

# Diseño de una arquitectura en FPGA para filtros espaciales en imágenes

Sergio Guzmán-Martínez (1), Carlos Rodríguez-Doñate (2)

1 Ingeniería en Comunicaciones y Electrónica, Universidad de Guanajuato | <sgm\_1210@hotmail.com>

2 Departamento de Estudios Multidisciplinarios, División de Ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato  
| c.rodriguezdonate@ugto.mx

## Resumen

El filtrado espacial en procesamiento digital de imágenes es una de las operaciones más utilizadas para diferentes metodologías de procesamiento de imágenes, por lo cual, la arquitectura que realizara este cálculo debe de ser muy eficiente. En este trabajo se presenta el diseño de una arquitectura en FPGA para realizar el filtrado espacial de imágenes con una máscara de 3x3, la cual se basa en los principios de computo paralelo y pipeline, donde se logró obtener una eficiencia de 200 fps para imágenes VGA. La etapa experimental se realizó con una cámara digital de 5 megapíxeles y el kit de desarrollo DE1-soc de altera, ocupando menos del 1% de la lógica disponible del FPGA Cycone V.

## Abstract

Spatial filtering in digital image processing is one of the most commonly used operations for different image processing methodologies, thus, architecture to conduct this calculation must be very efficient. This paper presents the design of an architecture in FPGA to perform spatial filtering of images with a mask of 3x3, which is based on the principles of parallel computing and pipeline, where he obtained an efficiency of 200 fps for VGA images . The pilot phase was performed with a 5 megapixel digital camera and Development kit Altera DE1-soc, occupying less than 1% of available FPGA logic Cycone V.

## Palabras Clave

Procesamiento digital de imágenes, Arquitectura en FPGA, Procesamiento en tiempo real, VHDL,

## INTRODUCCIÓN

El procesamiento digital de imágenes consiste en mejorar la calidad de la imagen digital para hacer notorios diferentes detalles que se desean resaltar de la imagen, el cual, se puede llevar a cabo por medio de diferentes técnicas de procesamiento, que consisten en modificar los píxeles de la imagen digital.

En la actualidad, el procesamiento digital de imágenes es de gran relevancia para diferentes tipos de aplicaciones, como medicina, robótica, seguridad, entre otras [1]. Por lo cual, en el mundo científico se han desarrollado varios softwares que facilitan la implementación de diferentes algoritmos de procesamiento de imágenes [2], con los cuales se logra obtener un procesamiento en tiempo real, teniendo por objetivo facilitar la interacción con actividades de la vida cotidiana, como es el caso del trabajo presentado en [3], donde se propone el monitoreo en tiempo real del consumo eléctrico de los hogares a partir de las fotografías tomadas a los medidores analógicos. Dentro de las plataformas tecnológicas para lograr obtener procesamiento de imágenes en tiempo real, se han destacado los FPGA (*Field Programmable Gate Array*), porque son capaces de realizar procesamientos de forma paralela, pipeline y trabaja a alta velocidad, además de ser dispositivos de bajo costo, donde los diseños basados en lenguaje de descripción de hardware como es VHDL (*Very High Speed Integrated Circuit*) se puede lograr la portabilidad.

Como se muestra en la literatura [1-5], el procesamiento espacial en imágenes es una de las operaciones más comunes y parte fundamental para diferentes metodologías de procesamiento como detección de bordes, eliminación de ruido, detección de movimiento, entre otras. Este tipo de procesamiento consiste en procesar los píxeles de una imagen por medio de grupos de píxeles (vecindad), que se compone de un pixel central y los que lo rodean, así el valor de los píxeles que rodean al central influyen en el pixel procesado de salida. [4 - 5]. Este tipo de filtrado en imágenes se realiza por medio de la convolución en 2 dimensiones, definida por la ecuación (1):

$$G(p, q) = h(j, i) * f(p - j, q - i) \quad (1)$$

Donde  $h(j, i)$  es conocida como máscara,  $f(p, q)$  representa la imagen que se filtrará y  $G(p, q)$  es la imagen filtrada. Normalmente, la máscara es una matriz cuadrada de tamaño impar formada por un pixel central y sus vecinos (píxeles que lo rodean).

