

## OPTIMIZACIÓN E IMPLEMENTACION DE UN SISTEMA RECONOCEDOR DE ROSTROS

## OPTIMIZACIÓN AND IMPLEMENTATION OF A SYSTEM RECOGNITION OF FACES

Jorge Alberto Del Carpio Salinas<sup>1</sup>, José Antonio Huamán Layme<sup>2</sup>

### RESUMEN

*Los Sistemas de seguridad controlados por características de tipo biométrico experimentan un creciente interés frente a alternativas tradicionales. Este éxito se debe en gran medida a que, cuando una persona pretende acceder a un sistema, la decisión se toma en base a características específicas de esa persona, y no en base a lo que conoce o a lo que posee (Tarjetas magnéticas, claves, etc.); En los últimos años el gran desarrollo de los sistemas de información unido al abaratamiento y masificación de computadoras y sensores ha provocado un creciente interés por sistemas que permitan establecer la identidad de un individuo en forma automatizada. Frente a esto, en este trabajo se describe e implementa un reconocedor de rostros utilizando las técnicas más exitosas en el campo de la biometría basadas en métodos Estadísticos como son: Descomposiciones del tipo Eingenface y Modelo Oculito de Markov Embebido (HMME). El primer método genera una representación lineal reducida de las imágenes de rostros de manera que cada rostro sea proyectado en un espacio de dimensionalidad reducida donde se llevara a cabo el reconocimiento. El segundo método genera un modelo de estados, basados en la correspondencia entre secuencia de observación y secuencias de estados. Para ello se ha empleado una base de datos de rostros obtenida con estudiantes de la Universidad Nacional de Ingeniería UNI, una cámara, una tarjeta digitalizadora y el sistema se implemento casi en tiempo real empleando el C++.*

*Palabras Clave.- Base de datos, Biometría, Cámara, Sensores, Eigenfaces, Métodos estadísticos, Modelo de Markov.*

### ABSTRACT

*The Security systems controlled by characteristics of biométrico type as opposed to experience an increasing interest traditional alternative. This success must to a great extent to that, when a person tries to accede to a system, the decision is taken on the basis of specific characteristics of that person, and on the basis of which does not know or which has (Magnetic cards, key, etc.); In the last years the great development of the information systems together with the reduction in price and masificación of computers and censors has caused an increasing interest by systems that allow to establish the identity of an individual in automated form. As opposed to this, in this work a recognition of faces is described and implemented using the most successful techniques in the field of the biometry based on methods Statistical as they are: Decompositions of the type Eingenface and Hidden Model of Markov Embedded (HMME). The first method generates linear a representation reduced of the images of faces so that each face is projected in a space reduced where the recognition*

---

<sup>1</sup>Dr. Docente investigador de la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Ingeniería, <sup>2</sup>Ing. Egresado de la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Ingeniería.

*was carried out. The second method generates a model of states, based on the correspondence between sequence of observation and sequences of states. For it use a data base of faces with students of the UNI, a camera, card digitizer and was developed software in C++.*

*Key words.- Statistical data base, Biometry, Camera, Sensors, Eigenfaces, Statistical methods, Model of markov.*

## INTRODUCCIÓN

El reconocimiento facial es una tecnología biométrica emergente, debido a que se pueden utilizar cámaras y base de datos de rostros con la que se puede operar pasivamente, es decir, sin que las personas se puedan dar cuenta de que están siendo examinados, estas pueden ser usados en tareas de verificación de identidad.

De acuerdo a estudios y pronósticos [1], algunas aplicaciones en las cuales los sistemas biométricos en base a rostros poseen superioridad, es en aplicaciones gubernamentales en los cuales se necesita identificar individuos utilizando base de datos de rostros de los mismos (DNI, pasaporte, licencia de conducir, etc.), algunos ejemplos de estas aplicaciones son el control de identidad en elecciones.

Aunque aun no se ha encontrado una solución óptima, existen tres marcos teóricos en base a los cuales se ha diseñado con éxito sistemas de reconocimiento de rostros, estos son:

El análisis estadístico, más concretamente los métodos que se trataran en este artículo, las redes neuronales [2] que son de uso general para la resolución de problemas de reconocimiento de patrones. Los métodos deformables que agrupados bajo este nombre se encuentran métodos muy diversos.

El éxito de las descomposiciones de tipo Eigenspace es debido a que el espacio de rostros (dado por los vectores reducidos) tiene menor dimensionalidad que el espacio de imágenes (dado por el número de píxeles en las imágenes) esto hace posible realizar el reconocimiento debido a que disminuye la data original y hace mas viable el cálculo computacional.

Un HMM uni-dimensional, puede ser generalizado para dar una apariencia a una estructura de dos dimensiones, que permitir a cada estado HMM uni-

dimensional a ser un HMME. De esta manera, el HMME consiste en un conjunto de súper estados con un conjunto de estados embebidos.

Los súper estados pueden entonces ser usados para un modelo de datos de dos dimensiones a lo largo de una dirección y con un modelo HMM embebido de los datos a lo largo de otra dirección.

## METODO DE DESCOMPOSICIÓN

### PCA (análisis de componentes principales)

Esta técnica denominada también Eigenfaces reduce el número de operaciones haciendo computacionalmente abordable el problema de reconocimiento. La idea es aprovechar la redundancia del conjunto de entrenamiento. Se trata de una aplicación particular del método que en estadística se conoce como análisis de componentes principales [3] y en teoría de señales como transformada de Karhunen-Loeve (KLT).

