



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Trabajo Práctico III

Develando la mentira de los megapíxeles

Métodos Numéricos  
Segundo Cuatrimestre de 2014

Integrante	LU	Correo electrónico
Aldasoro Agustina	86/13	agusaldasoro@gmail.com
Bouzón María Belén	128/13	belenbouzon@hotmail.com
Cairo Gustavo Juan	89/13	gjcairo@gmail.com



Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

## Resumen

En el presente trabajo se pretende analizar y comparar distintas alternativas que pretenden dar respuesta al problema de demosaicing. Para ello, las mismas serán implementadas y sometidas a experimentación sobre fotografías crudas sintéticas específicamente seleccionadas para poner de manifiesto las ventajas e inconvenientes del empleo de cada método.

### Palabras clave:

- Filtro Bayer
- Demosaicing
- Interpolación
- Artifacts

## Índice

<b>1. Introducción Teórica</b>	<b>3</b>
1.1. Inconvenientes de aproximar mediante el Polinomio de Lagrange . . . . .	3
1.2. Artifacts . . . . .	3
<b>2. Desarrollo</b>	<b>4</b>
2.1. Vecino más cercano . . . . .	4
2.2. Interpolación Bilineal . . . . .	4
2.3. Interpolaciones Direccionales . . . . .	5
2.4. Algoritmo de Malvar, He y Cutler . . . . .	6
2.5. Exploración de Artifacts . . . . .	6
<b>3. Resultados y discusión</b>	<b>7</b>
3.1. MHC . . . . .	7
3.2. Análisis cualitativo de los algoritmos . . . . .	7
3.3. Análisis de tiempos . . . . .	7
<b>4. Conclusiones y trabajo futuro</b>	<b>8</b>
<b>5. Apéndices</b>	<b>8</b>
5.1. Apéndice A: Enunciado . . . . .	8
5.2. Apéndice B: . . . . .	8
<b>6. Referencias</b>	<b>9</b>

## 1. Introducción Teórica

Promediando la década de 1990 comenzaron a introducirse al mercado las cámaras digitales de uso hogareño, empezando a sustituir desde entonces a las ampliamente difundidas cámaras de film. Mientras que estas últimas capturaban la imagen de forma mecánica y sin intercesión de un procesamiento digital (¿? Asegurarse cómo es y explayarse. Estuve buscando y no entendí o no encontré nada. me parece que así esta bien), las cámaras electrónicas emplearon una nueva tecnología. Esta tecnología distingue entre en el momento de obtención de un conjunto de datos cerca del objetivo – donde se obtiene una imagen que llamaremos “cruda” – y una etapa de procesamiento automático conjunto a la reconstrucción de la imagen final.

En el presente trabajo nos interesará analizar un fragmento de dicha etapa conocido como *demosai-cing*. Para entender de qué consta, es menester comprender previamente la forma en que se capturan las imágenes al utilizar cámaras digitales.

Este tipo de cámaras poseen un sensor CCD compuesto de una matriz de elementos fotosensibles, cada uno de los cuales es capaz de captar la intensidad de la luz que llega a ese punto a través de la lente.

La tarea de capturar todos los colores reflejados por el objetivo que se quiere fotografiar no está encomendada a ninguna posición en particular, sino que cada una de ellas tiene asignada – de acuerdo a algún patrón definido – la captura de la información perteneciente a un solo color. Cada píxel de la imagen va a estar dividido en tres canales ya que la vamos a almacenar mediante RGB (Red, Green, Blue).

Ahora bien, esto genera una imagen que se aproxima a la realidad sólo en parte ya que para cada píxel de la imagen tenemos la información de sólo uno de sus tres canales. A causa de esto es que la misma debe ser reconstruida algorítmicamente, interpolando los valores de los colores que no fueron almacenados explícitamente para cada pixel.

*\*insertar imágenes por aquí y por allá\*:*

Filtro Bayer

Otros filtros (?)

