

**IMPLEMENTACIÓN DEL ESTÁNDAR DE COMPRESIÓN DE VIDEO
H.264, UNA PROPUESTA AL APROVECHAMIENTO ÓPTIMO DEL
CANAL DE COMUNICACIÓN.**

Ing. Ángel Luis Zuriarraín Sosa¹

*1. Universidad de Matanzas “Camilo Cienfuegos”, Vía Blanca
Km.3, Matanzas, Cuba.*

Resumen.

En el presente trabajo se presenta una propuesta de solución al aprovechamiento del ancho de banda a través de la utilización de técnicas de compresión de la información, utilizando el estándar H.264. Para la realización de esta propuesta se realizó un estudio teórico detallado del formato de compresión de video H.264 que se apoya en las principales etapas de este códec: predicción, transformación, cuantificación, codificación y el diseño de filtros. Por lo que se garantiza la incidencia directa en la transmisión de información mínima necesaria y la reconstrucción de la imagen en el receptor.

La implementación y puesta en práctica se realiza en Matlab, desarrollando un software que permite el procesamiento de imágenes y flujos de videos de forma eficiente. Se muestra la capacidad de reconstrucción de la imagen o flujo de video por el receptor, demostrándose la versatilidad de esta técnica para el uso más eficiente del ancho de banda.

Palabras claves: Imagen; Ancho de banda; Eficiencia.

Introducción

En la actualidad existe una tendencia acelerada hacia la digitalización de la información en sentido general, de lo cual las imágenes, la transmisión de video y la televisión no han sido excepción. La introducción de la imagen digital constituyó un salto trascendental en el procesamiento de imágenes pues fue necesario implementar nuevas técnicas de codificación, filtrado e incluso se trabaja con el concepto: cuantificación, propio de la información digital.

El estándar H.264, es una norma que define un códec de video de alta compresión, capaz de proporcionar una buena calidad de imagen con bitrate notablemente inferiores a los estándares previos. Es una realidad, que estos estándares se utilizan cada vez con mayor frecuencia en aplicaciones que precisan de rapidez y a la vez poca pérdida de calidad de imagen.

La flexibilidad del códec H.264, ha permitido su utilización en distintas aplicaciones como el DVD de alta definición, (Blu-ray entre otros), la difusión de vídeo digital, desde streaming a la TV de alta definición, el almacenamiento de vídeo en línea (como YouTube), la telefonía móvil 3G y en software como QuickTime, Flash y el sistema operativo de Apple, MacOS X, así como en consolas de videojuegos como PlayStation 3.

Aunque el códec se considera como un algoritmo, se debe tener en cuenta que realmente es el resultado de varios procesos o algoritmos: transformaciones, predicciones, codificación y filtrados. Entre el gran número de herramientas que existen, escoger Matlab para desarrollar este proyecto no es casual. Matlab contiene herramientas destinadas al procesamiento de señales e imágenes, y cuenta con un ambiente amistoso para el trabajo con arreglos y matrices.

En el presente trabajo se muestra la implementación de los algoritmos que componen el códec H.264, en Matlab. Aparecen las imágenes obtenidas después de ser ejecutados los

algoritmos desarrollados y se analizan en cuanto a los parámetros definidos por el códec previamente.

Desarrollo

1. Herramientas para el procesamiento de imágenes

Matlab ofrece un grupo de herramientas útiles para el procesamiento de imágenes. Este grupo de herramientas soporta un amplio rango de operaciones en el procesamiento de imágenes, el cual incluye:

- ✓ Transformaciones espaciales en las imágenes.
- ✓ Operaciones morfológicas.
- ✓ Operaciones de bloques.
- ✓ Filtros lineales y diseño de filtros.
- ✓ Transformadas.
- ✓ Análisis y realce de imágenes.
- ✓ Registro de imágenes.
- ✓ Restricción de la región de operación.

Es importante resaltar que estas herramientas ofrecidas por Matlab pueden ser ampliadas al usar archivos creados por el usuario o al combinar varias herramientas.

2. Herramientas para el procesamiento de señales

Las herramientas para el procesamiento de señales soportan un amplio rango de operaciones. Entre las operaciones que soporta Matlab se encuentran: el diseño de filtros, la puesta en funcionamiento, el modelado y el análisis espectral de las señales. Las herramientas se dividen en dos categorías: las funciones de línea de comandos y las funciones de usuario gráfico.

