

Título : " UMBRALIZACION ADAPTATIVA EN TIEMPO REAL ".

Por :

M.A. Fernández (*) y J. Mira (**).

(*) Departamento de Informática. Universidad de Castilla la Mancha.
E. U. P. A.

Campus Univ.- Avda. de España s/n. Albacete (02006).

(**) Departamento de Informática y Automática. Facultad de Ciencias
Físicas de la U.N.E.D.

Aptdo. 60141. Madrid (28040).

Abstract:

Este trabajo presenta un algoritmo para el cálculo de umbral adaptativo en cada punto de imagen. Se realiza un estudio sobre la realización hardware para su funcionamiento en tiempo real, y se comentan los resultados obtenidos mediante una simulación de este sistema realizada en C sobre una tarjeta comercial de digitalización de imagen. Se comentan también otras posibles aplicaciones de este algoritmo en diferentes campos de tratamiento de la señal.

Palabras Clave: UMBRALIZACION / ADAPTATIVA / TIEMPO REAL / TRATAMIENTO DE SEÑAL / VISION / SUAVIZADO / PASO BAJA.

Planteamiento del problema:

Cuando intentamos clasificar los puntos de una imagen, umbralizando alguna de sus características, de nivel de gris o el valor obtenido al someter la imagen a cualquier tipo de filtro, de borde, o de otro tipo [1]; el cálculo del umbral, juega un papel decisivo en el éxito de nuestro propósito [2] [3]. En multitud de escenas, aparecen zonas de imagen de diferente naturaleza en las que el parámetro clasificador toma valores muy distintos. Por lo tanto,

si deseamos clasificar adecuadamente toda la imagen, el cálculo del umbral debe realizarse atendiendo al comportamiento del entorno del punto a umbralizar, ya que en la imagen completa aparecerán multitud de elementos que nos llevarían a definir un umbral " general ", poco fino para las zonas de imagen en las que el parámetro clasificador toma valores mas sutiles [4]. Un ejemplo claro de este caso, sería la detección de un misil o blanco lejano sobre un fondo celeste, acompañado en la escena por elementos de nubes, oleaje u horizontes montañosos.

El propósito de este trabajo es, producir un sistema capaz de resolver este problema trabajando en tiempo real. Supondremos inicialmente, que los datos que vamos a procesar proceden de una cámara de video con un estándar de señal CCIR. Esto nos dá una frecuencia de información, con una digitalización a 10 Mhz., de un nuevo dato cada 100 nseg, bien sea este dato un nivel de gris, bien corresponda a la respuesta de un filtro actuando sobre sobre la señal original.

En 100 nseg., no es fácil el análisis del histograma del entorno del punto [5], por lo que planteamos, para el ajuste de umbral, calcular la media de los valores del entorno del punto y definir el umbral ponderado este valor [6].

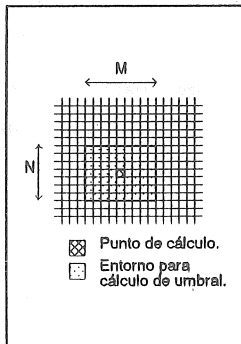


Fig. 1. Entorno.

Hemos considerado en este trabajo, como entorno que interviene en el cálculo de umbral para cada punto de imagen, el conjunto de puntos que le rodea, imponiendo para este entorno una geometría rectangular. Como podrá verse mas adelante, nuestro sistema es flexible en cuanto a las dimensiones del rectángulo que definimos como entorno, así como al lugar geométrico que ocupa el punto de cálculo dentro de este (ver fig. 1).

El cálculo que deberemos realizar, sera pues la suma de los valores de los puntos del entorno definido alrededor del punto de cálculo, dividir por el número de éstos, y ponderar el resultado mediante una constante para generar un umbral más o menos exigente. No es pues la dificultad del cálculo lo que complica la solución, sino el volumen de éste.

Si tomamos como dimensiones para el entorno, por ejemplo 10 por 10 puntos de imagen, tendremos que operar con 100 sumandos, ponderar este valor y luego

comparar con el umbral en tan solo 100 nseg. Estas son dimensiones un tanto reducidas. Sería lo normal, dimensiones de entorno mayores como 40 por 30 o 70 por 100, por ejemplo, esto nos llevaría a tener que operar con 1200 o 7000 sumandos cada 100nseg., si deseamos realizar el cálculo en tiempo real.

Algoritmo para computación en tiempo real :

Podemos dividir el proceso en cuatro etapas distintas.