En este punto adquieren relevancia las diferentes propuestas de reconstrucción de la imagen original. Esto conlleva la ejecución de diversos procedimientos, entre los que se encuentran el demosaicing, el análisis del balance de blancos, la curva tonal, la saturación y el contraste y la compresión final de la imagen. Como nuestra propuesta se aboca al primer paso mencionado, exploraremos en el presente trabajo sólo un subconjunto de una serie de variaciones no finita que admiten métodos de debayering como ser la interpolación por asignación de valores próximos, la interpolación bilineal y interpolación direccional.

### 1.1. Inconvenientes de aproximar mediante el Polinomio de Lagrange

*Explicar aca porque no se usa, con eso del simulador*

### 1.2. Artifacts

*Aca explicar un poco que onda, que son los artifacts*

## 2. Desarrollo

Dada una *imagen cruda*, los valores que contiene en sus píxeles para cada canal los asumimos reales y ciertos. Por lo tanto, resta averiguar para cada uno de ellos los valores de los dos canales para los cuales dicha información se encuentra ausente.

### 2.1. Vecino más cercano

La implementación más sencilla para la estimación de los canales desconocidos de cada pixel consiste en otorgar a cada canal “vacío” el valor real más próximo que corresponda a dicho color. Este método es algorítmicamente asequible, pero no tiene en cuenta dos factores:

▷ Por un lado, al otorgar el valor de un sólo pixel cercano, surge la necesidad de definir arbitrariamente cuál se debe escoger como valor referencial en casos en que un conjunto de pixeles son equidistantes al analizado y no existe un criterio claro acerca de cuál de ellos debe ser considerado el “más próximo”. Que esta decisión impacte en el resultado de forma inocua o parcial o totalmente nociva dependerá en cierta medida de la fotografía que se esté analizando.

El ejemplo más trivial se encuentra en una imagen monocromática: al ser los valores reales idénticos para todos los pixeles, tomar las magnitudes de cualquier otra posición para cada canal correspondiente no afectará el resultado final. Sin embargo, si consideráramos una imagen formada por columnas alternadas de un pixel de ancho rojas (#0000FF) y blancas (FFFFFF), entonces si para definir los canales verde y azul de un pixel rojo se tomara como posiciones más cercanas aquel que se encuentra debajo suyo y el que está a la derecha del mismo (respectivamente), el resultado sería un pixel fucsia (#FF00FF). En cambio, si los pixeles elegidos fueran su inmediato derecho y en el que se encuentra a su diagonal inferior derecha, la posición analizada debería manifestar el color blanco (#000000).

Si bien se trata de un ejemplo de alcance limitado, resulta suficiente para ilustrar los potenciales problemas de la aplicación de este algoritmo.

[Aca va la fotito que tiene que armar Belu](#)

▷ Por otra parte - y en cierta forma vinculado al ejemplo anterior - este método no especifica ninguna distinción respecto de la forma de proceder en casos de borde o de superficies parejas.

### Decisiones tomadas en el Algoritmo de Vecino Más Cercano

[Aca va todo eso de poner fotos de peor caso para cada mas cercano derecha izuquierda arriba abajo, y que elegimos irnos hacia la derecha.](#)

### 2.2. Interpolación Bilineal

Como la estructura del filtro Bayer sólo asegura para los píxels nativamente azules o rojos tener todas sus posiciones vecinas netamente verdes, aquel es el único caso en que el cálculo de un canal se reduce a computar el promedio de los valores superior, inferior, derecho e izquierdo al espacio que se quiere interpolar.

Sin embargo, si se quisiera obtener el valor de azul o rojo que correspondería asignar a un pixel nativamente verde, el entramado de Bayer obligaría a estimarlo a partir de sus dos valores más cercanos ubicados a la derecha e izquierda o arriba y abajo dependiendo de la ubicación del píxel.

Por otro lado, los valores del canal azul para un pixel rojo (y los del canal rojo para un pixel azul) se calcularían tomando en cuenta las cuatro posiciones diagonales a la analizada. Este proceso no se realizaría simplemente tomando el promedio de sus cuatro diagonales sino que requeriría tomar dos pixels cercanos que pertenezcan a la misma fila o columna, cuyo canal real sea el verde, calcular sus

valores de azul/rojo promediando los de las casillas nativas azules/rojas más próximas y posteriormente realizar un nuevo promedio entre los valores resultantes.

