

El Método IRDB: Aprendizaje Incremental para el Reconocimiento de Caras

O. Déniz, M. Castrillón, J. Lorenzo, M. Hernández
Universidad de Las Palmas de Gran Canaria
Departamento de Informática y Sistemas
Edificio de Informática y Matemáticas
Campus Universitario de Tafira
35017 Las Palmas - España
{odeniz,mcastrillon,jlorenzo,mhernandez}@dis.ulpgc.es

Resumen: *En un problema complejo como es el reconocimiento de caras, donde se utilizan espacios de representación de un número alto de dimensiones, es de gran importancia emplear toda la información disponible. En particular, habrá casos en los que, estando el sistema de reconocimiento en funcionamiento, se acumulará una gran cantidad de información en forma de imágenes faciales con su etiqueta asociada. En este trabajo se propone un algoritmo para utilizar esta información, caso de estar disponible. El algoritmo presenta la principal característica de ser incremental, con lo que la eficiencia del mismo no degenera con el número de imágenes acumuladas. Por otra parte, el algoritmo hace uso de la combinación temporal de los resultados de clasificación obtenidos para las distintas imágenes de la secuencia de entrada. Mediante experimentos realizados con secuencias obtenidas con un sistema de reconocimiento hardware-software, se analiza el rendimiento de dicho método, así como sus limitaciones y posibles ampliaciones.*

Palabras clave: visión por computador, reconocimiento de caras, fusión temporal, aprendizaje incremental

1 Introducción

El problema del reconocimiento de caras ha generado una ingente cantidad de trabajos de investigación en los últimos años. Aunque es una tarea de enorme complejidad, algunos sistemas han demostrado ser capaces de lograr un rendimiento aceptable en condiciones restringidas. Sin embargo, en la mayoría de los trabajos aparecidos se realizan experimentos en condiciones no reales de funcionamiento, como por ejemplo usar solo una imagen

para el reconocimiento. Varios autores han demostrado que el uso de la información que proporciona una secuencia de imágenes mejora notablemente la respuesta del sistema, en comparación con el uso de una sola imagen. Algunos de los sistemas descritos modifican la representación o los clasificadores utilizados para tener en cuenta la información de las secuencias. Otros simplemente realizan una fusión de los resultados de clasificación. Un ejemplo del primer caso aparece en [1], donde se caracterizan las caras mediante trayectorias en un espacio de representación. El reconocimiento se hace comparando la trayectoria correspondiente a la secuencia a reconocer con las de secuencias de entrenamiento. En problemas de verificación, en [3] se muestra como el uso de múltiples imágenes permite reducir el error hasta en un 40%. En el mismo trabajo se observa que la reducción en el error es grande al principio pero tiende a disminuir conforme se acumulan más imágenes. En [2] se genera un subespacio de representación con la secuencia de imágenes de entrada, que se compara con subespacios obtenidos en la fase de entrenamiento, apreciándose un aumento de la robustez del sistema ante cambios de expresión y de pose. Por otro lado, con referencia a los trabajos en los que se realiza una fusión de los resultados, no existe un método de fusión/combinación predominante; los más utilizados son la regla de la media [7], la suma [8] o el máximo [4, 5, 6]. Desde un punto de vista práctico, se emplea generalmente una fusión de los resultados de clasificación, pues es tanto conceptual como computacionalmente la opción menos compleja.

