

RAMA: Reconocedor Automático de Matrículas de Automóviles

Fernando Martín Rodríguez y Xulio Fernández Hermida

Departamento de Teoría de la Señal y Comunicaciones.
Universidad de Vigo.

E.T.S.I.T. Ciudad Universitaria S/N. 36200 Vigo (Pontevedra). España.

Tel: +34-986-812131, FAX: +34-986-812116

E-mail: fmartin@tsc.uvigo.es, xulio@tsc.uvigo.es

Abstract. In this work we present a system that reads automatically a car license plate using machine vision. In the paper first sections, we describe methods that are used to locate and segment the characters. Afterwards we describe all the pattern recognition methods that we tested and we see their different results. The final sections of the paper give the global system results and also our conclusions and future lines of work. Some of the techniques that we will describe come from the “state of the art”, but others were developed for this particular problem. This system has many practical applications such as parking check in/out, stolen car detection, security applications...

Palabras clave: Visión por Computador, Morfología Matemática, Redes Neuronales, OCR.

1. Introducción

Nuestro punto de partida fue una imagen del coche con placa de matrícula legible. En principio, queríamos que nuestro sistema reconociese por igual la matrícula trasera o la delantera. Tampoco queríamos “atar” el sistema a un tipo de placas determinado, esto es: que en la medida de lo posible, fuésemos capaces de reconocer matrículas de diferentes nacionalidades.

Empezamos trabajando con imágenes en escala de grises confiando en que las matrículas están diseñadas para ser reconocidas con facilidad, esto es: entre los caracteres y el fondo siempre hay una diferencia de brillo importante. En este punto debemos distinguir dos tipos de matrículas: las que tienen caracteres oscuros sobre un fondo claro (como las matrículas españolas antiguas o las nuevas matrículas de la comunidad europea) y las que tienen caracteres claros sobre un fondo oscuro (como las antiguas matrículas portuguesas y francesas). Como veremos, nuestro sistema es capaz de tratar placas de los dos tipos distinguiéndolas de forma automática.

Otra clasificación preliminar la establecimos entre las matrículas que tienen un renglón de caracteres (las rectangulares, que son las más habituales) y las que tienen dos renglones (las cuadradas, que aparecen en algunas motocicletas y también en algunos furgones). Nuestro sistema es capaz de distinguir automáticamente de qué tipo de placa se trata y tratarla adecuadamente.

Una primera versión del sistema se publicó en [1] en el año 1997. En ese momento el sistema ya tenía todos los elementos funcionales. A lo largo de los siguientes años

se han ido implementando nuevos métodos de reconocimiento, segmentación y localización de la placa.



Figura 1. Ejemplo de una imagen de partida con la placa de matrícula visible.

2. Localización y Segmentación de Caracteres

En esta sección vamos a ver cómo el sistema es capaz de encontrar los caracteres que después va a reconocer. Se trata de una etapa fundamental y que suele ser también la de mayor dificultad y la responsable de la mayor parte de los errores.

2.1 Localización de Posibles Placas

Para esta parte se han desarrollado dos métodos que expondremos brevemente: el primero basado en gradientes y el segundo basado en morfología matemática.

2.1.1 Método Basado en Gradientes

La idea básica es muy simple: los caracteres provocan picos en el gradiente. Además, esos picos siguen un patrón muy definido: si los caracteres son oscuros y el fondo es claro, a un valor de gradiente fuerte y negativo seguirá (en la misma línea y muy pegado) otro valor fuerte y positivo, la distancia entre ambos valores es una aproximación del grosor del trazo (en el caso de caracteres claros sobre fondo oscuro, ocurrirá exactamente lo contrario).

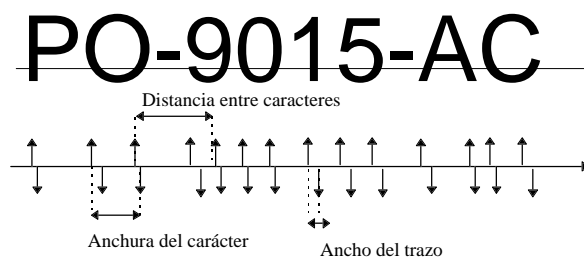


Figura 2. Método del Gradiente aplicado en una línea de imagen que contiene una placa de matrícula.

