



# REPRESENTACIÓN DE CARAS MEDIANTE EIGENFACES

*Luis Lorente Giménez*

*Proyectista del Dpto. de Teoría del Señal y Comunicaciones, Grupo de Procesado de Imagen,  
Universidad Politécnica de Catalunya  
alorente@gps.tsc.upc.es*

## 1. INTRODUCCIÓN

El proyecto de fin de carrera en el que estoy trabajando desde hace casi un año consiste en el desarrollo de un sistema de reconocimiento de caras. Las tecnologías de reconocimiento de caras (*face recognition technologies* o FRT) son un área de investigación muy activa en los últimos años que engloba varias disciplinas como el procesado de imagen, redes neuronales, *pattern recognition* y visión por ordenador. Tiene varias aplicaciones potenciales, tanto comerciales como de seguridad, como por ejemplo la identificación de fotos de carnets, pasaportes o identificación en tiempo real de personas desde imágenes de cámaras de vigilancia. El reconocimiento de caras, aunque fácilmente realizable por las personas, es difícilmente implementable de una manera totalmente automatizada en ordenador.

El objetivo de un sistema de reconocimiento de caras es, generalmente, el siguiente: dada una imagen de una cara "desconocida" (o imagen de test) encontrar una imagen de la misma cara en un conjunto de imágenes "conocidas" (conjunto de entrenamiento).

Esto, que es tan sencillo de decir, plantea unas terribles dificultades para su implementación en un sistema automático. Las personas reconocemos las caras con gran facilidad, rapidez e incluso en circunstancias adversas de iluminación, orientación y otros factores. Sin embargo los sistemas automáticos de reconocimiento por ordenador todavía están muy lejos de la efectividad del cerebro humano, aunque los resultados son esperanzadores. Las mayores dificultades a las que se puede enfrentar un sistema de reconocimiento de caras son: las variaciones en la expresión de las caras, las variaciones en las condiciones de iluminación y las rotaciones en profundidad. Las rotaciones en profundidad son aquellas en que la cara no gira en el plano de la imagen y por lo tanto parte de la cara puede quedar oculta.

## 2. PARTES DE UN SISTEMA DE RECONOCIMIENTO DE CARAS

A pesar de la gran variedad de sistemas existentes, todo sistema de FR puede dividirse en tres partes:

- preprocesado de las imágenes: consiste en intentar compensar todo lo que puede provocar que dos imá-

genes de la misma cara sean diferentes. Esto incluye normalizar el tamaño y el contraste de la imagen. A veces también se intentan compensar los cambios de iluminación, la rotación y otras características de la imagen que pueden perjudicar los resultados del sistema. Existe una buena cantidad de investigaciones dedicadas únicamente a esta parte, debido a su complejidad.

- extracción de características: en esta fase se extraen una serie de valores característicos de cada imagen, como pueden ser los coeficientes de algún desarrollo, la salida de un filtro, etc. Independientemente de su origen estos valores deben intentar caracterizar con la mayor exactitud cada cara (lo que se considera eficiencia) y, al mismo tiempo, deben tener capacidad de discriminación. Esto significa que los valores extraídos de las imágenes de una cara y los de las imágenes de otras cara deben formar dos grupos lo más compactos y separados posible

- comparación de características: se comparan los valores característicos de la imagen de test (la que se quiere reconocer) con los de las imágenes de entrenamiento y se calcula una medida de semejanza. Los métodos van desde la distancia euclídea (considerando que el conjunto de valores característicos forman un vector) a otros mucho más sofisticados. La imagen de entrenamiento que más semejante sea a la de test se considerará que es de la misma persona

En una primera fase que llamaremos fase de entrenamiento, las dos primeras partes se aplican a las imágenes del conjunto de entrenamiento, para extraer las características que caracterizan a las imágenes de este conjunto. Posteriormente, en la fase de test, las imágenes del conjunto de test que queremos reconocer pasan por las dos primeras partes y la fase de comparación determina finalmente la imagen de entrenamiento que más se parece a la de test.

La parte más importante es la de extracción de características. En ella, se intenta caracterizar una cara con independencia de factores externos como la iluminación, expresión, etc. En general el objetivo es hallar una serie de características de la imagen que pueden ser desde las dimensiones de los elementos de la cara como la boca, ojos, etc. hasta los coeficientes que nos da un filtro o una transformada al aplicarlo a la imagen de la cara. Lo importante es que caractericen con gran fidelidad a cada cara y que sean invariantes a los cambios en

los factores externos que se han mencionado anteriormente.