3. Funciones de línea de comandos y objetos

Las funciones de línea de comandos y objetos están disponibles en las siguientes categorías:

- ✓ El diseño de filtro discreto, el análisis, y la puesta en práctica.
- ✓ El diseño de filtro de analógico, el análisis, y la puesta en práctica.
- ✓ Transformaciones de sistema lineales.

- ✓ Funciones de marco o ventana.
- ✓ Análisis espectral.
- ✓ Transformadas.
- ✓ Procesamiento estadístico de la señal.
- ✓ Modelado.
- ✓ Pronóstico lineal.
- ✓ Procesamiento de señal de *Multirate*.
- ✓ Generación de señales.

4. Interfaz de usuario gráfico

Es una plataforma de usuario gráfico interactivo en la cual se encuentran disponibles las siguientes aplicaciones:

- ✓ Diseño de filtro y análisis.
- ✓ Diseño de ventana y análisis.

5. Herramientas para el diseño de filtros

Las herramientas utilizadas para el diseño de filtros son consideradas una técnica muy poderosa y avanzada para diseñar, simular, y analizar filtros digitales. Extiende la capacidad del software en el procesamiento de señal con las arquitecturas de filtros y los métodos de diseño para aplicaciones de procesamiento digital de imágenes en tiempo real complicadas, incluyendo filtrado adaptativo, filtrado *multirate* así como las transformaciones de filtro.

6. H.264 empleando Matlab

La implementación del códec H.264 en Matlab se puede dividir en cinco bloques fundamentales: un bloque destinado a la predicción; uno encargado de aplicar la transformada discreta del coseno a la imagen y otro bloque encargado de la cuantificación. Posteriormente se codifica la información para ser transmitida y para poder reconstruir la imagen se implementa el filtrado, encargado de eliminar el efecto de bloques (Ver Figura1).

En la Figura 1 se muestra el diagrama general del códec H.264:

- ✓ La imagen es recibida y tomada como fuente de referencia para aplicar la predicción en la siguiente imagen.
- ✓ Se aplica la transformada discreta del coseno a la imagen, dividiéndola en bloques de 4x4.

- ✓ Posteriormente se pasa a la cuantificación de los coeficientes obtenidos por la DCT.
- ✓ La información cuantificada es codificada entrópicamente, para una mejor transmisión o almacenamiento de la misma.
- ✓ Al ser recibida la información debe ser decodificada, mediante el proceso inverso a la codificación.
- ✓ La información decodificada se encuentra cuantificada, por lo que se debe aplicar el proceso inverso de la cuantificación, para que se le pueda aplicar la transformada del coseno discreto inversa (IDCT).
- ✓ Al aplicar la transformada del coseno discreto inverso (IDCT), la imagen que se recupera presenta cierta degradación en la calidad, pues una parte de la información ha sido descartada o predicha.
- ✓ Con el objetivo de mejorar la calidad de percepción de la imagen y eliminar el efecto de bloques (introducido por la predicción) se pasa al filtrado de la imagen: filtro deblocking.

Es importante aclarar, que en este proyecto se implementará el códec aplicado a una imagen, es decir, a un solo frame. En la Figura 3.1 aparece representada la etapa de predicción con líneas discontinuas, lo cual significa que este bloque no será implementado en Matlab. Cuando se realiza el análisis de un *frame* o imagen no es necesario aplicar técnicas de predicción, la compresión está basada netamente en la DCT y la cuantificación de la información.

Es importante resaltar que en el presente trabajo de diploma propone el algoritmo para implementar la predicción por software, aunque no se implementa por falta de tiempo. Todos los resultados obtenidos en la compresión de imágenes son igualmente válidos para la compresión de video, pues un flujo de video puede ser interpretado como una secuencia de imágenes mostradas cada determinado intervalo de tiempo.

Matlab constituye una herramienta muy útil para el desarrollo de este proyecto:

- ✓ Presenta un ambiente agradable y amistoso para el usuario.
- ✓ Ofrece al programador una herramienta de ayuda poderosa y fácil de acceder.
- ✓ Matlab tiene implementada un grupo de funciones y operaciones matemáticamente complicadas, que son utilizadas en el procesamiento de imágenes.
- ✓ Matlab constituye una de las herramientas más importantes en el trabajo con matrices, de aquí que la imagen sea representada como una matriz y cada coeficiente de esta matriz puede ser interpretada como un pixel de la imagen.
- ✓ Fácil implementación y creación de filtros.