El problema del reconocimiento de imágenes faciales presenta una dificultad tal, que toda la información que se pueda aprovechar es poca. El ciclo de vida de un sistema práctico se dividiría en dos etapas: inducción del clasificador (a partir de un conjunto de imágenes de entrenamiento) y funcionamiento propiamente dicho. Desde un punto de vista computacional no resulta práctico generar un nuevo clasificador cada vez que en la etapa de funcionamiento se obtenga nueva información supervisada, pues el coste depende del número de muestras utilizadas. En [9] se describe una técnica de aprendizaje automático aplicada al reconocimiento de caras, si bien no se emplea información supervisada, sino que la propia salida del sistema se utiliza para actualizar la representación interna. Si el sistema funciona con bajo error, dicho método funcionará correctamente. Sin embargo, cuando el sistema se equivoque con frecuencia aparecerá una tendencia a empeorar. Por otra parte, según la aplicación concreta pueden aparecer situaciones en las que se disponga de información supervisada. Por ejemplo, el propio individuo puede identificarse voluntariamente, o bien el propio sistema puede descubrir la identidad del individuo por otros medios. Una solución para aplicaciones de este tipo se describe en [10], donde se emplea un árbol de decisión para espacios de gran dimensión. Cada nodo del árbol representa un espacio obtenido mediante PCA. El árbol se actualiza dinámicamente mediante mecanismos de olvido y crecimiento controlado. En [11] se describe el sistema *Argus* para reconocimiento de visitantes. El cometido de *Argus* es detectar y reconocer las personas que llegan a la puerta de un edificio, así como notificar la llegada a las personas residentes en el edificio relacionadas con el visitante. Cuando *Argus* se equivoca, el usuario residente en el edificio puede facilitar la identidad de la persona en la puerta (caso de conocerla y de querer hacerlo), o aceptar la notificación como correcta. *Argus* emplea un reconocimiento basado en imágenes almacenadas y el clasificador del

vecino más cercano, con lo que esta información del usuario residente puede aprovecharse fácilmente, almacenando las nuevas imágenes obtenidas.

En este trabajo se describe un método para aprovechar la información supervisada que pueda obtenerse en la etapa de funcionamiento, con el fin de mejorar la precisión del sistema. El método presenta como principales características el uso de la combinación de los resultados de clasificación de los frames de las secuencias y la incrementalidad. En la Sección 2 se describe el método propuesto. El algoritmo correspondiente se presenta en la Sección 3. Los experimentos realizados para comprobar su rendimiento se explican en la Sección 4. Por último, en la Sección 5 se detallan las conclusiones más importantes.

2 El método IRDB

El método propuesto, que denominaremos IRDB (*Incremental Refinement of Decision Boundaries*), se aplica sobre un esquema de decisión como el representado por la regla:

$$\text{Si } m = \arg \max_{i=1, \dots, z} d_i(\vec{x}) \Rightarrow \vec{x} \in C_m, \quad (1)$$

donde $d_i(\vec{x})$ es la estimación o medida de pertenencia a la clase i que proporciona el clasificador, y las clases se denotan por C_1, \dots, C_z . Estando el sistema en funcionamiento se dispone de un conjunto de n muestras etiquetadas $(\vec{x}^j; d_1^j, \dots, d_z^j)$, con $j = 1, \dots, n$. Con el fin de aprovechar esta información para mejorar la precisión del clasificador se utiliza como nueva regla de decisión:

$$\text{Si } m = \arg \max_{i=1, \dots, z} F_i(\vec{p}_i; d_i(\vec{x})) \Rightarrow \vec{x} \in C_m, \quad (2)$$

donde las funciones F_i actúan como modificadores de las salidas que proporciona el clasificador y están además parametrizadas por los vectores \vec{p}_i , de P elementos. Los vectores de parámetros deberán ser aquellos que mejor se ajusten a la relación $d_i^j = F_i(\vec{p}_i; d_i(\vec{x}^j))$, para $j = 1, \dots, n$. Estos vectores \vec{p}_i pueden asignarse uno a uno de forma subóptima mediante:

$$\vec{p}_i^* = \arg \min_{\vec{p}_i} \left[\hat{P}(\text{error} | \vec{x} \in C_i, \vec{p}_i) - \hat{P}(\text{acierto} | \vec{x} \notin C_i, \vec{p}_i) \right] ; i = 1, \dots, z, \quad (3)$$

donde las probabilidades \hat{P} han de estimarse a partir de las muestras etiquetadas disponibles. En la Figura 1 se muestra el esquema general de funcionamiento del método.

