Computación y Sistemas Vol. 6 No. 3 pp. 183 - 195
© 2003, CIC - IPN. *ISSN 1405-5546* Impreso en México

Filtros Robustos RM-KNN con Diferentes Funciones de Influencia para Supresión de Ruido Impulsivo en Imágenes Digitales

Robust RM-KNN Filters with Different Influence Functions for Removal of Impulsive Noise in Digital Images

Francisco Gallegos[1], Volodymyr Ponomaryov[2], Oleksiy Pogrebnyak[3] y Luis Niño de Rivera[2]
[1]Escuela Superior de Ingeniería Mecánica y Eléctrica, U.P.A.L.M, Zacatenco
Av. IPN s/n, Col. Lindavista, C.P. 07738, México, D.F.
[2]Escuela Superior de Ingeniería Mecánica y Eléctrica, U.P. Culhuacán del IPN
Av. Sta. Ana 1000, Col. San Francisco Culhuacán, C.P. 04430, México, D.F.
[3]Centro de Investigación en Computación del IPN
Av. Juan de Dios Bátiz s/n Col. Lindavista C.P. 07738, México D.F.
E-mails: fcogf@hotmail.com, vponomar@ipn.mx, olek@cic.ipn.mx

Artículo recibido en Diciembre 12, 2002; aceptado en Marzo 15, 2003

Resumen

Presentamos la implementación de filtros robustos para imágenes con supresión de ruido impulsivo y preservación de detalles. Los esquemas de filtrado usan una técnica similar al filtro KNN para proveer la preservación de detalles finos y la combinación de estimadores-M con el estimador de la mediana o Wilcoxon proveen la supresión de ruido impulsivo. Usamos diferentes tipos de funciones de influencia en el estimador-M para proveer una mejor supresión de ruido impulsivo. El filtrado de secuencias de vídeo corrompidas con ruido impulsivo demuestra que los métodos propuestos potencialmente proveen una solución para mejorar la calidad de las transmisiones de TV/Vídeo. La eficiencia de los filtros propuestos fue evaluada por numerosas simulaciones. La implementación de los filtros propuestos fue realizada en tiempo real mediante el uso del DSP TMS320C6701.

Palabras Clave: Filtros de orden estadístico, filtros RM-KNN, DSP TMS320C6701.

Abstract

We present the implementation of robust image filtering for impulsive noive suppression with detail preservation. The filtering schemes use a similar to KNN filter technique to provide fine detail preservation and the redescending M-estimators combined with the median or Wilcoxon estimator to provide impulsive noise rejection. We use different types of influence functions in the M-estimator to provide better impulsive noise suppression. The filtration of image sequences corrupted by impulsive noise demonstrates that the proposed methods potentially could provide a solution to quality TV/Video transmission. The efficiency of the proposed filters has been evaluated by numerous simulations. The implementation of proposed filters was realized in real-time by means of use of DSP TMS320C6701.

Keywords: Order Statistics Filters, RM-KNN filters, DSP TMS320C6701.

1 Introducción

Diferentes clases de filtros han sido propuestas para remover el ruido en las imágenes (Pitas y Venetsanopoulos, 1990; Astola y Kuosmanen, 1997). Los filtros son clasificados en diferentes categorías dependiendo de aplicaciones específicas. Los filtros lineales son eficientes para remover el ruido Gaussiano pero tienen un pobre desempeño ante el ruido impulsivo. Los filtros no lineales son diseñados para suprimir ruido de diferente naturaleza (Gaussiano, multiplicativo, etc.), pueden remover el ruido impulsivo y pueden preservar los contornos y detalles finos (Pitas y Venetsanopoulos, 1990; Astola y Kuosmanen, 1997). Muchos tipos de filtros no lineales están basados en la teoría de estimación robusta, especialmente en la de orden estadístico (Pitas y Venetsanopoulos, 1990; Astola y Kuosmanen, 1997).

Sin embargo, la primera teoría de estimación robusta (estimadores-*M*) fue realizada por Huber (Huber, 1981). Por otra parte, los estimadores-*R* robustos fueron desarrollados por Hodges y Lehmann (Hodges y Lehmann, 1963). Estos estimadores fueron originalmente derivados de la teoría de estadísticas de rangos. La mediana y la combinación lineal de orden estadístico (estimadores-*L*) fueron propuestas y usadas como estimadores robustos por Tukey, David y otros (Pitas y Venetsanopoulos, 1990; Hampel *et al.*, 1986). En la actualidad, un gran campo dentro del orden estadístico ha sido desarrollado para la estimación robusta.

El análisis hecho para el desempeño de varios tipos de estimadores robustos esta basado en las funciones de influencia. Estas funciones son herramientas muy poderosas

para la descripción de robustez, y para el diseño de estimadores robustos (Astola y Kuosmanen, 1997; Hampel *et al.*, 1986).

Los filtros basados en la estadística de rangos han recibido considerable atención debido a sus excelentes propiedades de supresión de ruido y preservación de detalles. Ejemplos de filtros basados en el orden de rangos son los bien conocidos *mediano*, KNN (*K-nearest neighbor*), filtro *L*, diferentes filtros de switcheo, filtros neuronales, etc. Los filtros *mediano multietapa*, CWM (*center weighted median*), WM (*weighted median*), WOS (*weighted order statistics*), *stack*, *permutación* y RCRS (*rank conditioned rank selection*) han sido utilizados como filtros suavizantes en aplicaciones de restauración donde una imagen es corrompida por ruido. Los filtros CS (*comparison and selection*), LUM (*lower-upper-middle*) y WMMR (*weighted majority of samples with minimum range*) han sido propuestos para mejoramiento de contornos (Astola y Kuosmanen, 1997; Bovik, 2000).

Por otro lado, también hay que considerar que los dispositivos para procesamiento de señales en tiempo real son muy importantes en un gran número de aplicaciones (Bovik, 2000; Kotropoulos y Pitas, 2001). En diferentes aplicaciones de procesamiento de imágenes como son los casos del análisis de imágenes médicas (ultrasonido, tomografía, etc.), radar, percepción remota, etc., es importante que los resultados sean en tiempo real para un análisis preciso para cada una de las aplicaciones en cuestión (Bovik, 2000; Kotropoulos y Pitas, 2001).

En sistemas de visión por computadora, los robots inteligentes responden a cambios ocurridos en su sensór de visión CCD (*Charge Coupled Device*) cuando ocurre una reacción en su entrada de visión. Por esta razón, las imágenes digitales adquiridas en sistemas de visión de robots deberán tener buena resolución y estar libres de ruido. Otro caso en el cual es importante la realización en tiempo real es en el procesamiento de vídeo, su transmisión y recepción, en las cuales las secuencias de vídeo deberán ser procesadas en tiempo real mediante algún algoritmo que restaure sus características, por ejemplo, la restauración o supresión de ruido de las secuencias ocasionada durante transmisiones de TV/vídeo (Bovik, 2000; Kotropoulos y Pitas, 2001).

En este artículo presentamos algoritmos robustos tipo RM para el filtrado de imágenes que proveen una buena preservación de los detalles finos con una buena supresión de ruido impulsivo (Ponomaryov y Pogrebnyak, 1999). El filtro diseñado usa el valor del pixel central de la ventana de filtrado para proveer la preservación de detalles finos y la combinación de estimadores-*M* con el estimador de la *mediana* o Wilcoxon proveen la supresión de ruido impulsivo. El estimador RM usa una función redescendiente similar a la del filtro KNN. Los algoritmos resultantes usan diferentes funciones de influencia en los estimadores robustos RM, tales como las funciones *simple de corte, corte mediana, Hampel, seno de Andrew, Tukey y Bernoulli* (Ponomaryov y Pogrebnyak, 1999; Gallegos-Funes, *et al.*, 2001; Hampel *et al.*, 1986).

Resultados experimentales del filtrado de diferentes imágenes y señales de vídeo son presentados. Extensivas simulaciones demostraron que el filtro propuesto provee mejor supresión de ruido y preservación de detalles que otros filtros basados en la estadística de orden. El criterio usado para evaluar el rendimiento de supresión de ruido fue la *relación pico-señal-a-ruido* (PSNR) y para evaluar la preservación de detalles finos el criterio usado fue el *error absoluto medio* (MAE) (Astola y Kuosmanen, 1997; Bovik, 2000; Kotropoulos y Pitas, 2001; Mitra y Sicuranza, 2001). Las propiedades determinísticas y estadísticas de los filtros diseñados son discutidas. También, presentamos la implementación del filtro propuesto mediante el uso del DSP TMS320C6701 de punto flotante que garantiza una realización en tiempo real del algoritmo en aplicaciones de procesamiento de imágenes y vídeo. Resultados experimentales del tiempo necesario para realizar los algoritmos propuestos son presentados en este artículo.

