

```

function [alfa,CD,JN]=afegirlletra
(alfal,C,JS,LN,Cnova,Vnova,n)
% Explicar les variables d'entrada en ordre. Despres les
variables de
% sortida.

%Funcio que afegeix una lletra al cataleg. Afegeix tan la
grafia (vector de caracter) com la
%imatge com els quocients de valors singulars.

%Variables de entrada

%n es la posicio on anira la lletra

    %Velles
%Grafies del cataleg
%Quocients del cataleg
%Imatges del cataleg
    %Noves

%LN es la grafia de la nova lletra 'Nova lletra'
%Cnova es el quocient de valors singulars de la nova lletra
%Vnova la matriu de la nova lletra

%Variables de sortida.

%La grafia amb la posicio correcta
%Els quocients amb la posicio correcta
%El matriu de imatge de la nova lletra.

[~,b]=size(alfal);

if n==1 % Per si la letra coincide con el primer valor (Per
example la lletra 'a' no afegida)
    h=C(n:end); %matrius de coeficients svd
    lk=JS(n:end); % matriu de vectors per veure la imatge
CD=[Cnova,h];
alfa=[LN,alfal(n:end)];
JN=[Vnova,lk];
elseif n==(b+1) % Per si la lletra coincideix amb l'ultim
valor (exemple la ' Z')
    mn=JS(1:end);
    gs=C(1:end); %matrius de coeficients svd
    CD=[gs,Cnova];
    JN=[mn,Vnova];
    alfa=[alfal(1:end),LN];

else
h=C(n:end);
sd=C(1:(n-1));
mt=JS(n:end);
rs=JS(1:(n-1));

```

```
CD=[sd,Cnova,h];  
JN=[rs,Vnova,mt];  
alfa=[alfa(1:n-1),LN,alfa(n:end)];  
end
```

```

%%%Catalog.

% El scrip Catalog creara el catalog per despres fer
l'analisi.I guardarem
% tot el necessari fer l'analisi

alfa=['a':'z','A':'X','0':'6','9','(',')'];
%'A':'X','0':'9','(',')'] %Surten totes les lletres menys la Y
i Z mayuscula.
%alfa=
['a':'v','x':'z','A':'J','L':'P','R':'V','0':'9','(',')'];
%'A':'X','0':'9','(',')'] %Surten totes les lletres menys la Y
i Z mayuscula.
load('quijote.mat','V','d');
% dni=d;
% % eliminar rarezas
% tac=['A':'X','0':'9','(',')'];
% tap=length(tac);
% for i=1:tap
%     rap=tac(i); %Por ejemplo todo sin mayusculas ni nada
%     d=strrep(d,rap,'');
%     dni=strrep(dni,rap,'*');
% end % en el texto
%
% gep=strfind(dni,'*');
% %Eliminar los no deseados
% g=0;
% for k=gep
%     for counter=k-g+1:size(V,2)
%         V{counter-1}=V{counter};
%     end
%     V(end)=[];
%     g=g+1;
% end

%Formació del catalog.
me=length(alfa);
for i=1:me
    te=strfind(d,alfa(i));
    met=length(te);
    if met>10
% comprobar(V,te(1:10),10,10); %Escollim la que ens vagi
millor de cada lletra
    else
        % comprobar(V,te,10,10);
    end
end
%la i la segunda% la j la segunda% la l la tercera% la n la
quinta% la q la segunda
%la s la segunda% % la t la tercera% la u la cuarta% la w la
tercera% la I la quarta
% la K la segunda% la M la segunda% la N la segunda% la R la
segunda% la T la segunda
% la W la segunda % Les de mes amb la primera serveix

%TOTES LES IMATGES
mt=

```

```

['a':'h','k','m','o':'p','r','v','x':'z','A':'H','J','L','O':'
Q','S','U':'V','X','O':'6','9','(',')'];% 'A':'H','J','L','O':'
Q','S','U':'V','X','O':'9','(',')'];
depn=length(mt);
for i= 1:depn
    te=strfind(d,mt(i));
    cap1=strfind(alfa,mt(i));
    car=te(1);
    Cat{cap1}=V{car};
end
%Lletres que no agafo la primera solucio

mant='ijlnqstuwIKMNRTW';    %IKMNRTW';
m=[2,2,3,5,2,2,3,4,3,4,2,2,2,2,2,2]; %4,2,2,2,2,2,2,2];
depd=length(mant);
for i= 1:depd
    te=strfind(d,mant(i));
    cap=strfind(alfa,mant(i));
    dal=m(i);
    cal=te(dal);
    Cat{cap}=V{cal};
end

% [a,b]=size(Cat);
% nan=1:b;
% comprobar(Cat,nan,10,10)

load('quijote300.mat'); %Nova solucio!!
%Canviar lletres por numeros.
din=length(d);
for i=1:din    % Longitud de la solucio
    aux=d(i); %Lletra per lletra
    aux1=strfind(alfa,aux);
    solucio(i)=aux1; %Es la solucio bona
end
%Obtenir el numero minim de costa per saber fins on podem fer
el calcul de
%valors singulars
for k=1:din
f=size(V{k});
fe(k)=min(f); %Minim dels dos costats de la matriu

end
h=min(fe); %Minim de totes les matrius

%Tenim que el minim es 19. Llavors podrem fer fins a 18.

%Quocients de valors singulars del Cataleg

for i=1:me
    vs=svd (Cat{i}); %Valores singulares
    C{i}=vs(2:7)/vs(1); %Quocients de valors singulars .Podem fer
    8 despres jugarem amb aquest valor
    %De les solucions

```

```

end
C=cell2mat(C); %Convertim la matriu celular en matriu abans de
guardar
%Quocients de valors singulars de les lletres.
for i=1:din
    vs=svd (V{i}); %Valores singulares
    qs{i}=vs(2:7)/vs(1); %Quocients de valors singulars .Podem fer
    8 despres jugarem amb aquest valor
    %De les solucions
end

save
('Estadistica_quijote300prueba.mat','d','qs','V','solucio','al
fa','C','Cat')

    % Guardam tot el necessari. % alfa es el vector de lletres
del Cataleg.

    % d es la solucio pero amb lletres (en principi no ens fera
falta).
%qs es el quocients de valors singulars de cada imatge
    % V es la matriu de imatges
    % 'solucio' es la solucio amb numeros
    %alfa es el vector de lletres del Cataleg
    %C es la matriu de quocients de valors singulars del cataleg
    %Cat es la imatge de les lletres del Cataleg.