Si el número total de muestras etiquetadas utilizadas para obtener las estimaciones (Q) es pequeño, el resultado puede ser peor que el obtenido con la regla de decisión original. No obstante, puesto que, para $i = 1, \dots, z$:

$$\lim_{Q \rightarrow \text{inf}} \frac{\left[\hat{P}(\text{error} | \vec{x} \in C_i, \vec{p}_i) - \hat{P}(\text{acierto} | \vec{x} \notin C_i, \vec{p}_i) \right]}{\left[P(\text{error} | \vec{x} \in C_i, \vec{p}_i) - P(\text{acierto} | \vec{x} \notin C_i, \vec{p}_i) \right]} = \quad (4)$$

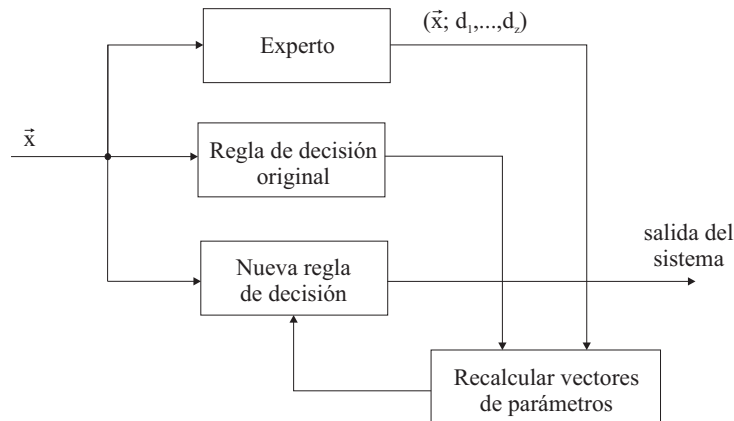


Figura 1. Esquema de funcionamiento del método IRDB. El cálculo de los vectores de parámetros solo se realiza cuando alguna entidad (experto) proporciona etiquetas para las muestras de entrada.

el método tenderá a dar un resultado mejor conforme se acumulen muestras.

El objetivo es pues obtener los vectores de parámetros \vec{p}_i que mejor se ajusten a las muestras etiquetadas, de forma incremental y en un tiempo no indeterminado, puesto que la operación habrá de hacerse en línea. El algoritmo que permite realizar esto se describe en la siguiente sección.

3 Algoritmo

La realización concreta del método IRDB que se estudia en este trabajo corresponde al uso de las funciones $F_i = d_i(\vec{x}) + p_i$; es decir, se utiliza un peso aditivo para cada salida de clase. El uso de pesos es equivalente al uso de un determinado umbral de decisión, por lo que para dos clases el método puede verse como una continua búsqueda de una posición más ventajosa en la curva ROC (*Receiver Operating Characteristic*) del sistema. Por otra parte, las entradas d_i que utiliza el método son tanto las salidas del clasificador como el resultado de la fusión o agregación temporal de las salidas del clasificador. Esta decisión responde a que uno de los objetivos es mejorar la tasa de aciertos de reconocimiento de secuencias. En concreto, se pretende mejorar la respuesta de reconocimiento en función del número de frames considerados. En funcionamiento real no está determinado el número de frames que se pasarán a la etapa de clasificación (aunque puede fijarse). Una forma simple de medir esta respuesta es el área bajo la curva que representa tasa de acierto frente a n^o de frames de la secuencia, calculada como la sumatoria S del porcentaje de subsecuencias de test acertadas para cada valor de número de frames considerados. Se asume que el número de frames que tendrá una secuencia de entrada es desconocido a priori y puede variar entre 1 y un valor máximo prefijado (que puede estar relacionado con el tiempo máximo de respuesta del sistema). Con el fin de facilitar la explicación del algoritmo, a partir de este punto supondremos que el número de clases z es igual a 2.

