	Universidad de Oviedo 2014.	Tratamiento Digital Imágenes
	Modificaciones sobre el histograma	

## Contents

- [Histogram equalization & Histogram Matching](#)
- [Ecuálización del histograma](#)
- [Histogram matching](#)
- [Compensación del sesgo del Color \(color bias\). Balance contrast enhancement technique \(BCET\)](#)

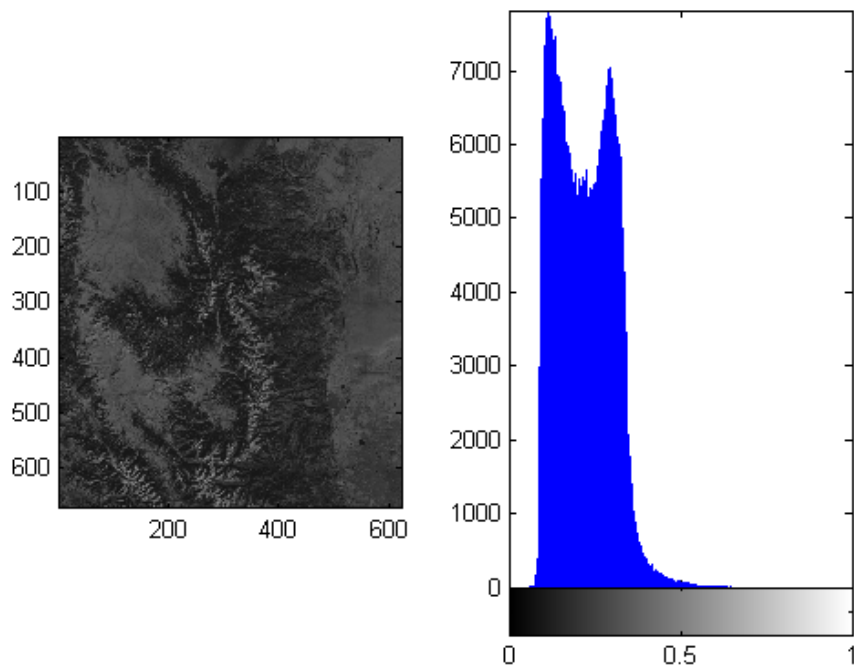
## Histogram equalization & Histogram Matching

Es una técnica automática de mejora del contraste que trata de encontrar un histograma para la imagen de entrada que sea igual al de una distribución uniforme. En esta situación se supone que el contraste sería máximo puesto que cada nivel de todo el rango dinámico disponible estará ocupado por la misma cantidad de píxeles y por tanto se maximizarán las diferencias entre ND's en la imagen. La técnica de ecualización del histograma sirve de paso intermedio para la denominada correspondencia de histogramas, en la que se busca "igualar" (o al menos hacer semejantes los histogramas de dos imágenes).

```
clear, close all
```

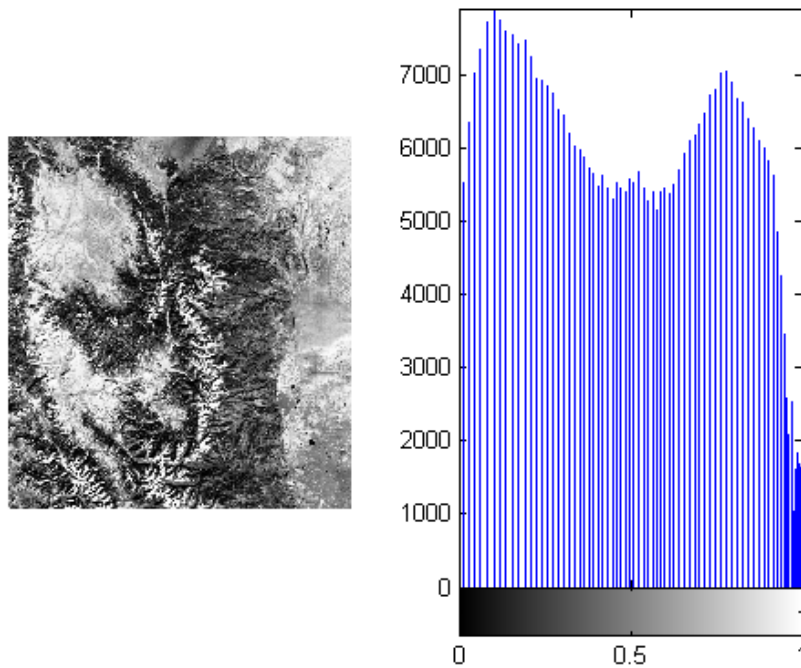
Tomemos la banda 8 del sensor pancromático de la plataforma Landsat.

```
I=imread('B8_ColoradoRED.tif');  
I=im2double(I(:,:,1));  
subplot(1,2,1)  
subimage(I)  
subplot(1,2,2)  
imhist(I(:,:,1))
```