```

```

%Funcio Comprobar

%Ens ajuda a verificar que les imatges siguin les correctes.

%Divideix la finestra gràfica en varies subfinestres de
dimensio (axb). On cada subfinestra
%correspon a una imatge. Les imatges poden ser de tamanys
diferents.
% Les imatges es mouen de esquerra a dreta i de dalt a baix.
Quan les
% imatges insertades a la finestra arriba al maxm genera una
nova
% finestra (figure). I aixi fins que mostri totes les imatges
que volem.

% Exemple.

% Tenim una matriu celular de 1000 imatges i volem que
llegeixi les 400
% primeres.

% Variable de entrada matriu de imatges (V)
%de (1:400)
%a=20; b=5; generar 4 figures cadascuna de elles tindra 100
imatges (20
%files i 5 columnes). llegides de esquerra a dreta i de dalt
a baix. Aixi
%la imatge (de=124) estarà a la figura 2. Fila 2
columna 4.

%Aquesta funció és molt útil per comprobar cada imatge esta al
lloc
%correcte.

% Variables:

% V es la matriu celular que conte les diferents imatges, les
imatges poden
% tenir tamanys diferents.

% de es el vector que indica dins del a la matriu celular que
imatges volem
% llegir.

%
%a=vertical. (nombre de filas)
%b=horitzontal (nombre de columnas)

%

function comprobar (V,de,a,b)
m=1;
g=1;

for k=de

```

```

        if m==(a*b+1)
            figure %Mostrar imagen. Quan la lletra arriba a la
lletra 100 ens mostra la imatge.
m=1;%Reinicio a per tal de que comencem a contabilitzar de
nou.

g=1; % Reinicio para añadir la letra que falta! (Aquest pas es
impresindible per afegir la primera lletra de les noves
imatges(101,201,301....)
t=k; % Li dic que em guardi la regio de la imatge que podiam
perdre (101,201,301,....)
g=g+1; % Para que entre dentro del bucle
        elseif g==2 %

subplot(a,b,(g-1));imshow(1-V{t}) %imagen de 101,201,301,401,
PODEM POSAR 1 EN COMPTES DE (G-1) es el mateix
        subplot(a,b,(m)); imshow(1-V{k}) % Imagen que falta para
mostrar que començara sempre en m=2 ja que la m=1 salta del
primer bucle

g=g+1;

        else
            subplot(a,b,m); imshow(1-V{k})% las primeras letras
que de 1 al a*b
            end
            m=m+1; %Saltar la imagen que no se ve
        end
            figure %Mostrar la ultima imagen.

```

```

% Scrip que genera aleatoriament solucions estadístiques per
comprobar una
% millora.
tblanca=0.6:0.05:1;
r0b=0.6:0.025:0.8;
tnegra=1;
Nb=1:10;
r0n=0.1:0.025:0.3;
Nn=1:10;

aux1=length(tblanca);
aux2=length(r0b);
aux4=length(Nb);
aux5=length(r0n);
aux6=length(Nn);
for t=1:100

    aux7=randperm(aux1,1);
    aux8=randperm(aux2,1);
    aux10=randperm(aux4,1);
    aux11=randperm(aux5,1);
    aux12=randperm(aux6,1);
    aux13(t)=tblanca(aux7);
    aux14(t)=r0b(aux8);
    tnegra=1;
    aux15(t)=Nb(aux10);
    aux16(t)=r0n(aux11);
    aux17(t)=Nn(aux12);
    Mejoras(aux13(t),aux14(t),aux15(t),aux16(t),aux17(t),1);
    Estadistica
    nueva(t)=[encertsrel(20)]
end

%Las mejores iteraciones son 60 86 66 y 2;

% tblanca=[0.8,0.6,0.65,0.65];
% r0b=0.6500    0.7750    0.6000    0.8000;
%      NB 6      5      3      6
%      r0n 0.2750    0.2500    0.1500    0.2750
%
%      Nn 8      5      1      8

    plot(nueva,'b')
hold on
    title('Percentatge de encerts vs Variables modificades');
    xlabel('Variables modificades');
    ylabel('Percentatge de encert');
    hold off
figure

    save
('nuevas.mat','nueva','aux13','aux14','aux15','aux16','aux17')
;

```



```

function [deteccion]=detec(probar,f,nq)
% nq es la quantitat quocients que definirem
% f es el primer valor de els pesos on augmentaran linialment.
%load('Estadistica.mat','d','qs','V','alfa','C','Cat')
load(probar,'d','qs','V','alfa','C','Cat')
% Variables de tinta blanca tblanca [1-0,6] r0b=[0,6-0.8] Si
el text te
% poques taques valora fer Nb=0 (No aplicar flux blanc).
[o,p]=size(V);

for k=1:p
    qs{k}=qs{k}(1:nq); % nq es la cantidad de valores propios
que vamos a utilizar
end

%ANALISIS

Catalogo=C(1:nq,:);
[aCat,bCat]=size(Catalogo);

%ETAPA 1

for k=1:p

    Mqs=(qs{k}*ones(1,bCat)); %N=#columnes del cataleg, Mqs
es matriu que te N
    %columnes amb qs repetit

%Etapa 2 Previa anàlisi afegir pesos
    wp=sqrt(linspace(f,1,nq));
    % pesos alternatius que decreixen mes rapidament que l'arrel
quadrada
    %wp=logspace(1+(f-1)/100,0,nq);
    % final del vector de pesos exponencial alternatiu
    wn=(wp'*ones(1,bCat));

%Etapa 3 Diferencia con pesos afegidos
    qsDif=(Catalogo.*wn-Mqs.*wn);
%Etapa 3

    prodqsDif=(qsDif')*qsDif;

%ETAPA 4

dists=diag(prodqsDif); % vector fila amb la diagonal de
prodqsDif, te
% les normes de les distancies a la 2

%ETAPA 5

[a,b]=min(dists); % vector fila amb la diagonal de prodqsDif,
te
% les normes de les distancies a la 2

deteccion(k)=b;
end