Tal como se explicó en la Sección 2, para asignar los pesos es necesario obtener las estimaciones frecuenciales $\hat{P}(\text{error}|\vec{x} \in C_i, \vec{p}_i)$ y $\hat{P}(\text{acierto}|\vec{x} \notin C_i, \vec{p}_i)$, o lo que es igual, $\hat{P}(\text{error}|\vec{x} \in C_i, F_i - d_i(\vec{x}))$ y $\hat{P}(\text{acierto}|\vec{x} \notin C_i, F_i - d_i(\vec{x}))$. Estas estimaciones se representarán por 2 histogramas, $\vec{H}_i = (H_{i,1}, \dots, H_{i,nb})$ y $\overline{HNO}_i = (HNO_{i,1}, \dots, HNO_{i,nb})$, caracterizados ambos por los rangos (r_0, \dots, r_{nb}) , siendo nb un parámetro fijado a priori. Los histogramas, 2 para cada clase i , se calculan como se explica a continuación. Dada una muestra \vec{x} etiquetada que llega, si pertenece a la clase i (para i variando de 1 a z) y la regla de decisión original se equivoca (es decir, si $\vec{x} \in C_i$ y $\arg \max_{j=1, \dots, z} d_j(\vec{x}) \neq i$), calculamos el valor $p = \max_{j=1, \dots, z} (d_j(\vec{x})) - d_i(\vec{x})$, que es la diferencia entre la salida obtenida por la clase ganadora y la obtenida por la clase i . Este valor p se añade entonces al histograma \vec{H}_i : si $r_k \leq p < r_{k+1} \Rightarrow H_{i,k+1} = H_{i,k+1} + 1$. Si por el contrario \vec{x} no pertenece a la clase i y la regla de decisión original acierta (es decir, si $\vec{x} \in C_l, l \neq i$ y $\arg \max_{j=1, \dots, z} d_j(\vec{x}) = l$), se calcula el valor $p = \max_{j=1, \dots, z} (d_j(\vec{x})) - d_i(\vec{x})$ y se añade al histograma \overline{HNO}_i . Una vez se ha añadido la nueva información a los histogramas se procede a la asignación de pesos. Para obtener el peso a aplicar a una clase i se emplea (3), relación que se traduce en calcular el máximo de la resta de los histogramas \vec{H}_i y \overline{HNO}_i . Esto es, se adjudica el peso que, con respecto a todas las muestras etiquetadas consideradas eliminaría una gran cantidad de errores y haría perder pocos aciertos. El peso a adjudicar es pues uno de los valores r_0 a r_{nb} . Una vez se asigna un peso p_i , y antes de proceder a adjudicar el siguiente, es necesario actualizar todos los histogramas para mantener la coherencia del proceso. Una forma de hacer esto sería recalcular los histogramas desde el principio, volviendo a considerar cada muestra etiquetada (y empleando el peso p_i recién asignado). Sin embargo, esto convertiría al proceso en no incremental. Es posible hacer la actualización mediante modificaciones a los histogramas si se cumple la condición $r_i + r_j = r_k$, para $i, j, k = 0, \dots, nb$ y j e i tales que $r_i + r_j \leq r_{nb}$. De esta forma aseguramos que, al asignar un peso, la modificación que hay que hacer a los histogramas son simples desplazamientos. Por ejemplo, si tenemos $r = (0, 0.2, 0.4, 0.6, 0.8, 1)$ y los valores de p que se han obtenido para la clase 1 son 0.1, 0.3, 0.5, 0.1, 0.5, 0.1, el histograma \vec{H}_1 sería $\vec{H}_1 = (3, 1, 2, 0, 0)$. Al asignar un peso $p_1 = 0.2$, el nuevo \vec{H}_1 sería $\vec{H}_1 = (1, 2, 0, 0, 0)$. El valor 3 pasaría a \overline{HNO}_2 , puesto que corresponde a fallos de la clase 1 que tras la asignación se convertirán en aciertos de la clase 2. Utilizando la notación multiclase de la Sección 2, esta condición puede expresarse como $F_i(\vec{r}_j, \vec{r}_k) = \vec{r}_l$, para $j, k, l = 0, \dots, nb$ y j y k tales que $F_i(\vec{r}_j, \vec{r}_k) \leq \vec{r}_{nb}$. Para el caso particular de $F_i = d_i(\vec{x}) + p_i$, es sencillo ver que la condición se cumple si los histogramas son equiespaciados.