En esta investigación el diseño de la arquitectura se realizó para una máscara de 3X3, que se define por la ecuación (2).

$$h(i, j) = \begin{bmatrix} C_{(1,1)} & C_{(1,2)} & C_{(1,3)} \\ C_{(2,1)} & C_{(2,2)} & C_{(2,3)} \\ C_{(3,1)} & C_{(3,2)} & C_{(3,3)} \end{bmatrix} \quad (2)$$

Donde,  $C_{(i,j)}$  representan los coeficientes de la máscara. El proceso de la convolución de la imagen con la máscara, consiste en multiplicar los coeficientes con una vecindad en la imagen del mismo tamaño que la máscara y que al sumarlos se obtendrá un pixel procesado. El proseo completo de filtrado se realizar mediante el recorrido de la máscara sobre toda la imagen.

En este trabajo se presenta una arquitectura basada en FPGA para el filtrado espacial en imágenes digitales con una máscara de tamaño 3 X 3. Cabe destacar que el diseño realizado se basa en los principios de computo paralelo y pipeline, por lo cual, se logran procesar 200 cuadros por segundo (fps, *frames per secon*) para imágenes VGA, además, la arquitectura es capaz de procesar la información enviada directamente de la cámara digital.

## MATERIALES Y MÉTODOS

Como se mencionó en la sección anterior, en este trabajo se propone una arquitectura hardware basada en FPGA, para el filtrado espacial en imágenes, donde se utilizó una cámara digital de 5 megapíxeles, un módulo LCD de 7" con una resolución de 800 X 480 con 24 bits en color y el kit DE1-Soc de Altera, el cual cuenta con el FPGA Cyclone V y una memoria SDRAM (*synchronous dynamic random-access memory*) de 64 Mb, la cual se muestra en la Imagen 1.

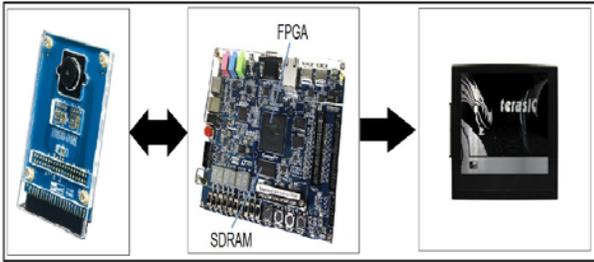


IMAGEN 1: Esquema general de la metodología propuesta.

### Diseño de arquitectura en hardware.

El diseño de la arquitectura para el filtrado espacial de imágenes se muestra en la Imagen 2 que representa el procesamiento, y consta de 3 etapas: ordena y almacena renglones, filtrado espacial, truncado.

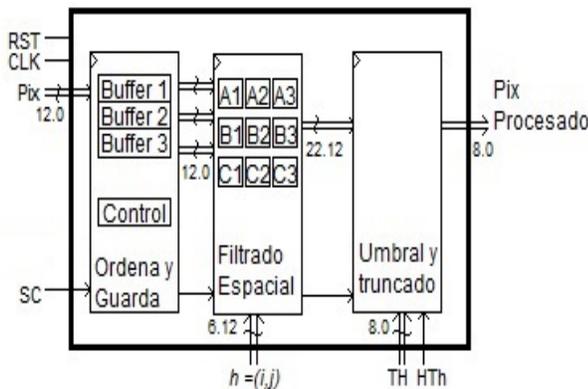


IMAGEN 2: Esquema de las etapas de la arquitectura de procesamiento diseñada.

