

Filtrado de Imágenes y Detección de Orillas Utilizando un Filtro Promediador Móvil Multipunto Unidimensional

Mario A. Bueno^a, Josué Álvarez-Borrego^b, Leonardo Acho^a y Vitaly Kober^c
mbueno@cicese.mx, josue@cicese.mx, leonardo@citedi.mx, vkober@cicese.mx

^aCITEDI-IPN¹, 2498 Roll Dr. 757, Otay Mesa
San Diego CA, USA

^bCICESE, División de Física Aplicada, Depto. Óptica

^cCICESE, División de Física Aplicada, Depto. de Ciencias de la Computación
Km. 107 Carretera Tijuana-Ensenada, B.C. México, CP 22860

Keywords: Area Técnica: Procesamiento Digital de Señales

RESUMEN

En el procesamiento de imágenes hay una variedad algoritmos de filtraje que se han descrito a través del tiempo en diferentes escritos [1][2]. En imágenes digitales es posible implementar técnicas de procesamiento digital como segmentación y detección de bordes [3]. Aún en la actualidad algoritmos eficientes sobre la utilización de máscaras especialmente en la detección de bordes presentan un considerable interés en el procesamiento de imágenes [4][5]. Este trabajo presenta la utilización de un filtro promediador móvil multipunto unidimensional (FPMMU), implementado como un filtro pasa-baja en imágenes diversas y de copéodos [6]. El objetivo es generar una señal discreta unidimensional usando una máscara de pixeles vecinos de 3x3, 5x5 y 7x7, después utilizando el FPMMU obtenemos la señal discreta filtrada la cual es restada de la imagen original produciendo la imagen final, mostrándose la detección de los bordes. Los resultados obtenidos son visualmente mejores que los obtenidos por otros métodos.

DESCRIPCIÓN DEL SISTEMA

Un filtro FPMMU puede ser usado como un filtro pasa-baja aplicado a una señal discreta unidimensional [7]:

$$y(n) = [x(n-1) + x(n) + x(n+1)] / c \quad n = 0, \pm 1, \pm 2, \dots \quad c = 3, 5, 7 \quad (1)$$

donde $x(n)$ es la entrada de la señal discreta, $y(n)$ es la salida de la señal discreta correspondiente al promedio del valor enésimo de la señal discreta de entrada $x(n)$, $x(n-1)$ es el valor previo del enésimo elemento y $x(n+1)$ el valor siguiente, c es el tamaño de la máscara de pixeles del filtro utilizado. Este promedio es calculado para diferentes elementos de $x(n)$ de acuerdo al tamaño n de la señal $x(n)$. Por lo tanto como estamos trabajando con un promedio de valores, este filtro se puede utilizar como un filtro pasa-bajas [7][8]. Por ejemplo, considere que $x(n)$ es obtenida mediante un muestreo continuo de la señal $x(t) = 5\cos(2\pi(10)t) + \cos(2\pi(100)t)$ con un muestreo a 0.003 segundos.

¹ Centro de Investigación del Instituto Politécnico Nacional localizado en Tijuana, B.C., México.

La señal muestreada correspondiente $x(n)$ se muestra en la Fig. 1a. La señal de salida $y(n)$ obtenida mediante la utilización de la ecuación (1) con una máscara de 3 píxeles ($c=3$) es mostrada en la Fig. 1b. Observe que $x(n)$ contiene algo de ruido el cual es atenuado en $y(n)$.

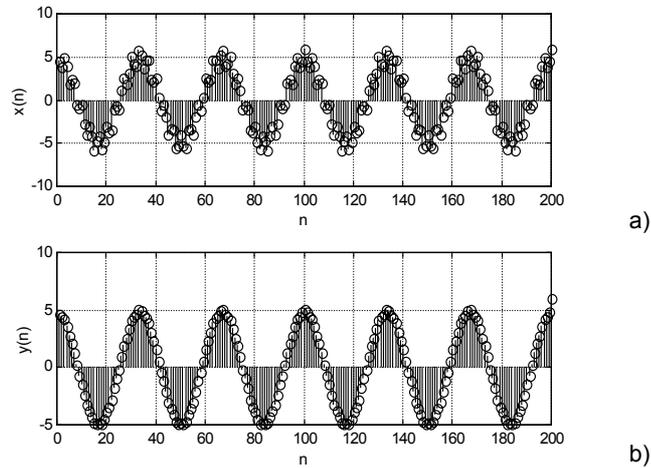


Fig. 1 - Resultados numéricos.

