

Estudio de algoritmos de reducción de ruido en video

Study of algorithms for video noise reduction

SANDRA CANCINO SUÁREZ¹, ENRIQUE ESTUPIÑÁN ESCALANTE¹, MARCELA RODRÍGUEZ RAMÍREZ¹,
DOMINIK MAKOWSKI²

1. Ingeniería Electrónica, Escuela Colombiana de Ingeniería Julio Garavito.

2. M. Sc. Eng. Electronics and Telecommunication, Poznan University of Technology.

Recibido: 30/10/2009 Aceptado: 20/11/2009

Disponible en <http://www.escuelaing.edu.co/revista.htm>

Resumen

En este artículo se describe la implementación de algoritmos para reducción de ruido. Estos algoritmos se aplican a videos con diferentes características de movimiento, esto es: bajo, medio o alto, realizando medidas objetivas y subjetivas para evaluar qué tan efectivos son, tanto en su implementación en el *hardware* DSP como en *software*. A pesar de las diferencias en los videos que se van a evaluar, el algoritmo que presenta mejores resultados en la medida objetiva es Alpha-Trimmed, incluso con mejores resultados en la implementación en *hardware* que en simulación. En la medida subjetiva el algoritmo más favorecido es el Mejor Vecino, aunque no por mucho margen.

Palabras claves: reducción de ruido, procesamiento de video, secuencia de imágenes, algoritmos de filtrado, DSP.

Abstract

This paper describes the implementation of noise reduction algorithms. These algorithms are tested on videos with different characteristics of low, medium and high movement, measuring objective and subjective measurements to evaluate their effectiveness either in DSP hardware or software implementation. Neither complexity nor easiness of implementation in real time is evaluated. In this case, when the algorithm increases its complexity it becomes more efficient in removing noise.

Keywords: Algorithms, DSP, FPGA, noise reduction, images sequence, video, video processing.

INTRODUCCIÓN

Durante las últimas décadas el procesamiento de imágenes ha tenido desarrollo para el aprovechamiento de datos multimedia, tanto en aplicaciones tradicionales como la televisión o en aplicaciones en internet. Su rápida evolución ha exigido el mejoramiento de plataformas de desarrollo en *hardware*, así como el de diversos algoritmos que permitan dar respuesta a los retos de procesamiento de gran cantidad de información que puede obtenerse de una señal de video.

En el caso de las imágenes y su procesamiento, se han utilizado ampliamente en diversas áreas del conocimiento, proporcionando información para toma de decisiones. Existe la tendencia a tratar el video como una secuencia de imágenes, buscando utilizar los recursos disponibles tanto de *hardware* como de *software* de tal manera que se pueda extraer información que permita la generación de señales de control, alertas o simplemente diagnóstico respecto a un proceso específico. Sin embargo, cuando se analiza un conjunto de cuadros o imágenes al mismo tiempo, mucha información acerca de la dinámica, esto es, el movimiento de los objetos en el video, la información estadística contribuye a un mejor procesamiento, la cual no puede reconstruirse a partir de una sola imagen.

El análisis de la señal de video se remonta a la detección de movimiento en sistemas de seguridad, pero actualmente su aplicación se puede observar casi en cualquier área del conocimiento. En diferentes fuentes es posible encontrar infinidad de áreas para la aplicación del procesamiento y análisis de la señal de video, entre las que es factible mencionar la videotelefonía, la radiología, el área médica en sus diversas especialidades, la sismología, la videoconferencia, la oceanografía. Es claro que la cantidad, naturaleza y tipo de información obtenida se relacionan directamente con la aplicación [1].

Una vez que la señal de video se obtiene de la fuente, ésta debe procesarse con el fin de incrementar la calidad de la señal de video y de mejorar el desempeño de las tareas subsecuentes. La señal de video requiere *software* y *hardware* especializado y eficiente para su proceso debido a que la señal de video es multidimensional, pues tiene las dos dimensiones espaciales, dependientes de la calidad de la fuente, y una temporal, haciendo uso extensivo de recursos como el ancho de banda o de almacenamiento [2].

El procesamiento de la señal de video es necesario, independientemente de la calidad de la cámara utilizada, pues en el momento de la adquisición toda señal de video sufre degradación a diferentes escalas. Dependiendo de la aplicación, el ancho de banda es un parámetro clave que afecta su transmisión y la capacidad de almacenamiento, dado que las técnicas usadas actualmente requieren en muchos casos la compresión del video, introduciendo diferentes elementos no deseados en la señal que se consideran ruido y que deberán eliminarse o por lo menos reducirse en el video procesado. El procesamiento para remover el ruido también puede incluir información acerca del movimiento de los objetos, y ésta puede utilizarse en posteriores procesamientos.