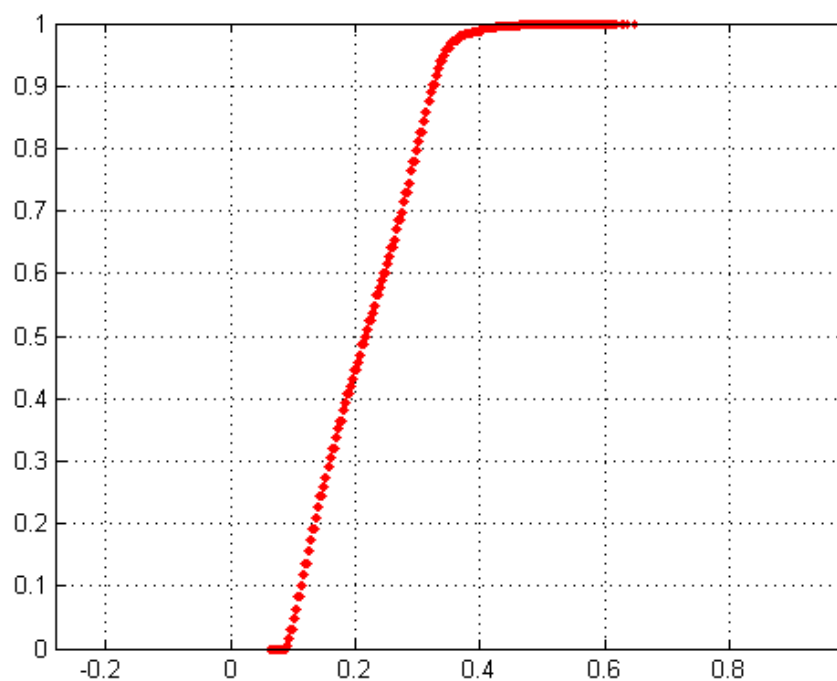
### Ecualización del histograma

```
J=histeq(I,256);  
figure  
subplot(1,2,1)  
imshow(J)  
subplot(1,2,2)  
imhist(J)  
axis tight
```



Podemos ver la función de transferencia haciendo

```
figure  
plot(I(:), J(:), '.r')  
axis equal  
grid on
```



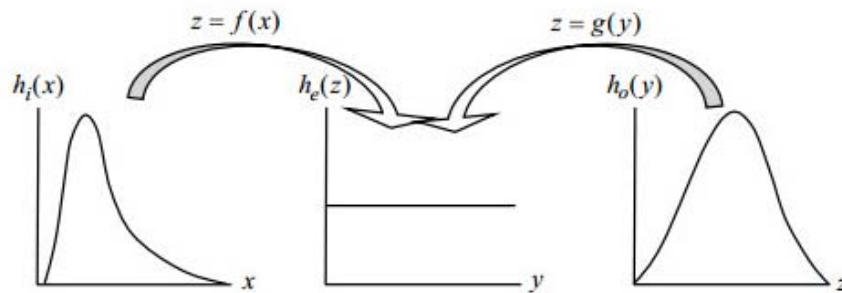
En la práctica el contraste obtenido es demasiado alto puesto que las escenas naturales tienden a seguir histogramas más

cercanos al histograma normal. Además el histograma es sólo aproximadamente uniforme puesto que la perfecta adecuación se podría conseguir en el caso de imágenes "no discretas" (continuas).

