

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials

**RECONEIXEMEN DE TEXT MANUSCRIT
MITJANÇANT LA SVD**

ANNEX

Autor: José de la Puente Puigjaner
Director: Jaume Amorós Torrent
Convocatòria: Gener 2017



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Sumari

SUMARI	2
A. CODI	3
A.1 FUNCIONS PRINCIPALS.....	3
A.2 FUNCIONS SECUNDÀRIES	7
A.3 CONTROL DE LES FUNCIONS.....	10
A.4 CREACIÓ DE CATALEGS	11
A.5 FUNCIONS DE FLUX DE TINTA	12
A.6 FUNCIONS AUXILIARS	13
B. CATÀLEGS.....	16
C. TEXTOS.....	20

A. Codi

En aquest apartat es presenta tot el codi desenvolupat per al reconeixement de text, així com altres funcions utilitzades per a la investigació.

A.1 Funcions principals

segmentacio.m

```
function [lletres,linies,ll_linia]=segmentacio(imatge,format)
% lectura de tiff b/n de texte, identificacio lletres,
% emmagatzemat en estructura d'imatges
% Versio per text recte, amb linees i paraules separades per espais en
% blanc. pero amb taques, fent eliminacio de taques en la llista de linees,
% paraules, lletres basada en eliminar aquelles que siguin molt mes petites
% que el tamany promig
% INPUT:fitxer=nom complet del fitxer de la imatge, que ha de ser en blanc
% i negre, format=format del fitxer imatge
% OUTPUT:lletres=matriu de 4 columnes, cada fila es
% fila inicial,fil final, col. inicial, col. final d'una lletra de la paraula
% (indexs en la imatge),linies=matriu amb 2 columnes que
% te una fila per cada linia de text de la imatge, la fila es fila inicial,
% fila final d'aquesta fila de texte en la imatge, ll_linia=lletres en cada
% linia (matriu de 2 cols, cada fila es lletra inicial, lletra final de la
% linia)

% Jaume Amoros, Jose de la Puente, UPC, Barcelona
% 2017/12/01

% provat i funciona

% llegeix la imatge
A = imread(imatge,format);
A = im2double(A(:,:,1));
% fa 1=negre, 0=blanc
A=1-A;

% deteccio de linies (s'assumeix que el text esta escanejat recte
% i hi ha espai en blanc entre linees)
% pixels negres per cada fila
pixlin=sum(A,2).';
% intervals de linees de pixels en blanc i no en blanc
[~,linies]=intervals_zero(pixlin);
% eliminacio de linies creades per una taca
mitja_linia=mean(linies(:,2)-linies(:,1)); %amplada mitjana de les linies
linies(linies(:,2)-linies(:,1)<mitja_linia*0.6,:)=[]; %elimina les linies amb
amplada menor a la mitja ponderada

lletres=[];
% les llistes de lletres per paraula i per linia necessiten inici dummy
```

```

% provisional
ll_linia=zeros(size(linies,1)+1,2);
% per cada linea: deteccio de paraules (text escanejat recte implica
% separacio per espais en blanc amples)
for k=1:size(linies,1)
    altlinia=linies(k,2)-linies(k,1)+1;
    Afila=A(linies(k,1):linies(k,2),:);
    pixcol=sum(Afila,1);
    % espais en blanc i lletres
    [~,lletres_linia]=intervals_zero(pixcol);

    % guarda posicio de les lletres (imatge, linia, lletra)
    fila_lletres=zeros(size(lletres_linia,1),4);
    for l=1:size(lletres_linia,1)
        % ajusta finestra a la lletra (les columnes son ajustades, les
        % files no)

        Alletra=A(linies(k,1):linies(k,2),lletres_linia(l,1):lletres_linia(l,2));
        pixlinll=sum(Alletra,2).';
        indnz=find(pixlinll>0);
        poslletra=[linies(k,1)+indnz(1)-1,linies(k,2)+indnz(end)-
altlinia,lletres_linia(l,1),lletres_linia(l,2)];
        fila_lletres(l,:)=poslletra;
    end;

    % eliminacio de lletres creades per una taca
    mitja_lletra=mean(fila_lletres(:,4)-fila_lletres(:,3)); %amplada mitjana
de les lletres
    fila_lletres(fila_lletres(:,4)-fila_lletres(:,3)<mitja_lletra*0.2,:)=[];
%elimina les lletres amb amplada menor a la mitja ponderada

    lletres=[lletres;fila_lletres];

    ll_linia(k+1,1)=ll_linia(k,2)+1;
    ll_linia(k+1,2)=size(lletres,1);

end;

% traiem l'inici dummy provisional de les llistes de paraules i linees
ll_linia(1,:)=[];

save ('imatgebinaria.mat', 'A')