- ✓ La versión Matlab 2010rb incorpora la capacidad de exportar hacia lenguaje ensamblador de algunos FPGA.

Resultado de la discusión.

A continuación se realizará un análisis de la influencia del parámetro Q_p en la compresión de la imagen. Se muestran tres valores de Q_p : $Q_p=0$, $Q_p=26$ y $Q_p=51$. Se analizan estos valores pues $Q_p=0$, $Q_p=51$ son valores críticos y $Q_p=26$ es una media entre los valores críticos.

En la Figura 2 se muestra una secuencia de imágenes obtenidas a partir de los algoritmos anteriormente expuestos para un $Q_p=0$. Al obtener la DCT prima (Figura 1(a)) se aprecian niveles normales de información en la imagen, correspondientes con la cantidad de información típica, después de aplicada la transformada. La Figura 1 (b) muestra el resultado de la transformada con cuantificación. En esta ocasión se aprecia cómo un valor pequeño de Q_p , influye en una menor compresión de la imagen. Se puede observar que la etapa de cuantificación es la de mayor peso en la decisión de la compresión. En la Figura 1 (c) la DCT prima recuperada presenta diferencias notables con respecto a la Figura 1 (a), esto es justificado por la pérdida de información que introduce el proceso de cuantificación.

En la Figura 2 (d) y 2 (e) se muestra la imagen original y la reconstruida respectivamente. A simple vista resulta casi imposible notar diferencia, lo cual se explica por el valor de Q_p (menor valor posible), que incide directamente como parámetro fundamental en la calidad o compresión de la imagen. En la Figura 2 no se muestra la diferencia entre la imagen original y la reconstruida pues es casi imperceptible.

En la Figura 3 se muestra una secuencia de imágenes similar a la mostrada anteriormente, pero con la diferencia de que el valor de Q_p ha sido modificado. En esta ocasión $Q_p=26$, lo cual es un valor intermedio del rango posible (0-51). En la Figura 3 (b) se puede observar como la DCT prima se transformó en una imagen con muy poca información no nula, lo cual demuestra la acción directa del Q_p al ser aumentado. La imagen (c) refleja la obtención de la DCT prima y muestra como la reconstrucción de imágenes a partir de una menor cantidad de información distorsiona la imagen.

En la Figura 3 (e) se muestra la imagen reconstruida, en la cual se aprecian rastros de la compresión, pues al compararla con la imagen original (d), se pueden encontrar algunas diferencias. En la Figura 3 (f) se muestra la diferencia entre la imagen original y la reconstruida. Se puede apreciar que existe realmente una gran cantidad de valores nulos, o casi nulos (color oscuro), por lo que la diferencia es muy pequeña.

Al analizar detenidamente la Figura 3 (f) se observa que la mayor cantidad de incongruencias entre la imagen original y la reconstruida se encuentra en los bordes, resultado que justifica el posterior uso de un filtro. El uso del filtro garantiza la corrección o alisamiento, principalmente en el área de los bordes, como será mostrado posteriormente.

Es importante resaltar el ventajoso compromiso que existe, para este valor de Q_p (26), si se compara la cantidad de información que se reduce con la calidad de la imagen reconstruida.

En la Figura 4 se muestra el conjunto de imágenes obtenidas en los algoritmos descritos por un valor de $Q_p=51$. Al tomar el Q_p máximo para la compresión se pretende ilustrar el compromiso indestructible entre compresión y calidad de la imagen reconstruida. En la Figura 4 (b) se aprecia que la información a transmitir tiene un alto grado de valores nulos, por lo que el resultado de la DCT con cuantificación es una imagen prácticamente oscura en su totalidad.

La DCT prima reconstruida (Figura 4 (c)) difiere notablemente de la DCT prima, lo cual es obvio pues un gran porcentaje de la información fue descartada en el proceso de cuantificación.

Para $Q_p=51$, la imagen reconstruida difiere notablemente de la imagen original e introduce en gran medida distorsión, no solo en los bordes, sino también en el resto de la imagen. Por este motivo, aunque el filtro actuó positivamente sobre la imagen, no se logra una imagen con la calidad mínima necesaria. Por estos motivos se recomienda no usar un valor de Q_p tan elevado. No se justifica el alto nivel de compresión con la mala calidad de la imagen.