Durante las últimas décadas se han desarrollado varias arquitecturas de *hardware* especializado junto con tecnologías que suplen, cada vez con mayor exactitud y rapidez, los extensos requisitos de este tipo de procesamiento [3]. Entre estos dispositivos se destacan, por su alto desempeño en la ejecución de algoritmos, las FPGA (*Field Programmable Gate Array*), que poseen la característica de ser reconfigurables en *hardware* para soportar las aplicaciones propuestas por los usuarios, y los DSP (*Digital Signal Processor*), que se pueden programar en lenguajes de alto nivel, como C++.

El proyecto trabaja con un DSP TMDSEV643, puesto que es una alternativa adaptable y suficiente para los requerimientos de diseño presentados.

El costo computacional del procesamiento de una señal de video está determinada en primera instancia por el tipo de tarea que se va a desarrollar, el tipo de procesador que realiza la tarea, la cantidad de datos procesados, además de la eficiencia de los algoritmos desarrollados [4].

Con el fin de simplificar la tarea de programación en el DSP, se hace una revisión de la bibliografía existente para estudiar cuáles algoritmos son los más prometedores en la reducción del ruido. Posteriormente se hace su simulación, para lo cual se elige un *software* que proporcione las herramientas suficientes de desarrollo, que maneje lenguaje de alto nivel, que permita optimizar el costo computacional y que sea capaz de establecer la comunicación con el procesador deseado; por tales razones, se selecciona el *software* de Simulink para trabajar.

La organización del artículo es la siguiente: primero se presentan los conceptos generales del video digital

y del ruido asociado a él; posteriormente, se describen las características que se tomaron en cuenta para la selección de los algoritmos de procesamiento de video presentados y su descripción; después se presentan los resultados y la discusión de la evaluación de desempeño hecha a los algoritmos seleccionados en su implementación en el *hardware* DSP TMDSEV643 y en el *software* Simulink. Finalmente, se hacen las conclusiones del estudio.

INFORMACIÓN BÁSICA SOBRE EL VIDEO DIGITAL

Conceptos generales del video digital

El video (V) usado como entrada para los algoritmos de reducción de ruido está compuesto por imágenes I espaciadas temporalmente por un delta de tiempo t que depende del formato del dispositivo de adquisición utilizado:

$$V = [I_0, I_1, \dots, I_{t-1}, I_t, I_{t+1}] \quad (1)$$

Cada píxel que forma parte de la imagen en escala de grises (I) del video V se define como una función bidimensional de la siguiente manera:

$$\text{píxel de } I_t = p(I_t) = f(x,y) \quad (2)$$

Donde f corresponde a una función bidimensional que define el valor del píxel y sus coordenadas en una imagen I_t que forma parte del video [5].

Tipos de ruido en imágenes y video digital

Es sabido que uno de los tipos de ruido que más afectan a las imágenes es el ruido gaussiano, presente en sensores o circuitos electrónicos defectuosos durante el proceso de adquisición, transmisión y almacenamiento. De igual manera, cuando existe una conmutación en los dispositivos de captura de imágenes se hace presente el ruido tipo bipolar o unipolar, comúnmente llamado ruido sal y pimienta.

Por lo anterior, el ruido tipo gaussiano y el ruido tipo sal y pimienta son utilizados en la práctica para el modelamiento matemático del video [6].

EL MÉTODO

Aspectos que hay que evaluar y selección de los algoritmos para reducción de ruido

Uno de los criterios de selección de algoritmos de procesamiento más prometedores para reducción de ruido en video consiste en implementar filtros que reduzcan el ruido tipo gaussiano de manera óptima como el filtro promedio, o como el filtro de mediana en el caso del ruido sal y pimienta, pero que favorezcan la conservación de información original en el video [7].

Sin embargo, al aplicar al video un filtro promedio o un filtro de mediana, que se han concebido inicialmente para el tratamiento de imágenes, se corre el riesgo de suponer que las imágenes que componen el video son temporalmente independientes, llevando a un procesamiento con resultados no óptimos. Es así como la correlación temporal entre las imágenes del video debe tomarse en cuenta a la hora de proponer buenos algoritmos de procesamiento de video. Los algoritmos de promedio temporal y mediana temporal son una variación de los filtros anteriormente descritos, que considera la dependencia temporal de las imágenes que conforman el video.