La formulación matemática del problema es el siguiente: Sea  $N$  el número de píxeles de cada imagen y  $M$  el número de imágenes en la base de datos de entrenamiento. La base de datos puede expresarse entonces de forma compacta como  $X = [X_1 \dots X_M]$ , donde cada  $X_i$  (vector de  $N$  componentes) es el resultado de reordenar la imagen  $i$ -ésima del conjunto de entrenamiento como un vector  $v_i$  y restarla posteriormente la media.

$$\bar{v} = M^{-1} \sum_{j=1}^M v_j \quad (1)$$

Con este método lo que se hace es intentar conservar la mayor parte de la información que se tiene de  $X$  en otra matriz  $u = [u_1 \dots u_{M'}]$   $M' < M$  y de manera que el conjunto que forman los vectores columna sea ortonormal. El  $k$ -ésimo vector  $u_k$ , es elegido tal que:

$$\lambda_k = [(u_k^T X_1)^2 + (u_k^T X_2)^2 \dots (u_k^T X_M)^2] / M \quad (2)$$

Es máximo, sujeto a.

$$u_l^T u_k = \begin{cases} 1 & l = k \\ 0 & \text{otro} \end{cases} \quad (3)$$

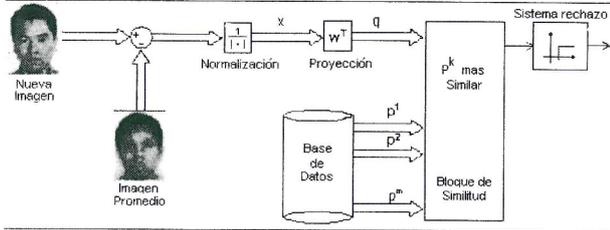


Fig. 1 Diagrama de reconocimiento por PCA.

Los vectores  $u_k$  y escalares  $\lambda_k$  son los Eigenvectores y Eigenvalores, respectivamente de la matriz de covarianza  $C$ .

$$C = (X_1 X_1^T + X_2 X_2^T + \dots + X_M X_M^T) / M = A A^T \quad (4)$$

Donde la matriz  $A = [X_1 X_2 \dots X_M]$ . La matriz  $C$ , sin embargo es de dimensión grande para tamaños típicos de imágenes. Se requiere un método computacional eficiente [4] para encontrar estos Eigenvectores, el cual se describe a continuación:

Sí el número de puntos de datos en el espacio imagen, es menor que la dimensión del espacio ( $M < N$ ) habrá únicamente  $M - 1$ , en lugar de  $N$  eigenvectores.

Considere los eigenvectores  $s_i$  de  $A^T A$  tal que

$$A^T A s_i = \lambda_i s_i \quad (5)$$

Multiplicando ambos lados por  $A$ , tenemos:

$$A A^T A s_i = \lambda_i A s_i \quad (6)$$

Donde podemos ver que  $A s_i$  son los eigenvectores de  $C = A A^T$ . Siguiendo con el análisis, construimos la matriz  $L = A^T A$  de tamaño  $M \times M$  donde  $L_{m \times m} = X_m^T X_m$  y se encuentra los  $M$  eigenvectores

$s_l$  de  $L$ .

Estos vectores determinan combinaciones lineales del conjunto de entrenamiento para formar las eigenfaces  $u_l$ .

$$u_l = (s_{l1} X_1 + s_{l2} X_2 + \dots + s_{lM} X_M) \quad (7)$$

Para  $l = 1$  hasta

### Método de fisher

Al igual que el análisis de componentes principales PCA, el método de Fisher crea una matriz de proyección formada por un conjunto de rostros, que ahora llamaremos **Fisherfaces**, donde los rostros son proyectados y de esta forma representada en un nuevo espacio de menor dimensionalidad. La diferencia con PCA es que ahora no se busca mantener la topología del espacio original. El nuevo objetivo es llevar los rostros a un espacio donde se pueda distinguir de mejor forma las distintas clases entre sí [4].

El primer objetivo es construir una base  $w^k$ , tal que las proyecciones en los vectores maximicen un cierto indicador  $\gamma(w^k)$ . Para definir  $\gamma(u)$  como un vector unitario, primero debemos estimar información básica con respecto a las imágenes de entrenamiento y las clases  $C_i$  a las cuales pertenecen. Sean entonces  $m$  y  $m_i$  el valor medio de todas las imágenes de caras y los valores medios de cada clase  $C_i$  (con  $NC$  número de clases) estimados como el promedio de las imágenes de entrenamiento correspondientes. Sea  $P(C_i)$  la probabilidad asociada a la clase  $C_i$  dentro del espacio de los rostros. Entonces se define la cantidad  $\gamma(u)$  como:

$$\gamma(u) = \frac{S_b(u)}{S_w(u)} \quad (8)$$

Donde:

$$S_b(u) = \sum_{i=1}^{NC} P(C_i) (m^i - m) \mu^2 \quad (9)$$

Donde esto mide la separación entre los rostros promedio de cada clase con respecto al promedio global, en la proyección sobre el vector unitario  $\mu$ .

$$S_W(\mu) = \sum_{i=1}^{NC} P(C_i)(x^i - m^i)\mu^2 \quad (10)$$

mide la separación entre imágenes de una misma clase  $C_i$  en torno a su valor medio, en la proyección sobre  $\mu$ . De esta forma al maximizar  $\gamma(\mu)$  ecuación (8) se trata de que las proyecciones sobre  $\mu$  se logren separar las clases entre si y aglomerar las imágenes de una misma clase.

