

Procesamiento Digital de Imágenes

Aritmética de pixels.

Claudio Delrieux

Laboratorio de Ciencias de las Imágenes – UNS - CONICET

cad@uns.edu.ar

PDI – Aritmética de pixels

Ya le hemos perdido el miedo a acceder a los valores de los pixels para modificarlos y de esa forma procesar convenientemente algunos aspectos de una imagen digital. Veremos ahora cómo (y por qué) combinar dos o más imágenes.

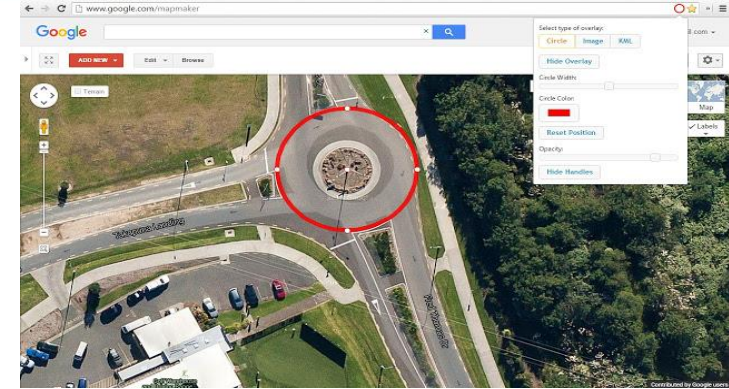
Básicamente podríamos clasificar las operaciones que combinan dos imágenes en dos tipos, las *cuasi-sumas*, y las *cuasi-restas*. Supongamos que queremos combinar la imagen A con la imagen B y obtener una nueva imagen C (todas de idéntica resolución).

En una cuasi-suma, la luminancia (o intensidad de los canales RGB) de la imagen C es una función *monótona no decreciente* de la luminancia (o RGB) de B. Y en una cuasi resta, es a la inversa.

PDI – Aritmética de pixels

Qué utilidad tiene este tipo de operaciones?

Todas las operaciones de cuasi-suma permiten realizar superposiciones de imágenes (cross-fade, overlay, máscaras, etc.).



PDI – Aritmética de pixels

Las cuasi-restas, por su parte, son de gran utilidad en procesamiento de imágenes, porque la diferencia entre imágenes permite revelar cambios, movimientos, etc.



PDI – Aritmética de pixels

Podemos sumar directamente dos pixels en una operación al estilo de la siguiente?

$$C[i, j] := A[i, j] + B[i, j]$$

Claramente no, porque $A[i, j]$ y $B[i, j]$ contienen enteros largos que codifican de alguna manera los valores independientes de RGB del pixel (i, j) .

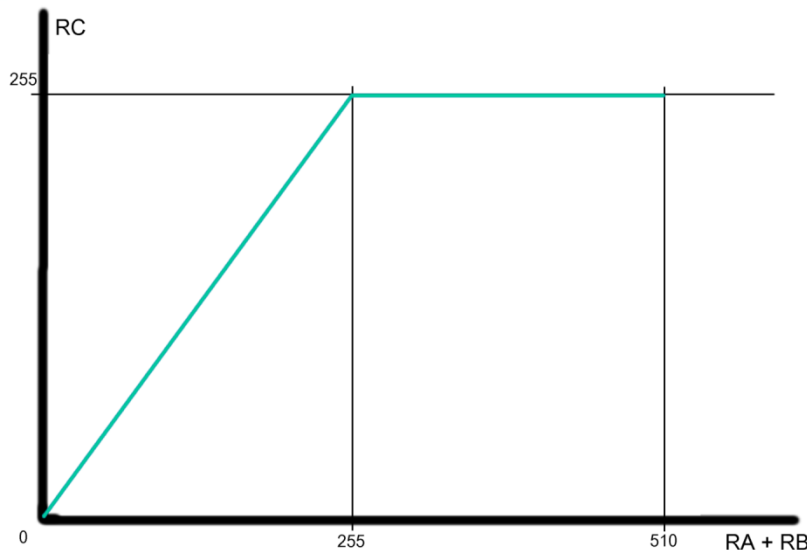
Es decir, es necesario sumar independientemente los bytes de RGB.

