

Diseño de una arquitectura en FPGA para algoritmo demosaicing mediante interpolación bilineal

Miguel Ángel Valdés Ochoa (1), Dr. Eduardo Cabal Yépez (2)

1 Ingeniería en Comunicaciones y Electrónica, Universidad de Guanajuato | Dirección de correo electrónico: ayelog@hotmail.com

2 Departamento de Estudios Multidisciplinarios, División de Ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato | Dirección de correo electrónico: e.cabalyopez@gmail.com

Resumen

La mayoría de las cámaras digitales usan un solo lente para captar toda la escena y para poder obtener una imagen a color es necesario que dicha escena llegue a un arreglo de sensores llamado Mosaico de Bayer, donde cada sensor filtra una longitud de onda específica (RGB). En este trabajo se propone una arquitectura en FPGA para reconstruir la imagen de una cámara digital, utilizando la interpolación bilineal como algoritmo demosaicing, obteniendo como resultado las tres matrices de color, consumiendo menos del 1% de los recursos y con un procesamiento superior al tiempo real alcanzando los 200 cuadros por segundo (fps) para una imagen VGA.

Abstract

Most digital cameras use a single lens to capture the whole scene, and to obtain a color image it is necessary that this scene comes from an array of sensors called Bayer mosaic, where each sensor filters a specific wavelength (RGB). This paper proposes an FPGA architecture to reconstruct the image from a digital camera using bilinear interpolation as demosaicing algorithm, resulting into three color matrices, that consumes less than 1% of resources at a higher processing than real time, reaching 200 frames per second (fps) for a VGA image.

Palabras Clave

Interpolación Bilineal; Algoritmo demosaicing; FPGA; Procesamiento de imágenes;

INTRODUCCIÓN

Los sistemas con sensores ópticos hoy en día tienen una gran importancia en el desarrollo tecnológico. En robótica, por ejemplo, un sensor óptico puede proveer más información para el sistema que la mayoría de los otros sensores. Un sistema de visión robótica comúnmente están compuestos por un solo sensor óptico y un procesador de información, como puede ser una PC [1].

La mayoría de los sensores ópticos usan un solo lente para capturar las escenas, al capturar dicha escena, cada pixel es captado por un sensor con un filtro que deja pasar un color primario específico, sea Rojo, Verde o Azul. Esta matriz de sensores es conocida como Matriz o Mosaico de Bayer [2]. Este sistema fue inventado por el Dr. Bryce Bayer, de la empresa Kodak en 1975. Una vez creada la matriz de Bayer es necesario llevar un proceso de decodificación para poder extraer la información de dicha matriz y reconstruir la imagen completa. Existen diversos métodos ya establecidos como son: Interpolación bilineal, Interpolación bicúbica y la Interpolación de orden superior, siendo la interpolación bilineal la más práctica para implementarse en hardware [3].

En este proyecto se desarrolla la arquitectura en hardware del algoritmo desmosaicing (interpolación bilineal) para la matriz de Bayer donde se logra un procesamiento superior al tiempo real para una imagen VGA mediante su implementación en FPGA. Las pruebas se realizaron con un sensor óptico Li-5m03 y un FPGA DE1 SoC de Altera.

MATERIALES Y MÉTODOS

Para el procesamiento de decodificación de la Matriz de Bayer mediante la interpolación bilineal, son necesarios tres pixeles, los cuales entran a una serie de registros y se desplazan para crear la máscara que irá haciendo un recorrido a través de

toda la imagen. De dicha máscara salen las señales de cada pixel central y sus vecinos, con los cuales se realiza las operaciones de interpolación a cada una de las matrices de cada color y calcular el valor del pixel central. El cálculo del valor de pixel central es básicamente un promedio obtenido de los 2 o 4 valores de los pixeles vecinos [4], estos dependiendo de la posición en el desplazamiento como se ve en la imagen 1 y ecuaciones (1) y (2). El mecanismo aplicado para la matriz R es el mismo para las otras matrices.