```

signatures.m

```

function [signatura]=signatures(lletres,dimensio,base)
% Diagonalitza totes les lletres del text i guarda les seves components
%INPUT: -lletres: matriu amb les coordenades de les lletres
%- dimensio: nombre de valors singulars a guardar
% - base: nombre de vectors singulars a guardar

```



```

%OUTPUT: -signatura: estructura amb els quocients dels valors 2 primers
%valors singulars (sense el primer) a la primera cel·la, i els 3 primers
%vectors singulars d'arribada i de sortida a la 2a i 3a cel·la. Un afila
%per cada lletra del text

%Jose de la Puente, UPC, Barcelona
%2016/12/06

%Provada i funciona

load ('imatgebinaria.mat','A') %matriu de tot el text
n_lletres=size(lletres,1);

S=cell(n_lletres,3);

for i=1:n_lletres
    lletra=A(lletres(i,1):lletres(i,2),lletres(i,3):lletres(i,4));
%localitzacio de la lletra en la matriu
    [U,D,V]= svd(lletra);
    D=diag(D);
    S{i,1}=D(2:dimensio+1)/D(1); %Quocients dels valors singulars sense el
primer (que sempre es 1)
    S{i,2}=V(:,1:base); %Vectors singulars d'arribada
    S{i,3}=U(:,1:base); %Vectors singulars de sortida
end

signatura=S;

end

```

OCR.m

```

function
[deteccio_numero,deteccio_lletra]=OCR(signatura,nom_cataleg,dimensio,base,taxa
_d,taxa_u,taxa_v)
% comparacio de les lletres del text amb els patrons
%OUTPUT: - deteccio_numero: vector amb els index de les paraules del
%cataleg que ha detectat
% - deteccio_lletra: cadena d'strings amb el reconeixement del text

%Jose de la Puente, Jaume Amoros, UPC, Barcelona
%2016/12/14

%Provat i funciona

load(nom_cataleg);

deteccio_numero=zeros(1,size(signatura,1));
display(size(signatura,1));

for i=1:size(signatura,1)

```

```

distancies_globals=zeros(1,size(cataleg,1));

Vs=signatura{i,2}; %vectors d'arribada de la lletra
Us=signatura{i,3}; %vectors de sortida de la lletra

for j=1:size(cataleg,1)

    Vc=cataleg{j,2}; %vectors d'arribada de la lletra del cataleg
    Uc=cataleg{j,3}; %vectors de sortida de la lletra del cataleg

    distancies_D=(signatura{i,1}-cataleg{j,1}(1:dimensio))'; %distancies
dels valors singulars

    distancies_D=taxa_d.*distancies_D; %distancia valors singulars
ponderada
    distancia_V=zeros(1,base);
    distancia_U=zeros(1,base);

    for k=1:base
        %v arribada lletra
        vs=Vs(:,k)*ones(1,size(Vc,1));
        vs=vs';
        vs=vs(:);
        vs=vs*sqrt(size(Vc,1)); % normalitzacio corregida

        %v arribada cataleg
        vc=Vc(:,k)*ones(1,size(Vs,1));
        vc=vc';
        vc=vc(:);
        vc=vc*sqrt(size(Vs,1)); % normalitzacio corregida

        %distancia vectors arribada
        dist=(1/sqrt(size(Vc,1)*size(Vs,1)))*min(norm(vs-vc),norm(vs+vc));
        distancia_V(k)=dist;

        %u sortida lletra
        us=Us(:,k)*ones(1,size(Uc,1));
        us=us';
        us=us(:);
        us=us*sqrt(size(Uc,1)); % normalitzacio corregida

        %u sortida cataleg
        uc=Uc(:,k)*ones(1,size(Us,1));
        uc=uc';
        uc=uc(:);
        uc=uc*sqrt(size(Us,1)); % normalitzacio corregida

        %distancia vectors sortida
        dist=(1/sqrt(size(Uc,1)*size(Us,1)))*min(norm(us-uc),norm(us+uc));
        distancia_U(k)=dist;
    end
    %calcul distancia global amb la lletra j del cataleg

