

# OPTIMIZACIÓN DEL ALGORITMO DEMOSAICING PARA LA ADQUISICIÓN DE UNA IMAGEN A TRAVÉS DE UN SENSOR ÓPTICO

Valdes Ochoa Miguel Ángel (1), Cabal Yépez Eduardo (2)

1 [Ingeniería en Comunicaciones y Electrónica, Universidad de Guanajuato] | Dirección de correo electrónico: [ma.valdesochoa@gmail.com]

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

## Resumen

Para poder obtener una imagen a color, RGB, de un sensor óptico es necesario una etapa de procesamiento donde la imagen a color es reconstruida por algún algoritmo Demosaicing a partir de la imagen cruda que provee el sensor en un mosaico de Bayer, la cual contiene la información de los píxeles previos a cualquier procesamiento. En este trabajo se evalúan tres algoritmos demosaicing en software para determinar cuál de ellos es el más adecuado para una posterior implementación en hardware. Se propone una arquitectura en FPGA para el algoritmo de interpolación bilineal debido a los resultados obtenidos en la evaluación.

## Abstract

An RGB image from an optical sensor is obtained by applying a processing stage where the color image is reconstructed through an demosaicing algorithm from the raw image provided by the optical sensor in a Bayer filter mosaic, which contains the information of pixels prior to any processing. In this paper three demosaicing algorithms are evaluated in software to determine which one is more suitable for a later implementation in hardware. An FPGA architecture is proposed for the bilinear interpolation algorithm because of the results obtained from the evaluation

## Palabras Clave

1. Interpolación. 2. Demosaicing. 3. Imagen Cruda. 4. Hardware. 5. MATLAB

## INTRODUCCIÓN

Como usuario, al tomar una fotografía con algún tipo de cámara digital se espera obtener una imagen con una alta gama de colores, contrastes y matizados. Sin embargo, para llegar a ese punto es necesario llevar a cabo un procesamiento previo que sirva como decodificador entre la imagen que provee el sensor óptico y la imagen que espera el usuario.

Todo sensor óptico al tomar una fotografía provee una imagen carente de colores más allá de una escala de grises llamada imagen cruda (véase IMAGEN (1) [1]). La imagen cruda posee la información de cada pixel sin procesar, obtenida directamente del sensor óptico.



IMAGEN 1: Imagen cruda sin procesar.

Un sensor óptico se compone de una matriz de filtros que dejan pasar un solo color primario específico [2], conocido como mosaico o patrón de Bayer. Un algoritmo decodificador o demosaicing es utilizado para obtener una imagen a color en formato RGB a partir de la información de los píxeles que contiene la imagen cruda codificada en el patrón de Bayer. Algunos de los algoritmos demosaicing más comunes son la interpolación bilineal, interpolación bicúbica y la correlación de color [3].

Para este proyecto se evaluaron dos algoritmos demosaicing, interpolación bilineal e interpolación bicúbica y se compararon contra el algoritmo demosaicing incluido en el toolbox de

procesamiento de imágenes de MATLAB, a fin de determinar cuál es el más apropiado para su implementación en hardware. La matriz de Bayer utilizada es la del sensor óptico Li-5m03 [4].

## MATERIALES Y MÉTODOS

Los algoritmos demosaicing tienen una similitud en funcionamiento, ya que se basan en la información de los píxeles vecinos para obtener el pixel central adyacente a ellos. La diferencia entre ellos es la cantidad de píxeles vecinos que necesitan para hacer el cálculo, la interpolación bilineal requiere 4 píxeles mientras que la interpolación bicúbica requiere 16.

Para la implementación de ambos algoritmos en software se diseñaron dos máscaras con la cantidad de píxeles correspondientes a cada algoritmo, pero con diferente ubicación de los mismos. Estas máscaras se encargan de hacer un recorrido a lo largo de toda la imagen cruda para extraer la información de los píxeles.

Se realizan dos recorridos en el caso de los colores rojo y azul, uno por cada máscara y uno para el color verde, para reconstruir en cada barrido algunos de los píxeles faltantes como se puede observar en la IMAGEN 2, donde IMAGEN 2a es el primer barrido y reconstruye los píxeles necesarios para el barrido en IMAGEN 2b. Es de notar que en cada máscara se consideran píxeles vecinos con diferente ubicación.