### 3. TIPOS DE TÉCNICAS DE RECONOCIMIENTO DE CARAS

Las técnicas de reconocimiento de caras se pueden clasificar en tres tipos, en función de los métodos utilizados para la extracción de las características de la cara:

- técnicas basadas en la extracción de características geométricas de la cara como las posiciones relativas y dimensiones de ciertos elementos de la cara (cejas, ojos, nariz, boca y contorno de la barbilla principalmente)
- técnicas basadas en procedimientos de *template matching* (emparejamiento de plantillas), en las que zonas de la imagen son comparadas con zonas equivalentes de otra imagen utilizando alguna distancia (pe la distancia Euclídea) para poder calibrar su grado de semejanza o discrepancia. Los sistemas que utilizan este tipo de técnicas carecen de la fase de extracción de características, ya que utilizan directamente la información de niveles de gris (o color) de los píxeles de la imagen
- técnicas basadas en la aplicación de transformadas: las características de la imagen son los coeficientes resultantes de aplicar algún procedimiento numérico o alguna transformada, como la transformada Karhunen-Loeve

A pesar de esta sencilla división, existen gran cantidad de técnicas y no resulta sencillo escoger entre ellas. Al comenzar el proyecto se definieron una serie características que debía cumplir el sistema y que limitaron las alternativas entre las que escoger. Se decidió buscar un sistema que utilizara una sola vista frontal de cada cara como conjunto de entrenamiento. En otras palabras, el sistema debía ser capaz de poder identificar a una persona teniendo sólo una imagen (una vista frontal) de ésta. Sin embargo se pueden encontrar técnicas de reconocimiento de caras que utilizan vistas laterales o perfiles, información 3D de la cara u otros tipos de información.

Tras una intensa etapa de documentación y búsqueda se decidió implementar un sistema de reconocimiento basado en la representación mediante *eigenfaces*. Una de las razones es que este tipo de sistemas tiene probablemente el mejor compromiso entre complejidad, rapidez de ejecución y resultados.

### 3. REPRESENTACIÓN DE CARAS MEDIANTE EIGENFACES

#### 3.1. Introducción

El origen de los sistemas de FR que utilizan *eigenfaces* se remonta a unos trabajos de Sirovich y Kirby ([1]). El objetivo de Sirovich y Kirby era caracte-

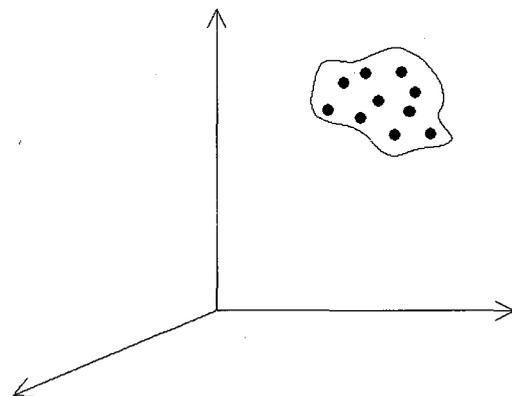
rizar un conjunto de caras con un mínimo número de parámetros. Para ello, se utilizaba el análisis de componentes principales o PCA (*principal components analysis*) para generar unas imágenes semejantes a caras llamadas *eigenpictures*. Las imágenes de caras se caracterizaban como una combinación lineal de *eigenpictures*. Fueron Turk y Pentland ([2]) quienes utilizaron esta representación de la imagen, mucho más compacta que los valores de los píxeles de la imagen, para implementar un sistema de FR, además de rebautizar las *eigenpictures* como *eigenfaces*.

El primer paso para comprender los sistemas de reconocimiento basados en *eigenfaces* es conocer la filosofía que se oculta detrás del PCA, que se utiliza para generarlas, y que permite un mejor entendimiento de la representación de las imágenes de caras mediante *eigenfaces*.

#### 3.2. Motivación

Consideremos el conjunto de todas las imágenes posibles cuyas dimensiones son  $w$  píxeles de ancho por  $h$  píxeles de alto. El número de componentes de estas imágenes es  $w \cdot h$ . Si concatenamos las filas de estas imágenes podemos considerarlas como vectores de dimensión  $w \cdot h$ . Utilizando esta representación vectorial, el conjunto de imágenes forman un espacio vectorial de dimensión  $w \cdot h$ . Este espacio se puede generar a partir de la base canónica  $(1,0,0,\dots,0)$ ,  $(0,1,0,0,\dots,0)$ , ...,  $(0,0,\dots,0,1)$ .

Consideremos ahora el conjunto de las imágenes de caras. En relación al resto de todas las posibles imágenes son todas relativamente semejantes: todas tienen los mismos elementos, situados de forma semejante, con textura parecida, etc. Además, los píxeles están altamente correlados con sus vecinos. Es razonable, por lo tanto, suponer que el conjunto de las imágenes de caras ocupa sólo una pequeña región de este espacio. Una consecuencia de esto es que la representación de las



**Figura 1.** Ejemplo simplificado del espacio de las imágenes (en este caso de dimensión 3) y región ocupada por las imágenes de caras.

imágenes de caras utilizando la base del espacio de las imágenes es claramente ineficiente.