## Histogram matching

El histograma uniforme puede servir de puente para conseguir cualquier otro histograma que se desee en la imagen. Este proceso se denomina histogram matching (correspondencia de histogramas) y se lleva a cabo en dos fases

```
figure
IM=imread('HistogramMATching.jpg');
imshow(IM)
```

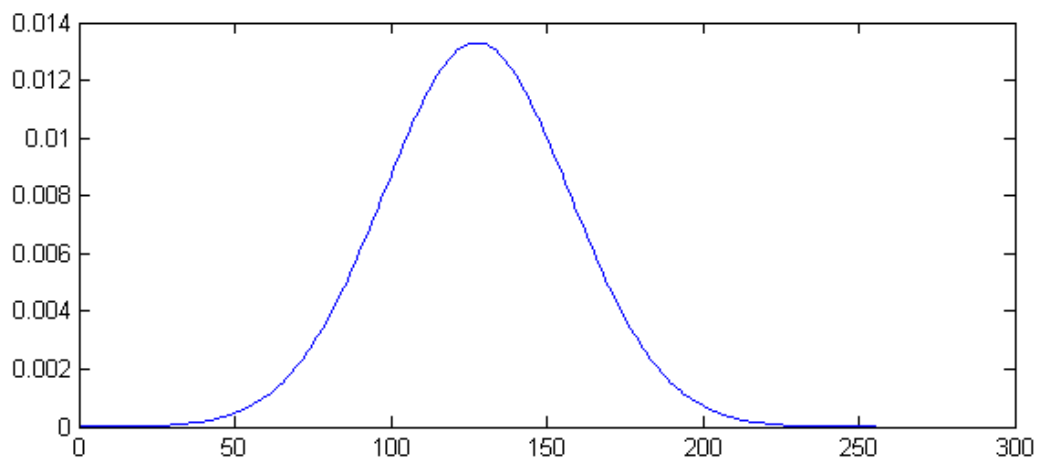


$$z = g(y) = f(x)$$

$$y = g^{-1}(z) = g^{-1}f(x)$$

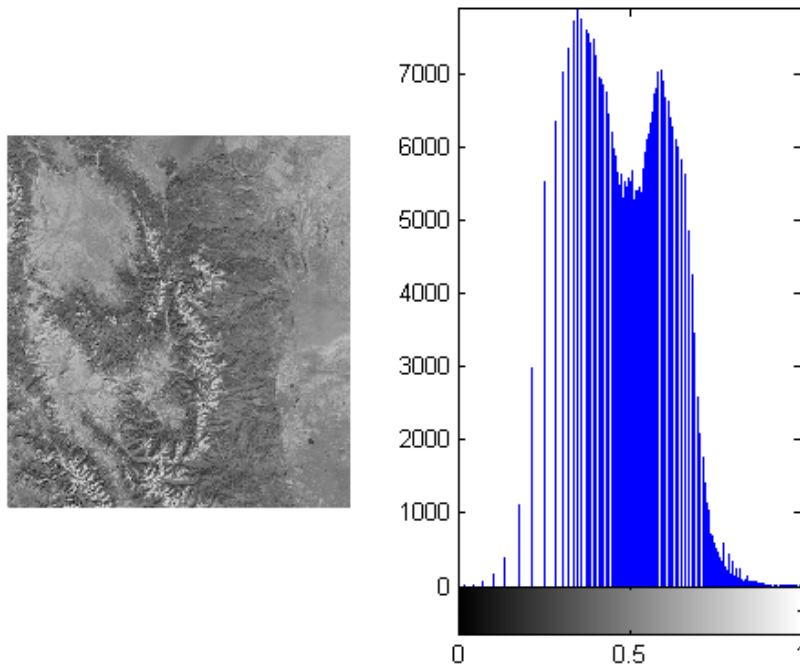
En MATLAB esto puede llevarse a cabo mediante la misma función `histeq` que permite introducir el histograma de una imagen buscada. Busquemos por ejemplo un histograma normal:

```
X=[0:255];
Y= normpdf(X,255/2,30);
plot(X,Y)
```



Ahora veamos que es capaz de hacer el comando `histeq` con nuestra imagen:

```
Jn=histeq(I,Y);
figure
subplot(1,2,1)
imshow(Jn)
subplot(1,2,2)
imhist(Jn)
axis tight
```



Puede verse claramente que intenta asemejarse al histograma normal con una mayor densidad en la zona central aunque sin llegar a perder su naturaleza bimodal.

