

Procesamiento en Tiempo Real de Secuencias de Vídeo usando Filtros de Orden Estadístico

Real Time Video Sequence Processing using Statistical Order Filters

Francisco Gomeztagle¹, Francisco Gallegos Funes²,
Volodymyr Ponomaryov¹ y Oleksiy Pogrebnyak³

¹ Escuela Superior de Ingeniería Mecánica y Eléctrica, U.P. Culhuacan del IPN
Av. Santa Ana 1000, Col. San Francisco Culhuacan, CP. 04430, México, D.F., México
email: yponomar@ipn.mx

² Escuela Superior de Ingeniería Mecánica y Eléctrica, U.P. Zacatenco del IPN
Av. IPN s/n, Col. Lindavista, CP. 07738, México D.F., México, email: fgallegosf@ipn.mx

³ Centro de Investigación en Computación del IPN, Av. Juan de Dios Bátiz s/n, Col. Lindavista, C.P. 07738, México D.F., México,
email: olek@pollux.cic.ipn.mx

Artículo recibido en Julio 28, 2004; aceptado Julio 08, 2005

Resumen

En este artículo se presentan los resultados del filtrado de ruido impulsivo usando los algoritmos RM-KNN justificados mediante criterios de rendimiento objetivo y subjetivo, la implementación en tiempo real de los filtros de tipo RM-KNN (Rank M-type K-nearest neighbor) para imágenes fijas y secuencias de video, así como su comparación con algunos filtros ya conocidos por su buen rendimiento y investigados en diferentes artículos. Los filtros propuestos reducen el ruido de tipo impulsivo y a la vez preservan los detalles finos de las imágenes para diferentes aplicaciones como video conferencia, percepción remota y visión artificial. El procesamiento de secuencias se realizó en tiempo real mediante el DSP TMS320C6711.

Palabras Clave: Filtrado de Imágenes, Filtros de Orden Estadístico, Filtros RM-KNN, DSP TMS320C6711.

Abstract

In this paper we present the results of the impulsive noise filtering using the RM-KNN (Rank M-type K-nearest neighbor) algorithms justified by means of objective and subjective criteria of performance, the implementation in real time of the RM-KNN type filters for fixed images and video sequences, as well as their comparison with some filters already known by their good performance and investigated in different articles. The proposed filters reduce the noise of impulsive type and at the same time they preserve the fine details of the images for different applications like video conferencing, remote sensing and computer vision. The data processing was realized in real time by the use of DSP TMS320C6711.

Keywords: Image Filtering, Order Statistic Filters, RM-KNN filters, DSP TMS320C6711.

1 Introducción

Es importante proponer un método de filtrado de imágenes y/o video que sea capaz de reducir el ruido que en muchos casos se puede modelar como ruido impulsivo, existen varias clasificaciones o categorías de filtros dependiendo de su aplicación, la familia de los filtros no lineales han sido diseñados para suprimir el ruido impulsivo y preservar contornos y detalles finos en las imágenes, varios de estos filtros están basados en las teorías de estimación robusta y en la de orden estadístico [1-3].

El propósito de este trabajo es realizar un filtrado de forma robusta sin importar la composición o naturaleza de las imágenes, es decir, pueden ser de banda visual o infrarroja. Por otro lado los filtros propuestos son de tipo RM-KNN y

están basados en la estimación de orden estadístico propuesta primeramente por Huber y Hampel, [4, 5] además utilizan las funciones de influencia simple, Hampel (Hampel's three part redescending), Seno de Andrew, Bernoulli y Tukey (Tukey's biweight) [4, 5].

Los filtros RM-KNN tales como MM-KNN y WM-KNN utilizan los estimadores- R mediano y Wilcoxon [6-8], ahora nosotros presentamos el filtro ABSTM-KNN, el cual usa el estimador- R Ansari-Bradley-Siegel-Tukey en su esquema de filtrado. Se muestran resultados de filtrado en secuencias de video mediante los filtros RM-KNN de acuerdo con los criterios PSNR y MAE. También se reconoce que el filtrado en tiempo real es importante en varias aplicaciones como el procesamiento de imágenes medicas, radar, percepción remota, video conferencias, etc. Finalmente, se presentan los tiempos de procesamiento obtenidos mediante la implementación de los filtros en el DSP TMS320C6701 el cual garantiza una realización en tiempo real de los algoritmos en aplicaciones de procesamiento de imágenes y video.

2 Criterios de Rendimiento de los Algoritmos en Presencia de Ruido

En esta sección se presenta el modelo de ruido usado así como los criterios de rendimiento de supresión de ruido y preservación de detalles para los filtros propuestos y otros utilizados como comparativo.

2.1 Modelo de Ruido

De diferentes modelos de ruido [1-3] nosotros empleamos el siguiente modelo de ruido para el caso de ruido impulsivo,

$$x(i, j) = n_{im}(S(i, j)) \quad (1)$$

donde $S(i, j)$ es la imagen original, $x(i, j)$ es la imagen degradada y $n_{im}(S(i, j))$ esta dada por,

$$n_{im}(S(i, j)) = \begin{cases} \text{una espiga con probabilidad } P \\ S(i, j) \text{ en otro caso} \end{cases} \quad (2)$$

La espiga o ruido impulsivo puede o no afectar el valor de un píxel directamente a cualquiera de los 8 bits del valor del píxel entre el rango de 0 a 255.