Además, debido a que todas las imágenes de caras son relativamente semejantes, podemos suponer que las imágenes de caras están en una zona reducida del espacio de las imágenes (llamémoslo el espacio de las caras). En ese caso, podríamos intentar hallar una base de vectores cuyo número sería mucho menor que el de una base del espacio de las imágenes. A pesar de que no existe ninguna evidencia al respecto, la rapidez con la que el cerebro humano es capaz de reconocer las caras sugiere que su representación en el cerebro es de una dimensionalidad baja, aunque nadie sabe cual es ni como se maneja la información. En la Figura 1 se muestra un ejemplo simplificado en el que el espacio de las imágenes es de dimensión tres (imágenes de tres pixels). Finalmente, llegamos a la conclusión debería ser posible generar las imágenes de caras (o al menos una muy buena aproximación) con muchos menos vectores que los necesarios para representar el espacio de las imágenes completo.

El objetivo pasa a ser encontrar esta base de vectores. El PCA (análisis de componentes principales o *principal components analysis*) permite hallar una nueva base con un máximo de eficiencia. El PCA fue desarrollado por estadísticos, aunque también ha sido formulado en el campo de las redes neuronales. En este proyecto se utiliza la concepción estadística del PCA.

### 3.3. El PCA

Como ya se ha dicho, las componentes (los pixels) de las imágenes de caras están altamente correladas entre sí. El PCA se basa precisamente en las propiedades estadísticas de las imágenes y es un método óptimo para reducir el número de dimensiones necesarias para representar un conjunto de vectores (en nuestro caso, imágenes de caras).

Consideremos un vector aleatorio  $\mathbf{X}$  de dimensión  $n$ , con su correspondiente media  $\mu$

$$\mu = E\{\mathbf{X}\}$$

La matriz de covarianza se define como

$$\Sigma = E\{(\mathbf{X} - \mu)(\mathbf{X} - \mu)'\} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix}$$

donde  $\sigma_{ij}$  representa la covarianza entre el componente número  $i$  y el número  $j$  del vector  $\mathbf{X}$ . El análisis de componentes principales consiste en hallar los vectores propios de esta matriz y expresar  $\mathbf{X}$  en función de estos vectores. De esta manera, el vector aleatorio  $\mathbf{X}$  se puede

representar sin error mediante una combinación lineal de vectores de la forma

$$\mathbf{X} = \sum_{i=1}^n y_i \mathbf{A}_i$$

donde los vectores  $\mathbf{A}_i$  son los vectores propios de la matriz de covarianza. La cualidad más importante de esta representación es que si queremos representar  $\mathbf{X}$  con sólo  $m$  (menor que  $n$ ) componentes la mejor elección posible en términos del error que se comete son los  $m$  vectores propios de la matriz de covarianza con mayores valores propios asociados.

### 3.4. Aplicación del PCA para la representación de caras: las eigenfaces

Como se acaba de ver, el PCA se reduce básicamente a encontrar los vectores propios de la matriz de covarianza  $\Sigma_{\mathbf{X}}$  del vector al que aplicamos el PCA. Apliquemos toda esta teoría a nuestro problema particular de reconocimiento de caras: nuestro objetivo es caracterizar un conjunto de caras (que serán las del conjunto de entrenamiento) en un espacio de menor dimensionalidad.

Tal y como se ha expuesto anteriormente, consideremos las imágenes de caras de este conjunto (de anchura  $w$  y altura  $h$ ) como realizaciones del vector aleatorio  $\mathbf{X}$  de dimensión  $w \cdot h$ , con su correspondiente vector media  $\mu$ . A este vector, al que llamaremos vector cara, deseamos aplicarle el PCA.

La media de este vector cara no la podemos obtener al no conocer la función de densidad de probabilidad, pero la podemos estimar a partir las imágenes que tenemos:

$$\mu \approx \frac{1}{N} \sum_{i=1}^N \mathbf{X}_i$$

La matriz de covarianza tampoco la conocemos pero se también se puede estimar:

$$\Sigma_{\mathbf{X}} \approx \frac{1}{N} \sum_{i=1}^N (\mathbf{X}_i - \mu)(\mathbf{X}_i - \mu)' = \frac{1}{N} \mathbf{M}_{\mathbf{X}} \mathbf{M}_{\mathbf{X}}'$$

Llegados a este punto, en teoría sólo queda el cálculo de los vectores propios de la matriz de covarianza, para lo que existen varios algoritmos automáticos fácilmente implementables. Pero la implementación de este algoritmo plantea un problema: la cantidad memoria necesaria para almacenar la matriz es demasiado grande. Si los vectores cara de la base de entrenamiento son de dimensión  $w \cdot h$ , las dimensiones de la matriz de covarianza estimada  $\Sigma_{\mathbf{X}}$  serán de  $w \cdot h$  de ancho y  $w \cdot h$  de alto. Por ejemplo, si utilizamos imágenes de entrenamiento con unas dimensiones contenidas de  $w=128$  y  $h=128$ , el número de elementos de la matriz de covarianza

sería de  $128^4$ , o sea más de 268 millones, lo que está fuera del alcance de las memorias de los ordenadores actuales.

La solución es aplicar una SVD ( descomposición en valores singulares o *singular value decomposition* ) a la matriz  $M_x$ . Esta operación descompone una matriz en un producto de tres matrices

$$M_x = UA^{1/2}V^T$$

donde si  $M_x$  es una matriz de  $w \times h$  filas y  $N$  columnas con  $w \times h$  mayor que  $N$  (lo que siempre será nuestro caso) la matriz  $U$  es de las mismas dimensiones y las matrices  $V$  y  $A^{1/2}$  son cuadradas de dimensiones  $N$  por  $N$ . Además, la matriz  $A^{1/2}$  es diagonal.

La propiedad que nos es útil de la SVD es que las  $N$  columnas de la matriz  $U$  son  $N$  vectores propios de la matriz  $XX^T$ , y los elementos de la matriz  $A^{1/2}$  son las raíces cuadradas de los valores propios correspondientes a estos vectores propios. Si recordamos la estimación de la matriz  $\Sigma_x$  vemos que esta no era exactamente  $XX^T$ , sino que además se dividía este producto por  $N$ . Pero esto no importa ya que los vectores y valores propios de una matriz y los de la misma matriz dividida por un escalar son iguales, salvo por el hecho de que los valores propios también quedan divididos por el mismo número.

Este método está al alcance de un ordenador, ya que la matriz  $M_x$  es de dimensiones  $w \times h$  por  $N$ , y  $N$  (que es el número de caras del conjunto de entrenamiento) suele estar alrededor de un centenar, lo que la convierte en una matriz manejable por un ordenador. Sin embargo este método no calcula todos los vectores propios de  $S_x$  ya que esta matriz, al ser real y simétrica, tiene siempre  $w \times h$  vectores propios (en la práctica muchos más que  $N$  que es el número de imágenes del conjunto de entrenamiento). Pero se puede demostrar que los  $N$  vectores propios conseguidos con la SVD son los únicos vectores propios de  $\Sigma_x$  que pueden tener un valor propio distinto de cero. Desde el punto de vista teórico del PCA, esto quiere decir que el resto de vectores propios (los que siempre tienen un valor propio igual a cero) no capturan ninguna información del vector  $X$  al que estamos aplicando el PCA, y por lo tanto los eliminaríamos del análisis tal y como se ha explicado en el punto anterior.

Volviendo a nuestro problema de representación, todo esto significa que los  $N$  vectores propios que genera la SVD son capaces de caracterizar a los  $N$  vectores correspondientes al conjunto de imágenes de caras de entrenamiento, y el resto de vectores propios sólo nos podrían servir para abarcar el resto del espacio original de  $X$  (el espacio de las imágenes de  $w$  por  $h$  pixels), pero no nos servirían para representar las caras. De entre estos  $N$  vectores propios el número de ellos que contienen información útil (valor propio mayor de cero) es igual al número de vectores linealmente independientes del conjunto de entrenamiento menos uno (ya que al restar la media se reduce siempre en uno el número de vectores linealmente independientes).

De esta manera obtenemos los deseados vectores propios que definen nuestro nuevo espacio, el espacio de las caras. Estos vectores tienen realmente apariencia de caras, y por ello han sido llamados primero *eigenpictures* (imágenes propias) y posteriormente *eigenfaces* (caras propias). En la figura 2 se muestran las 5 primeras *eigenfaces* generadas con un conjunto de 57 imágenes de entrenamiento.



**Figura 2.** Cara media y 5 primeras *eigenfaces* de un total de 56 generadas a partir de 57 imágenes de caras sin barba ni gafas. La posición de los ojos y la boca está normalizada y los componentes vectores propios han sido remapeados en el rango  $[0,255]$  para poder visualizarlos

### 3.5. Reducción de la dimensionalidad

Llegados a este punto, la situación es la siguiente: tenemos un conjunto de  $N$  imágenes de caras  $X_i$  ( que utilizamos como conjunto de entrenamiento ) que queremos representar en un espacio de menor dimensión. Para ello hemos considerado estas imágenes como realizaciones del proceso aleatorio vectorial "imágenes de caras" y hemos aplicado el PCA a este vector. Como resultado, obtenemos  $N$  vectores propios (*eigenfaces*) ortonormales y con  $N-1$  de ellos se pueden generar sin error las  $N$  caras del conjunto de entrenamiento. En la figura 3 se muestra de una manera gráfica parte de la representación de una de las caras utilizadas para generar las *eigenfaces* de la figura 2.

Para hallar las coordenadas  $Y_i$  de una cara  $X_i$  sobre este nuevo espacio no tenemos más que proyectar  $X_i$

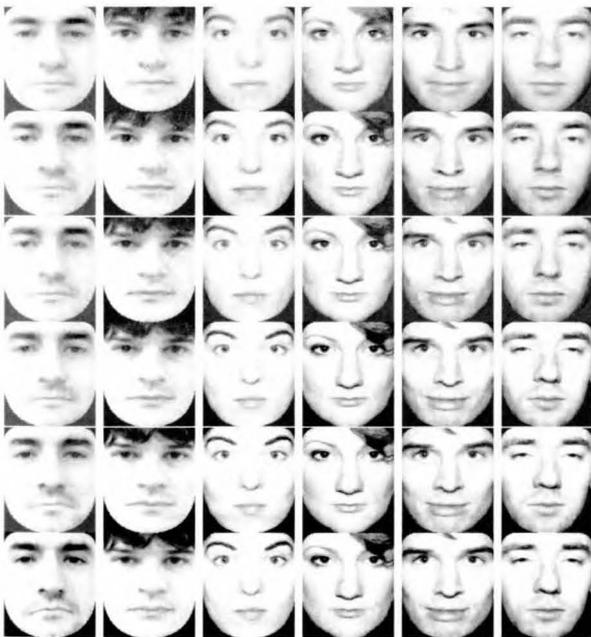


**Figura 3.** Representación de una cara a partir de las *eigenfaces*: a la cara se la resta la cara media y esta diferencia se puede expresar como una combinación lineal de las *eigenfaces*. Para que la representación sea exacta la imagen debe estar incluida en el conjunto al que se la aplica el PCA (el conjunto de entrenamiento)

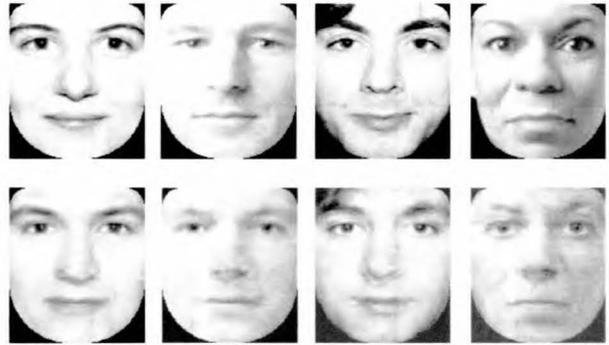
sobre las *eigenfaces*. Para hallar la proyección de  $\mathbf{X}_i$  sobre cada *eigenface* bastará con calcular el producto escalar, ya que las *eigenfaces* son ortonormales. Por lo tanto, si las columnas de la matriz  $\mathbf{A}$  son las *eigenfaces*, hallaremos las nuevas coordenadas en el espacio de las caras de la siguiente manera:

$$Y_i = P^T X_i$$

Al haber N-1 *eigenfaces* la dimensión de los vectores  $\mathbf{Y}_i$  es N-1 (recordemos que N es el número de imágenes de entrenamiento). Esto supone una fuerte reducción de dimensionalidad frente al espacio de las imágenes (w\*h) pero esto no es ninguna sorpresa ya que como ya se ha comentado obviamente para representar a N vectores a los que se ha restado su media basta con N-1 vectores. La verdadera reducción de dimensionalidad de la representación mediante *eigenfaces* consiste en utilizar sólo las *eigenfaces* con mayor valor propio asociado. Recordemos que el valor propio asociado a una *eigenface* es igual a la variancia de la proyección de las caras sobre esa *eigenface*. De esta manera se puede reducir notablemente el número de dimensiones de la representación sin perder apenas calidad en la representación de las caras. En este aspecto, las *eigenfaces* consiguen la máxima eficiencia que se puede conseguir ya que las primeras *eigenfaces* (las de mayor valor propio asociado) consiguen capturar la información más importante de las caras, la que explica la máxima variancia, mientras que las últimas apenas aportan información (sus valores propios son muy bajos). Dicho de otra manera: la mejor representación posible de un conjunto de N imágenes de caras utilizando sólo M dimensiones (con  $M < N$ ) se consigue proyectándolas



**Figura 4.** Representaciones de 6 imágenes de caras utilizando las primeras 8, 16, 24, 32, 40, 48 y 56 *eigenfaces* de un total de 56 *eigenfaces*, generados a partir de un PCA de 57 imágenes de caras y ordenadas de mayor a menor valor propio. La reconstrucción incluye la suma de la cara media para poder apreciar mejor la calidad



**Figura 5.** Imágenes de personas no incluidas en el conjunto de entrenamiento representadas con un conjunto de 56 *eigenfaces*. El error es considerable debido a que el número de *eigenfaces* es demasiado bajo

sobre las M *eigenfaces* con mayor valor propio. En la figura 5 se puede ver la evolución de la calidad en la representación de algunas caras del conjunto de entrenamiento utilizado para generar las *eigenfaces* de la figura 2. Como se puede apreciar, con un número muy bajo de *eigenfaces* se consigue una representación con una notable calidad, lo que demuestra que las primeras *eigenfaces* consiguen capturar la mayor parte de la información de las caras.

Como se ha dicho anteriormente, si las *eigenfaces* se generan a partir de un conjunto de entrenamiento suficientemente grande de imágenes entonces son capaces de representar con fidelidad imágenes de caras de personas que no están en el conjunto de entrenamiento. Para ello el tamaño del conjunto de entrenamiento debe ser como mínimo de unas 100 imágenes. En el presente proyecto no se disponía de tal número y por lo tanto no se consigue una buena representación. En la figura 5 se muestran algunos ejemplos de reconstrucción de caras de personas no incluidas en el conjunto de entrenamiento.

