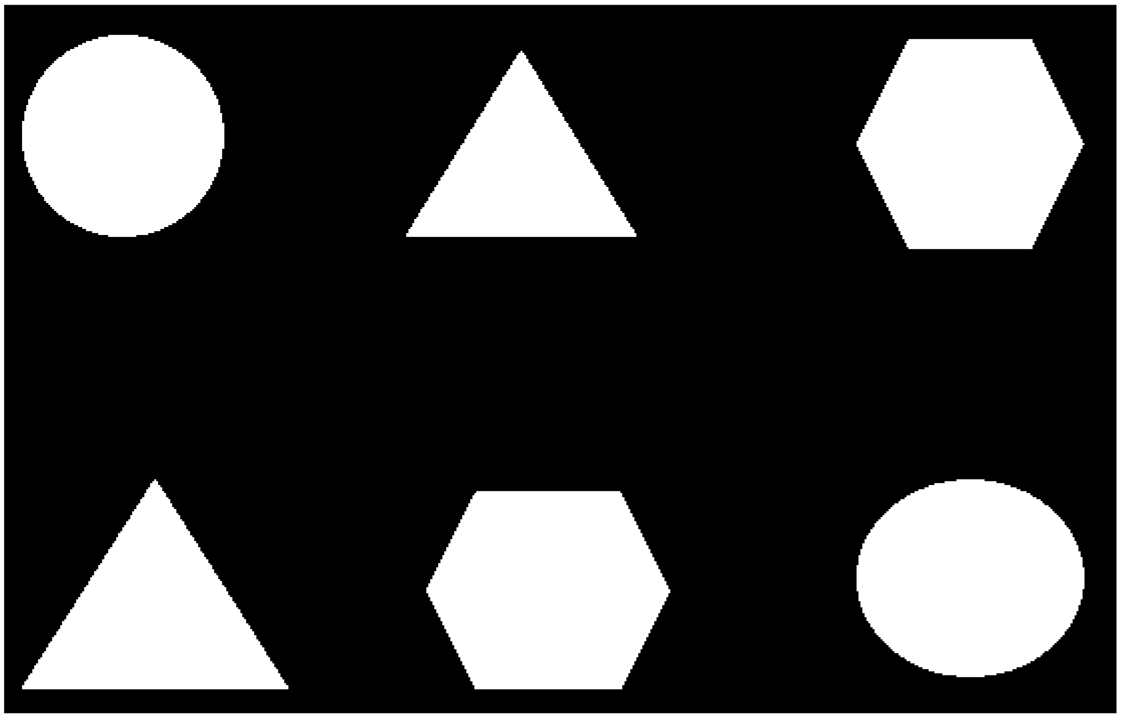
Entrega 8 - 18/04

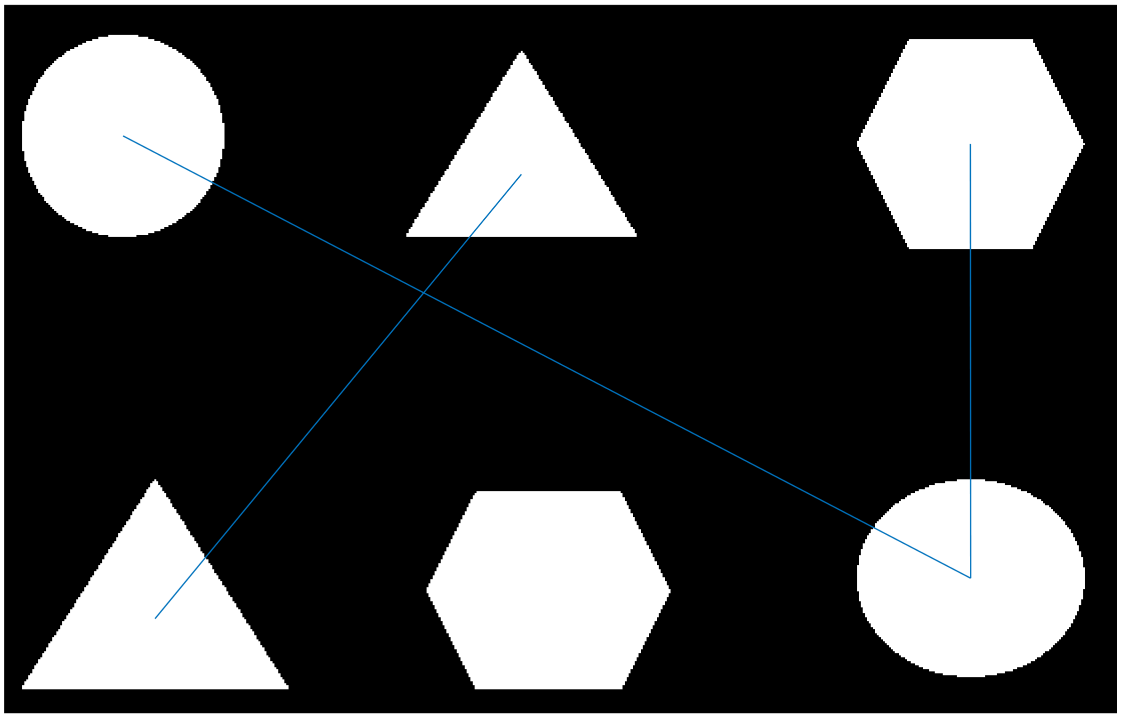
Transformada de Fourier: Convolució amb sinus.

```
I = rgb2gray(imread("figures.png"));
BW = I < 200;
imshow(BW);
```



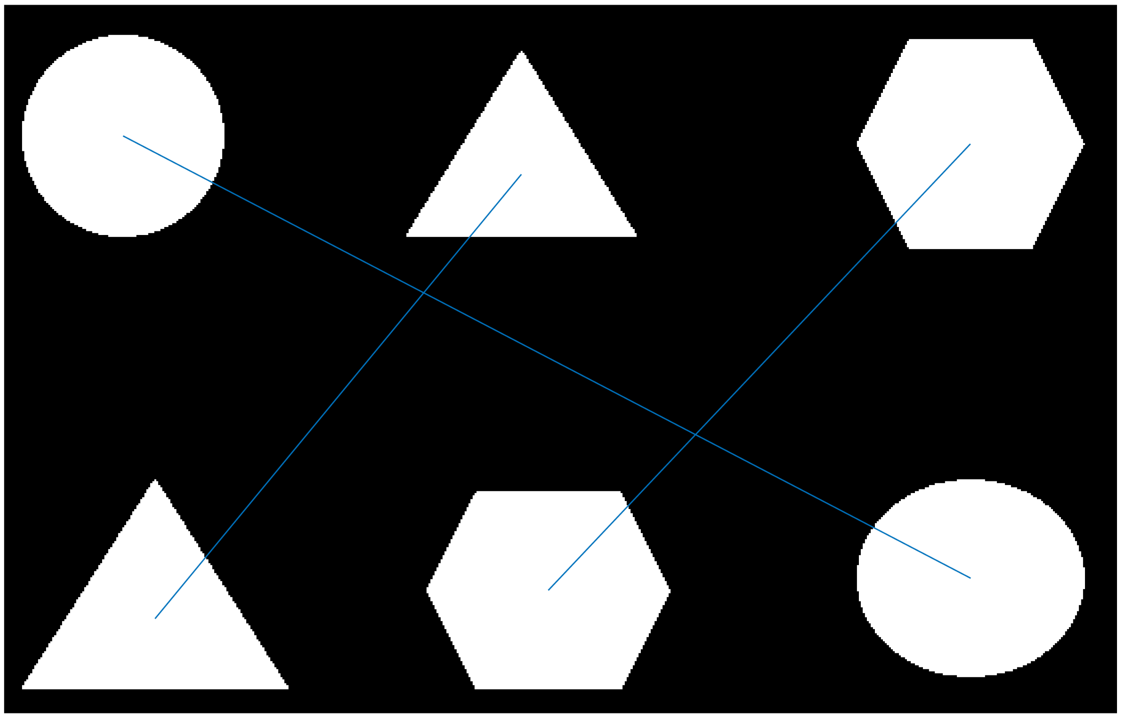
```

BWU = BW;
BWU(end/2:end,:) = 0; %imshow(BWU);
BWD = BW;
BWD(1:end/2,:) = 0; % imshow(BWD);

% segmenta d'esquerra a dreta
CCU = bwconncomp(BWU);
CCD = bwconncomp(BWD);

propsU = regionprops('table',CCU,'Centroid','BoundingBox','Circularity');
propsD = regionprops('table',CCD,'Centroid','BoundingBox','Circularity');

numObjU = CCU.NumObjects; % num obj trobats a la pilla d'adalt
numObjD = CCD.NumObjects; % num obj trobats a la pilla d'abaix

% Primera prova
```

```
FU = propsU.BoundingBox(:,3)./propsU.BoundingBox(:,4); % ratios component a component
FD = propsD.BoundingBox(:,3)./propsU.BoundingBox(:,4);
% normalitzar
FU = FU./max(FU);
FD = FD./max(FD);


% quina figura te el ratio més proper
Assig = dsearchn(FD,FU);

imshow(BW);
for i=1:numObjU
    line([propsU.Centroid(i,1), propsD.Centroid(Assig(i),1)],[propsU.Centroid(i,2), propsD.Cent
end
```