En cuanto al coste computacional del algoritmo IRDB, es dependiente del número de trozos a usar en los histogramas (nb), del número de clases z , y del proceso de generación de subsecuencias. Si se quiere aprovechar toda la información disponible, deberían generarse todas las combinaciones posibles de frames de la secuencia. Dado que la cantidad de combinaciones posibles es un número combinatorio es necesario escoger otra opción. Puede usarse, por ejemplo, un número determinado de las subsecuencias posibles. El efecto de esto sería únicamente el de retrasar el aprendizaje, puesto que se estaría utilizando menos información que la disponible en cada secuencia. Por otro lado, el usar la

información de una secuencia de entrada, que pertenece a una única clase, provocaría un desequilibrio en los histogramas que a su vez produciría un resultado incorrecto. Para evitar esto se replica la información obtenida en pasos anteriores para las otras clases. Alternativamente, puede optarse por actualizar los pesos solo cuando se tenga el mismo número de muestras para ambas clases. Por otra parte, el coste de almacenamiento del método es $O(z \cdot nb^{P(z-1)})$, siendo P el número de elementos de los vectores de parámetros, lo cual puede representar una restricción según el número de clases del problema.

4 Experimentos

Con el fin de analizar el rendimiento del método IRDB se realizaron experimentos con secuencias reales de imágenes de caras. Las secuencias utilizadas en los experimentos se obtuvieron con el sistema DESEO (Detección y Seguimiento de Objetos) [12]. DESEO es un sistema hardware-software capaz de realizar detección y seguimiento de personas en tiempo real, empleando información de movimiento y/o color de la piel. Las imágenes que proporciona DESEO son procesadas para confirmar que realmente estamos ante una cara relativamente frontal y, si es así, normalizarla para su posterior reconocimiento. El proceso completo se describe con más detalle en [13]. El resultado neto es un conjunto de imágenes de la cara, normalizadas y listas para su reconocimiento, ver Figura 2.



Figura 2. Dos secuencias de imágenes faciales normalizadas, obtenidas con el sistema descrito en la Sección 4, y utilizadas en los experimentos.

Los experimentos se realizaron con secuencias de imágenes obtenidas con el sistema descrito. Por un lado se empleó un problema de 2 clases: se utilizaron 10 secuencias, una por individuo, cada una de 167 imágenes, 1670 imágenes distintas en total. Las clases están constituidas por: personas vinculadas al laboratorio de trabajo de los autores y personas que, siendo habituales, no están directamente vinculadas al laboratorio. Por otro lado se realizaron experimentos utilizando 5 clases: una secuencia por individuo, siendo el objetivo en este caso reconocer la identidad del mismo. Todas las imágenes utilizadas en los experimentos tienen un tamaño de 39x43 pixels. De cada secuencia, se tomaron 3 imágenes de entrenamiento del clasificador, 50 como información supervisada para el algoritmo, y el resto para test. Sobre el conjunto de las imágenes de entrenamiento se aplicó la técnica PCA (*Principal Component Analysis*). Cada experimento se realizó diez

veces, cambiando cada vez el orden de las imágenes de cada secuencia, aleatoriamente. Los resultados finales son la media de los diez resultados parciales. Las imágenes de test de cada secuencia se tomaron de n en n , con solapamiento (ver Figura 3), donde n es el número de imágenes consideradas, que va de 1 a 10. La generación de subsecuencias para el algoritmo IRDB se hizo de la misma forma. Como clasificador se utilizó en un caso el vecino más cercano (utilizando la media como prototipo y distancia euclídea) y en otro un clasificador basado en SVM (*Support Vector Machines*), con kernel de función de base radial. El valor del parámetro nb , que indica el número de trozos a usar en los histogramas, se fijó para todos los experimentos en 20.

