



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

1. Introducción Teórica	3
2. Desarrollo	4
2.1. Vecino más cercano	4
2.2. Interpolación Bilineal	4
2.3. Interpolaciones Direccionales	4
2.4. Algoritmo de Malvar, He y Cutler	5
3. Resultados y discusión	6
3.1. MHC	6
4. Conclusiones y trabajo futuro	7
5. Apéndices	7
5.1. Apéndice A: Enunciado	7
5.2. Apéndice B:	14
6. Referencias	15

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

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.

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.

▷ 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.

2.2. Interpolación Bilineal

Como la estructura del filtro Bayer sólo asegura para los pixels 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 los valores de la columna o de la fila, respectivamente, puesto que la intersección entre las columnas que capturan el color rojo y el verde y las que toman este último junto al azul es vacía.

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 posiciones diagonales a la analizada. Esto requeriría tomar dos pixels cercanos cuyo canal real sea el verde y cuyas filas/columnas resulten paralelas entre sí, 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.

En la ilustración se puede ver que esa cuenta equivale a hacer el promedio entre los valores diagonales más cercanos, puesto que $((a+b)/2 + (c+d)/2)/2 = (a+b+c+d)/4$.

Yo aca “copiaría” lo del enunciado con el mismo dibujito y todo, esta mal?

2.3. Interpolaciones Direccionales

Idem ↑

2.4. Algoritmo de Malvar, He y Cutler

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.

4. Conclusiones y trabajo futuro

5. Apéndices

5.1. Apéndice A: Enunciado



Develando la mentira de los megapíxeles

Motivación

El mundo de los *recruiters* está en constante movimiento. Ya todos están enterados que los alumnos de Métodos Numéricos estuvieron resolviendo problemas de procesamiento de imágenes no triviales como denoising basado en diferencias finitas, y que además manejan varios temas idóneos para el área como la descomposición en valores singulares y el cálculo de autovalores.

Una empresa muy famosa por la calidad de sus productos en procesamiento de imágenes, Adobby, está interesada en reclutar recursos humanos para sus distinguidas posiciones de *Image processing Ninja Gurú Jedi Master*. Sus recruiters saben que los alumnos en estos momentos están aprendiendo sobre métodos de interpolación y están muy interesados en ver el desempeño en la resolución de algún problema de procesamiento de imágenes con los conceptos que están adquiriendo.

Contexto

El mundo de la fotografía digital trajo aparejados nuevos desafíos algorítmicos. Desde el momento en que el fotógrafo presiona el disparador hasta que se puede visualizar la imagen en la pantalla de la cámara o en una computadora, existen varias cuestiones técnicas a resolver para lograr el objetivo.

Uno de los problemas a resolver en la mayoría de las cámaras digitales actuales es el de *demosaicing*.

Si bien usualmente las publicidades de cámaras digitales indican que las mismas capturan una cierta cantidad de píxeles de información, la realidad es que la mayoría solo captura un tercio de la información que publicita. Esto sucede porque el sensor de la cámara no está preparado para capturar todo tipo de luz, si no que cada uno de los píxeles puede captar la luz de un color específico entre rojo, verde y azul.

La decisión de que color capturar en cada pixel da lugar a lo que se conoce como un *Color Filter Array (CFA)*. En la figura 1 se puede ver el CFA más utilizado, el *Bayer array*. Dicho CFA es por lejos el más utilizado en las cámaras digitales actuales, a tal punto que el problema de demosaicing muchas veces se lo refiere como *debayering*. Sin embargo, existen algunos productores de cámaras digitales que deciden utilizar otros entramados por diversos motivos.

El problema de demosaicing consiste en obtener una imagen con información en los 3 canales de colores rojo, verde y azul en cada pixel, a partir de la información capturada por el sensor mediante el Bayer array. Para esto se asume que la información captada por el sensor es correcta y se trata de inferir los valores de los dos canales faltantes en cada uno de los píxeles de la imagen.

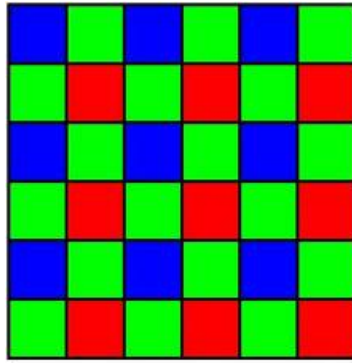


Figure 1: Bayer filter array

Los métodos

Existen varios métodos para abordar el problema de demosaicing, y no queda claro que haya un método que sea mejor que todos los demás. Un aspecto importante del problema, que hace que sea entendible que no haya un único método para resolverlo, es el hecho de que a partir de una entrada válida el problema admite muchas soluciones factibles.

Entre los métodos posibles se encuentran:

Vecino más cercano

Es un método muy simple en el que se rellenan los valores faltantes copiando al vecino más cercano del pixel que tenga una observación real en el mismo canal que se está tratando averiguar.

