra fusión, y así sucesivamente hasta recorrer todo el contorno, luego se unen las líneas, y queda formado el polígono.
3. Una *técnica que divide lados* del contorno, consiste en subdividir lados sucesivamente hasta que se cumpla un criterio dado. Por ejemplo, un criterio podría ser que la distancia perpendicular máxima desde un lado del contorno a la línea que une sus dos extremos no exceda un umbral preestablecido. Si lo sobrepasa, entonces el punto más alejado se convierte en un vértice, subdividiendo así el lado en dos sublados. Para comenzar, se tomaría la línea que une los dos puntos más alejados del contorno.

1.8.1.3. Firmas

Una firma es una representación funcional unidimensional de un contorno, y se puede generar de varias formas (pero sin importar cual sea, lo que se busca es reducir la representación del contorno a una función unidimensional). Una de las formas es representar la *distancia desde el centro* al contorno como una función del ángulo. Es decir que para cada ángulo θ , se tomará la distancia desde el centro hasta el contorno.

Esta técnica es invariantes con respecto a la translación, pero no con respecto a la rotación y a la escala. Se puede normalizar con respecto a la rotación seleccionando como punto de partida ($\theta = 0$) la distancia mayor desde el centro.

Para los cambios de tamaño, se puede normalizar el resultado al *escalar* todas las funciones de tal manera que siempre abarquen el mismo rango de valores. Por ejemplo, [0, 1].

1.8.1.4. Lados del contorno

A veces es útil descomponer un contorno en lados, para reducir así la complejidad del contorno, y simplificar el proceso de descripción. Esto es muy útil cuando el contorno presenta una o más concavidades significativas que contienen información sobre la forma. Para esto se puede usar el *cerco conexo*.

El *cerco conexo* H de un conjunto arbitrario S es el conjunto convexo más pequeño que contiene a S . El conjunto diferencia $H-S$ se llama *deficiencia convexa* D del conjunto S . Para dividir el contorno de la región, se sigue el contorno de S , y se marcan aquellos puntos donde se hace una transición hacia dentro o fuera de un componente de la deficiencia convexa.

1.8.1.5. El esqueleto de una región

Una forma de representar la forma de una región es a través de su esqueleto, el cual se obtiene mediante un algoritmo de reducción (llamado esqueletización).

Para una región R con borde B , se define su esqueleto de la siguiente manera:

Para cada punto p de R , se encuentra su vecino más próximo en B . Si p tiene más de un vecino de estos, se dice que pertenece al eje medio (esqueleto) de R . El concepto de más próximo depende de la definición de distancia usada.

La implementación directa de esto es imposible por razones de cálculo, por lo cual se usan algoritmos de reducción que van suprimiendo los puntos del margen de una región, cumpliendo con las siguientes restricciones:

1. No eliminar puntos extremos.
2. No romper la continuidad.
3. No causar excesiva erosión en la región.

1.8.2. Descriptores de contorno

1.8.2.1. Algunos descriptores simples

- La *longitud* de un contorno, que se puede obtener de forma aproximada contando la cantidad de pixels que lo forman.
- El *diámetro* de un contorno B , el cual es la máxima distancia entre dos puntos cualquiera de B , la línea que une a esos dos puntos (los dos más distantes), se llama *eje mayor* del contorno: $\text{Diam}(B) = \max_{i,j}(D(p_i, p_j))$.
- La *curvatura* se define como el ritmo de variación de la pendiente, y, en general, es muy difícil de calcular, usándose sobre todo para clasificar segmentos en cóncavos y convexos.
- Número de forma

El *número de forma* de un contorno se define como la *primera diferencia* (se obtiene contando en el sentido contrario a las agujas del reloj el número de direcciones que separan dos elementos adyacentes del código de cadena, y su primer valor se obtendrá comparando el último elemento del código cadena con el primero) de módulo más pequeño. El *orden n* de un número de forma se define como el número de dígitos de su representación.

- Otros más complejos como descriptores de Fourier o Momentos.

1.8.3. Descriptores de región

1.8.3.1. Algunos descriptores simples

- El *área* de una región se define como el número de pixels contenido en su contorno.
- El *perímetro* de una región es la longitud de su contorno.
- La *densidad* de una región es $\{\text{perímetro}^2 / \text{área}\}$.

Las tres son descriptores de región. Las dos primeras varían de acuerdo al tamaño, mientras que la tercera no.

- Otras medidas sencillas utilizadas como descriptores de región incluyen la *media* y la *mediana* de los niveles de gris, los valores *mínimo* y *máximo* del nivel de gris, y el *número de pixels* con valores por encima y por debajo de la media.

1.8.3.2. Descriptores topológicos

Las propiedades topológicas son útiles para descriptores de regiones. La *topología* es el estudio de las propiedades de una figura a las que no afecta ninguna deformación (estiramientos, rotación, etc.).

- Un descriptor topológico podría ser el *número de huecos* de la región.
- Otra propiedad útil para la descripción de regiones es el *número de componentes conexas*. Una componente conexa de un conjunto es un subconjunto de tamaño máximo tal que cualquier par de puntos del mismo se puede unir por una curva conexa trazada completamente dentro del subconjunto.
- Otra propiedad es el *número de Euler* (E), donde $E = C - H$, siendo C el número de componentes conexas, y H el número de huecos.

1.8.3.3. Textura

Un método importante para la descripción de regiones consiste en cuantificar su contenido de *textura*. Los tres métodos más usados para describir la textura de una región son:

1. Métodos estadísticos: proporcionan características de textura tales como suavidad, rugosidad, granulosidad, y otras similares.
2. Métodos estructurales: tratan de la composición de primitivas de imágenes.
3. Métodos espectrales: se basan en el espectro de Fourier.

1.8.4. Morfología

La morfología, en el contexto de morfología matemático, es una herramienta para extraer componentes de una imagen que sean útiles en la representación y descripción de la forma de una región, tales como contorno, esqueleto, y cerco conexo. También son de interés las técnicas morfológicas para el pre y postprocesado, tales como filtrado morfológico, la reducción y el recortado.

El lenguaje de la morfología matemática es la *teoría de conjuntos*. Los conjuntos representan las formas de los objetos de una imagen.

1.8.5. Descriptores relacionales

En este tipo de descriptores, se cuenta con *elementos primitivos* (representados por letras minúsculas) que representan una determinada forma. Estos elementos se relacionan de diferente manera para describir la estructura deseada. La forma en que se relacionen las primitivas varía, pudiendo formar *cadenas* de primitivas, o bien *árboles*.

1.9. Reconocimiento e Interpretación

El análisis de imágenes es un proceso que consiste en describir, identificar y comprender los patrones que son relevantes en el rendimiento de un trabajo basado en imágenes. Uno de los principales objetivos del análisis de imágenes por computadora consiste en dotar a una máquina, en algún sentido, de la capacidad de aproximación, similar a la de los seres humanos.

De este modo, un sistema automático de análisis de imágenes debería ser capaz de ofrecer varios grados de inteligencia. El concepto de inteligencia se podría definir para estos casos en base a ciertas características:

1. La capacidad de extraer la información de interés.
2. La capacidad de aprender a partir de ejemplos, y de generalizar este conocimiento para que se pueda aplicar en nuevas y diferentes circunstancias.
3. La capacidad de realizar inferencias a partir de una información incompleta.

1.9.1. Elementos del análisis de imágenes

Se puede dividir las técnicas de análisis de imágenes en tres áreas básicas:

1. **Procesado de bajo nivel:** trata de un tipo de funciones a las que se puede considerar como reacciones automáticas, y que no requieren inteligencia por parte del sistema de análisis de imágenes. Abarca a la adquisición y preprocesado de imágenes.

2. **Procesado de nivel intermedio:** se refiere a la labor de extracción y caracterización de los componentes de la imagen que se obtienen de un proceso de bajo nivel. Abarca la segmentación y la descripción de imágenes. Se necesita cierto comportamiento inteligente para realizar estas funciones.
3. **Procesado de alto nivel:** comprende el reconocimiento y la interpretación de imágenes. Estos dos procesos tienen un gran parecido con lo que se conoce como *conocimiento inteligente*.

Hay diferentes tipos de métodos para el reconocimiento y la interpretación:

1. **Métodos de decisión teórica:** se basan en la representación de patrones en forma vectorial, y en la búsqueda posterior de aproximaciones que permitan agrupar y asignar estos patrones vectoriales a las diferentes clases de patrones.
2. **Métodos estructurales:** los patrones se representan en forma simbólica (cadena, árbol, etc.), y los métodos de reconocimiento se basan en el emparejamiento de símbolos o en modelos que tratan a los patrones de símbolos como secuencias de un lenguaje artificial.
3. **La interpretación:** consiste en asignar un significado a un conjunto de elementos reconocidos de la imagen.

1.9.2. Patrones y clases de patrones

Un *patrón* es una descripción estructural o cuantitativa de un objeto o de alguna otra entidad de interés en una imagen, y está formada por uno o más descriptores (llamados características).

Una *clase de patrón* es una familia de patrones que comparten algunas propiedades comunes, y se representan por w_1, \dots, w_M , donde M es el número de clases.

Las tres representaciones de patrones principalmente utilizadas en la práctica son los vectores (para descripciones cuantitativas), y las cadenas y los árboles (para descripciones estructurales). Los patrones vectoriales se representan utilizando letras minúsculas en negrita, como **x,y,z**, y tienen la forma:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

donde cada componente x_i representa el i -ésimo descriptor, y n es el número de descriptores. Los patrones vectoriales se representan como columnas. El concepto clave a tener en cuenta es que la selección de la(s) medida(s) utilizada(s) para obtener cada componente de un patrón vectorial tiene una profunda influencia en el rendimiento final de análisis de imágenes, ya que

de esta selección depende el grado de separabilidad entre clases. Al seleccionar el tipo de medidas, un patrón vectorial pasa a ser la representación completa de cada ejemplo físico.

En algunas aplicaciones, las características de los patrones se describen mejor utilizando relaciones estructurales. En estos casos en que las medidas cuantitativas acerca de cada característica junto con las relaciones espaciales entre dichas características determinan la pertenencia a cada clase, generalmente se resuelven mejor por técnicas estructurales. Las representaciones en forma de cadena generan adecuadamente patrones de objetos y otras entidades cuya estructura se basa en conexiones relativamente sencillas de primitivos, normalmente asociadas a formas de bordes o contornos. Una técnica más potente para muchas aplicaciones consiste en la utilización de descriptores en forma de árbol.

1.9.3. Métodos de decisión teórica

Se basan en la utilización de *funciones de decisión* (o *discriminantes*), las cuales permiten decidir a que clase de patrones (w_1, \dots, w_M) pertenece un patrón \mathbf{x} . Para lo cual tendremos M funciones ($d_1(\mathbf{x}), \dots, d_M(\mathbf{x})$) que tengan la siguiente propiedad:

- Si un patrón \mathbf{x} pertenece a la clase w_i , entonces $d_i(\mathbf{x}) > d_j(\mathbf{x}) \quad j = 1, \dots, M; i \neq j$.

En caso de empate en el valor de dos o más funciones (y que no haya otro con valor mayor), se resuelve arbitrariamente.

La *frontera de decisión* que separa la clase w_i de la clase w_j viene dada por valores de \mathbf{x} tales que: $d_i(\mathbf{x}) = d_j(\mathbf{x})$. En la práctica se utiliza una función $d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$ para representar la frontera, tal que:

- Si $d_{ij}(\mathbf{x}) > 0$ entonces $\mathbf{x} \in w_i$.
- Si $d_{ij}(\mathbf{x}) < 0$ entonces $\mathbf{x} \in w_j$.

1.9.3.1 Clasificador de mínima distancia: Supongamos que cada clase de patrones está representada por un *vector prototípico* (o *medio*):

$$m_j = \left(\frac{1}{N_j} \right) \sum_{\mathbf{x} \in w_j} \mathbf{x} \quad j = 1, \dots, M$$

donde N_j es el número de patrones vectoriales de la clase w_j . Una forma de determinar la pertenencia a una clase de un patrón vectorial desconocido \mathbf{x} consiste en asignarlo a la clase del prototípico más próximo. Si se usa la distancia euclídea para determinar la distancia, tenemos que: $D_j(\mathbf{x}) = \| \mathbf{x} - m_j \|$, $j = 1, \dots, M$, donde $\| \mathbf{a} \| = (\mathbf{a}^T \mathbf{a})^{1/2}$ es la norma de euclídes. Si $D_j(\mathbf{x})$ es la menor distancia, entonces se asigna la \mathbf{x} a la clase w_j .

1.9.3.2. Clasificadores estadísticos óptimos

1.9.3.2.1 Fundamentos: Es posible obtener una técnica de clasificación óptima en el sentido de que, por término medio, su empleo tenga asociada la probabilidad más baja de cometer errores de clasificación.

La probabilidad de que un patrón concreto \mathbf{x} , provenga de una clase w_i se representa como $p(w_i|\mathbf{x})$. Si el clasificador decide que \mathbf{x} proviene de w_j cuando en realidad proviene de w_i , incurre en un error L_{ij} . Como el patrón puede provenir de una de las M clases, el error medio que se produce al asignar \mathbf{x} a la clase w_j es:

$$r_j(\mathbf{x}) = \sum_{k=1}^M L_{kj} p(w_k|\mathbf{x})$$

Esta ecuación se suele llamar *riesgo medio condicional* o *perdidas*. Lo cual se puede ver como:

$$r_j(\mathbf{x}) = \sum_{k=1}^M L_{kj} p(\mathbf{x}/w_k) P(w_k)$$

donde $p(\mathbf{x}/w_k)$ es la función densidad de probabilidad de los patrones de la clase w_k , y $P(w_k)$ es la probabilidad de ocurrencia de la clase w_k .

El clasificador tiene M clases posibles para elegir a la hora de clasificar un patrón desconocido dado. Se calculan los errores medios $r_i(\mathbf{x})$ para $i=1,\dots,M$, y se asigna \mathbf{x} a la clase w_i tal que $r_i(\mathbf{x}) < r_j(\mathbf{x})$ $i,j=1,\dots,M$, y $i \neq j$. Este clasificador se llama *clasificador de Bayer*.

En muchos casos, el error de una decisión correcta es 0, y para una incorrecta 1. De esta forma, el error medio se puede ver como: $r_j(\mathbf{x}) = p(\mathbf{x}) - p(\mathbf{x}/w_j) P(w_j)$, como $p(\mathbf{x})$ es igual para todos los $r_i(\mathbf{x})$, entonces, se podría decir que el clasificador de Bayer asigna un patrón \mathbf{x} a la clase w_i si: $d_i(\mathbf{x}) > d_j()$ $i,j = 1,\dots,M$, $i \neq j$, siendo $d_i(\mathbf{x}) = p(\mathbf{x}/w_i) P(w_i)$ para $i = 1,\dots,M$.

Si todas las clases son equiprobables, $P(w_j) = 1/M$, para $i = 1,\dots,M$, en caso contrario dicha probabilidad se puede inferir del problema.

Para el caso de $p(\mathbf{x}/w_j)$, es más difícil de calcular o estimar. Generalmente se supone que tiene la forma de la distribución *Gaussiana*. Cuando más se aproxime a la realidad esta suposición, más se aproximara el clasificador de Bayer al mínimo error medio de clasificación.

1.9.3.2.2 El clasificador de Bayer para patrones gaussianas: Para esta distribución, tenemos que:

$$p(\mathbf{x}/w_j) = \{1/[(2\pi)^{n/2} |C_j|^{1/2}]\} \exp[-(1/2)(\mathbf{x} - m_j)^T C_j^{-1} (\mathbf{x} - m_j)]$$

donde m_j es el vector medio, y C_j la matriz de covarianza:

$m_j = E_j\{\mathbf{x}\}$ $C_j = E_j\{(\mathbf{x} - m_j)(\mathbf{x} - m_j)^T\}$ donde $E_j\{..\}$ representa el valor esperado del argumento para los patrones de la clase w_j , $|C_j|$ es el determinante de C_j , y n es la dimensión de los patrones vectoriales.

Al aproximar el valor esperado E_j por el valor medio de las cantidades en cuestión, se obtiene una estimación para:

$$m_j = (1/N_j) \sum_{\mathbf{x} \in w_j} \mathbf{x}$$

y para

$$C_j = (1/N_j) \sum_{\mathbf{x} \in w_j} \mathbf{x} \mathbf{x}^T - m_j m_j^T$$

donde N_j es el número de patrones vectoriales de la clase w_j .

Como la densidad Gaussiana tiene forma exponencial, conviene trabajar con el logaritmo neperiano de la función de densidad. Es decir:

$$d_j(\mathbf{x}) = \ln [p(\mathbf{x}/w_j) P(w_j)] = \ln [p(\mathbf{x}/w_j)] + \ln [P(w_j)]$$

Con esto llego a que:

$$d_j(\mathbf{x}) = \ln [P(w_j)] - (1/2) \ln |C_j| - (1/2) [(\mathbf{x} - m_j)^T C_j^{-1} (\mathbf{x} - m_j)] \quad \text{para } j = 1, \dots, M$$

será la función de decisión de Bayer para clases de patrones gaussianas que tienen una función error 0-1.

El clasificador de mínima distancia es óptimo, según Bayer, si:

1. Las clases de patrones son gaussianas.
2. Todas las matrices de covarianza son iguales a la matriz identidad.
3. Todas las clases son equiprobables.

En este caso la función de decisión será:

$$d_j(\mathbf{x}) = (\mathbf{x}^T m_j) - (1/2) (m_j^T m_j) \quad \text{para } j = 1, \dots, M.$$

1.9.4. Métodos estructurales

Este tipo de métodos tratan de lograr el reconocimiento del patrón poniendo énfasis en las relaciones estructurales inherentes a la forma de dichos patrones.

1.9.4.1. Correspondencia de números de forma

Se puede formular un procedimiento análogo al concepto de la mínima distancia para la comparación de contornos que se describen mediante números de forma.

El *grado de similitud*, k , entre dos contornos (formas) A y B, se define como el mayor orden para el que aún coinciden sus correspondientes números de forma.

Se define la *distancia* entre dos formas A y B como la inversa del grado de similitud:

$$D(A,B) = 1 / k$$

Esta distancia tiene las siguientes propiedades:

1. $D(A,B) \geq 0$.
2. $D(A,B) = 0$ si $A = B$.
3. $D(A,C) \leq \max[D(A,B), D(B,C)]$.

Tanto k como D pueden ser utilizados para comparar dos formas. En el primer caso, cuando mayor es k, mayor será la similitud. En el segundo, cuando menor es D, mayor será la similitud.

1.9.4.2. Correspondencia de cadena

Supongamos que se codifican dos contornos A y B en las cadenas representadas por a_1, a_2, \dots, a_n y b_1, b_2, \dots, b_m respectivamente. Supóngase que M es el número de correspondencias entre ambas cadenas, teniendo en cuenta que se produce una correspondencia en la k-esima posición si $a_k = b_k$. El número de símbolos que no se corresponden es: $Q = \max(|A|, |B|) - M$, donde $|\arg|$ es la longitud o número de símbolos de la cadena arg.