La primera etapa se compone de tres buffers de 1024 direcciones construido por bloques RAM embebidos del FPGA, esta etapa recibe los píxeles (pix) enviados por el sensor digital (validos mientras SC este activa) y los almacenará en buffers. Al recibir tres renglones, se procede a enviar cada ciclo de reloj los píxeles de los tres buffers de forma paralela y al concluir con él envío de la información, el nuevo renglón enviado por el sensor se sobre escribirá en el primer buffer, ya que no será necesario para la segunda etapa. En la etapa de filtrado se realiza la operación de convolución, por lo cual se almacenan los nueve píxeles de la vecindad (renglón 1: A1, A2, A3;

renglón 2: B1, B2, B3; 3: renglón 2: C1, C2, C3) y se multiplican con los coeficientes de la máscara de forma paralela, y la serie de sumas que concluirán con el pixel procesado, se realizan por etapas pipeline logra obtener un dato procesado por ciclo de reloj. Las operaciones de esta etapa se realizan en punto fijo, donde los coeficientes tienen un formato de 6.12 (6 bits parte entera y 12 bits parte decimal), y los píxeles de entrada con formato 12.0. La tercera etapa, consiste en el truncamiento del resultado de la convolución para realizar un ajuste a 8 bits y si se requiere un nivel de umbral (Th) a la salida del filtro, la señal Hth lo activará con un nivel alto. Para procesar la información de las tres componentes enviadas por el sensor (RGB), solo se duplica la etapa de procesamiento como se muestra en la imagen 3.

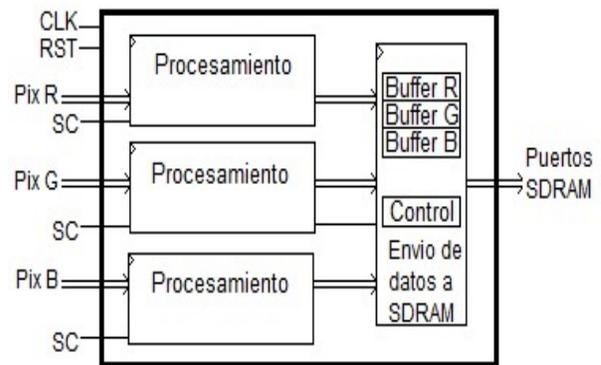


IMAGEN 3: Esquema general de la arquitectura diseñada del procesamiento y envío de datos a SDRAM.

Como se puede observar en la imagen 3, la información procesada se envía a una última etapa que se encarga de almacenar la información en la SDRAM. Para optimizar y facilitar el acceso a la memoria, en cada dirección de la SDRAM se almacenan dos píxeles por cada componente, donde tres direcciones de la memoria tendrán la información de dos píxeles procesador, almacenando primero dos píxeles de la componente R, después dos de la componente G y finalmente dos de la componente B, este proceso se repetirá para almacenar un renglón completo de la imagen. De esta forma el LCD puede acceder a la memoria de forma eficiente.

## RESULTADOS Y DISCUSIÓN

Entre los resultados destacables de este trabajo, se encuentra el consumo de recursos del FPGA Cyclone V, ya que se utilizaron 412 bloques lógicos que representa el 1 %, 98304 bits de memoria embebida correspondientes al 2% y 15 multiplicadores de 18 X 18 bits que es el 17% del dispositivo.

Referente al tiempo de procesamiento, en la Tabla 1 se presenta la eficiencia en fps para diferentes tamaños de imágenes, logrando superar el estándar de procesamiento en tiempo real que es de 30 fps. Esta información se obtuvo utilizando una frecuencia de 96 MHz que corresponde a la máxima frecuencia de operación de la cámara utilizada.

Tabla 1: Eficiencia de la arquitectura para diferentes tamaños de imágenes.

Tamaño de imagen	Cuadros por segundo (fps)
1. 256 X 256	872
2. 640 X 480	200
3. 800 X 480	164
4. 800 X 600	131
5. 1024 X 1024	60

Finalmente, la imagen 4 muestra los resultados de una imagen procesada por la arquitectura diseñada, donde la máscara que se utilizó fue la de Sobel Horizontal y Sobel Vertical, que se definen por la ecuación (3) y (4), respectivamente. En la Imagen 4 a) Se muestra la imagen a procesar adquirida por el sensor.