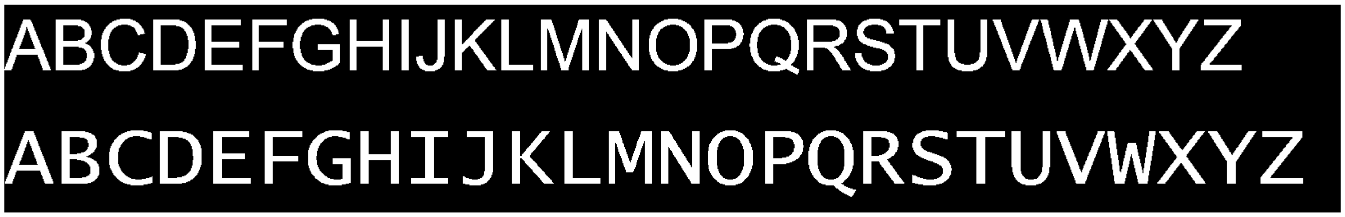


```


% Segona prova
FU = [propsU.Circularity]; % ratios component a component
FD = [propsD.Circularity];
% normalitzar
FU = FU./max(FU);
FD = FD./max(FD);


% quina figura te el ratio més proper
```

```
Assig = dsearchn(FD,FU);

imshow(BW);
for i=1:numObjU
    line([propsU.Centroid(i,1), propsD.Centroid(Assig(i),1)],[propsU.Centroid(i,2), propsD.Cent
end
```



```
J = rgb2gray(imread("Abecedari.png"));
BW = J < 200;
imshow(BW);
```



