

Procesado de la imagen digital

- Contenidos:
 - Procesamiento en el dominio espacial.
 - Procesamiento puntual con una o varias imágenes.
 - Procesamiento de vecindad.

FSIV - UNIVERSIDAD DE CORDOBA

Métodos en el dominio espacial

- Métodos puntuales con una imagen:

- Procesado del rango dinámico:

- Normalización del rango dinámico.

- Expansión/Contracción: “Corrección de gamma”.

- Ecualización del histograma.

- Lookup tables.

- Cambios de espacio de color.

$$g(x, y) = T[f(x, y)]$$

Métodos en el dominio espacial

- Normalización del rango del rango dinámico:

Lineal: $[\min V, \max V] \rightarrow [A, B]$

FSIV - UNIVERSIDAD DE CORDOBA

No Lineal: media 0, varianza 1.0

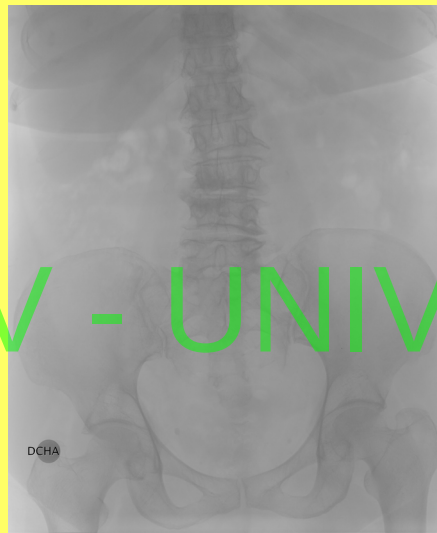
Métodos en el dominio espacial

- Ajuste de gamma, contraste y brillo.

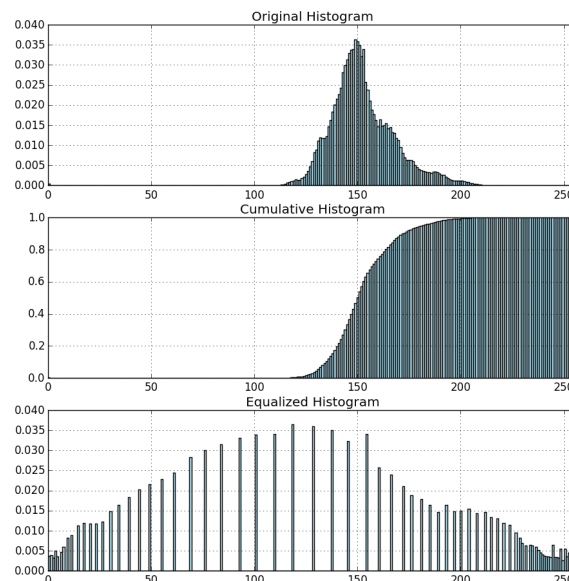
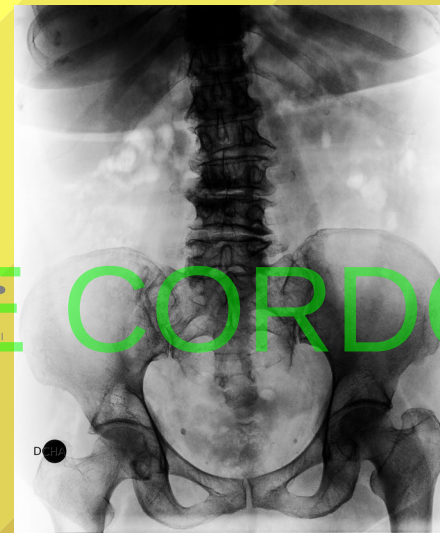
FSIV - UNIVERSIDAD DE CORDOBA

Métodos en el dominio espacial

- Procesamiento del histograma: ecualización.