En forma análoga se realiza el procedimiento seguido en PCA. Para encontrar una base  $w^k$  que cumple esta propiedad entonces expresamos a:

$$S_b(\mu) = \mu^T S_b \mu \quad (11)$$

$$S_W(\mu) = \mu^T S_W \mu \quad (12)$$

Donde:

$$S_b(\mu) = \sum_{i=1}^{NC} P(C_i)(m^i - m)(m^i - m)^T \quad (13)$$

$$S_W(\mu) = \sum_{i=1}^{NC} P(C_i)(x^i - m)(x^i - m)^T \quad (14)$$

Ambas matrices pertenecen a  $R^{N \times N}$ , son simétricas y definidas semi-positivas. A  $S_b$  se le denomina "matriz de dispersión de rostros entre clases" y a  $S_W$  "matriz de dispersión de rostros dentro de las clases". Ahora podemos reescribir la ecuación (8) como:

$$\gamma(\mu) = \frac{\mu^T S_b \mu}{\mu^T S_W \mu} \quad (15)$$

Ahora, siguiendo los mismos pasos de PCA, escribimos la ecuación anterior como:

$$\gamma(\mu) = \frac{\mu^T S_W^{-1} S_b \mu}{\mu^T S_W \mu} \quad (16)$$

es claro que el sistema que estamos buscando está dado por:

$$S_W^{-1} S_b w^k = \lambda_k w^k \quad (17)$$

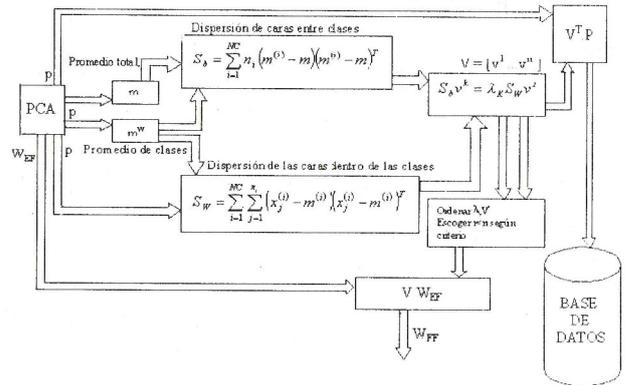


Fig. 2 Etapas necesarias para obtener las fisherfaces.

Reemplazando esta ecuación en la anterior vemos que  $\gamma(w^k) = \lambda_k$ . Análogamente a PCA, y actuando por contradicción, es fácil demostrar que no existe un vector distinto de  $w_1$  con un coeficiente mayor. Luego, los vectores  $w_k$  corresponden a la base óptima, es decir, maximizan en forma sucesiva (con los  $K$  ordenados de mayor a menor).

En la Fig. 2 se muestra un diagrama de bloques de todas las etapas necesarias para obtener las Fisherfaces. Si observamos la Fig. 2, la matriz de proyección PCA es  $W_{EF} \in R^{N \times m}$  y la matriz de proyección de Fisher en el espacio reducido es  $V \in R^{m \times r}$  (donde  $r \leq m$  es el número de componentes escogidas para una buena discriminación de clases), entonces la matriz de proyección definitiva del método de Fisher (FF)  $W_{FF} \in R^{N \times r}$  será:

$$W_{FF} = W_{EF} V \quad (18)$$

### METODOS EMPLEADOS CON HMM

#### HMM unidimensional

Un HMM uni-dimensional es una cadena de Markov con un número finito de estados no observados. Aunque los estados de Markov no son directamente observables, cada estado tiene una probabilidad de distribución asociada con un conjunto de posibles observaciones.

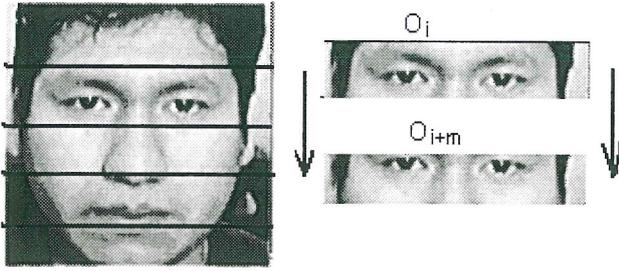


Fig. 3 Formación de los vectores de observación.

De esta manera, cuando el HMM está en el estado  $i$ , la salida (observación) está determinada de acuerdo a una función de densidad de probabilidad, a menudo gaussiana o una mezcla gaussiana  $P$ .

Para determinar estadísticamente un HMM, es necesario una matriz de transición, un estado inicial de la distribución de probabilidad y un conjunto de función de densidad de probabilidad asociados con la observación de cada estado Fig. 4.

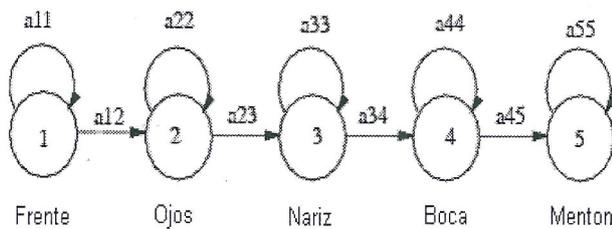


Fig. 4 Estados para un HMM unidimensional.

### HMM embebido (HMME)

Un modelo HMM uni-dimensional puede ser generalizado, para dar una apariencia a una estructura de dos dimensiones, que permite a cada estado HMM uni-dimensional a ser un HMM. De esta manera, el HMM consiste en un conjunto de súper estados con un conjunto de estados embebidos. Los súper estados pueden entonces ser usados para un modelo de datos de dos dimensiones a lo largo de una dirección y con un modelo HMM embebido de los datos a lo largo de otra dirección. Este modelo difiere del verdadero HMM de dos dimensiones por las transiciones entre los estados en diferentes súper estados que no son permitidos. Por consiguiente, este es descrito como un HMM embebido. Los elementos de un HMME son:

1. El número de súper estados,  $N_0$ , y el conjunto de súper estados,  $S_0 = \{S_{0,i} \mid 1 \leq i \leq N_0\}$ .
2. La distribución inicial del súper estado.  $\Pi_0 = \{\pi_{0,i}\}$ , donde  $\pi_{0,i}$  son las probabilidades iniciales del súper estado  $i$  en el tiempo cero.
3. El súper estado de la matriz de probabilidad de transición  $A_0 = \{a_{0,ij}\}$ , donde  $a_{0,ij}$  es la probabilidad de transición desde el súper estado  $i$  para el súper estado  $j$ .

Los parámetros de los HMM embebidos incluyen:

- El número de estados embebidos en el  $K^{\text{th}}$  súper estado,  $N_1^{(k)}$ , y un conjunto de estados embebidos,  $S_1^{(k)} = \{S_{1,i}^{(k)}\}$ .
- El estado inicial de distribución,  $\Pi_1^{(k)} = \{\pi_{1,i}^{(k)}\}$ , donde  $\pi_{1,i}^{(k)}$  son las probabilidades que existen en el estado  $i$  de súper estado  $k$  en tiempo cero.
- El estado de la matriz de probabilidad de transición,  $A_1^{(k)} = \{a_{1,ik}^{(k)}\}$ ; que especifican las probabilidades de transición desde el estado  $k$  al estado  $j$ .
- Finalmente, la matriz de probabilidad de estado de observación  $B^{(k)} = \{b_i^{(k)}(O_{t_0,t_1})\}$ ; donde  $O_{t_0,t_1}$  representa el vector de observación como la fila  $t_0$  y columna  $t_1$ .

Los estados son caracterizados por funciones de densidad de observaciones continuas. La función de densidad de probabilidad es típicamente usada como una mezcla finita de la forma:

$$B_i^{(k)}(O_{t_0,t_1}) = \sum_{m=1}^M c_{im}^{(k)} N(O_{t_0,t_1}, \mu_{im}^{(k)}), \quad (19)$$

Donde  $1 \leq i \leq N_1^{(k)}$ ,  $c_{im}^{(k)}$  es el coeficiente mixto, dado que el  $m^{\text{th}}$  es el estado  $i$  del súper estado  $k$ .

$N(O_{t_0,t_1}, \mu_{im}^{(k)}, U_{im}^{(k)})$  es una pdf gaussiana con un vector  $\mu_{im}^{(k)}$  y matriz Covarianza  $U_{im}^{(k)}$ .

Sea  $\Lambda^{(k)} = \{\Pi_1^{(k)}, A_1^{(k)}, B^{(k)}\}$  el conjunto de parámetros que define el  $k^{\text{th}}$  súper estado. Usando una notación taquigráfica, un HMM embebido es definido de la siguiente manera:

$$\lambda = (\Pi_0, A_0, \Lambda) \quad (20)$$

donde:

$$\Lambda = \{\Lambda^{(1)}, \Lambda^{(2)}, \dots, \Lambda^{(N_0)}\}$$

A pesar de la complejidad respecto a un HMM unidimensional, un HMM embebido es más apropiado para datos de dos dimensiones y tiene una complejidad proporcional a la suma de los cuadrados del número

de estados,  $\sum_{k=1}^{N_0} (N_1^{(k)})^2$ . La estructura de estados de los modelos de la cara y la probabilidad de transición de «no cero» del HMM embebido son mostradas en la Fig. 5.

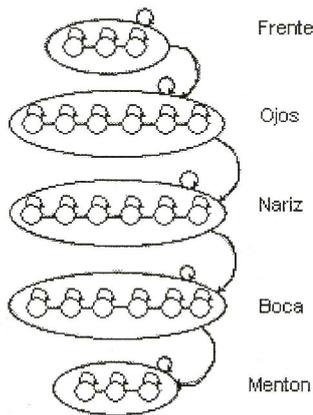


Fig. 5 Estados en un HMM embebido.

HMM va de izquierda a derecha, este modelo es apropiado para imágenes de rostros siempre y cuando los rostros estén de frente ya que estos preservan una estructura de súper estados desde arriba hacia abajo y también al mismo de izquierda a derecha de estados dentro de cada esos súper estados.

Comparando con las otras estructuras, un HMM embebido tiene las siguientes ventajas:

1. La complejidad es reducida en términos de entrenamiento y reconocimiento.
2. La mejor estimación inicial del modelo de parámetro que puede ser obtenido.
3. La estructura en dos dimensiones de los datos es desde luego preservado sin que se use un estado extra de fin de línea que incremente la complejidad de este modelo.
4. El estado de HMM embebido son descritos por una simple densidad gaussiana pdf.

$$b_i^{(k)}(O_{t_0:t_1}) = N(O_{t_0:t_1}, \mu_i^{(k)}, U_i^{(k)}) \quad (21)$$

5. Y la matriz de covarianza es asumida diagonal.

### Los vectores de observación

La secuencia de observación es generada usando la técnica mostrada en la Fig. 6, donde una ventana de  $P \times L$ , escanea la imagen de izquierda a derecha, y de arriba hacia abajo.

Al sobreponer entre la ventana adyacente: es  $M$  líneas en la dirección vertical y  $Q$  columnas en la dirección horizontal.

La observación de vectores consiste de todos los valores de píxeles desde cada uno de los bloques, y por consiguiente la dimensión del vector de observación es  $L \times P$ .