Se aplica por lo tanto ésta idea para crear un filtro pasa-baja a una imagen digital formada por una escala de grises, donde la señal discreta $x(n)$ es producida como sigue:

$$\begin{matrix}
 x(0) \\
 x(1) \\
 x(2) \\
 x(3) \\
 x(4) \\
 x(5) \\
 x(6) \\
 x(7) \\
 x(8)
 \end{matrix}
 =
 \begin{matrix}
 A(i-1, j-1) \\
 A(i-1, j) \\
 A(i-1, j+1) \\
 A(i, j+1) \\
 A(i+1, j+1) \\
 A(i+1, j) \\
 A(i+1, j-1) \\
 A(i, j-1) \\
 A(i, j)
 \end{matrix}
 \tag{2}$$

La señal discreta $x(n)$ es equivalente a la secuencia obtenida siguiendo la ruta marcada en la Fig. 2 alrededor de $A(i,j)$. Aquí, $A(i,j)$ representa el n ésimo elemento de la imagen de entrada $A(x,y)$ en el dominio espacial.

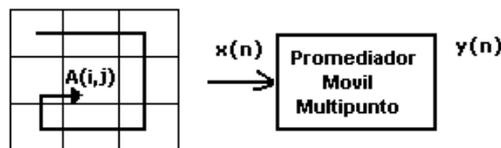


Fig. 2 – Descripción del sistema para un FPMMU con máscara de píxeles de 3x3.

Cuando la señal filtrada $y(n)$ es calculada por medio de la ecuación (1), obtenemos su promedio; por ejemplo para un filtro FPMMU con una máscara de pixeles de 3x3:

$$\bar{y} = \frac{1}{7} \sum_{n=1}^7 y(n) \quad (3)$$

Observe que la sumatoria va desde $n=1$ hasta $n=7$ satisfaciendo el límite de la secuencia de $x(n)$ en (2); en otras palabras, cuando $n=1$ en (1), usamos el primer valor del elemento de $x(n)$ en (2), $x(0)$, y cuando $n=7$ en (1), usamos el último valor del elemento de $x(n)$ en (2), $x(8)$. La imagen de salida $B(i,j)$ en la localización espacial (i,j) es obtenida mediante la siguiente ecuación:

$$B(i, j) = \bar{y} \quad (4)$$

Este proceso se repite hasta recorrer cada uno de los pixeles de toda la imagen de entrada $A(x,y)$ y obtener la imagen de salida $B(x,y)$. Después de realizar el proceso anterior, necesitamos corregir el rango dinámico de los pixeles de la imagen de salida mediante la ecuación siguiente:

$$B_T(i, j) = 255[B(i, j) - I(\min_{i,j}(B(i, j)))] / \max_{i,j}(B(i, j)) \quad (5)$$

donde se supone que la imagen está expresada en una escala de grises con 256 niveles de grises.

EXPERIMENTOS NUMÉRICOS

Aplicando el algoritmo anterior con una máscara de 3x3 pixeles a la imagen mostrada en la Fig. 3a, la imagen filtrada en las bajas frecuencias es obtenida en la Fig. 3b. El proceso de restar la imagen mostrada en la Fig. 3b de la imagen original mostrada en la Fig. 3a, produce una imagen conteniendo sólo las altas frecuencias mostrada en la Fig. 3c la cual es bastante similar a una imagen conteniendo sólo las orillas obtenida por algún otro método de detección de orillas.

Para comparar nuestros resultados, procesamos la imagen que se muestra en la Fig 3a mediante el algoritmo de Robert's [3] el cual nos detecta las orillas de una imagen dada, los resultados se muestran en la Fig. 3d. Note que algunas orillas no se muestran en la Fig. 3d, mientras que en el procesamiento de la imagen con nuestro algoritmo hay una detección mas robusta y precisa de las orillas, observe la Fig. 3c.

