

Filtrado en Tiempo Real Basado en Estimadores Robustos

Francisco J. Gallegos-Funes¹, Volodymyr I. Ponomaryov¹,
Oleksiy B. Pogrebnyak², Luis Niño de Rivera¹

Instituto Politécnico Nacional

¹Escuela Superior de Ingeniería Mecánica y Eléctrica

Av. Santa Ana 1000, Col. San Francisco Culhuacan, 04430, México D.F., México

²Centro de Investigación en Computación

Av. Juan de Dios Batiz s/n, Col. Lindavista, C.P. 07738, México D.F., México

Tel/Fax (52-55)5656-2058, Email: fcogf@hotmail.com

Resumen

En este artículo se presentan las capacidades y características en tiempo real de un filtro robusto para suprimir el ruido impulsivo en aplicaciones de procesamiento de imágenes. La implementación en tiempo real del filtro propuesto fue realizada en el DSP TMS320C6701. Extensivas simulaciones en imágenes conocidas demostraron que el desempeño del filtro propuesto es mejor que otros filtros no lineales tanto en supresión de ruido y preservación de detalles. Finalmente, simulamos secuencias de video corrompidas por ruido impulsivo para demostrar que el método propuesto potencialmente provee una solución en tiempo real para mejorar la calidad de las transmisiones de televisión y video.

Palabras clave: Procesamiento de imágenes, DSP TMS320C6701.

1. Introducción.

Diferentes clases de filtros han sido propuestas para remover el ruido en las imágenes [1-3]. 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, pueden remover el ruido impulsivo y garantizan la preservación de detalles. Los filtros basados en el orden de rangos han recibido considerable atención debido a sus excelentes propiedades de supresión de ruido y preservación de detalles [1, 2]. 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 [1-4]. Los filtros CS (*comparison and selection*), LUM (*lower-upper-middle*) son propuestos para mejoramiento de contornos [2, 4].

En este artículo, presentamos un esquema robusto de filtrado para aplicaciones en procesamiento de imágenes. Nosotros introducimos el filtro MM-KNN que está basado en estimadores robustos combinados R y M para calcular la estimación robusta de los píxeles encerrados en la ventana de filtrado [5, 6]. El filtro propuesto usa el valor del píxel central en la ventana de filtrado para proveer la preservación de detalles y el estimador M combinado con el estimador de la *mediana* proveen suficiente supresión de ruido impulsivo. Varios tipos de funciones de influencia [1, 7] en el estimador M son usados para proveer una mejor supresión de ruido impulsivo.

El resto de este artículo es organizado como sigue: Formulamos el esquema del filtrado propuesto. Resultados experimentales del filtrado de imágenes conocidas mediante el uso de nuestro método y otros métodos basados en la mediana son presentados. Se muestra el tiempo de procesamiento de nuestro método y de varios filtros utilizados como comparativo mediante el uso del DSP de punto flotante TMS320C6701. Finalmente, presentamos algunos resultados experimentales en secuencias de vídeo usando nuestro algoritmo, y mostramos nuestras conclusiones.

2. Formulación e Implementación.

Huber propuso los estimadores M como una generalización de los estimadores de máximo de verosimilitud (MLE) [2, 4]. Su definición es dada por una función ρ $\{\rho(X) = \ln(F(X))\}$ conectada con la función de densidad de probabilidad $F(X)$ de una muestra de datos $X_i, i = 1, \dots, N$ [2]:

$$\hat{\theta} = \arg \min_{\theta \in \Theta} \sum_{i=1}^N \rho(X_i - \theta) \quad (1)$$

La estimación del parámetro local θ puede ser encontrada mediante el cálculo de la derivada parcial de ψ (con respecto a θ).

$$\sum_{i=1}^N \psi(X_i - \theta) = 0 \quad (2)$$

donde θ es un parámetro local.

La solución del estimador robusto M para θ es determinado imponiendo ciertas restricciones en la función de influencia $\psi(X)$ o en las muestras $X_i - \theta$, llamada censorización o "trimming". La técnica estándar para el estimador asume el uso del método iterativo de Newton que puede ser simplificado por un algoritmo de un paso para calcular la estimación M del valor de la media de θ [2]

$$\theta_M = \frac{\sum_{i=1}^N X_i \tilde{\psi}(X_i - \text{MED}\{\mathcal{X}\})}{\sum_{i=1}^N 1_{[-r,r]}(X_i - \text{MED}\{\mathcal{X}\})}, \quad (3)$$

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

La simple restricción del rango de $\psi(X)$ es el estimador M de Huber [5, 7]

$$\tilde{\psi}_b(X) = \text{MIN}(b, \text{MAX}(X, -b)) = [X]_{-b}^b \quad (4)$$

Otra manera para encontrar la función $\tilde{\psi}(X)$ es la de eliminar los valores lejanos de una muestra de datos. Este método es llamado estimaciones M mínimas. Hampel demostró en [7] que la función de corte mediana es la más reducida estimación robusta M . Anteriormente, usamos la función simple de corte. Existen diferentes funciones de influencia propuestas en la literatura. Nosotros también usamos las funciones de influencia Hampel, Andrews, Tukey y Bernoulli [2, 7].

