
```

im=imread("peppers.png");
figure,imshow(im),title("input image");

% Representación 3D de los colores de la imagen
[files,cols,changet]=size(im);
vect=reshape(double(im),files*cols,3);
figure,scatter3(vect(:,1),vect(:,2),vect(:,3),1)
xlabel('R'), ylabel('G'), ylabel('B')
title('espai RGB');

% Creamos k clusters
k=3;
% Devuelve el id del cluster de cada píxel y el centro de los k clusters
% Al introducir estocasticidad al principio pueden variar los resultados en
% cada ejecución
[labels,cluster_ctr]=kmeans(vect,k,"Distance","cityblock");

eti=reshape(labels,files,cols);
figure,imshow(eti,[]),title("imagen etiquetada");

rgb=ind2rgb(eti,cluster_ctr/255);
figure,imshow(rgb,[]),title("imagen rgb");

figure,scatter3(vect(:,1),vect(:,2),vect(:,3),1,labels)
xlabel('R'), ylabel('G'), ylabel('B')
title('espai RGB con etiquetas');

% Qué pasa al usar kmeans con HSV en vez de RGB
hsv=rgb2hsv(im);
hs=hsv(:, :, 1:2);
vect2=reshape(hs,files*cols,2);

k=2;
[labels2,cluster_ctr2]=kmeans(vect2,k,"Distance","cityblock");

eti2=reshape(labels2,files,cols);
figure,imshow(eti2,[]),title("imagen etiquetada"), colormap prism;

% El Hue es cíclico por lo que puede acabar separando los colores rojos
% (hue=0) en dos grupos distintos. K-means no trabaja con ciclos.
figure,scatter(vect2(:,1),vect2(:,2),1,labels2)
xlabel('H'), ylabel('S')
title('espai Hue-sat');

% Podemos transformar (hue, sat) a (sat*sin(hue), sat*cos(hue))
hs2=hsv(:, :, 1:2);
vect2=reshape(hs2,files*cols,2);
vect3=vect2;

% Multiplicamos por 2pi para pasar de [0,1] a [0,2pi]
for i=1:262144
    vect3(i,1)=vect2(i,2)*sin(vect2(i,1)*2*pi);

```

```

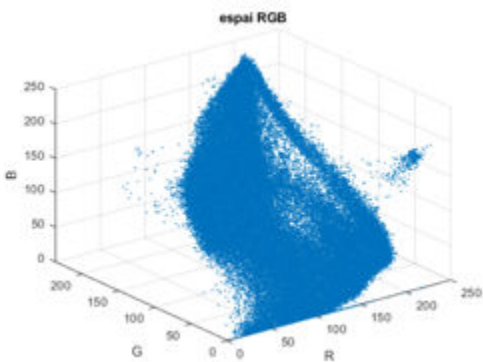
    vect3(i,2)=vect2(i,2)*cos(vect2(i,1)*2*pi);
end

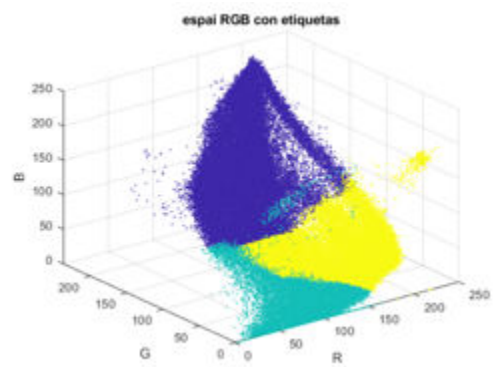
k=2;
[labels3,cluster_ctr3]=kmeans(vect3,k,"Distance","cityblock");

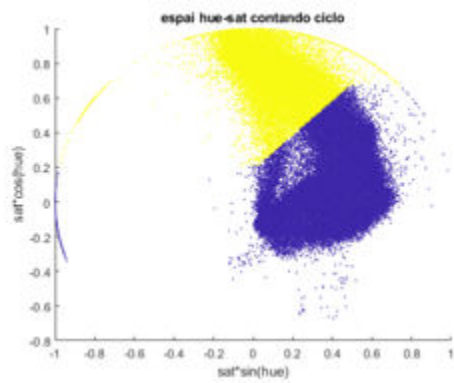
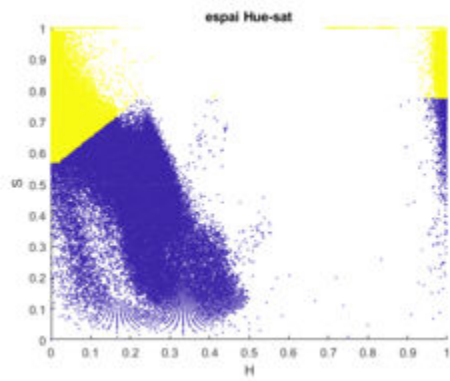
eti3=reshape(labels3,files,cols);
figure,imshow(eti3,[]),title("imagen etiquetada"), colormap prism;

figure,scatter(vect3(:,1),vect3(:,2),1,labels3)
xlabel('sat*sin(hue)'), ylabel('sat*cos(hue)')
title('espai hue-sat contando ciclo');
clear;