Una sencilla medida de similitud entre A y B es el cociente $R = M / Q$.

Para determinar la similitud entre A y B, se debe rotar una de las cadenas (tratándola como una cadena circular), tomando cada elemento de dicha cadena como inicio, sacando el R correspondiente, y luego quedarse con el mayor de los valores de R como grado de similitud. Para evitar esto, se podría tener algún método para determinar en ambas cadenas el mismo punto de partida.

1.9.4.3. Métodos sintácticos

La idea consiste en la especificación de un conjunto de *primitivas* de patrones, un conjunto de reglas (en forma de *gramática*) que gobiernan su interrelación, y un *reconocedor* (denominado *autómata*) cuya estructura está determinada por el conjunto de reglas que impone la gramática para reconocer si la cadena pertenece a la gramática

1.9.5. Interpretación

1.9.5.1. Fundamentos

La interpretación de imágenes por medio de una computadora es un proceso muy complejo, para el cual debemos:

1. **Limitar la generalidad del problema:** se debe limitar lo más que se pueda las circunstancias desconocidas para simplificar el problema.

2. **Intentar incorporar el conocimiento humano al proceso:** para esto se requiere la elección de un formalismo para expresar el conocimiento. Las tres técnicas más importantes son:
 - a) La Lógica Formal (generalmente se basan en el cálculo de predicado de primer orden).
 - b) Redes Semánticas (grafos dirigidos con leyendas).
 - c) Sistemas de Producción (o sistemas basados en reglas).

1.9.5.2. Tipos de conocimiento

Al conocimiento se lo puede clasificar como:

1. **Conocimiento procedimental:** Se refiere a operaciones tales como la selección de algoritmos y el ajuste de los parámetros de los mismos (como elegir los valores de umbral).
2. **Conocimiento visual:** Trata de los distintos aspectos de la formación de imágenes.
3. **Conocimiento universal:** Indica el conocimiento global en el ámbito de un problema.

Los dos primeros se utilizan más para los niveles de procesado bajo e intermedio, y el tercero para el nivel alto.

Independientemente del tipo de conocimiento, se debe tratar que la representación del conocimiento sea lo más independiente posible de la aplicación.

Capítulo 2 – Sistemas Biométricos

Introducción

Entenderemos por *sistema biométrico* a un sistema automatizado que realiza labores de biometría. Es decir, un sistema que fundamenta sus decisiones de reconocimiento mediante una característica personal que puede ser reconocida o verificada de manera automatizada. En esta sección son descritas algunas de las características más importantes de estos sistemas.

2.1. 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:

1. *Conocimiento*: la persona tiene conocimiento (por ejemplo: un código),
2. *Posesión*: la persona posee un objeto (por ejemplo: una tarjeta), y
3. *Característica*: la persona tiene una característica que puede ser verificada (por ejemplo: una de sus huellas dactilares).

Cada uno de los indicadores anteriores genera una estrategia básica para el proceso de identificación personal. Además pueden ser combinados con el objeto de alcanzar grados de seguridad más elevados y brindar, de esta forma, diferentes niveles de protección. Distintas situaciones requerirán diferentes soluciones para la labor de identificación personal. Por ejemplo, con relación al *grado de seguridad*, se debe considerar el valor que está siendo protegido así como los diversos tipos de amenazas. También es importante considerar la reacción de los usuarios y el costo del proceso.

2.2. Características de un indicador biométrico

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

1. *Universalidad*: cualquier persona posee esa característica;
2. *Unicidad*: la existencia de dos personas con una característica idéntica tiene una probabilidad muy pequeña;
3. *Permanencia*: la característica no cambia en el tiempo; y
4. *Cuantificación*: la característica puede ser medida en forma cuantitativa.

Los requerimientos anteriores sirven como criterio para descartar o aprobar a alguna característica como *indicador biométrico*. Luego de seleccionar algún indicador que satisfaga los requerimientos antes señalados, es necesario imponer restricciones prácticas sobre el sistema que tendrá como misión recibir y procesar a estos indicadores. En la siguiente sección se presentan estas restricciones.

2.3. Características de un sistema biométrico para identificación personal

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:

1. *El desempeño*, que 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. El objetivo de esta restricción es comprobar si el sistema posee una exactitud y rapidez aceptable con un requerimiento de recursos razonable.
2. *La aceptabilidad*, que 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. Factores psicológicos pueden afectar esta última característica. Por ejemplo, el reconocimiento de una retina, que requiere un contacto cercano de la persona con el dispositivo de reconocimiento, puede desconcertar a ciertos individuos debido al hecho de tener su ojo sin protección frente a un "aparato". Sin embargo, las características anteriores están subordinadas a la aplicación específica. En efecto, para algunas aplicaciones el efecto psicológico de utilizar un sistema basado en el reconocimiento de características oculares será positivo, debido a que este método es eficaz implicando mayor seguridad.
3. *La fiabilidad*, que refleja cuán difícil es burlar al sistema. El sistema biométrico debe reconocer características de una persona viva, pues es posible crear dedos de látex, grabaciones digitales de voz prótesis de ojos, etc. Algunos sistemas incorporan métodos para determinar si la característica bajo estudio corresponde o no a la de una persona viva. Los métodos empleados son ingeniosos y usualmente más simples de lo que uno podría imaginar. Por ejemplo, un sistema basado en el reconocimiento del iris revisa patrones característicos en las manchas de éste, un sistema infrarrojo para chequear las venas de la

mano detecta flujos de sangre caliente y lectores de ultrasonido para huellas dactilares revisan estructuras subcutáneas de los dedos.

2.4. Arquitectura de un sistema biométrico para identificación personal

Los dispositivos biométricos poseen tres componentes básicos. El primero se encarga de la adquisición análoga o digital de algún indicador biométrico de una persona, como por ejemplo, la adquisición de la imagen de una huella dactilar mediante un escáner. El segundo maneja la compresión, procesamiento, almacenamiento y comparación de los datos adquiridos (en el ejemplo una imagen) con los datos almacenados. El tercer componente establece una interfaz con aplicaciones ubicadas en el mismo u otro sistema. La arquitectura típica de un sistema biométrico se presenta en la figura 1. Esta puede entenderse conceptualmente como dos módulos:

1. *Módulo de inscripción (enrollment module)* y
2. *Módulo de identificación (identification module)*.

El módulo de inscripción se encarga de adquirir y almacenar la información proveniente del indicador biométrico con el objeto de poder contrastar a ésta con la proporcionada en ingresos posteriores al sistema. Las labores ejecutadas por el módulo de inscripción son posibles gracias a la acción del lector biométrico y del extractor de características.

El primero se encarga de adquirir datos relativos al indicador biométrico elegido y entregar una representación en formato digital de éste. El segundo extrae, a partir de la salida del lector, características representativas del indicador. El conjunto de características anterior, que será almacenado en una base de datos central u otro medio como una tarjeta magnética, recibirá el nombre de template. En otras palabras un template es la información representativa del indicador biométrico que se encuentra almacenada y que será utilizada en las labores de identificación al ser comparada con la información proveniente del indicador biométrico en el punto de acceso.

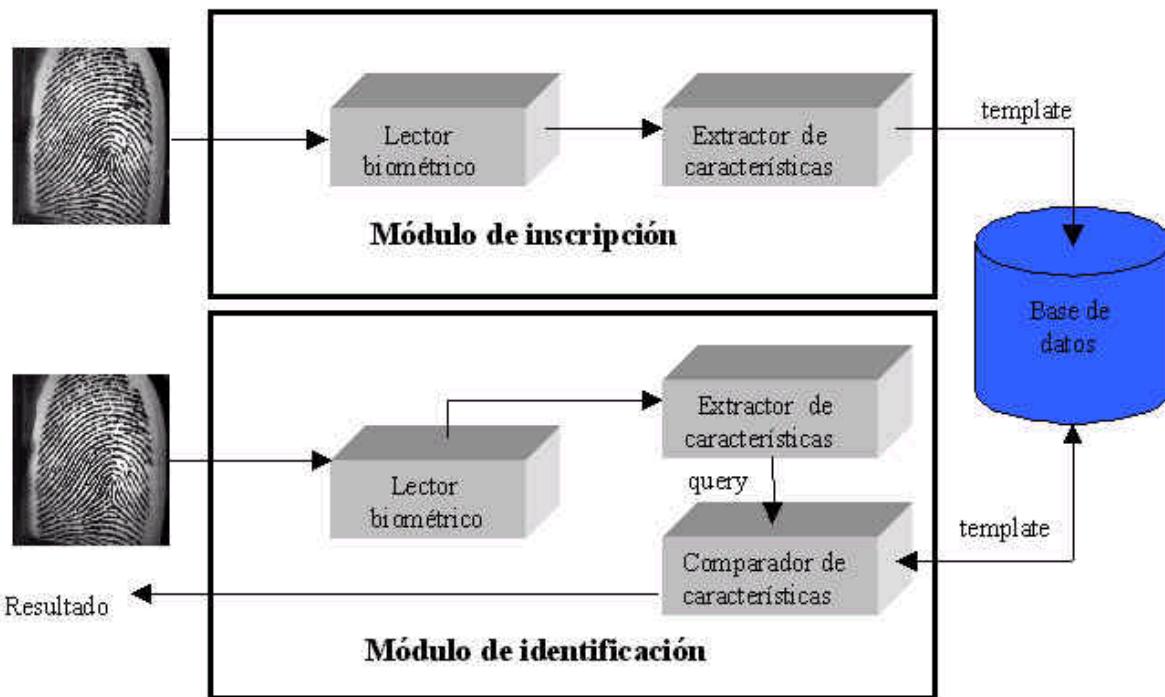


Figura 1: Arquitectura de un sistema biométrico para identificación personal, aquí ejemplificado con huellas dactilares.

El módulo de identificación es el responsable del reconocimiento de individuos, por ejemplo en una aplicación de control de acceso. El proceso de identificación comienza cuando el lector biométrico captura la característica del individuo a ser identificado y la convierte a formato digital, para que a continuación el extractor de características produzca una representación compacta con el mismo formato de los *templates*. La representación resultante se denomina *query* y es enviada al comparador de *características* que confronta a éste con uno o varios *templates* para establecer la identidad.

El conjunto de procesos realizados por el módulo de inscripción recibe el nombre de *fase de inscripción*, mientras que los procesos realizados por el módulo de identificación reciben la denominación de *fase operacional*. A continuación se entregan detalles de esta última.

2.5. Fase operacional de un sistema de identificación personal.

Un sistema biométrico en su fase operacional puede operar en dos modos:

1. *Modo de verificación*, o
2. *Modo de identificación*

Un sistema biométrico operando en el modo de verificación comprueba la identidad de algún individuo comparando la característica sólo con los templates del individuo. Por ejemplo, si una

persona ingresa su nombre de usuario entonces no será necesario revisar toda la base de datos buscando el template que más se asemeje al de él, sino que bastará con comparar la información de entrada sólo con el template que está asociado al usuario. Esto conduce a una comparación uno-a-uno para determinar si la identidad reclamada por el individuo es verdadera o no. De manera más sencilla el modo de verificación responde a la pregunta: ¿eres tú quién dices ser?.

Un sistema biométrico operando en el modo de identificación descubre a un individuo mediante una búsqueda *exhaustiva* en la base de datos con los templates. Esto conduce a una comparación del tipo *uno-a-muchos* para establecer la identidad del individuo. En términos sencillos el sistema responde la pregunta: ¿quién eres tú?.

Generalmente es más difícil diseñar un sistema de identificación que uno de verificación. En ambos casos es importante la exactitud de la respuesta. Sin embargo, para un sistema de identificación la rapidez también es un factor crítico. Un sistema de identificación necesita explorar toda la base de datos donde se almacenan los templates, a diferencia de un sistema verificador.

De la discusión anterior resulta obvio notar que la exigencia sobre el extracto y el comparador de características es mucho mayor en el primer caso.

2.6. Exactitud en la identificación: medidas de desempeño

La información provista por los templates permite particionar su base de datos de acuerdo a la presencia o no de ciertos patrones particulares para cada indicador biométrico. Las "clases" así generadas permiten reducir el rango de búsqueda de algún template en la base de datos. Sin embargo, los templates pertenecientes a una misma clase también presentarán diferencias conocidas como *variaciones intraclase*. Las variaciones intraclase implican que la identidad de una persona puede ser establecida sólo con un cierto nivel de confianza. Una decisión tomada por un sistema biométrico distingue "personal autorizado" o "impostor". Para cada tipo de decisión, existen dos posibles salidas, verdadero o falso. Por lo tanto existe un total de cuatro posibles respuestas del sistema:

1. Una persona autorizada es aceptada,
2. Una persona autorizada es rechazada,
3. Un impostor es rechazado,
4. Un impostor es aceptado.

Las salidas números 1 y 3 son correctas, mientras que las números 2 y 4 no lo son. El grado de confidencia asociado a las diferentes decisiones puede ser caracterizado por la distribución

estadística del número de personas autorizadas e impostores. En efecto, las estadísticas anteriores se utilizan para establecer dos tasas de errores.

1. Tasa de falsa aceptación (*FAR*: False Acceptance Rate), que se define como la frecuencia relativa con que un impostor es aceptado como un individuo autorizado,
2. Tasa de falso rechazo (*FRR*: False Rejection Rate), definida como la frecuencia relativa con que un individuo autorizado es rechazado como un impostor.

La *FAR* y la *FRR* son funciones del grado de seguridad deseado. En efecto, usualmente el resultado del proceso de identificación o verificación será un número real normalizado en el intervalo [0, 1], que indicará el "grado de parentesco" o correlación entre la característica biométrica proporcionada por el usuario y la(s) almacenada(s) en la base de datos. Si, por ejemplo, para el ingreso a un recinto se exige un valor alto para el grado de parentesco (un valor cercano a 1), entonces pocos impostores serán aceptados como personal autorizado y muchas personas autorizadas serán rechazadas. Por otro lado, si el grado de parentesco requerido para permitir el acceso al recinto es pequeño, una fracción pequeña del personal autorizado será rechazada, mientras que un número mayor de impostores será aceptado. El ejemplo anterior muestra que la *FAR* y la *FRR* están íntimamente relacionadas, de hecho son duales una de la otra: una *FRR* pequeña usualmente entrega una *FAR* alta, y viceversa, como muestra la figura 2.

El grado de seguridad deseado se define mediante el umbral de aceptación u , un número real perteneciente al intervalo [0,1] que indica el mínimo grado de parentesco permitido para autorizar el acceso del individuo.

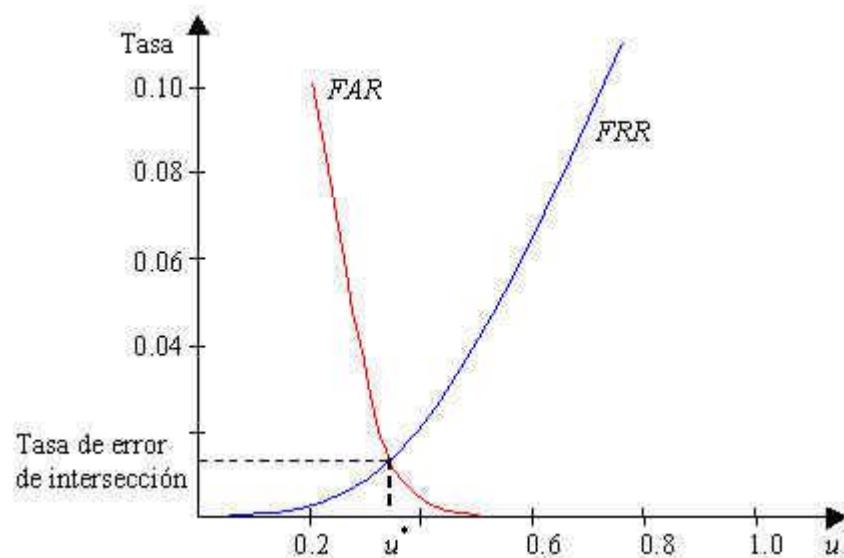


Figura 2. Gráfica típica de la tasa de falso rechazo (FRR) y la de falsa aceptación (FAR) como funciones del umbral de aceptación u para un sistema biométrico.

La FRR es una función estrictamente creciente y la FAR una estrictamente decreciente en u . La FAR y la FRR al ser modeladas como función del umbral de aceptación tienen por dominio al intervalo real $[0,1]$, que es además su recorrido, puesto que representan frecuencias relativas. La figura 2 muestra una gráfica típica de la FRR y la FAR como funciones de u . En esta figura puede apreciarse un umbral de aceptación particular, denotado por u^* , donde la FRR y la FAR toman el mismo valor. Este valor recibe el nombre de *tasa de error de intersección (cross-over error rate)* y puede ser utilizado como medida única para caracterizar el *grado de seguridad* de un sistema biométrico. En la práctica, sin embargo, es usual expresar los requerimientos de desempeño del sistema, tanto para verificación como para identificación, mediante la FAR. Usualmente se elige un umbral de aceptación por debajo de u^* con el objeto de reducir la FAR, en desmedro del aumento de la FRR.

2.7. Sistemas biométricos actuales.

En la actualidad existen sistemas biométricos que basan su acción en el reconocimiento de diversas características, como puede apreciarse en la figura 3. Las técnicas biométricas más conocidas son nueve y están basadas en los siguientes indicadores biométricos:

1. Rostro,
2. Termograma del rostro,
3. Huellas dactilares,
4. Geometría de la mano,
5. Venas de las manos,
6. Iris,
7. Patrones de la retina,
8. Voz,
9. Firma.