2.2 Criterios de Rendimiento del Filtrado

Los criterios usados para evaluar las propiedades de supresión de ruido impulsivo y preservación de detalles de diferentes filtros son los siguientes [1-3]:

La *relación pico señal a ruido* (PSNR),

$$\text{PSNR} = 10 \cdot \log \left[\frac{(255)^2}{\text{MSE}} \right] \text{ dB} \quad (3)$$

y el *error absoluto medio* (MAE),

$$\text{MAE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |S(i, j) - \hat{S}(i, j)| \quad (4)$$

donde $S(i, j)$ es la imagen original, $\hat{S}(i, j)$ es la imagen restaurada, M y N son el tamaño de la imagen y MSE es el error cuadrático medio,

$$\text{MSE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [S(i, j) - \hat{S}(i, j)]^2 \quad (5)$$

Además, para una comparación visual humana subjetiva se presenta la imagen de error dada como,

$$\text{Er}(i, j) = [S(i, j) - \hat{S}(i, j)] \quad (6)$$

3 Los Estimadores RM

En esta sección se describen los estimadores RM los cuales están basados en los estimadores $-R$ y $-M$ y que sirvieron como base para el desarrollo de los filtros de tipo RM.

Los estimadores- R son una clase de estimadores robustos no paramétricos basados en cálculos de rango. Un test de rangos esta basado en el test estadístico [5],

$$S = \frac{1}{m} \sum_{i=1}^m a_i(R_i) \quad (7)$$

donde los coeficientes o pesos a_i son generados por una función J de la siguiente manera:

$$a_i = (m+n) \int_{(i-1)/(m+n)}^{i/(m+n)} J(s) ds \quad (8)$$

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 una distribución Gaussiana, la función $J(s) = \Phi^{-1}(s)$ nos da la solución de acuerdo al criterio de van der Waerden; para una distribución logística la solución esta determinada por la estimación de Hodges-Lehman $J(s) = s - \frac{1}{2}$ y el criterio se transforma al test de Wilcoxon. Finalmente, para una distribución Laplaciana el criterio de signos esta dado en la forma $J(s) = -1$ para $s < 1/2$, $J(s) = 1$ para $s > 1/2$. Este criterio determina el algoritmo de la mediana [5].

Cuando ninguna información *a priori* de la distribución de datos o momentos no esta disponible, el criterio de signos es el más poderoso asintóticamente dentro de los estimadores- R . Cuando la densidad de probabilidad es una función simétrica, el criterio de Wilcoxon es el más poderoso en el mismo sentido. Otros estimadores- R se puede derivar usando diferentes funciones $J(s)$. En este artículo, nosotros proponemos usar la función *Ansari-Bradley-Siegel-Tukey*

$$J(s) = \left| s - \frac{1}{2} \right| - \frac{1}{4} \quad (\text{Huber, 1981}) \text{ con coeficientes } a(i) = \frac{2i - \frac{3}{2}N - 1}{2N} \text{ dados por la ec. (7), entonces el estimador-}R$$

correspondiente esta dado por,

$$\theta_{\text{ABST}} = \text{med} \left\{ \begin{array}{ll} X_{(i)}, & i \leq N/2 \\ \frac{1}{2}(X_{(i)} + X_{(j)}), & i > N/2 \end{array} \right\} \quad (9)$$

La estimación (9) esta compuesta del estimador mediano para la primera mitad de la muestra y para la segunda, el estimador de Wilcoxon es usado.

La forma generalizada del máximo de verosimilitud (maximum likelihood) para los estimadores robustos M fue propuesta por Huber [1, 3, 5]. Su definición esta dada por una función $\rho(X) = -\ln(f(X))$ conectada con la función de densidad de probabilidad $f(X)$ y una muestra de datos $X_i, i = 1, \dots, N$: $\theta = \arg \min_{\theta \in \Theta} \sum_{i=1}^N \rho(X_i - \theta)$. 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 de influencia $\psi(X, \theta) = \frac{\partial}{\partial \theta} \rho(X, \theta)$:

$$\sum_{i=1}^n \psi(X_i - \theta) = 0 \quad (10)$$

La técnica usada para el cálculo de la estimación- M esta basada en el método iterativo de Newton [5]:

$$\hat{\theta}^{(q)} = \frac{\sum_{i=1}^N w[(X_i - \hat{\theta}^{(q-1)})/S_0] X_i}{\sum_{i=1}^N w[(X_i - \hat{\theta}^{(q-1)})/S_0]} \quad (11)$$

donde la función $w(u) = \psi(u)/u$ esta determinada con base en la función de influencia, y $\hat{\theta}^{(q)}$ es la estimación- M del parámetro de localización θ para la iteración q y S_0 es la estimación del parámetro de escala. Usualmente $\hat{\theta}^{(0)} = \text{MED}\{X_i\}$ es la estimación de los datos iniciales y $S_0 = \text{MED}\{X_i - \hat{\theta}^{(0)}\} = \text{MAD}(X_i)$ es la mediana de las derivaciones absolutas de la mediana.

La restricción más simple sobre el rango de la función de influencia es el límite del estimador- M de Huber [4]

$$\varphi_b(X) = \text{MIN}(b, \text{MAX}(X, -b)) = [X]_{-b}^b \quad (12)$$

En este artículo, nosotros también usamos las funciones de influencia simple (skipped mean), *Hampel*, *seno de Andrew's*, *Tukey* y *Bernoulli* [3, 5].