Nuestra idea fue aprovechar el patrón anterior, esto es: calcular el gradiente horizontal y buscar las líneas (o, mejor dicho, los trozos de línea o segmentos) que cumplan el

patrón anterior. Cuando tenemos el patrón dado en varios segmentos de líneas consecutivas, podemos intentar unirlos para formar un rectángulo (siempre que los segmentos estén aproximadamente alineados en vertical). Estos rectángulos ya son las posibles placas que se entregarán a la etapa siguiente. Podemos postprocesar esos rectángulos uniendo los que están muy pegados. Este proceso va a unir en una sola placa los dos renglones de las placas cuadradas, pero la información de que una placa viene de la unión de otras se debe conservar (porque en el caso de las placas cuadradas ayuda a la segmentación de renglones).

Nótese que este método (y también el siguiente) encuentra una serie de posibles placas. Esas placas van a ser procesadas todas por separado y no se tomará la decisión sobre cuál es la correcta hasta la etapa de segmentación de caracteres. Debido a esta forma de trabajar, encontraremos como posibles placas zonas tales como pegatinas, las letras de la marca del coche y otras zonas que produzcan un patrón de gradiente similar al de la placa. Ya veremos cuál es el método que se usa para discernir la verdadera placa. De momento, sólo mencionaremos una decisión de diseño: si en la imagen de entrada hay más de una placa, el sistema siempre se queda con la de mayor tamaño.

Para la implementación concreta utilizamos el conocido operador de Sobel [2] para calcular el gradiente. Sobre el resultado impusimos un umbral que se calcula fijando un porcentaje (por ejemplo un 5%) sobre el histograma, esto es: van a superar el umbral el 5% de los valores de gradiente. Se trata de umbral e histograma en valor absoluto; el gradiente se convierte a tres posibles valores: 1, 0 y -1.

2.1.2 Método Morfológico

La base de este método es un modelo diferente (aunque en el fondo comparte la misma idea del anterior): los caracteres son objetos formados por trazos finos y colocados sobre un fondo de color opuesto. Eso permite detectarlos con el operador morfológico [3] llamado “top-hat”, que se basa en borrar los caracteres y obtener la resta entre la imagen original y la imagen procesada.



Figura 3. Izquierda: imagen inicial. Centro: imagen procesada (sin caracteres). Derecha: “top-hat” obtenido restando la segunda imagen de la primera.

El borrado de caracteres se consigue, en este caso, aplicando un “closing” (utilizando morfología para escala de grises) a la primera imagen con un elemento estructurante que no quepa dentro del trazo de las letras. Si las letras fuesen negras sobre fondo blanco, habría que aplicar un “opening”. Este es el primer problema de este método: hay que aplicarlo dos veces para conseguir encontrar los dos tipos de matrícula.

Para completar la localización se binariza la imagen que resulta del “top-hat”, eligiendo un umbral con el conocido método de Otsu [4]. Después, se aplican técnicas de morfología binaria para conseguir unir todos los caracteres en un rectángulo y borrar todo lo que no pertenezca a dicho rectángulo. Para ello, hay que suponer conocida la máxima distancia entre caracteres y la máxima y mínima altura de éstos. Esta es la limitación principal del segundo método: sólo se puede aplicar si el tamaño de los caracteres es conocido. Generalmente, esto significa que se puede aplicar cuando la distancia entre la cámara y la placa es constante.

Una vez aislada, la placa que resulta suele ser un poco más pequeña que la real por lo que se agranda con una dilatación. Esta operación tiene el efecto de unir los renglones de las placas cuadradas.

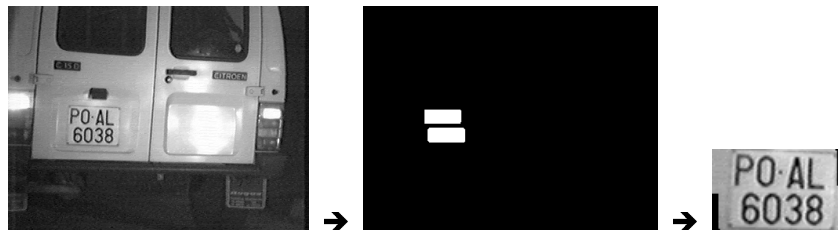


Figura 4. Izquierda: imagen inicial. Centro: detección de placas (renglones). Derecha: detalle de la placa que se acaba detectando.