PDI – Aritmética de pixels

Pero ahora nos encontramos con un nuevo problema: no podemos sumar directamente dos bytes y colocar el resultado dentro de un solo byte, porque la aritmética “no cierra”.

Una forma posible para arreglar este problema es “recortar” el valor de la suma (*clamping* o anclaje).

Vemos en este ejemplo cómo quedaría el valor del canal rojo de la imagen C (RC) en función de RA y RB.



PDI – Aritmética de pixels

Una forma de implementar esta cuasi-suma es entonces sumar independientemente los bytes de los tres primarios, y coercionar independientemente cada uno si se pasa de 255. A esta forma de “cerrar” la aritmética de la cuasi-suma la denominamos “*suma clampeada en RGB*”.

Uno de los problemas, sin embargo, de esta solución simple, es lo que se denomina “*deriva cromática*” (*color shift*), dado que al clampearse o saturarse independientemente los primarios, el resultado es que los colores finales de los pixel se alteran.

PDI – Aritmética de pixels

Por ejemplo en la suma de estas dos imágenes, se puede observar cómo la suma, además de saturar en muchos sectores de la imagen, produce colores no presentes en los originales (ver sobre todo el tope de la fuente en la imagen C).

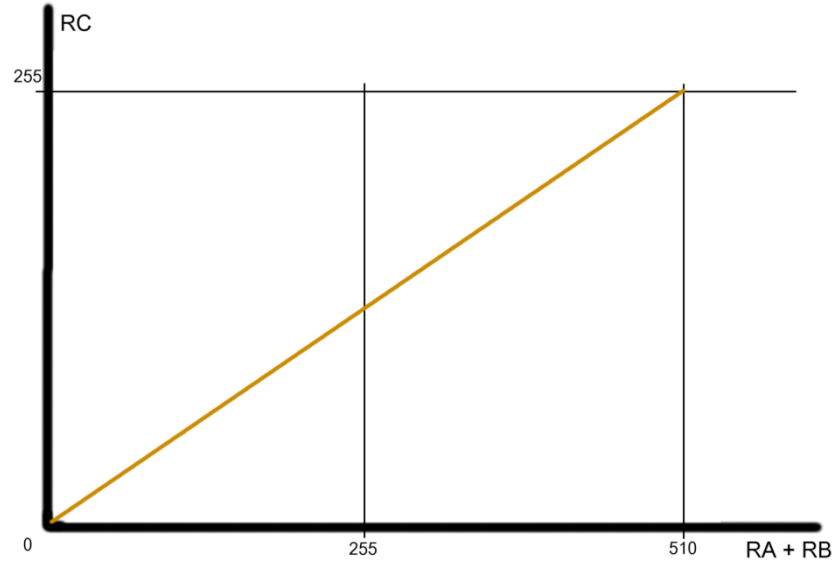


Esto se debe a que esos pixels eran rosados originalmente, pero al sumarles el verde de la imagen A, el verde “saturó” independientemente del rojo (que siguió en su nivel original).

PDI – Aritmética de pixels

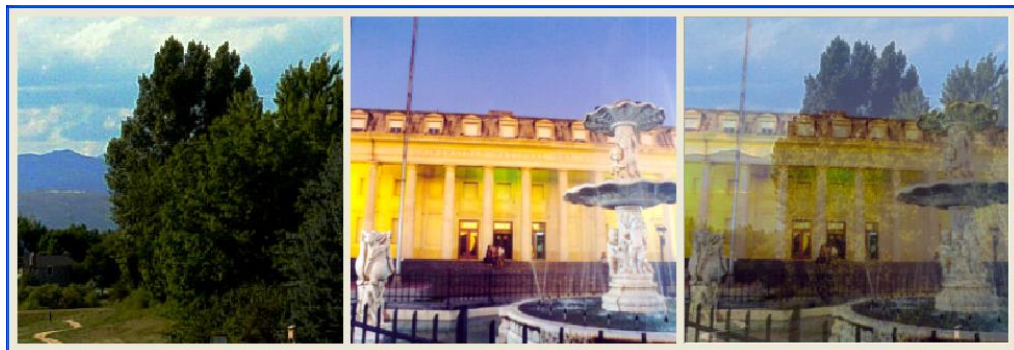
Una forma de evitar ese y otros problemas consiste en buscar diferentes formas de “sumar” los colores de las imágenes sin que existan discontinuidades.