2 Modelo de Degradación para Imágenes

Existen diferentes modelos de ruido (Astola y Kuosmanen, 1997; Bovik, 2000; Kotropoulos y Pitas, 2001; Mitra y Sicuranza, 2001). Nosotros usamos el siguiente modelo de ruido para el caso de ruido impulsivo.

$$x(i,j) = n_{im}(e(i,j)) \ , \qquad (1)$$

donde $e(i,j)$ es la imagen original, $x(i,j)$ es la imagen degradada y $n_{im}(e(i,j))$ es la función $n_{im}(e\ (i,j)) = \begin{cases} M(i,j) & \text{con probabilid ad } p \\ e(i,j) & \text{con probabilid ad 1-}p \end{cases}$, donde $M(i,j)$ son valores aleatorios distribuidos uniformemente en un rango de 0 a 255.

Si tomamos en cuenta la ecuación (1) la solución para decrementar la influencia de ruido consiste en diseñar algoritmos de filtrado con propiedades robustas que puedan remover el ruido impulsivo y puedan preservar los contornos y detalles finos. Para este propósito, proponemos el uso de un filtro de tipo no lineal que use estimadores robustos en su ventana de filtrado.

3 Filtros Tipo RM

En esta sección se describen los estimadores-*R* así como también los estimadores-*M* y su solución mediante el uso de diferentes funciones de influencia. Los estimadores robustos *R* y *M* sirvieron como base para el desarrollo de los estimadores combinados RM, estos nuevos estimadores son presentados aquí y sirvieron como base para el desarrollo de los filtros RM.

3.1 Estimadores-R

Los estimadores-*R* son una clase de estimadores robustos no paramétricos basados en cálculos de rango. Considere dos muestras de test de rangos $x_1,\ldots,x_m$ y $y_1,\ldots,y_n$ como dos muestras con distribuciones $H(x)$ y $H(x+\Delta)$, donde Δ es el corrimiento de localización desconocido. Sea R_i el rango de X_i en la muestra de tamaño $N=m+n$. Un test de rangos de $\Delta=0$ para $\Delta>0$ esta basado en el test estadístico (Hampel *et al.*, 1986)

$$S=\frac{1}{m}\sum_{i=1}^{m}a_i(R_i) \qquad (2)$$

Usualmente asumimos que los coeficientes o pesos a_i son generados por alguna función J de la siguiente manera:

$$a_i=(m+n)\int_{(i-1)/(m+n)}^{i/(m+n)}J(s)ds \qquad (3)$$

La función $J(s)$ es simétrica en el censo de $J(1-s)=-J(s)$ y satisface $\int J(s)ds=0$ y los coeficientes a_i satisfacen $\sum_{i=1}^{n}a_i=0$. Para la selección de la función $J(s)$ asumimos que la función de prueba $f_0(X)$ es simétrica y realizando (Hampel *et al.*, 1986)

$$J(f_0(X))=-c\frac{f_0'(X)}{f_0(X)}, \quad c\neq 0 \qquad (4)$$

y para cuando $f_0(X)$ es asimétrica no se tendrá eficiencia completa en los estimadores-*R* encontrados.

De acuerdo con lo anterior, el estimador mediano es óptimo en el caso de una función de distribución Laplaciana $f_0(X)=\frac{1}{2}e^{-|x|}$ y con una función $J(t)=\begin{cases}-1 & t<\frac{1}{2}\\ 1 & t>\frac{1}{2}\end{cases}$ dada por (4) y coeficientes $a_i=\begin{cases}1 & i=(N+1)/2\\ 0 & \text{cualquier otro caso}\end{cases}$ dados por (3) el estimador mediano se deriva mediante la estima de Hodghes-Lehmann mediante el uso de los coeficientes a_i en la eq. (2) de tal forma que:

$$\hat{\theta}_{\text{med}}=\begin{cases}\frac{1}{2}\left(X_{(n/2)}+X_{(1+n/2)}\right) & \text{para } n \text{ par}\\ X_{(1+n/2)} & \text{para } n \text{ impar}\end{cases} \qquad (5)$$

donde $X_{(j)}$ es el elemento de rango j. Este es el mejor estimador cuando ninguna información *a priori* de la distribución de datos de forma y momentos de X_i no esta disponible (Astola y Kuosmanen, 1997; Hampel *et al.*, 1986).

De la misma forma, el estimador Hodges-Lehmann con una función $J(t)=\left|t-\frac{1}{2}\right|$ esta relacionado con el test de Wilcoxon y una función de distribución logística $f_0(x)=\frac{1}{1+e^{-x}}$ con coeficientes dados por $a_i=\frac{2i-N-1}{2N}$ y el correspondiente estimador de rango es el estimador-*R* Wilcoxon:

$$\hat{\theta}_{\text{Wil}}=\underset{i\leq j}{\text{med}}\left\{\frac{1}{2}\left(X_{(i)}+X_{(j)}\right),\ i,j=1,\ldots N\right\} \qquad (6)$$

donde $X_{(i)}$, $X_{(j)}$ son los elementos con rango i y j respectivamente. La ecuación (6) es robusta y es la mejor estimación cuando la distribución de datos inicial tiene forma simétrica, por lo que este test es conocido como el test asintóticamente más poderoso (Astola y Kuosmanen, 1997; Hampel *et al.*, 1986).

3.2 Estimadores-M

La forma generalizada del máximo de verosimilitud (maximum likelihood) para los estimadores robustos *M* fue propuesta por Huber (Hampel *et al.*, 1986; Pitas y Venetsanopoulos, 1990; Astola y Kuosmanen, 1997). Su definición es dada por una función $\rho(X)=\ln(f(X))$ conectada con la función de densidad de probabilidad $f(X)$ para una muestra de datos X_i, $i=1,\ldots,N$:

$$\hat{\theta}=\underset{\theta\in\Theta}{\arg\min}\sum_{i=1}^{N}\rho(X_i-\theta) \qquad (7)$$

La estimación para el parámetro de localización θ puede ser encontrado mediante el cálculo de la derivada parcial de ρ (con respecto a θ) introduciendo la función:

$$\psi(X,\theta)=\frac{\partial}{\partial\theta}\rho(X,\theta):$$

$$\sum_{i=1}^{n}\psi(X_i-\theta)=0\ , \qquad (8)$$

La solución de θ para el estimador *M* tiene algunas limitantes: la desviación de la función $\psi(X)$ llamada *winsorización* o de la desviación de $(X_i - \theta)$ llamada *censorización o trimming*. La técnica usada para el cálculo de la estimación *M* esta basada en el método iterativo de Newton (Astola y Kuosmanen, 1997):

$$\theta_{\mathbf{M}} = \frac{\sum_{i=1}^{N} X_i \tilde{\psi}(X_i - \mathrm{MED}\{\vec{X}\})}{\sum_{i=1}^{N} 1_{[-r,r]}(X_i - \mathrm{MED}\{\vec{X}\})} \quad , \qquad (9)$$

donde $\tilde{\psi}$ es la función normalizada ψ : $\psi(X) = X\tilde{\psi}(X)$. Es evidente que la fórmula (9) representa la media aritmética de $\sum_{i=1}^{n} \psi(X_i - \mathrm{MED}\{\vec{\mathbf{X}}\})$ evaluada en el intervalo $[-r, r]$.

La restricción más simple sobre el rango de $\psi(X)$ es el límite del estimador-*M* de Huber (Ponomaryov y Pogrebnyak, 1999)

$$\tilde{\psi}_b(X) = \mathrm{MIN}(b, \mathrm{MAX}(X, -b)) = [X]_{-b}^{b} . \qquad (10)$$

Otra vía para derivar la función $\tilde{\psi}(X)$ consiste en eliminar los valores lejanos (por peso) de una muestra de datos. Este paso es también llamado mínima estimación-*M*. Hampel (1986) demostró que la función de influencia de corte mediana (*skipped median*) es la mínima estimación robusta *M*. Anteriormente, también usamos la función simple de corte. Hay otras funciones de influencia conocidas en la literatura. También usamos las funciones de influencia de *Hampel, seno de Andrews, Tukey y Bernoulli* (Hampel *et al.*, 1986; Astola y Kuosmanen, 1997).

