# LOCALIZACIÓN Y SEGMENTACIÓN DE CARACTERES EN FORMULARIOS MANUSCRITOS

*Fernando Martín Rodríguez* *David López Crespo* Francisco Parada Loira

Departamento de Tecnologías de las Comunicaciones
Universidad de Vigo
fmartin@tsc.uvigo.es dlopez@tsc.uvigo.es fparada@tsc.uvigo.es

**RESUMEN**

En este trabajo se describe un sistema capaz de encontrar y segmentar caracteres manuscritos procedentes de formularios (la estructura del formulario es conocida). Los formularios se usan profusamente en las tareas administrativas de múltiples organizaciones. Este sistema es la primera parte de un lector automático que podría automatizar las tareas de introducción de datos (falta el reconocedor). En el estado del arte actual existen más sistemas de este tipo, pero todavía sufren de muchos problemas [1].

## 1. INTRODUCCIÓN

Este sistema va a trabajar con imágenes en escala de grises (8 bits por píxel) obtenidas mediante escaneado El objetivo final es la segmentación total de los caracteres presentes en la imagen inicial. Esto es: la obtención de imágenes de pequeño tamaño que recorten (lo más exactamente posible) a cada uno de los caracteres iniciales.

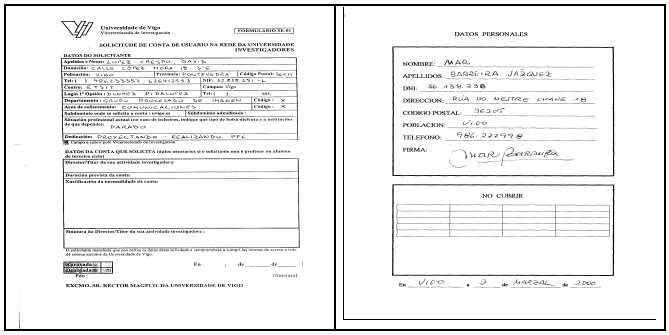


Figura 1. *Ejemplos de formularios.*

Vamos a describir ahora la estructura del resto del artículo:

- En la sección 2 (suposiciones de partida), se describen las suposiciones iniciales que usa el sistema. La suposición básica es el conocimiento de la estructura del formulario. Para describir esta estructura se ha definido un formato de fichero especial. También comentaremos las "normas de escritura" para obtener un buen funcionamiento.
- En la sección 3 (métodos), se explican todos los algoritmos que usa el sistema en el mismo orden de su aplicación: preprocesado (binarización y corrección de inclinación), encajado y extracción de líneas y, por último, la segmentación propiamente dicha.
- En la sección 4 (resultados), se exponen los resultados finales: porcentajes de acierto, conclusiones y líneas futuras.

## 2. SUPOSICIONES DE PARTIDA

### 2.1. Definición del Formulario

Los formularios se describen a través de ficheros ***.FRM**. Un fichero FRM describe a todos los formularios que tengan la misma estructura: número y posición de los campos a rellenar (se puede definir un fichero FRM para el formulario de matrícula, otro para el formulario de solicitud de título...).
El formato FRM es un formato ASCII que se basa en 5 palabras clave. La palabra **RESOLUCION** indica la resolución de trabajo con el tipo de formularios de interés (normalmente, trabajamos siempre con 200 dpi's). **L-VERT** y **L-HOR** sirven para indicar las posiciones de la primera línea horizontal significativa y la primera línea vertical, el usuario debe señalar cuáles son estas líneas. Esas líneas serán usadas en el proceso de encajado del formulario (determinación del desplazamiento entre el formulario de entrada y el formulario descrito por el fichero **FRM**).
Las palabras más importantes son **MACRO-CUADRO** y **CUADRO**, que se usan para definir los diferentes cuadros de texto presentes en el formulario. La organización es jerárquica: los macrocuadros están numerados, los cuadros siempre están dentro de macrocuadros y también están numerados. Un cuadro corresponde con un campo a ser rellenado.

```
RESOLUCION:200 ppp
L-VERT:148
L-HOR:335
MACRO-CUADRO_1: 507,148          1061,1082
        CUADRO_1_1 576,472       618,1036
```

Figura 2. *Ejemplo de fichero FRM.*

### 2.2. Normas de Escritura

Se supone que el usuario ha seguido las normas habituales para escribir en formularios. Esto es: ha escrito en letras mayúsculas y sin unir caracteres (el sistema es capaz de separar caracteres unidos "por accidente" pero no escritura continua).

