

# **Transformaciones en la escala de grises**

Proceso Digital de Imágenes

David Guillermo Morales Sáez

## Escalado

El escalado consiste en expandir el rango de niveles de gris de una imagen. Para ello, se ha utilizado el siguiente algoritmo en Matlab:

```
function [ salida ] = escalado( imagen )  
    minimo = min(min(imagen));  
    maximo = max(max(imagen));  
    diferencial = maximo - minimo;  
    auxiliar = 255/diferencial  
    salida = (imagen-minimo)*255/diferencial;  
    imshow(salida);  
end
```

con el cual se ha obtenido el siguiente resultado:

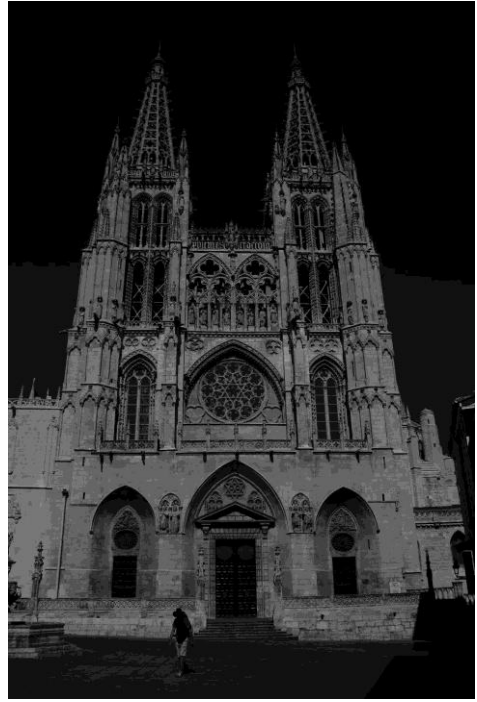


## Stretching

Esta técnica consiste en aplicar varias transformaciones lineales por tramos para mejorar el contraste de aquellas partes que sea necesario. Para ello, se ha utilizado el siguiente algoritmo en Matlab:

```
function [ salida ] = stretching( img, umbral1, umbral2, alpha, beta,
gamma)
    L = max(max(img));
    [ x,y ] = size(img);
    va = 0;
    vb = 0;
    for i=1:x
        for j=1:y
            if img(i,j)<umbral1
                salida(i,j) = alpha*img(i,j);
            else
                if img(i,j)<umbral2
                    if img(i,j)==umbral1
                        salida(i,j) = alpha*img(i,j);
                        va = alpha*img(i,j);
                    else
                        salida(i,j) = beta*(img(i,j)-umbral1)+va;
                    end
                else
                    if img(i,j)==umbral2
                        salida(i,j) = beta*(img(i,j)-umbral1)+va;
                        vb = beta*(img(i,j)-umbral1)+va;
                    else
                        salida(i,j) = gamma*(img(i,j)-umbral2)+vb;
                    end
                end
            end
        end
    end
end
```

con lo cual, se han obtenido los siguientes resultados:

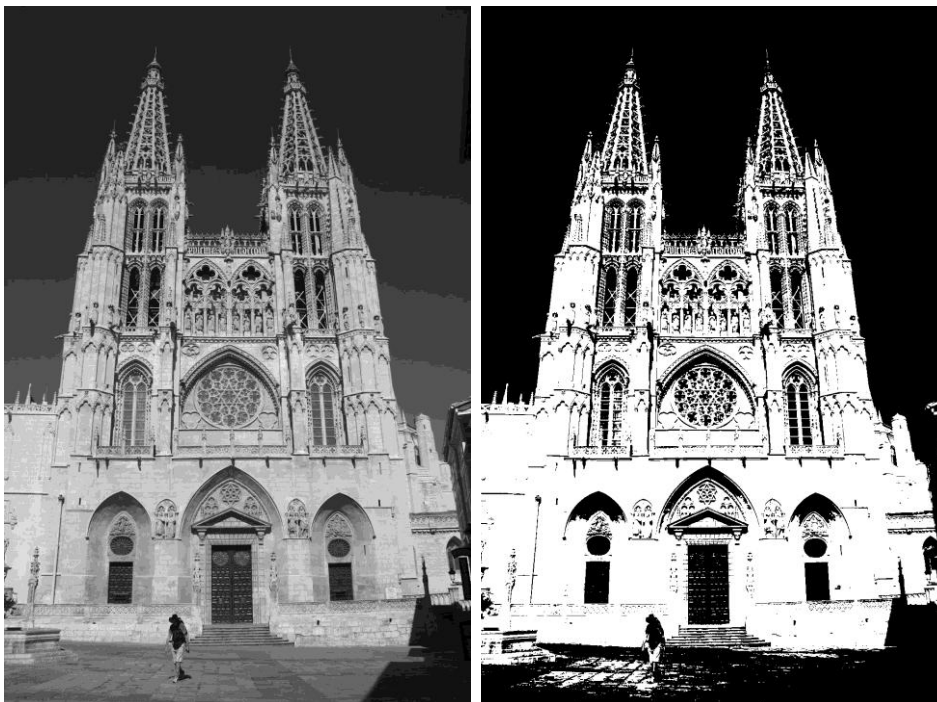


## Truncamiento

El truncamiento es un caso particular del stretching donde sólo se lleva a cabo una transformación en el segundo segmento, quedando el resto a 0. Para ello, se ha utilizado el siguiente algoritmo en Matlab:

```
function [ salida ] = truncar( img, umbral1, umbral2, beta )
    L = max(max(img));
    [ x,y ] = size(img);
    vb = 0;
    for i=1:x
        for j=1:y
            if img(i,j)<=umbral1
                salida(i,j) = 0;
            else
                if img(i,j)<umbral2
                    salida(i,j) = beta*(img(i,j)-umbral1);
                else
                    if img(i,j)==umbral2
                        salida(i,j) = beta*(img(i,j)-umbral1);
                        vb = beta*(img(i,j)-umbral1);
                    else
                        salida(i,j) = vb;
                    end
                end
            end
        end
    end
end
```

con el cual se ha obtenido el siguiente resultado:



## Extracción de bits

La extracción de bits consiste en extraer el n-ésimo bit más significativo de cada píxel de la imagen. Para ello, se ha utilizado el siguiente algoritmo de Matlab:

```
function [ salida ] = extraccionbits( im, n )  
    [ x,y ] = size(im);  
    salida = im;  
    for i=1:x  
        for j=1:y  
            salida(i,j) = bitset(salida(i,j),n);  
        end  
    end  
end
```

con el cual se ha obtenido el siguiente resultado:

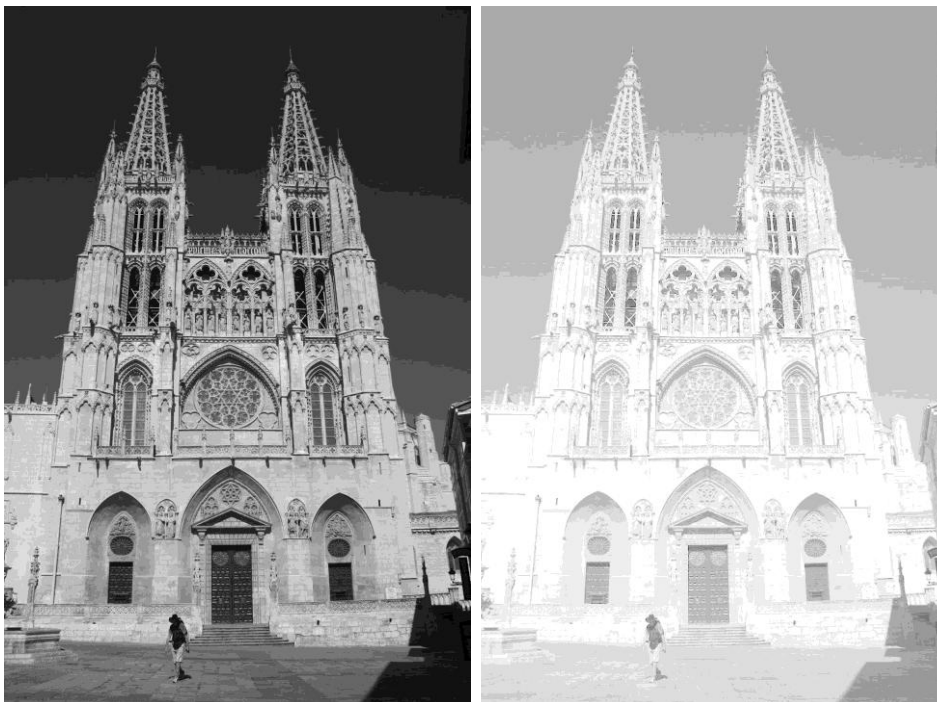


## Compresión de rango

Esta transformación consiste en modificar el rango dinámico de los datos de una imagen comprimiéndola de forma no lineal, realzando en mayor medida los píxeles de menor magnitud frente a los de mayor magnitud. Para ello se ha utilizado el siguiente código en Matlab:

```
function [ salida ] = compresionrango( img, c)  
    salida = im2uint8(mat2gray(c*log10(1+double(img))));  
end
```

con el cual se ha obtenido el siguiente resultado:





## Corrección Gamma

La corrección gamma modifica el rango dinámico de los datos de una imagen mediante una función exponencial. Para ello, se ha utilizado el siguiente código en Matlab:

```
function [ salida ] = correcciongamma( img, correccion, gamma, epsilon )  
    salida =  
    im2uint8( mat2gray( correccion * power( double( img + epsilon ), gamma ) ) );  
end
```

con el cual se han obtenido los siguientes resultados:

