

ALGORITMO DEMOSAICING PARA LA RECONSTRUCCIÓN DE UNA IMAGEN PROPORCIONADA POR UN SENSOR ÓPTICO

Vargas Rojas, Pedro Luis (1), Cabal Yopez, Eduardo (2)

¹ [Ingeniería en Sistemas, Politécnico de la costa atlántica] | [pedro.vr096@gmail.com]

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

Resumen

Para reducir el costo y tamaño de las cámaras digitales, utilizan un solo sensor cubierto por una matriz de filtrado de color, con el fin de registrar toda la información posible de los canales RGB. La imagen producida por el sensor óptico está conformada por pixeles con un solo valor de los canales RGB, por lo que se requiere calcular los dos valores faltantes para generar la imagen a todo color. Demosaicing, es el proceso de interpolación con el que se calculan los valores de los canales faltantes. En este proyecto se evaluaron 2 algoritmos demosaicing de diferentes complejidad y desempeño, con el objetivo de hallar uno de fácil implantación. En el desarrollo del proyecto, se propone un algoritmo que cumpla las características requeridas.

Abstract

To reduce the cost and size of digital cameras, they use a single sensor covered by a color filter matrix in order to record all possible information of the RGB channels. The image produced by the optical sensor is made up of pixels with a single value of the RGB channels, so it is necessary to calculate the two missing values to generate the full color image. Demosaicing, is the interpolation process with which the values of the missing channels are calculated. In this project, two demosaicing algorithms of different complexity and performance were evaluated, in order to find one that is easy to implement. In the development of the project, an algorithm is proposed that meets the required characteristics.

Palabras Clave

1. Interpolación. 2. Demosaicing. 3. Matriz de filtrado de color.

INTRODUCCIÓN

En las últimas décadas, se han desarrollado múltiples aplicaciones para las cámaras digitales, siendo de gran utilidad para complementar las funciones de teléfonos inteligentes, drones, PC's e incluso automóviles [1].

Debido a su alta demanda la mayoría de las cámaras digitales utilizan un único sensor para reducir su costo y su tamaño [2]. Esto influye en la imagen capturada puesto que un único sensor de imagen sólo obtiene un canal de color en cada posición de píxel [3], por consiguiente, es necesario establecer las regiones donde se percibirán las intensidades de luz de longitud de onda roja, verde y azul [4], tal como se muestra en la imagen 1.

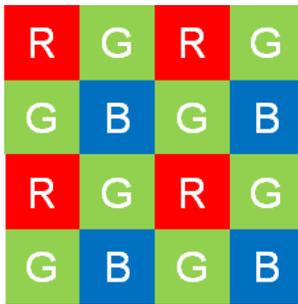


IMAGEN 1: 4 x 4 Matriz de filtrado de color Bayer. Elaboración propia.

La imagen 1 muestra el patrón de Bayer, que es la matriz de filtrado más común, conformada por un 50% de píxeles verdes y el otro 50% de píxeles rojos y azules, a causa de que la vista humana es más sensible al verde que al rojo o al azul [5].

Sin embargo, ¿Cómo se obtiene la imagen a todo color si se necesitan los tres canales? Existe un proceso en el cual se deben interpolar los dos colores perdidos a partir de las muestras circundantes, este proceso es conocido como demosaicing [2].

Para una excelente interpolación de los canales faltantes, diferentes métodos de demosaicing se han propuesto para mejorar la precisión de la interpolación y preservar los detalles de borde y textura [6].

Algunos otros métodos que tienen una complejidad computacional baja a diferencia de los anteriores, pero también traen inconvenientes, tales como efectos de franja, colores falsos y efectos de cremallera, especialmente en la región de borde [2].

De lo anterior, se infiere que existen diversidad algoritmos Demosaicing de complejidad y calidad variante, por esta razón, en este paper, se analiza el desempeño y complejidad computacional de dichos algoritmos. Para lo anterior se evaluaron dos algoritmos demosaicing, interpolación bilineal e interpolación bicúbica y se propuso uno algoritmo con baja complejidad computacional y una calidad aceptable. A continuación, se definirá los algoritmos a evaluados.

Interpolación Bilineal

El método de interpolación bilineal, calcula el valor del píxel faltante promediando los valores de los píxeles vecinos. Dicho método tiene aritmética más simple y consume menos tiempo de procesado. Los valores rojos, verdes y azules se estiman con las siguientes ecuaciones:

$$(1) f(x) = \frac{x_{i-1,j-1} + x_{i-1,j+1} + x_{i+1,j-1} + x_{i+1,j+1}}{4}$$

$$(2) f(x) = \frac{x_{i,j-1} + x_{i+1,j} + x_{i,j-1} + x_{i-1,j}}{4}$$

Donde x , es el píxel faltante y el subíndice indica la posición de los demás píxeles con respecto al faltante. La ecuación (1) se emplea para calcular los valores rojos y azules faltantes, mientras que la (2) se utiliza para los valores verdes faltantes. En la imagen 2, podemos observar gráficamente las ecuaciones.