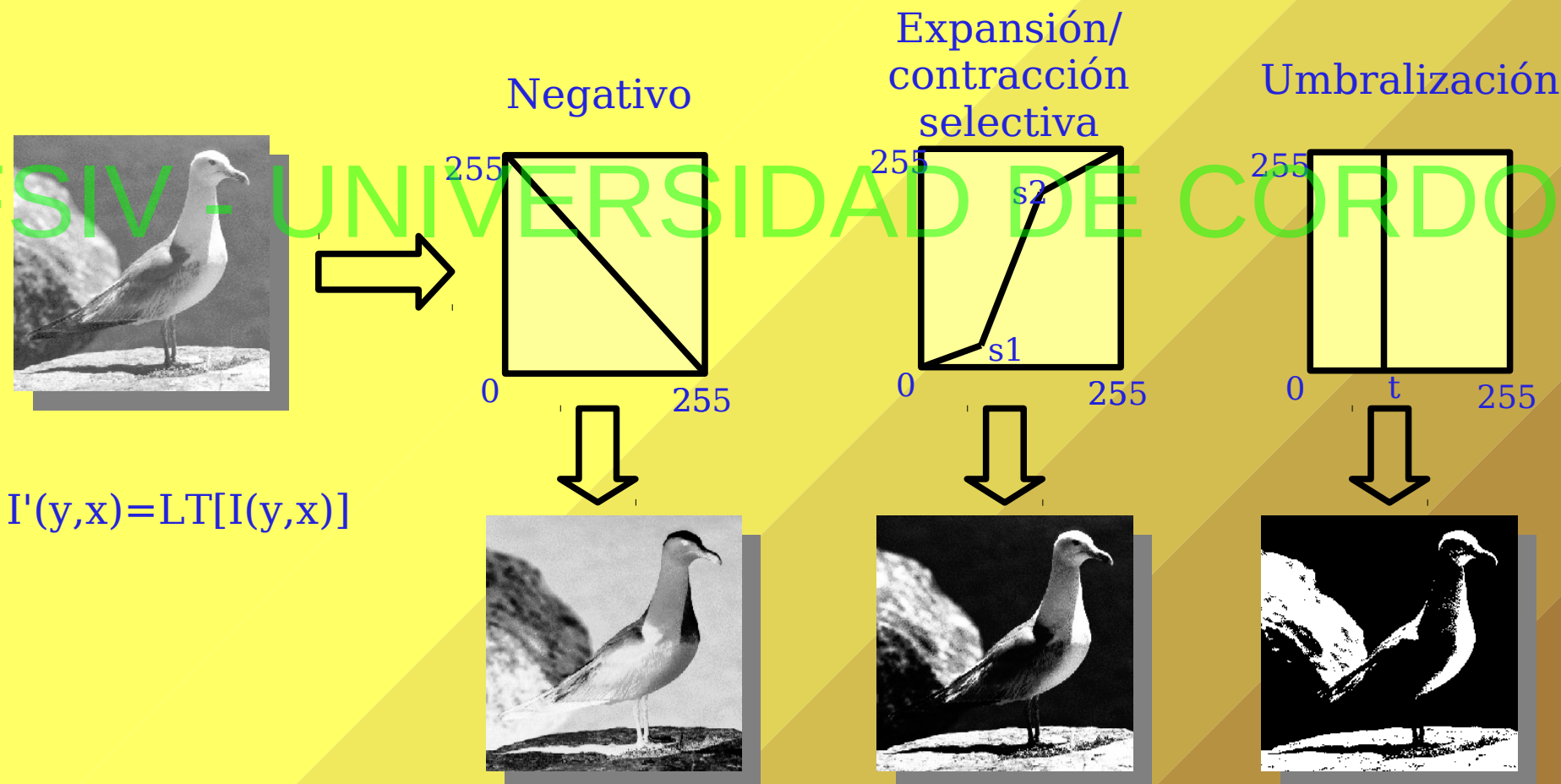


Ecualización



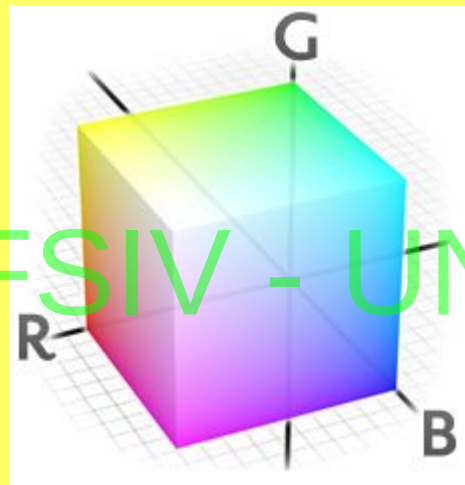
Métodos en el dominio espacial

- Look-up tables.

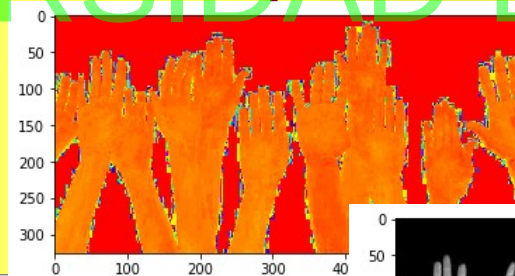