2.1.3 Comparación e Integración

Hemos visto que el método morfológico tiene más limitaciones que el primero (y en las pruebas produjo mayor número de placas falsas). Los porcentajes obtenidos (por número de placas acertadas) son muy similares (ambos por encima del 95%). Sin embargo, el método morfológico mostró un mejor comportamiento en cierto tipo de placas. Podemos ver un ejemplo en la siguiente figura:



Figura 5. Izquierda: imagen segmentada usando gradientes. Derecha: imagen segmentada morfológicamente.

Vemos que el método morfológico se ve menos afectado por elementos extraños como los faros, cerradura y similares. Esos elementos producen máximos de gradiente pero no aparecen en el “top-hat”. Además, el método morfológico se mostró capaz en un tipo de matrículas que son casi imposibles para el método del gradiente: las motocicletas (en este caso, el número de gradientes que no proceden de la placa es enorme).

La solución pasa por combinar ambos métodos cuando sea posible (si la distancia no es fija, la morfología no vale). Aplicando ambos métodos obtendremos el doble de placas y aquellas con mucho solapamiento serán las más probables.

2.2 Binarización

Para la binarización también diseñamos dos métodos que luego decidimos combinar. El primero es el método de Otsu [4], el segundo es un método desarrollado por nosotros y que, para este caso, ha resultado superior.

2.2.1 Método de Otsu

Empezamos implementando el método de Otsu original (Figura 6) y también probamos una versión interpolada basada en [5]. Esta segunda versión se mostró capaz de tratar con placas donde la iluminación no era todo lo uniforme que sería deseable (Figura 8).



Figura 6. Izquierda: imagen que ha salido del localizador. Derecha: Binarización con el método de Otsu, umbral único.



Figura 7. Izquierda: imagen con iluminación incorrecta. Derecha: el umbral de Otsu interpolado resuelve el problema.

2.2.2 Método del Gradiente

El método del gradiente se basa en la siguiente idea: los valores grandes de gradiente separan a los píxeles en dos clases claramente diferenciadas. Se trata de aprovechar el gradiente que ya teníamos calculado de la etapa de localización. Podemos recorrer la matrícula y clasificar cada píxel. Eso nos permite obtener sumas y medias para cada clase. El umbral se calcula como:

$$\text{Umbral} = (\mu_N N_B + \mu_B N_N) / (N_N + N_B) \quad (1)$$

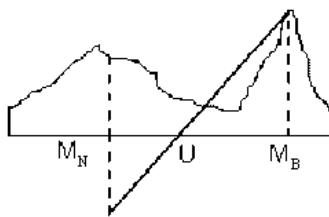


Figura 8. Interpretación geométrica de la Fórmula 1.

Donde μ_N es la media de los píxeles negros, μ_B es la media de los píxeles blancos y N_N y N_B son el número de píxeles negros y blancos, respectivamente. Para entender bien esta fórmula hemos dibujado una interpretación geométrica (Figura 8).

Este método fue comparado con el de Otsu original sobre más de 50 coches y se concluyó que las imágenes binarizadas con el método del gradiente eran más fáciles de segmentar. Como ejemplo, podemos ver el resultado con este método para la imagen de la Figura 7 (Figura 9).

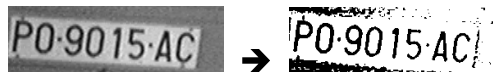


Figura 9. Izquierda: imagen que ha salido del localizador. Derecha: Binarización con el método del gradiente (no aparecen caracteres pegados a los bordes).

2.2.3 Comparación y Combinación de Ambos Métodos

Como dijimos, el método del gradiente se comporta mejor que el Otsu normal y además es muy rápido. El Otsu interpolado es el mejor de todos pero también es lento y sólo se requiere en casos de iluminación incorrecta como el de la Figura 7.

Con todo esto en mente diseñamos un método para combinar eficazmente el método del gradiente y el Otsu interpolado. Se trata de empezar aplicando el método del gradiente, medir la calidad del umbral obtenido y, si es pobre, aplicar Otsu interpolado.

Para medir la calidad del umbral, medimos cuantos píxeles de la imagen están en un entorno de ese umbral. Si el porcentaje es demasiado alto el umbral no es bueno ya que no se está encontrando el valle central del histograma bipolar.