## 3. MÉTODOS

### 3.1. Preprocesado

El preprocesado consiste en la aplicación de los procesos siguientes:

- Binarización: para ello se implementa el algoritmo clásico de Otsu [2].
- Corrección de la Inclinación: primero se detecta usando la transformada de Hough [3] y después se corrige con un giro.

### 3.2. Encajado y Extracción de Líneas

El encajado consiste en encontrar el desplazamiento (tanto horizontal como vertical) del formulario de entrada respecto a uno de referencia (el que se describe en el fichero **FRM**). Para esto se sigue un procedimiento en dos pasos:

- Primero se encuentra una primera aproximación buscando la primera línea horizontal significativa (recordar que su posición venía en el fichero **FRM**). Por supuesto, también se busca la primera línea vertical significativa.
- Se prueba con cierto intervalo en torno a la aproximación inicial. Se halla la correlación (muy rápida por ser imágenes binarias) entre la imagen original desplazada y la imagen de referencia. Por supuesto, se toma el máximo.

Las líneas (o campos) de texto corresponden con los cuadros descritos en el fichero **FRM**, con lo que una vez encajado se recortan sin más.

### 3.3. Segmentación

La segmentación se realiza en dos fases:

- Gruesa: se calcula la proyección horizontal. El sistema deduce los umbrales a aplicar y realiza una primera segmentación. También se estudia la distancia típica entre caracteres (mediana de las distancias), cuando esa distancia se supera ostensiblemente se considera que hay un espacio (separación de palabras). Esta etapa separa las palabras correctamente. Cuando se encuentran caracteres demasiado pequeños se eliminan (ruido, signos de puntuación... ).
- Fina: esta segunda parte da por buena la segmentación de palabras pero estudia más detenidamente la segmentación de caracteres. Primero se halla (con la mediana) la anchura típica del carácter. Cuando ese valor se supera en más del 60%, suponemos que hay dos caracteres unidos "por accidente". Para separarlos se busca un punto de ruptura usando primero la proyección horizontal. Si no se encuentra un punto claro, se utiliza la correlación de dos columnas consecutivas ("Break Cost" [4]).

Nótese que el resultado está organizado jerárquicamente, esto es: dividimos las líneas en palabras y las palabras en caracteres.

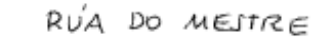


Figura 3. *El sistema es capaz de separarla E y la S en la figura*

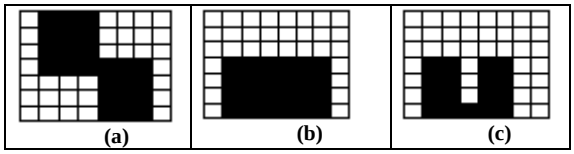


Figura 4. *Diferencias entre la proyección (b) y el "break-cost" (c) (la imagen original es a).*

## 4. RESULTADOS

### 4.1. Porcentajes de Acierto

El sistema se ha probado con 31 formularios de tipos distintos. Los resultados obtenidos fueron.

| | |
|---|---|
| Caracteres Totales: | 3563 |
| Caracteres Correctos: | 3516 |
| Caracteres Falsos: | 89 |
| Caracteres Perdidos: | 47 |

Con lo que obtenemos:
% Acierto = C/NT*100=98.7%
% Falsos = F/(C+F)*100=2.5%

### 4.2. Conclusiones

Se ha desarrollado un segmentador con resultados aceptables y funcionamiento muy sencillo. El sistema es la primera parte de un reconocedor de formularios completo. El sistema completo tiene muchas aplicaciones en la automatización de los procesos de introducción de datos.

### 4.3. Líneas Futuras

Este trabajo admite muchas mejoras, algunas de ellas son:

- Desarrollo de un OCR basado en redes neuronales [5].
- Aprovechamiento de ese reconocedor para hacer segmentación realimentada [4].
- Mejora de la segmentación de caracteres unidos.

## 5. REFERENCIAS

[1] Varios. "Design, Integration and Evaluation of Form-Based Handprint and OCR System". NISTIR 5932. 1996.

[2] N. Otsu. "A Threshold Selection Method for Gray Level Histograms". IEEE transactions on System, Man and Cibernetics. 1979.

[3] G.S.D. Farrow et al. "Detecting the Skew Angle in Document Images". Signal Processing: Image Communication, Vol. 6. 1994.

[4] S. Tsujimoto, H. Asada. "Resolving Ambiguity in Segmenting Touching Characters". Structured Document Image Analysis. Springer-Verlag. 1992.

[5] D. Cruces, F. Martín. "Printed and Handwritten Digits Recognition Using Neural Networks". ICSPAT-98, Vol 1, pp 839-843. Toronto (Canadá). 1998.