Una propuesta para mejorar las propiedades robustas de los estimadores $-R$ y $-M$ mediante el uso de los estimadores RM, consiste en el cambio de la media vía la mediana o otros estimadores de tipo Wilcoxon, o Ansari-Bradley-Siegel-Tukey [6-8]. De este modo, obtenemos los siguientes estimadores iterativos combinados RM:

Mediana tipo-M (MM),

$$\theta^{(q)}_{\text{MM}} = \text{MED} \left\{ X_i, \varphi_b(X_i - \theta^{(q-1)}), i = 1, \dots, N \right\} \quad (13)$$

Wilcoxon tipo-M (WM),

$$\theta^{(q)}_{\text{WM}} = \text{MED}_{i \leq j} \left\{ \frac{1}{2} \left[X_i \varphi_b(X_i - \theta^{(q-1)})_+ \right], i = 1, \dots, N \right\} \quad (14)$$

así como el nuevo estimador ABST tipo-M (ABSTM),

$$\theta^{(q)}_{\text{ABSTM}} = \text{MED} \left\{ \begin{array}{l} X_i \tilde{\psi}(X_i - \theta^{(q-1)}) \quad , \quad i \leq \left\lfloor \frac{N}{2} \right\rfloor \\ \frac{1}{2} \left[X_i \tilde{\psi}(X_i - \theta^{(q-1)}) + \right. \\ \left. X_j \tilde{\psi}(X_j - \theta^{(q-1)}) \right] \quad , \quad \left\lfloor \frac{N}{2} \right\rfloor < i \leq N \end{array} \right\} \quad (15)$$

4 Esquema del Filtrado RM-KNN

El filtrado RM-KNN utiliza un detector de impulsos que se puede presentar como [7, 8]:

$$S(x(i, j)) = \frac{\text{med}\{x(i, j) - x(i+m, j+n)\}}{\text{MAD}\{x(i, j)\}} + \frac{1}{2} \frac{\text{MAD}\{x(i, j)\}}{\text{med}\{x(i+l, j+k)\}} \quad (16)$$

donde MAD es la mediana de las desviaciones absolutas de la mediana y es calculada como [5],

$$\text{MAD}\{x(i, j)\} = \text{med}\{x(i+k, j+l) - x(i+m, j+n)\} \quad (17)$$

El detector de impulsos se aplica para determinar el número de píxeles vecinos cercanos K_c , que se tomaran en cuenta, a es el parámetro que controla la preservación de detalles finos; K_{\min} es el número mínimo de vecinos necesarios para remover el ruido, K_{\max} es el máximo número de vecinos. En el caso de una ventana de 3x3, $K_{\min}=5$ y $K_{\max}=8$. El valor K_c se determina de acuerdo con,

$$K_c(i, j) = \left[K_{\min} + aS(x(i, j)) \right] \leq K_{\max} \quad (18)$$

El filtro resultante puede ser de tipo mediano MM-KNN, Wilcoxon WM-KNN [7, 8], y el filtro Ansari-Bradley-Siegel-Tukey ABSTM-KNN propuesto en este artículo.

4.1 Estructura de los filtros RM-KNN

Aquí, presentamos la estructura de los filtros MM-KNN, WM-KNN, y ABSTM-KNN. Los filtros mencionados se derivan de los estimadores iterativos combinados RM (13), (14), y (15). El filtro MM-KNN es propuesto por la siguiente ecuación [7]:

$$\tilde{S}_{\text{MMKNN}}^{(q)}(i, j) = \text{med}\{g^{(q)}(i+m, j+n)\} \quad (19)$$

y el filtro WM-KNN esta dado por [9]:

$$\hat{S}_{\text{WMKNN}}^{(q)}(i, j) = \text{med}\left\{ \frac{g^{(q)}(i+m, j+n) + g^{(q)}(i+k, j+l)}{2} \right\} \quad (20)$$

y finalmente el filtro ABSTM-KNN propuesto aquí tiene tal forma

$$S_{\text{ABSTKNN}}^{(q)}(i, j) = \text{med} \left\{ \begin{array}{l} g^{(q)}(i+m, j+n) \quad j, j \leq \frac{N}{2} \\ \frac{g^{(q)}(i+m, j+n) + g^{(q)}(i+m, j+n)}{2} \quad i > \frac{N}{2}, i \leq j \end{array} \right\} \quad (21)$$

En las ecuaciones (19) a (21) $g^{(q)}(i+m, j+n)$ 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 $\hat{S}_{\text{RMKNN}}^{(q-1)}(i, j)$. K_c es el número actual de pixeles vecinos cercanos determinado en ec. (18). El estimador inicial es $\hat{S}_{\text{RMKNN}}^{(0)}(i, j) = x(i, j)$ y $\hat{S}_{\text{RMKNN}}^{(q)}(i, j)$ denota la estimación en la iteración q , $x(i, j)$ es el pixel actual o central contaminado por ruido en la ventana de filtrado, El tamaño de la ventana de filtrado es $N = (2L+1)^2$, $m \leq m_1$, $n \leq n_1$, y $m, n, m_1, n_1 = -L, \dots, L$. El algoritmo finaliza cuando $\hat{S}_{\text{RMKNN}}^{(q)}(i, j) = \hat{S}_{\text{RMKNN}}^{(q-1)}(i, j)$. Usualmente es necesario 2 ó 3 pasos para satisfacer esta condición.