Para determinar la eficiencia de nuestro algoritmo extendimos las pruebas numéricas y procesamos una imagen de copépodo [6] con otros algoritmos diferenciadores detectores de orillas, tales como Prewitt[7], Sobel[7] y Robert's, los resultados obtenidos se muestran en la Fig. 4. En la Fig 4a se muestra la imagen procesada con el algoritmo de Prewitt, en la Fig. 4b se muestra la imagen procesada con el algoritmo de Sobel, en la Fig 4c se muestra la imagen procesada con el algoritmo de Robert's y en la Fig 4d se muestra la imagen procesada con el algoritmo FPMMU utilizado una máscara de 3x3 pixeles. Podemos extender la capacidad de filtrado del algoritmo cambiando la dimensión de la máscara de procesamiento, por ejemplo podemos utilizar máscaras de 3x3, 5x5, 7x7, 9x9 pixeles.

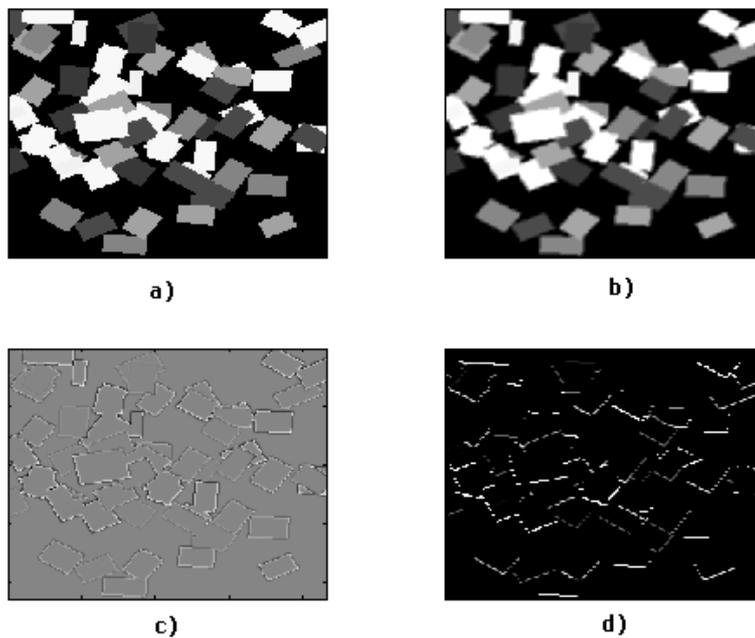


Fig. 3 – Experimentos numéricos.

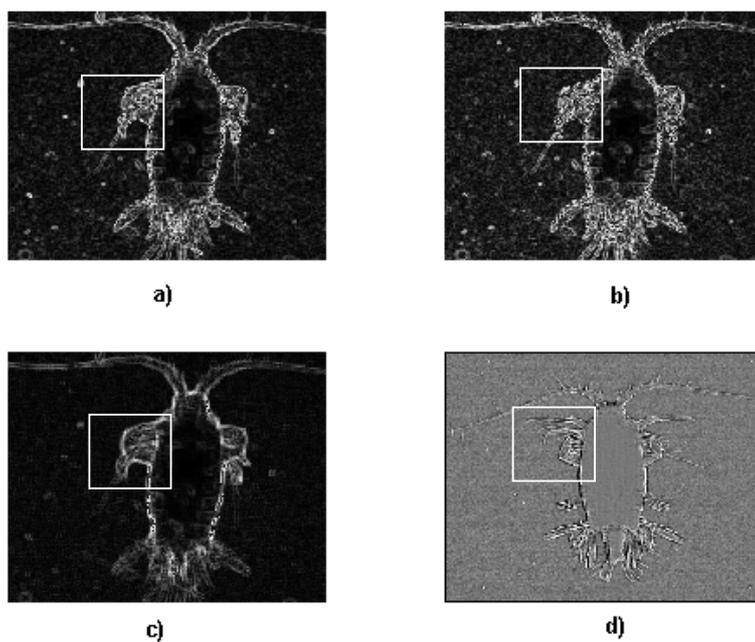


Fig. 4 – Experimentos numéricos adicionales con imágenes de copéodos.

En la Fig 4 se muestran las ventanas utilizadas para comparar la eficiencia de los diferentes algoritmos probados, esta ventana fue tomada de la imagen original, la cual es mostrada en la Fig 5, para tener una referencia visual de la misma y así observar los detalles finos de la imagen original y compararlos visualmente. En la Fig 6a tenemos los resultados de la ventana amplificada utilizando Prewitt, en la Fig 6b tenemos los resultados utilizando Sobel, en la Fig 6c tenemos los resultados utilizando Roberts y en la Fig 6d tenemos los resultados utilizando FPMMU. Comparando las imágenes que se muestran en la Fig 6, observamos que el algoritmo propuesto preserva los detalles finos mientras que los otros algoritmos pierden y distorsionan las orillas de la imagen original.