Pero esto tiene dos importantes desventajas: primero, el valor del píxel no representa una medida robusta, ya que ellos tienden a ser sensibles a los ruidos, rotación e iluminación de las imágenes, segundo, la dimensión del vector de observación es larga y el costo computacional es complejo y esto no sería eficiente en aplicaciones en tiempo real.

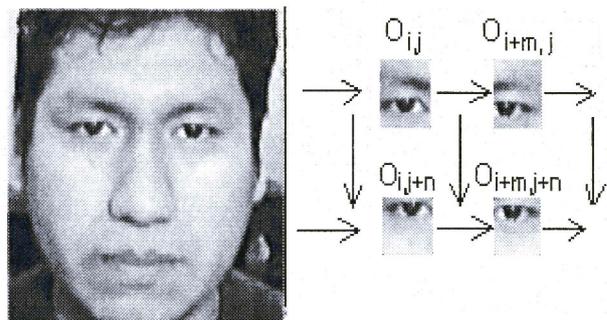


Fig. 6 Generación de vectores de observación para HMME.

Para ello en nuestro modelo HMM embebido, el vector de observación se forma desde los coeficientes 2D-DCT de cada bloque de la imagen.

La propiedad de comprensión y decorrelación es de 2D-DCT por imágenes iniciales, los cuales, son adecuados para el uso como vectores de observación.

Específicamente, los coeficientes dentro de una ventana rectangular por encima de bajas frecuencias en el dominio de 2D-DCT, donde más se encuentre la energía de la imagen, son usados como vectores de observación.

Usando los coeficientes de 2D-DCT mas bien los valores de pixeles, se reducen dramáticamente el tamaño del vector de observación y, por consiguiente, decremanta la complejidad del sistema reconocedor.

Para nuestro experimento  $L=8$ ,  $P=10$  se usan 6 coeficientes 2D-DCT desde cada bloque que son usados como vectores de observación. Por consiguiente, el tamaño del vector de observación es reducido en 13 tiempos comparado con el método que usa la intensidad de del píxel como el vector de observación.

**Modelo de entrenamiento de rostros**

Cada persona en la base de datos es representado por un HMM embebido. Un conjunto de imágenes representa diferentes instantes del mismo rostro a ser usado en el entrenamiento.

Los vectores de observación son extraídos desde cada bloque para ser usados para entrenar los modelos así como lo muestra la Fig. 7.

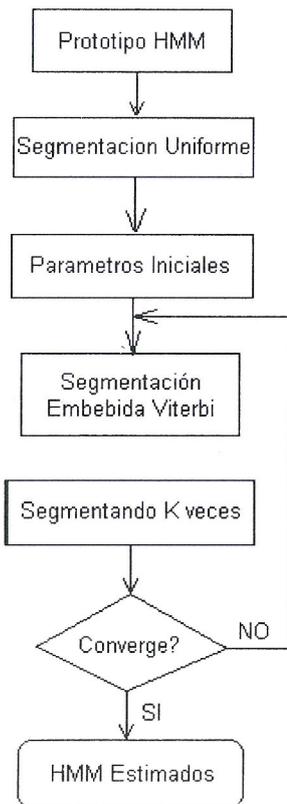


Fig. 7 Esquema de entrenamiento del HMME para rostros.

Según el número de súper estados, estados en cada súper estado, y la estructura del HMM embebido, los datos se segmentan uniformemente para obtener una estimación inicial de los parámetros del modelo. Primero, las observaciones de arriba hacia abajo HMM son segmentadas en no súper estados verticales, entonces, los datos que corresponden para cada uno de estos súper estados es segmentado uniformemente desde izquierda a derecha empezando en  $N_1^{(k)}$  estados. Para la siguiente iteración, la segmentación uniforme es reemplazada por una doble segmentación embebida con el algoritmo de Viterbi Fig. 7. La doble segmentación embebida del algoritmo de Viterbi se ilustra en la Fig. 8.

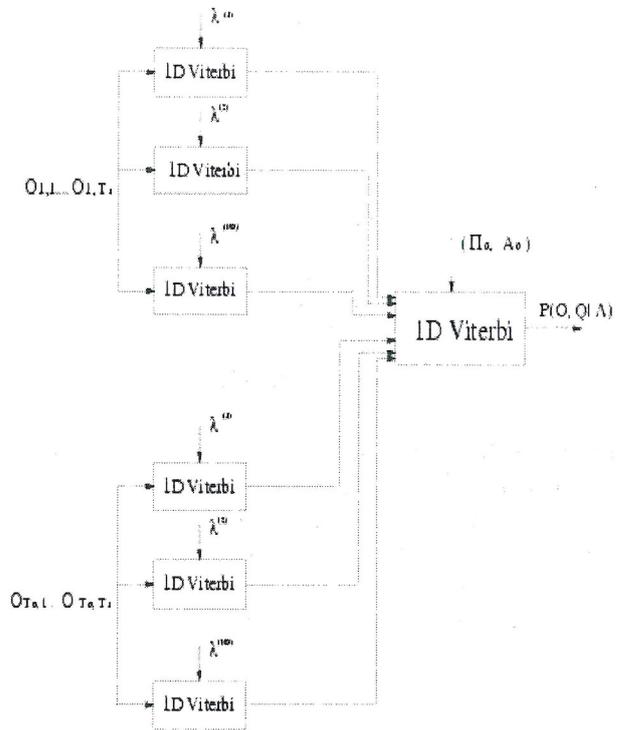


Fig. 8 Empleo del algoritmo de Viterbi.