```



```
%Eliminar una cell array buida.
function V=eliminarCellarray(V,gep)
%gep son es el vector que indica els index que volem eliminar
V es la cell
%array inicial. la sortida sera la cellarray.
g=0;
for k=gep
    for counter=k-g+1:size(V,2)
        V{counter-1}=V{counter};
    end
    V(end)=[];
    g=g+1;
end
```

```

%La funcio/scrip estadistica se encarregara de treure
estadistiques de les
%zones detectades pel metode.

letm='no'; %Dibujar los errores de letras minusculas
let='si'; %Dibujar los errores de todas las letras
%La variable modnq si volem fer l'analisi de nq i modpesos es
si volem
%fer analisis de pesos.
%probar='Estadistica_minusculas.mat';

Mejoras
probar='Estadistica_quijote300_mejorado.mat';
load(probar,'solucio','V','d','alfa','lim');

modnq='si';
modpesos='no';
if modnq=='si' %Si volem modificar la nq....
    f=1;
    for i=1:lim-1

        nq=i;
        deteccion=detec(probar,f,nq);
        diferencies=deteccion-solucio;
        posencerts=(diferencies==0);
        encerts(i)=sum(posencerts);
        encertsrel(i)=encerts(i)/length(solucio);
    end
    plot(encertsrel,'b')
    hold on
    title('Percentatge de encerts vs Nombre de quocients
singulars');
    xlabel('quocients singulars');
    ylabel('Percentatge de encert');
    hold off
    figure
    end
    if modpesos=='si'
        nq=8;
        r=1:50:201;
        contar=1;
        for i=r

            f=i;
            deteccion=detec(probar,f,nq);
            diferencies=deteccion-solucio;
            posencerts=(diferencies==0);
            encerts(contar)=sum(posencerts);
            encertsrel(contar)=encerts(contar)/length(solucio);
            contar=contar+1;
        end

        plot(encertsrel,'b')
        hold on
        set(gca,'XTick',[1:5]);
        set(gca,'XTicklabel',r);
        title('Porcentatge de encerts vs Variacio de pesos');

```

```

xlabel('Variació de pesos (Pes inicial)');
ylabel('Porcentatge de encert');
hold off
figure
end

%quines lletres fallen

error1=find(posencerts==0); %Matriu de errors
letras=solucio(error1); %Es la solucio Real.
err=deteccion(error1); % Es la solucio que hem dona el meu OCR
mir=[letras;err]; %Solucio real mes OCR
doc=length(letras); %Longitud de la solucio real
h=length(alfa);
mat=zeros(1,h);
for k=1:h
    g=1;
    for i=1:doc
if letras(i)==k
    mat(k)=g;
    g=g+1;
else
end
    end
end

if letm=='si'
mat=(mat./sum(mat))*100;
bar(mat)
hold on
set(gca,'XTick',[1:26])
set(gca,'XTicklabel',
['a';'b';'c';'d';'e';'f';'g';'h';'i';'j';'k';'l';'m';'n';'o';'
p';'q';'r';'s';'t';'u';'v';'w';'x';'y';'z'])
title('Percentatge de errors (minuscules)');
xlabel('lletres');
ylabel('Percentatge de errors');
hold off
figure
else
end

tap=length(alfa);
if let=='si'
mat=(mat./sum(mat))*100;
bar(mat)
hold on
set(gca,'XTick',[1:tap])
set(gca,'XTicklabel',
['a';'b';'c';'d';'e';'f';'g';'h';'i';'j';'k';'l';'m';'n';'o';'
p';'q';'r';'s';'t';'u';'v';'w';'x';'y';'z';'A';'B';'C';'D';'E'
;'F';'G';'H';'I';'J';'K';'L';'M';'N';'O';'P';'Q';'R';'S';'T';'
U';'V';'W';'X';'0';'1';'2';'3';'4';'5';'6';'9';'(';')'])
title('Percentatge de errors de totes les lletres');
xlabel('lletres');
ylabel('Percentatge de errors');
hold off

```

```

figure
else
end

% Analisis de lletra per lletra

[a,b]=sort(mat,'descend'); %Reagrupa les lletres de mes gran
a mes petit.

%Queda amb les lletres que tinc un error significatiu. Per
exemple el mes
%grans que el 3%
sign=a>3; %Fins al percentatge de error que analitzarem
lim=sum(sign); %Sumem per saber fins quina lletra parar
Letrasm=b(1:lim); %Agafem les lletres que tene mes del 2%
Percen=a(1:lim); %Son els percentatges de error de les
lletres amb mes del 3%
LetrasmS=alfa(Letrasm);
for i=1:(length(Letrasm))
for k=1:(length(letras))
if letras(k)==Letrasm(i);
Posicio(i,k)=k;
else
end
end
end

for i=1:(length(Letrasm))
aux86=Posicio(i,:);
aux87=aux86(aux86>0);
ErrLetras{i}=err(aux87); %Cada lletra amb les lletres que
ha detectat erroneament
end
aux90=length(alfa);

for i=1:(length(Letrasm)) % Les lletres que analitzarem

aux88=ErrLetras{i};
aux89=length(aux88);
for k=1:aux90 %Lletra per lletra
g=1; %Reiniciamos la cuenta
for f=1:aux89 %Mirar si es repeteixen en tot el vector
if k==aux88(f);
mal(i,k)=g; % Aquesta matriu la fila es cada lletra y
les columnes els errors de les lletres
g=g+1;
else
end
end
end
end
end
% Exemple mal (1,1). La lletra que analitzem la t i cuantes
"a" detecta

%Per a minuscules
minc='no';
if minc=='si' %Letrasm

```

```

        for i=1:length(LetrasM)
            aux200=mal(i,:);
            aux201=sum(aux200);
malM=aux200./aux201*100;
bar(malM)
hold on
set(gca,'XTick',[1:26])
    set(gca,'XTicklabel',
['a';'b';'c';'d';'e';'f';'g';'h';'i';'j';'k';'l';'m';'n';'o';'
p';'q';'r';'s';'t';'u';'v';'w';'x';'y';'z'])
aux100=LetrasM(i);

aux101='La lletra ";
aux102='" amb un percentatge de error del ';
aux103=round(a(i)); %Percentatge de error ardonit
aux104=' aproximadament (nomes minuscules)';
aux105=[aux101,aux100,aux102,num2str(aux103),aux104];

title(aux105);
xlabel('Lletres');
ylabel('Percentatge que el programa ha detectat malament');
hold off
figure
    end
else
end

    minTot='no';
    tap1=length(alfa);
    if minTot=='si' %LetrasM
        for i=1:length(LetrasM)
            aux200=mal(i,:);
            aux201=sum(aux200);
malM=aux200./aux201*100;
bar(malM)
hold on
set(gca,'XTick',[1:tap1])
    set(gca,'XTicklabel',
['a';'b';'c';'d';'e';'f';'g';'h';'i';'j';'k';'l';'m';'n';'o';'
p';'q';'r';'s';'t';'u';'v';'w';'x';'y';'z';'A';'B';'C';'D';'E';
'F';'G';'H';'I';'J';'K';'L';'M';'N';'O';'P';'Q';'R';'S';'T';'
U';'V';'W';'X';'0';'1';'2';'3';'4';'5';'6';'9';'('';')'])
aux100=LetrasM(i);
aux101='La lletra ";
aux102='" amb un percentatge de error del ';
aux103=round(a(i)); %Percentatge de error ardonit
aux104=' aproximadament (totes les lletres)';
aux105=[aux101,aux100,aux102,num2str(aux103),aux104];

title(aux105);
xlabel('Lletres');
ylabel('Percentatge que el programa ha detectat malament');
hold off
figure
    end
else
end