Por ejemplo podemos reemplazar la función anterior (clampeada) por una lineal, que equivale en realidad a utilizar en la imagen C el promedio de los niveles de los primarios de las imágenes A y B.



PDI – Aritmética de pixels

Como se puede observar, al combinar las imágenes de esta manera no se producen alteraciones en el color, pero en cambio aparece otro problema, y es que la imagen C es en general más oscura que las imágenes originales.



Esto se debe a que el nivel de luminancia es el promedio de las luminancias de las imágenes A y B, es decir que siempre que alguna de esas imágenes sea oscura, eso hará que la imagen C sea oscura en ese mismo sector.

PDI – Aritmética de pixels

Una forma de evitar estos dos problemas consiste en desacoplar la luminancia de la cromaticidad de las imágenes, y realizar la cuasi-suma en luminancias. Esto precisamente lo podemos lograr trabajando en el espacio YIQ.

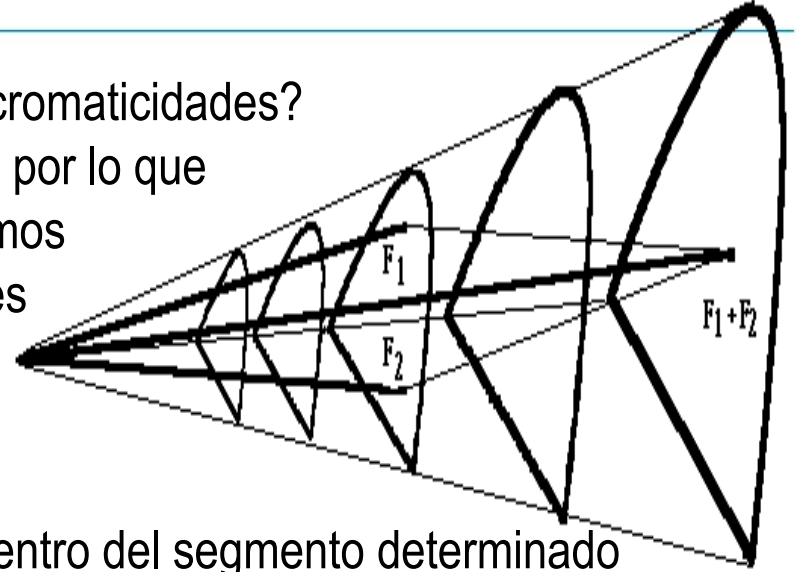
La luminancia de la imagen C será la cuasi-suma de las luminancias de las imágenes A y B, clampeada o promediada:

$YC := YA + YB$; If $YC > 1$ then $YC:=1$; {suma clampeada}

$YC := (YA + YB) / 2$; {suma promediada}

PDI – Aritmética de pixels

Y cuál es el criterio a adoptar para “combinar” las cromaticidades?
Claramente la cromaticidad no es un valor escalar, por lo que sumar es incorrecto en el espacio YIQ. Si recordamos nuestro gráfico del espacio XYZ, vemos dos colores F_1 y F_2 que se suman, formando un nuevo color F_3 utilizando la “ley del paralelogramo”.



Eso quiere decir que la cromaticidad de F_3 yace dentro del segmento determinado por F_1 y F_2 (ver que el vector F_3 corta al plano que contiene a F_1 y F_2 en algún lugar intermedio entre estos dos colores).