4.2 Funciones de Influencia Usadas

Para realizar los algoritmos de los filtros mencionados se usaron las siguientes funciones de influencia en el estimador- M [3, 5]:

Función de influencia Simple,

$$\psi_{\text{cut}(r)}(X) = \begin{cases} X, & |X| \leq r \\ 0, & |X| > r \end{cases} \quad (22)$$

Función de influencia Andrew,

$$\psi_{\text{sin}(r)}(X) = \begin{cases} \sin(X/r), & |X| \leq r\pi \\ 0, & |X| > r\pi \end{cases} \quad (23)$$

Función de influencia Tukey,

$$\psi_{\text{bi}(r)}(X) = \begin{cases} X^2(r^2 - X^2), & |X| \leq r \\ 0, & |X| > r \end{cases} \quad (24)$$

Función de influencia Hampel,

$$\psi_{\alpha, \beta, r}(X) = \begin{cases} X, & 0 \leq |X| \leq \alpha \\ \alpha \cdot \text{sgn}(X), & \alpha \leq |X| \leq \beta \\ \alpha \frac{r - |X|}{r - \beta}, & \beta \leq |X| \leq r \\ 0, & r \leq |X| \end{cases} \quad (25)$$

Función de influencia Bernoulli,

$$\psi_{\text{ber}(r)}(X) = X^2 \sqrt{r^2 - X^2} \cdot 1_{[-r, r]}(X) \quad (26)$$

donde r es el límite de las funciones de influencia, y α, β para la función de Hampel.

5 Resultados de Simulación

En esta sección, se presenta la simulación de resultados del filtrado impulsivo en la imagen Lena y las secuencias de video “Miss America”, “Lab View” y “Flowers” 256 niveles de gris, las cuales son degradadas por diferentes

porcentajes de ruido impulsivo. Las secuencias de video tienen el formato QCIF con 30 cuadros por segundo y un tamaño de 144x176 píxeles. La Figura 1 muestra las imágenes y secuencias utilizadas.

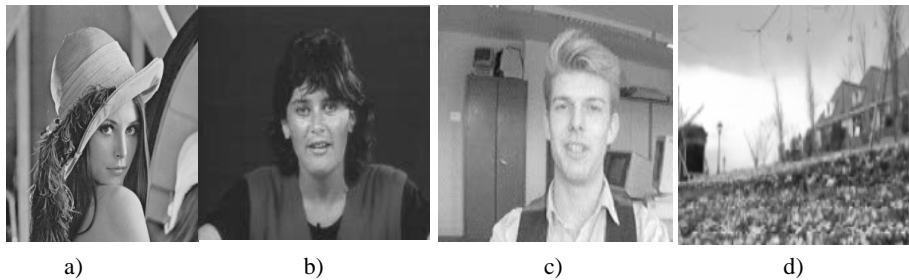


Fig. 1. Imagen y secuencias utilizadas. a) imagen Lena, b) secuencia de video “Miss America”, c) secuencia de video “Lab View”, d) secuencia de video “Flowers”.

Los rendimientos de los filtros RM-KNN con las funciones de influencia *simple*, *Hampel*, *seno de Andrews*, *Tukey* y *Bernoulli* fueron comparados con los filtros *mediano*, *weighted median* (WM) [2, 3], *adaptive center weighted median* (ACWM) [9], *rank order mean* (ROM) [10], *minimum-maximum exclusive mean* (MMEM) [11] y *local linear minimum mean square error* (LMMSE) [12].

5.1 Resultados del Filtrado de Imágenes Estáticas

La investigación del comportamiento de los filtros de orden estadístico consiste en la degradación de las imágenes con ruido impulsivo en un rango de 0 a 20% y luego filtrar éstas, usando varios filtros de orden estadístico. Finalmente, el comportamiento de las imágenes resultantes se evalúa con los criterios objetivos PSNR y MAE. Cabe recordar que el rendimiento de un filtro es mejor que otro cuando los valores de PSNR son mayores y cuando los valores de MAE son menores.

Los resultados del filtrado de la imagen “Lena” son presentados en la Tabla 1 para los filtros usados como comparativo y los filtros MM-KNN, WM-KNN así como el filtro propuesto ABSTM-KNN con diferentes funciones de influencia. De la Tabla 1, se puede observar que los filtros RM-KNN proveen mejor supresión de ruido y preservación de detalles que los filtros usados como comparativo. También puede verse que el filtro ABSTM-KNN provee en algunas ocasiones mejores resultados que los filtros MM-KNN y WM-KNN.

Tabla 1. Resultados PSNR y MAE para diferentes filtros en la imagen “Lena” degradada con diferentes porcentajes de ruido impulsivo.