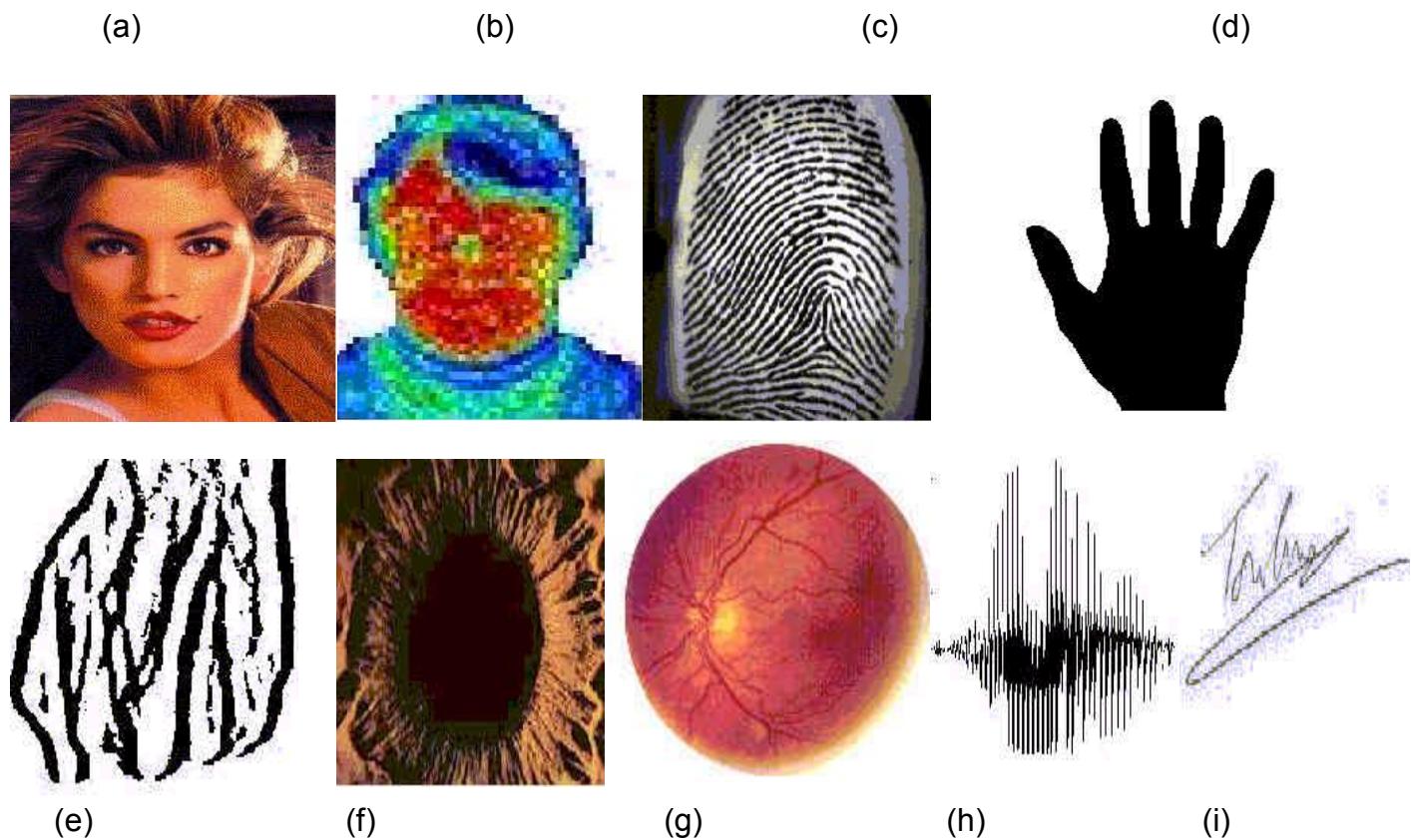


Figura 3. Técnicas biométricas actuales: (a) Rostro, (b) Termograma Facial, (c) Huella dactilar, (d) Geometría de la mano, (e) Venas de la mano, (f) Iris, (g) Patrones de la retina, (h) Voz e (i) Firma.

Cada una de las técnicas anteriores posee ventajas y desventajas comparativas, las cuales deben tenerse en consideración al momento de decidir que técnica utilizar para una aplicación específica. En particular deben considerarse las diferencias entre los métodos anatómicos y los de comportamiento. Una huella dactilar, salvo daño físico, es la misma día a día, a diferencia de una firma que puede ser influenciada tanto por factores controlables como por psicológicos no intencionales. También las máquinas que miden características físicas tienden a ser más grandes y costosas que las que detectan comportamientos. Debido a diferencias como las señaladas, no existe un único sistema biométrico que sea capaz de satisfacer todas las necesidades. Una compañía puede incluso decidir el uso de distintas técnicas en distintos ámbitos. Más aún, existen esquemas que utilizan de manera integrada más de una característica para la identificación. Por ejemplo, se integran el reconocimiento de rostros y huellas dactilares. La razón es que el reconocimiento de rostros es rápido pero no extremadamente confiable, mientras que la

identificación mediante huellas dactilares es confiable pero no eficiente en consultas a bases de datos. Lo anterior sugiere el utilizar el reconocimiento de rostros para particionar la base de datos.

Luego de esto comienza la identificación de la huella. Los resultados alcanzados por el sistema conjunto son mejores que los obtenidos por sus partes por separado. En efecto, las limitaciones de las alternativas por separado son soslayadas, logrando además respuestas exactas con un tiempo de proceso adecuado. En la figura 4 se presenta un esquema de división de las características biométricas.

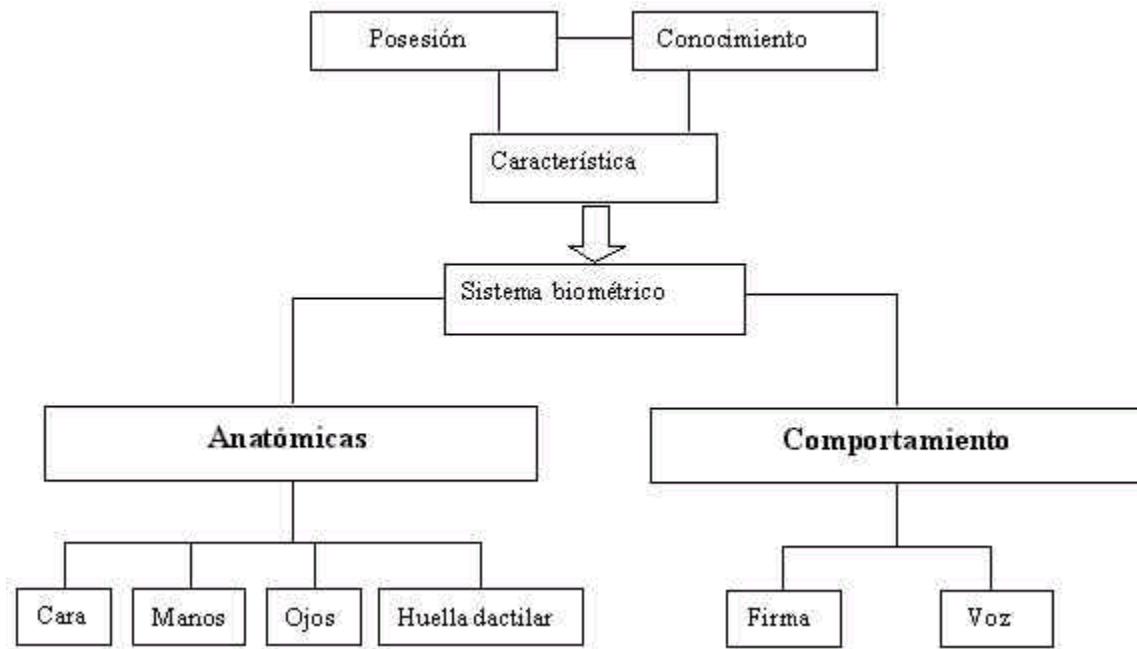


Figura 4. División de las características biométricas para identificación personal.

Capítulo 3 – Reconocimiento de Patrones

3.1. Introducción

El reconocimiento de patrones es utilizado actualmente para la solución de tareas tales como el reconocimiento de caracteres, de huellas digitales y reconocimiento del habla, entre muchos otros. Por esta razón, desde hace varios años se han venido desarrollando diferentes metodologías que intentan resolver este tipo de problemas.

3.2. Metodologías para el reconocimiento de patrones

De manera general, podemos decir que existen tres metodologías básicas para el reconocimiento de patrones: las heurísticas, las matemáticas y las lingüísticas.

3.2.1. Heurísticas

Esta metodología es la que hace uso de la experiencia y la intuición humana. Por lo general, los sistemas que son desarrollados bajo estos métodos, están hechos a la medida del problema que se desea resolver.

3.2.2. Matemáticas

Este tipo de metodologías hacen uso de las propiedades comunes de los patrones y se basan en las reglas de clasificación formuladas en un marco matemático. Además, este enfoque se divide a su vez en otras dos categorías: las determinísticas y las estadísticas. Entre los métodos determinísticos podemos encontrar, por ejemplo, los algoritmos de aprendizaje iterativo tales como los algoritmos utilizados para el entrenamiento de redes neuronales.

Por otra parte, dentro de los métodos estadísticos podemos encontrar a los clasificadores basados en las reglas de clasificación de Bayes.

3.2.3. Lingüísticas (sintácticas)

Por último encontramos las metodologías lingüísticas en las cuales se hace uso de los elementos primitivos que componen a los patrones (sub-patrones) y la relación que existe entre ellos.

3.3. Etapas para el reconocimiento de patrones

De manera general, cuando se intenta resolver algún problema que implique el reconocimiento de patrones se deben de tomar en cuenta las siguientes etapas:

- Adquisición de datos.
- Preprocesamiento.
- Extracción de características.
- Reconocimiento.

3.3.1. Adquisición de datos

Para poder realizar el reconocimiento es necesario, realizar o implementar alguna etapa de adquisición de los datos que describen al patrón que se desea clasificar. Esto implica que, el método que se utilizará para realizar la adquisición de los datos, dependerá de la naturaleza del patrón. Por ejemplo, si se desea desarrollar un sistema para reconocer caracteres, esta etapa se podría realizar con la ayuda de una cámara digital o utilizando un escáner.

3.3.2. Preprocesamiento

Una vez que ya se cuenta con los datos que describen a cada uno de los patrones, es más conveniente realizar una etapa de preprocesamiento sobre cada uno de ellos en lugar de ser dados como entrada del sistema tal y como fueron obtenidos durante la etapa de adquisición de datos.

El realizar un preprocesamiento sobre los datos tiene varias ventajas; la principal de ellas es que puede reducir la dimensionalidad de los datos, lo cual mejora substancialmente la ejecución del sistema, sobre todo cuando se utiliza una metodología como la de redes neuronales.

Dentro de esta etapa se puede preprocesar el patrón de entrada de tal forma que todos los patrones tengan el mismo tamaño (escala) consiguiendo con esto que el sistema sea invariante al escalamiento. Además de esto, también se busca lograr que el sistema sea invariante a la traslación. Cuando un sistema es invariante a la traslación y al escalamiento de los patrones, se dice que el sistema cuenta con un conocimiento previo.

3.3.3. Extracción de características

Uno de los principales problemas en el reconocimiento de patrones, es encontrar una manera óptima de representar la información original que describe a cada uno de los patrones. Este problema es conocido como extracción de características. Este proceso de extracción de características trata de reducir la cantidad de información (reducción de dimensionalidad) que representa a cada uno de los patrones, obteniendo de esta forma, un vector de características que represente de la mejor manera posible al patrón original.

La extracción de características debe de cumplir con las siguientes condiciones:

- La dimensionalidad del vector de características debe de ser menor que la del patrón original.
- Las características deben representar una codificación óptima de la entrada, perdiendo la información que no sea muy importante.

Especialmente, cuando se realiza el reconocimiento de patrones utilizando redes neuronales, la extracción de características debe tratar de obtener un vector de características con una dimensionalidad mucho menor a la del patrón original, puesto que, un vector con una dimensionalidad más pequeña que sea dado como entrada a la red neuronal tiene varios beneficios. En primer lugar, la cantidad de pesos que deben de ser aprendidos es menor, y en segundo lugar, al tener menos pesos, el tiempo de entrenamiento puede ser reducido considerablemente.

3.3.4. Reconocimiento

Finalmente, una vez que se han realizado las etapas de preprocesamiento y extracción de características, se procede ha realizar el objetivo principal del sistema: el reconocimiento. De manera general, la etapa de reconocimiento consiste en obtener una salida del sistema como respuesta a un conjunto de estímulos dados como entrada (patrones).

Capítulo 4 – Introducción a las Redes Neuronales

Las redes neuronales o modelos conexionistas como también se les conoce, 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 procesamiento llamados neurones artificiales, los cuales tienen conexiones de entrada (dendritas) que provienen de otros neurones y conexiones de salida (axón) hacia otros neurones. Para poder obtener un buen desempeño, los neurones de una red neuronal están masivamente interconectados con otros neurones mediante la serie de conexiones tanto de entrada como de salida, las cuales tienen un cierto peso.

De manera general las redes neuronales presentan tres principales características: aprendizaje, generalización, adaptabilidad. La característica de aprendizaje se refiere a que al igual que el cerebro humano, 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 los neurones que conforman a la red neuronal. Por otra parte, la capacidad de generalización significa que se pueden obtener salidas razonables cuando se usan entradas diferentes a las utilizadas durante el proceso de entrenamiento. Por último, la característica de adaptabilidad significa que una red neuronal puede ser reentrenada para funcionar adecuadamente ante cambios en su medio ambiente.

Existen 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. Una de las arquitecturas mas utilizadas para el reconocimiento de rostros es la red alimentada hacia delante multicapas o red de perceptrones multicapas (MLP).

Básicamente, 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. Este tipo de redes comúnmente son entrenadas utilizando el algoritmo de retro-propagación de error el cual consta de dos fases. La primera, es una fase hacia delante, en la cual la información colocada en los neurones 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 los neurones que la conforman, la respuesta a la información dada como entrada. Durante esta fase, todos los pesos de las conexiones entre neurones 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 los neurones de la capa de salida.

Capítulo 5 - Sistemas de Reconocimiento de Rostros

5.1. Introducción

Como se mencionó previamente, el reconocimiento de rostros es una área dentro del reconocimiento de patrones que ha sido investigada desde hace varios años. Desde entonces se han propuesto varias técnicas para resolver esta tarea, entre ellas podemos mencionar las siguientes: las técnicas que utilizan igualamiento de plantillas, cálculo de *eigenfaces*, características geométricas, y los métodos que utilizan redes neuronales, entre otros. Para la implementación de este tipo de sistemas, independientemente de la técnica o metodología que sea implementada, se utilizan generalmente dos conjuntos de datos. El primer conjunto de datos es el que es utilizado durante la etapa de aprendizaje, el cual es comúnmente llamado *conjunto de entrenamiento*. Se debe de tratar que los patrones que integran este conjunto, sean lo más diferente posible entre sí, y que además, representen al problema, para poder obtener un buen porcentaje de generalización. El segundo conjunto de patrones, es el que se utiliza durante la etapa de reconocimiento y es llamado *conjunto de prueba*.

En la mayoría de estas técnicas, el reconocimiento de rostros involucra (siguiendo los lineamientos del reconocimiento de patrones en general) alguna de las siguientes etapas:

- Adquisición de la imagen
- Preprocesamiento de la imagen
- Extracción de características,
- Alguna forma de entrenamiento,
- por último, la etapa de reconocimiento.

Adquisición de la imagen

Esta etapa, se define como concepto básico tanto del procesamiento de imágenes como del reconocimiento de patrones, por lo cual en este caso cabe se mencionan los medios de obtención de las imágenes de rostros más utilizados: videocámara, cámara digital o escaneado de fotografía entre otras.

Preprocesamiento de la imagen

Dado que las imágenes de los rostros regularmente son tomadas en momentos diferentes, esto trae como consecuencia que las diferentes imágenes de una misma persona tengan variaciones en cuanto a la iluminación, la orientación, y el tamaño del rostro. Por tal motivo, es necesario que la imagen sea preprocesada antes de que pueda ser utilizada. Entre

las tareas de preprocesamiento más comunes podemos encontrar las siguientes: extraer la imagen del rostro de una imagen más grande que contenga información irrelevante para el reconocimiento; normalización en cuanto al tamaño, es decir, que todas las imágenes de los rostros tengan un tamaño similar, y la aplicación de algún método de filtrado para mejorar la calidad de la imagen.

Extracción de Características

La extracción de características es una de las etapas de la cual depende en gran medida el buen desempeño del sistema de reconocimiento de rostros. El objetivo principal de esta etapa es extraer la información más discriminante de un rostro, eliminando la información que resulte irrelevante para el reconocimiento. En esta etapa podemos encontrar desde las técnicas que obtienen las características de manera manual, hasta técnicas mas sofisticadas que extraen las características de forma automática. En varias técnicas, las principales características que son extraídas de los rostros son aquellas que describen a cada uno de los componentes básicos del rostro tales como los ojos, la nariz, la boca, la barba, las cejas, y por supuesto, la relación que existe entre ellos, es decir, la distancia entre cada uno de estos componentes.

Entrenamiento

El entrenamiento básicamente consiste en utilizar alguna forma de aprendizaje que le permita al sistema “aprender” los rostros que constituyen el conjunto de entrenamiento. El tipo de entrenamiento que sea utilizado para el aprendizaje, dependerá en gran medida, de la metodología que se esté utilizando para el reconocimiento. Por ejemplo, al utilizar una metodología como la de redes neuronales, el entrenamiento consiste en obtener los valores correspondientes a cada una de las conexiones (pesos) que constituyen a la red neuronal. Para este tipo de metodologías existen diferentes algoritmos para realizar el entrenamiento, entre los cuales el algoritmo de retro-propagación es uno de los más populares.

Reconocimiento

La última etapa que comúnmente es realizada en un sistema de reconocimiento de rostros, es la etapa de reconocimiento. Esta etapa básicamente consiste en alimentar al sistema con imágenes de personas diferentes a las utilizadas durante el entrenamiento, esperando obtener como resultado, alguna forma de codificación que nos permita identificar de que persona se trata.

Cabe acotar que estas etapas parecerían seguir un orden secuencial de ejecución y tradicionalmente fueron tratadas de esta manera. Sin embargo, es más apropiado

considerarlas como un conjunto de módulos o tareas que cooperan entre si, tanto desde el punto de vista computacional como psicológico.

5.2. Investigaciones sobre reconocimiento de rostros

5.2.1. Historia del reconocimiento de rostros

El tema de reconocimiento de rostros es tan viejo como la visión por computadora, tanto por la importancia del tema, como por el interés teórico de los científicos cognitivos. Pese al hecho que otros métodos de identificación (tales como las huellas digitales, o búsqueda de iris) pueden ser más precisos, el reconocimiento de rostros ha tenido siempre un mayor enfoque de las investigaciones, debido a su naturaleza no-invasiva y debido a que es el método primario de las personas para identificar a otras.

Possiblemente el más famoso y uno de los primeros ejemplos, en el área es el desarrollado por Kohonen, quien demostró que una simple red neuronal podía realizar reconocimiento de rostros para imágenes de rostros alineadas y normalizadas. El tipo de red que el utilizó computaba una descripción de rostro, aproximando los eigen vectors de la matriz de autocorrelación de la imagen del rostro; estos eigen vectors son actualmente conocidos como 'eigenfaces.'

El sistema de Kohonen no fue un éxito en la práctica, sin embargo, tal vez por la necesidad de un alineamiento y normalización precisos. En los años subsecuentes muchos investigadores probaron esquemas de reconocimiento de rostros, basados en bordes, distancia entre rasgos y otros acercamientos sobre redes neuronales. Mientras muchos fueron exitosos sobre pequeñas bases de datos de rostros alineados, ninguno pudo resolver el problema más realista de grandes bases de datos donde la ubicación y escala del rostro es desconocido.