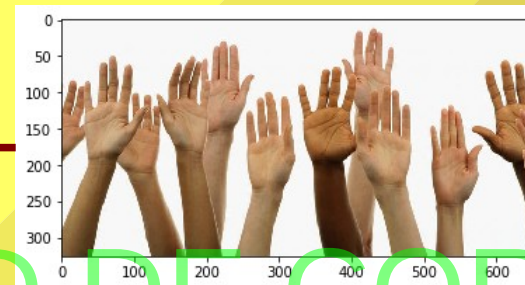


Métodos puntuales

- Cambios del Espacio de Color.

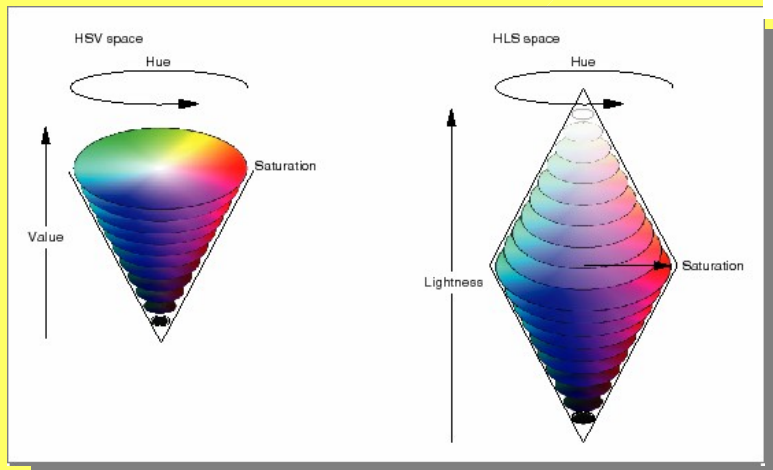
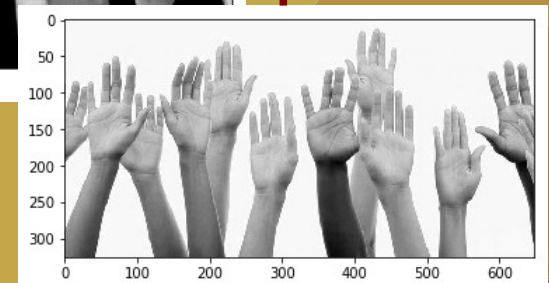
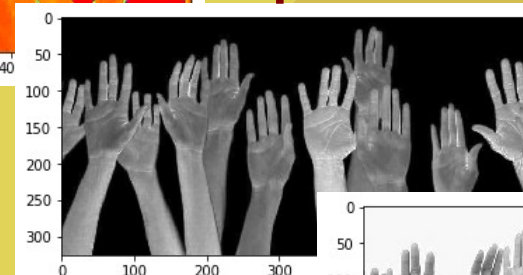