Sin embargo, lo que siempre se consigue es una buena representación de imágenes de caras que tienen alguna otra imagen (de la misma cara o persona) en el conjunto de entrenamiento, aunque estas tengan variaciones en la expresión respecto a la imagen del conjunto de entrenamiento. En estos casos, aunque la imagen representada con las *eigenfaces* tenga un error considerable la identidad de la cara siempre se preserva con gran fidelidad, lo que es especialmente importante de cara a su aplicación al reconocimiento de caras. En la figura 6 vemos unos ejemplos que ilustran este hecho.

La figura 4 da una idea visual de la información capturada por las primeras *eigenfaces* pero ¿hasta qué punto consiguen concentrarse esta información en un bajo número de *eigenfaces*? En la figura 7 se muestra la gráfica de la evolución de los valores propios (iguales a la variancia de la proyección de las caras sobre la *eigenface* asociada), ordenados de mayor a menor. La gráfica sigue una evolución semejante a una exponencial,



**Figura 6.** Primera fila: imágenes pertenecientes al conjunto de entrenamiento. Segunda fila: imágenes de las mismas personas con cambios de expresión. Tercera fila: imágenes de la segunda fila representadas a partir de las eigenfaces generadas con las imágenes de la primera

gráfica sigue una evolución semejante a una exponencial, lo que demuestra la gran concentración de variancia que se produce en las primeras *eigenfaces*. Esto se refleja claramente en la calidad obtenida en la representación. Para expresar esta calidad numéricamente debemos hallar una medida del error cometido al representar una cara. Si la representación de una cara  $X$  con  $M$  *eigenfaces* (de un total de  $N$ ) es

$$\tilde{X} \approx \mu + \sum_{i=1}^M y_i A_i$$

donde  $\mu$  es el vector media  $A_i$  son las eigenfaces ordenadas de mayor a menor valor propio y  $y_i$  la proyección de  $X$  sobre  $A_i$ , el error de representación se define como

$$e = \left( \frac{\|X - \tilde{X}\|}{\|X\|} \right)^2$$

El error está normalizado y su rango de valores va de cero a uno. Sin embargo, debido a que la media es una buena estimación de todas las caras en la práctica los valores siempre son mucho más reducidos. La evolución de sus valores se ve en la figura 8 para los tres casos de representación vistos. Como se podía esperar, esta gráfica confirma que con un bajo número de *eigenfaces* se consiguen representaciones con bajos porcentajes de error.

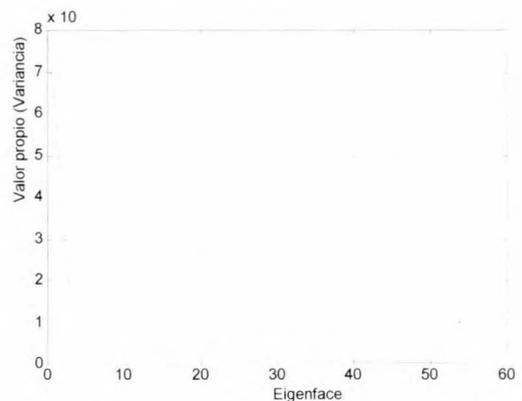
#### 4. EL SISTEMA DE RECONOCIMIENTO DE CARAS

En el capítulo anterior se ha visto la conveniencia de representar las imágenes de caras mediante *eigenfaces*. En este capítulo se verá como se aplica esta representación al reconocimiento de caras. Recordaremos el planteamiento del problema a resolver por un sistema de reconocimiento de caras: dado un conjunto de imágenes de caras que llamaremos conjunto de entrenamiento, el

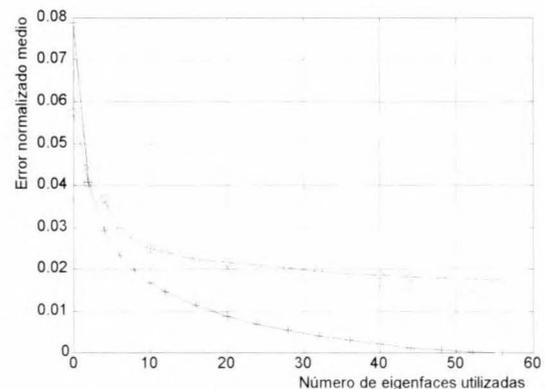
sistema debe ser capaz de emparejar una nueva imagen con la imagen del conjunto de entrenamiento que pertenece a la misma persona.

#### 4.1. Etapas de entrenamiento y de test

Para que el sistema esté preparado para realizar la parte de comparación es necesario disponer de las características de las imágenes de entrenamiento. Por ello es necesario realizar una vez una etapa de entrenamiento. Esta consiste en aplicar las dos primeras fases a las imágenes del conjunto de entrenamiento. A partir de entonces se puede efectuar el reconocimiento con un conjunto de imágenes de test. El esquema general se muestra en la Figura 9.