Interpolación bilineal

Es un método que se compone de sucesivas interpolaciones lineales para aproximar el valor de una función de dos variables, a partir de puntos cercanos al que se quiere averiguar.

Supongamos que en la figura 2 se quiere aproximar el valor del pixel P en el canal rojo. Para esto vamos a utilizar los valores adquiridos en los píxeles 1, 2, 3 y 4 que sí tienen información fehaciente en el canal deseado. La utilización de la interpolación bilineal en este caso se basa en interpolar primero en dirección horizontal, y luego en dirección vertical.

En un primer paso, la interpolación lineal en dirección horizontal de los píxeles 2 y 3 da como resultado la aproximación del canal rojo para el pixel 2-3. Análogamente, el mismo paso aplicado a los píxeles 1 y 4, resulta en la aproximación del canal rojo en el pixel 1-4.

Una vez que se tienen hechas las interpolaciones horizontales, se procede a realizar la interpolación vertical de los píxeles 1-4 y 2-3 para hallar la aproximación del valor del canal rojo para el pixel P .

Esta misma idea puede ser adaptada a otros casos similares que se necesiten abordar.

Interpolaciones direccionales

Muchos de los métodos de demosaicing se basan en la interpolación por direcciones. La idea de estos métodos es poder resolver un problema más simple, como es la interpolación en

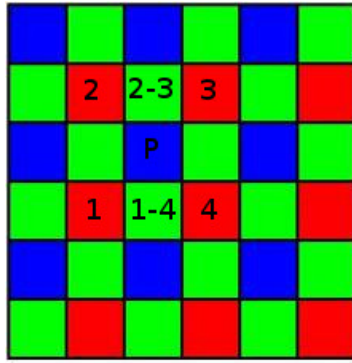


Figure 2: Bayer filter array

una única dimensión, para luego generar la información del pixel en cuestión utilizando las diferentes informaciones direccionales conseguidas.

Uno de los ejemplos más simples dentro de este tipo de métodos sería el de, dado un pixel que queremos averiguar un cierto canal, averiguar su posible valor mediante una interpolación polinómica de toda la fila. Luego, averiguar otro posible valor mediante la interpolación polinómica de toda la columna en la que se encuentra dicho pixel. Una vez obtenido estos dos valores, potencialmente diferentes, se toma el promedio de ellos para generar el valor final del pixel en ese canal.

En esta familia de métodos existen dos aspectos importantes a resolver para generar un algoritmo. En primer lugar, se debe resolver el aspecto de la información direccional. En este punto, las dos decisiones más importantes consisten en determinar que direcciones se van a utilizar (dado que estos algoritmos no tienen porque limitarse unicamente a las direcciones horizontal y vertical) y luego, dado una dirección, de que manera se va a interpolar el valor buscado.

Una vez que se posee toda la información direccional pertinente, se debe decidir de que manera se va a mezclar la información de las diferentes direcciones para generar el valor final. En el caso del ejemplo explicado, la decisión se basó en simplemente dar el mismo peso a la informacion horizontal y a la vertical, tomando un promedio de ambos valores. Sin embargo, existen otras formas de utilizar la información para intentar generar mejores resultados.

Una manera bastante utilizada de juntar la información de las diferentes direcciones es usando el gradiente del pixel que se quiere interpolar. El gradiente, o el conjunto de derivadas direccionales, es un indicador de la variación de color en un determinado pixel. Si el gradiente tiene un valor grande en una determinada dirección, esto significa que en esa dirección el color que se está teniendo en cuenta está sujeto a una variación alta. En general, el hecho de que de un pixel a un vecino haya una variación de color muy grande, está asociado a la presencia de un borde de algún objeto en la imagen. En el caso de un algoritmo ideal, sería interesante poder reconocer estos bordes para que la información de color de un lado del borde no afecte a la interpolación de los colores desconocidos del otro lado. En la práctica, esto se traduce en encontrar una aproximación del gradiente y luego darle más peso a la información correspondiente a la dirección con menor valor en su derivada, ya que está asociada a una dirección en donde el color no está sufriendo grandes cambios.

En el caso de utilizar las direcciones horizontal y vertical, una forma muy común para estimar el gradiente es aproximar las derivadas en cada dirección mediante diferencias finitas de primer orden, lo cual se traduce simplemente en tomar la diferencia de los vecinos inmediatos al pixel estudiado en la dirección correspondiente. Por ejemplo, si buscamos estimar la derivada horizontal en el canal verde del pixel (x, y) de la figura 3, lo podemos hacer mediante la fórmula

$$|\partial_x G(x, y)| \simeq |G(x - 1, y) - G(x + 1, y)| \quad (1)$$

en donde G denota la utilización de la información del canal verde. Se destaca que en la fórmula expuesta se utiliza el valor absoluto, dado que lo que interesa es conocer la intensidad con la que el color verde está cambiando y no en que sentido sucede este cambio.