3.3 Estimadores-RM

Para mejorar las propiedades robustas de los estimadores *R* y *M* mediante el uso de las estimaciones *R* (5) y (6) consiste en un procedimiento similar a la media aritmética (Ponomaryov y Pogrebnyak, 1999; Gallegos-Funes, *et al.*, 2001). De este modo, obtenemos las estimaciones combinadas RM siguientes:

$$\theta_{\mathrm{medM}} = \mathrm{med}\{X_i \tilde{\psi}(X_i - \mathrm{med}\{\vec{\mathbf{X}}\})\} \qquad (11)$$

$$\theta_{\mathrm{WilM}} = \underset{i \le j}{\mathrm{med}}\left\{\frac{1}{2}\left[X_i \tilde{\psi}(X_i - \mathrm{med}\{\vec{\mathbf{X}}\}) + X_j \tilde{\psi}(X_j - \mathrm{med}\{\vec{\mathbf{X}}\})\right]\right\} \qquad (12)$$

Hay que hacer notar que el estimador RM (11) es usualmente la mediana cuando la función ψ es representada por (10). Si la función ψ es descrita por la función simple de corte, esta se transforma en la función de corte mediana. Es posible suponer que las propiedades robustas de los estimadores RM pueden exceder las propiedades robustas de los estimadores básicos *R* y *M*.

3.4 Filtros RM-KNN

Para incrementar la robustez de los filtros estándar, es posible emplear diferentes métodos conocidos en la teoría de estimación robusta, por ejemplo, la *censorización* (Pitas y Venetsanopoulos, 1990; Astola y Kuosmanen, 1997). Una propuesta conocida para incrementar la calidad de filtración vía preservación de contornos y detalles de pequeña escala en las imágenes consiste en el uso de K elementos de una muestra de datos, los cuales son encerrados con el valor del pixel central de la ventana de filtrado. De tal manera, se encuentra el conocido algoritmo para filtrado de imágenes KNN (*K-nearest neighbor*) (Astola y Kuosmanen, 1997):

$$\hat{e}_{\mathrm{KNN}}(i, j) = \frac{1}{K} \sum_{m=-L}^{L} \sum_{n=-L}^{L} \psi(x(i+m, j+n)) x(i+m, j+n) \qquad (13)$$

donde *m, n* = -*L*...*L* y $\psi(x(i+m, j+n)) =$

$$\begin{cases} 1, \text{ si } x(i+m, j+n) \text{ es uno de los } K \text{ valores encerrados} \\ \quad \text{en la ventana de filtrado con respecto a } x(i, j) \\ 0, \text{ cualquier otro caso} \end{cases}$$

Para mejorar la robustez del filtro KNN proponemos el uso de los estimadores combinados RM (11) y (12).

El filtro *mediano tipo-M KNN* (filtro MM-KNN) es determinado de la siguiente manera (Ponomaryov y Pogrebnyak, 1999; Gallegos-Funes *et al.*, 2001):

$$\hat{e}^{(q)}_{\mathrm{MMKNN}}(i, j) = \mathrm{med}\{g^{(q)}(i+m, j+n)\} \qquad (14)$$

donde $g^{(q)}(i+m, j+n)$ es un conjunto de K_c pixeles con peso de acuerdo a la función $\psi(X)$ usada y encerrados por valor en la ventana del filtro con la estimación del paso previo $\hat{e}^{(q-1)}_{\mathrm{MMKNN}}(i, j)$. El estimador inicial es $\hat{e}^{(0)}_{\mathrm{MMKNN}}(i, j) = x(i, j)$ y $\hat{e}^{(q)}_{\mathrm{MMKNN}}(i, j)$ denota la estimación en la iteración *q*. El tamaño de la ventana de filtrado es $N = (2L+1)^2$ y $m, n = -L, \ldots, L$. El algoritmo finaliza cuando $\hat{e}^{(q)}_{\mathrm{MMKNN}}(i, j) = \hat{e}^{(q-1)}_{\mathrm{MMKNN}}(i, j)$. Usualmente es necesario 3 ó 4 iteraciones *q* para satisfacer esta condición.

El filtro *Wilcoxon tipo-M KNN* (WM-KNN) es dado por:

$$\hat{e}^{(q)}_{\mathrm{WMKNN}}(i, j) = \mathrm{med}\left\{\frac{g^{(q)}(i+m, j+n) + g^{(q)}(i+m_1, j+n_1)}{2}\right\} \qquad (15)$$

donde $g^{(q)}(i+m, j+n)$ y $g^{(q)}(i+m_1, j+n_1)$ son pixeles que forman un conjunto de K_c pixeles con peso de acuerdo a la función $\psi(X)$ usada y encerrados por valor en la ventana del filtro con la estimación del paso previo $\hat{e}^{(q-1)}_{\mathrm{WMKNN}}(i, j)$. El estimador inicial es $\hat{e}^{(0)}_{\mathrm{WMKNN}}(i, j) = x(i, j)$ y $\hat{e}^{(q)}_{\mathrm{WMKNN}}(i, j)$ denota la estimación en la iteración *q*. El tamaño de la ventana de filtrado es $N = (2L+1)^2$, $m \le m_1$, $n \le n_1$, y $m, n, m_1, n_1 = -L, \ldots, L$. El algoritmo finaliza cuando

$\hat{e}^{(q)}_{\mathrm{WMKNN}}(i,j)=\hat{e}^{(q-1)}_{\mathrm{WMKNN}}(i,j)$. Usualmente es necesario 2 ó 3 pasos para satisfacer esta condición.

En ambos filtros $x(i,j)$ es el pixel actual o el pixel central contaminado por ruido en la ventana de filtrado y K_c es el número actual de pixeles vecinos cercanos, éste refleja la actividad local de los datos y la presencia de picos y es determinado por (Ponomaryov y Pogrebnyak, 1999; Gallegos-Funes *et al.*, 2001):

$$K_c(i,j)=\left[K_{\min}+a\,\mathrm{S}(x(i,j))\right]\le K_{\max} \qquad (16)$$

$$\mathrm{S}(x(i,j))=\frac{\mathrm{med}\{|x(i,j)-x(i+m,j+n)|\}}{\mathrm{MAD}\{x(i,j)\}}+0.5\frac{\mathrm{MAD}\{x(i,j)\}}{\mathrm{med}\{x(i+k,j+l)\}}, \qquad (17)$$

donde a controla la sensitividad del filtro para una buena detección de detalles; K_{min} es el número mínimo de vecinos para remover el ruido; K_{max} es el número máximo de vecinos que son usados para una buena detección de los contornos y los detalles suaves; $\mathrm{S}(x(i,j))$ es el detector de impulsos; y MAD (*the median of absolute deviations from median*) esta definida como

$$\mathrm{MAD}\{x(i,j)\}=\mathrm{med}\{|\mathrm{med}\{x(i+k,j+l)\}-x(i+m,j+n)|\}.$$

En nuestros experimentos una ventana de filtrado 3x3 fue usada. El número de vecinos ceercanos $K_{min}=5$ fue encontrado de numerosas simulaciones en diferentes imágenes con diferentes niveles de ruido y el valor de $K_{max}=(N-1)$ depende de la ventana de filtrado sin contar el pixel central. Para mejorar el rendimiento de los filtros propuestos (14) y (15) en presencia de ruido impulsivo proponemos el uso del filtro mediano estándar. De numerosas simulaciones obtuvimos que cuando $K_c>7$ los filtros MM-KNN y WM-KNN pueden ser sustituidos por el filtro *mediano 3x3* y cuando $K_c>350$ pueden ser sustituidos por el filtro *mediano 5x5*. Esto es, cuando $K_c>7$, no se necesita el filtrado fino de técnica KNN y por ello, podemos emplear el filtro *mediano 3x3*. Este efecto se puede explicar considerando que los cálculos de la estimación KNN de 7 pixeles de entre el número total de 9 pixeles produce resultados similares al filtro *mediano 3x3*. Además, cuando el estimador de la actividad local de los datos produce un valor de K_c más grande que el número de píxeles en la ventana *3x3*, en este caso mayor que 350, significa que podemos ampliar la ventana del filtro mediano hasta *5x5*.

Finalmente, el diagrama de flujo del algoritmo del filtro RM-KNN (filtros MM-KNN y WM-KNN) de acuerdo a las ecuaciones (14) a (17) se muestra en la Figura 1.