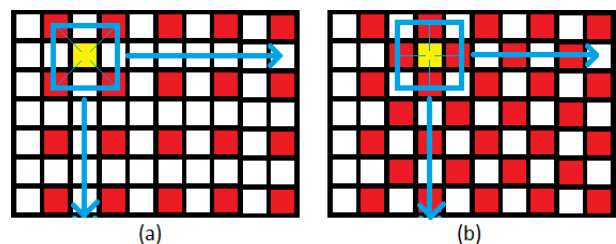


IMAGEN 2: Barrido de máscara para reconstruir el canal rojo a partir de una imagen cruda por interpolación bilineal. (a) Primer barrido, (b) segundo barrido.

La máscara calcula el pixel central mientras hace el recorrido dadas las formula (1).

$$P_{ix} = \frac{P_{ix1} + P_{ix2} + \dots + P_{ixn}}{n} \quad (1).$$

Donde  $P_{ix}$  es el pixel interpolado y  $P_{ix1}, P_{ix2}, \dots, P_{ixn}$ , son los vecinos cercanos utilizados en el proceso.

Para la interpolación bicúbica es el mismo proceso con la diferencia de que la máscara es mayor, ya que esta toma los valores de 16 pixeles vecinos como se había mencionado anteriormente.

A las imágenes reconstruidas se les comparo con la reconstrucción incluida en el toolbox de procesamiento digital de imágenes de MATLAB. Tomando la imagen original como referencia se les aplicaron dos métricas a las tres reconstrucciones, error cuadrático medio (MSE por sus siglas en ingles) y la Relación Señal a Ruido de Pico (PSNR por sus siglas en ingles).

## RESULTADOS Y DISCUSIÓN

La reconstrucción se realizó de manera satisfactoria para todos los métodos, sin embargo, la interpolación bilineal fue la que tuvo mejores resultados a la vista de usuario, esto tomando en cuenta la reconstrucción del toolbox de MATLAB como marca de referencia. En la IMAGEN 3 se pueden observar las 3 reconstrucciones comparadas con la imagen original. Las diferencias son mínimas a simple vista y se ubican en los bordes de cada una de las reconstrucciones, las cuales pierden algunos pixeles.

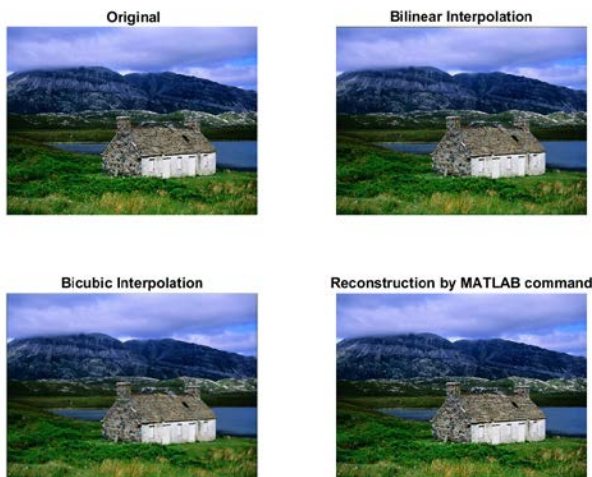
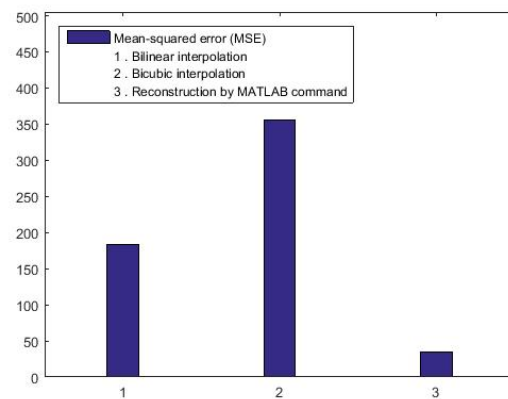


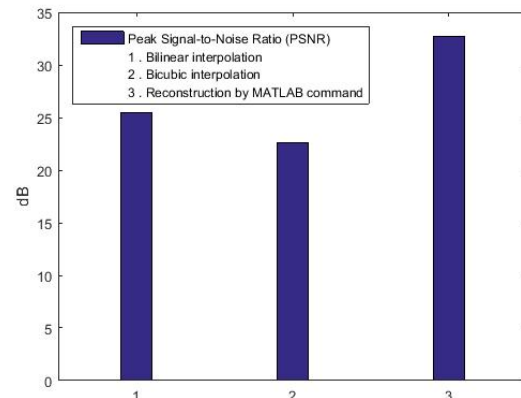
IMAGEN 3: Reconstrucciones por los diferentes algoritmos demosaicing.