Para mejorar las propiedades robustas de los estimadores R y M mediante el uso de estimaciones R consiste en un procedimiento similar a la media aritmética [5, 6, 9]:

$$\theta_{\text{med}M} = \text{MED}\{X_i \tilde{\psi}(X_i - \text{MED}\{\mathcal{X}\}), i = 1, \dots, N\} \quad (5)$$

Este estimador es llamado estimador combinado RM. Hay que hacer notar que el estimador RM (5) es usualmente la mediana cuando la función ψ es representada por la ec. (4). Si la función $\tilde{\psi}$ es descrita por la función simple de corte, esta se convierte en la función de corte mediana. Es posible suponer que las propiedades de los estimadores robustos RM excedan las propiedades robustas de los estimadores de base R y M .

Para incrementar la robustez de los filtros estándar, es posible emplear diferentes métodos conocidos en la teoría de estimación robusta [1, 2]. Una propuesta para incrementar la calidad de filtración vía preservación de detalles en las imágenes, consiste en el uso de K elementos de una muestra de valores los cuales son encerrados con respecto al valor del pixel central en la ventana de filtrado. Este método es conocido como el algoritmo de filtrado KNN [2].

Para mejorar la robustez del filtro KNN para que pueda preservar los contornos y detalles de pequeña escala en presencia de ruido impulsivo, propusimos el uso de la estimación RM (5). El filtro resultante “Median M-type K-Nearest Neighbor”, que nosotros llamamos simplemente como el algoritmo MM-KNN es escrito como [6, 9]:

$$\hat{f}_{\text{MMKNN}}^{(w)}(i, j) = \text{MED}\{h^{(w)}(i+m, j+n)\} \quad (6)$$

donde $h^{(w)}(i+m, j+n)$ es un conjunto K_{close} de valores de pixeles con peso de acuerdo con la función de influencia $\tilde{\psi}(X)$ encerrados en la ventana de filtrado en la estimación obtenida en el paso previo $\hat{f}_{\text{MMKNN}}^{(w-1)}(i, j)$. La estimación inicial es $\hat{f}_{\text{MMKNN}}^{(0)}(i, j) = x(i, j)$ y $\hat{f}_{\text{MMKNN}}^{(w)}(i, j)$ denota la estimación en la iteración w . $x(i, j)$ es el pixel actual o pixel original contaminado por ruido en la ventana de filtrado. La ventana de filtrado satisface la expresión $N = (2L+1)^2$ y $m, n = -L, \dots, L$. El número actual de pixeles vecinos cercanos $K_{\text{close}}(i, j)$ refleja la actividad local de los datos y la presencia de ruido impulsivo y es determinado como [5, 6]

$$K_{\text{close}}(i, j) = [K_{\text{min}} + aD_S(i, j)] \leq K_{\text{max}} \quad (7)$$

Aquí, el parámetro a controla la sensibilidad del filtro para detectar los detalles en las imágenes. K_{min} es el número mínimo de pixeles vecinos para remover el ruido y K_{max} es el número máximo de vecinos que restringen los contornos y el emborronamiento de los detalles. $D_S(i, j)$ es el detector de picos [6], y MAD es la mediana de las desviaciones absolutas de la mediana [2], los cuales son definidos como:

$$D_S(i, j) = \frac{\text{MED}\{|x(i, j) - x(i+m, j+n)|\}}{\text{MAD}\{x(i, j)\}} + 0.5 \frac{\text{MAD}\{x(i, j)\}}{\text{MED}\{x(i+k, j+l)\}}, \quad (8)$$

$$\text{MAD}\{x(i, j)\} = \text{MED}\{x(i+m, j+n) - \text{MED}\{x(i+k, j+l)\}\} \quad (9)$$

En nuestros experimentos, una ventana 3x3 es aplicada. El algoritmo finaliza cuando $\hat{f}_{\text{MMKNN}}^{(w)}(i, j) = \hat{f}_{\text{MMKNN}}^{(w-1)}(i, j)$. Además, los valores de los parámetros del filtro MM-KNN y de las funciones de influencia fueron encontrados después de numerosas simulaciones [8, 9]. Las funciones de influencia anteriormente mencionadas en el filtro propuesto (6) proveen una buena supresión de ruido

impulsivo [2, 6, 7]. También propusimos la aplicación del filtro mediano estándar para mejorar la habilidad de supresión de ruido del filtro MM-KNN en presencia de ruido impulsivo. Las simulaciones mostraron que cuando $K_{close} > 7$ el filtro MM-KNN puede ser sustituido por el filtro *mediano 3x3* y cuando $K_{close} > 350$ usamos el filtro *mediano 5x5* [6, 8, 9].