3.5 Funciones de Influencia

Las funciones de influencia utilizadas como funciones de restricción en el filtro (14) y (15) para derivar los estimadores robustos M aplicables para el procesamiento de imágenes se muestran en la Tabla 1. Estas funciones proveen buena supresión de ruido impulsivo (Pitas y Venetsanopoulos, 1990; Astola y Kuosmanen, 1997; Hampel *et al.*, 1986).

4 Rendimiento del Filtro Propuesto

De diferentes pruebas de simulación, se obtuvieron las propiedades de los nuevos algoritmos de filtrado (14) y (15), y comparamos estas con diferentes filtros propuestos en la literatura. El criterio usado para comparar el rendimiento de supresión de ruido de los filtros fue la *relación pico de señal a ruido* (PSNR) y para evaluar la preservación de contornos y detalles finos el criterio usado fue el *error absoluto medio* (MAE).

$$\mathrm{PSNR}=10\cdot\log\left[\frac{(255)^2}{\mathrm{MSE}}\right]\ \mathrm{dB}\ , \qquad (18)$$

$$\mathrm{MAE}=\frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}|x(i,j)-\hat{e}(i,j)| \qquad (19)$$

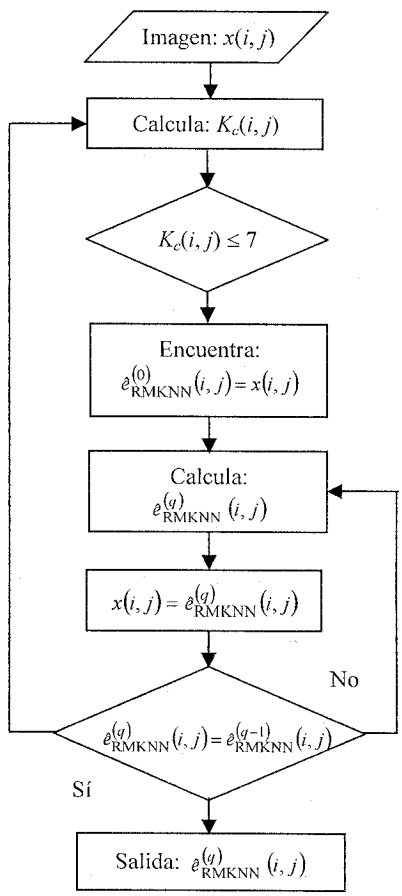


Figura 1: Diagrama de flujo del algoritmo RM-KNN.

Función de Influencia	Fórmula
Simple de corte	$\psi_{\mathrm{cut}(r)}(X)=\begin{cases} X, \lvert X\rvert \le r \\ 0, \lvert X\rvert > r \end{cases}$
Corte mediana	$\psi_{\mathrm{med}(r)}(X)=\begin{cases} \mathrm{sgn}(X), \lvert X\rvert \le r \\ 0, X > r \end{cases}$
Hampel	$\psi_{\alpha,\beta,r}(X)=\begin{cases} X, \ 0 \le \lvert X\rvert \le \alpha \\ \alpha \cdot \mathrm{sgn}(X), \ \alpha \le \lvert X\rvert \le \beta \\ \alpha \dfrac{r-\lvert X\rvert}{r-\beta}, \ \beta \le \lvert X\rvert \le r \\ 0, \ r \le \lvert X\rvert \end{cases}$
Seno de Andrews	$\psi_{\sin(r)}(X)=\begin{cases} \sin(X/r), \lvert X\rvert \le r\pi \\ 0, \lvert X\rvert > r\pi \end{cases}$
Tukey	$\psi_{\mathrm{bi}(r)}(X)=\begin{cases} X^2\left(r^2-X^2\right), \lvert X\rvert \le r \\ 0, \lvert X\rvert > r \end{cases}$
Bernoulli	$\psi_{\mathrm{ber}(r)}(X)=X^2\sqrt{r^2-X^2}\cdot 1_{[-r,r]}(X)$

Tabla 1: Funciones de influencia usadas en el filtrado propuesto para derivar los estimadores robustos *M*.

donde $\mathrm{MSE}=\frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left[x(i,j)-\hat{e}(i,j)\right]^2$ es el *error cuadrático medio*, $x(i, j)$ es la imagen original; $\hat{e}(i, j)$ es la imagen restaurada; y *M, N* es el tamaño de la imagen.

Los rendimientos de los filtros MM-KNN y WM-KNN con las funciones de influencia *simple de corte*, *corte mediana*, *Hampel*, *seno de Andrews*, *Tukey* y *Bernoulli* fueron comparados con los filtros *mediano*, *weighted median* (WM) (Brownrigg, 1984; Yli-Harja, *et al.*, 1991), *FIR median hybrid* (FIRMH) (Wichman, *et al.*, 1990), *lower-upper-middle* (LUM) (Hardie y Boncelet, 1993), *tri-state median* (TSM) (Chen, *et al.*, 1999), *adaptive center weighted median* (ACWM) (Chen, *et al.*, 2001), *rank order mean* (ROM) (Abreu, *et al.*, 1996), *minimum-maximum exclusive mean* (MMEM) (Wei-Yu y Ja-Chen, 1997) y *local linear minimum mean square error* (LMMSE) (Ôzkan, *et al.*, 1993).

Para determinar las propiedades de supresión de ruido impulsivo de los filtros propuestos (14) y (15), la imagen de prueba "Lena" en niveles de gris de 256x256 pixeles fue degradada con 5%, 10% y 15% de ruido impulsivo. Los valores PSNR y MAE para diferentes filtros se muestran en la Tabla 2. Los valores en "negrita" representan los mejores valores en PSNR y MAE para los filtros propuestos y para los filtros de comparativo. La Tabla 2 muestra que los filtros propuestos MM-KNN y WM-KNN tienen una mejor supresión de ruido y preservación de detalles en comparación a los filtros de comparativo, no importando los niveles de ruido impulsivo.

Otra prueba fue desarrollada en las imágenes "Airfield", "Barbara", "Boat", "Goldhill", "Lena" y "Peppers" en niveles de gris de 256x256 pixeles las cuales fueron degradadas con 20% de ruido impulsivo. Los valores PSNR y MAE para estos filtros se muestran en la Tabla 3. En "negrita" se representan los mejores valores para los filtros propuestos y para los filtros de comparativo.

De la Tabla 3 podemos observar que los filtros propuestos proveen mejor supresión de ruido impulsivo y preservación de detalles que los filtros utilizados como comparativo. Los filtros MM-KNN y WM-KNN proveen resultados similares pero el filtro MM-KNN provee una mejor supresión de ruido y preservación de detalles.

Algoritmo	Porcentaje de ruido impulsivo					
	5%		10%		15%	
	PSNR	MAE	PSNR	MAE	PSNR	MAE
Mediano 3x3	24.99	8.63	24.27	9.54	23.64	10.16
WM 3x3	25.56	6.90	24.02	7.48	24.41	**8.08**
FIRMH 5x5	27.69	6.12	25.63	7.51	23.79	9.03
LUM 5x5	28.02	6.56	26.17	7.84	24.49	9.04
TSM 3x3	26.90	7.38	24.12	9.50	23.98	10.35
ACWM 3x3	27.73	7.35	26.83	8.00	**26.09**	8.45
ROM 3x3	27.49	7.64	26.51	8.34	25.76	8.79
MMEM 3x3	**28.24**	**5.82**	**27.04**	**6.77**	25.67	**8.08**
LMMSE 3x3	28.00	7.43	26.33	8.44	25.41	9.19
MM-KNN (Simple)	26.77	7.45	25.55	8.20	25.29	8.20
MM-KNN (Corte)	28.46	5.72	27.68	6.22	26.89	6.74
MM-KNN (Hampel)	**28.55**	**5.64**	**27.72**	**6.16**	**26.96**	**6.68**
MM-KNN (Andrew)	28.48	5.71	27.70	6.21	26.93	6.72
MM-KNN (Tukey)	28.39	5.75	27.60	6.25	26.87	6.77
MM-KNN Bernoulli	28.40	5.75	27.60	6.25	26.87	6.77
WM-KNN (Simple)	28.10	5.84	27.21	**6.49**	**26.31**	**7.08**
WM-KNN (Corte)	28.08	5.84	27.19	**6.49**	26.30	7.09
WM-KNN (Hampel)	27.70	6.22	**27.48**	6.74	25.90	7.22
WM-KNN (Andrew)	**28.13**	**5.83**	27.16	6.51	26.29	7.11
WM-KNN (Tukey)	27.58	6.65	26.75	7.20	25.15	7.76
WM-KNN Bernoulli	27.33	6.44	26.68	7.11	25.15	7.65

Tabla 2: Resultados comparativos PSNR en dB y MAE en la imagen "Lena" corrompida por 5%, 10% y 15% de ruido impulsivo.