1) Realización del sumatorio de todos los puntos del entorno:

Proponemos para realizarlo, el siguiente algoritmo que denominamos de "Cómputo Acumulado".

Descompondremos el sumatorio como indica la figura 2.

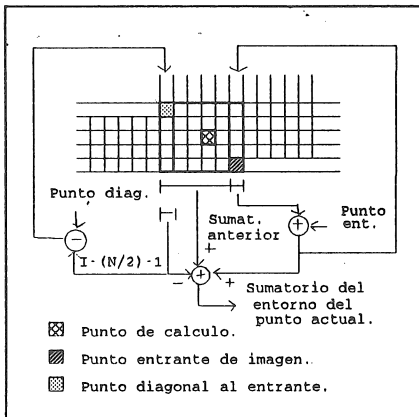


Fig. 2. Algoritmo.

El cálculo se simplifica, si consideramos que al realizar el sumatorio de los valores del entorno de un punto de imagen, podemos utilizar los cálculos que realizamos para su vecino inmediatamente anterior. Es decir, para calcular el nuevo sumatorio, tan solo tendremos que restar al anterior la columna de entorno que desaparece de éste, y sumarle los valores de la columna que para el nuevo punto de cálculo entra a formar parte del entorno. Sería, por tanto, necesario

almacenar los datos requeridos para realizar estas operaciones. Este almacenamiento se realiza mediante memorias tipo FIFO.

El grado de complicación del cálculo realizado, así como el volumen de este, no dependen de el tamaño del entorno que elijamos, ni de la posición que ocupe dentro de este el punto a umbralizar.

2) Cálculo del valor medio:

Podemos optar en este punto, por dividir el sumatorio realizado por el numero de puntos, o multiplicar por este el valor del punto de cálculo.

Para los puntos cercanos al borde de imagen, se genera un tratamiento especial que realiza la media, solamente con los puntos de su entorno que

forman parte de la escena.

3) Cálculo del umbral. El tipo de umbral a utilizar dependerá del problema concreto. Nosotros hemos trabajado con dos tipos de umbral:

- En un caso fijamos el umbral, incrementando la media calculada, con una constante fija en toda la imagen.

- En el otro caso incrementamos la media con una constante distinta en cada punto y proporcional a la media obtenida para su entorno.

4) El ultimo paso consiste en someter el valor del punto de cálculo, al umbral calculado.

Obtenemos de esta manera una imagen binarizada, utilizando un umbral adaptativo dependiente del valor de los puntos del entorno de cada punto de imagen.

Estudio de viabilidad hardware :

Podemos ver un esquema de bloques para una realización hardware que permite la computación en tiempo real, en la figura 3. Consta básicamente de memorias de almacenamiento, un multiplicador, un comparador y dos sumadores de alta velocidad. El almacenamiento de los valores de pixel se resuelve utilizando memorias estáticas de alta velocidad del tipo IDT71B024, y memorias FIFO, del tipo IDT72215A, para almacenar los datos de las columnas internas y externas al entorno del punto de calculo. El bloque de lógica de control, resuelve el flujo de datos entre operadores y memorias, incluyendo el

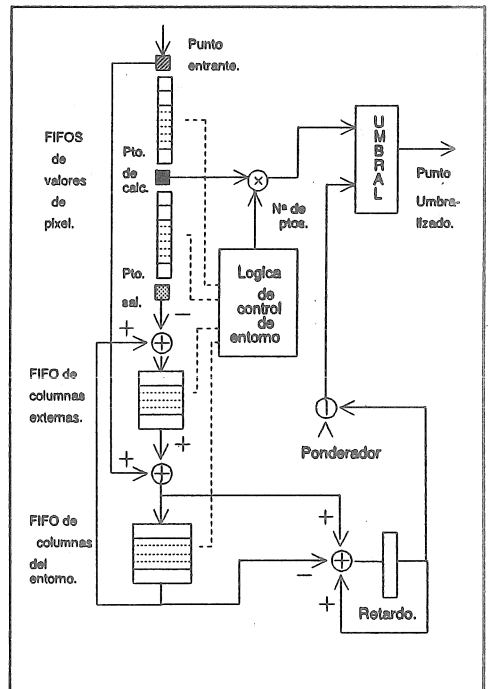


Fig. 3. Esquema de bloques HW.

control de número de puntos de entornos no completos por localizarse en los

bordes de imagen. El resto de las operaciones de suma y multiplicación se realizan de manera segmentada utilizando EPLD de ALTERA del tipo EP600 y EP900, al igual que el bloque de lógica de control.