El doble algoritmo de segmentación de Viterbi, ilustrada en la figura anterior, consiste en los siguientes pasos:

- Primero, la segmentación de Viterbi es aplicada a cada fila de la imagen, y la probabilidad:

$$P(O_{t_0,1} \dots O_{t_0,T_1}, q_{1,1}^{(t_0)} \dots q_{1,T_1}^{(t_0)} | \lambda^{(k)}) \quad (22)$$

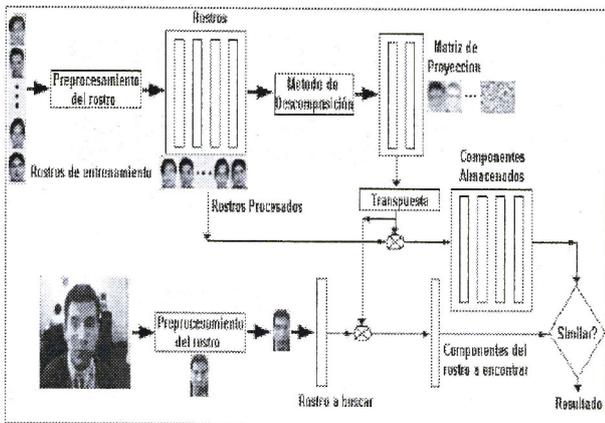
$$1 \leq k \leq N_0$$

son calculados, donde  $q_{1,t_1}^{(i_0)}$ ,  $1 \leq t_1 \leq T_1$  representa el estado de un súper estado asignado para la observación  $O_{i_0,t_1}$ . La probabilidad de los estados y observaciones en una fila, da el modelo de súper estado, obtenido desde la segmentación de Viterbi, representa la probabilidad de súper estados.

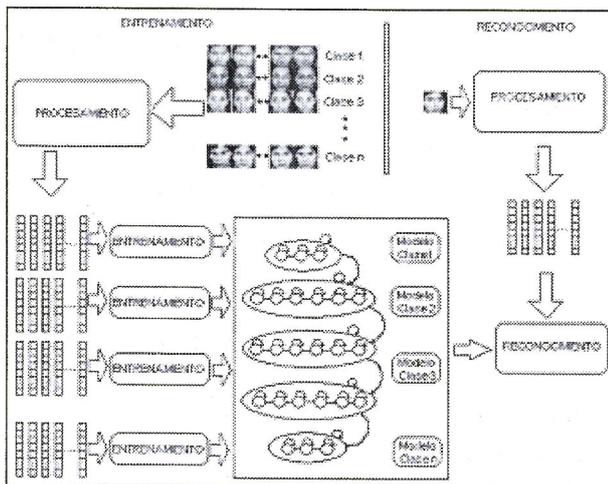
- La probabilidad del súper estado conjuntamente con la probabilidad de transición de súper estados A o la probabilidad inicial de súper estados  $\Pi_0$ .

**PROCESO DE IDENTIFICACIÓN**

Las figuras siguientes mostrarán los procesos por los dos métodos respectivos. En la Fig. 9, se muestra el proceso de reconocimiento usando el método de descomposición y en la Fig. 10 el proceso de reconocimiento usando HMM.



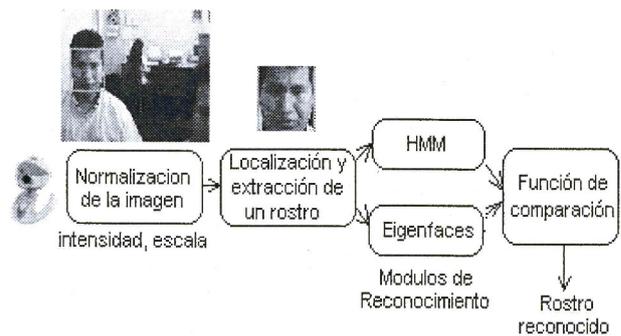
**Fig. 9** Proceso de reconocimiento usando el método de descomposición.



**Fig. 10** Proceso de reconocimiento usando HMM.

**PRUEBAS DEL SISTEMA**

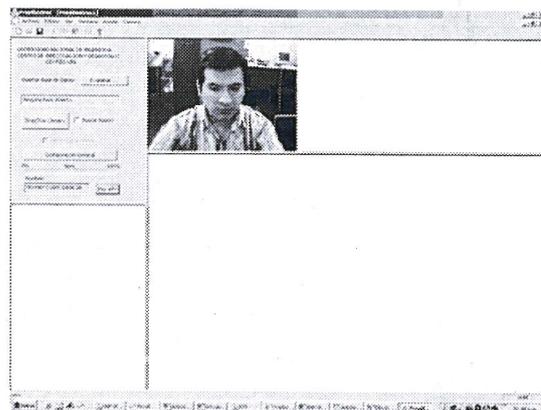
Se realizaron pruebas al sistema utilizando un potente software matemático: Matlab [5], que posee rutinas matemáticas, como es el manejo de matrices de gran dimensión, pero de capacidad limitada. Por lo que para su implementación se recurrió a utilizar el lenguaje C, y diseñar el software por Visual C++. Entonces se buscaron las librerías adecuadas para utilizar HMM, siendo elegida la librería OpenCV ya que posee una gran cantidad de funciones de procesamiento de imágenes, además de que estas librerías de Intel son optimizadas para PC Pentium®.