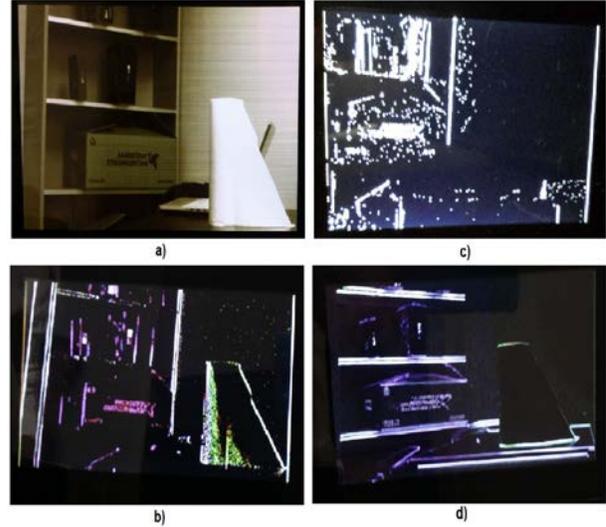


IMAGEN 4: Resultados experimentales, a) Imagen original, b) Bordes horizontales (RGB), c) Bordes horizontales con un solo canal de procesamiento, d) Bordes verticales (RGB).

La imagen 4 b) muestra la imagen filtrada para detección de bordes horizontales con el operador Sobel, con un nivel de umbral de 100; Imagen 4 d) presenta una imagen filtrada para detección de bordes verticales usando el operador Sobel, con un nivel de umbral de 100. Finalmente si se desea procesar una imagen en escala de grises solo se deberá usar un canal de procesamiento, la cual se presenta en la imagen 4 c) donde el nivel de umbral es de 32.

Operador Sobel para bordes horizontales:

$$H_{SH}(i,j) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (3)$$

Operador Sobel para bordes verticales:

$$H_{SV}(i,j) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (4)$$

## CONCLUSIONES

Las pruebas realizadas para la unidad de procesamiento se realizaron con el sistema de captura de imágenes y desplegado con la pantalla LCD, donde se configuró la etapa de procesamiento para detectar bordes con el operador Sobel, logrando obtener un procesamiento en tiempo real de 164 fps para una imagen de 800 X 480. Por esta razón, la unidad de procesamiento puede ser aplicado para algoritmos que utilicen este tipo de filtrado y que requieran de un procesamiento en tiempo real, como es el caso de estimación de movimiento, rastreo de objetos, entre otras. Referente al consumo de hardware, se logró diseñar un sistema con bajo consumo de hardware ya que se utilizó menos del 1% de las memorias embebida y unidades lógicas del FPA Cyclone V.

Como trabajos futuros, se pretende integrar esta unidad de procesamiento a proyectos de visión robótica, los cuales se encuentran desarrollando en la universidad de Guanajuato.

## AGRADECIMIENTOS

A la Dirección de Apoyo a la Investigación y Posgrado (DAIP) de la Universidad de Guanajuato, por el apoyo para realizar el proyecto y a Altera University Program, por la donación de equipo utilizado para el proyecto.

## REFERENCIAS

- [1] Tomasi, M., Vanegas, M., Barranco, F., Díaz, J., & Ros, E. (2010). High-performance optical-flow architecture based on a multi-scale, multi-orientation phase-based model. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(12), 1797-1807.
- [2] Castillo, A., Ortegón, J., Vázquez, J., & Rivera, J. (2014). Virtual Laboratory for Digital Image Processing. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 12(6), 1176-1181.
- [3] Tang, Y., Ten, C. W., Wang, C., & Parker, G. Extraction of Energy Information From Analog Meters Using Image Processing. *IEEE Transactions on Smart Grid*, 6 (4), 2032 – 2040.
- [4] Ramírez, J. M., Flores, E. M., Martínez-Carballido, J., Enriquez, R., Alarcón-Aquino, V., & Báez-López, D. (2010, February). An

FPGA-based architecture for linear and morphological image filtering. In *20th International Conference on Electronics, Communications and Computer (CONIELECOMP)*, 90-95

[5] Hanumantharaju, M. C., Ravishankar, M., & Rameshbabu, D. R. (2013). Design and FPGA implementation of an 2D Gaussian surround function with reduced on-chip memory utilization. In *IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 604-609.