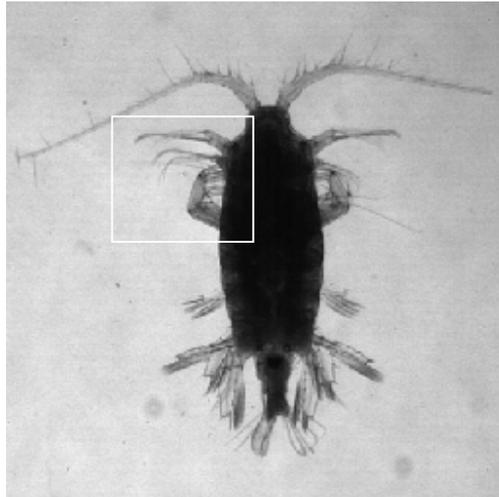


Fig. 5 – Imagen original del copépodo.

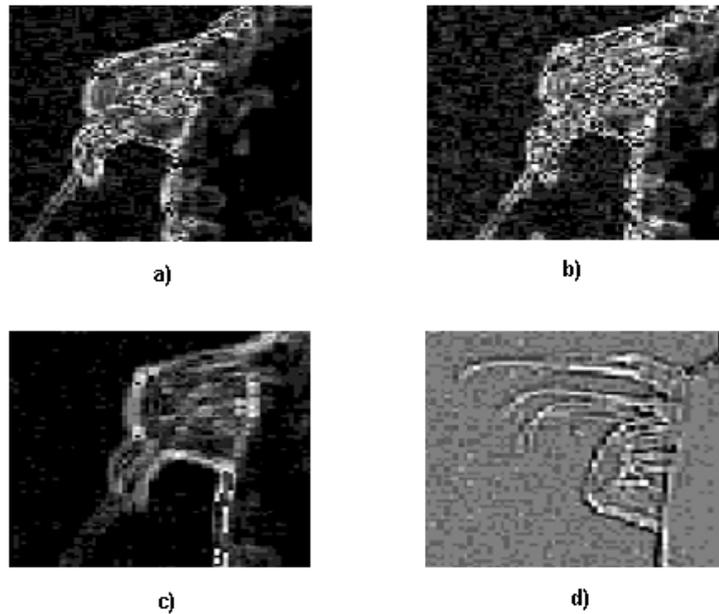


Fig. 6 – Preservación de los detalles .

CONCLUSIONES

El filtrado de imágenes FPMMU se extendió a señales bi-dimensionales discretas representadas por imágenes en escala de grises y se obtuvieron imágenes filtradas conteniendo bajas frecuencias. Se pudo realizar la detección de las orillas mediante la resta de las imágenes con bajas frecuencias de las imágenes originales. El algoritmo FPMMU descrito se comparó contra otros algoritmos mostrando resultados satisfactorios ya que preserva los detalles finos del organismo procesado.

Agradecimientos: Parte de este trabajo está soportado por el Consejo Nacional de Ciencia y Tecnología (CONACYT, proyecto 36075-B).

REFERENCIAS

1. Lim, Jae S. Two-Dimensional Signal and Image Processing. Englewood Cliffs, NJ: Prentice Hall, 1990. pp. 478-488.
2. Parker, James R. Algorithms for Image Processing and Computer Vision. New York: John Wiley & Sons, Inc., 1997. pp. 23-29.
3. Canny, John. "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986. Vol. PAMI-8, No. 6, pp. 679-698.
4. Ando S., IEEE Trans. Pattern Anal. Machine Intell., **22**, no. 3, 252-265. (2000)
5. Heath et al., Computer Vision and Image Understanding. **69**, no. 1 , 38-54. (1997-1998)
6. J. Álvarez-Borrego. et al., Journal of Plankton Research. Short Comm. **25**, 229-233, (2003)
7. González R. and Woods R. *Digital Image Processing 2nd Edition*, Prentice Hall, USA. 2002. pp. 119-123,131.
8. Proakis J. G. and Manolakis D. G., *Digital Signal Processing*, Prentice Hall, Third Edition, Upper Saddle River, New Jersey, USA. pp. 331-333.