Kirby y Sirovich (1989) más tarde introducen una manipulación algebraica que hace fácil el calcular directamente los eigenfaces, y muestran que menos de 100 son requeridos para codificar con precisión imágenes alineadas y normalizadas. Turk y Pentland (1991) luego demostraron que el error residual de codificar usando los eigenfaces puede ser usado tanto para detectar rostros en un espacio natural, como para determinar la ubicación precisa y escala del rostro en una imagen. Luego demostraron que juntando este método para detección y localización de rostros con el método de reconocimiento utilizando eigenfaces, uno puede conseguir seguramente, reconocimiento en tiempo real de los rostros en ambientes mínimamente restringidos. Esta demostración que técnicas de reconocimiento simples pueden ser combinadas para crear sistemas útiles, inició una explosión de interés en el área de reconocimiento de rostros. En los últimos 20 años, los investigadores han realizado grandes progresos y propuesto varias estrategias, la mayoría orientadas al reconocimiento de imágenes

de rostros frontales en pequeñas bases de datos, con determinadas restricciones. Mientras que recientemente, se ha prestado mucha atención a grandes bases de datos, con múltiples posiciones, reconocimiento de rostros en tiempo real, con menos enfoque a la imagen en sí.

Tanto en los 70s como en los 80s, los métodos tradicionales como feature-based y estrategias de matching de template-base eran muy populares, mientras que en los 90s se generaron métodos como transformation-based y ANN-base que reportaron una mejor performance.

5.2.2. Actuales

Enseguida se presentan algunos de los trabajos en el área de reconocimiento de rostros más relevantes de los últimos años. Roberto Brunelli y Tomaso Poggio, comparan dos estrategias sobre una base de datos común (imágenes frontales de los rostros de 47 personas; 26 masculinos y 21 femeninas, cuatro imágenes por persona). Ellos desarrollaron e implementaron dos nuevos algoritmos; el primero consiste en el cálculo de un conjunto de características geométricas, tales como ancho y alto de la nariz, posición de la boca y forma de la barba, y el segundo método está basado en el igualamiento de plantillas en niveles de gris. Ellos reportaron haber obtenido un 90% de reconocimiento usando características geométricas y un reconocimiento perfecto usando igualamiento de plantillas.

Meziane Yacoub y Younes Bennani proponen un método para la selección de características para detectar e identificar zonas discriminatorias en la capa de entrada de un modelo conexionista. El objetivo es seleccionar características simples que tengan un buen poder discriminatorio. El método que proponen es el ε HVS (HVS Extendido) que está basado en la medida HVS (Heurística para la Selección Variable) que ha mostrado resultado exitoso sobre Redes de Perceptrones Multicapa para la selección variable. Esta medida estima la contribución discriminatoria a la decisión final del clasificador de un peso dado. HVS permite identificar y seleccionar las variables más importantes en el espacio de características. Esto es logrado al eliminar las unidades redundantes y que tienen una baja contribución.

Ellos utilizaron un modelo conexionista convolucional llamado TDNN (de sus siglas en inglés Time Delay Neural Networks). Ellos utilizan una base de datos que consiste de 92 imágenes diferentes (4 personas diferentes, 23 imágenes por persona). La arquitectura del sistema que ellos utilizaron tiene dos módulos: el primero es un extractor de características con dos capas ocultas, con pesos compartidos, produciendo en la última capa oculta un mapa de características de la imagen presentada en la entrada, el segundo módulo es un clasificador que recibe como entrada el mapa de características obtenido anteriormente. La salida del clasificador es la identidad de la imagen presentada. La capa de salida tiene tantas unidades

como personas se deseé identificar. Ellos reportaron un porcentaje de reconocimiento del 100% sobre su base de datos.

Otro nuevo enfoque para el reconocimiento de rostros es el que proponen Oliver de Vel y Stefan Aeberhard. Aquí ellos clasifican las técnicas para el reconocimiento de rostros en dos tipos, al primero lo llaman reconocimiento basado en características geométricas, el cual consiste en obtener un conjunto de características del rostro tales como ancho de la nariz, forma de la barba, espesor de las cejas, etc. Por otro lado, al segundo tipo lo llaman reconocimiento basado en imágenes, en donde mencionan que esta metodología de reconocimiento tienen una mayor simplicidad comparada con las metodologías basadas en características geométricas. Dentro de este tipo de metodologías podemos encontrar enfoques como los basados en plantillas y los basados en redes neuronales. Pero debido a que estos métodos, por lo general necesitan bastante tiempo para su entrenamiento y han sido probados con bases de datos de rostros relativamente pequeñas, ellos proponen una tercera metodología, a la cual denominan: reconocimiento de rostros basado en líneas, la cual, tiene un buen porcentaje de reconocimiento, es tolerante a rotaciones, a variaciones en la escala y es invariante a la posición del rostro. Esta técnica consiste en representar a cada uno de los rostros como un conjunto de segmentos de líneas aleatorias obtenidas de la imagen de niveles de gris y utiliza un clasificador *nearest-neighbor*. Para la evaluación del desempeño de su sistema, utilizan la base de datos del “Olivetti & Oracle Research Laboratory” (ORL) y la base de datos del “University of Bern” (UB). Para realizar los experimentos, combinaron las dos bases de datos con el fin de obtener una base de datos más grande. Para la etapa de entrenamiento utilizaron 5 imágenes por persona y las otras 5 imágenes restantes las utilizaron para la etapa de reconocimiento. Como resultados de sus experimentos, reportan haber obtenido un porcentaje del 98% sobre la base de datos combinada, utilizando imágenes aleatorias para el entrenamiento y un porcentaje del 100%, utilizando un conjunto de imágenes seleccionadas para el entrenamiento.

5.3. Ventajas del Reconocimiento de Rostros

Los métodos más comunes de verificación e identificación en la actualidad son los sistemas de Password/PIN (Personal Identification Number). Debido a que estos sistemas tienen problema con falsificaciones, robos y lapsus de memoria por parte del usuario, se ha desarrollado un importante interés en los sistemas de identificación biométricos, que usan técnicas de reconocimiento de patrones, para identificar personas usando sus características fisiológicas. Las huellas digitales son un clásico ejemplo de biométrica; existen nuevas tecnologías que incluyen reconocimiento de retina e iris. Pese a que son interesantes para

transacciones bancarias o entrada a áreas de seguridad, estas tecnologías tienen la desventaja que son un poco molestas tanto físicamente como socialmente. Esto es así porque requieren que el usuario posicione su cuerpo relativo a algún tipo de sensor, y luego espere un segundo para ‘identificarse’ el mismo. Mas aún , existe un aspecto psicológico a la interacción: debido a que las personas no pueden reconocer otras personas usando estos sistemas, este tipo de identificación no tiene lugar en la interacción humana normal y las estructuras sociales.

Entonces, mientras estas cosas son muy útiles para aplicaciones de alta seguridad (hacen que los sistemas se vean más precisos), son exactamente lo opuesto a lo que se requiere cuando se construye una tienda que reconoce a sus mejores clientes, o un centro de información automatizado que recuerda a sus visitantes, o una casa ‘inteligente’ que conoce a las personas que viven en ella. El reconocimiento de rostros y el de voz tienen un lugar asegurado en estos ambientes inteligentes de siguiente generación. – son capaces de reconocer a la distancia sin requerir una interacción de ‘espera e identificación’, son usualmente pasivos (no requieren generación de iluminación electromagnética especial), no restringen el movimiento del usuario, y son tanto de bajo consumo como baratos. Probablemente, más importante aún, sin embargo, es el hecho de que las personas al poder identificar otras personas por su rostro y voz, por lo tanto se sienten más confortables en el uso de estos sistemas que usan reconocimiento de voz y/o rostros.

Las ventajas del reconocimiento de rostros son, entre otras, las siguientes :

- Proceso pasivo (no es necesario conocer o controlar el proceso de captura/comparación)
- Puede usarse en situaciones donde solamente se tienen fotos de personas
- Puede utilizar la ayuda humana: no se necesita un aprendizaje especial como en el uso de huellas digitales, geometría de la mano, etc.
- Una alta aceptación social por parte de la gente en general
- El Hardware más sofisticado puede consistir en una sencilla cámara de vídeo; barato

5.4. Aplicaciones

Un sistema biométrico basado en reconocimiento de rostros tiene un gran número de aplicaciones:

sistemas de seguridad, identificación criminal, procesamiento de imágenes y películas, interacción hombre-máquina. Desafortunadamente, el desarrollo de modelos computacionales de reconocimiento de rostros es una tarea dificultosa; esto es así debido a que todavía no conocemos como el cerebro humano reconoce los rostros.

El reconocimiento automático de rostros involucra la resolución de algunos problemas complejos:

- localización del rostro en escenarios complejos

- invarianza a posición e iluminación
- invarianza a cambios en las expresiones
- invarianza a bigotes, barba, corte de pelo ,anteojos, sombrero, etc.

El reconocimiento de rostros se usa también en verificaciones uno-a-uno, como en los siguientes ejemplos:

- Maquinas de cheques usan face recognition para verificar los depositantes de cheques.
- FaceVACS es un sistema seguro de depósitos que utiliza face recognition.
- El proyecto de frontera US/Mexico, SENTRI, usa el software Visionics face recognition en conjunto con verificación de voz.
- India Oil Co. usa Miros face recognition para una aplicación de tiempo y asistencia.
- Existen también aplicaciones end-user que reemplazan los login de Windows 95/98/NT con face recognition.

5.5. Tecnologías

Las tecnologías desarrolladas para el reconocimiento de rostros, son muchas y variadas, pero todas consisten en tres etapas básicas :

1. Segmentación of scenes from cluttered scenes
 - mapa de bordes, distintas heurísticas
2. Extracción de características de la región del rostro
 - características “holísticas” (donde cada rasgo es una característica del rostro en general)
 - características parciales (pelo, nariz, boca, ojos, etc.)
3. Decisión
 - identificación
 - reconocimiento de la persona
 - categorización

En general, una cámara es usada para adquirir la imagen del rostro, desde una distancia cercana. El sistema, luego analiza la geometría del rostro, tal como la distancia entre los ojos y la nariz.

La mayoría de los sistemas incluyen una función de localización de rostros que busca rostros en toda la imagen.

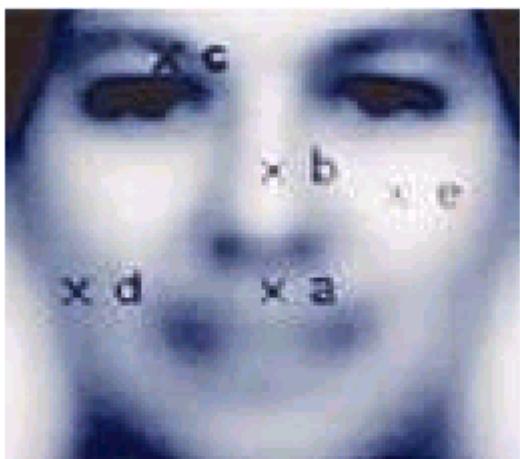
La descripción general del problema de Reconocimiento de Rostros (Face Recognition, FR) puede ser formulada como sigue:

Dada una imagen digitalizada obtenida de alguna forma, se debe identificar una o más personas, realizando el matching de el rostro en la imagen contra todo rostro conocido y almacenado en la base de datos de rostros. Existen, para ello, varias tareas diferentes:

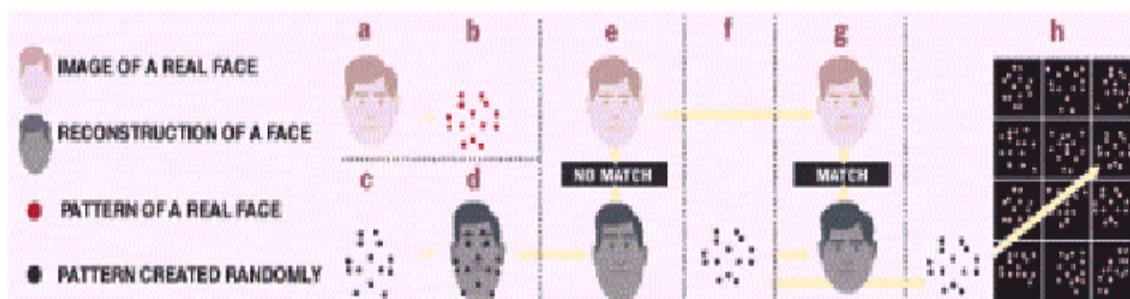
- detección de rostro y tracking
- extracción de características faciales, análisis de expresiones faciales
- categorización de rostros por raza, sexo o edad
- verificación de identidad y reconocimiento.

5.5.1. Análisis de Características Locales (Local Feature Analysis, LFA)

LFA se deriva del método eigenface pero supera alguno de sus problemas, al no ser sensible a deformaciones del rostro y a cambios de iluminación y poses. LFA considera características individuales en vez de contar con solo una representación global del rostro. El sistema selecciona una serie de bloques que definen mejor un rostro individual. Estas características con los bloques de construcción con los cuales todas las imágenes de rostro pueden ser construidas. El procedimiento comienza recolectando cada una de las fotografías de rostros y extrayendo de cada una de ellas su eigenface. Aplicando el LFA, el sistema selecciona el subconjunto de bloques de construcción(es decir características), en cada rostro que difiere más de los otros. Cualquier rostro seleccionado, puede ser identificado con al menos 32 o 50 de estos bloques. Los puntos más característicos (como se muestra a la izquierda) son la nariz, cejas, boca y las áreas donde la curvatura de los huesos cambia. Los patrones deben ser flexibles (“elásticos”) para poder describir posibles movimientos o cambios de expresión. La computadora conoce que esos puntos, como las ramas de un árbol en un día ventoso, se pueden mover ligeramente de un lado al otro del rostro, en combinación con los otros, sin perder la estructura básica que define ese rostro.



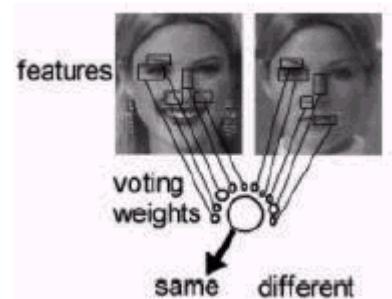
Para determinar la identidad de alguien, (a) la computadora toma una imagen de esa persona y (b) determina el patrón de puntos que hacen que ese individuo difiera lo más posible de la otra gente. Luego, el sistema comienza a crear patrones, (c) tanto al azar o (d) basándose en el promedio eigenface. Para cada selección, (e) la computadora construye una imagen de rostro y la compara con el rostro a ser identificado. (f) Nuevos patrones son creados hasta (g) que una imagen de rostro que matchea con el objetivo puede ser identificada. Cuando se encuentra un match, la computadora busca en su base de datos para realizar un matching pattern de una persona real (h), como se muestra a continuación.



5.5.2. Redes Neuronales

Según el reconocimiento de rostros ha sido desarrollado desde eigenfaces al análisis de características locales, tres temas de performance han permanecido: encontrar un rostro en imágenes con fondos complejos, acomodar condiciones de luminosidad variables y mejorar la precisión en el reconocimiento. La tecnología de Redes Neuronales [17] ha tratado con todos estos asuntos en el siguiente paso en la evolución de la tecnología de reconocimiento de rostros.

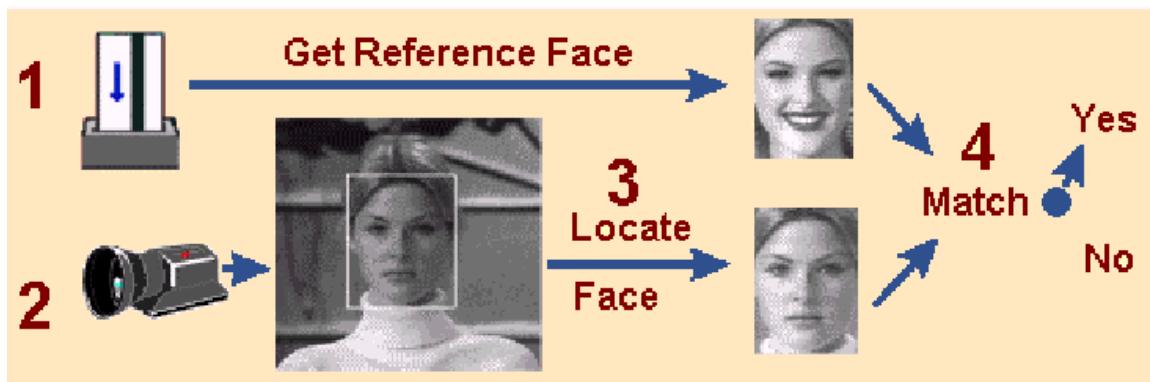
Tanto la imagen de un rostro o un instante de vídeo, incluyen uno o más rostros y un fondo. El fondo puede contener objetos o texturas variadas, así como otra gente, moviéndose o no. En medio de todos los posibles objetos en una imagen, el primer paso es aislar un rostro de esta imagen, siempre que exista (Face Isolation). Este paso es conseguido con una red neuronal, que se entrena para reconocer como se ve un rostro a través de diferentes estilos de pelo, condiciones de luminosidad, color de la piel y género así como variaciones en el tamaño del rostro, ubicaciones y orientaciones. Este entrenamiento de la red neuronal se realiza con cientos de miles de ejemplos de rostros. Para vídeo, un rostro puede ser seguido continuamente, realizando el proceso de face isolation por cada uno de los cuadros de vídeo.



Una vez que el rostro es aislado y localizado, facciones del rostro se extraen como elementos de contraste visual, tales como los ojos, lados de la nariz, boca, cejas, líneas de la mejilla y otras (Feature Extraction) como se muestra a la derecha.

Las características son cuantificadas, normalizadas y comprimidas en un template code. La combinación de face isolation y face encoding toma 1 segundo en un Pentium II.

El tamaño de un template code es de aproximadamente 1,300 bytes (Face Template). Para verificar a alguien, primeramente se realiza un requerimiento de identificar que persona debería ser, en un proceso mostrado más abajo. Este requerimiento es de la forma de un user ID que puede ser un PIN, código de barras, tarjeta, token, u otra biométrica. El user ID solicitado es utilizado para buscar el template que representa la verdadera identidad de la persona (reference template), que está almacenado en la base de datos. Luego, el programa matchea el reference template conocido contra el template del rostro a reconocer y produce un puntaje de matching. Si este puntaje, entre el template del rostro a analizar y el template almacenado es más grande que un umbral predefinido, los dos rostros se suponen como que provienen de la misma persona. Caso contrario, se supone que provienen de distintas personas.

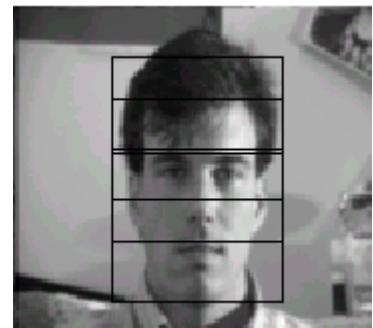


En la identificación de una persona desconocida contra la base de datos de personas, el template del rostro se compara contra todos los reference templates en la base de datos y luego los puntajes de matching son ordenados de mayor (el mejor) a menor puntaje. Si el puntaje mayor (o los puntajes mayores) está (están) por encima de un umbral de matching, entonces es el (son los) mejor(es) candidato(s) para identificar a la persona desconocida. El proceso de matching involucra una red neuronal en la cual las características de los dos rostros deciden si los rostros son similares o diferentes, como se mostró más arriba. Además, a cada decisión se le adjudica un peso que es ajustado en el proceso de aprendizaje. El aprendizaje se realiza por el método de prueba y error. Inicialmente, todos los pesos son iguales. Luego, la red neuronal se expone a pruebas, cada una con un par de rostros. Si la red neuronal produce la respuesta correcta en la decisión de si los rostros son los mismos o no, no hay lugar para el aprendizaje. Sin embargo, si la red neuronal produce una respuesta errónea, los pesos para las características son ajustados a la respuesta correcta. Repitiendo este proceso de aprendizaje sobre muchos ejemplos de rostros, la red neuronal automáticamente ajusta todos los pesos de las características, para producir la diferenciación más óptima entre pares de rostros similares y diferentes.