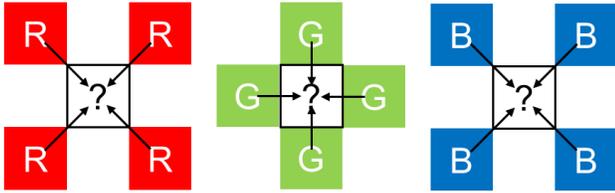


IMAGEN 2: Representación gráfica de la interpolación bilineal.
Elaboración propia.

Interpolación Bicubica

La interpolación bicubica es un método complejo a comparación de otros métodos de interpolación, pero produce mejores resultados, conservando detalles y sin tantos inconvenientes como otros métodos. En este caso se toman una muestra de 16 píxeles, y toma el valor de píxel vecino en una forma de dos cúbicos [4]. Entonces se interpola el área mediante la primera interpolación de las cuatro columnas y luego la interpolación de los resultados en la dirección horizontal [7].

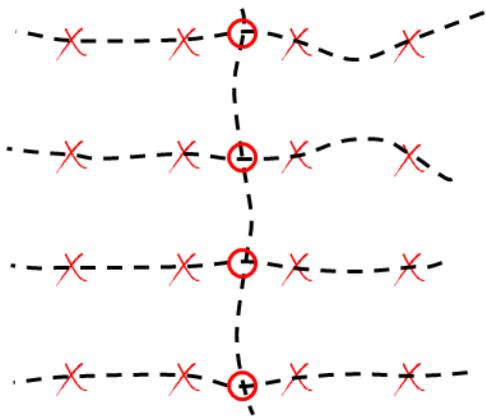


IMAGEN 3: Representación gráfica de la interpolación bicubica.
Elaboración propia.

MATERIALES Y MÉTODOS

La metodología implementada para desarrollar exitosamente la investigación, consistió en una serie de pasos expuesta a continuación:

- 1- Se realizó el análisis de los algoritmos bilineal y bicubica, considerando las

diferencias de complejidad y calidad de los algoritmos.

- 2- Dicho análisis se evaluaba una fácil implementación y una complejidad lineal, por lo que el algoritmo bilineal es el ideal
- 3- Basado en el método de interpolación lineal, se desarrolló un algoritmo que produce un resultado de interpolación aceptables considerando su baja complejidad computacional.

Para la elaboración del algoritmo demosaicing, se utilizó la metodología de prueba y error, en la cual se simulaban las imágenes proporcionada por el sensor óptico, produciendo así, una imagen con el filtrado de Bayer.



IMAGEN 4: Recreación una imagen, proporcionada por un sensor óptico. Elaboración propia.

En la imagen 3, se observa una imagen producida de la simulación, la cual se utilizará para probar diferentes métodos de interpolación. Cada prueba era comparada con la imagen original y la mejor fue seleccionada.

A continuación, las herramientas con las que se realizaron las pruebas y se desarrolló el algoritmo.

Herramientas

Matlab

Matlab, es un programa matemático, que ofrece un entorno de cálculo técnico de altas precisiones para cálculo numérico, se utiliza para el procesamiento de señales, aprendizaje automático, procesamiento de imágenes, visión artificial, comunicaciones, finanzas computacionales, diseño de control robótico y muchos campos más.

Matlab cuenta con un lenguaje propio, su lenguaje está basado en matrices que expresa las matemáticas computacionales.

RESULTADOS Y DISCUSIÓN

Se propuso un algoritmo de interpolación de baja complejidad computacional y de desempeño aceptable basado en la interpolación bilineal. A continuación, se mostrará el diagrama de flujo del algoritmo mencionado