Los resultados obtenidos sin utilizar IRDB se muestran en la Tabla I. Todas las curvas acierto- n° de frames de la secuencia de los experimentos resultaron monótonas crecientes. Como reglas de agregación se utilizaron la media y la mayoría de votos. Asimismo, para convertir los valores de salida de los clasificadores al rango $[0,1]$, se utilizó la función de mapeo $y = 1/(1 - e^{-\frac{x-\mu}{\sigma}})$ donde μ y σ son respectivamente la media y la desviación típica de los valores obtenidos para el conjunto de entrenamiento.

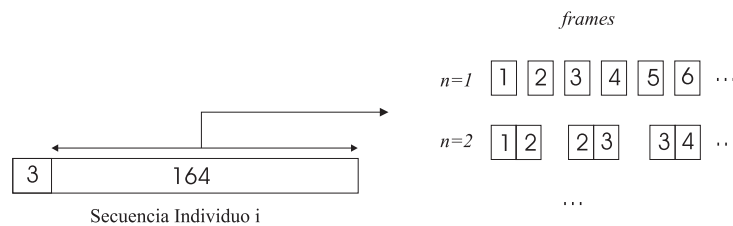


Figura 3. Formación de las secuencias utilizadas en los experimentos.

Tabla I. Resultados de sumatoria de porcentajes de aciertos (S) y máximo porcentaje alcanzado de secuencias de test acertadas (MAX), para los distintos clasificadores y reglas de agregación.

Clasificador utilizado	Nº Clases/Agregación	S	MAX
Vecino más cercano	2 / Media	732.40	74.88
	2 / Mayoría de votos	721.48	74.55
	5 / Media	745.65	77.23
Clasificador SVM	2 / Media	782.05	79.96
	2 / Mayoría de votos	770.85	79.26
	5 / Media	779.22	80.49

Los resultados obtenidos utilizando IRDB se muestran en las Tablas II y III, para 2 clases, y IV para 5 clases. En la primera columna (F) aparece el número de frames acumuladas, en la segunda columna (C) la clase de la que se acumulan los frames. Con respecto a la segunda línea de la tabla, n es el número máximo de frames de las subsecuencias que se generan. N es el valor máximo posible, que en este caso es $max(\text{frames de la secuencia de entrada}, 10)$. Puede observarse que el método IRDB consigue mejorar la precisión del reconocimiento, tanto más cuanto mayor el número de frames acumuladas. También se observa el efecto positivo de generar subsecuencias con $n=N$, con respecto a usar $n=1$ (en el caso de $n=1$ el método no usa agregación de resultados).

Tabla II. Resultados obtenidos con el algoritmo IRDB, usando la regla de agregación de la media.

F	C	Vecino más cercano				Clasificador SVM			
		n=N		n=l		n=N		n=l	
		S	MAX	S	MAX	S	MAX	S	MAX
4	1	711.83	72.67	709.05	72.391	760.35	77.78	770.45	78.47
	2	724.33	74.38	722.09	73.802	812.13	83.36	793.87	81.23
6	1	729.37	75.03	719.21	73.601	815.02	84.02	798.17	81.80
	2	734.28	75.71	724.26	74.262	815.19	83.89	812.54	83.33
8	1	734.28	75.71	722.46	74.011	814.46	83.77	815.02	83.71
	2	743.08	76.77	728.02	74.592	813.52	83.39	814.59	83.66
25	1	748.22	76.87	732.57	74.961	815.89	84.05	814.58	83.49
	2	751.87	77.54	728.96	74.562	820.87	84.81	805.46	82.33
50	1	747.59	76.94	732.78	74.891	820.98	84.46	810.38	83.06
	2	751.02	77.50	736.47	75.592	822.60	84.89	817.82	83.84

Tabla III. Resultados obtenidos con el algoritmo IRDB, usando la regla de agregación de la mayoría de votos.