2.2.4 Determinación del Color Activo

Con color activo nos referimos al color de los caracteres (foreground color). Aunque tengamos la imagen binarizada, todavía no sabemos si este color es blanco o negro. Esta decisión hay que tomarla para cada placa (ya que se puede llegar a tener en la misma imagen inicial una matrícula de cada tipo). Nótese, que los dos métodos de localización que presentamos son capaces de deducir esta información como dato adicional: el primer método sabe el color de los caracteres por el orden en que encuentra los valores de gradiente; mientras que el segundo método también lo sabe por el tipo de “top-hat” que le llevó a cada placa. Esa decisión puede ser reforzada contando el número de píxeles blancos y negros: los de fondo deberían ser mayoría. Eso es mejor hacerlo en el centro de la placa para evitar errores debidos a objetos extraños que se hayan unido a la placa.

Una vez determinado el color activo, si es el blanco, realizamos un negativo de esa placa. A partir de este punto, el color activo será siempre el negro.

2.3 Corrección de la Inclinación

Para corregir la inclinación, primero calculamos el ángulo usando el algoritmo que se presenta en [6] (usando la transformada de Hough) y después realizamos un giro.

Para el cálculo del ángulo no realizamos la transformada a todos los píxeles sino sólo a los del contorno horizontal, esto es: los píxeles donde hay un salto del blanco al negro en cada columna de la imagen.

Para optimizar el giro no usamos el método habitual de invertir la matriz de giro. Lo que hacemos es aplicar un giro directo sólo a los píxeles negros y colocar los píxeles girados sobre una nueva imagen, inicialmente blanca.

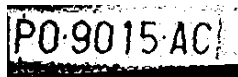


Figura 10. Imagen de la Figura 9 corregida por este método. En el giro se ha tenido en cuenta que los límites de la imagen han crecido. Hay cierto ruido granular debido al redondeo de coordenadas.

2.4 Segmentación

La segmentación en este sistema ha sido implementada de forma casi artesanal. Además, cada parte se ha diseñado de forma que sea capaz de funcionar, aun en caso de errores leves en las etapas anteriores. Por ejemplo: como paso previo a la segmentación se intenta eliminar los bordes de la matrícula (que, muchas veces, aparecen en las placas binarizadas). Sin embargo, nuestro segmentador es capaz de trabajar también en presencia de bordes. Pero cuando el borde ha sido eliminado, el funcionamiento del segmentador será menos propenso a errores.

Además, es en esta etapa donde se decide cuál es la placa correcta. La segmentación se va realizando en todas las placas ordenadas por su área (en orden decreciente). Cuando se encuentra una placa que contiene entre 4 y 12 caracteres correctos se considera esa como la buena y no se sigue segmentando. Normalmente, las placas incorrectas no producen caracteres en la segmentación. Este método es muy simple pero se ha mostrado eficaz ya que las placas falsas son poco probables. Si hay dos placas correctas en la imagen, se encontrará la mayor.

2.4.1 Segmentación de Líneas (Detección de Matrículas Cuadradas)

El sistema aquí podría fiarse de la información que tiene de la etapa de localización; ya que, normalmente, las placas cuadradas se detectan como dos renglones que después se unen. Sin embargo, eso no es siempre así; por ejemplo cuando la placa cuadrada está inclinada los renglones tienden a mezclarse. Por eso, aquí se realiza una comprobación adicional usando la proyección vertical. La proyección debería tener un mínimo local en el centro, similar (tal vez algo mayor) a los que aparecen en los extremos. Ese mínimo (si existe) se usa como línea de ruptura.

2.4.2 Segmentación de Caracteres Individuales

Para la segmentación de cada renglón se ha implementado un método que recorre una línea central buscando caracteres. El recorrido empieza en el centro del rectángulo y va avanzando alternativamente a izquierda y derecha. Cada vez que se encuentra una mancha conectada se estudia para determinar si es un carácter o no. Si se avanza un

espacio prudencial sin encontrar nada, el recorrido se para por ese lado, aun sin llegar al final. A las manchas se les aplican muchas reglas: tamaño mínimo y máximo, razón de aspecto y otras similares. Cuando se encuentra una mancha que puede corresponder a dos caracteres pegados se estudia si hay un punto óptimo para partirla (usando proyección horizontal). Una vez extraídos todos los caracteres se eliminan aquéllos que tengan una altura o un área excesivamente diferente al resto. Por anchura, sólo se eliminan los que tienen un valor excesivo respecto del resto (no se elimina nada por poca anchura debido a caracteres como la **I** o el **1**). Este método se ha depurado a través de un largo proceso de ensayo y error.