Tomando como referencia la ilustración 1 se puede ver que el cálculo recién descripto equivale a calcular el valor en el canal rojo para el píxel x. Lo cual consistiría en:

Hacer el promedio entre  $a$  y  $b$  y luego, ubicar ese valor para el canal rojo para el píxel ubicado arriba de  $x$ .

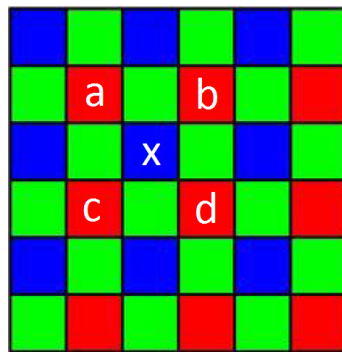
Hacer el promedio entre  $c$  y  $d$  y luego, ubicar ese valor para el canal rojo para el píxel ubicado debajo de  $x$ .

Una vez obtenidos estos dos valores, promediarlos y ubicarlos en el canal rojo de  $x$ .

Llegando así a que realizar este cálculo es equivalente a realizar el promedio entre los valores de sus diagonales:

$$\frac{a+b}{2} + \frac{c+d}{2} = \frac{a+b+c+d}{4}$$

Figura 1:



## 2.3. Interpolaciones Direccionales

Se trata de una gran cantidad de métodos que buscan comenzar a resolver el problema de demosaicing realizando inicialmente una interpolación en una dirección y combinando con algún criterio determinado el resultado estimado con el obtenido de una aproximación realizada en otra dirección.

La implementación de este tipo de métodos exige tomar al menos dos decisiones de peso que distinguirán a unos de otros:

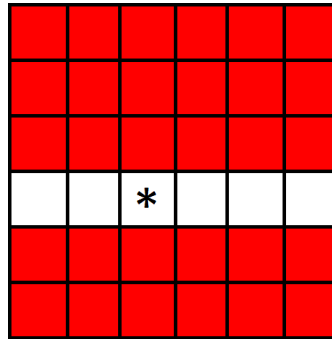
- De qué forma se realizarán las interpolaciones (en qué direcciones y mediante qué mecanismo de interpolación)
- Con qué criterio y consideraciones combinar la información proporcionada por las direcciones escogidas.

Para escoger alguna opción apropiada corresponde analizar las características del Color Filter Array con el que se va a trabajar, puesto que no todas las direcciones van a aportar en la totalidad de los casos la misma información.

En cuanto a la combinación de los resultados de las distintas interpolaciones, es conveniente tener presente que el promedio de los mismos no es siempre una buena opción. Por ejemplo, supongamos que se analiza una imagen que presenta en un sector la siguiente característica:

Si se interpolara utilizando la dirección vertical el valor de la posición denotada con una estrella, todo indicaría que el resultado debería generar el color rojo. Al hacerlo horizontalmente se llegaría a

Figura 2:



la conclusión de que la posición en cuestión debería ser blanca. Promediando #FF0000 con #FFFFFF resultaría el valor #FF8080, que expresa un color rosado.

Ciertamente esto no es lo esperado y existen mecanismos que permiten corregir este error. Uno de ellos consiste en calcular el gradiente del color para utilizarlo al combinar los resultados, de modo tal que si el mismo fuera de gran magnitud (es decir, la variación del color es considerable) entonces el peso que debe tener el valor calculado en el resultado final debería ser menor a aquel que fue calculado en una dirección cuyo gradiente fuera menor (es decir, que se tratara de una superficie más cromáticamente homogénea).

No sé qué más poner acá =/ re cortina.-

## Versiones de Splines

Hablar de todos sus tipos

Decir que bilineal y poner 0.5 queda igual (o casi igual je), usar psnr.

Y ver cual vamos a dejar

Y que usamos NATURALES

## 2.4. Algoritmo de Malvar, He y Cutler

## 2.5. Exploración de Artifacts

Ver si hay otros artifacts mas para agregar.