Otra característica importante que hace que el procesamiento de video difiera del tratamiento de imágenes consiste en la necesidad de conservar sus elementos dinámicos presentes en el movimiento. De esta manera, el procesamiento de video requiere algoritmos que incluyan adaptabilidad de filtros y sus parámetros, lo que permite eliminar ruido y preservar la información relativa al movimiento del video. De aquí la selección de algunos algoritmos ya propuestos por algunos autores, como el del mejor vecino, Alpha-Trimmed, y el algoritmo de CCD, y en algunos casos, la generación de nuevas propuestas a través de la combinación de las mejores características de cada uno de ellos, como es el caso del algoritmo del mejor vecino y mediana temporal, y finalmente, CCD y promedio temporal [8].

Descripción de los algoritmos seleccionados e implementados

A continuación se describen los algoritmos seleccionados para la reducción del ruido que afecta al video V . *Algoritmo de promedio temporal.* Para este tipo de procesamiento se utilizan los valores de los píxeles que se

encuentran en la misma posición, tanto en la imagen actual I_t como en la imagen anterior I_{t-1} , y la imagen siguiente I_{t+1} del video (figura 1).

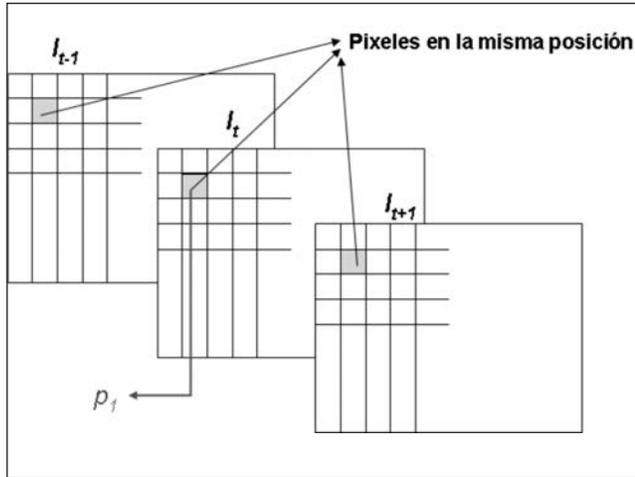


Figura 1. Píxeles en la misma posición en tres imágenes consecutivas del video.

El píxel procesado p_1 que se encuentra en la imagen actual I_t adquiere un nuevo valor p_{1n} , que corresponde al promedio de éste con los píxeles correspondientes a la misma posición, pero en las imágenes anterior I_{t-1} y siguiente I_{t+1} :

$$P_{1n} = \frac{1}{3} [P_{1(t-1)} + P_{1(t)} + P_{1(t+1)}] \quad (3)$$

Si se incrementa el número de imágenes para el cálculo de la anterior ecuación, se obtiene una mayor reducción de ruido en el video, puesto que se toma un mayor número de píxeles para el promedio con el que se calcula el nuevo valor del píxel procesado. Sin embargo, esto conduce a un efecto borroso debido a los elementos dinámicos del video.

Algoritmo de mediana temporal. Este algoritmo utiliza los 27 valores de los píxeles compuestos por las máscaras 3×3 de tres imágenes consecutivas del video denominadas imagen anterior I_{t-1} , actual I_t y siguiente I_{t+1} (figura 2).

Para calcular el nuevo valor $p_{nt}(x, y)$ del píxel central de la máscara 3×3 en la imagen actual I_t , se recurre a la ecuación 4:

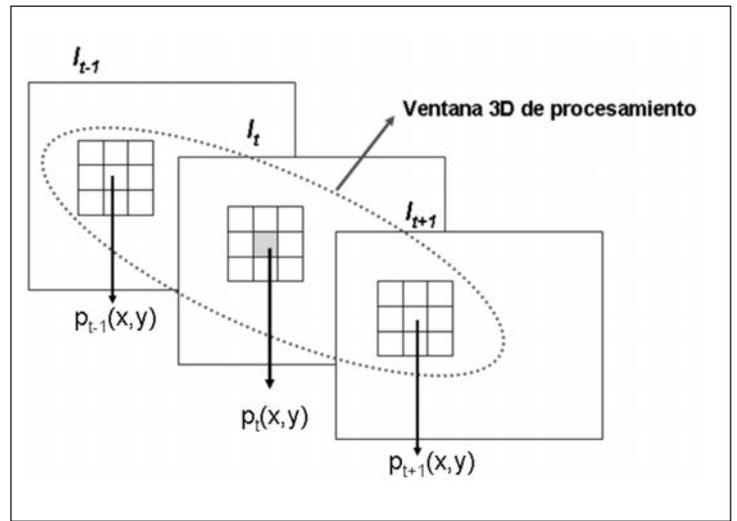


Figura 2. Ventana de procesamiento tridimensional en el video.