Diferentes Filtros	Porcentaje de ruido impulsivo									
	0%		5%		10%		15%		20%	
	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE
Mediano	23.23	8.17	23.21	8.22	23.10	8.37	22.98	8.50	22.91	8.64
ROM	24.86	6.55	24.96	6.56	24.85	6.93	24.73	7.07	24.51	7.38
MMEM	23.90	8.80	21.37	10.56	19.63	12.73	18.46	14.62	17.66	16.35
ACWM	23.46	7.94	23.44	7.99	23.37	8.08	23.23	8.19	23.14	8.33
WM	23.69	8.06	24.50	7.83	24.43	7.93	24.31	8.05	24.26	8.13
LMMSE	23.40	13.27	22.37	13.37	21.24	13.97	19.97	15.14	18.90	16.82
MM-KNN Simple	31.74	2.91	31.08	3.20	30.21	3.61	29.30	4.04	26.71	6.05
MM-KNN Andrew	31.73	2.91	30.11	3.33	28.98	3.82	28.23	4.27	27.47	4.75
MM-KNN Tukey	31.74	2.92	30.89	3.24	29.96	3.68	28.93	4.15	28.08	4.65
MM-KNN Hampel	31.70	2.95	31.02	3.25	30.06	3.67	29.20	4.10	28.47	4.54
MM-KNN Bernoulli	31.74	2.92	30.97	3.23	30.01	3.66	28.92	4.14	28.08	4.64
WM-KNN Simple	31.74	2.92	31.05	3.22	30.16	3.63	29.27	4.06	28.48	4.52
WM-KNN Andrew	31.75	2.91	30.40	3.38	29.18	3.94	28.11	4.48	27.60	4.92
WM-KNN Tukey	31.74	2.92	28.86	3.88	27.50	4.72	26.49	5.41	26.20	5.80
WM-KNN Hampel	31.74	2.92	29.80	3.56	28.66	4.17	27.80	4.73	27.27	5.19
WM-KNN Bernoulli	31.73	2.94	28.89	3.88	27.48	4.69	26.45	5.39	26.11	5.80
ABSTM-KNN Sim.	31.75	2.90	31.07	3.20	30.19	3.62	29.28	4.05	28.51	4.50
ABSTM-KNN And.	31.75	2.86	30.06	3.44	28.66	4.09	27.66	4.68	27.17	5.13
ABSTM-KNN Tuk.	31.80	2.90	30.10	3.40	28.70	4.10	27.70	4.70	27.20	5.10
ABSTM-KNN Ham.	31.70	2.90	31.10	3.20	30.20	3.62	29.30	4.05	28.50	4.51
ABSTM-KNN Bern.	31.80	2.86	30.10	3.44	28.70	4.09	27.70	4.68	27.20	5.13

5.2 Resultados del Filtrado de Secuencias de Vídeo

Las secuencias de vídeo al igual que la imagen Lena fueron degradadas con diferentes porcentajes de ruido impulsivo y también se filtraron usando diferentes filtros de orden estadístico [13].

En la Tabla 2 presentamos los resultados del filtrado en la secuencia de vídeo “Lab View” degradada con porcentajes de 5, 10, 15 y 20% de ruido impulsivo para el primer cuadro o imagen, tanto para los filtros de comparativo como los filtros propuestos de tipo de RM-KNN. De la Tabla 2, se observa que aun y cuando los filtros propuestos tiene mejores resultados que los filtros de comparativo, en general los mejores resultados son para el filtro WM-KNN en especial para la función de influencia simple, aunque la función Hampel en el filtro MM-KNN resulta mejor que en los filtros WM-KNN y ABSTM-KNN.

Procesamiento en Tiempo Real de Secuencias de Video usando Filtros de Orden Estadístico

Tabla 2. Resultados PSNR y MAE para diferentes filtros en el primer cuadro de la secuencia de video “Lab View” degradada con diferentes porcentajes de ruido impulsivo.

Diferentes Filtros	Porcentaje de ruido impulsivo							
	5%		10%		15%		20%	
	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE
ACWM	20.31	8.07	20.25	8.20	20.18	8.39	20.09	8.57
Mediano	19.54	8.23	19.45	8.36	19.37	8.55	19.24	8.73
WM	20.28	4.96	20.54	5.83	20.94	6.78	20.01	8.30
ROM	20.26	10.01	20.20	10.21	20.12	10.58	19.93	11.09
Wilcoxon	19.15	13.70	18.44	16.15	17.71	18.93	16.97	21.68
LMMSE	18.67	15.85	9.44	26.53	17.25	17.16	16.51	18.73
MMEM	11.52	8.17	10.81	10.71	18.84	7.89	7.26	25.67
MM-KNN Simple	20.84	7.69	20.80	7.79	20.71	7.99	20.63	8.19
MM-KNN Andrew	20.96	6.31	20.74	6.76	20.53	7.27	20.29	7.84
MM-KNN Tukey	20.82	6.42	20.56	6.90	20.36	7.42	19.89	8.18
MM-KNN Hampel	21.15	5.91	21.07	6.22	20.97	6.57	20.85	7.02
MM-KNN Bernoulli	21.11	6.00	20.99	6.35	20.85	6.76	20.75	7.19
WM-KNN Simple	21.15	5.93	21.07	6.24	20.97	6.59	20.85	7.03
WM-KNN Andrew	21.11	6.01	21.00	6.38	20.89	6.79	20.74	7.28
WM-KNN Tukey	21.02	6.24	20.86	6.75	20.74	7.20	20.53	7.78
WM-KNN Hampel	21.15	5.91	21.07	6.23	20.97	6.58	20.85	7.02
WM-KNN Bernoulli	21.01	6.25	20.85	6.75	20.73	7.19	20.52	7.77
ABSTM-KNN Simple	19.92	6.87	19.88	7.18	19.79	7.58	19.72	7.89
ABSTM-KNN Andrew	19.87	7.03	19.78	7.46	19.65	7.98	19.55	8.40
ABSTM-KNN Tukey	19.87	7.03	19.78	7.46	19.65	7.98	19.55	8.40
ABSTM-KNN Hampel	19.92	6.87	19.88	7.18	19.79	7.58	19.72	7.89
ABSTM-KNN Bernoulli	19.87	7.03	19.78	7.46	19.65	7.98	19.55	8.40

En la Tabla 3 se muestran los resultados para el primer cuadro de las secuencias de video “Miss America” y “Flowers” cuando fueron degradadas con 20% de ruido impulsivo. De la Tabla 3 podemos observar que los mejores resultados para la secuencia “Flowers” se dan en las funciones de influencia Hampel y Bernoulli en los filtros MM-KNN, WM-KNN y ABSTM-KNN. Por otro lado, en la secuencia “Miss America” los mejores resultados se dan con el filtro MM-KNN Hampel y con los filtros WM-KNN y ABSTM-KNN con las funciones de influencia Hampel y Simple.