Debido a que las redes neuronales aprenden de la experiencia, realizan un mejor trabajo de acomodar variaciones en las condiciones de luminosidad y mejora la precisión sobre cualquier otro método. Después que la red neuronal ha sido entrenada, estará lista para su uso en aplicaciones de verificación e identificación.

5.5.3. Modelo Markov (Hidden Markov Model, HMM)

Fernando Samaria [5] [15] es quien describe e introduce recientemente, un framework basado en Hidden Markov Model (HMM) para reconocimiento de rostros y detección de rostros. HMM es un conjunto de modelos de estadística usados para caracterizar las propiedades estadísticas de una señal. HMM consiste de: (1) una oculta, inobservable cadena de Markov con un número finito de estados, una matriz de probabilidades de transición de estados y una distribución inicial de probabilidades de estados, y (2) un conjuntos de funciones de densidad de probabilidad asociados a cada estado.



Los vectores de observación que se utilizan para caracterizar los estados del HMM son obtenidos utilizando los coeficientes de la Transformada Karhunen-Loeve (KLT). El método de reconocimiento presentado, reduce significativamente la complejidad computacional de anteriores sistemas de reconocimiento de rostros basados en HMM, además de mejorar levemente la tasa de reconocimiento. Consistentemente con el modelo HMM del rostro, se introduce un acercamiento a la detección de rostros basándose en HMM, usando las mismas técnicas de extracción de características usadas para el reconocimiento de rostros.

El sistema esencial de reconocimiento de rostros basado en HMM se describe de la siguiente manera : Para un rostro frontal, las regiones del rostro significativas, tales como el pelo, frente, ojos, nariz y boca van apareciendo en un orden natural de arriba hacia abajo, aún si las imágenes experimentan pequeñas rotaciones en el plano de la imagen y/o rotaciones en el plano perpendicular al plano de la imagen.

Cada una de estas regiones del rostro se asignan a un estado, en un continuo HMM de una dimensión y de izquierda a derecha.

Comparando con los clásicos métodos basados en templates, los métodos basados en HMM ofrecen un framework más flexible para detección y reconocimiento, y pueden ser usados más eficientemente en sistema de escalas invariantes.

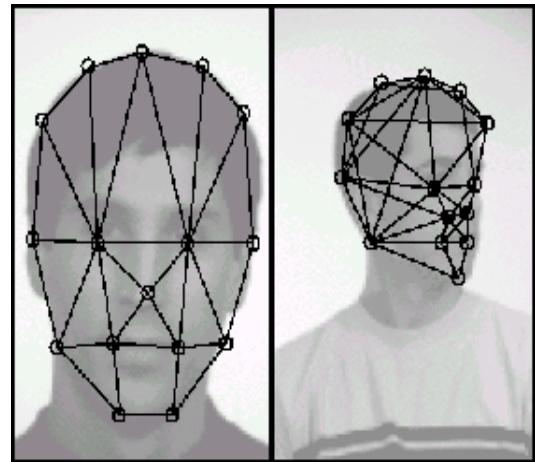
Las investigaciones realizadas sobre esta nueva técnica de reconocimiento de rostros basándose en HMM, la precisión de este método respecto a las variaciones en las condiciones de luminosidad y eficiencia, sugieren que es método muy prometedor.

Aun cuando hace pocos años ha visto la luz, el modelado HMM de rostros responde muy bien ante un amplio rango de orientaciones y expresiones de los rostros.

5.5.4. Elastic Bunch Graph Matching

El principal referente de este modelo es Laurenz Wiskott [7]. Se presenta un sistema para reconocimiento de rostros a partir de imágenes simples en una gran base de datos conteniendo solo una imagen por persona. La tarea es difícil debido a las variaciones de la imagen en términos de posición tamaño, expresión y poses. El sistema evita tomar la mayoría de estas variaciones, extrayendo descripciones concisas del rostro en la forma de grafo de la imagen. En este, los puntos fiducial en el rostro (ojos, boca, etc.) son descriptos como conjuntos de componentes wavelet (jets). La extracción por grafo de imagen esta basada en una nueva técnica, el grafo de grupo (bunch graph), que es construido a partir de un pequeño conjunto de muestra de grafos de imagen. El reconocimiento, luego, se basa en una comparación directa de los grafos de imagen.

Este sistema tiene un importante núcleo de estructura que toma en cuenta, el hecho que las imágenes de objetos coherentes tienden a trasladar, rotar y deformar el plano de la imagen. La representación básica de un objeto, es un grafo etiquetado; los arcos son etiquetados con información de distancia y los nodos son etiquetados con respuestas wavelet localmente empaquetados en jets. Los modelos de grafo almacenados, pueden ser matcheados con nuevas imágenes para generar grafos de imagen, que luego podrán ser incorporados a una galería y transformarse en grafos modelo. Como se usan los Wavelets, son robustas a cambios moderados de luminosidad, pequeños cambios y deformaciones. Los grafos modelo, pueden ser fácilmente trasladados, orientados o deformados, durante el proceso de matching, compensando de este modo los grandes cambios de las imágenes. Desafortunadamente, teniendo una galería con solo una imagen por persona, no se provee suficiente información para manejar rotación en profundidad.

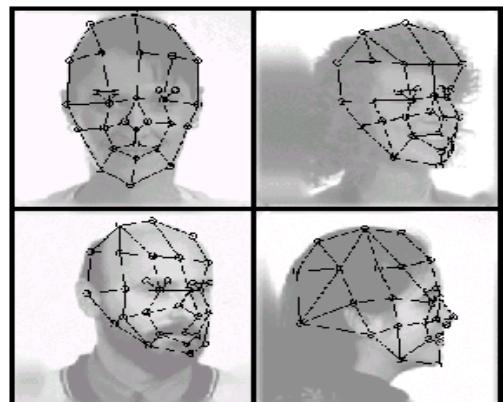


Esta estructura general, es muy útil para el manejo de cualquier tipo de objeto (no solo rostros, en principio) y puede ser suficiente para discriminar entre tipos de objetos estructuralmente diferentes. Sin embargo, para discriminar dentro de una clase de objetos, de lo cual el reconocimiento de rostros es un ejemplo, es necesario contar con información específica de una estructura común a todos los objetos en la clase. Esto es crucial para la extracción de esas características estructurales de la imagen que son importantes para la discriminación ("para saber donde mirar y a que hay que prestarle atención"). Es este tipo de sistema, información específica de la clase tiene la forma de bunch graphs, una por cada

posición, que son pilas de un número moderado (por ejemplo, 70 en los experimentos de Wiskott) de rostros diferentes, jet-sampled en un conjunto apropiado de puntos fiducial (localizados sobre ojos, boca, contorno, etc.). Los Bunch graphs son tratados como entidades combinatorias en las cuales, para cada punto fiducial, un jet de una muestra diferente del rostro, puede ser seleccionado, de modo de crear un modelo altamente adaptable.

Este modelo es matcheado con nuevas imágenes de rostros para encontrar con seguridad los puntos fiducial en la imagen. Se extraen los Jets y su posición relativa a estos puntos y son combinados en un grafo de imagen, una representación del rostro que no tiene restos de variación debido al tamaño, posición (u orientaciones en el plano).

Un bunch graph es creado en dos etapas. Su estructura cualitativa como grafo (un conjunto de nodos y arcos) además de la correspondiente asignación de etiquetas (jets y distancia) para una imagen inicial es hecha manualmente, mientras que el grueso bunch graph se extrae de manera semiautomática desde imágenes de muestra, realizando el matching entre ellas y los bunch graph.



Los grafos de imágenes son muy robustos a rotaciones pequeñas en profundidad del rostro. Los grandes ángulos de rotación, i.e. diferentes poses, son manejadas con la ayuda de bunch graphs con una estructura de grafo diferente, y las correspondencias entre nodos en diferentes poses realizadas de forma manual.

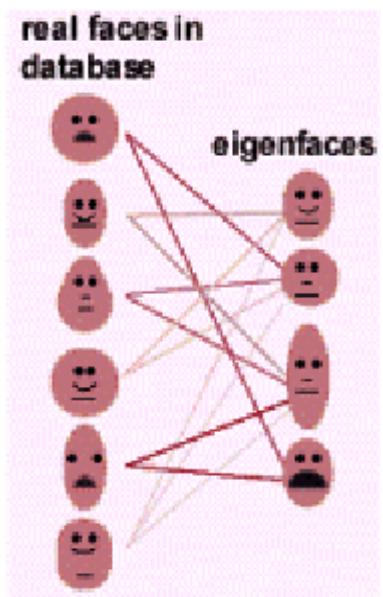
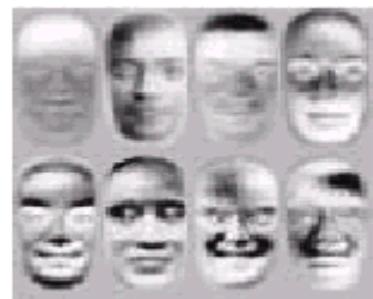
Después de esta preparación, el sistema puede extraer de imágenes simples descripciones concisas e invariantes del rostro en la forma de grafos de imagen (denominados grafos modelo cuando se almacenan en una galería). Estos contienen toda la información relevante para la tarea de discriminar el rostro. Para el reconocimiento en sí, los grafos de imagen pueden ser comparados con los grafos modelo a un costo pequeño de computación, evaluando únicamente la similitud de los jet.

En síntesis, este sistema se basa al máximo en una estructura general de información (grafos etiquetados con respuestas wavelet) y propiedades de transformación general. Estas son realizadas manualmente, pero debido a la generalidad y simplicidad que poseen, el esfuerzo es mínimo. Como se ha descripto, el sistema hace uso de estructuras de grafo para objetos específicos hechas “manualmente” y procedimiento de moderada intensidad de trabajo para generar los bunch-graphs.

Los investigadores están tratando de eliminar esta necesidad de intervención manual y del “trabajo adivinando”, con la ayuda de métodos de estimación estadísticos. Sin embargo, hay que recalcar que este sistema necesita solo un pequeño número de ejemplos para manejar la compleja tarea de reconocer un rostro.

5.5. Eigenfaces

Un gran número de imágenes de rostros se coleccionan en una base de datos. Luego, combinando todas las imágenes (tal como se muestra a la derecha), se crea un conjunto de eigenfaces, -arreglos de dos dimensiones de los rostros de áreas claras y áreas oscuras, y mirando a lo que es común a los grupos de individuos y donde más difieren. Los Eigenfaces trabajan como 'componentes de rostro'. Así como un color puede ser creado, mezclando los colores primarios, una imagen puede ser construida agregando conjuntamente, con diferentes intensidades de luz, un número de eigenfaces (cerca de 100 es suficiente para identificar la mayoría de las personas, pero solo para una posición y una condición de luminosidad relativa al rostro) relativamente pequeño. Para identificar un rostro, el programa compara las características de su eigenface, que son codificadas en números llamado template, con aquellos en la base de datos, seleccionando los rostros cuyos templates matchean el objetivo más cerca, como se muestra a la derecha.



Aspectos Matemáticos del Cálculo de EigenFaces

Para este algoritmo, se comienza asumiendo que existe un conjunto de rostros de entrenamiento de dimensión $N \times N$ cada uno :

$$\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_M,$$



Base de conocimiento de ejemplo

Como primer paso, estas imágenes de $\mathbf{N} \times \mathbf{N}$ son reordenadas como un vector de una dimensión $1 \times \mathbf{N}^2$ usando un ordenamiento por fila (es decir, tal cual un orden lexicográfico). El rostro promedio se define como :

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

y luego se le resta a cada una de las imágenes de entrenamiento, de forma que obtenemos un nuevo conjunto de vectores :

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

La idea principal es la de realizar un Análisis de componentes principales (PCA, principal component analysis) para obtener un conjunto de vectores ortogonales que representan óptimamente la distribución de la información.

Se realiza una descomposición SVD (single value decomposition) en la matriz de covarianza C de los vectores 'phi'

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = \frac{1}{M} A A^T,$$

donde

$$A = [\Phi_1 \quad \Phi_2 \quad \dots \quad \Phi_M].$$

La descomposición SVD esta dada por

$$C = U \Sigma V^T$$

donde las columnas de U, u^k conforman los eigenvectores (eigenfaces) del espacio de rostros (eigenspace).

Hay que tener en cuenta que al ser C una matriz de $N^2 \times N^2$, la descomposición SVD no es computable en tiempo y espacio de memoria. Sin embargo, es posible calcular la descomposición SVD de

$$L = \frac{1}{M} A^T A = U' \Sigma' V'^T$$

que es un sistema de $M \times M$. Luego, los eigenvectores u se calculan como una combinación lineal de las columnas de A y las columnas de la matriz V

$$\mathbf{u}_t = \sum_{k=1}^M v_{tk} \Phi_k.$$



Representación como una imagen de los vectores pertenecientes al eigenspace

El análisis PCA consiste, luego, en usar los mas significativos M' eigenfaces para crear un subespacio de rostros.

Un rostro puede ser aproximado como una combinación lineal de los eigenvectores del subespacio; esto es equivalente a decir que una cara se proyecta en el subespacio creado por los M' eigenvectores. Los coeficientes de proyección de cualquier rostro son fácilmente calculados por :

$$\omega_k = \mathbf{u}_k^T (\Gamma - \Psi) \quad k = 1, 2, \dots, M'$$

formando el vector

$$\Omega^T = [\omega_1 \quad \omega_2 \quad \dots \quad \omega_{M'}]$$

que es usado como una representación compacta de la imagen de entrada.



Proyección de los rostros originales sobre el eigenspace.

El reconocimiento de un rostro, luego, consiste en encontrar el rostro del conjunto de entrenamiento que minimiza la “distancia de rostro” con respecto a la imagen de entrada, es decir, que rostro de la base se asemeja al rostro de entrada al sistema. Descripto matemáticamente tendremos que :

$$\min \left(\|\Omega - \Omega_i\|^2 \right) \quad i = 1, 2, \dots, M .$$

La forma en que un rostro minimiza con otro, será variada, aunque podemos mencionar la distancia euclídea como la más elemental, y alguna otra técnica más sofisticada como mixture-distance.

PARTE III

ASPECTOS DE IMPLEMENTACION

Capítulo 6 – Estudio Preliminar

6.1. Análisis del sistema de reconocimiento a implementar

Una vez estudiadas y analizadas las distintas técnicas de reconocimiento de rostros, se optó por elegir una para su implementación en algún lenguaje de programación.

El modelo a seguir es el de Eigenfaces, por varias razones :

- El abundante material que se encontró (tanto papers como web-sites de varias universidades, investigadores y estudiantes) en relación a otras técnicas (por ejemplo de **HMM**, solo se encontraron papers de quien lo ideó)
- Otros modelos lo utilizan como piedra fundamental (por ejemplo **HMM** o algunos modelos de redes neuronales)
- Aparente simplicidad de implementación contra buenos resultados en grandes bases de datos (si bien los autores de otros modelos o técnicas, no le daban buenos resultados, investigadores independientes mostraron su alta eficacia)
- Es una técnica resistente a variaciones de luminosidad, posición y características del rostro. No todas las técnicas tienen tolerancia a estas tres simultáneamente.
- Se realiza bajo un proceso netamente automático, en comparación a otros que necesitan (sobre todo en la etapa de entrenamiento, como **Elastic Matching**) intervención humana

En base a todas estas características, que lo hacen de por sí interesante, es de esperar que una vez obtenido resultados a nivel implementación y salida, sentar bases para posteriores trabajos.

6.2. Herramientas de Desarrollo

Una de las herramientas de gran utilidad en el estudio de las diferentes técnicas del procesamiento de imágenes es el KHOROS, ya que incluye múltiples módulos implementados de distintos filtros morfológicos, funciones matemáticas,, Lógicas, de interfase gráfica, de realce, de contraste, erosión, dilatación, histogramas, transformaciones, complemento, umbralamiento de las imágenes, entre otras funciones.

KHOROS posee una herramienta visual (Cantata) donde se crean áreas de trabajo y en las mismas se pueden confeccionar redes de ejecución de procesos sobre imágenes, que funcionan mediante pasaje de tokens, aplicables a una imagen de entrada.

Con número de muestras (imágenes de rostros, obtenidas de diversas fuentes) se probaron las capacidades que KHOROS provee, aplicando filtros y técnicas para conocer que

se podría llegar a realizar, para de esta forma tomar un punto de partida a nivel de implementación del desarrollo realizado posteriormente.

Como primer lenguaje de programación, utilizamos ADA, para implementar rutinas que permitieran obtener imágenes bitmap, así como definimos un TAD matriz, y las operaciones propias de creación de “eigenfaces”, con lo cual obtuvimos un prototipo que servía para identificar el sujeto que está en la base de conocimiento que más se aproximara al rostro a reconocer. Todavía no se incorporaba en este prototipo, la opción que de decidir si los rostros eran tan semejantes como para suponer que eran pertenecientes al mismo sujeto, o el hecho de decidir que el rostro a reconocer no se encuentra en la base de las imágenes.

Posteriormente, se migraron las rutinas a lenguaje C, con un doble objetivo : por un lado testear la performance entre dos lenguajes de programación (la cual por su parte no decayó, pese a que estimabamos que la migración a C sería más veloz) y por el otro crear una librería (“eigen-lib”) utilizable desde cualquier lenguaje que pueda importar funciones C.

Finalmente, se optó por desarrollar una aplicación bajo el paradigma de programación visual (para un rápido y ameno uso del sistema) utilizando Borland C++ Builder 4.0, con lo cual utilizar las funciones C fue transparente.

6.3. Características de la Entrada

6.3.1. Obtención de la muestra

La base de datos de Rostros utilizada (llamada por sus creadores 'La Base de Datos de Rostros ORL' [18]), contiene un conjunto de imágenes de rostro tomadas entre Abril de 1992 y Abril de 1994 en el laboratorio AT&T de Cambridge. Esta base de datos se uso primeramente en el contexto de un proyecto de sistema de reconocimiento de rostros llevado a cabo con la colaboración del “Grupo de Voz, Visión y Robótica” del Departamento de Ingeniería de la Universidad de Cambridge.

Existen diez imágenes de rostros de cada una de las 40 personas. En la mayoría de los casos , las imágenes fueron sacadas distintas veces (entre las fechas anteriormente citadas) persiguiendo el objetivo de obtener variaciones en la luminosidad de las imágenes, así como distintas expresiones faciales (ojos abiertos/cerrados, sonriendo o no) y detalles en el rostro (con o sin anteojos). Todas las imágenes fueron tomadas contra un fondo oscuro y homogéneo, con las personas de pie, en una posición frontal (que en algunos casos tienen un pequeño movimiento del rostro).