$$P_{nt}(x,y) = \text{mediana} [f_{(-1,-1)}(I_t), f_{(1,-1)}(I_t), f_{(-1,1)}(I_t), f_{(0,0)}(I_t), p_t(x,y)]$$

Donde

$$f_{(i,j)}(I_t) = \frac{1}{3} * [p_{t-1}(x-i, y-j) + p_t(x, y) + p_{t+1}(x+i, y+j)]$$

(4)

Como se observa en la ecuación 4, el cálculo del píxel procesado $p_{nt}(x,y)$ toma en cuenta la trayectoria de movimiento de un elemento dinámico en el video. Así, la posición de un elemento en movimiento en la imagen anterior es diferente en la imagen actual y en la siguiente del video, y su variación sigue una curva claramente definida. Esta característica del algoritmo de mediana temporal permite reducir ruido sin afectar la información de movimiento del video [8].

Algoritmo del mejor vecino. En este algoritmo también se utilizan los 27 valores de los píxeles de las máscaras 3×3 de las tres imágenes consecutivas del video, que conforman una ventana tridimensional de procesamiento (figura 3).

Los píxeles son agrupados en un vector ordenado X:

$$X = [p_1, p_2, \dots, p_n] \quad (5)$$

Donde p_1 corresponde al píxel central de la máscara 3×3 ubicada en la imagen actual I_t (figura 3). El criterio de ordenamiento del vector es:

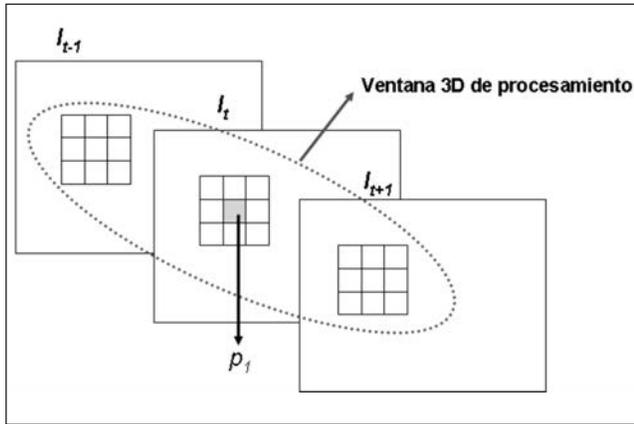


Figura 3. Ventana de procesamiento tridimensional en el video.

$$|p_i - p_i| \leq |p_i - p_j| \quad (6)$$

Donde $i = 2 \dots N$, $j = i \dots N$, siendo N el número total de los píxeles que se utilizan para el procesamiento (los que se abarcan con las tres máscaras) y que en este caso en particular es igual a 27.

El píxel p_i es procesado y se le asigna un nuevo valor denominado p_{in} :

$$P_{in} = \frac{1}{M} \sum_{i=1}^N P_i \quad (7)$$

En un caso en particular, si el parámetro M de la ecuación 7 se iguala al valor del parámetro N , el algoritmo se reduce a un filtro promedio temporal aplicado a la ventana tridimensional.

En otro caso en particular, si el parámetro M es igual a uno, el algoritmo no cambia el valor del píxel procesado, es decir, p_{in} es igual a p_i [9].

Algoritmo Alpha-Trimmed. Al igual que el algoritmo anterior, éste utiliza los valores de los 27 píxeles ubicados en la ventana tridimensional formada por máscaras 3×3 en tres imágenes consecutivas del video (figura 3).

Un vector ordenado Y contiene los valores de los píxeles mencionados:

$$Y = [p_1, p_2, \dots, p_n] \quad (8)$$

Donde p_1 corresponde al píxel central de la máscara ubicada en la imagen actual I_t . Los píxeles p_2 hasta p_n son ordenados de la siguiente manera:

$$p_1 > p_2 \dots > p_n \quad (9)$$

Al píxel procesado p_1 se le asigna un nuevo valor p_{in} , que depende de la siguiente ecuación:

$$P_{in} = \frac{1}{\theta} \sum_{i=\text{floor}(\alpha N)+1}^{N-\text{floor}(\alpha N)} P_i$$

$$\theta = N - 2 * \text{floor}(\alpha N) \quad (10)$$

Cuando el parámetro α es igual a 0,5, el algoritmo se convierte en una variación del filtro de mediana temporal. Por otro lado, si α es igual a 0 el algoritmo pasa a ser un filtro promedio temporal aplicado a la ventana tridimensional [9].