Tabla 3. Resultados PSNR y MAE para diferentes filtros en el primer cuadro de las secuencias de video “Flowers” y “Miss America” degradadas con 20% de ruido impulsivo.

Diferentes Filtros	Flowers		Miss America	
	PSNR	MAE	PSNR	MAE
ACWM	6.01	47.17	25.11	4.44
Mediano	6.09	36.14	24.48	5.40
WM	6.27	38.33	24.63	3.85
ROM	5.49	45.13	24.94	5.93
Wilcoxon	5.90	40.29	18.57	16.84
LMMSE	5.82	48.47	11.64	24.05
MMEM	6.10	37.29	18.33	9.48
MM-KNN Simple	6.28	36.89	25.28	4.08
MM-KNN Andrew	6.25	35.15	23.21	8.15
MM-KNN Tukey	6.18	35.46	18.91	8.65
MM-KNN Hampel	6.35	34.12	25.50	3.56
MM-KNN Bernoulli	6.33	34.42	22.75	4.99
WM-KNN Simple	6.35	34.18	25.40	3.66
WM-KNN Andrew	6.32	34.65	24.55	4.15
WM-KNN Tukey	6.29	34.87	24.68	4.97
WM-KNN Hampel	6.35	34.14	25.38	3.68
WM-KNN Bernoulli	6.28	35.12	24.68	4.97
ABSTM-KNN Simple	6.35	34.16	24.93	4.00
ABSTM-KNN Andrew	6.31	34.86	20.38	9.38
ABSTM-KNN Tukey	6.31	34.86	20.38	9.38
ABSTM-KNN Hampel	6.35	34.16	24.93	4.00
ABSTM-KNN Bernoulli	6.31	34.86	20.38	9.38

De las Tablas 2 y 3 se observa que las variaciones usando las funciones de influencia aun cuando son pequeñas entre las imágenes estáticas (en el caso de la imagen “Lena”) y en secuencias de video en el desempeño de los filtros MM-KNN, WM-KNN y ABSTM-KNN son debidas a la diferente composición de las imágenes y cuadros de las secuencias de video haciendo que cada filtro con cada función de influencia tenga una respuesta única. Podemos decir que las funciones de influencia con mejor desempeño en cualquier filtro RM-KNN son las funciones Hampel y Simple.

5.3 Procesamiento en DSP

El procesamiento en tiempo real se realizo en el DSP TMS320C6711 de Texas Instruments, él cual permite hasta 1 millón de operaciones por segundo [14]. Los tiempos de procesamiento de diferentes filtros incluyen el tiempo de adquisición, procesamiento y almacenamiento, y están dados en segundos.

La tabla 4 presenta el tiempo de procesamiento para las secuencias “Flowers”, “Miss America” y “Lab View” para los filtros usados como comparativo. De la Tabla 4, se puede ver que los tiempos de procesamiento no cambian para las diferentes secuencias debido a que los cuadros son del mismo tamaño, y que los filtros realizan casi el mismo número de operaciones de filtrado para los diferentes cuadros. El uso del DSP TMS320C6711 permite que los resultados del tiempo de procesamiento sean bastante similares y solo cambian en la séptima u octava cifra significativa.

Tabla 4. Resultados del tiempo de procesamiento para diferentes filtros usados como comparativo en todas las secuencias de video utilizadas.

Filtros de Comparativo	Tiempo de Procesamiento (seg)
Mediano	0.006550
Entrenamiento ROM	0.050000
ROM	0.009250
ACWM	0.106200
WM	0.033900

La Tabla 5 muestra los resultados del tiempo de procesamiento para las secuencias de video “Miss America” y “Lab View” degradadas con 20% de ruido impulsivo mediante los filtros MM-KNN, WM-KNN y ABSTM-KNN. En esta Tabla se presentan los tiempos de procesamiento mínimo y máximo para la secuencias de video utilizadas. Estos tiempos fueron encontrados al procesar todos los cuadros o imágenes de las secuencias de video.

Tabla 5. Resultados del tiempo de procesamiento para las secuencias de vídeo “Miss America” y “Lab View” degradadas con 20% de ruido impulsivo mediante los filtros MM-KNN, WM-KNN y ABSTM-KNN.