El análisis de estos valores de Q_p permite determinar el rango o valor óptimo que este debe tomar. Hasta el momento, un análisis cualitativo descarta los valores extremos: $Q_p=0$ y $Q_p=51$. El menor valor es descartado pues la calidad de la imagen reconstruida no justifica la pobre compresión, apenas un 8%. Sin embargo existen otros valores de Q_p en los cuales no se percibe diferencia y presentan un mayor porcentaje de compresión de la imagen. En el caso de $Q_p=51$ si bien es cierto que se obtienen compresiones del 30% no se justifica la mala calidad de la imagen reconstruida.

La Figura 5 muestra dos gráficos a partir de los cuales se puede realizar un análisis cuantitativo sobre el impacto del parámetro de cuantificación (Q_p) en la compresión de la imagen. El gráfico mostrado en la Figura 5 (a) evidencia el incremento de valores nulos en la matriz que define la imagen comprimida al aumentar el valor de Q_p . Resulta de particular interés el comportamiento de esta curva, por lo que se puede arribar a las siguientes conclusiones:

- ✓ En el rango de Q_p de 0 – 3 la cantidad de valores descartados o truncados en la imagen no reciben un cambio considerable, se comporta prácticamente igual a los efectos de la percepción de la imagen.
- ✓ En el caso del rango que va desde $Q_p=4$ hasta $Q_p=17$ la gráfica muestra un incremento de la pendiente considerable, por lo que se puede afirmar que en este rango de valores, la imagen será más susceptible a cambios en el Q_p .
- ✓ El rango de 18 a 35 constituye el intervalo donde la cantidad de valores desechados o truncados se comportan con mayor estabilidad, prácticamente de forma lineal, lo cual da un rango aconsejable u óptimo a la hora de la elección de un valor de Q_p .
- ✓ Cuando Q_p excede el valor 40, la curva experimenta un decremento de la pendiente colocándose de forma casi paralela a las abscisas, lo cual puede interpretarse como un comportamiento casi constante en la compresión y sin embargo la calidad de la

imagen sí se ve afectada de forma drástica. Por lo que es aconsejable no asignar a Q_p un valor mayor que 40.

En la Figura 6 (b) aparece una gráfica donde se compara el parámetro de calidad “VC” o variable de calidad. Este parámetro es una medida de la degradación en la calidad de la imagen. Se define fundamentalmente por la cantidad de información descartada y que no es posible recuperar, así como por la calidad de percepción de la imagen recuperada. La gráfica mostrada permite arribar a un grupo de conclusiones con respecto al rango de valores que puede tomar Q_p :

- ✓ Para valores de Q_p menores que 18 se aprecia un comportamiento caótico en cuanto a los saltos del parámetro “VC”; no se establece un valor fijo en cuanto a la variación de calidad de la imagen. Se puede considerar un comportamiento inestable que puede sufrir variaciones al cambiar de imágenes.
- ✓ En el caso de $Q_p=18$ hasta $Q_p=35$ el comportamiento de la curva es prácticamente lineal y los intervalos en los que varía la variable de calidad son prácticamente constantes. Dentro de este rango de valores una variación de Q_p progresiva se correspondería con una variación de la calidad de la imagen también progresiva, lográndose un comportamiento lineal entre estas variables.
- ✓ Para valores de Q_p mayores que 40 la gráfica comienza a mostrar una inestabilidad en la relación lineal existente entre el parámetro de calidad y el Q_p en el rango de 18 a 35. La calidad de la imagen comienza a degradarse de forma más notable.

Se puede concluir, al analizar ambas gráficas, que la imagen no mantiene un resultado estable en la compresión para todo el rango de Q_p . Se puede afirmar que el rango de valores aconsejable u óptimo para la elección del Q_p está entre 18 y 35.

Después de reconstruida la imagen es sometida a un filtrado para eliminar el efecto de bloques que se introduce durante la predicción y durante el proceso de reconstrucción de la DCT prima (Ver Figura 7). En esta figura se muestra la DCT prima antes de ser cuantificada (a) y a continuación la DCT prima recuperada (b). En la Figura 7 se puede verificar cómo el proceso de cuantificación influye negativamente en la calidad de la imagen reconstruida.