Simulación :

Para la simulación del sistema se ha trabajado en un entorno compuesto por una cámara de televisión CCD, blanco y negro de norma CCIR, una placa comercial de digitalización de imagen de la casa Matrox, y un grupo de ordenadores tipo PC. El sistema se ha simulado mediante rutinas en C, y se ha valorado digitalizando varias imágenes significativas y sometiénolas a proceso según el método y algoritmos propuestos. Los resultados obtenidos se comentan a continuación.

Resultados y Conclusiones :

Como muestra de los resultados obtenidos, exponemos aquí algunos ejemplos significativos.

Se ha sometido la imagen de la figura 4, sobre la que se imprime el histograma de la misma, a varios tipos de filtrado basados en el nivel de gris, obteniendo los resultados que comentamos a continuación.

- La fig. 5, muestra el resultado de someter la imagen a un umbral fijo en todos sus puntos. La elección de este umbral se ha realizado analizando el histograma. El umbral se fijo en el nivel de gris 135 (punto 1 de la fig. 4). Puede verse que conseguimos segmentar solo una zona de la imagen, siendo defectuosa la umbralización en zonas de distinto nivel de gris.

- La fig. 6, podemos ver el resultado de someter la imagen a un umbral del mismo tipo, pero con valor 176 (punto 2 de la fig. 4). Conseguimos segmentar otra zona de la imagen, pero perdemos gran cantidad de información en las zonas anteriormente bien segmentadas.

Un umbral de valor igual a la media de todos los valores de los puntos de imagen, da resultados mas uniformes pero de peor calidad, como puede verse en la figura 7.

Utilizando el algoritmo que proponemos, puede someterse la imagen a umbrales adaptativos, consiguiendo resultados como los que muestran, para distintos tamaños de entorno y tipo de umbral, las figuras 8, 9, 10.

- Fig. 8 . Umbral adaptativo:

Umbral = media de los valores del entorno de 20 x 20.

Se detecta la información en todas las zonas de imagen. En zonas de

niveles de gris muy similares, se detectan los puntos con pequeñas diferencias y el ruido.

- Fig. 9 . Umbral adaptativo:

Umbral = media de los valores del entorno de 60 x 60.

Los resultados son similares al caso anterior, pero puede apreciarse la diferencia de comportamiento en las fronteras.

- Fig. 10 . Umbral adaptativo:

Umbral = media de los valores del entorno de 20 x 20 + el mismo valor multiplicado por una cte.=0.05 en cada punto.

Con un umbral como este, adaptativo, pero mas exigente, eliminamos el ruido en las zonas donde no existe mucha diferencia de niveles de gris.

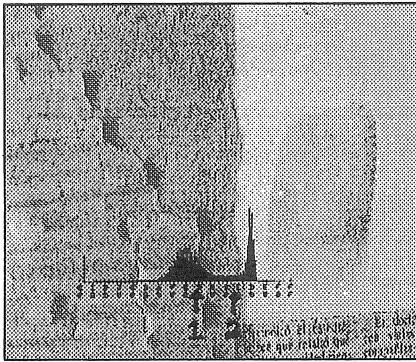


Fig. 4. Imagen original con histograma de la misma.

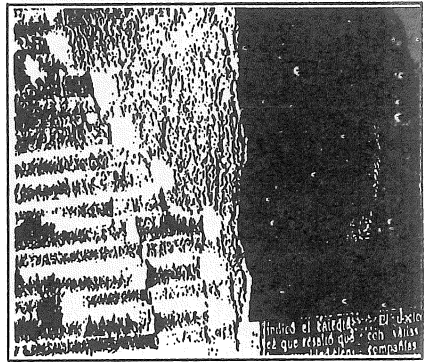


Fig. 5. No adaptativo.
Umbral punto 1 fig. 4.

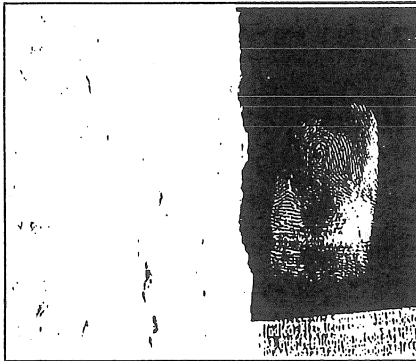


Fig. 6. No adaptativo.
Umbral punto 2 fig. 4.