Algoritmo del mejor vecino y mediana temporal. Este filtro de reducción de ruido es una combinación del algoritmo del mejor vecino y el algoritmo de mediana temporal. Inicialmente, este algoritmo realiza una evaluación que define si un cambio notable a lo largo del video corresponde a algún tipo de ruido, o si, por el contrario, corresponde al movimiento de un elemento dinámico en el video. El criterio de decisión se basa en la sustracción entre la imagen actual I_t del video y la imagen inmediatamente anterior I_{t-1} . El resultado de la anterior sustracción consiste en una imagen donde se elimina la información constante o estática de las imágenes I_t e I_{t-1} , conservando la información dinámica de los elementos en movimiento reflejada en esas imágenes [10]. A continuación se presenta un diagrama explicativo de la sustracción descrita anteriormente.

A continuación se presenta un diagrama de flujo que describe las etapas principales del algoritmo (figura 5).

Como se describe en el diagrama anterior, el histograma de la imagen resultado del valor absoluto de la resta de I_t e I_{t-1} se utiliza para calcular un umbral. Este cálculo se realiza a través de la detección de los picos del histograma que representan los píxeles estáticos y dinámicos de las imágenes consecutivas. A continuación, los píxeles de la imagen del valor absoluto de la resta mencionada son comparados con el umbral hallado para definir el algoritmo de procesamiento. Así, se aplica el

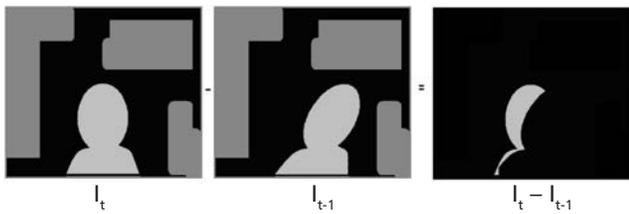


Figura 4. Diagrama del proceso de sustracción entre imágenes consecutivas del video.

algoritmo del mejor vecino cuando se asegura que los cambios que se presentan en las imágenes consecutivas del video se deben a la existencia de ruido. Por otro

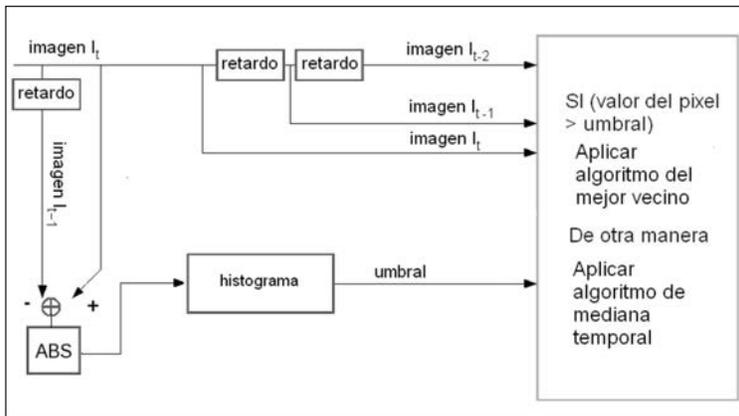


Figura 5. Diagrama de flujo del algoritmo del mejor vecino y mediana temporal.

lado, el algoritmo aplica un filtro de mediana temporal si se determina que los cambios se deben simplemente a movimiento dentro del video [8].

Algoritmo CCD y promedio temporal. Al igual que el algoritmo anterior, se utiliza el valor absoluto de la resta entre imágenes consecutivas del video para estimar el movimiento de sus elementos dinámicos. A renglón seguido se describe el diagrama de flujo de este algoritmo (figura 6).

La diferencia principal de este algoritmo con respecto al anterior consiste en la adición de un bloque de decisión, el cual se encarga de determinar si el resultado de la resta entre la imagen anterior y la actual tiene un carácter estacionario o dinámico. La decisión se toma por medio de un análisis estadístico de los píxeles que forman parte de la imagen, resultado del valor absoluto de la resta descrita anteriormente. Después de determi-

nar la decisión, el algoritmo aplica un filtro de promedio temporal si el carácter del resultado es estacionario, es decir, cuando el movimiento a lo largo del video es prácticamente nulo, y por tanto se pueden promediar