3. Métodos de Reconocimiento de Patrones

3.1 El método de la Proyección

En este método obtenemos las proyecciones verticales y horizontales del carácter a reconocer y las comparamos con un alfabeto de caracteres posibles. La distancia mínima determina el carácter ganador.

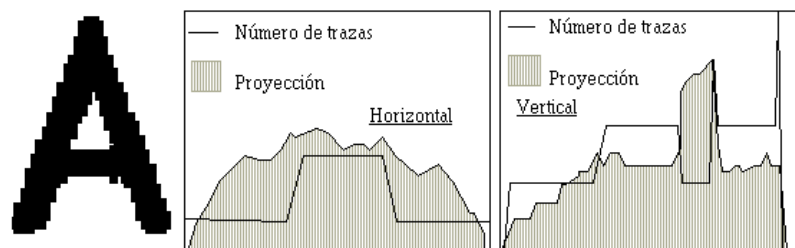


Figura 11. Ejemplo de vectores de proyecciones y trazas. En el cálculo de trazas hay que tener cuidado con el ruido granular ya que puede incrementar la cuenta.

Para comparar vectores de proyección de tamaños distintos (suponemos que el carácter de entrada siempre es más pequeño que el modelo) hemos desarrollado una función de comparación heurística. Además de la proyección obtenemos el vector de trazas. Éste se construye contando el número de trozos negros que se encuentran al recorrer cada línea o columna de la imagen. Podemos decir que el vector de trazas divide a la curva de proyecciones en zonas. Cada valor de la proyección de entrada se va a comparar con un trozo del modelo (así se compensa la diferencia de tamaños). Si la traza de entrada en ese punto no coincide con alguna traza del subvector del modelo, se aplica una penalización especial. Las distancias horizontal y vertical, simplemente, se suman.

3.2 Métodos Estructurados

Llamamos métodos estructurados a aquéllos que se componen de una extracción de características y un clasificador. Vamos a exponer 2 métodos para cada parte lo que nos permite combinarlos de 4 formas distintas. Antes de la extracción, siempre se realiza un proceso de escalado (manteniendo la relación de aspecto) y otro de

centrado (añadiendo columnas en blanco) que permite trabajar con caracteres de tamaño fijo [7].

3.2.1 Extracción Basada en el Gradiente de Kirsch

Esta extracción se basa en obtener los componentes de la imagen en cuatro direcciones principales: horizontal, vertical, diagonal primera (desde la esquina superior izquierda hasta la inferior derecha) y diagonal segunda. Esto se consigue utilizando el operador gradiente de Kirsch [2] (por ejemplo, el gradiente horizontal nos da la componente de dirección vertical). Combinando las componentes direccionales y la imagen original, tras una serie de escalados y diezmados se obtiene un vector de 80 números reales. Éste es el vector de características.

Nosotros hemos calculado el vector de características de Kirsch para más de 1000 caracteres extraídos de matrículas reales. Después, calculamos el parámetro Λ de Wilks [8] para comprobar si existía separabilidad en esos datos. Al obtener un valor del orden de 10^{-20} se demuestra que sí existe y que es muy buena (el parámetro debería resultar menor que 1, un valor menor implica mayor separabilidad).

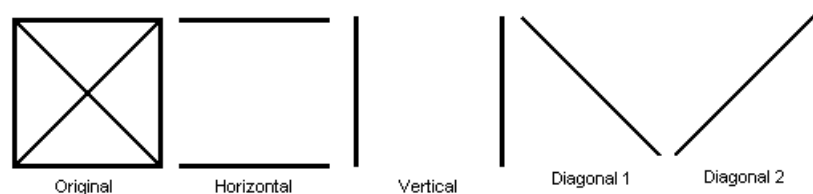


Figura 12. Ejemplo simple del significado de las características direccionales.

3.2.2 Extracción Basada en la Transformada de Karhunen-Loève

La transformada KLT [9] se basa en diagonalizar la matriz de autocorrelación de los datos. Después de eso, es posible expresar esos datos como combinación lineal de sus "componentes principales" o autovectores. Esas componentes tienen mayor importancia cuanto mayores sean los autovalores asociados. Por tanto, es posible retener un número de componentes que serán los que se usen para reconocer (los demás, ni siquiera se calculan).

Nosotros decidimos escalar los caracteres a un tamaño 8x8 y después retener los 32 primeros coeficientes de la KLT (la mitad). El parámetro de Wilks para este caso resulta de orden de 10^{-12} con lo que se demuestra la existencia de separabilidad.