```



```
%Funcio lectora:  
%Rep el nom del fitxer grafic de la imatge de texte,  
%l'importa a Matlab i el deixa com a una matriu double amb 1  
=blanc 0=negre.  
  
function A=fitxer2matbin(fitxer,format)  
  
A_ini = imread (fitxer,format);  
% Per treballar amb la matriu cal que sigui de doble precisio  
A = im2double(A_ini(:,:,1));
```

```

% Flux de tinta Blanca N passos "Eliminacio de taques negres"

%Flux de tinta blanca

% Funcio per a la restauracio i eliminacio d'imperfeccions de
textos
% escanejats o fotocopiats en blanc i negre. Aquesta funcio
duu a terme tot el
%procediment necessari a traves d'operacions matricials.

%Cal prendre tambe com a paremetre d'entrada el nombre
d'iteracions NB,
%PARAMETRES: tblanc quan valguin 1. No afegirem tinta blanca.
%Si val mes de 1 afegirem tinta i i si es mes petit treure.
(Per afegir negre valors petits).

% els indexs de repartiment de tinta 'rv' (index
% vertical) i 'rh' (index horitzontal) seran iguals, en
principi, de tal manera que
%  $2*rv+2*rh+r0b = 1$ .  $rv=rh$ . Llavors la variable que sera fixada
per l'usuari
% ser el r0b que fitxa el porcentatge de tinta negra que no
flueix en cada
% pas.

% 1=pixel blanc, 0=pixel negre.

% Variables de tinta blanca tblanca [1-0,6] r0b=[0,6-0.8] Si
el text te
% poques taques valora fer Nb=0 (No aplicar flux blanc).

function Afinal=fluxdetintablanca(A,Nb,r0b,tblanca)
% Es busca la mida de la imatge. En tractar-se d'una imatge en
blanc i
% negre, la matriu
%serie bidimensional.
[m,n] = size(A);

rv=tblanca*(1-r0b)/4;
rh=tblanca*(1-r0b)/4;

% Sinicialitzen les matrius auxiliars.
Aux_ext = ones(m+2,n+2);
A_ext = Aux_ext;
A_dalt = Aux_ext;
A_baix = Aux_ext;
A_dreta = Aux_ext;
A_esq = Aux_ext;
%
for i=1:Nb
A_ext(2:m+1,2:n+1) = A;
A_dalt(1:m,2:n+1) = A;
A_baix(3:m+2,2:n+1) = A;
A_dreta(2:m+1,3:n+2) = A;
A_esq(2:m+1,1:n) = A;

```

```
A_ext = rv*(A_dalt + A_baix) + rh*(A_dreta + A_esq)+r0b*A_ext;  
A = A_ext(2:m+1,2:n+1);
```

```
end;
```

```
%A=max(A,An); % aquest max fa que els pixels que eren blancs  
continuin sent blancs
```

```
Afinal=round(A);
```

```

%Flux de tinta negra

% Funcio per a la restauracio i eliminacio d'imperfeccions de
textos
% escanejats o fotocopiats en blanc i negre. Aquesta funcio
duu a terme tot el
%procediment necessari a traves d'operacions matricials.

%Cal prendre tambe com a paremetre d'entrada el nombre
d'iteracions NB,
%PARAMETRES: tblanc quan valguin 1. No afegirem tinta blanca.
%Si val mes de 1 afegirem tinta i i si es mes petit treure.
(Per afegir negre valors petits).

% els indèx de repartiment de tinta 'rv' (index
% vertical) i 'rh' (index horitzontal) seran iguals, en
principi, de tal manera que
%  $2*rv+2*rh+r0b = 1$ .  $rv=rh$ . Llavors la variable que sera fixada
per l'usuari
% ser el r0b que fitxa el porcentatge de tinta negra que no
flueix en cada
% pas.

% 1=pixel blanc, 0=pixel negre.

%Hem de vigilar ja que hi ha mes tinta negra que tinta
%blanca a tots els textos per tan si afegi tinta negra
(tnegre=1.02)
%llavors (afegim un 4% pero amb les iteracions al final acabes
afegim molt...)
%i pot pasar que tot el texte quedi molt enegrit.

%Les variables per tinta negra recomenades son.
%Variables per a tinta negra (tnegra valors petits 4%[tnegra=
1,02] i
%r0n=[0.1-0.3]. Nn=6;

function Afinal=fluxdetintanegra(A,Nn,r0n,tnegra)
%Definim la funcio A_ini perque es una variable especificada
en el subprograma
A_ini=A;
[m,n] = size(A);
% Es busca la mida de la imatge. En tractar-se d'una imatge en
blanc i
% negre, la matriu
%serie bidimensional.

rv=tnegra*(1-r0n)/4; % repartiment per vertical y horitzontal
igual
rh=tnegra*(1-r0n)/4;

% Fem el canvi
A=1-A;

% Sinicialitzen les matrius auxiliars.
Aux_ext = zeros(m+2,n+2);
A_ext = Aux_ext;

```

```

A_dalt = Aux_ext;
A_baix = Aux_ext;
A_dreta = Aux_ext;
A_esq = Aux_ext;

for i=1:Nn %En principi pocs passos

A_ext(2:m+1,2:n+1) = A;
A_dalt(1:m,2:n+1) = A;
A_baix(3:m+2,2:n+1) = A;
A_dreta(2:m+1,3:n+2) = A;
A_esq(2:m+1,1:n) = A;
A_ext = rv*(A_dalt + A_baix) + rh*(A_dreta + A_esq)+r0n*A_ext;
%
A = A_ext(2:m+1,2:n+1);

end;

A=1-A; %Desfem el canvi

A=min(A,im2double(A_ini)); % aquest min fa que els pixels que
eren negres continuin sent negres

Afinal=round(A);