3. Rendimiento del Filtro MM-KNN.

El filtro MM-KNN con diferentes funciones de influencia fue evaluado, y su desempeño fue comparado con diferentes filtros no lineales [8-12]. El criterio usado para comparar el rendimiento en supresión de ruido de diferentes filtros fue el *pico de la relación señal a ruido* (PSNR) y para la evaluación de la preservación de detalles el criterio usado fue el *error absoluto medio* (MAE) [1,2].

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

$$MAE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i, j) - \hat{f}(i, j) \quad , \quad (11)$$

donde $MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [f(i, j) - \hat{f}(i, j)]^2$ es el *error cuadrático medio*,

$f(i, j)$ es la imagen original; $\hat{f}(i, j)$ es la imagen restaurada; y M, N es el tamaño de la imagen.

Para determinar las propiedades de supresión de ruido impulsivo de diferentes filtros, las imágenes “Airfield”, “Barbara”, “Boat”, “Bridge”, “Goldhill” y “Peppers” (imágenes en niveles de gris de 256x256 pixeles), fueron degradadas con 20% de ruido impulsivo. El rendimiento PSNR y MAE del filtro MM-KNN 3x3 con las funciones de influencia de *corte mediana*, *Hampel*, *Andrews*, *Tukey* y *Bernoulli* fueron comparados con el rendimiento de los filtros WM 3x3 (*weighted median*), TSM 3x3 (*tri-state median*), ACWM 3x3 (*adaptive center weighted median*), ROM 3x3 (*rank order mean*), MMEM 3x3 (*minimum-maximum exclusive mean*) y LMMSE 3x3 (*Local Linear Minimum Mean Square Error*). Los filtros usados para comparar nuestro método fueron computados de acuerdo a sus referencias [10-16]. La razón de haber escogido estos filtros para compararlos con el nuestro es que su rendimiento ha sido comparado con diferentes filtros conocidos y sus ventajas han sido demostradas. Los valores PSNR y MAE son mostrados en las Tablas 1 y 2 respectivamente.

La Tabla 1 muestra que el rendimiento PSNR del filtro propuesto MM-KNN es mejor en comparación a los filtros utilizados como comparativo, (especialmente cuando el porcentaje de ruido es mayor al 5%). Resultados similares fueron obtenidos al calcular los valores MAE (ver Tabla 2). Uno puede ver que el filtro ROM tiene en algunos casos los mejores valores de supresión de ruido y preservación de detalles, pero la ventaja de nuestro procedimiento es que el filtro propuesto MM-KNN no usa datos de entrenamiento y los parámetros del filtro propuesto pueden ser tomados como constantes. Además, el filtro ROM presenta los mejores resultados visuales en preservación de detalles, pero su desventaja es que tiende a variar grandemente el valor del pixel central en la ventana de filtrado, esto puede ser fácilmente visualizado en imágenes a color. El parámetro a del filtro propuesto 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.

Tabla 1. Valores PSNR en dB para diferentes imágenes corrompidas por 20% de ruido impulsivo.

Filtro	Airfield	Barbara	Boat	Bridge	Goldhill	Peppers
3x3 WM	22.40	21.38	22.20	22.88	24.21	24.68
3x3 TSM	21.13	21.77	21.47	20.57	23.11	22.96
3x3 ACWM	22.97	23.67	23.26	23.14	24.84	25.18
3x3 ROM	23.08	23.84	23.40	23.06	24.82	25.04
3x3 MMEM	22.69	22.93	22.92	22.47	24.16	24.40
3x3 LMMSE	23.03	22.79	22.29	22.58	24.15	24.75
MM-KNN (Corte)	23.21	23.83	23.67	22.90	25.45	26.37
MM-KNN (Hampel)	23.24	23.87	23.60	22.90	25.50	26.48
MM-KNN (Andrew)	23.23	23.85	23.63	22.93	25.48	26.40
MM-KNN (Tukey)	23.23	23.86	23.64	22.92	25.50	26.41
MM-KNN (Bernoulli)	23.24	23.86	23.64	22.92	25.50	26.41

Tabla 2. Valores MAE para diferentes imágenes corrompidas por 20% de ruido impulsivo.