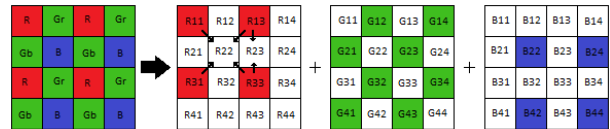


IMAGEN 1: Interpolación bilineal para cada matriz.

$$R_{22} = \frac{R_{11} + R_{13} + R_{31} + R_{33}}{4} \quad (1)$$

$$R_{23} = \frac{R_{13} + R_{33}}{2} \quad (2)$$

Implementación en Hardware

La descripción de la arquitectura consta de un bloque general que contiene una máquina de estados que controla cuando son válidos los datos que se reciben del sensor y cuando deben ser procesados, un contador que sincroniza y controla cuales y cuando los datos de salida son los correctos y la tercera parte es donde se encuentra el algoritmo desmosaicing, el cual mediante las señales procedentes de los pixeles Pix1, Pix2 y Pix3 almacenados en la máscara de entrada consigue los valores de los pixeles vecinos, los cuales entran a una serie de sumadores y registros para realizar el promedio y el truncamiento necesario para poder al final obtener en cada salida (R,G y B) un bus de datos truncado a 8 bits como un estándar para su exhibición en VGA, véase imagen 2 para referencia de bloques.

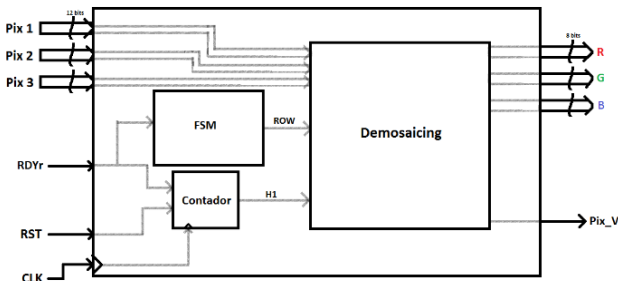


IMAGEN 2: Diagrama a bloques de implementación.

Trabajo experimental

La tarjeta FPGA utilizada es una DE1-SoC [5], modelo de la compañía Altera, la cual cuenta con una memoria SDRAM de 64 MB, una velocidad de reloj de 50 MHz y un puerto VGA, siendo estos los recursos utilizados. Para el sintetizado y compilación se utilizó el software Quartus II en su versión Web libre. El sensor óptico empleado es una cámara CMOS Li-5m03 de 5Mp, que cuenta con un arreglo de pixeles de 2752x2004, capaz de alcanzar una frecuencia de 96 MHz, de la compañía LEOPARD iMAGIN, que contiene un sensor interno MT9P031 [6] de la compañía ON Semiconductor.

RESULTADOS Y DISCUSIÓN

Se realizó una simulación para verificar el funcionamiento del sistema como se muestra en la imagen 3, y con esto se obtuvo el valor de latencia de 4 ciclos de reloj que corresponden a la entrada de los datos válidos y la salida de los primeros datos ya procesados.

En la simulación tenemos la señal de reloj Clk, la señal de control de línea valida RDYr, las entradas de los pixeles Pix1, Pix2 y Pix3, las salidas de datos para cada color R, G y B y la señal de control de salida de datos validos Pix_Valid, respectivamente.

El consumo de recursos se obtuvo mediante la herramienta de síntesis Quartus II y los resultados

se muestran en la tabla 1, donde se observa que el consumo es menor al 1%.

Las pruebas de procesamiento para diferentes tamaños de imágenes se muestran en la tabla 2 donde para tamaños menores a los 1024x1024 se observa que el procesado está por encima del tiempo real.

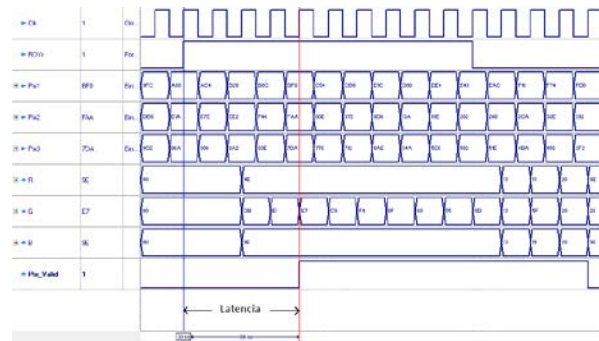


IMAGEN 3: Simulación VHDL.