**Fig. 11** Proceso de reconocimiento general del sistema.

**Adquisición de la imagen**

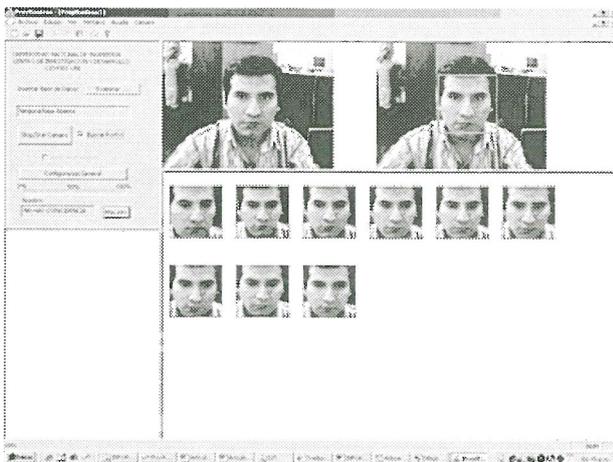
Para la adquisición de imágenes se procedió a diseñar el software de tal manera que utilice toda la capacidad del hardware, es por ello que se recurrió a utilizar DirectX. Específicamente DirectShow, con esto la toma de imágenes en tiempo real es óptima. La iluminación debe ser constante en un determinado ambiente, ya que esto afecta la eficiencia del reconocimiento.



**Fig. 12** Adquisición de imagen de video.

### Localización y extracción de un rostro

Para la localización del rostro se procedió a utilizar un clasificador (Particularmente llamado *cascade of boosted classifiers* [6]) que consiste en tener plantillas de rostros de diferentes tamaños (en cascada), y aplicar algún método (boosted), que clasifica cada etapa de la cascada. La función de Visual C++ llamada *cvHaarDetectObjects* que viene en la librería de OpenCV busca la región rectangular en una imagen dada, y retorna los rostros como una secuencia de rectángulos. Esta función escanea la imagen en varios tiempos y con diferentes escalas Fig. 13.



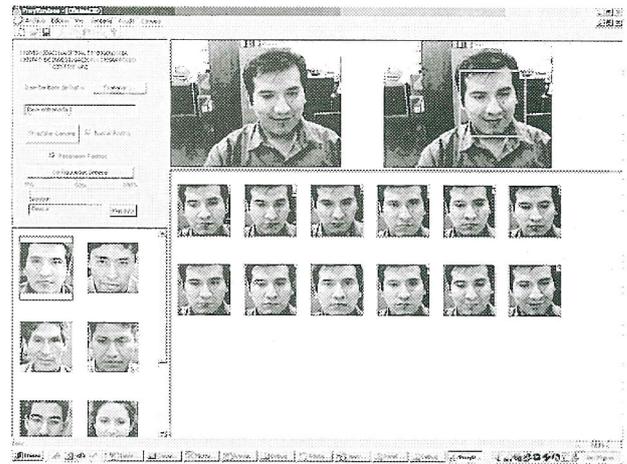
**Fig. 13** Adquisición de rostros y captura automática de los rostros en una imagen de video en tiempo real.

### Entrenamiento del sistema

Para el entrenamiento del sistema se utilizó una base de datos de rostros de los participantes de los proyectos en el Centro de Investigación y Desarrollo CID-FIEE. Estas imágenes de rostros se guardaron en formato comprimido jpg y con una resolución de 140x140 píxeles. Para utilizar el método de descomposición se utilizó el software en Matlab, y se entrenó con la base de datos de imágenes. Para utilizar el método de HMME, se realizó el software en Visual C++, pues, se requiere que el procesamiento se realice casi en tiempo real y a mayor velocidad. El entrenamiento por cada método se realizó de acuerdo a la Fig. 8 y 11.

### Reconocimiento de rostros

El Reconocimiento de rostros se realiza una vez que hayamos insertado una base de datos de rostros ya entrenados. Para esto el software desarrollado busca los rostros en una imagen de video, los encuentra y tiene la opción de que estas fotos se almacenen en una carpeta, a la vez se irá comparando con la base de datos de rostros y los estará mostrando en pantalla cada vez que reconozca un rostro, este proceso se realiza casi en tiempo real, de acuerdo a la capacidad y procesamiento de la PC.



**Fig. 14** Reconocimiento de rostros de acuerdo a una base de datos entrenada por HMME.

## RÉSULTADOS

Para los resultados se considero en general que el número de clases representa a la cantidad de personas entrenadas y el número de patrones por clase se refiere a la cantidad de rostros por persona entrenada. Se realizó dos tablas con los métodos utilizados, obteniéndose los resultados que a continuación se muestran.

### Resultados con los métodos de descomposición

La tabla 1, muestra los resultados obtenidos con el método de descomposición, el reconocimiento se realiza para un conjunto de 10 rostros por persona incluidos los rostros entrenados.

**Tabla 1. Resultados con el método de descomposición.**

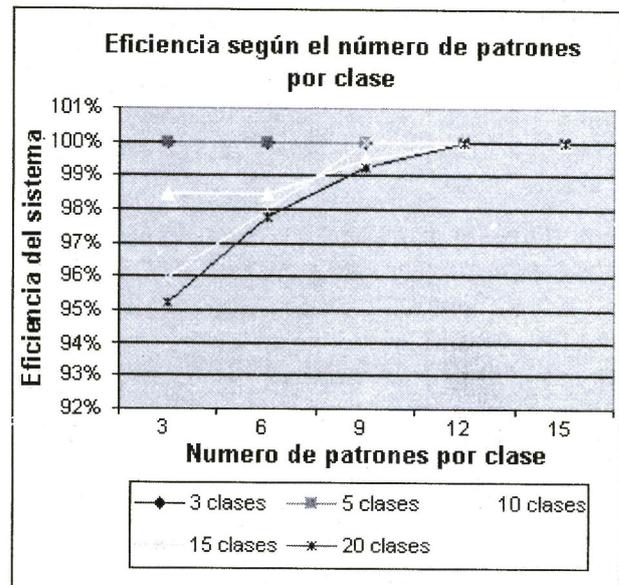
Número de Clases	Patrones por clase	Reconocimiento (10 por cada Clase)		Eficiencia
		Reconoce	No reconoce	
3		30	0	100%
5	3	50	0	100%
10	3	90	10	90%
15	3	126	24	84%
20	3	148	52	74%
	3			
Número de Clases	Patrones por clase	Reconocimiento (10 por cada Clase)		Eficiencia
		Reconoce	No reconoce	
10		90	10	90%
10	3	92	8	92%
10	4	98	2	98%
10	5	100	0	100%
10	6	100	0	100%
	7			