Filtro	Airfield	Barbara	Boat	Bridge	Goldhill	Peppers
3x3 WM	10.67	11.61	10.61	12.76	9.96	7.81
3x3 TSM	13.44	13.86	13.13	15.62	12.43	11.38
3x3 ACWM	10.57	11.08	10.57	11.85	10.43	9.21
3x3 ROM	10.42	10.96	10.55	11.81	10.57	9.62
3x3 MMEM	12.23	12.09	11.36	13.57	11.09	9.67
3x3 LMMSE	11.24	12.10	10.28	12.91	11.08	9.70
MM-KNN (Corte)	10.45	10.42	9.75	12.61	9.58	7.18
MM-KNN (Hampel)	10.42	10.39	9.75	12.59	9.50	7.18
MM-KNN (Andrews)	10.44	10.40	9.76	12.59	9.53	7.15
MM-KNN (Tukey)	10.45	10.41	9.76	12.63	9.56	7.18
MM-KNN (Bernoulli)	10.46	10.41	9.76	12.63	9.56	7.18

Para las imágenes de prueba el valor óptimo de la función de *Hampel* es $\beta=90$. El valor óptimo del parámetro r para la función *Andrews* es $r=35$, para la función de *Tukey* $r=15$ y para la función de *Bernoulli* $r=20$. En las Figuras 1 y 2 se presentan las imágenes procesadas para las imágenes "Barbara" y "Goldhill" mostrando la supresión de ruido impulsivo de acuerdo a las Tablas 1 y 2.

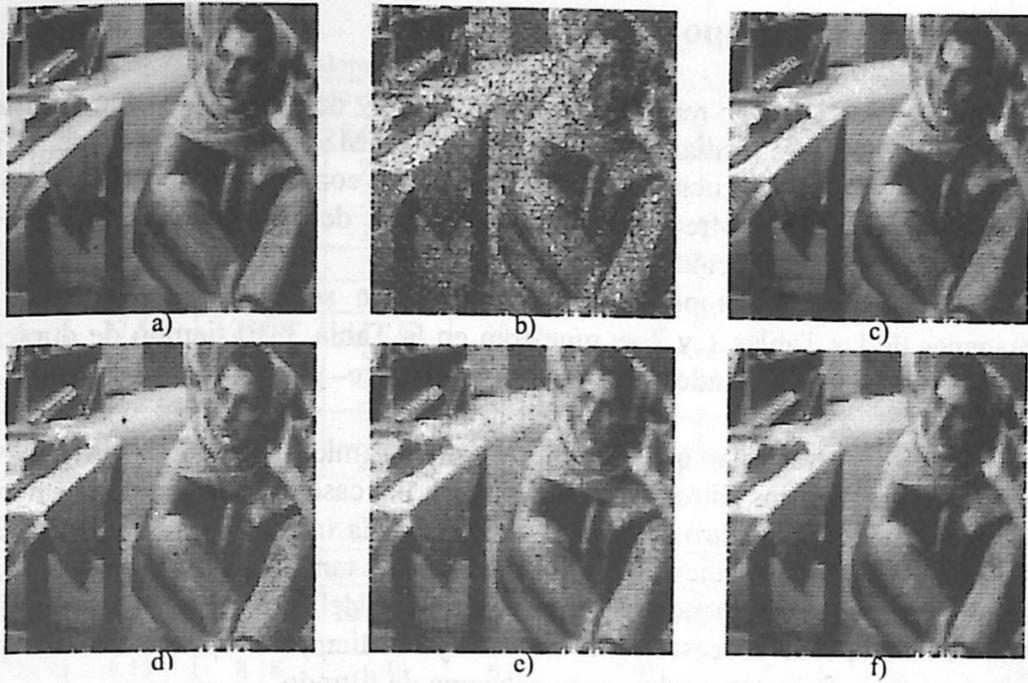


Figura 1. Supresión de ruido impulsivo en la imagen "Barbara". a) Imagen original, b) Imagen degradada con 20% de ruido impulsivo, c) Imagen restaurada con el filtro ACWM, d) Imagen restaurada con el filtro ROM, e) Imagen restaurada con el filtro MMEM, f) Imagen restaurada con el filtro MM-KNN (Corte).