3.2.3 Clasificación Utilizando Redes Neuronales Probabilísticas

La Red Neuronal Probabilística o PNN (Probabilistic Neural Network) [10] se basa en el cálculo de la probabilidad a posteriori, esto es: se trata de un clasificador que emplea el método de máxima verosimilitud. Para ello, se realiza un modelado gaussiano de las características de entrada, centrado en las muestras de entrenamiento.

La PNN no tiene una verdadera etapa de entrenamiento sino que recorre todo el conjunto clasificado en cada ejecución. Esto da lugar a un método muy lento. En nuestra implementación, hemos hecho una reducción aleatoria del conjunto de

entrenamiento que permite mejorar mucho los tiempos (nos hemos quedado con 20 muestras por clase, elegidas aleatoriamente).

3.2.3 Clasificación Utilizando Perceptrón Multicapa

El perceptrón multicapa o MLP (Multi Layer Perceptron) [11] se basa en el entrenamiento de los pesos de redes neuronales artificiales.

Nosotros hemos empleado dos esquemas diferentes: para la extracción de Kirsch usamos un esquema de conexiones reducidas descrito en [7] (que refleja el significado geométrico de esta extracción) y para la extracción KLT usamos un esquema de conectividad total. El algoritmo de entrenamiento empleado fue el **back-error-propagation** [11] normal.

3.3 Un Método Híbrido

Este método consiste en aplicar primero el método de la proyección para seleccionar un conjunto de candidatos y después aplicar la PNN sólo a esos candidatos. El número de candidatos que elegimos es de 5 como máximo, esto es: seleccionamos los 5 caracteres de menor distancia a la entrada; pero si hay un salto importante entre el segundo y el tercero (por ejemplo), sólo cogeremos 2. Gracias a esto, a veces, no hay que usar la PNN porque el primer método ya obtuvo resultados concluyentes.

3.4 Pruebas, Resultados y Comparación de Métodos

Estos resultados proceden de la prueba con 217 caracteres extraídos de 30 placas. Los reconocedores fueron entrenados con 1054 caracteres extraídos de 155 placas. Los conjuntos de prueba y entrenamiento son disjuntos. Los resultados fueron:

Extracción Características	Clasificador	PNN:	MLP:
Componentes de Kirsch:		97.68%	99.07%
KLT:		96.29%	98.61%
Método de la Proyección:			91.74%
Método Híbrido:			97.92%

Tabla 1. Porcentajes de acierto.

El método híbrido llega a mejorar la exactitud de la PNN (debido a la conjunción de dos métodos, con ventajas y desventajas complementarias). El método de la proyección es el más simple y el menos preciso. Ahora veremos los tiempos medios de ejecución (medidos en un Intel Pentium MMX a 133 MHz):

Extracción Características	Clasificador	PNN:	MLP:
Componentes de Kirsch:		98.44	31.03
KLT:		51.38	2.03
Método de la Proyección:			15.26
Método Combinado (apartado 2.2.5.6):			52.14

Tabla 2. Tiempos medios de ejecución (en milisegundos).

La extracción de características KLT es más rápida que la de Kirsch. Además, el reconocedor MLP es mucho más rápido que la PNN. Nótese además que el método combinado presenta un valor intermedio entre los dos métodos componentes.

A la vista de estos resultados pensamos en elegir como OCR definitivo la extracción KLT seguida de MLP. Este método es, con mucho, el más rápido y su porcentaje de acierto es plenamente aceptable. Sin embargo, debemos tener en cuenta que el número de muestras de entrenamiento es, teóricamente, insuficiente para un MLP (debería ser, al menos, 5 veces el número de pesos del MLP) con lo que ésta podría ser una elección arriesgada. ¿Por qué, aun estando infraentrenado, ha dado buenos resultados con conjuntos de entrenamiento y prueba disjuntos? Tal vez, la se trata de que este es un problema donde la generalización es fácil por ser caracteres impresos, siempre en mayúscula y con fuente constante (recuérdese la gran medida de separabilidad obtenida). Si aún así decidimos descartar el MLP, el método híbrido ofrece unas características muy interesantes tanto en tiempo como en acierto. Los resultados globales de la sección 4 se han obtenido con este último OCR.