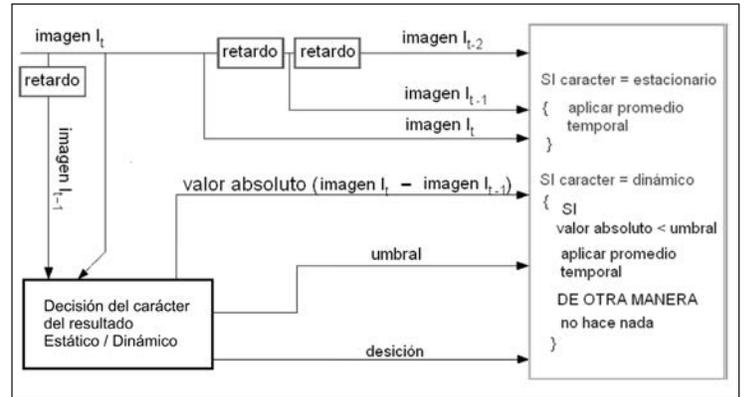


Figura 6. Diagrama de flujo del algoritmo CCD y promedio temporal.

píxeles para eliminar el ruido que pueda afectar el video sin perder información. Si el carácter del resultado es dinámico, se define por medio de un umbral determinado experimentalmente el grado de movimiento que existe en el video para aplicar un filtro de promedio temporal (si el movimiento es escaso), o no procesar y conservar los valores originales de los píxeles si el grado de movimiento es alto, para evitar la pérdida de información del video original [11] [12].

Selección de videos de prueba

Se realiza la evaluación de cinco algoritmos de procesamiento de video basados en los seleccionados y descritos anteriormente. Para este propósito se utilizan tres videos diferentes, a los cuales se les adiciona ruido gaussiano con una media de cero y una desviación estándar de 10:

- Video 1 (ice60.avi): contiene como características principales movimientos rápidos e impredecibles de personas, fondo semidinámico y un rango limitado de niveles de gris. La fuente del video original se encuentra en <ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/>. La resolución espacial que se trabaja es de 144 x 176 píxeles, a una velocidad de 30 fps.

- Video 2 (highway60.avi): contiene como características principales movimientos rápidos y predecibles de líneas de carretera, fondo semiestático y muy pocos niveles de gris. La fuente del video original se encuentra en <http://trace.eas.asu.edu/yuv/>. La resolución espacial que se trabaja es de 144 x 176 píxeles, a una velocidad de 30 fps.
- Video 3 (container60.avi): contiene como características principales un movimiento lento y predecible, fondo estático y una gran diversidad de niveles de gris. La fuente del video original se encuentra en http://meru.cecs.missouri.edu/free_download/videos/. La resolución espacial que se trabaja es de 144 x 176 píxeles, a una velocidad de 30 fps.

La selección de estos videos se basa en la necesidad de probar el desempeño de los algoritmos que hay que evaluar, en muestras que difieren en ciertas particularidades como su cantidad de niveles de gris, su fondo y sus elementos dinámicos.

Adicionalmente, existen dos tipos de resultados en la evaluación de los algoritmos: los cuantitativos, que utilizan como medida principal la relación señal a ruido pico o PSNR, y los cualitativos, que están definidos por un valor numérico asignado por un observador.

El PSNR, medida expresada en decibeles, puede ser útil en cierto grado para comparar y evaluar los resultados de reconstrucción del video original. Un valor alto del PSNR indica una alta precisión de reconstrucción, es decir, que cuanto mayor sea el PSNR del video procesado y el original, mayor será la similitud entre los dos. De acuerdo con la bibliografía, se considera que un valor entre 30 y 40 dB indica buena calidad en una imagen [13].

RESULTADOS Y DISCUSIÓN

Seguidamente se presentan los resultados cuantitativos del PSNR del video filtrado con respecto al original, en la implementación en el *hardware* DSP TMDSEV643.

En la tabla anterior se observa que para el video ice60.avi el algoritmo que mejor lo filtró fue el Alpha-Trimmed, seguido del promedio temporal. Para el video highway60.avi, el mejor algoritmo de procesamiento fue el Alpha-Trimmed, seguido del mejor vecino. Para el video container60.avi, el mejor algoritmo

de filtraje fue el Alpha-Trimmed, seguido del algoritmo de mejor vecino y mediana temporal.

La evaluación cualitativa se hace a través del criterio de un observador o persona completamente ajena al proyecto. Ésta consiste en una valoración entre 1 y 5,

Tabla 1
PSNR del video procesado y el video original en la implementación en el *hardware* DSP

Tipo de Algoritmo	Video	PSNR Filtrado - Original
Algoritmo de promedio temporal	ice60.avi	21,38 dB
	highway60.avi	16,37 dB
	60.avi	16,9 dB
Algoritmo del mejor vecino y mediana temporal	ice60.avi	20,3 dB
	highway60.avi	16,2 dB
	container60.avi	21,71 dB
Algoritmo del mejor vecino	ice60.avi	20 dB
	highway60.avi	17,5 dB
	container60.avi	21,2 dB
Algoritmo Alpha-Trimmed	ice60.avi	21,75 dB
	highway60.avi	17,81 dB
	container60.avi	22,72 dB
Algoritmo CCD y promedio temporal	ice60.avi	19,74 dB
	highway60.avi	15,96 dB
	container60.avi	17,07 dB

siendo 1 la valoración del desempeño del algoritmo más baja y 5 la más alta. Además, el observador tiene en cuenta en su valoración la calidad del video procesado y su definición.