### Compensación del sesgo del Color (color bias). Balance contrast enhancement technique (BCET)

Es habitual que las distintas bandas sean recogidas por distintos sensores dentro del mismo instrumento. La sensibilidad de los mismos varía a causa de los filtros y de otros factores. El resultado es que el brillo de alguna de las bandas es significativamente diferente al de otras y esto provoca una desviación del color que produce una imagen muy poco natural. Para compensar este efecto sería necesario que todas las bandas tuvieran el mismo rango y valor medio. Es posible para ello plantear una función parabólica definida por 3 componentes que mantenga la forma básica del histograma y que a la vez ajuste el rango y el valor medio de la imagen. La función de transferencia básica será:

$$y = a \cdot (x - b)^2 + c$$

Los valores de los coeficientes se pueden deducir con cierta facilidad y son los siguientes. Sea  $l$ ,  $h$  y  $e$  los valores mínimo, máximo y medio de la imagen de entrada y  $L$ ,  $H$ , y  $E$  los valores homólogos para la imagen de salida. Las expresiones siguientes nos proporcionan los valores de  $a$ ,  $b$  y  $c$  para la función de transferencia parabólica:

$$b = \frac{h^2(E - L) - s(H - L) + l^2(H - E)}{2[h(E - L) - e(H - L) + l(H - E)]}$$

$$a = \frac{H - L}{(h - l)(h + l - 2b)}$$

$$c = L - a(l - b)^2$$

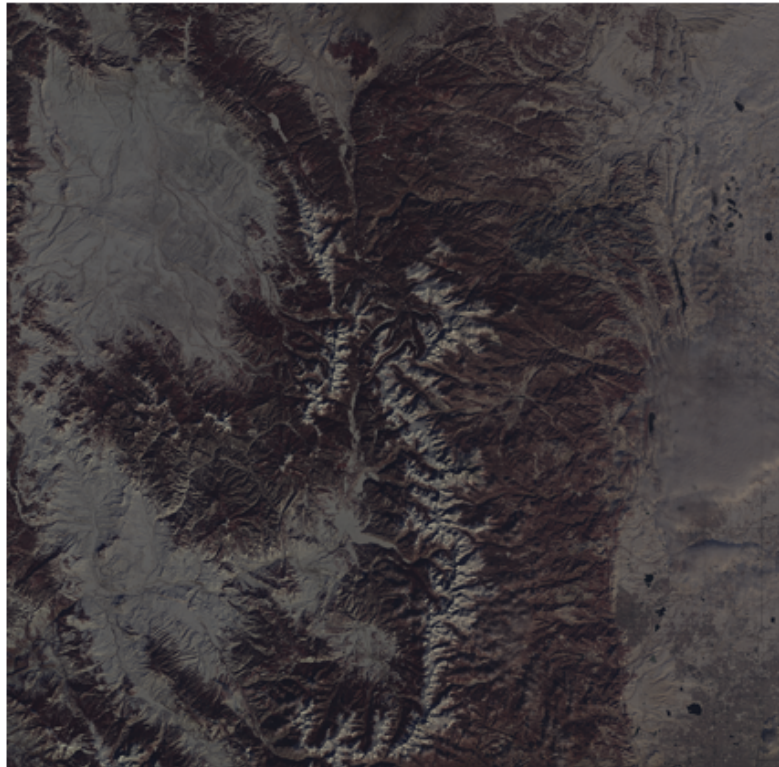
donde el factor  $s$  es la media de los cuadrados de los valores de la imagen de entrada:

$$s = \frac{1}{N} \sum_{i=1}^N x_i^2$$

Veamos un ejemplo con la composición de las bandas 5, 4 y 1 del instrumento OLI de Landsat 8 antes y después del BCET

```
close all,clc
RGB=im2double(imread('ColoradoLandsat8OLI541.tif'));
R=RGB(:,:,1); G=RGB(:,:,2); B=RGB(:,:,3);
figure
imshow(RGB)
title('Original')
```