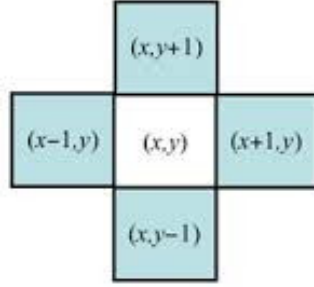


Figure 3: Vecindad 4-connected

Existen muchos métodos que se pueden enmarcar dentro de esta familia de métodos de interpolaciones direccionales, uno de los más conocidos se puede encontrar en [1].

Algoritmo de Malvar, He y Cutler

En [2] se presenta una mejora a los algoritmos básicos de demosaicing para conseguir un mejor resultado. La principal diferencia entre este método y los presentados anteriormente radica en el hecho de que la reconstrucción de cada canal no se hace en forma independiente sino que se utiliza la información de los 3 canales para la reconstrucción integral de la imagen. En este algoritmo en particular, primero se estiman los valores a determinar mediante la interpolación bilineal explicada anteriormente, y luego se corrigen estos valores en base a argumentos relacionados con las derivadas direccionales.

Enunciado

El objetivo del trabajo es experimentar posibles soluciones al problema de demosaicing utilizando las alternativas propuestas. Para esto, se considerará un contexto de aplicación a imágenes crudas sintéticas, las cuales serán obtenidas a partir de aplicar el Bayer Array a imágenes con los 3 canales completos. Este contexto de aplicación tiene sentido por dos lados diferentes. En primer lugar, el hecho de generar imágenes crudas sintéticas tiene como beneficio el poder comparar la calidad de la salida del algoritmo de demosaicing contra la imagen original. Además, dado que solamente se está trabajando sobre el algoritmo de demosaicing, no sería factible aplicar esto sobre datos crudos de una cámara digital, dado que

existen otros pasos necesarios para poder generar una imagen visible, que exceden al alcance de este trabajo.

Para este trabajo se deberán implementar los métodos presentados, específicamente:

- Vecino más cercano.
- Interpolación Bilineal.
- Interpolación por direcciones con posterior decisión sobre como utilizar los valores en las diferentes direcciones. En este punto se pide que entre las opciones que el grupo pueda barajar, esté contemplada la utilización de splines para interpolar toda una fila y toda una columna de una imagen, mezclando luego dichos valores mediante los argumentos asociados al gradiente explicados anteriormente.
- Algoritmo de Malvar, He y Cutler.

Una vez implementados los diferentes métodos, se debe experimentar sobre los mismos todo aspecto que el grupo crea que es interesante para un algoritmo de estas características. La experimentación deberá incluir como mínimo los siguientes aspectos:

- Tiempo de cómputo: Medición basada en la variación de todos los parámetros que se consideren importantes.
- Calidad cuantificable: Utilización de métricas de comparación de imágenes tales como el *Peak Signal-to-Noise Ratio*.
- Calidad subjetiva: Identificación y caracterización de *artifacts* presentes en las diferentes formas de resolución.

En todos los casos, es importante considerar qué condiciones de experimentación son interesantes variar para comparar las diferentes soluciones.

Un punto importante a remarcar es que en el Bayer Array la información no está distribuída de la misma manera para los 3 canales. El canal verde posee el doble de información que los otros dos canales, lo cual suele ser fundamentado desde el hecho de que el ojo humano es más sensible al color verde.

El color verde suele ser el más importante y el más estudiado en los algoritmos de demosaicing dada su importancia en la visualización de la imagen final. Es por esto que muchas veces los algoritmos de demosaicing tienen como primer paso fundamental la búsqueda de una buena interpolación de todo el canal verde, para luego pasar a generar los canales rojo y azul, usualmente en función del canal verde ya determinado.

En el caso de este trabajo, se puede realizar todo el estudio comparativo de los algoritmos observando solamente lo que sucede en el canal verde, dado que es el más importante y es el canal donde la información es más regular (dado que, por ejemplo, tenemos información del canal en todas las filas y columnas de los datos de entrada). En el caso de hacer todo el estudio sobre el canal verde y, de todas maneras, querer generar una imagen a color, se

puede fijar cualquier algoritmo para los demás canales y mantenerlo inmutable durante toda la experimentación.

Fechas de entrega

- *Formato Electrónico:* Jueves 13 de Noviembre de 2014, hasta las 23:59 hs, enviando el trabajo (informe + código) a la dirección `metnum.lab@gmail.com`. El subject del email debe comenzar con el texto [TP3] seguido de la lista de apellidos de los integrantes del grupo.
- *Formato físico:* Viernes 14 de Noviembre de 2014, en la clase de laboratorio.

Importante: El horario es estricto. Los correos recibidos después de la hora indicada serán considerados re-entrega.

References

- [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.

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.