Ver como arreglar el de Moire y bla...

Poner fotitos de ampliaciones donde se vean bien los artifacts.

### 3. Resultados y discusión

#### 3.1. MHC

Le da muchísima más definición/claridad a las fotos, y los bordes oscuros quedan prácticamente perfectos (comparando con bilineal, por ejemplo, aparecen colores fuertes alrededor de los bordes negros). Sigue pasando eso con los bordes blancos/mas claros, pero las zonas donde hay un salto grande entre claro y oscuro, mejora mucho.

#### 3.2. Análisis cualitativo de los algoritmos

Luego de haber generado las imágenes con todos los algoritmos que hemos propuesto, nos dispusimos a correr en Matlab (R) la diferencia entre ellas y la imagen original de la cual partimos. Luego de eso, invertimos los colores para poder apreciar mejor dónde se notaban las mayores diferencias. A continuación se incluyen las imágenes obtenidas con este proceso, en color blanco y negro.

Aca falta agregar todas las fotitos, Y HACERLO PARA SPLINES!!!!!!!!!!!!!!!!!!!!

En la imagen 2"se puede observar que, si bien al ejecutar el algoritmo de MHC disminuye abundantemente la diferencia con la imagen original, al tener una superficie en su mayoría de agua se dificulta a niveles algorítmicos aproximar el agua de una manera más real ya que esta no es para nada una superficie pareja. (SI EXPLICO EL PORQUE PASA ESTO, MANDO FRUTA ASI QUE NO VOY A HACERLO)'.

En la imagen 4"se puede apreciar el inconveniente de aproximar píxeles que son borde cuando estos son dados en abundancia y en líneas recta en su mayoría. Inclusive en el mejor caso de esta imagen, (la imagen creada mediante el Algoritmo de MHC) se puede apreciar una notable diferencia con la original en lo que respecta a los bordes. Esto es un ejemplo más de la existencia de artifacts.

En cuanto a la imagen 5"se aprecia el mismo inconveniente que en la imagen 1"dado a la existencia de agua, pero a diferencia cuando se corre el algoritmo de MHC la mejoría es notablemente mayor ya que la diferencia entre esta imagen y la foto original es muy pequeña.

En la imagen 9"se puede ver un comportamiento similar a lo ya estudiado, lo interesante de destacar en cuanto a esta foto es que al poseer un letrero es una imagen menos abstracta que las anteriores y por lo tanto, se adquiere una definición mayor cuando se tiene una diferencia menor con la original y así facilitar la lectura del cartel. aca pondria una fotito ahaciendo zoom en el cartel

La imagen 12.<sup>es</sup> una imagen con muchas líneas y dibujos con contornos esto se aprecia en la gran densidad que tiene la imagen que muestra las diferencias entre la imagen original y la imagen obtenida mediante el algoritmo de Vecino Más Cercano. Si observamos las imágenes obtenidas mediante la resta con sus respectivas originales se denota una gran diferencia entre el Algoritmo de Vecino Más Cercano y el Algoritmo de MHC.

Como conclusión, podemos inferir que los lugares donde se muestra una variación son siempre lugares pertenecientes a contornos o bordes de objetos en las imágenes. En cambio, en píxeles sobre superficies llanas o parejas del mismo cuerpo la diferencia entre la imagen original y nuestra imagen obtenida es nula.

#### 3.3. Análisis de tiempos

Aca hay que comparar los tiempos de ejecucion entre todos los algoritmos dispuestos.

## **4. Conclusiones y trabajo futuro**

## **5. Apéndices**

### **5.1. Apéndice A: Enunciado**

### **5.2. Apéndice B:**



## 6. Referencias

[1] Ron Kimmel. Demosaicing: image reconstruction from color ccd samples. *IMAGE PROCESSING, IEEE TRANSACTIONS ON*, 1999.

[2] Henrique S. Malvar, Li wei He, and Ross Cutler. High-quality linear interpolation for demosaicing of bayer-patterned color images. In *Proceedings of the IEEE International Conference on Speech, Acoustics, and Signal Processing*, 2004.