Original



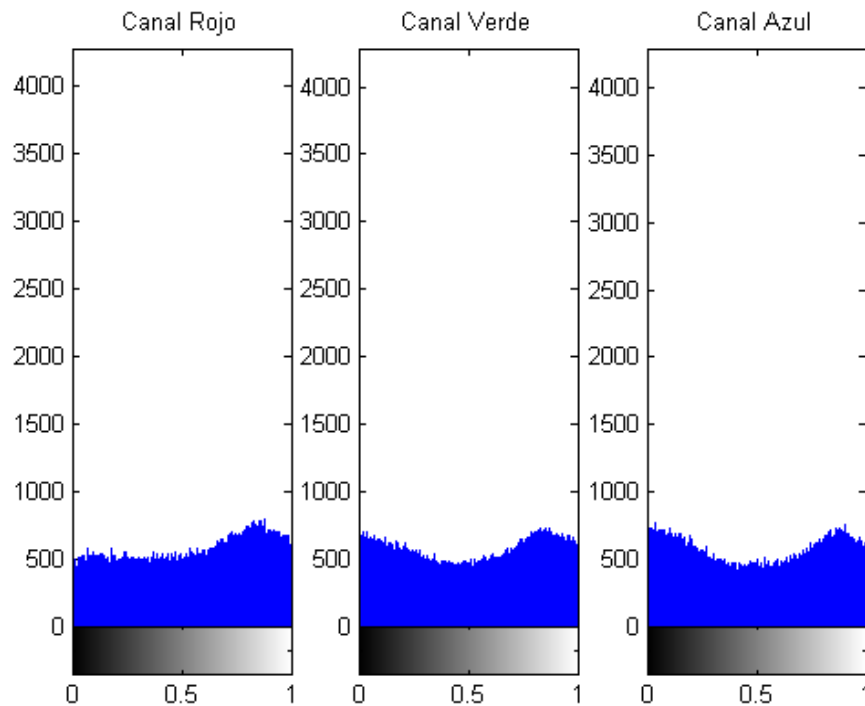
En primer lugar ajustemos el contraste del canal verde

```
Gc = imadjust(G,stretchlim(G,[0.1 0.9]),[]);
B=imadjust(B,stretchlim(B,[0.1 0.9]),[]);
R=imadjust(R,stretchlim(R,[0.1 0.9]),[]);
figure
subplot(1,3,1)
imhist(R)
title('Canal Rojo')
subplot(1,3,2)
imhist(Gc)
```

```

title('Canal Verde')
subplot(1,3,3)
imhist(B)
title('Canal Azul')

```



Ahora tomemos el verde como el canal que fijará el rango y la media de los demás canales.

```
H=max(Gc(:)); L=min(Gc(:)); E=mean(Gc(:));
```

Transformemos el Rojo y el Azul con la transformación BCET

```

R2=R.^2;
s=1/numel(R)*sum(R2(:));
h=max(R(:)); l=min(R(:)); e=mean(R(:));

b=[h^2*(E-L)-s*(H-L)+l^2*(H-E)]/[2*[h*(E-L)-e*(H-L)+l*(H-E)]];
a=(H-L)/[(h-l)*(h+l-2*b)];
c=L-a*(1-b)^2;

% La banda Roja transformada será:
Rc=a*(R(:)-b).^2+c;
Rc=reshape(Rc,size(R));

% Transformación del azul:
B2=B.^2;
s=1/numel(B)*sum(B2(:));
h=max(B(:)); l=min(B(:)); e=mean(B(:));

b=[h^2*(E-L)-s*(H-L)+l^2*(H-E)]/[2*[h*(E-L)-e*(H-L)+l*(H-E)]];
a=(H-L)/[(h-l)*(h+l-2*b)];
c=L-a*(1-b)^2;