En las imágenes (c) y (d) de la Figura 6 se muestra un zoom aplicado a cada DCT prima original y recuperada respectivamente. En la imagen (d) se aprecia cómo la carencia de información en la reconstrucción de la imagen provoca el efecto de bloque, pues una parte de la información es supuesta o predicha.

La existencia en mayor o menor medida de los bloques está dada por la cantidad de información que se utilice para la reconstrucción de la imagen; lo cual a su vez está determinado por el parámetro de cuantificación Q_p . El nivel de filtrado también está determinado por el factor Q_p (a partir de Q_p se determinan los umbrales α y β).

En la Figura 7 se muestra la acción del filtro *deblocking* para un valor elevado del Q_p , con el objetivo de ilustrar la acción del filtrado. En las imágenes (c) y (d) se muestra el zoom aplicado a las imágenes (a) y (b) respectivamente. Se puede apreciar a simple vista la acción sobre los bordes realizada por el filtro. El filtrado, al atenuar los cambios y hacerlos menos bruscos, logra crear la sensación de alisado, por lo que elimina los bloques tan contrastados existentes en la imagen a filtrar.

El filtrado, constituye uno de los grandes logros en este proyecto, pues cumple con la función de atenuar los efectos negativos de la compresión y propina un adecuado acabado a la imagen recuperada.

Imágenes

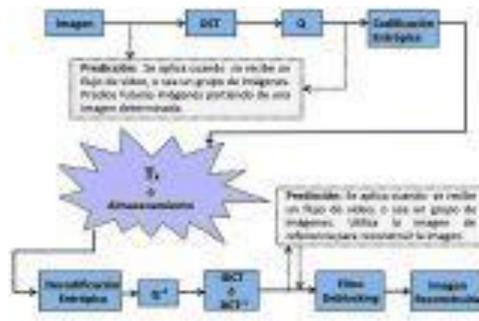


Figura 1 Diagrama del códec H.264.



Figura 2 Para $Q_p=0$: (a)DCT prima. (b)Imagen cuantificada. (c)DCT prima recuperada. (d)Imagen original. (e)Imagen recuperada.

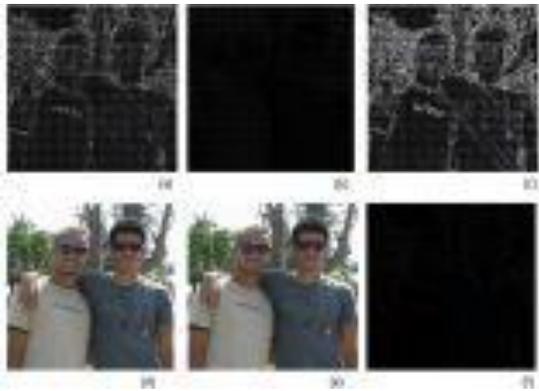


Figura 3 Para $Q_p=26$: (a)DCT prima. (b)Imagen cuantificada. (c)DCT prima recuperada. (d)Imagen original. (e)Imagen recuperada.(f)Diferencia Imagen original e Imagen recuperada.

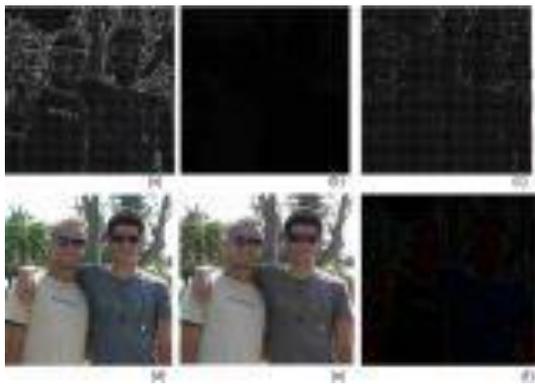


Figura 4 Para $Q_p=51$: (a)DCT prima. (b)Imagen cuantificada. (c)DCT prima recuperada. (d)Imagen original. (e)Imagen recuperada.(f)Diferencia Imagen original e Imagen recuperada.

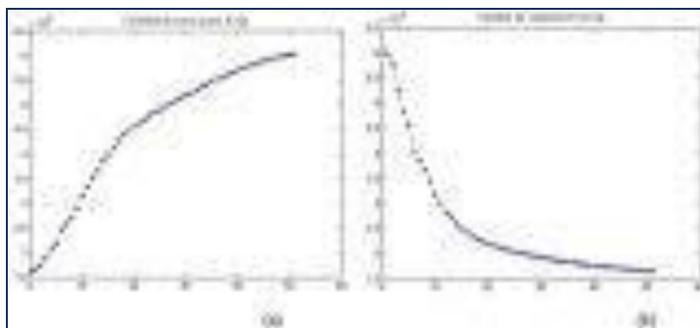


Figura 5 Gráficas de los parámetros del codec.