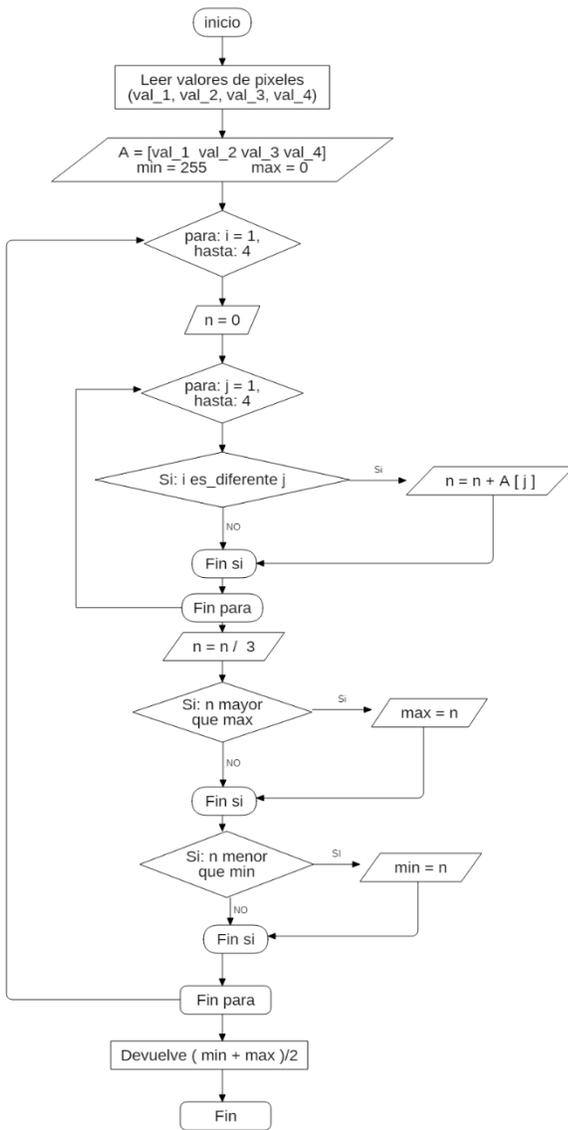


IMAGEN 5: Diagrama de flujo del algoritmo propuesto. Elaboración propia.

En la imagen 5, se observa que en el diagrama de flujo se leen 4 valores, dichos valores son los pixeles cercanos al faltante los cuales serán interpolados. La esencia de este algoritmo es calcular 4 promedios de tres valores, después de calcularlos se promedia el valor mayor y el valor menor obtenido para devolver el promedio faltante. En la siguiente imagen se modela dicho procedimiento.

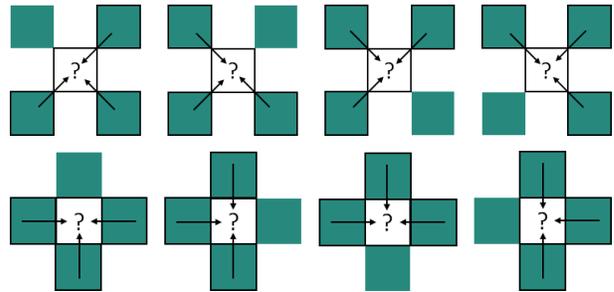


IMAGEN 6: Representación gráfica del algoritmo. Elaboración propia.

En la imagen 6 se representa el origen de los 4 promedios que se utilizaran para calcular el valor faltante. Una vez calculado, dicho valor será parte del origen de otra muestra a interpolar, tal y como se muestra en la siguiente imagen.

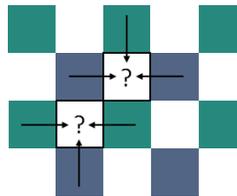


IMAGEN 7: Representación gráfica de la segunda parte algoritmo. Elaboración propia.

En la imagen 4 teníamos una imagen cruda, producida de un sensor óptico, la cual se le aplicó el algoritmo y el resultado es el siguiente:



IMAGEN 7: Imagen 4 con la interpolación propuesta aplicada. Elaboración propia.

CONCLUSIONES

Se desarrolló un algoritmo capaz de interpolar una muestra de píxeles para calcular los píxeles faltantes de la imagen y además cumple la sencillez requerida. El desempeño del algoritmo es aceptable considerando su baja complejidad computacional para generar toda la imagen a color.

Gracias a al análisis de datos hechos en Matlab se concluye con un algoritmo con él que se cumplen exitosamente los objetivos del proyecto.

AGRADECIMIENTOS

Agradeciendo de ante mano a Dios por brindarme la oportunidad de estar en este programa. Todas las personas que conocí y que brindaron un apoyo en estancia en Guanajuato tienen mi gratitud, así como la Universidad de Guanajuato quien es pilar de toda esta maravillosa oportunidad. Por último a mis padres que hicieron la mayor parte de esto posible, gracias.

REFERENCIAS

[1] Konstantinos N. Plataniotis (2005) Color Image Zooming on the Bayer Pattern

[2] Hao Ren, Lei Xie & Huifang Chen (2012) A Color Interpolation Scheme Based on Improved Bilinear Scheme for Bayer CFA Digital Cameras

[3] Jiaji Wu, Marco Anisetti, Wei Wu, Ernesto Damiani, and Gwanggil Jeon (2016) Bayer Demosaicking With Polynomial Interpolation

[4] K.Sudha Rani & W.Jino Hans (2013) FPGA implementation of Bilinear Interpolation Algorithm For CFA Demosaicking

[5] Mingyu Baek and Jechang Jeong (2014) Demosaicking Algorithm using High-order Interpolation with Sobel Operators

[6] Xiangdong Chen, Liwen He, Gwanggil Jeon & Jechang Jeong (2015), Multidirectional Weighted Interpolation and Refinement Method for Bayer Pattern CFA Demosaicking

[7] Paul Breeuwsma (2017). Cubic interpolation - Paulinternet.nl. Recuperado de <http://www.paulinternet.nl/?page=bicubic>