Filtros Propuestos	Secuencias de Vídeo			
	Lab View		Miss America	
	Mínimo	Máximo	Mínimo	Máximo
MM-KNN Simple	0.012516	0.012643	0.012605	0.012749
MM-KNN Andrew	0.020740	0.020820	0.025482	0.025698
MM-KNN Tukey	0.011440	0.011640	0.011435	0.011635
MM-KNN Hampel	0.014121	0.014255	0.012420	0.012630
MM-KNN Bernoulli	0.012471	0.012628	0.013122	0.013247
WM-KNN Simple	0.015250	0.015420	0.012516	0.012666
WM-KNN Andrew	0.013100	0.013210	0.017191	0.017406
WM-KNN Tukey	0.016170	0.016440	0.017917	0.018154
WM-KNN Hampel	0.015660	0.015840	0.012698	0.012878
WM-KNN Bernoulli	0.016350	0.016650	0.032641	0.033246
ABSTM-KNN Simple	0.076100	0.082380	0.076180	0.083220
ABSTM-KNN Andrew	0.087890	0.094510	0.087360	0.092760
ABSTM-KNN Tukey	0.086600	0.092900	0.087190	0.093540
ABSTM-KNN Hampel	0.092838	0.100967	0.088961	0.095152
ABSTM-KNN Bernoulli	0.105025	0.111347	0.102262	0.107664

La Tabla 6 presenta los resultados del tiempo de procesamiento mínimo y máximo para la secuencia de video “Flowers” degradada con el 20% de ruido impulsivo mediante el uso de los filtros WM-KNN y ABSTM-KNN.

Procesamiento en Tiempo Real de Secuencias de Vídeo usando Filtros de Orden Estadístico

De las Tablas podemos observar que con el tiempo de procesamiento obtenido podemos tener prácticamente un filtrado en tiempo real, ya que podemos filtrar hasta 30 cuadros por segundo, con los filtros WM-KNN y MM-KNN tenemos tiempos cercanos a los 0.01 seg. Notamos también, que el peor de los casos se presenta en el filtro ABSTM-KNN con tiempos promedio de 0.08 seg., vemos que el desempeño en el filtrado no se afecta de manera significativa al cambiar de un tipo de imagen a otra.

Tabla 6. Resultados del tiempo de procesamiento para la secuencia de vídeo “Flowers” degradada con 20% de ruido impulsivo mediante los filtros WM-KNN y ABSTM-KNN.

Filtros	Secuencia “Flowers”	
	Mínimo	Máximo
WM-KNN Simple	0.180160	0.192800
WM-KNN Andrew	0.228000	0.237000
WM-KNN Tukey	0.167640	0.181640
WM-KNN Hampel	0.159140	0.171928
WM-KNN Bernoulli	0.185930	0.203000
ABSTM-KNN Simple	0.087100	0.092400
ABSTM-KNN Andrew	0.087400	0.095700
ABSTM-KNN Tukey	0.090600	0.092900
ABSTM-KNN Hampel	0.093400	0.102200
ABSTM-KNN Bernoulli	0.075800	0.080400

5.4 Calidad Subjetiva de Imágenes de Secuencias Filtradas

Presentamos algunas imágenes obtenidas después del filtrado en las secuencias de vídeo con varios tipos de filtros con el objetivo de hacer una comparación visual.

En las Figuras 2, 3 y 4 se muestran los resultados visuales en las secuencias de vídeo “Flowers”, “Miss America” y “Lab View” respectivamente, degradadas con 20% de ruido impulsivo. En estas Figuras puede observarse que los mejores resultados visuales son proporcionados por los filtros propuestos.

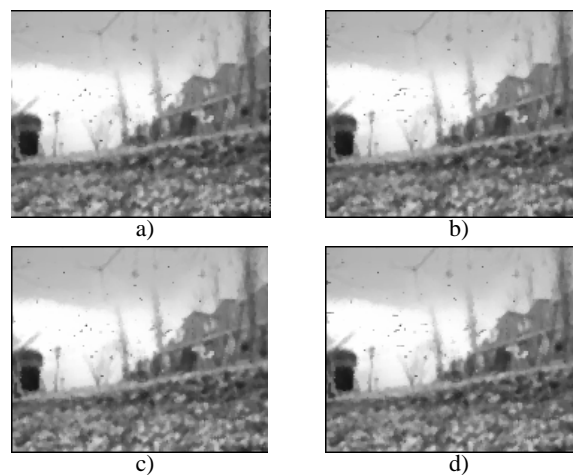
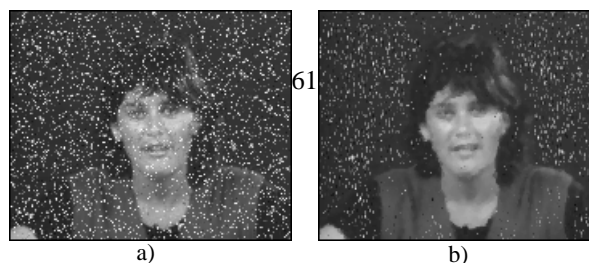


Fig. 2. Resultados visuales en un cuadro de la secuencia de vídeo “Flowers”. a) cuadro restaurado con el filtro ACWM, b) cuadro restaurado con el filtro MM-KNN Hampel, c) cuadro restaurado con el filtro ABSTM-KNN Simple, d) cuadro restaurado con el filtro WM-KNN Tukey.