El parámetro *a* de los filtros propuestos y los parámetros α, β y *r* de las funciones de influencia son muy importantes para obtener la suficiente supresión de ruido y preservación de detalles. Los valores para estos parámetros se muestran en la Tabla 4 y fueron encontrados haciendo variar éstos en numerosas simulaciones con diferentes imágenes y diferentes niveles de ruido.

Para las imágenes de prueba de las Tablas 2 y 3, encontramos que los valores del parámetro *a* (16) cambian en el caso de la función *simple* para diferentes imágenes y diferentes niveles de ruido, para el filtro MM-KNN $a=2$ y el filtro WM-KNN $a=3$. Para otras funciones de influencia usadas el parámetro *a* puede ser tomado como constante con un valor óptimo de $a = 4$ o $a = 5$ para el filtro MM-KNN y para el filtro WM-KNN $a=3$. Los valores de los parámetros de las funciones de influencia son: $r=90$ para la función de *corte*, $\alpha=10$, $\beta\le 90$ y $r=300$ para la función de *Hampel*. El parámetro *r* para la función del *seno de Andrew* es $r=35$, para la función de *Tukey* $r=15$ y para la función de *Bernoulli* $r=20$. En la Figura 2 se presentan las imágenes procesadas para la imágen "Peppers" mostrando la supresión de ruido impulsivo de acuerdo a la Tabla 3. En la Figura 3 se presentan las imágenes de error obtenidas a partir de los resultados de la Figura 2.

Algoritmo	Resultados comparativos PSNR en dB y MAE de diferentes imágenes corrompidas por 20% de ruido impulsivo											
	Airfield		Barbara		Boat		Goldhill		Lena		Peppers	
	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE
Mediano 3x3	21.24	11.43	21.62	14.32	20.80	14.51	22.41	14.46	23.09	10.95	22.83	11.24
WM 3x3	22.40	10.67	21.38	11.61	22.20	10.61	24.21	**9.96**	23.58	**8.67**	24.68	**7.81**
FIRMH 5x5	21.05	12.67	21.44	12.81	21.28	12.35	22.21	12.39	22.37	10.61	21.96	11.17
LUM 5x5	22.20	11.67	22.26	12.50	21.91	12.08	23.10	12.07	23.26	10.62	23.65	11.37
TSM 3x3	21.13	13.44	21.77	13.86	21.47	13.13	23.11	12.43	23.56	10.80	22.96	11.38
ACWM 3x3	22.97	10.57	23.67	11.08	23.26	10.57	**24.84**	10.43	**25.56**	8.75	**25.18**	9.21
ROM 3x3	**23.08**	**10.42**	**23.84**	**10.96**	**23.40**	10.55	24.82	10.57	25.20	9.11	25.04	9.62
MMEM 3x3	22.69	12.23	22.93	12.09	22.92	11.36	24.16	11.09	24.52	9.46	24.40	9.67
LMMSE 3x3	23.03	11.24	22.79	12.10	22.29	**10.28**	24.15	11.08	24.59	9.95	24.75	9.70
MM-KNN (Simple)	21.52	12.80	23.13	11.26	22.19	11.95	24.80	10.28	24.90	8.54	24.74	8.83
MM-KNN (Corte)	23.21	10.45	23.83	10.42	**23.67**	**9.75**	25.45	9.58	**26.38**	7.12	26.37	7.18
MM-KNN (Hampel)	**23.24**	**10.42**	**23.87**	**10.39**	23.60	9.75	**25.50**	**9.50**	26.33	**7.07**	**26.48**	7.18
MM-KNN (Andrew)	23.23	10.44	23.85	10.40	23.63	9.76	25.48	9.53	26.36	7.12	26.40	**7.15**
MM-KNN (Tukey)	23.23	10.45	23.86	10.41	23.64	9.76	25.50	9.56	26.32	7.13	26.41	7.18
MM-KNN (Bernoulli)	23.24	10.46	23.86	10.41	23.64	9.76	25.50	9.56	26.31	7.14	26.41	7.18
WM-KNN (Simple)	**23.24**	**10.43**	23.42	10.86	22.99	10.19	25.22	10.00	25.63	7.62	25.90	7.59
WM-KNN (Corte)	22.82	10.82	23.45	**10.81**	**23.02**	**10.15**	25.29	9.97	25.66	7.60	25.92	**7.55**
WM-KNN (Hampel)	22.72	10.86	23.38	10.85	22.84	10.26	25.19	10.00	25.45	7.67	25.60	7.63
WM-KNN (Andrew)	22.79	10.84	**23.49**	10.77	**23.02**	10.17	**25.36**	**9.92**	**25.68**	**7.56**	**26.07**	7.56
WM-KNN (Tukey)	22.30	11.23	23.02	11.34	22.43	10.74	24.95	10.46	24.88	8.11	24.96	8.12
WM-KNN (Bernoulli)	22.26	11.29	22.87	11.27	22.50	10.60	24.41	10.40	24.79	8.05	24.88	8.04

Tabla 3: Resultados comparativos PSNR en dB y MAE de diferentes imágenes corrompidas por 20% de ruido impulsivo.

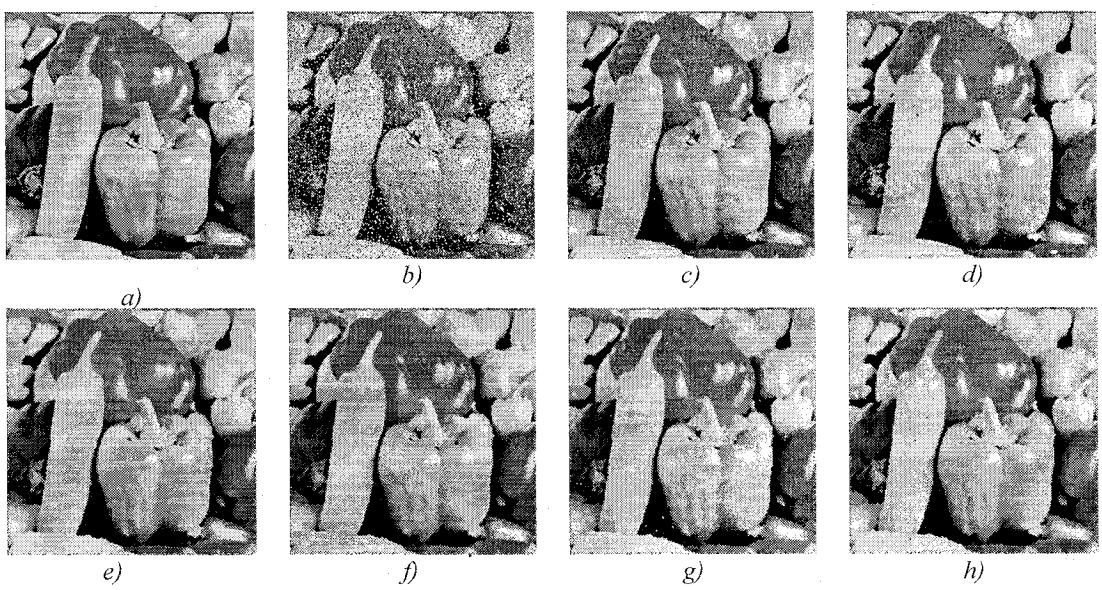


Figura 2: Imágenes de simulación para la supresión de ruido impulsivo en la imagen "Peppers" producidas por diferentes filtros. a) Imagen original de "Peppers"; b) Imagen degradada con 20% de ruido impulsivo, c) Imagen restaurada con el filtro ROM, d) Imagen restaurada con el filtro ACWM, e) Imagen restaurada con el filtro MM-KNN (corte), f) Imagen restaurada con el filtro WM-KNN (corte), g) Imagen restaurada con el filtro MM-KNN (Hampel), y h) Imagen restaurada con el filtro WM-KNN (Hampel).