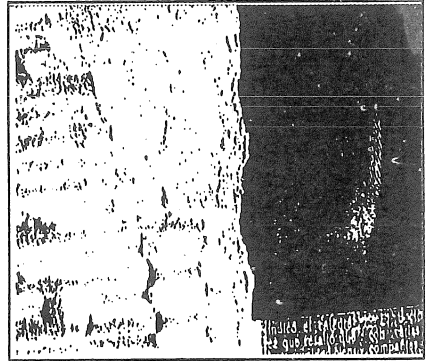


Fig. 7. No adaptativo.
Basado en media de pantalla

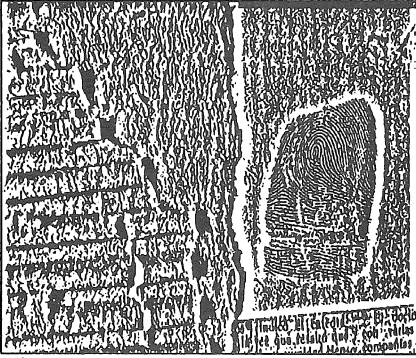


Fig. 8. Adaptativo. Media de entorno 20 x 20.

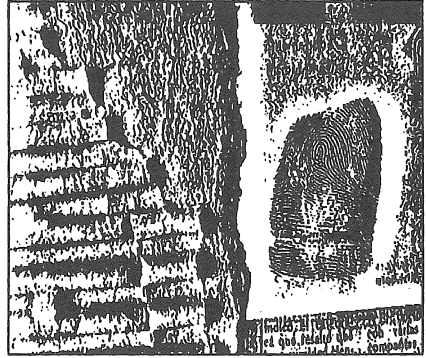


Fig. 9. Adaptativo. Media de entorno 60 x 60.

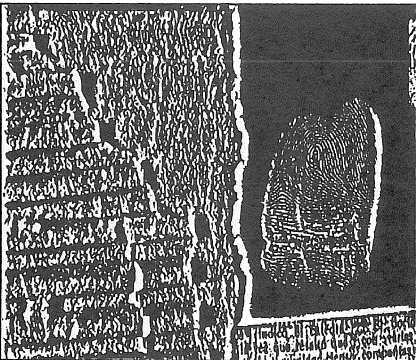


Fig. 10. Adaptativo. Media modificada entorno 20 x 20.

Podemos afirmar que el sistema de umbral adaptativo, basado en algoritmos de cómputo acumulado, es realizable mediante hardware para conseguir, en tiempo real, un valor de umbral en cada punto de imagen dependiente del valor de los puntos de su entorno.

Atendiendo a los resultados obtenidos en la fase de valoración, este tipo de umbralización consigue clasificar gran cantidad de información que no detectan los sistemas basados en umbrales fijos para toda la imagen. Puede destacarse que, el método que pondera la media para fijar el umbral, las dimensiones del entorno elegidas, y la situación del punto de cálculo dentro del entorno, juegan un papel importante en los resultados obtenidos, y pueden optimizarse para problemas concretos.

Este tipo de algoritmos de cómputo acumulable, realizado mediante un hardware similar al expuesto, puede ser utilizado para otras tareas referentes al proceso de imagen, como puedan ser el suavizado, o a otros campos de tratamiento de la señal que requieren el cálculo de la media de un entorno del punto en tiempo real.

- Agradecemos el apoyo económico prestado por los proyectos CICYT TIC-880315 y TIC-92-0136.
- Agradecemos a Luis Alonso Almansa Daimiel, la colaboración prestada en la fase de simulación.

Referencias :

- [1] Ikram E. Abdou & William K. Pratt. "Quantitative Desing and Evaluation of Enhacement/Thersholding Edge Getectors".
- [2] Ballard, Dana H. & Brown, Christopher M. . "Computer Visión". Prentice Hall (1982). Apartados 3.3.2 y 5.3.
- [3] Kohler, Ralf . "A Segmentation System Based on Thresholding". Computer Graphics and Image Processing (1981).
- [4] Robert J. Schalkoff. " Digital Image Processing and Computer Visión ". Wiley. Página 178 y siguientes.
- [5] fu K. S., Gonzalez R. C. & Lee C. G. S.. "Robótica: Control, Detección, Visión e Inteligencia". Apartados 7.6.5. y 8.2.2.
- [6] E. R. Davis. "Machine Vision: Theory, Algorithms, Practicalities". Academic Press. New York. Apartados 4.3 y 4.4.