Tabla 1: Recursos de la FPGA

Recursos de la FPGA	Utilizados	Disponibles
Registros	216	64,140
Memoria	0	4,065,280
Lógica	92 (< 1 %)	32,070

Tabla 2: Mediciones de tiempo en procesamiento para diferentes tamaños de imágenes

Tamaño de imagen	Cuadros por segundo (fps)
256x256	872
640x480	200
800x480	164
800x600	131
1024x1024	60

La prueba del sistema completo se muestra en la imagen 4, donde se tiene el sensor óptico Li-5m03 conectado al FPGA DE1 SoC. La imagen es desplegada en una pantalla LCD de 7 pulgadas. Como se puede observar, la imagen desplegada en la pantalla LCD presenta una saturación de color verde, esto debido a la vasta información para ese canal de color.

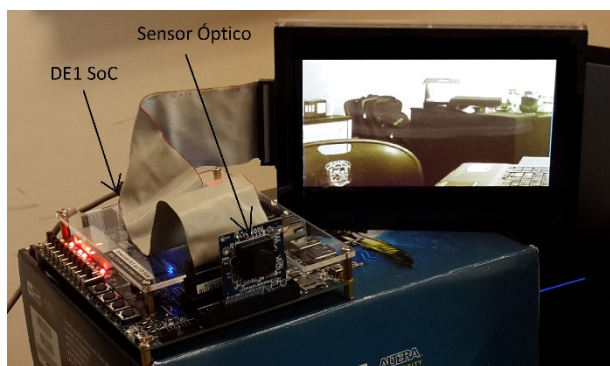


IMAGEN 4: Prueba del sistema.

CONCLUSIONES

El algoritmo de decodificación implementado consigue un procesamiento superior al tiempo real para imágenes VGA alcanzando una tasa de procesamiento de 200 fps con un uso mínimo de recursos para el FPGA Cyclone V. El algoritmo es una arquitectura portable ya que se utilizaron librerías estándar.

Esta es una etapa intermedia de un proyecto más ambicioso, teniendo como trabajo investigar una solución a la saturación de color para un solo canal, obteniendo así mejores imágenes y poder incorporar la arquitectura diseñada en el desarrollo de nanosatélites utilizados para realización de mapas cartográficos y monitoreo terrestre, así como en múltiples aplicaciones de visión robótica y procesamiento digital.

AGRADECIMIENTOS

A la Dirección de Apoyo a la Investigación y al Posgrado (DAIP) de la Universidad de Guanajuato, por el apoyo brindado en el proyecto, a la empresa Prefixa Vision Systems SA de CV por el equipo proporcionado, a Altera University Program por la donación de equipo, y a mis asesores, Dr. Carlos Rodríguez Doñate y Dr. Eduardo Cabal Yépez a quienes les debo la realización del proyecto, gracias por su apoyo y consejos.

REFERENCIAS

- [1] He, W., Yuan, K., Xiao, H., & Xu, Z. (2011). A High Speed Robot Vision System with GigE Vision Extension. International Conference on Mechatronics and Automation, Beijing, China, August 7 - 10.
- [2] Alleysson, D., Süsstrunk, S., & Héroult, J. (2005). Linear demosaicing inspired by the human visual system, IEEE Transactions on Image Processing, 14(4), 439-449.
- [3] Rani, K. S., & Hans, W. J. (2013). FPGA implementation of Bilinear Interpolation Algorithm For CFA Demosaicing. International Conference On Communication And Signal Processing, India, April 3-5.
- [4] Pei, S. C., & Tam, I. K. (2003). Effective Color Interpolation in CCD Color Filter Arrays Using Signal Correlation. IEEE Transactions On Circuits And Systems For Video Technology, 13(6), 503-513.
- [5] ALTERA, "DE1-SoC Development Kit". User Manual. April, 2015.
- [6] ON Semiconductor, "1/2.5-Inch 5 Mp CMOS Digital Image Sensor". MT9P031 Datasheet. 2005.