```

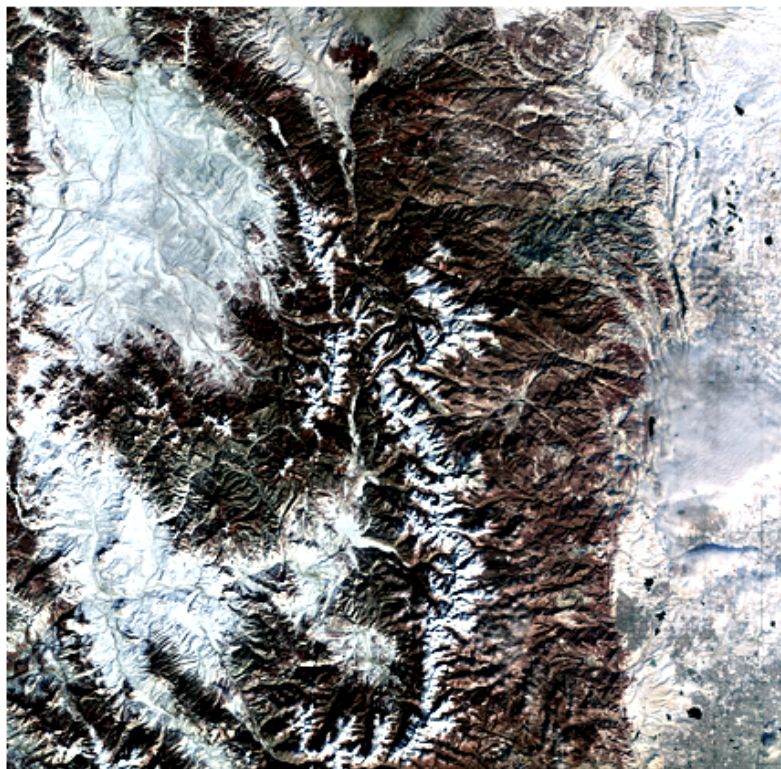


```
% La banda Azul transformada será:  
Bc=a*(B(:)-b).^2+c;  
Bc=reshape(Bc,size(B));
```

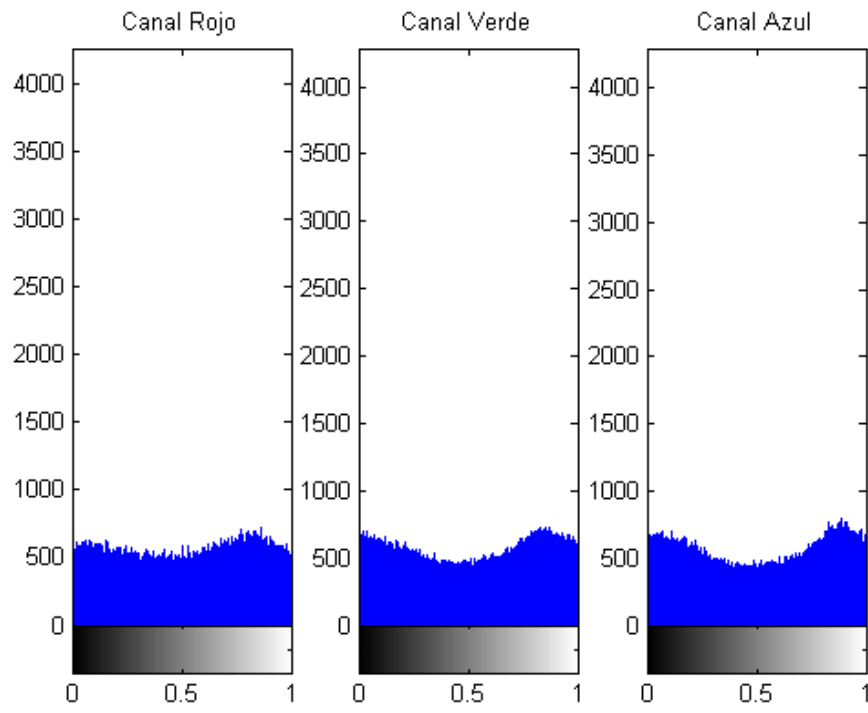
Ahora componemos la imagen final

```
RGBc(:, :, 1)=Rc; RGBc(:, :, 2)=Gc; RGBc(:, :, 3)=Bc;  
figure  
imshow(RGBc)  
title('BCET')  
figure  
subplot(1,3,1)  
imhist(Rc)  
title('Canal Rojo')  
subplot(1,3,2)  
imhist(Gc)  
title('Canal Verde')  
subplot(1,3,3)  
imhist(Bc)  
title('Canal Azul')
```

BCET







Published with MATLAB® R2012b