Los archivos se almacenaron en formato .PGM, y puede ser visualizados en sistemas UNIX/Linux usando el programa 'xv' o en sistemas Windows con el programa de procesamiento

de imágenes Lview Pro. El tamaño de cada imagen es de 92x112 pixels, a una escala de 256 grises.

La base de datos, luego consiste en la organización de las imágenes en 40 directorios (uno por cada persona), que tienen nombres de la forma sX, donde X indica el número de persona (entre 1 y 40). En cada uno de esos directorios, existen diez imágenes distintas de esa persona, que tienen nombres de la forma Y.pgm, donde Y es el número de imagen para ese sujeto (entre 1 y 10).

Posteriormente, se volvió a reformular la base de datos, convirtiendo las imágenes de los rostros a formato .bmp de 100x100 pixels, y además modificando el nombre de estas, pasando a tener la forma SiFj.bmp, lo que significa que es la foto número j del sujeto i. Dado este cambio, se elimina la necesidad de 40 directorios

6.3.2. Estudio de diversos formatos gráficos

Dado que las imágenes de la base de datos ORL, estaban guardadas en formato .pgm, se procedió al estudio de dicho formato de imagen.

El formato gráfico pgm (portable graymap format, formato de mapa de grises portable) es el más bajo común denominador entre los formatos de archivo gráfico de escala de grises. El formato interno es muy sencillo, con un header que incluye alto, ancho y valor máximo de gris. La sección de datos incluye líneas de hasta 70 caracteres ASCII decimales, donde carácter indica el nivel de gris, desde 0 (negro), hasta el máximo valor de gris (blanco).

Si bien es un formato sencillo, y luego, de estudiar y de obtener rutinas para leer dicho formato de imágenes, se decidió evaluar otros formatos de archivos de imagen, que incluyan la posibilidad de manejar color (si bien nuestra aplicación no utiliza imágenes en color, posibles avances sobre esta podrían llegar a manejarlo).

Se pensó, entonces, en el formato .bmp, ya que es simple su codificación, no posee compresión interna, con lo cual los rostros no perdían calidad de imagen. Además este formato fue elegido por estar familiarizados con su uso, y ser uno de los más divulgados y portables de los que se conocen en el mercado.

Los archivos BMP (Bitmap), tienen inicialmente un bloque de control con toda la información acerca de la imagen y de la paleta de colores que la misma utiliza, luego se encuentra la paleta de colores donde cada color se encuentra representado por un valor RGB (Red/Green/Blue) y por último los datos, ubicados secuencialmente por fila (desde la fila n-ésima a la 1º), donde la ubicación en la pantalla se obtiene de la siguiente forma;

Fila : altura imagen – (int(offset(dato) /ancho imagen))

Columna : offset(dato)- altura imagen – (int (offset(dato) / ancho imagen))

Capítulo 7 - Desarrollo de la aplicación

Para un claro y conciso entendimiento de los pasos seguidos en el desarrollo de la aplicación, revisitaremos los aspectos matemáticos del modelo de Eigenfaces, presentados anteriormente.

Como se recordará, este algoritmo asumía la existencia de un conjunto de rostros de entrenamiento de dimensión $N \times N$ (en nuestro caso $N=100$) cada uno. Este conjunto, estarán dados por un subconjunto de la base de datos ORL reformulada, y lo denominaremos de aquí en adelante *base de conocimiento*.

Como primer paso, estas imágenes de $N \times N$ son reordenadas como un vector de una dimensión $1 \times N^2$ usando un ordenamiento por fila (es decir, tal cual un orden lexicográfico).

```
Function Encolumnar(Im: in Imagen) return Matriz is
    X: Matriz(1..N*N,1..1);
    Begin
        For i in Im'range(1) loop -- Filas
            For j in Im'Range(2) loop --Columnas
                Elemento(X,(((i-1)*N)+j),1,float(Im(i,j)));
            end loop;
        end loop;
        return X;
    end Encolumnar;
```

Se define el rostro promedio entre todas las imágenes pertenecientes a la base de conocimiento.

```
Procedure PromedioDelImagenes is
    Begin
        For i in 1..Cant_I loop
            Promedio:= Promedio + Columna(Conjunto,i);
        end loop;
        Promedio:=(1/Cant_I) * Promedio;
    end PromedioDelImagenes;
```

Luego, este rostro promedio se le resta a cada una de las imágenes de entrenamiento, de forma de obtener un nuevo conjunto de vectores (una matriz donde cada columna representa una de las imágenes).

```
Function MatrizDeDiferencias return Matriz is
    A: Matriz(N*N, Cant_I):=Conjunto;
    Begin
        For i in 1..Cant_I loop
            Columna(A,i):=Columna(A,i)-Promedio;
        end loop;
        Return A;
    end MatrizDeDiferencias;
```

La idea principal es la de realizar un Análisis de componentes principales (PCA, principal component analysis) para obtener un conjunto de vectores ortogonales que representan óptimamente la distribución de la información (vectores propios o auto-vectores).

Se deberían obtener los vectores propios de la matriz de covarianza 'C' de la matriz de diferencias obtenidas en el paso anterior. Pero hay que tener en cuenta que al ser C una matriz de $N^2 \times N^2$, la obtención de los vectores propios no es computable en tiempo y espacio de memoria.

Sin embargo, es posible obtener los vectores propios de una matriz L de $M \times M$, donde la diferencia está en el orden en que se multiplican la matriz de diferencia y su traspuesta (en 'C' se multiplica la matriz por su traspuesta, mientras que en L se multiplica la traspuesta por la matriz de diferencias). Para esto, primero obtenemos la matriz L, realizando la matriz de covarianza reducida.

```
Function MatrizL(A: in Matriz) return Matriz is
    L: Matriz(Cant_I, Cant_I);
    Begin
        L:= Traspuesta(A) * A;
        L:= (1/Cant_I)* L;
        return L;
    end MatrizL;
```

y luego obtenemos sus vectores propios o auto-vectores, y calculamos a través de ellos los vectores propios de 'C'. Estos eigenvectores u se calculan como una combinación lineal de las columnas de la matriz de diferencia (A) y los vectores propios de L.

El análisis PCA consiste, luego, en usar los más significativos M' eigenfaces (o vectores propios) para crear un subespacio de rostros.

Matriz de autovectores de 'C' para sacar los patrones.

```
Procedure MatrizDeAutovectores (L: in Matriz; A: in Matriz) is
    Begin
        Autovectores(L);
        For i in 1..Cant_EF loop
            For k in 1..Cant_I loop
                Columna(U,i):= Columna(U,i) + (L(k,i)* Columna(A,k));
            end loop;
        end loop;
    end MatrizDeAutovectores;
```

Un rostro puede ser aproximado como una combinación lineal de los eigenvectores del subespacio; esto es equivalente a decir que una cara se proyecta en el subespacio creado por los M' eigenvectores. Los coeficientes de proyección de cualquier rostro son fácilmente calculados formando el vector que es usado como una representación compacta de la imagen de entrada.

```
Function SacarPatron(Im: in Matriz) return Matriz is
    P: Matriz(Cant_EF, 1);
    Valor: float;
    Begin
        For i in 1..Cant_EF loop
            Valor:=(Traspuesta(Columna(U,i))*(Im-Promedio))(1,1);
            P(i,1):=valor;
        end loop;
    return P;
end SacarPatron;
```

Para obtener la matriz con los patrones de cada rostro.

```
Procedure ObtenerPatrones is
    Begin
        For i in 1..Cant_I loop
            Columna(ConjuntoPatrones,i):= SacarPatron(Columna(Conjunto,i));
        end loop;
    end ObtenerPatrones;
```

El reconocimiento de un rostro, luego, consiste en obtener el patrón para esta imagen, y luego encontrar el rostro del conjunto de entrenamiento que minimiza la “distancia de rostro” con respecto a la imagen de entrada, es decir, que rostro de la base se asemeja al rostro de entrada al sistema. Si dicha distancia es mayor a un determinado umbral (definido como MaxDif) el sistema indicará que ningún rostro se asemeja a la entrada.

```
Function CompararPatrones (P: in Matriz; MaxDif: in float) return integer is
    aux: float;
    min: float:= MaxDif;
    pos: integer:=0;
    Begin
        For i in Cant_I loop
            aux:= Distancia(Columna(ConjuntoPatrones,i),P);
            if (aux < min) then
                min:=aux;
                pos:=i;
            end if;
        end loop;
        return pos;
    end CompararPatrones;
```

La forma en que un rostro minimiza con otro que utilizaremos es la de la distancia euclídea, que como ya se ha visto es una de las más elementales :

```
Function Distancia(P1: in Matriz; P2: in Matriz) return float is
    Dis: float:=0;
    Begin
        for i in 1..Cant_EF loop
            Dis:= Dis + ( (P2(i,1)-P1(i,1))**2 );
        end loop;
        return sqrt(Dis);
    end Distancia;
```

Observaciones

- Dentro de las funciones y procedimientos que han sido presentados, se ha utilizado un tipo "Matriz", el cual se define abstractamente como :

Type Matriz is array(Cantidad de Filas,Cantidad de Columnas) of integer;

y se aplicaron las siguientes operaciones sobre matrices:

Columna(M: in Matriz, i: in integer). Retorna la columna i de la matriz M como una Matriz de una sola columna.

Fila (M: in Matriz, i: in integer). Retorna la fila i de la matriz M como una Matriz de una sola fila.

Traspuesta(M: in Matriz) Retorna la traspuesta de M.

"+"(M1,M2: in Matriz). Retorna la matriz resultante de sumar M1 por M2.

"-"(M1,M2: in Matriz). Retorna la matriz resultante de restar M1 por M2.

"*"(M1,M2: in Matriz). Retorna la matriz resultante de multiplicar M1 por M2.

"*"(F: in float, M: in Matriz). Retorna la matriz resultante de multiplicar M por un valor constante F.

Autovectores(M: in out Matriz). Procedimiento que devuelve en M los autovectores de M, ordenados de mayor a menor según los autovalores. Cada fila es un autovector.

- Tambien, se definieron las siguientes variables :

N: integer Cantidad de pixels en una imagen (tanto alto como ancho, en nuestro caso toma valor 100).

Cant_I: integer Cantidad de imagenes en la base de entrenamiento.

Cant_EF: integer; Cantidad de EigenFaces usado. Debe ser menor o igual a Cant_I.

ConjuntoPatrones: Matriz(Cant_EF, Cant_I);

-- Conjunto de los patrones de cada imagen de la base.

-- Cada columna corresponde al patrón de una imagen.

Conjunto: Matriz(**N*N**, Cant_I);

-- Conjunto con las imagenes de la base.

-- Cada columna es una imagen representada en forma de vector columna

-- formado concatenando cada fila de la imagen.

Promedio: Matriz(**N*N**, 1); Imagen promedio de todas las imágenes de la base. Es una matriz de una sola columna.

U: Matriz(N*N, Cant_I); Matriz con los autovectores de la matriz de covarianza. Cada fila corresponde a un autovector.

- Por último se definen las funciones esenciales para el aprendizaje y el reconocimiento.

Función para el aprendizaje, donde C indica la cantidad de eigenfaces usados para representar los patrones.

```
Procedure Aprendizaje (C: in integer) is
  X: Matriz(Dim*Dim, Cant_I);
  L: Matriz(Cant_I, Cant_I);
  Begin
    if (C>Cant_I) then
      Cant_EF:=Cant_I;
    else
      Cant_EF:=C;
    end if;

    PromedioDelImagenes;
    X:=MatrizDeDiferencias;
    L:=MatrizL(X);
    MatrizDeAutevtores(L,X);
    ObtenerPatrones;
  end Aprendizaje;
```

Función para el reconocimiento, retorna la posición de la imagen en la base de conocimiento.

```
Function ReconocerImagen(Im: Matriz, MaxDif: integer) return integer is
  P: Matriz(Cant_EF,1);
  Id: integer;
  begin
    P:=SacarPatron(Im);
    Id:=CompararPatrones(P, MaxDif);
    return Id;
  end ReconocerImagen;
```

Capítulo 8 - Optimizaciones y Pruebas

Se realizaron una serie de pruebas para estudiar los resultados obtenidos por el sistema para distintas bases de conocimiento.

Para la creación de las bases de conocimiento, se fueron realizando subconjuntos de una base de rostros que incluye únicamente una imagen de rostro por persona, es decir , 40 imágenes de rostro en total. Es así que se conformaron bases de conocimiento incluyendo cinco, diez, quince, veinte, veinticinco, treinta, treinta y cinco y cuarenta imágenes. Es evidente que no sean realizado para las pruebas, todas las bases de conocimiento posibles ya que con n imágenes, la cantidad posible de bases de conocimiento es :

$$\binom{n}{40} = \frac{n!}{40!(n-40)!}$$

Luego, las bases de conocimiento con n imágenes de rostro se fueron formando tomando de a n personas (1 imagen de rostro por persona). Si la partición no se alcanzaba con la persona 40, se comenzaba nuevamente con la persona 1 hasta completar las n personas (por ejemplo, para particionar en 15 personas, se tomaron 3 bases de conocimiento de 1..15, 16..30, y la última 31..40 + 1..5)

En cuanto a las imágenes utilizadas para el reconocimiento formal, se tomaron los 400 rostros (10 de cada persona, tal cual se mencionó anteriormente) que posee la base de datos para testear contra la base de conocimiento.

Por otro lado se definió un “nivel de error” respecto al reconocimiento, cuyos valores pueden ser 0-2-5-8 o 10. Las bases de conocimiento fueron testeadas para cada uno de estos valores, para poder observar como se comporta el sistema en cada caso.

Cuando se utiliza un nivel de error 0, el rostro a reconocer debe ser casi idéntico al rostro que le corresponde (la misma persona con alguna variación de las anteriormente descriptas) dentro de la base de conocimiento, para obtener un resultado positivo, es decir, lo reconozca correctamente; a medida que se aumenta el de nivel de error, se permite una mayor diferencia entre el rostro a reconocer y el rostro que le corresponde. Cuando utilizamos un nivel de error 10, siempre obtendremos un resultado positivo (debido a que siempre devuelve el rostro que más se aproxime, asume un correcto reconocimiento).

Como consecuencia de esto, a medida que se aumenta el nivel de error, aumenta la cantidad de aciertos en el reconocimiento de los rostros de personas que poseen una imagen de rostro en la base de conocimiento, y disminuye la de las personas que no están en la base, hasta llegar a 0, con el nivel 10.

Para una mejor legibilidad de las pruebas realizadas, los resultados obtenidos en las pruebas de reconocimiento, se dividieron en dos partes: la primera registra los porcentajes de aciertos de las imágenes de rostros de la correspondiente base de conocimiento, y la otra con las de los que no están en la base.

Cabe destacar que un “acerto” en el reconocimiento para una imagen de un rostro de una persona que posee imagen en la base de conocimiento, corresponde a indicar como resultado esa imagen de la base de conocimiento, mientras que para una imagen de rostro de una persona que no se encuentra en la base de conocimiento, un “acerto” corresponde a informar que no existe en la base. Es por ello que los resultados pueden observarse y/o analizarse mejor, ya que no existe una dependencia con la cantidad de imágenes de rostros en la base de conocimiento.

Para cada una de las base de conocimiento utilizadas, se realizaron dos cuadros indicando para cada persona la cantidad de aciertos (de las 10 fotos de esa persona) en cada uno de los niveles de error utilizados (0-2-5-8-10), en un cuadro se detallan los resultados de las personas en la base de conocimiento, y en el otro las personas que no se encuentran en la misma.

A continuación se presentan, a modo informativo, algunos de los resultados realizados en base a pruebas, para bases de conocimiento de 5, 10, 15, 20, 25 y 40 imágenes de rostros.

Debido a que se han generado más testeos, se han tomado todos los resultados, y se realiza como análisis final, un cuadro resumen. Este cuadro, indica la cantidad de imágenes de rostros utilizadas en la base de conocimiento, y que rostros en particular la componen, obteniendo los porcentajes de aciertos de cada una en los distintos niveles de error, de los cuadros particulares.

Sujeto en la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
21	1	10	10	10	10
22	6	10	10	10	10
23	2	8	8	8	8
24	3	10	10	10	10
25	4	8	10	10	10
ACIERTOS	16	46	48	48	48
PROMEDIO	32%	92%	96%	96%	96%

Sujeto fuera de la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
1	10	10	7	0	0
2	10	4	0	0	0
3	10	8	0	0	0
4	10	6	0	0	0
5	10	0	0	0	0
6	10	2	0	0	0
7	10	10	1	0	0
8	10	10	4	0	0
9	10	7	0	0	0
10	10	6	0	0	0
11	10	10	9	1	0
12	10	6	0	0	0
13	10	1	0	0	0
14	10	10	5	0	0
15	10	10	0	0	0
16	10	9	2	0	0
17	10	10	5	0	0
18	10	0	0	0	0
19	10	10	6	0	0
20	10	7	1	0	0
26	10	2	0	0	0
27	10	10	10	0	0
28	10	10	10	0	0
29	10	3	0	0	0
30	10	0	0	0	0
31	10	7	4	0	0
32	10	9	7	0	0
33	10	4	0	0	0
34	10	1	0	0	0
35	10	7	1	0	0
36	10	9	1	0	0
37	10	10	4	0	0
38	10	0	0	0	0
39	10	2	0	0	0
40	10	4	0	0	0
ACIERTOS	350	214	77	1	0
PROMEDIO	100%	61%	22%	0%	0%

Cuadro 1) Una de las pruebas de reconocimiento con una base de conocimiento de 5 imágenes de rostros

Sujeto en la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
1	1	3	5	9	9
2	1	8	10	10	10
3	1	6	8	8	8
4	1	3	7	7	7
5	3	6	8	10	10
6	2	10	10	10	10
7	1	8	10	10	10
8	2	5	5	10	10
9	1	5	7	8	8
10	1	1	2	4	5
ACIERTOS	14	55	72	86	87
PROMEDIO	14%	55%	72%	86%	87%

Sujeto fuera de la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
11	10	10	10	0	0
12	10	9	3	0	0
13	10	10	7	0	0
14	10	10	9	2	0
15	10	10	8	0	0
16	10	9	4	0	0
17	10	10	8	1	0
18	10	10	8	0	0
19	10	10	8	0	0
20	10	10	9	1	0
21	10	10	7	0	0
22	10	10	10	3	0
23	10	10	9	2	0
24	10	10	9	2	0
25	10	8	0	0	0
26	10	9	1	0	0
27	10	10	9	1	0
28	10	10	3	0	0
29	10	10	10	2	0
30	10	10	1	0	0
31	10	7	5	1	0
32	10	10	10	2	0
33	10	10	10	0	0
34	10	10	10	0	0
35	10	10	7	1	0
36	10	8	6	0	0
37	10	9	5	0	0
38	10	10	9	1	0
39	10	10	9	4	0
40	10	8	3	0	0
ACIERTOS	300	287	207	23	0
PROMEDIO	100%	96%	69%	8%	0%