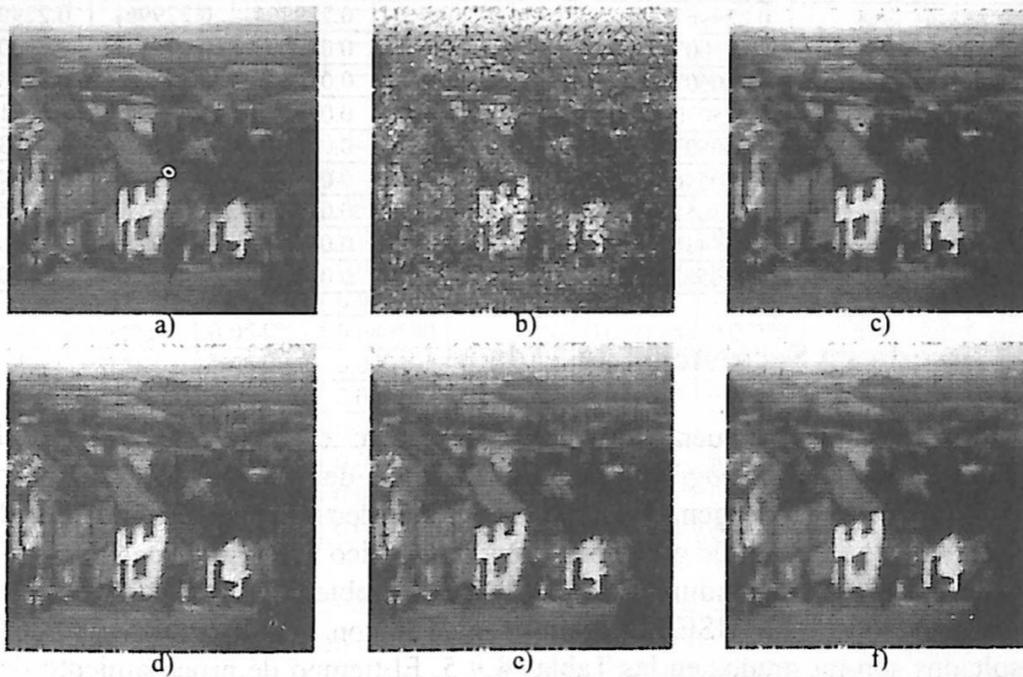


Figura 2. Supresión de ruido impulsivo en la imagen "Goldhill". a) Imagen original, b) Imagen degradada con 20% de ruido impulsivo, c) Imagen restaurada con el filtro ACWM, d) Imagen restaurada con el filtro ROM, e) Imagen restaurada con el filtro LMMSE, f) Imagen restaurada con el filtro MM-KNN (Andrews).

4. Análisis en Tiempo Real.

El análisis en tiempo real del filtro MM-KNN y de otros filtros fue realizado en varias imágenes mediante el uso del DSP TMS320C6701 [15] de Texas Instruments. Con 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 [17-19].

Los valores del tiempo de procesamiento en segundos realizado en las imágenes de las Tablas 1 y 2 se muestran en la Tabla 3. El tiempo de duración incluye el tiempo de adquisición, procesamiento y almacenamiento de las imágenes.

Los resultados revelan que el tiempo de procesamiento del filtro MM-KNN es mayor que el de otros filtros en la mayoría de los casos. La duración del tiempo de procesamiento del filtro propuesto depende de la imagen a procesar y casi no varía para diferentes niveles de ruido, estos valores también dependen del cálculo de las funciones de influencia y de los parámetros del filtro. En el caso del filtro ROM su tiempo de procesamiento no incluye el tiempo de procesamiento para calcular los coeficientes usados en su esquema de filtrado.

Tabla 3. Tiempo de procesamiento para diferentes imágenes corrompidas por 20% de ruido impulsivo.

Filtro	Airfield	Barbara	Boat	Bridge	Goldhill	Peppers
3x3 WM	0.020341	0.020168	0.020341	0.020341	0.020341	0.020341
3x3 TSM	0.054756	0.054758	0.054757	0.054756	0.054758	0.054759
3x3 ACWM	0.229967	0.229954	0.229960	0.229964	0.229961	0.229951
3x3 ROM	0.075007	0.075008	0.075008	0.075007	0.075008	0.075008
3x3 MMEM	0.040607	0.040620	0.040606	0.040607	0.040606	0.040618
3x3 LMMSE	0.075070	0.075118	0.075101	0.075105	0.075087	0.075140
MM-KNN (Corte)	0.060503	0.059214	0.061038	0.060503	0.060806	0.060382
MM-KNN (Hampel)	0.070546	0.070992	0.072360	0.071816	0.073086	0.071449
MM-KNN (Andrews)	0.075051	0.074246	0.074308	0.077871	0.079414	0.076526
MM-KNN (Tukey)	0.059436	0.059172	0.060846	0.060314	0.059951	0.059672
MM-KNN (Bernoulli)	0.066561	0.065529	0.069023	0.067937	0.067169	0.067001

5. Filtrado en Secuencias de Vídeo.

Simulamos una secuencia de vídeo mediante el uso de imágenes PET (*Positron Emission Tomography*) con una relación de muestreo de 10 imágenes por segundo. Cada imagen de la secuencia de vídeo es muestreada a 256x256 píxeles con 256 niveles de gris. La secuencia de vídeo fue degradada con 20% de ruido impulsivo. Los rendimientos PSNR y MAE obtenidos con los filtros WM, TSM, MMEM y LMMSE fueron comparados con el filtro propuesto y los resultados son mostrados en las Tablas 4 y 5. El tiempo de procesamiento de la secuencia de vídeo es presentado en la Tabla 6. Las imágenes procesadas 1 a 4 de la secuencia de vídeo son mostradas en la Figura 3 mediante el uso del filtro LMMSE y el filtro MM-KNN con la función de influencia de corte mediana.