H



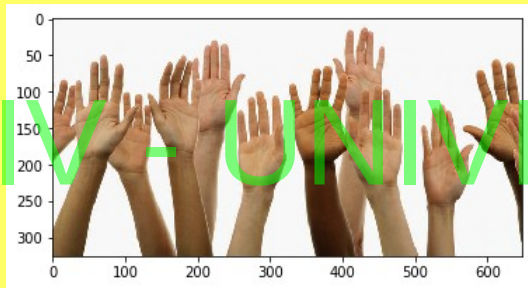
S

V/I

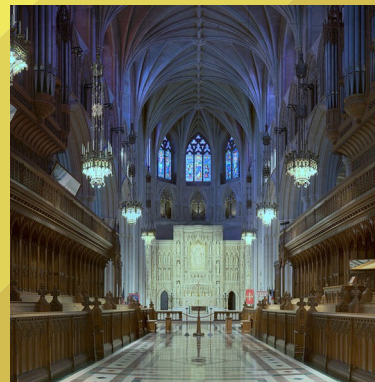
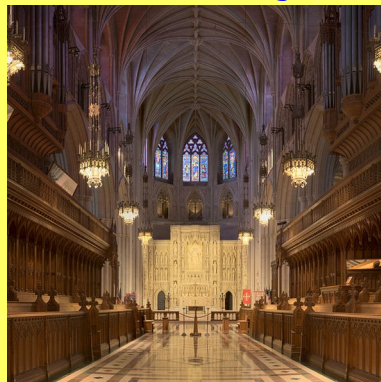


Métodos puntuales

- Equilibrado del color:
 - White patch:



- Gray World:



Métodos en el domino espacial

- Métodos puntuales con dos imágenes:
 - Operaciones aritméticas: + - * /
 - Operaciones lógicas: And, Or, Xor, Not.
 - Operaciones de comparación: <, <=, >, >=, ==, !=

$$g(x, y) = T[f(x, y), h(x, y)]$$

Métodos en el dominio espacial

- Métodos puntuales con dos imágenes:

– Operaciones aritméticas:

Suma.

$$g(x, y) = f(x, y) + \eta(x, y)$$

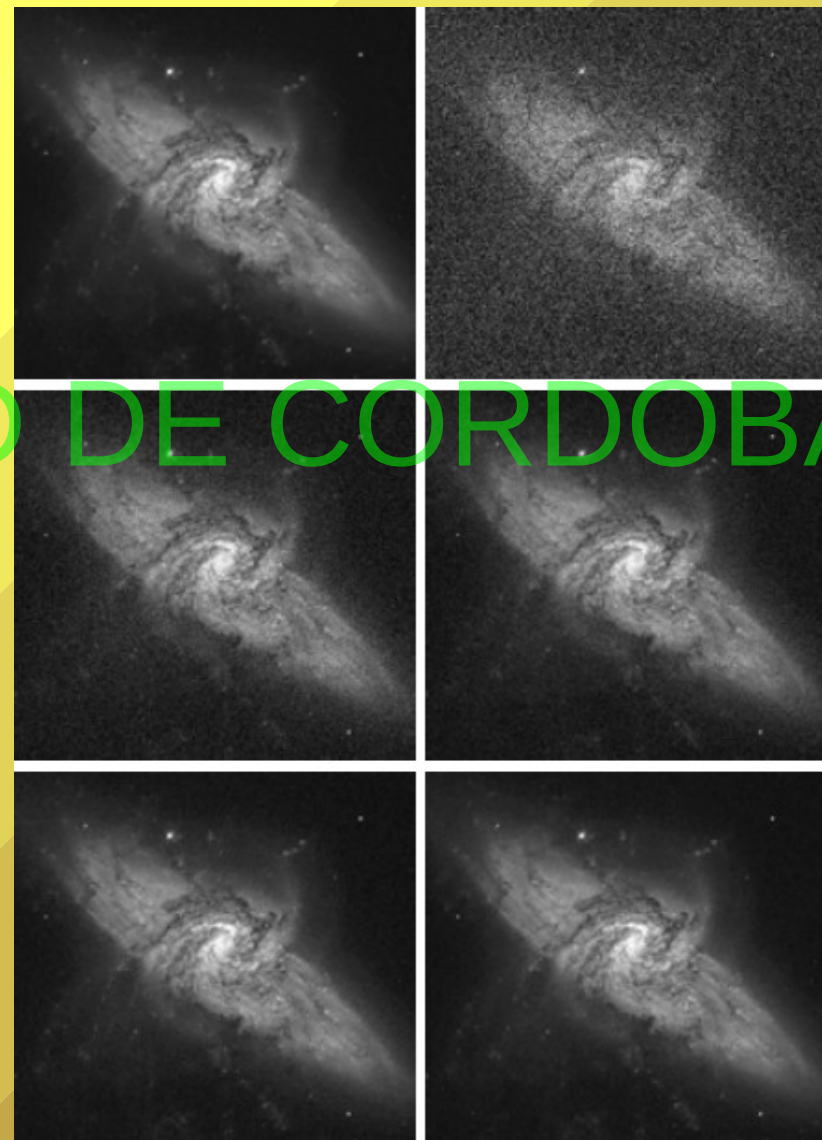
$$E[g(x, y)] = \bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^K g_i(x, y)$$