**Figura 7.** Gráfica de los valores propios asociados a las eigenfaces ordenados de mayor a menor. La disminución es muy rápida, lo que indica una gran concentración de la variancia de las imágenes en las primeras eigenfaces



**Figura 8.** Error de representación de las caras del conjunto de entrenamiento (media de las 57) en función del número de eigenfaces utilizadas para representar imágenes del conjunto de entrenamiento (+), otras imágenes de personas incluidas en el conjunto de entrenamiento con diferentes expresiones (X) y imágenes de caras "desconocidas" (O). El error decae rápidamente con las primeras eigenfaces y la reducción de error debida al resto de eigenfaces es mínima

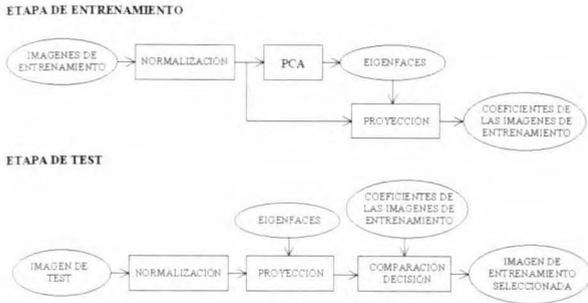


Figura 9. Esquema del sistema de reconocimiento de caras.

## 4.2. Implementación de las partes del sistema de reconocimiento

A continuación se describe la implementación concreta de las partes del sistema de reconocimiento.

### 4.2.1. Normalización o preprocesado

La fase de normalización es de una gran importancia en los sistemas que utilizan *eigenfaces*. Como se ha visto anteriormente, la potencia de las *eigenfaces* reside en su gran capacidad para caracterizar con un mínimo error una imagen de una cara. Esta capacidad siempre es muy alta con las imágenes con las que se han generado las *eigenfaces*, pero con otras imágenes de caras (pe. las que se tendrán que reconocer, que llamaremos conjunto de test) depende mucho del procesado que se le aplique a la imagen antes de proyectarla sobre las *eigenfaces*.

Nuestra implementación de esta fase intenta compensar las diferencias en los siguientes aspectos:

- **tamaño:** se normaliza la distancia entre los ojos y entre los ojos y la boca. En otras palabras: se normaliza la altura y la anchura de la cara. Como se ha comentado antes, esto supone una pequeña distorsión en la forma de la cara, pero los resultados demuestran una mejora. La localización de los ojos y la boca se realiza manualmente al no ser un objetivo del proyecto, pero existen métodos automáticos muy eficientes ([3])
- **rotación en el plano de la imagen:** utilizando las posiciones de los dos ojos, se efectúa una rotación de la imagen de manera que los dos ojos queden en horizontal
- **zona de la cara utilizada:** la cara se recorta de manera que no aparezcan zonas del fondo de la imagen ni zonas del pelo, cuyo aspecto es muy variable y que perjudicaría la robustez del sistema
- **contraste y nivel de iluminación:** se expande el rango de niveles de gris de la imagen al máximo. Al estar representadas el 8 bits, esto hace que el valor mínimo de los pixels pase a cero y el máximo a 255. Esta etapa sólo debería realizarse después de la de recorte de la cara, de lo contrario el pelo y el fondo podrían distorsionar los resultados

La normalización del tamaño y de la rotación se consigue realizando un nuevo muestreo sobre la imagen en el que las posiciones de los ojos y la boca sean fijas. El proceso se puede ver en la figura 10.

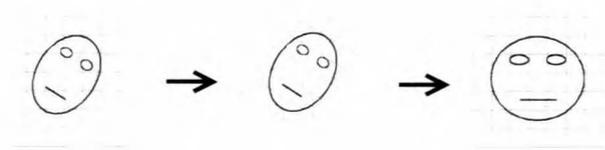


Figura 10. El proceso de remuestreo.

### 4.2.2. Cálculo de las eigenfaces

En esta fase se realiza el PCA con el que se extraen las eigenfaces, mediante los siguientes pasos:

- se calcula de la cara media y se resta de todas las caras normalizadas
- se forma la matriz  $M_x$  cuyas columnas son las imágenes del conjunto de entrenamiento normalizadas
- se realiza la SVD de la matriz  $M_x$ , dando como resultado tres matrices, una de ellas conteniendo los vectores propios (las *eigenfaces*) y otra las raíces cuadradas de los valores propios
- se guardan las *eigenfaces* así como la proyección de las imágenes de entrenamiento sobre ellas (coeficientes de las imágenes de entrenamiento)

### 4.2.3. Proyección sobre las eigenfaces

Las *eigenfaces* forman un conjunto ortonormal. Por lo tanto, para hallar la proyección de una imagen sobre ellas basta con realizar el producto escalar de la imagen sobre cada una de las *eigenfaces*.

### 4.2.4. Fase de comparación/decisión

El objetivo es determinar que imagen del conjunto de entrenamiento es más parecida a la imagen de test, a partir de sus representaciones mediante las eigenfaces (sus proyecciones). Para ello se compara el vector  $y_{TEST}$  formado por las proyecciones de la imagen de test sobre las eigenfaces con cada uno de los vectores  $y_{ENTI}$ . El criterio que se utiliza es el de la menor distancia euclídea, es decir, menor  $|y_{TEST} - y_{ENTI}|$



Figura 11. Ejemplo de la base de datos de Berna

## 5. RESULTADOS

### 5.1. Imágenes utilizadas

Para probar el sistema se utilizaron dos conjuntos de imágenes. La base de datos de Berna consiste en imágenes de 30 personas (todas de sexo masculino). De cada persona se seleccionaron una vista frontal, otra vista frontal con ligeros cambios de expresión y una vista lateral con un ligera rotación en profundidad. Varias personas llevan gafas y algunas barba. En la figura 11 vemos un ejemplo.