F	C	Vecino más cercano				Clasificador SVM			
		n=N		n=l		n=N		n=l	
		S	MAX	S	MAX	S	MAX	S	MAX
4	1	721.48	74.55	723.29	74.771	770.85	79.26	770.85	79.26
	2	727.46	75.17	727.46	75.172	775.98	79.72	775.98	79.72
6	1	727.46	75.17	727.46	75.171	779.07	80.21	777.43	79.91
	2	727.46	75.17	726.80	75.092	777.62	80.03	778.78	80.19
8	1	727.46	75.17	727.46	75.171	779.07	80.21	779.07	80.21
	2	730.73	75.48	726.80	75.092	777.62	80.03	777.62	80.03
25	1	730.39	75.56	730.73	75.481	778.03	80.11	779.74	80.32
	2	731.75	75.70	730.73	75.482	777.62	80.03	776.82	79.99
50	1	731.39	75.57	730.73	75.481	777.62	80.03	778.52	80.16
	2	731.39	75.57	729.86	75.472	778.41	80.10	778.52	80.16

5 Conclusiones y Trabajo Futuro

En los últimos años se ha comprobado empíricamente que la combinación temporal de los resultados de clasificación para los frames de una secuencia produce un mejor rendimiento del sistema de reconocimiento. En sistemas prácticos se usan reglas de combinación simples, como la media, el máximo o la mayoría de votos. Por otro lado, existen situaciones concretas en el funcionamiento de un sistema de reconocimiento en las que se dispone de información supervisada que, dada la complejidad del problema, no se debería desaprovechar. Ambos aspectos, de interés eminentemente práctico, han sido tratados en el método de aprendizaje incremental propuesto. Desde un punto de vista computacional, el método no degenera con el número de frames acumuladas. Existen numerosas mejoras potencialmente aplicables al método IRDB, siendo la más importante la disminución del coste de almacenamiento para número de clases altos. Una forma de aliviar esta limitación podría ser el uso de memoria paginada en disco, así como el uso de matrices escasas. Por otra parte algunos clasificadores se extienden al problema multiclase mediante varias soluciones biclásicas (p.ej. los basados en *Support Vector Machines*), y consiguientemente no sufrirían este problema de almacenamiento. También se plantea estudiar la posibilidad de usar otras funciones F_i de modificación de las salidas, que puedan proporcionar una

Tabla IV. Resultados para el reconocimiento de 5 individuos, usando agregación con la media.

F	C	Vecino más cercano				Clasificador SVM			
		n=N		n=1		n=N		n=1	
		S	MAX	S	MAX	S	MAX	S	MAX
4	1	751.09	77.81	745.84	77.231	779.22	80.49	779.22	80.49
	2	753.27	78.02	745.84	77.232	698.76	75.42	779.22	80.49
	3	753.27	78.02	745.84	77.233	735.80	77.64	745.53	78.42
	4	756.15	78.61	756.15	78.614	770.94	80.39	785.92	81.66
	5	749.89	77.92	744.20	77.245	809.81	84.46	795.55	82.79
6	1	749.89	77.92	749.58	78.021	794.44	82.69	769.32	80.30
	2	749.89	77.92	749.58	78.022	810.48	84.33	796.22	82.67
	3	749.89	77.92	749.58	78.023	810.48	84.33	796.22	82.67
	4	760.41	79.19	744.20	77.244	815.72	84.82	800.58	83.06
	5	766.78	79.78	750.57	77.835	815.72	84.82	800.58	83.06
8	1	760.41	79.19	744.20	77.241	815.72	84.82	789.72	82.33
	2	760.41	79.19	744.20	77.242	815.72	84.82	796.03	83.18
	3	760.41	79.19	744.20	77.243	815.72	84.82	796.03	83.18
	4	760.41	79.19	744.20	77.244	812.58	84.74	801.86	83.65
	5	766.78	79.78	750.76	77.835	812.58	84.74	801.86	83.65
25	1	772.17	80.56	750.76	77.831	827.16	86.44	791.01	82.92
	2	772.17	80.56	750.76	77.832	827.16	86.44	791.01	82.92
	3	772.17	80.56	750.76	77.833	827.16	86.44	791.01	82.92
	4	766.47	79.89	766.47	79.894	827.16	86.44	776.52	81.29
	5	768.13	80.10	766.47	79.895	841.11	88.26	806.89	83.92
50	1	768.13	80.10	761.09	79.111	841.11	88.26	806.89	83.92
	2	768.13	80.10	761.09	79.112	842.51	88.60	806.89	83.92
	3	768.13	80.10	761.09	79.113	842.51	88.60	806.89	83.92
	4	762.43	79.42	761.09	79.114	842.51	88.60	809.66	84.21
	5	768.13	80.10	766.47	79.895	842.51	88.60	798.74	83.50