Tabla 4. Valores PSNR en dB para la secuencia de video PET.

Núm. Sec.	Filtros de orden estadístico				Filtro MM-KNN			
	WM	TSM	MMEM	LMMSE	Corte	Hampel	Andrews	Bernoulli
1	23.53	27.90	27.68	29.23	33.04	33.75	32.76	33.00
2	27.49	28.21	27.78	30.12	35.21	36.63	34.93	35.52
3	29.01	28.34	28.16	30.07	37.06	39.03	36.83	37.14
4	30.04	28.79	27.73	31.53	38.51	42.75	37.28	38.47
5	26.58	27.96	27.72	30.66	35.19	35.94	34.39	34.67
6	27.58	28.08	27.77	30.70	36.97	38.75	36.43	36.62
7	32.41	28.12	28.54	31.44	45.90	42.09	41.23	42.46
8	32.36	28.17	28.16	31.89	39.67	42.64	38.42	39.51
9	30.65	28.48	28.17	32.31	44.12	42.50	41.74	43.23
10	30.22	29.08	28.29	31.42	43.18	42.45	40.80	41.90

Tabla 5. Valores MAE para la secuencia de video PET.

Núm. Sec.	Filtros de orden estadístico				Filtro MM-KNN			
	WM	TSM	MMEM	LMMSE	Corte	Hampel	Andrews	Bernoulli
1	8.16	8.26	6.63	7.05	5.17	5.26	5.13	5.24
2	8.17	8.18	6.49	6.85	4.97	5.05	4.90	5.03
3	8.00	8.18	6.41	6.80	4.80	4.82	4.72	4.79
4	7.73	8.01	6.38	6.30	4.56	4.60	4.38	4.55
5	7.96	8.23	6.41	6.57	4.77	4.85	4.70	4.80
6	8.08	8.25	6.68	6.68	4.83	4.91	4.78	4.86
7	7.81	8.08	6.16	6.54	4.46	4.45	4.38	4.44
8	7.60	8.02	6.01	6.71	4.20	4.22	4.14	4.19
9	7.37	7.96	6.09	6.18	4.24	4.19	4.16	4.18
10	7.48	7.66	5.88	6.22	4.02	4.06	3.96	4.01

Tabla 6. Tiempo de procesamiento para la secuencia de video PET.

Núm. Sec.	Filtros de orden estadístico				Filtro MM-KNN			
	WM	TSM	MMEM	LMMSE	Corte	Hampel	Andrews	Bernoulli
1	0.020341	0.054756	0.040582	0.075108	0.060746	0.069584	0.082740	0.068537
2	0.020341	0.054759	0.040588	0.075115	0.061255	0.071004	0.081302	0.067388
3	0.020341	0.054757	0.040592	0.075070	0.061181	0.070612	0.082707	0.067331
4	0.020340	0.054755	0.040590	0.075124	0.061824	0.071712	0.086542	0.068776
5	0.020341	0.054757	0.040592	0.075116	0.061456	0.072268	0.084483	0.070175
6	0.020341	0.054757	0.040585	0.075080	0.062739	0.072810	0.084752	0.069196
7	0.020341	0.054758	0.040597	0.075076	0.062859	0.072597	0.087664	0.068960
8	0.020341	0.054756	0.040596	0.075078	0.061319	0.070744	0.084072	0.069399
9	0.020341	0.054754	0.040591	0.075087	0.062859	0.073974	0.083702	0.070640
10	0.020341	0.054758	0.040596	0.075088	0.061536	0.071503	0.084774	0.070706

De los resultados obtenidos en las Tablas 4 y 5 se observa que el filtro MM-KNN tiene mejores propiedades de supresión de ruido y preservación de detalles en comparación con otros filtros, en el caso de imágenes PET su propia constitución en niveles de gris ayuda grandemente a un mejor filtrado. El filtro LMMSE fue el filtro comparativo que obtuvo los mejores resultados ya que esta diseñado especialmente para secuencias de video [14]. El filtro ROM no se

utilizo debido a que su esquema de filtrado requiere de datos de entrenamiento y no es factible su implementación en tiempo real. Con lo que respecta al tiempo de procesamiento, nuestra técnica tiene el mayor tiempo de procesamiento a excepción del filtro LMMSE. De los resultados de las Tablas 3 y 6, es claro que el sistema puede procesar 10 imágenes de 256x256 con un tiempo de espera de 0.01 a 0.03 seg., dependiendo de la función de influencia usada. En el caso de imágenes de 64x64 la técnica de filtrado propuesta puede suprimir el ruido impulsivo y preservar los detalles finos prácticamente a la velocidad de una película estándar en sistemas de visión por computadora.