Cuadro 2) Una de las pruebas de reconocimiento con una base de conocimiento de 10 imágenes de rostros

Sujeto en la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
1	1	1	2	3	3
2	1	6	8	8	8
3	1	5	8	9	9
4	1	3	7	7	7
5	3	5	7	10	10
6	1	5	10	10	10
7	1	7	10	10	10
8	1	5	5	10	10
9	1	5	7	8	8
10	1	1	2	4	7
11	1	5	6	7	7
12	1	3	5	7	7
13	2	8	9	10	10
14	1	5	5	5	8
15	2	5	5	5	5
ACIERTOS	19	69	96	113	119
PROMEDIO	13%	46%	64%	75%	79%

Sujeto fuera de la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
16	10	10	5	2	0
17	10	10	10	1	0
18	10	10	7	1	0
19	10	10	10	5	0
20	10	10	10	3	0
21	10	10	6	0	0
22	10	10	10	9	0
23	10	10	8	2	0
24	10	10	9	0	0
25	10	8	0	0	0
26	10	10	8	1	0
27	10	10	10	5	0
28	10	10	10	4	0
29	10	10	10	3	0
30	10	10	9	0	0
31	10	10	3	0	0
32	10	10	9	2	0
33	10	10	10	2	0
34	10	10	9	3	0
35	10	10	9	1	0
36	10	8	7	0	0
37	10	10	10	5	0
38	10	10	10	1	0
39	10	10	10	8	0
40	10	8	7	0	0
ACIERTOS	250	244	206	58	0
PROMEDIO	100%	98%	82%	23%	0%

Cuadro 3) Una de las pruebas de reconocimiento con una base de conocimiento de 15 imágenes de rostros

Sujeto en la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
1	1	1	3	4	7
2	1	8	9	10	10
3	1	3	8	9	9
4	1	3	6	7	7
5	2	5	5	9	10
6	1	3	9	10	10
7	1	6	10	10	10
8	1	5	5	9	10
9	1	4	6	6	6
10	1	1	2	2	3
31	1	4	4	5	5
32	1	1	4	7	7
33	1	5	8	10	10
34	1	6	10	10	10
35	1	1	1	2	2
36	1	2	2	2	2
37	2	5	5	5	9
38	1	5	9	10	10
39	1	4	7	7	7
40	1	2	4	5	5
ACIERTOS	22	74	117	139	149
PROMEDIO	11%	37%	59%	70%	75%

Sujeto fuera de la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
11	10	10	10	10	0
12	10	10	8	3	0
13	10	10	10	8	0
14	10	10	10	10	0
15	10	10	10	2	0
16	10	10	10	4	0
17	10	10	10	3	0
18	10	10	10	8	0
19	10	10	10	2	0
20	10	10	10	4	0
21	10	10	8	0	0
22	10	10	7	1	0
23	10	9	5	1	0
24	10	10	10	9	0
25	10	10	1	0	0
26	10	10	6	1	0
27	10	10	10	3	0
28	10	10	6	0	0
29	10	10	10	2	0
30	10	10	10	1	0
ACIERTOS	200	199	171	72	0
PROMEDIO	100%	100%	86%	36%	0%

Cuadro 4) Una de las pruebas de reconocimiento con una base de conocimiento de 20 imágenes de rostros

Sujeto en la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
16	1	1	1	2	2
17	1	2	3	3	3
18	1	2	5	8	8
19	2	4	8	9	9
20	1	4	4	6	6
21	1	5	10	10	10
22	1	10	10	10	10
23	1	3	3	5	5
24	1	4	9	10	10
25	1	5	7	8	9
26	1	3	7	8	8
27	1	5	6	6	7
28	1	3	4	4	4
29	1	5	5	5	5
30	1	6	7	9	9
31	1	4	4	5	5
32	1	1	4	7	7
33	1	5	8	8	8
34	1	6	9	9	9
35	1	1	1	1	1
36	1	2	2	2	2
37	2	5	5	5	5
38	1	5	8	8	8
39	1	2	5	6	6
40	1	2	4	5	5
ACIERTOS	27	95	139	159	161
PROMEDIO	11%	38%	56%	64%	64%

Sujeto en la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
1	10	10	10	3	0
2	10	10	10	8	0
3	10	10	6	1	0
4	10	10	8	0	0
5	10	10	8	0	0
6	10	10	10	7	0
7	10	10	10	7	0
8	10	10	10	10	0
9	10	10	6	0	0
10	10	10	7	1	0
11	10	10	10	10	0
12	10	10	8	2	0
13	10	10	7	0	0
14	10	10	10	9	0
15	10	10	10	4	0
ACIERTOS	150	150	130	62	0
PROMEDIO	100%	100%	87%	41%	0%

Cuadro 5) Una de las pruebas de reconocimiento con una base de conocimiento de 25 imágenes de rostros

Sujeto en la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
1	1	1	2	2	2
2	1	6	9	9	9
3	1	3	8	8	8
4	1	3	5	5	5
5	2	4	5	7	7
6	1	3	9	10	10
7	1	5	9	10	10
8	1	3	5	7	9
9	1	4	6	7	7
10	1	1	2	4	4
11	1	3	5	7	7
12	1	3	3	5	5
13	2	3	8	10	10
14	1	5	5	5	8
15	1	4	5	5	5
16	1	1	1	2	2
17	1	2	3	3	3
18	1	2	5	6	6
19	1	3	8	9	9
20	1	4	4	5	5
21	1	7	10	10	10
22	1	9	10	10	10
23	1	3	5	8	8
24	1	5	8	10	10
25	1	5	7	7	7
26	1	2	6	8	8
27	1	5	6	6	6
28	1	3	4	5	5
29	1	5	6	6	6
30	1	6	8	9	9
ACIERTOS	32	113	177	205	210
PROMEDIO	11%	38%	59%	68%	70%

Sujeto fuera de la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
31	10	9	6	0	0
32	10	10	10	2	0
33	10	10	2	0	0
34	10	10	8	6	0
35	10	10	7	3	0
36	10	10	7	1	0
37	10	10	6	4	0
38	10	9	1	0	0
39	10	10	5	0	0
40	10	9	7	0	0
ACIERTOS	100	97	59	16	0
PROMEDIO	100%	97%	59%	16%	0%

Cuadro 6) Una de las pruebas de reconocimiento con una base de conocimiento de 30 imágenes de rostros

Sujeto en la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
6	1	3	4	10	10
7	1	4	9	10	10
8	1	3	5	7	10
9	1	3	7	9	9
10	1	1	2	3	3
11	1	3	6	7	7
12	1	2	3	5	5
13	1	3	8	10	10
14	1	5	5	5	8
15	1	4	5	5	6
16	1	1	1	2	2
17	1	2	3	3	3
18	1	2	5	6	6
19	1	3	8	9	9
20	1	4	4	6	6
21	1	4	10	10	10
22	1	8	10	10	10
23	1	3	3	5	5
24	1	4	7	10	10
25	1	3	6	7	7
26	1	2	6	8	8
27	1	5	6	6	6
28	1	3	4	4	4
29	1	4	5	6	6
30	1	5	7	9	9
31	1	3	4	4	5
32	1	1	4	7	7
33	1	4	8	8	8
34	1	5	8	8	8
35	1	1	1	1	1
36	1	2	2	2	2
37	1	5	5	5	5
38	1	2	8	8	8
39	1	2	5	6	6
40	1	2	4	5	5
ACIERTOS	35	111	188	226	234
PROMEDIO	10%	32%	54%	65%	67%

Sujeto fuera de la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
1	10	10	10	5	0
2	10	10	7	2	0
3	10	10	8	0	0
4	10	10	9	1	0
5	10	10	10	1	0
ACIERTOS	50	50	44	9	0
PROMEDIO	100%	100%	88%	18%	0%

Cuadro 7) Una de las pruebas de reconocimiento con una base de conocimiento de 35 imágenes de rostros

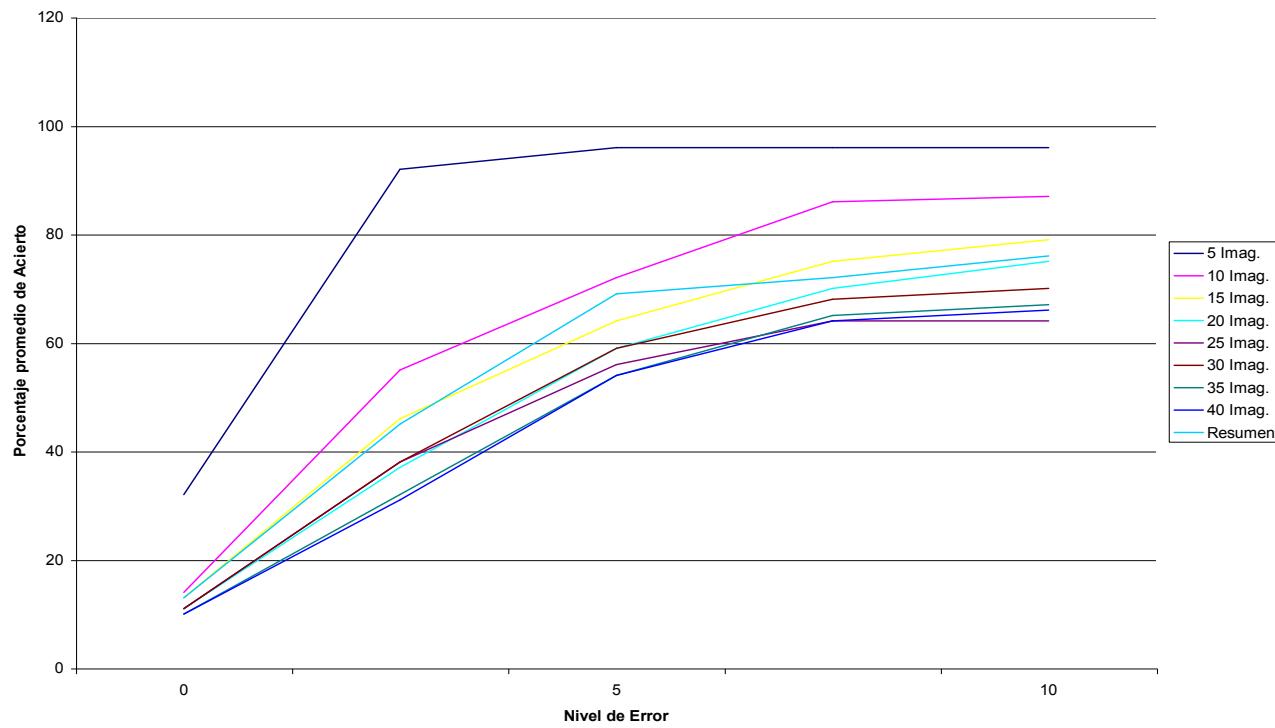
Sujeto en la Base	Aciertos de acuerdo al nivel de error permitido				
	0	2	5	8	10
1	1	1	1	2	2
2	1	5	8	9	9
3	1	3	8	9	9
4	1	2	4	5	5
5	2	4	5	8	8
6	1	3	4	10	10
7	1	4	9	10	10
8	1	3	5	6	10
9	1	3	6	8	8
10	1	1	2	2	3
11	1	3	6	7	7
12	1	1	3	5	5
13	1	3	8	10	10
14	1	5	5	6	9
15	1	4	5	5	5
16	1	1	1	2	2
17	1	2	3	3	3
18	1	2	5	6	6
19	1	3	8	9	9
20	1	4	4	6	6
21	1	4	10	10	10
22	1	8	10	10	10
23	1	3	3	5	5
24	1	4	7	10	10
25	1	3	6	7	7
26	1	2	6	8	8
27	1	5	6	6	6
28	1	3	4	4	4
29	1	4	5	6	6
30	1	5	7	9	9
31	1	3	4	5	5
32	1	1	4	6	6
33	1	4	8	8	8
34	1	5	8	8	8
35	1	1	1	1	1
36	1	2	2	2	2
37	1	5	5	5	5
38	1	2	8	8	8
39	1	2	6	6	6
40	1	2	4	5	5
ACIERTOS	41	125	214	257	265
PROMEDIO	10%	31%	54%	64%	66%

Cuadro 8) Prueba de reconocimiento con la base de conocimiento de 40 imágenes de rostros

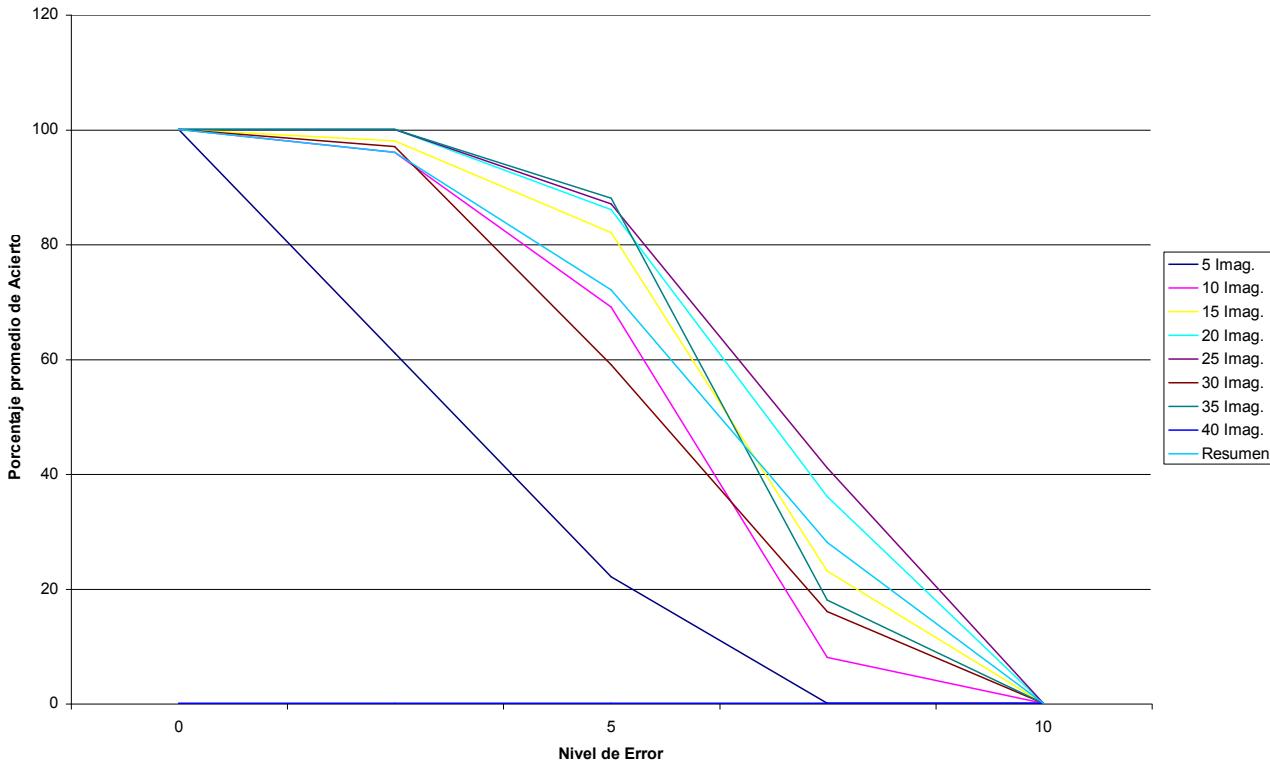
Tamaño de la base	Fotos en la Base	Personas en la base					Personas fuera de la base				
		0	2	5	8	10	0	2	5	8	10
5	1..5	16%	68%	90%	92%	92%	100%	78%	29%	0%	0%
	6..10	20%	64%	82%	88%	90%	100%	97%	83%	42%	0%
	11..15	16%	58%	74%	84%	90%	100%	94%	72%	26%	0%
	16..20	16%	42%	58%	76%	82%	100%	100%	84%	41%	0%
	21..25	32%	92%	96%	96%	96%	100%	61%	22%	0%	0%
	26..30	22%	70%	80%	84%	88%	100%	91%	63%	39%	0%
	31..35	12%	48%	62%	76%	90%	100%	99%	87%	64%	0%
	36..40	16%	52%	60%	78%	94%	100%	97%	81%	50%	0%
10	1..10	14%	55%	72%	86%	87%	100%	96%	69%	8%	0%
	11..20	12%	41%	55%	68%	77%	100%	100%	93%	57%	0%
	21..30	16%	68%	79%	83%	83%	100%	87%	46%	24%	0%
	31..40	13%	39%	59%	66%	80%	100%	98%	88%	61%	0%
15	1..15	13%	46%	64%	75%	79%	100%	98%	82%	23%	0%
	16..30	13%	53%	67%	71%	72%	100%	94%	64%	25%	0%
	1..5 - 31..40	13%	39%	59%	69%	74%	100%	98%	80%	34%	0%
	11..25	13%	48%	63%	71%	73%	100%	98%	65%	20%	0%
20	1..20	11%	36%	55%	67%	70%	100%	100%	85%	28%	0%
	21..40	11%	43%	62%	67%	68%	100%	100%	81%	43%	0%
	1..10 - 31..40	11%	37%	59%	70%	75%	100%	100%	86%	36%	0%
	11..30	12%	46%	62%	68%	69%	100%	98%	66%	21%	0%
25	1..25	11%	39%	60%	70%	72%	100%	98%	63%	18%	0%
	16..40	11%	38%	56%	64%	64%	100%	100%	87%	41%	0%
	8..32	10%	38%	56%	65%	68%	100%	99%	68%	25%	0%
	1..12 - 28..40	11%	36%	53%	66%	68%	100%	99%	84%	34%	0%
30	1..30	11%	38%	59%	68%	70%	100%	97%	59%	16%	0%
	11..40	10%	35%	55%	62%	64%	100%	100%	85%	34%	0%
	1..10 - 21..40	10%	38%	57%	69%	70%	100%	100%	89%	32%	0%
	1..20 - 31..40	10%	31%	51%	62%	65%	100%	99%	75%	13%	0%
35	1..35	10%	35%	55%	67%	69%	100%	100%	56%	8%	0%
	6..40	10%	32%	54%	65%	67%	100%	100%	88%	18%	0%
	3..37	10%	33%	53%	66%	68%	100%	100%	64%	16%	0%
	1..18 - 24..40	10%	32%	53%	63%	65%	100%	98%	72%	8%	0%
40	1..40	10%	31%	54%	64%	66%	-	-	-	-	-
PROMEDIO TOTAL		13%	45%	63%	72%	76%	100%	96%	72%	28%	0%

Cuadro 9) Cuadro resumen de las pruebas de reconocimiento realizadas

Comparación entre las Pruebas de Reconocimiento de Sujetos en la Base



Comparación entre las Pruebas de Reconocimiento de Sujetos fuera de la Base



CONCLUSIONES

Luego de la adquisición de los conocimientos necesarios para la realización del proyecto y habiendo completado todas las etapas de desarrollo, partiendo desde el análisis de requerimientos, los aspectos de implementación, puesta a punto hasta llegar a las evaluaciones, se impone la necesidad de elaborar las conclusiones generales.