PDI – Aritmética de pixels

Este tipo de operaciones aritméticas se conoce como *interpolación*, que seguramente hemos visto en otros lugares (<https://es.wikipedia.org/wiki/Interpolación>). En nuestro caso queremos que IC (resp. QC) sea una interpolación convexa de IA e IB (resp. QA y QB):

$$IC := (YA * IA + YB * IB) / (YA + YB) ;$$

$$QC := (YA * QA + YB * QB) / (YA + YB) ;$$

Cuanto más importante es YA respecto a YA+YB, más importancia tendrán IA y QA en el color final, y viceversa.

PDI – Aritmética de pixels

En la suma YIQ clampeada se puede observar que no se producen los efectos debidos a la saturación de cada primario independientemente de los demás.



Por otro lado, es interesante notar que la suma YIQ promediada produce el mismo resultado que la suma RGB promediada. Esto se debe a que el promedio es una operación lineal y por lo tanto conmuta respecto de otras operaciones lineales como por ejemplo el cambio de sistemas de coordenadas (RGB \rightarrow YIQ es un cambio de coordenadas).

PDI – Aritmética de pixels

Una última cuasi-suma de utilidad es la operación denominada “if-ligther”, que básicamente retiene un en el pixel de la imagen C el pixel correlativo de mayor luminancia entre la imagen A o B:



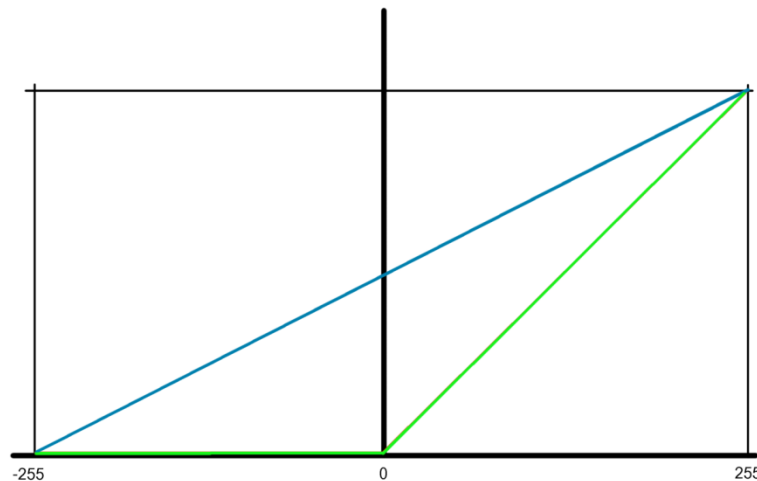
```
if  $Y_A > Y_B$  then  $\{Y_C := Y_A; I_C := I_A; Q_C := Q_A\}$   
else  $\{Y_C := Y_B; I_C := I_B; Q_C := Q_B\};$ 
```

Se puede realizar una operación análoga en RGB, pero es garantía de obtener resultados con colores poco adecuados.

PDI – Aritmética de pixels

Pasamos ahora a ver las cuasi-restas. En este caso los valores posibles de la resta entre bytes están entre -255 y 255, por lo que la forma de pensar las combinaciones “clampeada” y promediada son las mostradas en las curvas verde y azul.

Existen otras formas, por ejemplo retener el valor absoluto de la resta, pero por el momento nos concentramos en estas dos formas de cerrar la operación aritmética.



PDI – Aritmética de pixels

Vemos en la fila superior las dos imágenes originales y la resta en RGB clampeado. En la fila inferior la resta en RGB promediado (que coincide con YIQ promediado), y en IYQ clampeado.

Finalmente agregamos también el resultado de la operación “if-darker” que retiene en la imagen C el pixel más oscuro entre la imagen A y B.



PDI – Aritmética de pixels

Existen otras operaciones aritméticas, incluyendo el producto y el cociente, cada una de las cuales tiene determinados tipos de utilidad. Por el momento no profundizamos en esos detalles, pero básicamente la implementación de cualquier operación aritmética consiste en seguir los pasos indicados:

- En qué espacio cromático se realiza
- De qué manera se “cierra” la aritmética