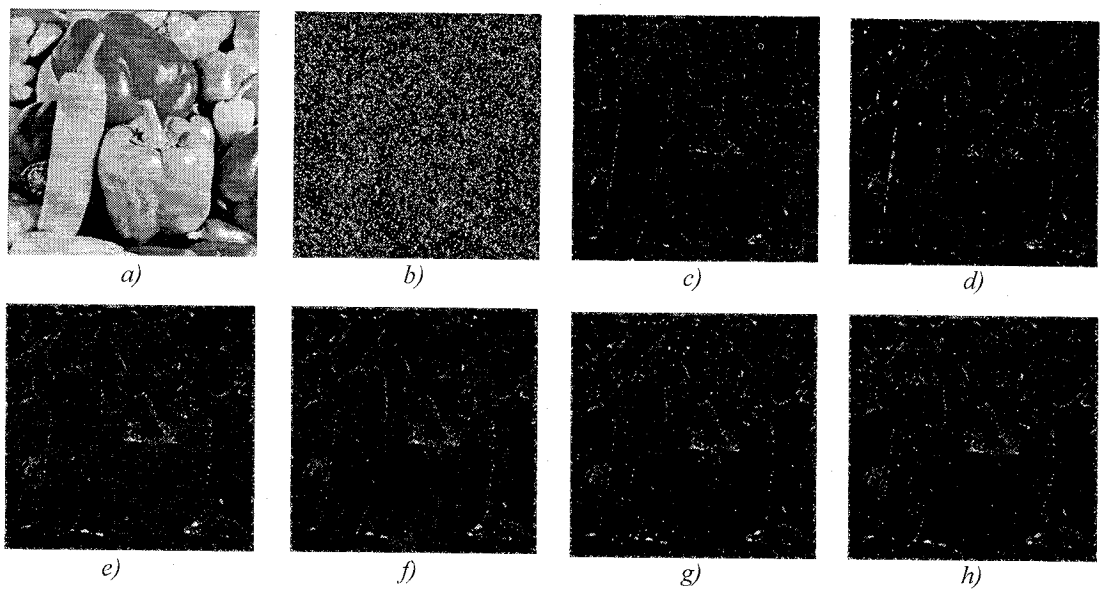


Figura 3: Imágenes de error de la imagen "Peppers" producidas por diferentes filtros. a) Imagen original de "Peppers"; b) Imagen de error degradada con 20% de ruido impulsivo, c) Imagen de error obtenida con el filtro ROM, d) Imagen de error obtenida con el filtro ACWM, e) Imagen de error obtenida con el filtro MM-KNN (corte), f) Imagen de error obtenida con el filtro WM-KNN (corte), g) Imagen de error obtenida con el filtro MM-KNN (Hampel), y h) Imagen de error obtenida con el filtro WM-KNN (Hampel).

Función de Influencia	Valores óptimos de los parámetros de los filtros						
	MM-KNN		WM-KNN		Ambos filtros		
	a	K_{min}	a	K_{min}	α	β	r
Simple	≥ 0.01	5	3.0	5	—	—	—
Corte	4.0	5	3.0	5	—	—	≤90
Hampel	5.0	5	3.0	5	10.0	≤90	300.0
Andrew	4.0	5	3.0	5	—	—	≤81.0
Tukey	4.0	5	3.0	5	—	—	≤255
Bernoulli	4.0	5	3.0	5	—	—	≤255

Tabla 4: Valores óptimos de los parámetros de los filtros propuestos con diferentes funciones de influencia.

5 Análisis en Tiempo Real

El análisis en tiempo real de los filtros MM-KNN y WM-KNN y de otros filtros propuestos como comparativo fueron realizados en las imágenes de las Tablas 2 y 3 mediante el uso del DSP TMS320C6701 de Texas Instruments (Texas Instruments, 1998). La capacidad del TMS320C6701 es de alto desempeño, la arquitectura VLIW (*advanced very long instruction word*) hace al DSP una buena herramienta para aplicaciones en multicanales y multifunciones. La ejecución de hasta 1 GFLOPS con una velocidad de reloj de 167 MHz, el 'C6701 ofrece soluciones efectivas de alto desempeño para los desafíos en la programación de DSPs. El 'C6701 tiene un conjunto completo de herramientas de desarrollo que incluyen: un compilador de C, un optimizador de ensamblador para simplificar la programación y las tareas, y una interfase de depurador para Windows para visualizar el código en ejecución.

Los algoritmos de los filtros fueron implementados en lenguaje C mediante el uso del TURBOC 2.0 para todas las rutinas, estructuras de procesamiento de datos y operaciones de bajo nivel de I/O. Compilamos estos programas en el compilador de C del 'C6701 para crear el archivo de ensamblador, el archivo objeto y el archivo ejecutable COFF (.out). El archivo .out fue simulado en la PC usando el simulador del 'C6701. Finalmente, este archivo fue cargado y ejecutado en la tarjeta 'C6701 usando la utilidad "COFF loader" (Texas Instruments, 2000).

Los valores del tiempo de procesamiento en segundos para los resultados de la Tabla 2 se muestran en la Tabla 5 para el caso del uso de los filtros MM-KNN y WM-KNN con algunas de las funciones de influencia utilizadas. De la Tabla 5 puede verse que estos tiempos casi no varían para diferentes niveles de ruido impulsivo. El tiempo de procesamiento para todos los filtros incluye el tiempo de adquisición, procesamiento y almacenamiento de datos. También se obtuvo el tiempo de procesamiento en segundos para otros filtros utilizados en las pruebas (ver Tabla 2): *Mediano* ≈ 0.0146, *WM* ≈ 0.0203, *FIRMH* ≈ 0.0609, *LUM* ≈ 0.4757, *TSM* ≈ 0.0547, *ACWM* ≈ 0.2299, *ROM* ≈ 0.0750, *MMEM* ≈ 0.0406 y el *LMMSE* ≈ 0.0751. Estos tiempos casi no varían para diferentes niveles de ruido impulsivo.

En la Tabla 6 se presenta el tiempo de procesamiento en segundos para las imágenes usadas en la Tabla 3. Los resultados revelan que el tiempo de procesamiento de los filtros MM-KNN y WM-KNN son mayores que el de otros filtros en la mayoría de los casos. Los filtros usados para comparar nuestro procedimiento fueron computados (Huang, *et al.*, 1979; Pitas, 1989) de acuerdo a sus referencias. En el caso del filtro ROM su tiempo de procesamiento no incluye el tiempo de procesamiento para derivar los coeficientes usados en su esquema de filtrado, este tiempo es mucho mayor comparativamente con el tiempo requerido para efectuar el filtrado.

El tiempo de procesamiento de los filtros MM-KNN y WM-KNN dependen de la imagen a procesar y casi no varía para diferentes niveles de ruido. Estos valores también dependen del cálculo y complejidad de las funciones de influencia y de los parámetros de los filtros propuestos. De los resultados de la Tabla 6, es claro que en el caso de imágenes en niveles de gris de 256x256 la técnica de filtrado propuesta puede procesar entre 8 a 14 imágenes en un segundo dependiendo la función de influencia utilizada.

Algoritmo	Tiempo de procesamiento para diferentes probabilidades de ruido impulsivo		
	5%	10%	15%
MM-KNN (Hampel)	0.0689	0.0689	0.0708
MM-KNN (Andrew)	0.0696	0.0697	0.0716
MM-KNN (Tukey)	0.0575	0.0588	0.0589
MM-KNN Bernoulli	0.0625	0.0651	0.0651
WM-KNN (Hampel)	0.0712	0.0723	0.0734
WM-KNN (Andrew)	0.0854	0.0821	0.0846
WM-KNN (Tukey)	0.0701	0.0701	0.0703
WM-KNN Bernoulli	0.0753	0.0743	0.0747

Tabla 5: Resultados comparativos del tiempo de procesamiento en la imagen "Lena" corrompida por 5%, 10% y 15% de ruido. impulsivo

6 Filtrado en Secuencias de Vídeo

En esta sección consideramos la aplicación de los filtros propuestos a señales de vídeo. Las secuencias de vídeo "Carphone" y "Miss America" de 176x144 pixeles fueron degradadas por 20% de ruido impulsivo. Los resultados del rendimiento PSNR, MAE y Tiempo de procesamiento en una imagen de cada secuencia de vídeo se muestran en la Tabla 7 mediante el uso de diferentes filtros.

De los resultados PSNR y MAE mostrados en la Tabla 7 se observa que los filtros propuestos proveen una mejor supresión de ruido y preservación de detalles que los filtros utilizados como comparativo. Comparando los resultados PSNR y MAE entre los filtros MM-KNN y WM-KNN se observa que son bastante similares, pero los mejores rendimientos son obtenidos mediante el filtro WM-KNN.

Con lo que respecta al tiempo de procesamiento de los filtros propuestos pueden procesar entre 17 a 33 imágenes de vídeo de 176x144 pixeles en un segundo dependiendo de la función de influencia utilizada. Es claro, que la técnica de filtrado propuesta puede procesar imágenes y vídeo de 64x64 pixeles en tiempo real proporcionando buena supresión de ruido y preservación de detalles en aplicaciones de visión por computadora.

Finalmente, los resultados visuales para la imagen "Carphone" se muestran en la Figura 4 mostrando la supresión de ruido impulsivo y preservación de detalles de los filtros propuestos de acuerdo a los resultados de la Tabla 7. En la Figura 5 se presentan las imágenes de error de las imágenes de la Figura 4, en las cuales se observa que los filtros propuestos proveen una mejor supresión de ruido y preservación de detalles en comparación con los filtros usados como comparativo.

Algoritmo	Tiempo de Procesamiento de diferentes imágenes corrompidas por 20% de ruido impulsivo					
	Airfield	Barbara	Boat	Goldhill	Lena	Peppers
Mediano 3x3	0.014634	0.014634	0.014635	0.014635	0.014634	0.014634
WM 3x3	0.020341	0.020168	0.020341	0.020341	0.020341	0.020341
FIRMH 5x5	0.060991	0.060991	0.060991	0.060991	0.060991	0.060991
LUM 5x5	0.475767	0.475737	0.475751	0.475755	0.475753	0.475757
TSM 3x3	0.054756	0.054758	0.054757	0.054758	0.054757	0.054759
ACWM 3x3	0.229967	0.229954	0.229960	0.229961	0.229960	0.229951
ROM 3x3	0.075007	0.075008	0.075008	0.075008	0.075008	0.075008
MMEM 3x3	0.040607	0.040620	0.040606	0.040606	0.040609	0.040618
LMMSE 3x3	0.075070	0.075118	0.075101	0.075087	0.075135	0.075140
MM-KNN (Simplest)	0.107967	0.111440	0.112401	0.117085	0.112582	0.112437
MM-KNN (Skipped)	0.060503	0.059214	0.061038	0.060806	0.059456	0.060382
MM-KNN (Hampel)	0.070546	0.070992	0.072360	0.073086	0.069599	0.071449
MM-KNN (Andrew)	0.075051	0.074246	0.074308	0.079414	0.072212	0.076526
MM-KNN (Tukey)	0.059436	0.059172	0.060846	0.059951	0.058984	0.059672
MM-KNN (Bernoulli)	0.066561	0.065529	0.069023	0.067169	0.065354	0.067001
WM-KNN (Simplest)	0.068654	0.076356	0.071661	0.075138	0.075791	0.075324
WM-KNN (Skipped)	0.109103	0.120524	0.116379	0.128009	0.114114	0.123113
WM-KNN (Hampel)	0.073683	0.078952	0.080814	0.082654	0.081409	0.080385
WM-KNN (Andrew)	0.092012	0.091715	0.094360	0.097901	0.094496	0.096882
WM-KNN (Tukey)	0.075313	0.082425	0.079893	0.080415	0.077514	0.079484
WM-KNN (Bernoulli)	0.069561	0.082425	0.085528	0.086193	0.083879	0.083257

Tabla 6: Resultados comparativos del tiempo de procesamiento de diferentes imágenes corrompidas por 20% de ruido impulsivo.

7 Conclusiones

Presentamos la implementación en tiempo real de filtros para imágenes y/o vídeo con preservación de detalles finos mediante el uso del DSP TMS320C6701. Los filtros propuestos están basados en estimadores combinados RM.

Los filtros MM-KNN y WM-KNN pueden preservar los detalles finos y pueden suprimir el ruido impulsivo mejor que otros filtros no lineales propuestos en la literatura. Las propiedades determinísticas y estadísticas de los filtros diseñados fueron investigadas y mostraron su efectividad. Los valores escogidos para los parámetros de los filtros MM-KNN y WM-KNN proveen una buena supresión de ruido y preservación de detalles.

Los resultados del tiempo total de procesamiento revelan que el tiempo de procesamiento de los filtros propuestos es mayor que el de otros filtros en la mayoría de los casos. El tiempo de procesamiento de los filtros MM-KNN y WM-KNN dependen de la imagen a procesar y casi no varía para diferentes niveles de ruido. Estos valores también dependen del cálculo y complejidad de las funciones de influencia y de los parámetros de los filtros propuestos. En el caso de imágenes en niveles de gris de 2 ó 3 veces menores a 256x256 pixeles la técnica de filtrado propuesta puede preservar los contornos y detalles de pequeña escala, y suprimir el ruido impulsivo prácticamente a la velocidad del film estándar en sistemas de visión por computadora. El tiempo de procesamiento puede disminuirse si se utiliza un DSP con mejor rendimiento que el TMS320C6701.

Algoritmo de filtrado	"Carphone"			"Miss America"		
	PSNR	MAE	Tiempo	PSNR	MAE	Tiempo
WM	23.8371	9.8099	0.008140	35.5312	5.7593	0.008140
TSM	21.5337	13.2486	0.021893	27.8978	8.7422	0.021893
ACWM	24.3492	9.7914	0.092028	29.2845	7.8611	0.092032
MMEM	23.6092	11.1146	0.016184	28.1068	7.0558	0.016174
LMMSE	24.1564	10.0469	0.030022	30.7171	7.1582	0.029955
MM-KNN (Simple)	24.5934	9.2738	0.042641	33.0876	5.9929	0.045078
MM-KNN (Corte)	24.7712	9.0844	0.024282	36.9980	4.2930	0.025650
MM-KNN (Hampel)	24.7632	9.1076	0.027058	39.7071	4.2394	0.029272
MM-KNN (Andrew)	24.7536	**9.0690**	0.032459	36.9580	4.2080	0.030950
MM-KNN (Tukey)	24.7463	9.1613	**0.023951**	37.2216	4.1404	**0.025183**
MM-KNN (Bernoulli)	24.7813	9.1592	0.027178	37.2288	4.1508	0.029142
WM-KNN (Simple)	**24.7834**	9.0841	0.028450	37.1768	**4.1361**	0.030718
WM-KNN (Corte)	24.0877	9.4744	0.055111	**40.6810**	4.6637	0.060505
WM-KNN (Hampel)	24.0223	9.4729	0.030512	35.5855	4.8066	0.032671
WM-KNN (Andrew)	24.0703	9.4602	0.035122	36.5601	4.8840	0.038897
WM-KNN (Tukey)	23.5984	9.8970	0.030399	27.2940	5.7717	0.031807
WM-KNN (Bernoulli)	23.6356	9.8620	0.032960	28.9192	5.9210	0.034230

Tabla 7: Resultados de rendimiento en secuencias de vídeo corrompidas con 20% de ruido impulsivo.

Figura 4: Imágenes de simulación para la supresión de ruido impulsivo en una imagen de la secuencia de vídeo "Carphone" producidas por diferentes filtros. a) Imagen original de "Carphone"; b) Imagen degradada con 20% de ruido impulsivo, c) Imagen restaurada con el filtro ACWM, d) Imagen restaurada con el filtro LMMSE, e) Imagen restaurada con el filtro MM-KNN (Hampel), f) Imagen restaurada con el filtro WM-KNN (simple), g) Imagen restaurada con el filtro WM-KNN (corte), h) Imagen restaurada con el filtro WM-KNN (Hampel).