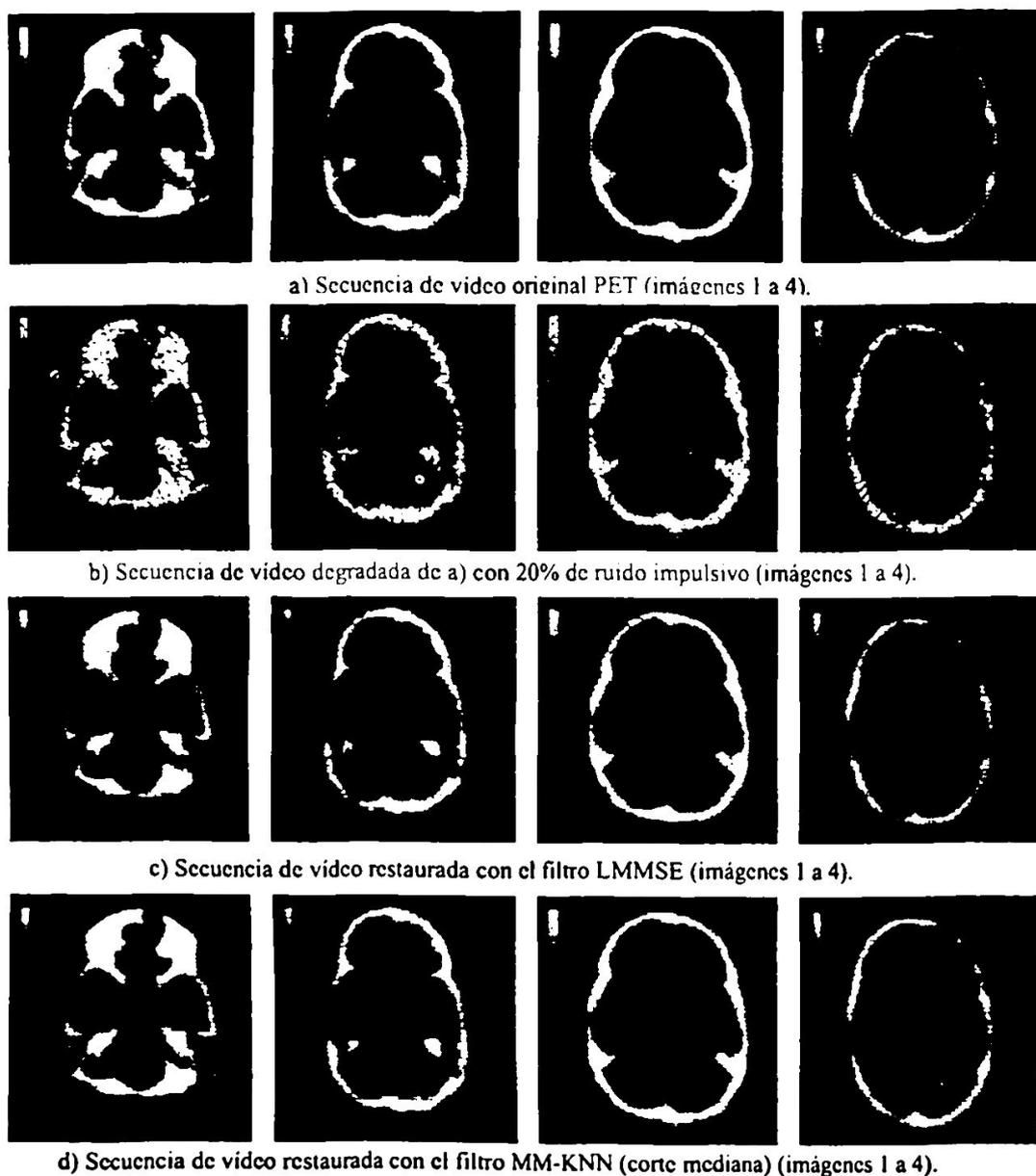


Figura 3. Resultados de supresión de ruido impulsivo en una secuencia de video PET.