$$E[f(x, y) + \eta(x, y)] = E[f(x, y)] + E[\eta(x, y)]$$

$$E[f(x, y)] = f(x, y)$$

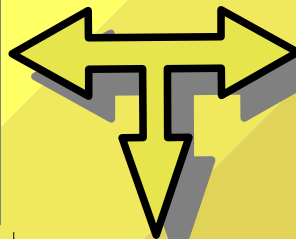
$$E[\eta(x, y)] = \sigma_{\bar{g}(x, y)}^2 = \frac{1}{K} \sigma_{\eta(s, y)}^2$$

K=8, 16, 64 y
128.



Métodos en el dominio espacial

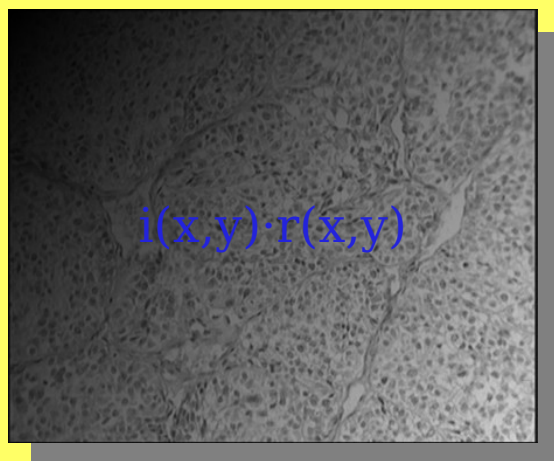
- Métodos puntuales con dos imágenes:
 - Operaciones aritméticas: Diferencia.



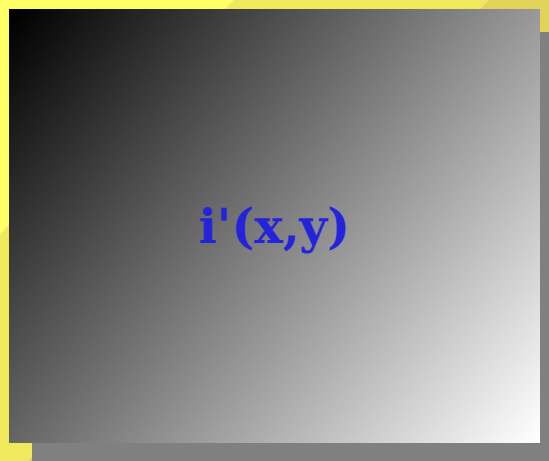
Métodos en el dominio espacial

- Métodos puntuales con dos imágenes:
 - Operaciones aritméticas: Multiplicar/División.