Para esto se deben contemplar el rendimiento que muestra la aplicación, las experiencias que aportó al grupo de desarrolladores y finalmente establecer algunos lineamientos futuros, en función de posibles mejoras al sistema o como forma de expansión del mismo.

En líneas generales, se puede esperar que el sistema se comporte, tolerante a imágenes de rostros con diferentes características de luminosidad, posiciones y expresiones, tal cual se requería, logrando identificar los rostros de la base de conocimiento que se le enseñe, de manera acertada. Es interesante, también el hecho que se provee un nivel de precisión (que hemos denominado nivel de error), con el cual se puede realizar un análisis del comportamiento del sistema frente a las variaciones en el rostro, tantas veces mencionadas.

Se puede explicitar, luego de estas consideraciones que la respuesta en líneas generales de sistema es altamente eficiente, por lo que se consideran cumplidas las expectativas que abarcaron el desarrollo de esta aplicación.

Por otro lado, deben destacarse la importante gama de conocimientos adquiridos por los integrantes del proyecto, durante el desarrollo del mismo. Se logró un aprendizaje de técnicas tratadas en procesamiento digital de imágenes, descubrir el significado de los sistemas biométricos, así como principalmente, se obtuvo un amplio conocimiento del funcionamiento y características de distintas arquitecturas y modelos de sistemas de reconocimiento de rostros, especializándose e implementando en particular uno de ellos: el modelo de Eigenfaces.

Se destaca el desarrollo de la aplicación y en particular de una librería para el reconocimiento de rostros utilizable en cualquier lenguaje de programación, logrando un producto terminado que no se limita solo a un trabajo de investigación teórica.

En cuanto a futuros desarrollos que se pueden establecer a partir del estado actual de este trabajo, incluyendo tanto aspectos teóricos como de implementación.

- Utilizar el modelo Eigenface en un reconocimiento de rostros sobre vídeo.
- Mejorar el sistema para trabajar sobre imágenes de rostros en colores.
- Desarrollo de un sistema (o sub-sistema) normalizador de rostros potente, de tal forma que la base de datos de rostros pueda componerse de cualquier imagen que posea al menos un rostro.
- Implementación de otros modelos de sistemas de reconocimiento de rostros, haciendo hincapié en intentar automatizar etapas del mismo que actualmente se realizan manualmente.

Finalmente, un desarrollo muy interesante sería el de acoplar el software a un hardware específico (o modelo a pequeña escala), que por ejemplo, abra una puerta si el rostro es reconocido satisfactoriamente.

ANEXOS

ANEXO A – BASE DE DATOS DE ROSTROS ORL (Olivetti Research Lab.)

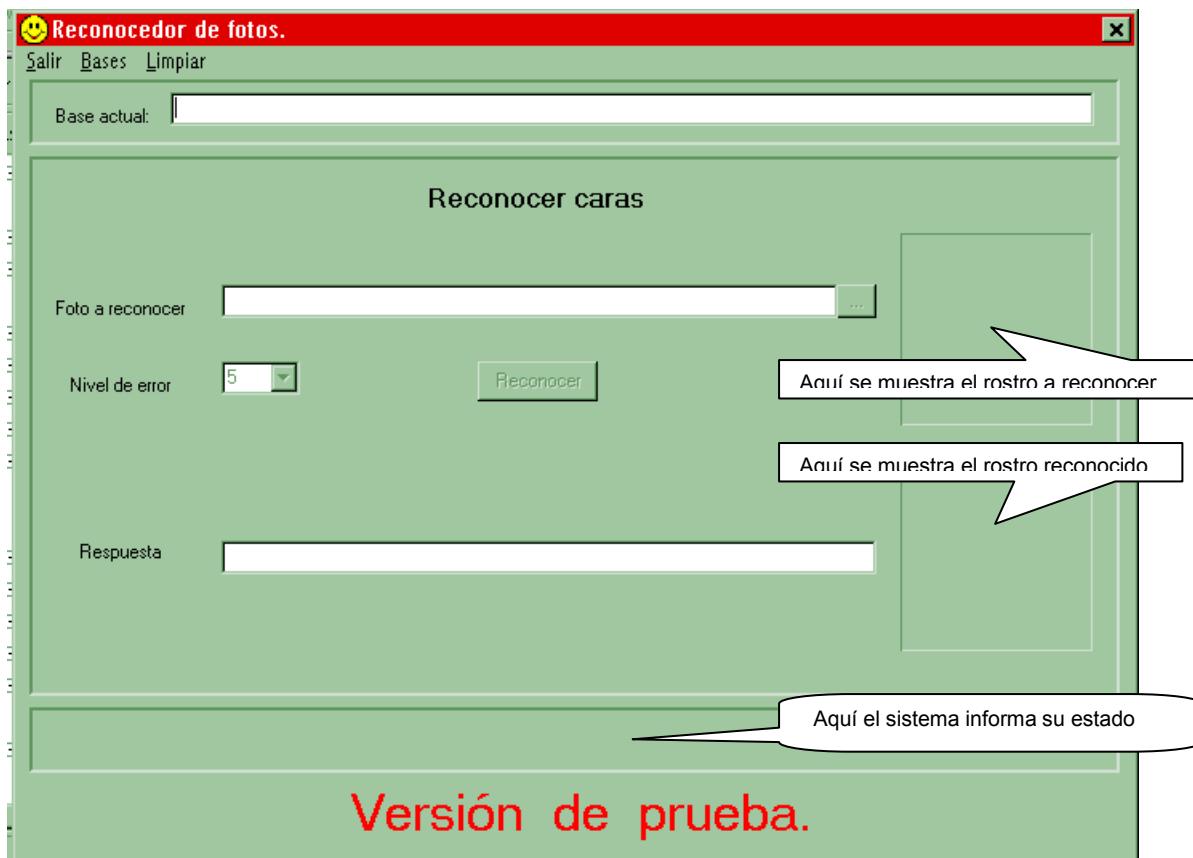




ANEXO B - Manual del usuario del sistema de reconocimiento de rostros

Se describe la funcionalidad y las opciones que posee el sistema de reconocimiento de rostros que se implementó para realizar testeos.

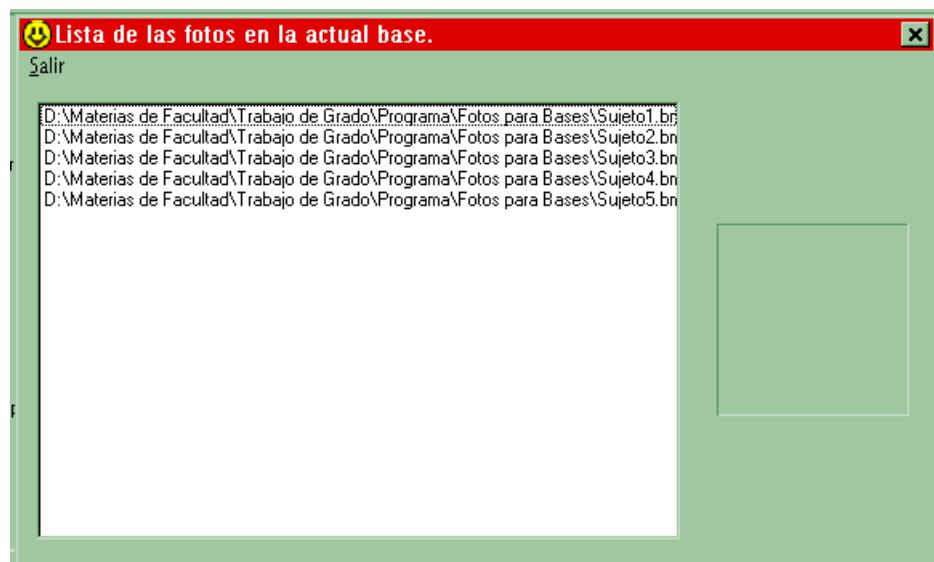
La pantalla principal, posee tres menús :



- El menú **SALIR** se utiliza para salir del sistema.
- El menú **BASES** tiene 4 opciones:
 1. **ABRIR** : con esta opción, es posible abrir una nueva base de conocimiento. Una vez que se selecciona una base el sistema indica que el próximo paso es el de realizar el cálculo (es decir que el sistema realice el aprendizaje de la base). Debido a que este paso es fundamental para un posterior reconocimiento, se deshabilitan las opciones referentes al reconocimiento en sí.
 2. **CALCULAR** : con esta opción se realiza el aprendizaje de la base de conocimiento que está abierta, tal como se mencionó anteriormente. Una vez que el proceso de aprendizaje ha concluido, el sistema indicará que ya se puede realizar el reconocimiento de rostros, por lo cual habilita las opciones de reconocimiento.
 3. **EDITAR** : con esta opción, se abre un editor de bases de conocimiento, en el cual agregaremos o eliminaremos imágenes de rostros a la base en edición.

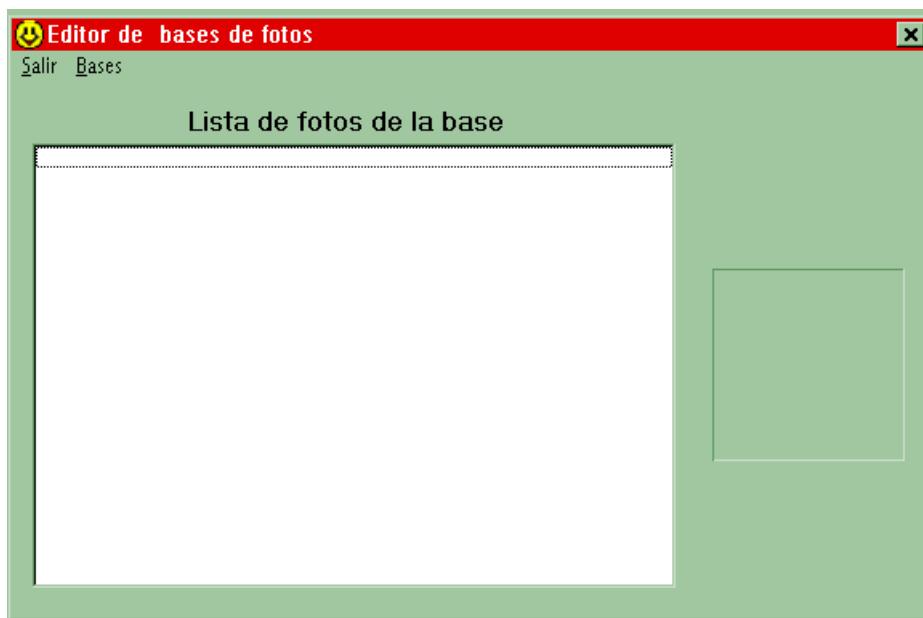
- 4. **VER** : con esta opción, se muestra la base de conocimiento abierta para poder observar las imágenes de los rostros que están en ella.
- El menú **LIMPIAR** se utiliza como “reset” del sistema, es decir, que no deja base de conocimiento abierta (y mucho menos calculada), así como retira toda información sobre el reconocimiento.
- El botón “...” es utilizado para poder seleccionar la imagen de rostro a reconocer.
- El **nivel de error** se utiliza para seleccionar un nivel, donde ya están predefinidos, niveles de error de 0-10.
- El botón **RECONOCER**, se utiliza para reconocer el rostro ingresado con el nivel de error indicado, devolviendo como resultado el rostro de la base de conocimiento y el "path" completo del resultado, o bien una leyenda indicando que no está en la base de conocimiento (es decir que el sistema no conoce a esa persona).

Como se mencionó anteriormente, con la opción **VER**, se muestra la base de conocimiento abierta, abriéndose la siguiente ventana:



- Con la opción **SALIR** se cierra dicha ventana.
- Se muestra una lista con el path y nombre de archivo de todas las imágenes de rostro pertenecientes a la base que está abierta en ese momento. Cuando se selecciona una, se muestra esa imagen de la persona.

También se describió la opción **EDITAR**, la cual abre el editor de bases:



- Con la opción **SALIR** se cierra el editor de bases.
- El menú **BASES** tiene 3 opciones:
 1. **NUEVA** : prepara la pantalla para crear una nueva base (deja en blanco la lista de las fotos de la base).
 2. **ABRIR** : abre una base ya creada para poder ver con que rostros está conformada, o bien para editarla (agregando o sacando alguna imagen de rostro).
 3. **GRABAR** : graba la base que se ha creado o modificado. Hay que tener en cuenta, que si se desea grabar con el nombre de la base que se ha abierto en el reconocedor, se presenta un cartel indicando que no se puede modificar por estar en uso.
- En la lista con las fotos existe un menú contextual que contiene:
 1. **ELIMINAR FOTO** : elimina el rostro que está seleccionado.
 2. **AGREGAR FOTO** : agrega uno o más rostros a la base.
- Cuando se selecciona un elemento de la lista se muestra la correspondiente foto.

PASOS PARA EL RECONOCIMIENTO DE UN ROSTRO

1. Abrir una base.
2. Calcular la base.
3. Seleccionar una imagen de entrada.
4. Modificar (si se desea) el nivel de error.
5. Presionar el botón Reconocer.

Bibliografía

Books & Papers

- [1] Rafael C. Gonzalez, Tratamiento Digital de Imágenes, Addison-Wesley Publishing Company, 1992
- [2] Gutiérrez González, L. 2000. Reconocimiento de rostros utilizando redes neuronales. Tesis Maestría. Ciencias con Especialidad en Ingeniería en Sistemas Computacionales. Departamento de Ingeniería en Sistemas Computacionales, Escuela de Ingeniería, Universidad de las Américas Puebla. Mayo.
- [3] M. Turk and A. Pentland, "Face recognition using eigenfaces", in Proceedings of International Conference on Pattern Recognition , pp. 586-591, 1991
- [4] Jose Gerardo Gonzalez, *Face Recognition and Detection Using Eigenfaces*, 1997
- [5] F. Samaria and S. Young, "HMM based architectures for face identification", *Image and Computer Vision*, vol. 12, pp 537-583, October 1994
- [6] Ara V. Nefian and Monson H. Hayes III, *Face detection and Recognition Using Hidden Markov Models*, in ICASSP98, 1998
- [7] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger and Christoph von der Malsburg *Face Recognition by Elastic Bunch Graph Matching*, in Intelligent Biometric Techniques in Fingerprint and Face Recognition, eds. L.C. Jain et al., publ. CRC Press, ISBN 0-8493-2055-0, Chapter 11, pp. 355-396, (1999)
- [8] Tou J. T., Gonzalez R. C.
Pattern Recognition Principles, Addison-Wesley Publishing Company, 1974.
- [9] Hrycej T., *Modular Learning in Neural Networks: A Modularized Approach to Neural Network Classification*. John Wiley & Sons, Inc. 1992.
- [10] Haykins S., *Neural Networks: A Comprehensive Foundation*. MacMillan College Publishing Company, 1994
- [11] Brunelli R., Poggio T., "Face Recognition: Features versus Templates", *IEEE Trans. Patt. Anal. And Mach. Intell.*, vol. 15, No. 10, pp. 1042-1052, 1993.
- [12] Jun Zhang, Young Yan, and Martin Lades. "Face Recognition: Eigenface, Elastic Matching, and Neural Nets." *Proceedings of the IEEE*. vol. 85. No. 9, pp.1422-1435, 1997.
- [13] Yacoub M., Bennani M., "Discriminative Feature Extraction and Selection applied to Face Recognition", *JCNN*, 1999.
- [14] Oliver de Vel, Stefan Aeberhard, "Line-Based Face Recognition under Varying Pose", *IEEE Trans. Patt. Anal. And Mach. Intell.*, vol. 21, No. 10, pp. 1081-1088, 1999.
- [15] F. Samaria, "Face Recognition Using Hidden Markov Models", PhD thesis, University of Cambridge, Cambridge, U.K. 1994.
- [16] S. Lin, S. kung, y L. Lin, "Face Recognition/Detection by Probabilistic Decision-Based Neural Network", *IEEE Trans. Neural Networks*, vol. 8, pp. 114-132, 1997.
- [17] Steve Lawrence, C. Lee Giles , Ah Chung Tsoi, Andrew D. Back "Face Recognition: A Convolutional Neural Network Approach". *IEEE Transactions on Neural Networks, Special Issue on Neural Networks and Pattern Recognition*, Volume 8, Number 1, pp. 98-113, 1997.

Recursos de Internet

- [18] AT&T Laboratories Cambridge, "The ORL Database of Faces",
ftp://ftp.uk.research.att.com:pub/data/att_faces.zip
<http://www.cam-orl.co.uk/facedatabase.html>
- [19] Graphics File Formats
<http://www.cs.ubc.ca/spider/ladic/text/pgm.txt>
- [20] Ryan Carmody/ James Bordley
CPSC 352 FACE RECOGNITION – ARTIFICIAL INTELLIGENCE (FALL 1999)
<http://shakti.trincoll.edu/~rcarmody/facerec.html>
- [21] Face Recognition Homepage
<http://www.mbhhs.edu/class/compsci/ai/1997/face.html>
- [22] The Face Detection Home Page (by Dr. Robert Frischholz)
<http://home.t-online.de/home/Robert.Frischholz/face.htm>
- [23] INFORME TECNICO DEL PROYECTO RECONOCIMIENTO DE ROSTROS POR INDEXADO APOYADO POR LA RED PARA EL DESARROLLO DE LA INFORMATICA (REDI-CONACYT)
<http://www.cic.ipn.mx/~project/identification/>
- [24] MIT Media Laboratory Vision and Modeling Group (VISMOD) Face Recognition Demo Page
<http://www-white.media.mit.edu/vismod/demos/facerec/>
- [25] SISTEMAS BIOMÉTRICOS: MATCHING DE HUELLAS DACTILARES MEDIANTE TRANSFORMADA DE HOUGH GENERALIZADA
Domingo Morales L. / Javier Ruiz-del-Solar
http://www2.ing.puc.cl/~iinq/ed429/sistemas_biometricos.htm
- [26] Investigadores de reconocimiento de rostros utilizando redes neuronales.
Hervé Abdi <http://wwwpub.utdallas.edu/~herve/>
http://wwwpub.utdallas.edu/~herve/face_lab.html
Dominique Valentin <http://www.u-bourgogne.fr/d.valentin>
A.J. O'Toole http://wwwpub.utdallas.edu/~otoole/face_try.html
- [27] Investigadores sobre aspectos Cognitivos (psicológicos) del reconocimiento de Rostros
Mike Burton Department Psychology - Face Recognition HomePage, University of Glasgow
<http://medusa.psy.gla.ac.uk/~mike/facerec.html>
Andy Ellis, Andy Young, Department of Psychology, University of York
<http://www.york.ac.uk/depts/psych/welcome.html>
Bob Johnston, School of Psychology, University of Birmingham.
<Http://www.birmingham.ac.uk/psychology/>
Andy Calder, MRC Cognition and Brain Sciences Unit, Cambridge
<http://www.mrc-apu.cam.ac.uk/>
John Vokey, Department of Psychology, University of Lethbridge, Canada.
<http://home.uleth.ca/psy/>
- [28] The Face Recognition Homepage
<http://www.cs.rug.nl/~peterkr/FACE/face.html>