6. Conclusiones.

Se presentó la implementación en tiempo real de un filtro robusto para supresión de ruido impulsivo y preservación de detalles mediante el uso del DSP TMS320C6701. El filtro robusto MM-KNN fue diseñado con diferentes funciones de influencia para aplicaciones en procesamiento de imágenes.

Extensivas simulaciones demostraron que desempeño del filtro propuesto tiene mejores propiedades de supresión de ruido y preservación de detalles que otros filtros no lineales presentados en la literatura.

El filtro propuesto provee una solución en tiempo real para la supresión de ruido y preservación de detalles en las transmisiones de televisión y vídeo para mejorar la calidad de éstas.

Es claro que en el caso de imágenes en niveles de gris de 64x64 la técnica de filtrado propuesta puede suprimir el ruido impulsivo y preservar los contornos y detalles finos prácticamente a la velocidad de una película estándar en sistemas de visión por computadora.

Los tiempos de procesamiento pueden ser reducidos si se utiliza un DSP con mayor desempeño que el TMS320C6701 tanto en velocidad de reloj como en ciclos de instrucción.

Agradecimientos. Los autores dan las gracias al CONACYT por el apoyo dado al proyecto de investigación 31978-A. Al Instituto Politécnico Nacional y al programa PROMEP SUPERA por los apoyos brindados.

Referencias

- [1] A. Bovik, *Handbook of Image and Video Processing*, Academic Press, San Diego CA, 2000.
- [2] J. Astola and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering*, CRC Press, Boca Raton-New York, 1997.
- [3] A. Murat, *Digital Video Processing*, Prentice Hall, Upper Saddle River, New Jersey, 1995.
- [4] G. R. Arce, "Multistage order statistic filters for image sequence processing," *IEEE Trans. Signal Process.* 39(5), 1146-1163, 1991.
- [5] V. I. Ponomaryov and O. B. Pogrebnyak, "Novel robust RM filters for radar image preliminary processing," *Journal of Electronic Imaging* 8(04), pp. 467-477, 1999.
- [6] V. I. Ponomaryov, F. J. Gallegos Funes, O. B. Pogrebnyak, Luis N. de Rivera, "Two-stage robust rank filter with fine detail preserving for the image processing," *Proc. SPIE Visual Information Processing X*, vol. 4388, 275-286, 2001.
- [7] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, W. A. Stahel, *Robust Statistics. The approach based on influence function*, Wiley, New York, 1986.

- [8] Francisco J. Gallegos Funes, Volodymyr I. Ponomaryov, Oleksiy B. Pogrebnyak, L. Niño-de-Rivera, "Estimadores robustos RM con diferentes funciones de influencia en el filtro KNN", Memorias del VI Taller Iberoamericano de Reconocimiento de Patrones "Avances Recientes en Análisis de Imágenes y Reconocimiento de Patrones TIARP 2001, pp. 1-21, México, D. F., Noviembre 2001.
- [9] F. J. Gallegos-Funes, V. Ponomaryov, S. Sadovnychiy, L. Niño-de-Rivera, "Median M-type K-nearest neighbour (MMKNN) filter to remove impulse noise from corrupted images", *IEE Electronics Letters*, 38(15), pp. 786-787, 2002.
- [10] T. Chen, K. Ma and L. Chen, "Tri-State Median filter for image denoising," *IEEE Trans. Image Process.* 8(12), pp. 1834-1838, 1999.
- [11] T. Chen and H. R. Wu, "Adaptive impulse detection using center-weighted median filters," *IEEE Signal Processing Letters*, 8(1), 1-3, 2001.
- [12] E. Abreu, 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.* 5(6), pp. 1012-1025, 1996.
- [13] H. Wei-Yu, and Lin Ja-Chen, "Minimum-maximum exclusive mean (MMEM) filter to remove impulse noise from highly corrupted images," *Electronics Letters*, 33(2), pp. 124-125, 1997.
- [14] M. K. Özkan, 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.
- [15] T. S. Huang, G. J. Yang and G. Y. Tang, "A Fast two-dimensional median filtering algorithm," *IEEE Trans. Acoust. Speech, Signal Process.* ASSP-27, pp. 13-18, 1979.
- [16] I. Pitas, "Fast algorithms for running order and max/min calculations," *IEEE Trans. Circuits and Systems*, 36(6), pp. 795-904, 1989.
- [17] Texas Instruments, *TMS320C6201/6701 Evaluation Module User's Guide, SPRU269D*, Texas Instruments Incorporated, 1998.
- [18] Texas Instruments, *TMS320C6000 Optimizing Compiler User's Guide, SPRU187G*, Texas Instruments Incorporated, 2000.
- [19] Texas Instruments, *TMS320C6000 Code Composer Studio Tutorial, SPRU301C*, Texas Instruments Incorporated, 2000.