En la aplicación de las métricas de evaluación con respecto a la imagen original, la reconstrucción por el algoritmo de interpolación bilineal obtuvo un resultado menor en el MSE y su PSNR fue mayor en comparación con el algoritmo de interpolación bicúbica, como se puede ver en la IMAGEN 4 e IMAGEN 5. Vale la pena recordar que se está buscando un algoritmo factible de implementar en hardware a través de un dispositivo FPGA, por lo que la reconstrucción a través del toolbox de MATLAB se utiliza únicamente como marca de referencia.



Algorithms, 1. Bilinear interpolation, 2. Bicubic interpolation, 3. MATLAB algorithm

IMAGEN 4: Resultados de Error Cuadrático Medio (MSE)



Algorithms, 1. Bilinear interpolation, 2. Bicubic interpolation, 3. MATLAB algorithm

IMAGEN 5: Resultados de Relación Señal a Ruido de Pico (PSNR)

Para la implementación en hardware se propone un diseño donde los canales azul y rojo tienen un sistema de retroalimentación, el cual es activado alternadamente para recompensar las líneas de pixeles faltantes debido a la configuración original

de la matriz de Bayer que el sensor óptico proporciona. Véase IMAGEN 6.

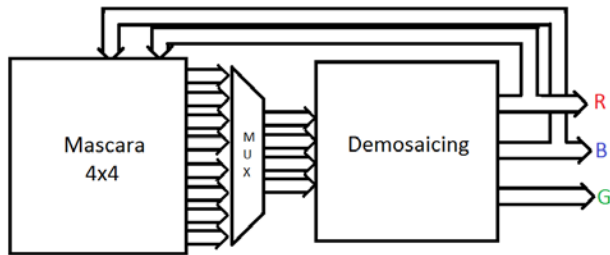


IMAGEN 6: Arquitectura en hardware propuesta

Con los sistemas de retroalimentación aseguramos siempre tener pixeles en los canales azul y rojo y así tener los 4 pixeles vecinos necesarios para la interpolación bilineal.

## CONCLUSIONES

Como se puede observar en la IMAGEN 4 e IMAGEN 5, los algoritmos implementados aun están un poco lejos de tener la calidad de la reconstrucción realizada por el toolbox de MATLAB; sin embargo, desde el punto de vista del usuario no se aprecian grandes diferencias en las reconstrucciones hechas en comparación con la imagen original y la reconstruida por el algoritmo de MATLAB. Para la implementación en hardware se ha optado por la implementación del algoritmo de interpolación bilineal debido a que éste es más sencillo en cuanto a su descripción en hardware y en cuanto a calidad se refiere se observa que supera al algoritmo de interpolación bicúbica.

La implementación en hardware se es el siguiente paso a seguir en un futuro cercano para este proyecto. Esto debido a que, pese a que la interpolación bilineal es el algoritmo más sencillo de los 2, esto no significa que su descripción en hardware sea sencilla, por lo que requiere tiempo adicional del definido en el verano de investigación. Sin embargo se le dará continuación inmediata ya que éste forma parte de otro proyecto más grande, donde se planean aplicaciones de monitoreo visual terrestre y otras aplicaciones de visión robótica.

## 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 Programs 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, consejos y paciencia.

## REFERENCIAS

- [1] Yuan, L., & Sun, J. (2011). High quality image reconstruction from raw and jpeg image pair. IEEE. International Conference on Computer Vision, pp. 2158-2165.
- [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] ON Semiconductor, "1/2.5-Inch 5 Mp CMOS Digital Image Sensor". MT9P031 Datasheet. 2005.