

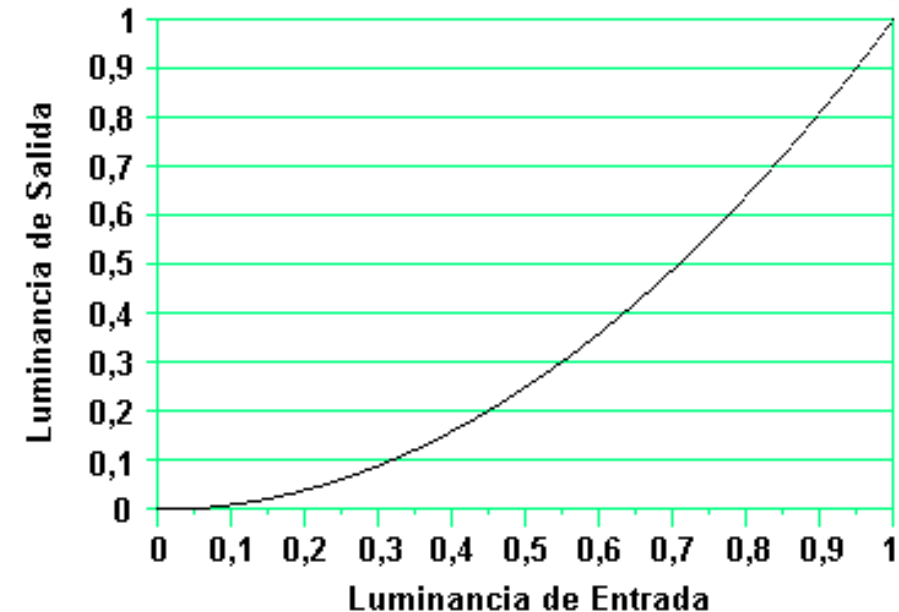
PDI – Histograma de luminancia

Razonando en forma simétrica, la función adecuada para mejorar una imagen muy iluminada será la función cuadrática:

$$Y' := Y^2; \text{ {o } \sqrt{Y} \text{ si el lenguaje tiene}}$$

Tanto $\sqrt{\cdot}$ como \cdot^2 mapean el intervalo $[0, 1]$ a $[0, 1]$ (Y' no se va de rango).

También, $\sqrt{\cdot}$ es siempre mayor a la identidad, o sea, todas las luminancias se incrementan (y viceversa con \cdot^2).



PDI – Histograma de luminancia

Cómo hacemos entonces para implementar estas ideas?. Básicamente reproducimos el workflow del trabajo práctico anterior, con algún cambio:

- 1- Normalizar los valores de RGB del pixel
- 2- RGB \rightarrow YIQ {utilizando la matriz correspondiente}
- 3- $Y' := f(Y)$; {sqrt() o sqr() según se elija}
- 4- $Y'IQ \rightarrow R'G'B'$ {notar que la cromaticidad (IQ) no se altera}
- 5- Convertir $R'G'B'$ a bytes y graficar el pixel

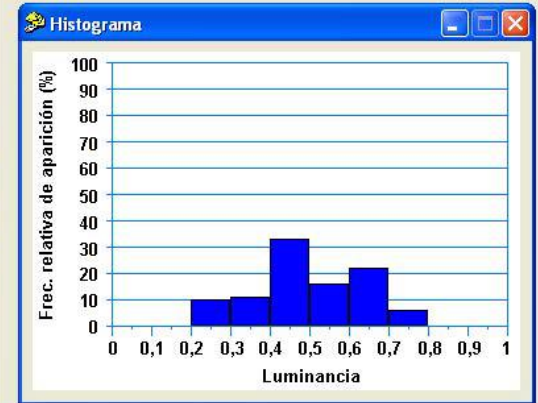
Fácil! No dejen de probarlo.



PDI – Histograma de luminancia

Vamos ahora a considerar el caso más complejo de una imagen con mal contraste o mal rango dinámico, como el que vimos arriba.