### Resultados obtenidos con HMME

La tabla 2, muestra los resultados del método de HMME y el modelo HMM embebido; el reconocimiento se realiza para un conjunto aproximado de 20 rostros por persona incluyendo las entrenadas.

**Tabla 2. Resultado con el método HMME.**

Número de Clases	Patrones por Clase	Reconocimiento (~20 por cada clase)		Eficiencia
		Reconoce	No Reconoce	
3	3	54	0	100%
3	6	54	0	100%
3	9	54	0	100%
3	12	54	0	100%
3	15	54	0	100%
5	3	94	0	100%
5	6	94	0	100%
5	9	94	0	100%
5	12	94	0	100%
5	15	94	0	100%
10	3	192	3	98.46 %
10	6	192	3	98.46 %
10	9	194	1	99.48%
10	12	195	0	100%
10	15	195	0	100%
15	3	287	12	95.98%
15	6	293	6	97.99%
15	9	298	1	99.96%
15	12	299	0	100%
15	15	299	0	100%
20	3	381	19	95.25%
20	6	391	9	97.75%
20	9	397	3	99.25%
20	12	400	0	100%
20	15	400	0	100%



**Fig. 14** Eficiencia según el algoritmo de HMME.

### Comparación entre los métodos utilizados

Según la tabla 1 y 2, observamos una diferencia entre los métodos utilizados. Con el método de descomposición se observa que a medida que aumentamos la cantidad de personas en nuestra base, el sistema baja su efectividad a 74%, en cambio usando el método de HMME y con las mismas bases de rostros el sistema llega a 95.25% y esto considerando solo tres rostros por persona, siendo esto el peor de los casos. Pero si se tiene una mayor cantidad de rostros por persona, el método de HMME converge con mayor rapidez con una buena eficiencia que utilizando el método de descomposición.

### CONCLUSIONES

Se concluye que el mejor entrenamiento se obtuvo con HMME, éste mostró una eficiencia mínima de 95.98% y ésta se debió a los rostros que no representaba bien a la persona y se consideró solo 3 rostros por persona haciéndola en el peor de los casos.

El método de descomposición solo puede ser utilizado para pequeñas bases de datos, y además se deben tener más fotos por persona para que sea afectivo el reconocimiento

El método de HMME puede ser utilizado para una mayor cantidad de personas, además es robusto para imágenes con menor escala y no es necesario que

tengan una resolución constante, como el caso del método de descomposición.

Para el mejor entrenamiento de rostros, este se debe hacer en el lugar donde se realizará el reconocimiento con ellos se consigue una mayor efectividad.

### TRABAJOS FUTUROS

Según lo visto en esta investigación, la utilización del modelo oculto de Markov (HMM) para imágenes de rostros resulta efectiva ya que se basa en características propias de un rostro. Esto hace pensar que también se pueden hacer reconocimiento de patrones con otras partes visibles del cuerpo, como por ejemplo, las manos que es utilizada para lectura de sordomudos. La utilización de modelos de Markov con Audio y video (Articulación de la boca al hablar) también proporciona más efectividad en reconocimiento de palabras, ya que frente a un ruido externo las imágenes determinarían la palabra correcta.

### AGRADECIMIENTOS

Al Instituto de Investigación de la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Ingeniería, por haber financiado este proyecto.

### REFERENCIAS

1. **Huang, J., Gutta, S., Wechsler, H.**, "Detection of Human Faces Using Decision Trees", 2nd International Conference on Automated Face and Gesture Recognition.
2. **Lawrence, G., Tsoi, Back.,** citeseer.nj.nec.com/lawrence96what.html "What Size Neural Network Gives Optimal Generalization?" Convergence Properties of Backpropagation (UMIACS-TR-96-22) 1996.
3. **Pentland, A. P., Moghaddam, B., Starner, T.**, "View-Based and Modular Eigenspaces for Face Recognition", IEEE Conference on Computer Vision & Pattern Recognition, 1994.
4. **Romdhani S.**, "Face Recognition Using Principal Components Analysis", Technical Report, University of Glasgow.
5. <http://www.mathworks.com/> The MathWorks Matlab and Simulink for Technical Computing.
6. **Viola, P.**, "Robust Real-time Object Detection" <http://research.microsoft.com/~viola/>
7. **Del Carpio Salinas, J. A., Huamán Layme, J., Marcelo Fernández, L.**, "Reconocimiento de Rostros". Facultad de Electricidad y Electrónica de la Universidad Nacional de Ingeniería, 2004, Lima, Perú.
8. **González, R. C., Woods, R. E.**, "Tratamiento Digital de Imágenes". Addison-Wesley Iberoamericana, S.A., 1996.

Correspondencia: huamanjose@speedy.com.pe

Recepción de originales: junio 2006

Aceptación de originales: setiembre 2006