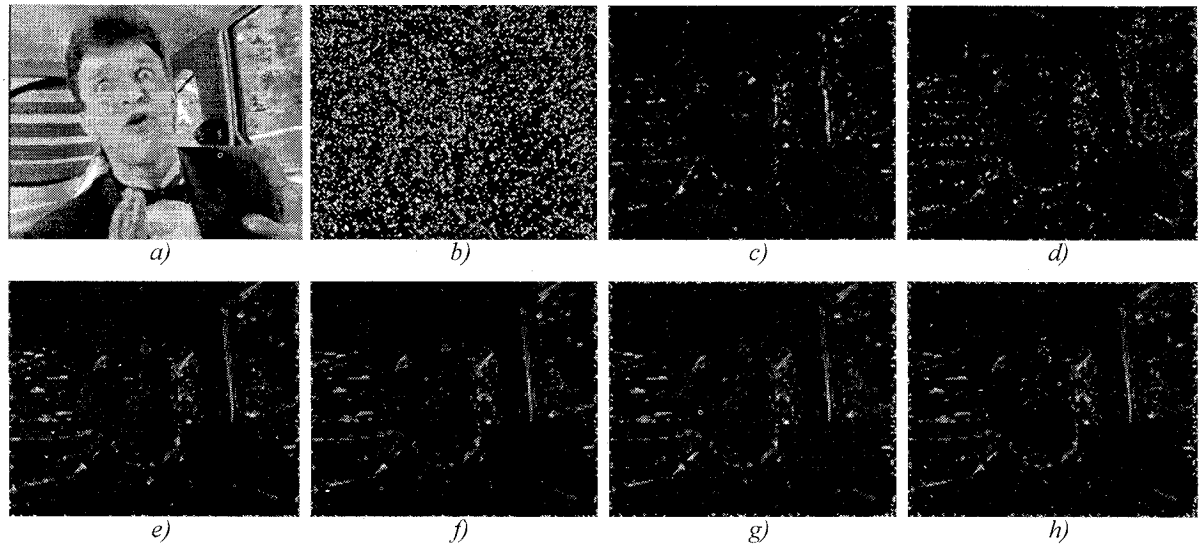


Figura 5: Imágenes de error para la supresión de ruido impulsivo en una imagen de la secuencia de vídeo "Carphone" producidas por diferentes filtros. a) Imagen original de "Carphone"; b) Imagen de error degradada con 20% de ruido impulsivo, c) Imagen de error obtenida con el filtro ACWM, d) Imagen de error obtenida con el filtro LMMSE, e) Imagen de error obtenida con el filtro MM-KNN (Hampel), f) Imagen de error obtenida con el filtro WM-KNN (simple), g) Imagen de error obtenida con el filtro WM-KNN (corte), h) Imagen de error obtenida con el filtro WM-KNN (Hampel).

Agradecimientos

Los autores dan las gracias al Instituto Politécnico Nacional y al programa PROMEP SUPERA por los apoyos brindados.

Referencias

Pitas I., and **A. N. Venetsanopoulos,** *Nonlinear Digital Filters: Principles and Applications,* Kluwer Academic Publisher, 1990.

Astola J., and **P. Kuosmanen,** *Fundamentals of Nonlinear Digital Filtering,* CRC Press, Boca Raton-New York, 1997.

P.J.Huber, *Robust Statistics*. Wiley, 1981.
J. L. Hodges and **E. L. Lehmann**. "Estimates of location based on rank tests". *Ann. Math. Statist.*, 34:598--611, 1963.

Hampel F. R.,. E. M. Ronchetti, P. J. Rouseew, and **W. A. Stahel,** *Robust Statistics. The approach based on influence function*, Wiley, New York, 1986.

Bovik A., *Handbook of Image and Video Processing*, Academic Press, San Diego CA, 2000.

Kotropoulos C., and **I. Pitas,** *Nonlinear Model-Based Image/Video Processing and Analysis*, John Wiley & Sons Inc., 2001.

Mitra S. K., and **G. L. Sicuranza,** *Nonlinear Image Processing,* Academic Press, San Diego CA, 2001.

Ponomaryov V. I., and **O. B Pogrebnyak,** "Novel robust RM filters for radar image preliminary processing", *Journal of. Electronic Imaging,* Vol. 8, No. 4, 1999, pp. 467-477.

Gallegos-Funes F. J., V. I. Ponomaryov, V. F. Kravchenko, and **L. Nino-de-Rivera,** "Robust Detail Preserving Image Filter by Means of Use DSP TMS320C6701", *Telecommunications and Radio Engineering*, Vol. 56, No. 1, 2001, pp. 79-82.

Brownrigg D. R. K., "The Weighted Median Filter", *Commun. Assoc. Comput. Mach*, Vol. 27, Aug. 1984, pp. 887-818.

Yli-Harja O., J. Astola and **Y. Neuvo,** "Analysis of properties of stack and weighted median filters", *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. 39, No. 2, 1991, pp. 395-410.

Wichman R., J. Astola, P. Heinonen and **Y. Neuvo,** "FIR-median hybrid filters with excellent trasient response in noisy conditions", *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. 38, Dec. 1990, pp. 2108-2117.

Hardie R. C., and **C. G. Boncelet,** "LUM filters: a class of rank order based filters for smoothing and sharpening", *IEEE Trans. Signal Processing*, Vol. 41, March 1993, pp. 1061-1076.

Chen T., K. Ma and **L. Chen**, "Tri-State Median filter for image denoising," *IEEE Trans. Image Process.* 8(12), pp. 1834-1838, 1999.

Chen T. and **H. R. Wu,** "Adaptive impulse detection using center-weighted median filters," *IEEE Signal Processing Letters*, 8(1), 1-3, 2001.

Abreu E., M. Lightstone, S. K. Mitra and **K. Arakawa,** "A new efficient approach for the removal of impulse noise from highly corrupted images", *IEEE Trans. Image Process,* Vol. 5, No. 6, 1996, pp. 1012-1025.

Wei-Yu H., and **Lin Ja-Chen,** "Minimum-maximum exclusive mean (MMEM) filter to remove impulse noise from highly corrupted images", *Electronics Letters*, Vol. 33, No. 2, 1997, pp. 124-125.

Özkan M. K., M. I: Sezan and **A. Murat**, "Adaptive motion-compensated filtering of noisy image sequences," *IEEE Trans. Circuits and Syst. For Video Tech.* 3(4), 277-290, 1993.

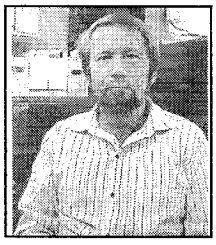
Huang T. S., G. J. Yang and **G. Y. Tang,** "A Fast two-dimentional median filtering algorithm", *IEEE Trans. Acoust. Speech, Signal Process.* ASSP-27, 1979, pp. 13-18.

Pitas I., "Fast algorithms for running order and max/min calculations", *IEEE Trans. Circuits and Systems*, Vol. 36, No. 6, 1989, pp. 795-904.

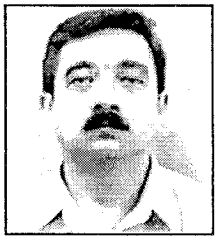
Texas Instruments, *TMS320C6201/6701 Evaluation Module User's Guide,* SPRU269D, Texas Instruments Incorporated, 1998.

Texas Instruments, *TMS320C6000 Optimizing Compiler User's Guide,* SPRU187G, Texas Instruments Incorporated, 2000.

Francisco Javier Gallegos-Funes, *es candidato a Doctor en Comunicaciones y Electrónica por el Instituto Politécnico Nacional. También es profesor investigador de la Escuela Superior de Ingeniería Mecánica y Eléctrica del Instituto Politecnico Nacional de México. Su principal actividad de investigación esta en el área de procesamiento de imágenes y vídeo, filtrado y sistemas en tiempo real.*



Volodymyr I. Ponomaryov, *es Doctor en Ciencias Técnicas del Instituto de Aviación de Jarkov, Ucrania (1981). Actualmente es profesor-investigador de la Escuela Superior de Ingeniería Mecánica y Eléctrica del Instituto Politécnico Nacional de México, ha escrito más de 120 publicaciones, 4 libros y ha desarrollado más de 20 patentes. Su principal actividad de investigación esta en el área de procesamiento de señales e imágenes, filtrado, detección, reconstrucción; SAR y Percepción Remota.*



Oleksiy B. Pogrebnyak, *es Doctor en Ciencias en Sistemas Radiotécnicos del Instituto de Aviación de Jarkov, 1991. Actualmente es profesor investigador del Centro de Investigación en Computación del Instituto Politécnico Nacional de México. Su principal actividad de investigación esta en el área de procesamiento de señales e imágenes: filtrado, detección, reconstrucción y compresión; radares; Percepción Remota.*

Luis Nino-de-Rivera, *es Doctor en Comunicaciones y Electrónica de la UAM, 1999. Actualmente es profesor investigador de la Escuela Superior de Ingeniería Mecánica y Eléctrica del Instituto Politécnico Nacional de México. Su principal actividad de investigación esta en el área de redes neuronales y filtros adaptivos.*