En la tabla 2 se presentan los resultados de la evaluación cualitativa de la implementación de los algoritmos en el *hardware* DSP TMDSEV643.

Según el criterio del observador, el mejor algoritmo para procesar los tres videos fue el algoritmo del mejor vecino, en la implementación en DSP.

En la tabla 3 se presentan los resultados cuantitativos del PSNR del video filtrado con respecto al original, en la implementación en el *software* Simulink.

Se puede observar en la tabla anterior que para el video ice60.avi el algoritmo que mejor funcionó fue el de promedio temporal, seguido del CCD y promedio temporal. Para el video highway60.avi, el mejor algoritmo de procesamiento fue el Alpha-Trimmed, seguido del promedio temporal. Para el video container60.avi el mejor algoritmo de filtraje fue el algoritmo de mejor vecino y mediana temporal, seguido del Alpha-Trimmed.

Tabla 2

Valoración del desempeño de los algoritmos por parte de un observador en la implementación en el *hardware* DSP

Algoritmo	Video	Observador Valoración
Algoritmo de promedio temporal	ice60.avi	3
	highway60.avi	3
	container60.avi	3
Algoritmo del mejor vecino y mediana temporal	ice60.avi	4
	highway60.avi	4
	container60.avi	4
Algoritmo del mejor vecino	ice60.avi	5
	highway60.avi	5
	container60.avi	5
Algoritmo Alpha-Trimmed	ice60.avi	4
	highway60.avi	4
	container60.avi	4
Algoritmo CCD y promedio temporal	ice60.avi	4
	highway60.avi	3
	container60.avi	3

En la tabla 4 se presentan los resultados de la valoración del desempeño de los algoritmos por parte del observador, en la implementación en el *software* Simulink.

Tabla 3

PSNR del video procesado y el video original en la implementación en el *software* Simulink

Algoritmo	Video	PSNR Filtrada - Original
Algoritmo de promedio temporal	ice60.avi	21,41 dB
	highway60.avi	16,35 dB
	container60.avi	16,87 dB
Algoritmo del mejor vecino y mediana temporal	ice60.avi	19,57 dB
	highway60.avi	15,6 dB
	container60.avi	21,51 dB
Algoritmo del mejor vecino	ice60.avi	17,39 dB
	highway60.avi	15,36 dB
	container60.avi	18,42 dB
Algoritmo Alpha-Trimmed	ice60.avi	19,45 dB
	highway60.avi	17,4 dB
	container60.avi	21,4 dB
Algoritmo CCD y promedio temporal	ice60.avi	20,02 dB
	highway60.avi	16,01 dB
	container60.avi	17,1 dB

En la tabla 4 se observa que, según el criterio del observador, el mejor algoritmo para procesar los tres videos en el *software* Simulink fue el algoritmo del mejor vecino. Así mismo, de acuerdo con esta evaluación, los algoritmos de promedio temporal y Alpha-Trimmed

tienen un desempeño aceptable, pero más bajo que los otros algoritmos estudiados.

Tabla 4

Valoración del desempeño de los algoritmos por parte de un observador en la implementación en el *software* Simulink

Algoritmo	Video	Observador Valoración
Algoritmo de promedio temporal	ice60.avi	3
	highway60.avi	3
	container60.avi	3
Algoritmo del mejor vecino y mediana temporal	ice60.avi	4
	highway60.avi	4
	container60.avi	4
Algoritmo del mejor vecino	ice60.avi	5
	highway60.avi	5
	container60.avi	5
Algoritmo Alpha-Trimmed	ice60.avi	3
	highway60.avi	3
	container60.avi	3
Algoritmo CCD y promedio temporal	ice60.avi	4
	highway60.avi	3
	container60.avi	3

En las figuras 7(a), 8(a) y 9(a), es posible observar un cuadro que forma parte del video de prueba ice60.avi. En las figuras 7(b) y 8(b) se muestra el procesamiento del cuadro por medio de la aplicación del algoritmo de promedio temporal y del algoritmo Alpha-Trimmed. El resultado es una disminución significativa del ruido que afecta a la imagen, creando sin embargo un efecto borroso en los elementos en movimiento del video, que se percibe en mayor medida en el resultado de aplicar el algoritmo de promedio temporal.