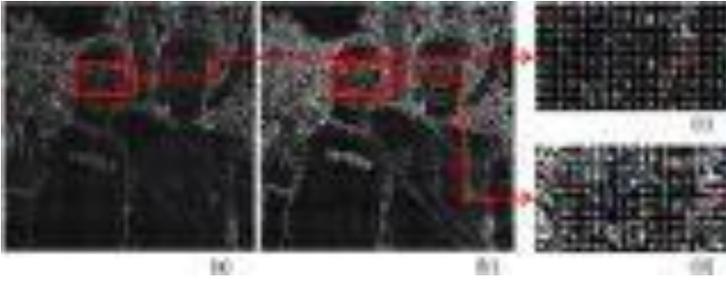


Figura 6 Impacto de la compresión de la DCT recuperada.



Figura 7 Efecto del filtrado.

Conclusiones

Con este trabajo se logró dar cumplimiento al objetivo inicial de estudiar e implementar técnicas de procesamiento digital al utilizar el códec H.264 para lograr un mejor aprovechamiento del canal. Esto se verifica debido a que:

Se estudiaron las bases que sustentan las técnicas de compresión y filtrado aplicadas a las imágenes digitales en escala de grises y en colores, que dentro del procesamiento de imágenes juegan un papel fundamental. Se realizó un estudio comparativo entre las diferentes técnicas de codificación de video, lo cual justifica la implementación del códec H.264 entre los diferentes formatos existentes. Se logró la implementación en Matlab de cada etapa en el proceso de codificación: Transformada Discreta del Coseno, cuantificación, filtro *deblocking* y el inverso de estos procesos. Se consiguió en este proyecto extender la compresión y filtrado a imágenes digitales con información de luma y croma (en colores). Se logró medir y cuantificar el grado de compresión de la imagen codificada, plasmándose estos resultados en tablas y gráficas que muestran el comportamiento de las imágenes al ser variados los parámetros de compresión. Se estableció un parámetro capaz de medir la calidad de la imagen reconstruida y se analizó el impacto del filtro en la misma.

Se elaboró una aplicación gráfica utilizando la herramienta matemática *Matlab* versión 7.9.0 [R2009b], basada en los algoritmos anteriores. Esta aplicación permite al usuario interactuar con los parámetros que determinan el nivel de compresión y filtrado, tratados en el proyecto. En la aplicación se puede observar el impacto que produce, sobre el flujo de video, la variación de los parámetros mencionados.

BIBLIOGRAFÍA

F. T. Ruiz, Sistemas Audiovisuales vol. II, 2000.

B. Jähne, Practical Handbook on IMAGE PROCESSING for SCIENTIFIC and TECHNICAL APPLICATIONS, S E C O N D E D I T I O N ed., 2004.

K. Jack, Video Demystified, a handbook for the digital engineer 2005.

W. K. PRATT, DIGITAL IMAGE PROCESSING, 2001.

R. V. Alonso, "IMPLEMENTACIÓN EN JAVA/C++ DEL ESTÁNDAR DE CODIFICACIÓN H.264," in ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) Madrid: UNIVERSIDAD PONTIFICIA COMILLAS, 2009.

B. Sheng, "AN IMPLEMENTED ARCHITECTURE OF DEBLOCKING FILTER FOR H.264/AVC," Department of Computer Science and Technology, Harbin Institute of Technology, China, 2009.

W.-Y. Wei, "Digital Video Compression Fundamentals and Standards," National Taiwan University Taipei, Taiwan, ROC, 2009.

A. J. Peter List, Jani Lainema, Gisle Bjøntegaard, and Marta Karczewicz, "Adaptive Deblocking Filter," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, vol. VOL. 13, NO. 7, JULY 2003.

C.-H. C. Chung-Ming Chen, "AN EFFICIENT VLSI ARCHITECTURE FOR EDGE FILTERING IN H.264/AVC," Proceedings of the third IASTED International Conference CIRCUITS, SIGNALS, AND SYSTEMS, 2005.