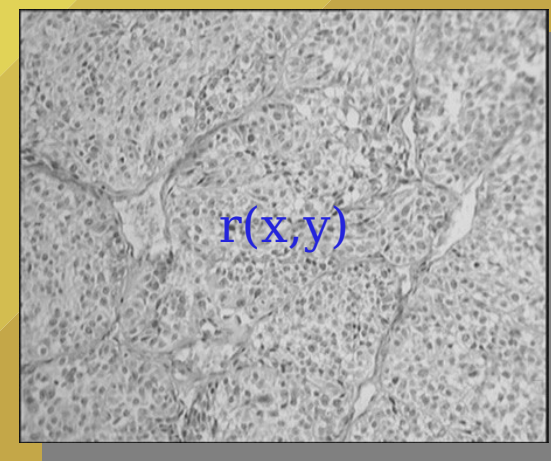
$$f(x,y) = \frac{i(x,y) * r(x,y)}{\bar{i}(x,y)} \approx r(x,y)$$



÷



=



Métodos en el dominio espacial

- Métodos puntuales con dos imágenes:

- Composición de imágenes:

- A nivel de bits (tipos enteros).

$$F(x,y) = [I1(x,y) \& M(x,y)] \mid [I2(x,y) \& \sim M(x,y)]$$

- En punto flotante:

$$f(x,y) = \alpha * i_1(x,y) + (1.0 - \alpha) * i_2(x,y)$$

Operaciones puntuales con OpenCV

Conversión de tipo:

```
//Gray/RGB [0,255] a flotante [0.0, 1.0]:  
img1.convertTo(img2, CV_32F, 1.0/255.0, 0.0);
```

```
//Gray/RGB [0.0,1.0] a uchar [0,255]:  
img1.convertTo(img2, CV_8U, 255.0, 0.0);
```

```
cv::Mat_<float>(A) //Casting a float
```

Sumar/Restar:

```
A += B;  
D = s2 - (A + B - (C+s1));
```

Multiplicar/dividir:

```
C = A.mult(B); //!!!Ojo: C = A*B es multiplicación matricial.  
A = B*s;  
A *= s;
```

Comparar (<,<=,>,>=,==,! =):

```
A = B>C;  
A = B>s;
```

Operaciones puntuales con OpenCV

Funciones generales (vectorización de código):

```
cv::pow(A, s, B); //Potencia B = A^s
cv::exp(A, B); //exponencial B = e^A
cv::log(A, B); //Log. natural B = ln(A)
cv::sqrt(A, B); //raíz cuadrada B = sqrt(A)
B=cv::abs(A); //valor absoluto B = |A|
cv::magnitude(A, B, C); //C=sqrt(A^2 + B^2)
C = cv::max(A, B); //C = max{A,B}
B = cv::max(A, s); //B = max{A, s}
cv::bitwise_and(A, B, C); // C = A & B
...
```

Concepto de máscara: Nos permite especificar sobre que pixeles operar.

```
cv::Mat mask = A > 128; //255 para todo pixel > 128.
//Calculamos el valor medio de los pixeles con valor > 128
mean = cv::mean(A, mask);
```

Operaciones puntuales con OpenCV

Gestión de los canales.

```
img.channels(); //Cuántos canales tiene la imagen.  
std::vector<cv::Mat> canales;
```

```
cv::split(img, canales); //Separa cada canal en una imagen.  
cv::merge(canales, img); //Combina canales en una imagen.
```

```
img.at<cv::Vect3b>(y, x)[0]; //Valor (byte) del canal 0 en pos  
(x,y)  
img.at<cv::Vect3f>(y, x)[1]; //Valor (float) del canal 1 en pos  
(x,y)
```


Referencias

- “Tratamiento digital de imágenes”, González R.C y otros. Addison-Wesley.
- Documentación de OpenCV.