```

```
function [characters]=lectora(text)
origen=fopen(text);
characters=fscanf(origen,'%s');
fclose(origen);
% que matlab llegeixi ttest caracter a caracter, el busqui al
cataleg i
% posi el seu numero de cataleg com a component del vector in.

%Entrada es el text original.
%Sortida es els caracters cargats en un vector de caracters.
```

```

function [vector]=letrasdobles(V,porcenG,plotG)

%Funcio que detecta zones conflictives. Aquesta funcio tracta
de detectar
%lletres que no se han pogut separar mitjançant la deteccio de
zones. Son
%lletres que no sortiran al alfabet.
%La variable de entrada: V son les diferents zones.

% porcenG Son las zones amb mes del x% percent amb mes
amplitud que la
% resta. Aquest valor pot variar bastant segons la tipologia
de lletra.
% Valors entre [1.5-1.9] aprox.

%La variable de sortida es el vector on indica les posicions
que
%posiblement erronees.
%Si plotG es igual a "si" mostrara per pantalla el plot

[c,d]=size(V);

for i=1:d
    [ffd,ffh]=size(V{i});
    largada(i)=ffh; %Vector de totes les amplades
end
larS=sum(largada); %Suma de totes les amplades
larM=larS/d;%media de les amplades per les zones.
largM=largada./larM; %Cada zona divideixo per la mitja.

dfad=1; %Inicialitzacio

for i=1:d
    fd=largM(i)>porcenG; %Es queda nomes amb les lletres mes
grans que el porcentatge que hem decidit (porcenG)
    if fd==1
vector(dfad)=i;
dfad=dfad+1;
    end
end

if plotG=='si'
ge=ones(d)*(porcenG);
plot(largM,'y') % Això ens servira per fernes una idea de quin
valor es el adequat. Mostra per pantalla
%La mitja de llargades. Com observarem el valor estara entorn
a 1. Haurem
%de veure els valors que es grans. Que corresponen a les
lletres amb mes
%amplitud.
hold on
plot (ge,'b');
h = legend('Letras','porcenG',2);
set(h,'Interpreter','none')
title('Promedio de largada de letras');
xlabel('Letras');

```

```
ylabel('Largada promedio');  
text(50,2.5,'La línea azul solo incluir letras grandes')  
hold off  
else  
end  
figure
```



```

function LimpiarDobles(V,dobles,caracters,lm)
% La funcio LimpiarDobles elimina totes les lletres dobles fen
un retall
% els retall sempre es fan de esquerra a detra. Si tenim mes
de dos lletres
% agafarem la primera lletra per l'esquerra i despres la
segona per la
% detra.

%Variable de entrada. Es el V es la matriu celular de lletres.
dobles es el
%vector de lletres conflictives. caracters son les lletres
reals i lm es el
%nom del fitxer on guardarem les lletres.

%Guardem les noves imatges una vegada retocades si la llargada
coincideix
%amb la llargada bona.

%Proceso semiautomatico
dec=0;
for i=dobles
    i=i+dec; %Saltar a letra aÑadida
    [a,b]=size(V);
    tres=V{i}; %Variable auxiliar per treballar si tenim 3
lletres. Guardem la lletra original
H=imcrop(V{i}); %Variable auxiliar (Primera lletra)
t= isempty(H);
    if t==1
        else
aux1=imcrop(V{i}); % Es la nova lletra que afegirem. Segona
lletra
V{i}=H; %Reescribim la primera imatge a la posicio on era la
imatge original. Recordem que hem de tallar de esquerra a
dreta.
Vn=Nuevaletra(V,aux1,i+1); %Nova lletra que afegirem
V=Vn;%Copiar
t=isempty(aux1); %Recordem que si hi ha tres lletres nomes
guardarem la primera.
dec=dec+1; %Contador de letras aÑadidas
if t==1 %Si tenemos 3 letras la letra guardaremos la primera y
la segunda no farem res.
Ha=imcrop(tres); %Variable auxiliar (segona lletra). Si tenim
3 lletres.
aux2=imcrop(tres); % La tercera lletra
Vn=Nuevaletra(V,Ha,i+1); %Nova lletra que afegirem
V=Vn;%Copiar
Vn=Nuevaletra(V,aux2,i+2); %Nova lletra que afegirem
V=Vn;%Copiar
dec=dec+1; %Contador de letras aÑadidas
else
end
end
end
    dep=length(caracters);
    [del,dee]=size(V);

```

```
save(lm, 'V')
if dep==dee
    save(lm, 'V') %Guardem si tot esta bé
else
    Vn=LimpiaRest(V, characters); %Es una funció que s'etjectu
    si la lletra no coincideix amb la solucio
    V=Vn;
    [del, dee]=size(V);
    if dep==dee
        save(lm, 'V') %Guardem si tot esta bé
    else
        ERROR
    end
end
end
```

```

function Vn=LlimpiarResta(V,caracters)

%La funcio LlimpiarResta se t'executara si no acabem de treure
totes les
%lletres.
%El vector errores es el vector de posibles lletres que
estinguin malalmen.
% Exemple en el quijote 2 falta per detectar 4 lletres i les
dobles lletres
% pegades que no trobem es la 'il'.
%%% Ejemplo quijote02
a=caracters;
b=strfind(a,'il'); %Aquest son els posibles errors.
mil=length(b);
dep=length(caracters);
[del,dee]=size(V);
rang=dep-dee+1; %Son les lletres que falten. Posem 1 per si a
cas.
for i=1:mil
g=(b(i)-rang):b(i);
error{i}=g;
end
errores=cell2mat(error);
dec=0;
for i=errores
i=i+dec; %Saltar a letra añadida
[a,b]=size(V);
tres=V{i}; %Variable auxiliar per treballar si tenim 3
lletres. Guardem la lletra original
H=imcrop(V{i}); %Variable auxiliar (Primera lletra)
t= isempty(H);
if t==1
else
aux1=imcrop(V{i}); % Es la nova lletra que afegirem. Segona
lletra
V{i}=H; %Reescribim la primera imatge a la posicio on era la
imatge original. Recordem que hem de tallar de esquerra a
dreta.
Vn=Nueva letra(V,aux1,i+1); %Nova lletra que afegirem
V=Vn;%Copiar
t=isempty(aux1); %Recordem que si hi ha tres lletres nomes
guardarem la primera.
dec=dec+1; %Contador de letras añadidas
end
end
end