mayor capacidad de aprendizaje.

Agradecimientos

El trabajo de O. Déniz está financiado por la beca *D260/54066308-R* de la Universidad de Las Palmas de Gran Canaria. Este trabajo se financió en parte con cargo a los proyectos de investigación DGUI-Gobierno de Canarias *PI2000/042* y UE/DGES *1FD1997-1580-C02-02*.

Referencias

- [1] Yongmin Li, Shaogang Gong, and Heather Liddell. Exploiting the dynamics of faces in spatio-temporal context. In *Procs. The Sixth International Conference on Control, Automation, Robotics and Vision (ICARCV2000)*, Singapore, December 2000.
- [2] Osamu Yamaguchi, Kazuhiro Fukui, and Ken ichi Maeda. Face recognition using temporal image sequence. In *IEEE Int'l Conference on Automatic Face and Gesture Recognition*, pages 318–323, Nara, Japan, 1998.
- [3] J. Kittler, J. Matas, K. Jonsson, and M.U. Ramos Sánchez. Combining evidence in personal identity verification systems. *Pattern Recognition Letters*, 18(9):845–852, 1997.

- [4] A. Jonathan Howell and H. Buxton. Towards unconstrained face recognition from image sequences. In *Procs. of the Second Int. Conference on Automatic Face and Gesture Recognition*, Killington, Vermont, October 1996.
- [5] H. Wechsler, V. Kakkad, J. Huang, S. Gutta, and V. Chen. Automatic video-based person authentication using the RBF network. In *First Int'l Conference on Audio and Video-Based Biometric Person Authentication (AVBPA)*, Crans-Montana, Switzerland, 1997.
- [6] T. Choudbury, B. Clarkson, T. Jebara, and A. Pentland. Multimodal person recognition using unconstrained audio and video. Technical Report TR-472, MIT Media Lab, 1998.
- [7] A. Senior. Recognizing faces in broadcast video. In *Int. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, Corfu, Greece, Sept. 1999.
- [8] S. McKenna and S. Gong. Recognising moving faces. In *Procs. of the NATO ASI on Face Recognition: From Theory to Applications*, Stirling, UK, 1997.
- [9] K. Okada and C. von der Malsburg. Automatic video indexing with incremental gallery creation: integration of recognition and knowledge acquisition. In *Procs. of the Third International Conference on Knowledge-Based Intelligent Information Engineering Systems*, pages 431–434, Adelaide, August 1999.
- [10] J. Weng, C.H. Evans, and W.S. Hwang. An incremental learning method for face recognition under continuous video stream. In *Procs. of the Fourth International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, March 2000.
- [11] R. Sukthankar and R. Stockton. Argus: The digital doorman. *IEEE Intelligent Systems and their applications*, 16(2):14–19, 2001.
- [12] F.M Hernández, J. Cabrera, M. Castrillón, and C. Guerra. DESEO: An active vision system for detection, tracking and recognition. In *Procs. of the Second International Conference on Automatic Face and Gesture Recognition*, Killington, Vermont, October 1996.
- [13] M. Castrillon, J. Lorenzo, M. Hernandez, and J. Cabrera. Before characterizing faces. In *IX Spanish Symposium on Pattern Recognition and Image Analysis*, Castellón, Spain, 2001.