Por otro lado, en la figura 9(b) se muestra la aplicación del algoritmo del mejor vecino, considerado el mejor de acuerdo con la evaluación cualitativa. Es posible observar que el algoritmo disminuye casi en su totalidad el ruido que afecta a la imagen, y que los elementos en movimiento del video y otros detalles, como los bordes de las regiones, se conservan en la imagen procesada con dicho filtro.

CONCLUSIONES

Según los resultados cuantitativos proporcionados, el mejor algoritmo implementado en el *hardware* DSP para

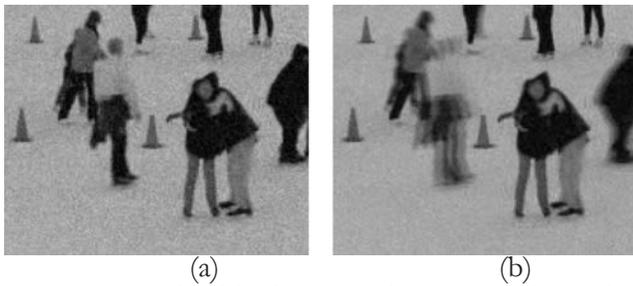


Figura 7(a). Cuadro del video de prueba ice60.avi. (b) Cuadro procesado con el algoritmo de promedio temporal.



Figura 8(a). Cuadro del video de prueba ice60.avi. (b) Cuadro procesado con el algoritmo Alpha-Trimmed.

el procesamiento de los tres videos con características diferentes fue el Alpha-Trimmed. Este algoritmo es una opción aceptable en los resultados con respecto a la implementación en el *software* Simulink, debido a que tiene la ventaja de contar con parámetros que pueden variar y acoplarse a las características de un video, logrando una reducción considerable de su ruido.



Figura 9(a). Cuadro del video de prueba ice60.avi. (b) Cuadro procesado con el algoritmo del mejor vecino.

Según los resultados cualitativos proporcionados, el algoritmo del mejor vecino se seleccionó como el más adecuado para mejorar la calidad y la definición de los tres videos de características diferentes. Aunque se esperaría que los resultados cuantitativos y cualitativos fueran parecidos, se debe tener en cuenta que el PSNR es una medida objetiva de similitud entre el video procesado y el original, medición que es mucho más difícil de determinar con el ojo de un observador.

REFERENCIAS

- [1] Quinell, Richard (2009, September). Smarter video analysis Techniques mine more data. *Electronic Design*, Cover Feature.
- [2] Bovik, A. (2009). *The essential guide of video processing*, 2nd ed. Academic Press.
- [3] Hines, G., Rahman, Z., Jobson, D. & Woodell, G. DSP implementation of the Retinex Image Enhancement Algorithm.
- [4] Tsuen, M., Kasturi, R. & Sivasubramaniam, A. (2003). A pipeline-based approach for scheduling video processing algorithms on NOW. *IEEE transactions on parallel and distributed systems*, vol. 14, No. 2.
- [5] Gonzalez, R., Woods, R. & Steven, L. (2004). *Eddins digital image processing using Matlab*, Prentice Hall.
- [6] Gonzalez, R. & Woods, R. (2002). *Digital image processing*, 2nd ed. Prentice Hall.
- [7] Pratt, William K. (2001). *Digital image processing: PIKS Inside*, 3rd ed. John Wiley & Sons, Inc.
- [8] Bovik, A. (2000). *Handbook of image and video processing*, Academic Press.
- [9] Zlokolica, V., Philips, W. & Van De Ville, D. (2002). Robust non-linear filtering for video processing. Gent, Bélgica: Ghent University.
- [10] Espinosa-Duró, V., Faúndez-Zanuy, M. & Ortega, J.A. (2004). *Face detection from a video camera image sequence*. Polytechnic University of Catalonia.
- [11] Chen, T-H., Lin, Z-H., Chen, C-H. & Kao, C-L. (2007). A fast video noise reduction method by using object-based temporal filtering, Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IHHMSP.
- [12] Hakan, N., Bengt, O. & Youshi, X. Spatio-Temporal Noise Reduction Asic for Real-Time Video Processing, ITE Mid Sweden University.
- [13] Kanjanawanishkul, K. & Uyyanonvara, B. (2003). Fast Adaptive Algorithm for Time-Critical Color Quantization Application, Proc. *Vllth Digital Image Computing: Techniques and Applications*. Sun C., Talbot H., Ourselin S. and Adriaansen T. (eds.), 10-12 Dec. Sydney.