```

```

%Funcion para separar las lineas.

%Aquesta funcio agafa una imatge que te com a variable de
entrada una imatge amb
%texte i divideix el texte en files. Ordenanles. Retorna una
matriu celular
%que cada index correspon a la fila ordenada de dalt a baix.
(Amb l'ordre
%de lectura). Els pixels del fons han de ser 0.

function [fila]=lineas(A)

%A es la matriu a la que volem ferli la separació de linies i
filas es la
%matriu celular

%Ens retorna les linies divides.

%TRACTAMENT 1. TALLAR LINES.

ter=sum(A');
terl=sum(ter)/length(ter); %Mitja de pixels.
min_pixel=terl; % Es evident que no cal arriba a la mitja de
pixels...

%En principi si les lines fossin rectes podriam passar ==0. ja
que hi
%hauria sempre una linea que sumaria 0 i separaria el gran
problema es que
%no hi ha zones que no sumen 0 pero han de separar lines per
això hem de
%seguir un proces iteratiu.

%Talla linies. Com bé sabem hem de talla linies abans de fer
%el analisis.

for k=0:4:min_pixel
    %hacer pasos de cuatro en cuatro y que me detecte las
zonas donde haya
    %0
    %pixels.

    c=find(sum(A')<=k); %Si les linies no estiguessin torçades
podriam posar 0 y separaria perfectament.
    %Agafem un valor horientatiu per exemple 30. Podriem agafar
un valor fin a
    %100 o 200. O més.

    f=length(c);
    [a,b]=size(A);

    if f==0
        %PASAR
    else

```

```

for i=1:f
    if i==f %Para LA ULTIMA FILA
        h{i}=im2double(A(c(i):a,1:b));

    else
h{i}=im2double(A(c(i):c(i+1),1:b)); %GENERAR FILAS (HACE
CORTES )
        end
    end

%Tractament 2. Eliminar filas VACIAS

%Eliminamos las filas vacias. Que tendran altitud de dos.
m=1;
suma=0;

    for i=1:f
[ab,cd]= size(h{i});
if ab>2 %ELIMINAR FILAS VACIAS
    te{m}=h{i};
    CCa=bwconncomp(te{m}); %Troba les regions connexes, sino
especificues res son els "8" veins. bwconncomp(A,4) 4 veins
    SAa = regionprops(CCa, 'Area'); %Busca les areas de les
regions conexas
    CAa = struct2cell(SAa); %Transforma en "matriu"
    to_pixela=cell2mat(CAa); %Vector de pixel

    mePia=sum(to_pixela)/length(to_pixela); %Mitjana entre el
total de pixel i les zones.
    pixel(m)=sum(to_pixela); %La suma de tots els pixels
    suma=suma+mePia; %SUMA tots el pixels divits per les
zones
        m=m+1; %NUMERO DE FILAS
    else
    end
    end

me=suma/(m-1); %MEDIA de zones

%Tractament 3. Eliminar las filas con pocos pixeles. al
realizar habra
%filas con uno o dos pixels de los cortes (filas torcidas).
%Esta funcion no funciona bien haremos la media de letras.

dd=1;
dar=0;
for i=1:(m-1)
    gd=pixel(i);
if gd*0.7>me % Si tots els pixels es mes gran que la mitja
multiplicada per 0.8 de una sola lletra fer.

        fila{dd}=te{i};
[moc,moc1]=size(fila{dd});
mg(dd)=moc; %Numero de filas
dar=dar+moc; %sumatorio de filas
dd=dd+1;

```

```
else
end

end
%Eliminar zonas de doble grosor.

buen=dar/(dd-1); %Todas las filas por las filas
mel=mg./buen; %MEDIA

abab=mel.*0.7>1; % Si multiplicamos la media por
baba=sum(abab);

if baba==0
    break %Cuando eliminemos las filas con doble grosor parar
so
%    %iterativo fijar un variable
else
end
    end
end
```

```

%Funcio escriptora
%Rep el nom de la matriu double al 0=negre 1=blanc i la guarda
com a fitxer
%N imatge del format demanat.
%ERROR NO EM FUNCIONA SI FICO LA VARIABLE MUDA H!!!
function matbin2fitxer(A,fitxer)

%Variable h de retorn de imwrite (val 1 si tot s'ha creat be
val 0 en cas
%contrari.)

imwrite(A,fitxer);

% Es guarda la imatge retocada en el directori de treball.
%nova_imatge = [imatge ' Nb=' num2str(Nb) ' ; Nn=' num2str(Nn)
' ; tnegra='
%num2str(tnegra) ' ; tblanca=' num2str(tblanca) ' ; r0b='
num2str(r0b) ' ; r0n=' num2str(r0n) ' .' format];
%imwrite(A,nova_imatge,format);

```

```

%Mejoras!!

%Introduim millores:
% Analitzem tots els valors propis
%Millorem cada lletra amb una aleatorietat guardem el resultat
load('nuevas.mat');
% [de,df]=sort(nueva,'descend');
t=79;
tblanca=aux13(t)-0.11;
r0b= aux14(t);
Nb=aux15(t);
tnegra=1;
r0n=aux16(t);
Nn=aux17(t);
load('Estadistica_quijote300','alfa','V','d','Cat','solucio');
[a,b]=size(V);
for i=1:b
A=V{i};
B=fluxdetintablanca(A,Nb,r0b,tblanca);
V{i}=fluxdetintanegra(B,Nn,r0n,tnegra);
end

clear aux13
clear aux14
clear aux15
clear aux16
clear aux17

[na,aux1]=size(V);
for i=1:aux1
    [aux2,aux3]=size(V{i});
    aux4(i)=aux2;
    aux5(i)=aux3;
end
aux6=sort(aux4,'descend');
aux7=sort(aux5,'descend');
aux8=aux6(1);
aux9=aux7(1);
aux10=[aux8,aux9];
aux11=sort(aux10);
lim=aux11(2); % Es el limit el numero maxim de matrius que
podem treballar.

%Quocients de valors singulars del Cataleg
[aux12,aux13]=size(Cat);

for i=1:aux13
vs=svd (Cat{i}); %Valores singulares
aux14=length(vs); % Llargada de la matriu.
aux15=lim-aux14; % Diferencia entre la matriu y la llargada
limit
aux16=zeros(aux15,1);
vs=[vs;aux16];
au=vs(2:lim)/vs(1);
C{i}=au;%Quocients de valors singulars .
%De les solucions
end