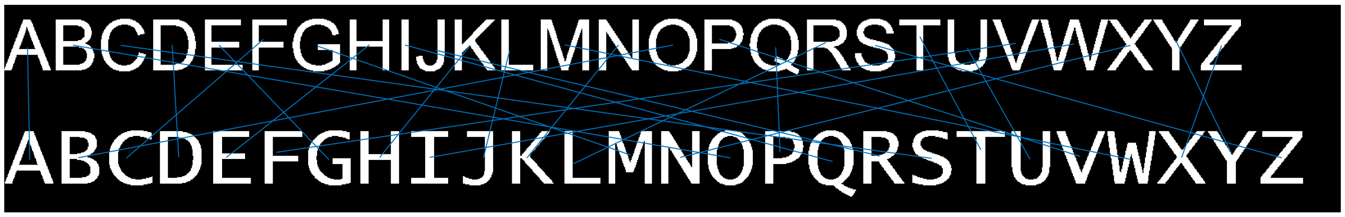
```
BWU = BW;
BWU(end/2:end,:) = 0;
```

```
Warning: Integer operands are required for colon operator when used as index.
```

```
BWD = BW;
BWD(1:end/2,:) = 0;
```

```
Warning: Integer operands are required for colon operator when used as index.
```

```

% segmenta d'esquerra a dreta
CCU = bwconncomp(BWU);
CCD = bwconncomp(BWD);

propsU = regionprops('table',CCU,'Centroid','BoundingBox','Circularity');
propsD = regionprops('table',CCD,'Centroid','BoundingBox','Circularity');

numObj = CCU.NumObjects; % num obj trobats a la pilla d'adalt
FU = [propsU.Circularity]; % ratios component a component
FD = [propsD.Circularity];

% normalitzar
FU = FU./max(FU);
FD = FD./max(FD);

A = zeros(numObj,numObj);
for i=1:numObj
    for j =1:numObj
        A(j,i) = norm(FU(i,:) - FD(j,:));
    end
end

Assig = zeros([numObj,1]);

% realitzem l'aparellament i eliminem les candidats aparellats
for k = 1:numObj
    [Amins,idx] = min(A); % minims de cada columna i la fila
    [Amin,Ai] = min(Amins); % minim global i la columna
    Aj = idx(Ai); % trobem fila i columna del minim
    Assig(Ai) = Aj; % assigem l'aparellament
    A(Aj,:) = Inf; % elimnar l'aparellament a la matriu de distancies
    A(:,Ai) = Inf;
end

imshow(BW);
for i=1:numObjU
    line([propsU.Centroid(i,1), propsD.Centroid(Assig(i),1)],[propsU.Centroid(i,2), propsD.Cent
end
```

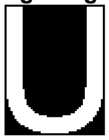


```
K = imread("head.png");
im = imresize(K,0.5);

rect = propsU.BoundingBox(21,:);
im = BWU(rect(2)-1:rect(2)+rect(4),rect(1)-1:rect(1)+rect(3));
```

```
Warning: Integer operands are required for colon operator when used as index.
Warning: Integer operands are required for colon operator when used as index.
```

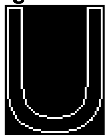
```
figure, imshow(im), title("imatge original");
```

imatge original



```

cont = xor(im, imerode(im, strel("disk",1)));
figure, imshow(cont), title("imatge contorns")
```

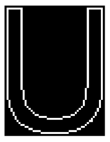
imatge contorns



```
[fila, col] = find(cont,1);
B = bwtraceboundary(cont, [fila,col], 'E');

% coordenades centrades
mig = mean(B);
Bc = B - mig; % traslladem a l'origen
s = Bc(:,1) + Bc(:,2) * 1i; % optional: passem a complexes
```

```
% Transformacio de fourier
z = fft(s);

ss = ifft(z);
aux = zeros(size(im));
files = round(real(ss) + mig(1)); % obtenim les coordenades i tornem a situar al centre
cols = round(imag(ss) + mig(2));
aux(sub2ind(size(aux), files,cols)) = 1; % posa els pixels del ss a 1
figure, imshow(aux), title("Imatge recuperada tots els components");
```

Imatge recuperada tots els components



```
N = 50;
zz = z;
zz(N+1:end-N) = 0;
ss = ifft(zz);
aux = zeros(size(im));
files = round(real(ss) + mig(1)); % obtenim les coordenades i tornem a situar al centre
cols = round(imag(ss) + mig(2));
aux(sub2ind(size(aux), files,cols)) = 1; % posa els pixels del ss a 1
figure, imshow(aux), title("Imatge recuperada, 50 components");
```

Imatge recuperada, 50 components