La base de datos de Stirling consiste en imágenes de 17 mujeres y 15 hombres, sin gafas ni barba. De cada persona se seleccionaron tres vistas equivalentes a las anteriores pero con un grado de dificultad más elevado que en la anterior base de datos: en la segunda vista frontal los cambios de expresión son más acusados y en la vista lateral la rotación hace que parte de la cara no se vea.

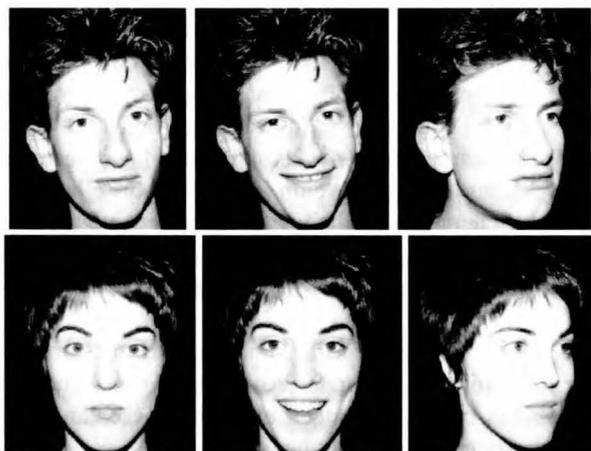


Figura 12. Ejemplo de imágenes de la base de datos de Stirling

### 5.2. Pruebas

Se realizaron dos pruebas. En ambas se utilizaron como conjunto de entrenamiento las 62 vistas frontales de personas diferentes.

En la primera prueba se utilizaron las 62 segundas vistas frontales como conjunto de test, mientras que en la segunda se utilizaron las vistas laterales. Los resultados se muestran en la figura 13. También se indican los resultados parciales con las imágenes de cada base de datos (con el conjunto de entrenamiento total de 62 imágenes).

Como se puede ver, los cambios de expresión apenas afectan a la eficacia del sistema. Sin embargo, las rotaciones en profundidad producen tal cambio en la imagen que la representación mediante *eigenfaces* deja

PRUEBA	CONJUNTO DE ENTREN.	CONJUNTO DE TEST	TOTAL	BERNA	STIRLING
Prueba 1	62 frontales (1 por pers.)	62 frontales (1 por pers.)	60/62	30/30	30/32
Prueba 2	62 frontales (1 por pers.)	62 laterales (1 por pers.)	29/62	25/30	4/32

Figura 13. Resultados del sistema

de ser adecuada y se producen errores. Particularmente con las imágenes de la base de datos de Stirling se producen muy malos resultados debido a que la rotación es muy pronunciada

## 6. CONCLUSIONES

El sistema que se ha presentado obtiene unos buenos resultados, pero con ciertas limitaciones. En concreto las rotaciones en profundidad producen cambios que la representación mediante *eigenfaces* no puede asimilar. Este problema es común a casi todos los sistemas de reconocimiento, y de momento no se ha conseguido resolver utilizando una sola imagen por persona en el conjunto de entrenamiento. Las únicas soluciones son utilizar un múltiples vistas de cada persona en el conjunto de entrenamiento, pero estas vistas normalmente no están disponibles y se necesitan muchas para poder abarcar todas las rotaciones posibles, necesiándose además muchas más memoria y cálculos independientemente del sistema utilizado.

Otro problema que no se ha tratado en este artículo pero que también produce dificultades son los cambios de iluminación, en concreto de la dirección de iluminación. En algunos trabajos de nuevo se recurre a tener múltiples vistas de cada persona (una para cada condición de iluminación diferente) para poder saber compensar esos cambios, pero los problemas siguen siendo los mismos, siendo el principal que normalmente sólo se va a disponer de una vista (frontal) para el conjunto de entrenamiento.

Como conclusión se puede decir que aunque están en continuo avance y los resultados son aceptables, los sistemas de reconocimiento de caras todavía deben superar algunas limitaciones.

## 7. BIBLIOGRAFÍA

- [1] L. SIROVICH AND M. KIRBY, "Low-dimensional procedure for the characterization of faces", J. Opt. Soc. Am. A, vol. 4, num. 3, 1987, pp. 519-524.
- [2] M. TURK AND A. PENTLAND, "Eigenfaces for recognition", J. Cognitive Neuroscience, vol. 3, num. 1, 1991.
- [3] A. PENTLAND, B. MOGHADDAM AND T. STARNER, "View-based and modular eigenspaces for face recognition", IEE Conf. On Computer Vision & Pattern Recognition, Seattle, WA, July 1994.