```



```

C=cell2mat(C); %Convertim la matriu celular en matriu abans de
guardar

%Quocients de valors singulars de les lletres.
[aux17,aux18]=size(V);

for i=1:aux1
    vs=svd (V{i}); %Valores singulares
    aux14=length(vs); % Llargada de la matriu.
    aux15=lim-aux14; % Diferencia entre la matriu y la llargada
limit
    aux16=zeros(aux15,1);
    vs=[vs;aux16];
    aul=vs(2:lim)/vs(1);
    qs{i}=aul; %Quocients de valors singulars .Podem fer 19
despres jugarem amb aquest valor
%De les solucions
end

%Aplicarem fluxs de tinta al cataleg;

save('Estadistica_quijote300
_mejorado.mat','alfa','V','d','Cat','solucio','C','qs','lim');

```

```

function [filas]=neteja(fila,porcenP)

% Funcio que elimina els pixels de dimensions mes petites.fila
es un matriu
% celular on en el nostre cas introduirem cada fila i li ferem
la seva
% neteja. La variable de sortida es la imatge netejada (sense
comes punts
% etc...). No treu parentesis ni corxetes.

%Valors acceptables.

%La variable porcenP serveix per eliminar pixels petits. Es una
variable
%percentual. Por exemple el 0.5 elimina els pixels que tenguin
el 50% de la
%mitja de els pixels de les zones.

%Si possem 0.4 elimina els pixels mes petits que el 60%. de la
mitja.
%Es a dir es porcen=1-%Percentatge que volem eliminar.

%Esta variable al ser percentual no depende de la resolucio
de la imatge.

%Farem un sistema estadistic per que serveixi automatizat.

[dc,dd]=size(fila);
for i=1:dd

CC=bwconncomp(fila{i}); %Troba les regions connexes, sino
especificues res son els "8" veïns. bwconncomp(A,4) 4 veïns
SA = regionprops(CC, 'Area'); %Busca les areas de les regions
connexes
CA = struct2cell(SA); %Transforma en "matriu"
to_pixel=cell2mat(CA); %Vector de pixel
mePi=sum(to_pixel)/length(to_pixel); %Mitjana entre el total
de pixel i les zones.

borrar=floor(mePi*porcenP);
% Eliminem les zones amb menys del 60 percent de la mitjana.
% %La dada aquesta es horientativa. Pero pot ser bastant real.
Ha de ser un
% valor enter.

filas{i}= bwareaopen(fila{i},borrar); %Funció per eliminar
zones amb menys de N pixels. primera variable es la matriu a
eliminar la segona els pixels mes petits que.

%imshow(filas{i}),figure %Si volem confirmar que esta be

end

```

```

function [Ve]=Nuevaletra(V,Vnova,n)
[a,b]=size(V);
%V es la matriu inicial
%Vnova es la matriu on anira enganxat
%n es la posicio.
if n==1 % Per si la letra coincide con el primer valor del
texte
    lk=V(n:end); % matriu de vectors per veure la imatge
    Ve=[Vnova,lk];
elseif n==(b+1) % Per si la lletra coincideix amb l'ultim
valor del texte
    mn=V(1:end);
    Ve=[mn,Vnova];
else
    mt=V(n:end);
    rs=V(1:(n-1));
    Ve=[rs,Vnova,mt];
end

```

```

function OCR(fitxer,format,porcenP,porcenG,plotG,Dlletres)

% Ens fa el OCR on les variables son:
%
% fitxer: la imatge que volem treballar.
% format:format en que treballarem
% porcenP es el porcenP de imatges petites que volem eliminar
porceP
%[0.3-0.6]
%porcenG que ens mostri per pantalla els elements mes grans
per detecta si
%hi ha zones conflictives. Aquest procediment es l'unic
procediment que no
%es pot fer automatic al menys cada tipologia de lletra tindra
un valor
%diferent de porcenG [1.6-1.8]
%Plot G si valen 'si' ens mostra els plots sino no fa res.
% El plotG s'ha de posar si la primera vegada per tenir les
mostres una
% vegada estem d'acord amb les mostres posarem no i se't
ejectura
% directament l'ultim pas. Això fa que el programa sigui molt
més rapid.

%Dlletres ens indica si hem realitzat la purga de dobles
lletres o no. Si
%posem 'si' vol que encara no l'hem fet.

%Una vegada seleccionat les variables de entrada ens mostra
per la pantalla
%la imatge llavors hem de click al ratolí la zona de inici i
arrossegar
%al ratolí on es crear una rectangle on situarem la imatge
que volem
%seleccionar. Una vegada seleccionat fem doble click amb el
ratolí cera la
%zona on ferem el OCR.

if plotG=='si'
A= fitxer2matbin(fitxer,format);

%A=imcrop(A_ini); % Seleccionem la zona de treball. Amb el
ratolí.
%Zona de tall si ens interessa
%Aquesta funcio dins del matlab ens deixa escullir la zona de
tall que
%volem per fer servir-la hem de click al ratolí la zona de
inici i arrossegar
%al ratolí on es crear una rectangle on situarem la imatge
que volem
%seleccionar. Una vegada seleccionat fem doble click amb el
ratolí cera la
%zona on ferem el OCR.
%Desde el punto de vista de un OCR pareix interesant tenir la
funcio
%aquesta pero matlab realitza la visualitzacio de les imatges

```

```

molt
%precariament per tant ferem el tall primer amb un programa de
imatges i
%despres pasarem la imatge.

Nn=0; r0n=0.2; tnegra=1;
A=fluxdetintanegre(A,Nn,r0n,tnegra);
Nb=0; tblanca=0.95; r0b=0.95;
A=fluxdetintablanca(A,Nb,r0b,tblanca);
A=1-A; % S'ha de fer canvi de pixels blancs a negres, per
veure les regions.
fila=lineas(A); %La funcion que nos separa las regione en
lineas.

filas=neteja(fila,porcenP); % Funcio que treu les comes, punts
i accents.. No treu els parentisis ni els corxetes

    [a,b]=size(filas);
    comprobar (filas,(1:b),15,1) %Mostrar filas netejades per
pantalla
V=ReconocimientoZonas(filas); % Funcio deteccio de zones.
detecta cada zona.
    [c,d]=size(V);
    comprobar (V,1:d,10,10) % Mostra totes les imatges per
pantalla. Totes les lletres.
dobles=letrasdobles(V,porcenG,plotG);
save('doblar.mat','V','dobles');
% Nos detecta las zonas problematicas. Enseña por pantalla un
plot con las
% diferentes zonas que vamos a mostrar. El plot es la imagen
porcentual del
% tamaño de las letras
else
end
load('doblar.mat');
%comprobar(V,dobles,10,10) %Mostra lletres amb mes amplitud
imar='.txt';
text=[fitxer,imar]; %El nom del fitxer coincidirà amb la
imatge llegida.
caracters=lectora(text); %Son totes les imatges reals.

imat='.mat';
lm=[fitxer,imat]; %El nom del fitxer coincidirà amb la imatge
llegida.

if Dlletres=='si'
LimpiarDobles(V,dobles,caracters,lm); %Funcio que neteja zones
problematicues
else
end