```





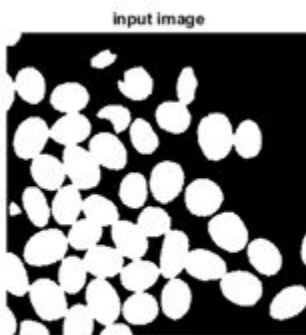
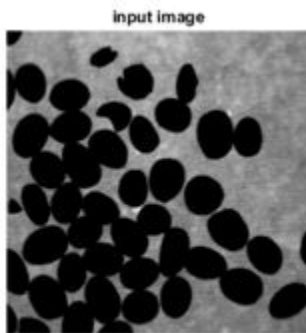


Ejercicio

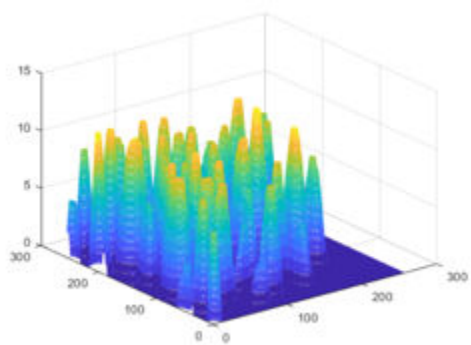
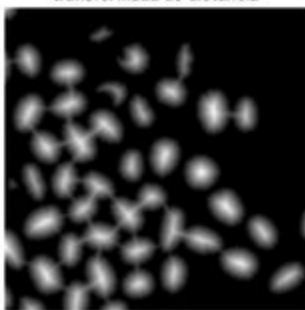
```
im=imread("cafe.tif");
figure,imshow(im),title('input image');
imbin=im < 3;
figure,imshow(imbin),title('input image');

% Observamos que la imagen tiene dos picos altos sin mínimos lo que hace
% difícil la inundación.
td=bwdist(~imbin);
figure,imshow(td,[],title('transformada de distancia'));
ee=strel('disk',4);
filt=imopen(td,ee);

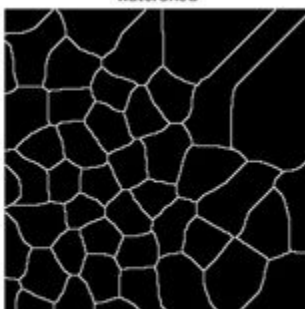
figure,mesh(filt);
segm=watershed(-filt);
figure,imshow(segm==0),title('watershed');
res=~im & segm;
figure,imshow(~res),title('cafes separados');
```



transformada de distancia



watershed



cafes separados



Published with MATLAB® R2024a