61

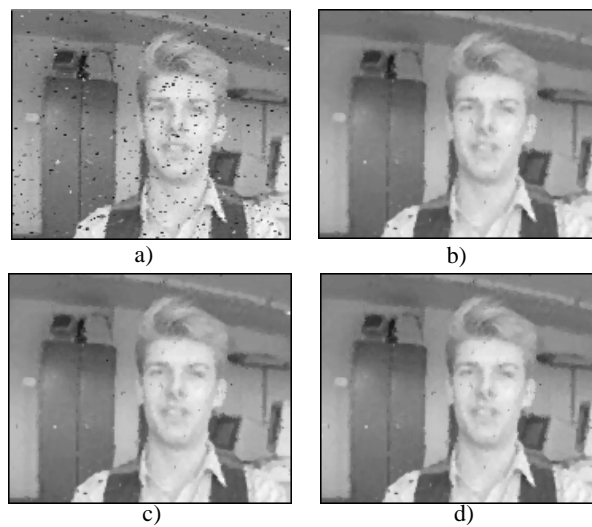


Fig. 4. Resultados visuales en un cuadro de la secuencia de video “Lab View”. a) cuadro restaurado con el filtro WM, b) cuadro restaurado con el filtro ABSTM-KNN Simple, c) cuadro restaurado con el filtro MM-KNN Bernoulli, d) cuadro restaurado con el filtro WM-KNN Simple.

6 Conclusiones

Se presento una comparación de rendimiento de los filtros robustos propuestos ABSTM-KNN, WM-KNN, MM-KNN y los filtros de orden estadístico usados como comparativo para su aplicación en procesamiento de secuencias de video y en imágenes fijas. La implementación en el DSP TMS320C6711 da como resultado un procesamiento en tiempo real o casi real. El filtrado propuesto demostró su robustez en la preservación de detalles finos y remueve bien el ruido impulsivo. Esta comparación se realizo tanto en rendimiento del filtrado y en sus tiempos de procesamiento. Realizamos pruebas en DSP en el filtrado de secuencias de video con formato QCIF para video conferencia y concluimos que se puede realizar en tiempo real ya que filtramos los cuadros de acuerdo al formato que es de 30 cuadros por segundo.

Agradecimientos. Los autores dan las gracias al Instituto Politécnico Nacional por los apoyos brindados.

References

1. **Pitas, I. ; and Venetsanopoulos, A. N.,** *Nonlinear Digital Filters: Principles and Applications*, (Kluwer Academic Publisher, 1990).
2. **Bovik, A.,** *Handbook of Image and Video Processing*, (San Diego: Academic Press, 2000).
3. **Astola, J. ; and Kuosmanen, P.,** *Fundamentals of Nonlinear Digital Filtering*, (Boca Raton-New York: CRC Press, 1997).
4. **Huber, P.J.,** *Robust Statistics*, (Wiley, 1981).
5. **Hampel, F. R. et al.,** *Robust Statistics. The approach based on influence function*, (New York: Wiley, 1986).
6. **Gallegos-Funes, F. et al.,** "Filtros Robustos RM-KNN con Diferentes Funciones de Influencia para Supresión de Ruido Impulsivo en Imágenes Digitales," *Computación y Sistemas*, 6 (3): 183-195 (2003).
7. **Ponomaryov, V. I. et al.,** "Real-Time Processing Scheme Based on RM Estimators," *Proc. SPIE Real Time Imaging VII*, 5012, (Santa Clara, USA, 2003).
8. **Gallegos-Funes, F. J.; and Ponomaryov, V. I.,** "Real-time image filtering scheme based on robust estimators in presence of impulsive noise," *Real Time Imag.*, 8 (2): 69-80 (2004).
9. **Chen, T. ; and Wu, H. R.,** "Adaptive impulse detection using center-weighted median filters," *IEEE Signal Processing Letters*, 8(1): 1-3 (2001).
10. **Abreu, E. et al.,** "A new efficient approach for the removal of impulse noise from highly corrupted images," *IEEE Trans. Image Process*, 5 (6): 1012-1025 (1996).
11. **Wei-Yu, H. ; and Ja-Chen, Lin.,** "Minimum-maximum exclusive mean (MMEM) filter to remove impulse noise from highly corrupted images," *Electronics Lett.*, 33 (2): 124-125 (1997).
12. **Özkan, M. K. ; Sezan, M. I. ; and Murat, A.,** "Adaptive motion-compensated filtering of noisy image sequences," *IEEE Trans. Circuits Syst. Video Tech.*, 3 (4): 277-290 (1993).
13. **Cote, G. et al.,** "H.263+: Video Coding at Low Bit Rates," *IEEE Trans. Circuits Syst. Video Technol.*, 8 (7): 849-866 (1998).
14. *Texas instruments Inc.: TMS320C6201/6701 evaluation module user's*, (Texas Ins., 1998).

Francisco Gomeztagle, et al.



Francisco Gomeztagle. Es Maestro en Ciencias en Microelectrónica por el Instituto Politécnico Nacional de México, 2004. Su principal actividad de investigación es en el área de procesamiento de imágenes y video.



Francisco J. Gallegos Funes. Es Doctor en Ciencias en Comunicaciones y Electrónica por el Instituto Politécnico Nacional de México. Es profesor investigador de la Escuela Superior de Ingeniería Mecánica y Eléctrica del Instituto Politécnico Nacional de México. Su principal actividad de investigación es en el área de procesamiento de imágenes y video, detección, filtrado; Percepción remota y sistemas en tiempo real.



Volodymyr I. Ponomaryov. Es Doctor en Ciencias Técnicas del Instituto de Aviación de Harkov, Ucrania, 1981. Actualmente es profesor investigador 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 es en el área de procesamiento de señales e imágenes: filtrado, detección, reconstrucción y compresión; SAR y Percepción Remota.



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