```

```

function [AB]=parrafos(A_ini,div)

%La funcion parrafo dividi la matriz en x trozos. La matriz
debe conter una imagen
% con texto

%La variable div indica el numero de trozos que queremos
dividir la imagen.
%La variable A es la matriz a la cual aplicaremos la funcion.

%AB es la matriz celular de salida una vez aplicado la
funcion. Donde
%estaran ordenadas de arriba a bajo tal como el orden de
lectura.

A=1-A_ini;
ter=sum(A');
ter1=sum(ter)/length(ter); %Mitja de pixels.
min_pixel=ter1;
largada=length(A);
division=largada/div;
for k=0:3:min_pixel
    %hacer pasos de cuatro en cuatro y que me detecte las
zonas donde haya
    %0
    %pixels.

    c=find(sum(A')<=k); %Si les linies no estiguessin torçades
podriam posar 0 y separaria perfectament.
    %Agafem un valor horientatiu per exemple 30. Podriem agafar
un valor fin a
    %100 o 200. O més.

    f=length(c);

    if f==0 %Sino detecta ninguna pasar

    else
        for ka=1:(f-1)

            if c(ka)+1==c(ka+1);
                c(ka+1)=0; %Eliminar zonas consecutivas
            else
                end

            end

        end
        le=c(:,(c')>0); %Quedarnos solo con las no consecutivas

        ge=length(le);

        gf=le(1)+division<=le((1):end); %Buscar valors que vagin de
el punt 1 al 1+division.
        re=sum(gf);
        if re>=1
            [ax,bx]=max(gf); %bx es el primer valor que dona 1 bxi

```

```

        puntos(1)=le(bx);
    else
    end
    %bxf=0; %Inicializacion
    for fde=1:(div-2) %Necesito la division menos 2.
    gf=le(bx)+division<=le((1:end));
    re=sum(gf);
    if re>1
        [ax,bx]=max(gf);%bx es el primer valor que dona 1
        puntos(fde+1)=le(bx);
        fde=fde+1;
    else
    end
    end
    if fde==(div-1)
        break
    else
    end
    end
    [a,b]=size(A_ini);
    puntos=[1,puntos,a]; %Son los puntos que generar el parrafo

    for i=1:fde+1
        AB{i}=im2double(A(puntos(i):puntos(i+1),1:b)); %GENERAR
    parrafos
    end
    % comprobar(h,(1:fde+2),1,1); Comprobado

```

```

%Aqui es realitzar els diferents proves

%Definim variables Quijote 1

porcenP=0.4; %% Porcentatge de eliminats [0.3-0.6]
porcenG=1.2; %% Que ens mostri per pantalla les regions
grans. possibles
%errors. [1.1-1.3]
%Con uno 1.2 en este caso detecta todas las zonaspegadas
plotG='si'; %Perque ens mostri el plot de les possibles lletres
"dobles". hem de posar 'si'
porcenG=1.2;
Dlletres='no'; %Si posem 'si' vol dir que encara no hem
depurat les dobles lletres.
%Quijote 2 al 5.
format='tif';

fitxer='quijote_01';
OCR(fitxer,format,porcenP,porcenG,plotG,Dlletres); %Correcto

% %Mostra 3. Platon
%
%
% porcenP=0.4; %% Porcentatge de eliminats [0.3-0.6]
% plotG='no'; %Perque ens mostri el plot de les possibles
lletres "dobles". hem de posar 'si'
% % Nn=6; r0n=0.2; tnegra=1;Nb=0;tblanca=0.95;r0b=0.95;
% porcenG=1.3;
% Dlletres='si'; %Si posem 'si' vol dir que encara no hem
depurat les dobles lletres.
% %Quijote 2 al 5.
% format='tif';
% % fitxer='platon_1(300pixels)';
% %fitxer='platon_2(300pixels)';
% %fitxer='platon_3(300pixels)';
% %fitxer='platon_4(300pixels)';
% fitxer='platon_5(300pixels)';
%
% OCR(fitxer,format,porcenP,porcenG,plotG,Dlletres);

```



```

function [V]=ReconocimientoZonas(filas)

%Aquesta funcio llegeix cada fila (guardades en una matriu
celular) i reconeix cada lletra i finalment les
%guarda en un matriu celular. Ordenades de esquerra a dreta i
de dalt abaix
%com l'ordre de lectura.

[a,b]=size(filas);
ffa=1;
for i=1:b
L2=bwlabel(filas{i});
S = regionprops(L2,'Area');% LA FUNCIO bwlabel(A); % fa el
MATEIX bwconncomp y labelmatrix amb una funcio, dona un
resultat
%
% % amb class "double". Doble precisio.
% %En resum: Troba les regions connexes, sino especificues res
son els "8"
% %veins. Crea la matriu de les estructures, etiquetan cada
zona. A cada zona
% %li correspon un valor. Y la lectura es de esquerra a dreta,
per
% columnes.
ga= struct2cell(S);
m=length(ga);

for k=1:m %En contrar las regiones
[r, c]= find(L2==k);%Troba les coordenades de cada regio.
t= bwselect(L2,c,r); %Selecciona els punts de las
coordenadas.
%Crea una matriu amb els punts de les coordenades
lleguides a la funcio anterior. Es a decir cada zona en una
matriu.
t = t(sum(t')>0,sum(t)>0); %Eliminar filas y columnas
de ceros. Això fa que només quedi la regio bona.
%Sino es fa això queda la matriu amb el tamany de la
matriu original.
t=im2double(t); %convertir la matriz logica en double
(SINO AL APLICAR LA SVD DONA ERROR, NO DONA UN RESULTAT
%JA QUE LA SVD NO DEIXA APLICARLA A MATRIUS LOGIQUES.
V{ffa}=t; %Guardem cada regio en una estructura celular.
% [ffd,ffh]=size(V{ffa});
% largada(ffa)=ffh;
ffa=ffa+1;
end
end

```