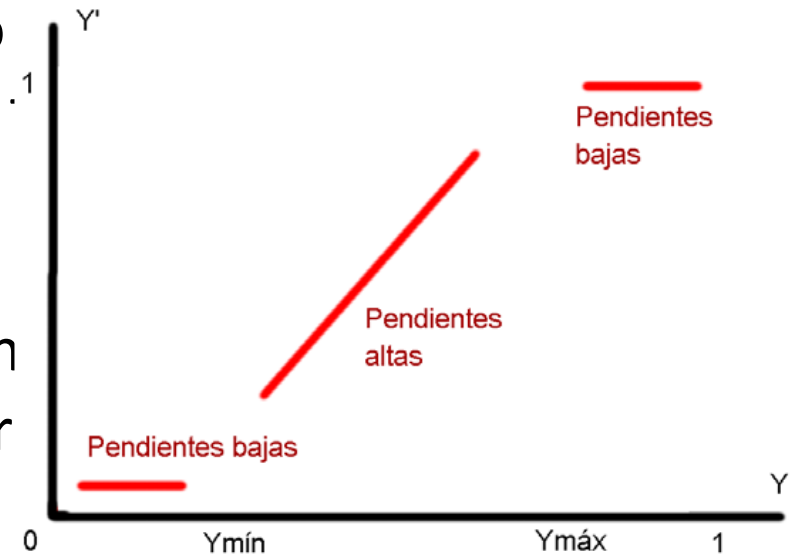
En este caso, no queremos ni oscurecer toda la imagen, ni aclararla toda. Más bien queremos que los pixels oscuros se vuelvan más oscuros y viceversa.



PDI – Histograma de luminancia

Una forma de entender cómo armar este nuevo es ver que en la imagen a procesar ningún pixel luminancia menor que $Y_{\text{mín}}$ ni mayor a $Y_{\text{máx}}$.

Por lo tanto, queremos desparramar las luminan que están entre estos dos valores, o sea mapear intervalo $[Y_{\text{mín}}, Y_{\text{máx}}]$ a $[0, 1]$.



Eso se puede lograr con una función que tenga pendientes bajas entre 0 y $Y_{\text{mín}}$, luego altas hasta $Y_{\text{máx}}$, y luego de nuevo bajas hasta 1.

PDI – Histograma de luminancia

Esta función se conoce como “lineal a trozos”.

Vemos un ejemplo donde $Y_{\text{mín}}=0.2$ y $Y_{\text{máx}}=0.8$.

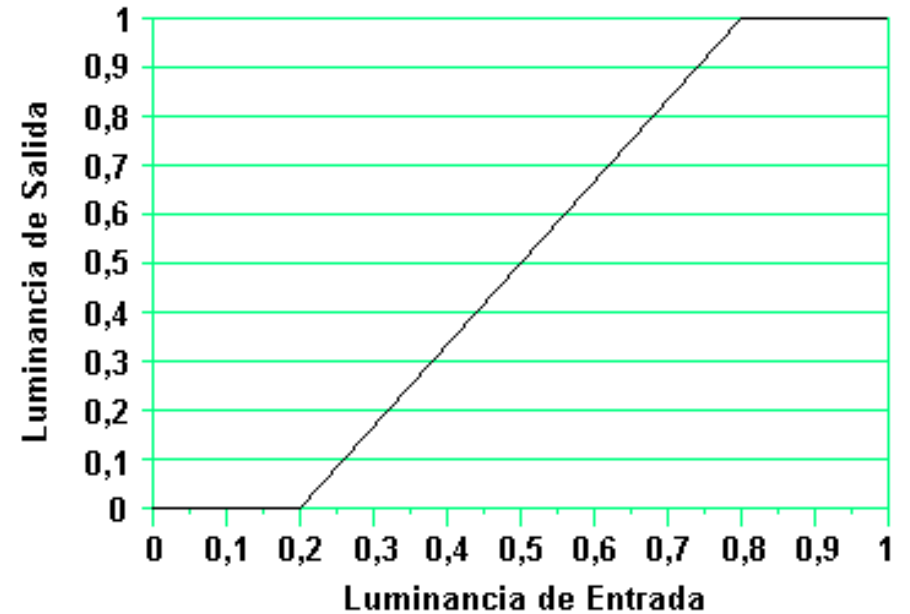
La forma de implementar esta función es por casos:

si $Y < Y_{\text{mín}}$ entonces $Y'=0$

si $Y > Y_{\text{máx}}$ entonces $Y'=1$

caso contrario, realizar interpolación:

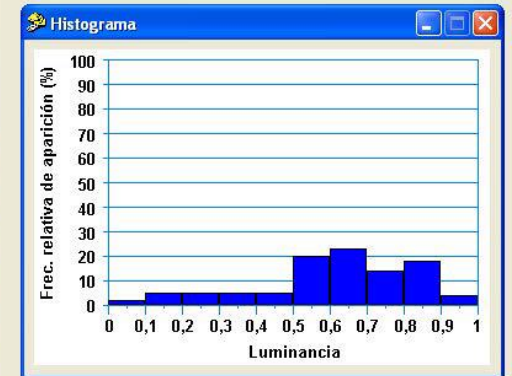
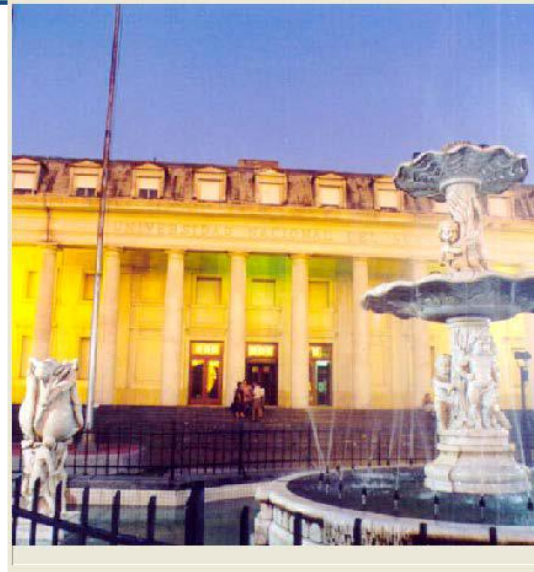
https://es.wikipedia.org/wiki/Regla_de_tres



PDI – Histograma de luminancia

Vemos que en efecto el resultado de este filtrado o procesamiento mejoró el contraste de la imagen, haciéndola más “vívida”.

El histograma, por su parte, es una versión “estirada” del anterior (ahora todas las luminancias son no nulas).



Un problema con este procesamiento es que debemos conocer o estimar Y_{\min} e Y_{\max} , pero se puede aplicar por prueba y error. Por eso es **FUNDAMENTAL** la experiencia.

PDI – Actividad práctica

Recapitulamos las ideas y pautas de trabajo.

Lo ideal es que desarrollen un aplicativo que pueda:

- 1- Abrir una imagen, y luego ver la imagen procesada.
- 2- Se debe poder elegir filtro raíz, cuadrado, o lineal a trozos.

Los controles para elegir los filtros los organizan como les resulte más cómodo.



PDI – Histograma de luminancia

Hay, además de estos tres filtros de manipulación de luminancia, otras ideas más generales (adaptativas, es decir, que tienen en cuenta el histograma de la imagen).

Las ideas subyacentes tienen que ver con qué se entiende por un histograma “correcto” o adecuado. Para algunos, el histograma tiene que poder modelarse con una distribución normal (Gaussiana), con una media 0.5 y variancia 1, mientras que otros opinan que todas las luminancias deberían tener igual frecuencia.

Estas dos técnicas se denominan, respectivamente *normalización* del histograma, y *ecualización* del histograma.



PDI – Histograma de luminancia

Estas técnicas no son extremadamente complejas. La normalización simplemente ajusta la media y variancia del histograma original a la Gaussiana de referencia por medio de una función sigmoïdal:

[https://en.wikipedia.org/wiki/Normalization_\(image_processing\)](https://en.wikipedia.org/wiki/Normalization_(image_processing)).

En el caso de la ecualización, la idea es asignar a cada pixel una luminancia proporcional a la cantidad de pixels menos luminosos que él mismo, lo cual es muy sencillo de computar con un histograma acumulativo:

https://en.wikipedia.org/wiki/Histogram_equalization

Estas ideas quedan para interesados en temas avanzados (y perdón por los links en inglés).