4. Pruebas y Resultados del Sistema Global

El sistema en su conjunto se probó con 185 imágenes de coches. La tasa de reconocimiento global (por placa) fue del 84.11%. Nótese que por carácter se obtuvieron porcentajes de más del 97% pero sin tener en cuenta los problemas debidos a la localización y segmentación. Aplicando técnicas de rechazo (el sistema renuncia a reconocer los casos que considera más difíciles, esos casos rechazados se pasan a un operador humano) se consiguió aumentar el porcentaje de acierto al 94.07% de los no rechazados. El porcentaje de coches rechazados fue del 10.59%.

El tiempo medio de ejecución del proceso completo es de 1.5 segundos por imagen (en un Intel Pentium MMX a 133 MHz).

5. Comparación con Proyectos Similares

Un sistema similar fue desarrollado en la Universidad de Tras-os-Montes e Alto Douro (Portugal) [12]. Este sistema consigue localizaciones correctas en el 100% de coches parados, el 99.5% de coches en movimiento con condiciones climáticas favorables y el 88.5% de coches en movimiento con condiciones adversas (nuestro sistema localiza una matrícula incompleta el 2% de las veces y no encuentra nada en otro 2%, acertando en el 96% restante; en esta prueba mezclamos coches parados y en movimiento, en condiciones buenas y malas). El OCR de este sistema obtiene un porcentaje sobre caracteres bien segmentados de 98.4%. El porcentaje global no aparece en la publicación.

También hemos estudiado las especificaciones de varios sistemas comerciales. Las prestaciones al usuario son similares a las del nuestro pero ahí acaba la comparación porque ninguno da datos de los porcentajes de acierto, simplemente afirman que son altos y tampoco mencionan nada sobre los métodos empleados.

6. Conclusiones

Hemos utilizado técnicas que son individualmente muy simples pero su adecuada combinación nos ha permitido desarrollar un sistema potente y con muchas aplicaciones prácticas.

7. Líneas Futuras

Actualmente, ya estamos probando un nuevo reconocedor basado en la distancia de Hausdorff. En el futuro, nos gustaría probar con sistemas que son capaces de realizar el reconocimiento sin segmentación [13].

Agradecimientos. Los autores desean expresar su agradecimiento a todos los estudiantes que colaboraron haciendo que este sistema llegara a ser una realidad (y a los que siguen colaborando en su evolución). A saber: José Luis Fernández Lijó, Fidel Pita Sande, Miguel Pérez Iglesias, Miguel Ángel Piñero Martínez, Juan Antonio López Bueno y Maite García Saburido.

Referencias

- [1] X. Fernández et al. "An O.C.R. for V.L.P.'s (Vehicle License Plates)". Proceedings of ICSPAT-97. San Diego. Septiembre.
- [2] A. K. Jain. "Fundamentals of Digital Image Processing". Ed: Prentice-Hall. 1989.
- [3] J. Serra. "Image Analysis Using Mathematical Morphology". Ed: Academic Press. 1984.
- [4] N. Otsu. "A Threshold Selection Method for Gray Level Histograms". IEEE Transactions on System, Man and Cybernetics. Enero, 1979.
- [5] J. Ohya, A. Shio, S. Akamatsu. "Recognizing Characters in Scene Images". IEEE Transactions on Pattern Analysis and Machine Intelligence. Febrero, 1994.
- [6] G. S. D. Farrow, M. A. Ireton, C. S. Xydeas. "Detecting the Skew Angle in Document Images". Signal Processing: Image Communication. 1994.
- [7] D. Cruces Álvarez, F. Martín Rodríguez, X. Fernández Hermida. "Printed and Handwritten Digits Recognition Using Neural Networks". ICSPAT-98. Toronto.
- [8] M. M. Tatsuoaka. "Multivariate Analysis". John Wiley & Sons. 1971.
- [9] P.J. Grother. "Karhunen-Loève Extraction for Neural Handwritten Character Recognition". Applications of Artificial Neural Networks III. Orlando. 1992.
- [10] J. L. Blue et al. "Evaluation of Pattern Classifiers for Fingerprint and OCR Applications". Pattern Recognition. Vol 27. 1994.
- [11] S. Haykin. "Neural Networks, A Comprehensive Foundation". Ed. McMillan.
- [12] J. Barroso et al. "Number Plate Reading Using Computer Vision". ISIE-97. Guimarães.
- [13] M. A. Özdil, F.T. Yarman-Vural, N. Arica. "Optical Character Recognition Without Segmentation". Lecture Notes on Computer Science. Nº 1311, Vol 2. 1997.