```

```

distancia_global=norm(distancias_D)+sqrt((taxa_v.^2)*(distancia_V.^2).'+sqrt(
(taxa_u.^2)*(distancia_U.^2).');
    distancias_globals(j)=distancia_global;
end
    [~,index]=min(distancias_globals); %busca la distancia global minima
d'aquesta lletra i
    deteccio_numero(i)=index;
    display([num2str(i) '/' num2str(size(signatura,1))]); %per veure en
pantalla l'estat del proces de reconeixement
end

deteccio_lletra=alfa(deteccio_numero);
end

```

A.2 Funcions secundàries

interval_zero.m

```

function [nuls,nonuls]=interval_zero(v)
% subdivisio d'un vector en subinterval segons si el valor de les
% components arriba o no a la cota preestablerta (0)
%
% Jaume Amoros, UPC, Barcelona
% 2016/4/5
%
% Funciona amb un petit error:
% No detecta components nulles o no nulles aïllades, nomes
% subinterval de longitud almenys 2
% Si els espais son suficientment grans no afecta

% intervals de zeros
w=[false v<=0 false];
iniciis=find(w(2:end) & ~w(1:end-1));
finals=find(~w(2:end) & w(1:end-1))-1;
nuls=[iniciis.' finals.'];

% intervals de positius estrictament
w=[false v>0 false];
iniciis=find(w(2:end) & ~w(1:end-1));
finals=find(~w(2:end) & w(1:end-1))-1;
nonuls=[iniciis.' finals.'];

```

lectura.m

```

function [reallletres, realparaules, reallines]=lectura(fitxer)

```

```

% funcio que rep un fitxer de texte i prepara la seva llista de lletres,
% paraules, linees en format compatible amb el de numeritzacio.
% INPUT: fitxer=nom complet del fitxer de texte
% OUTPUT: reallletres=cadena de texte amb totes les lletres que hi havia a
% fitxer, en l'ordre en que hi apareixien (descartant espais en blanc i
caracters de final de linea),
% realparaules= matriu de 2 columnes que llista les paraules (=trossos de
texte
% delimitat per espais en blanc), cada fila indica posicio de les lletres
% inicial i final de la paraula en lletres, reallinees= matriu de 2 columnes
% que llista les linees del texte (descartant linees en blanc), cada fila
% indica posicio de les lletres inicial i final de la linea en lletres

% Jaume Amoros, UPC, Barcelona
% 2016/4/8

% provada i funciona

entrada=fopen(fitxer,'r','n','UTF-8');

reallletres=[];
realparaules=[0 0];
reallinees=[0 0];
linea=fgetl(entrada);

while ischar(linea)
    % descomposa linea en paraules
    linea=strtrim(linea);
    % Opcional aqui: eliminar de linea signes de puntuacio si no participen
    % en el OCR
    C=strsplit(linea,{' ',' '});

    longlinea=0;
    for k=1:length(C),
        if length(C{k})>0 % problema amb les linees en blanc
            reallletres=[reallletres C{k}];
            realparaules=[realparaules;
realparaules(end,2)+1,realparaules(end,2)+length(C{k})];
            longlinea=1;
        end;
    end;
    if longlinea==1
        reallinees=[reallinees; reallinees(end,2)+1,realparaules(end,2)];
    end;

    linea=fgetl(entrada);
end;

fclose(entrada);

realparaules(1,:)=[];
reallinees(1,:)=[];

```


resultats.m

```
function resultats(directori,nom,nom_cataleg,deteccio_lletra,reallletres)
% calcula els resultats i els guarda a un fitxer anomenat resultats_ +
% 'nom'

%Jose de la Puente, UPC, Barcelona
%2016/12/20

fitxer_resultats=[directori 'resultat_' nom '.txt']; %fitxer on es guarda

encerts=0;
for i=1:size(deteccio_lletra,2) %calcular els encerts
    if i>size(reallletres,2)
        break
    end
    if deteccio_lletra(i)==reallletres(i)
        encerts=encerts+1;
    end
end

llargada=num2str(size(reallletres,2));
t_encerts=num2str((encerts/size(reallletres,2))*100);
encerts=num2str(encerts);

fitxer=fopen(fitxer_resultats,'wt');
fprintf(fitxer,['Cataleg: ' nom_cataleg]);
fprintf(fitxer, '\n');
fprintf(fitxer,['Llargada: ' llargada]);
fprintf(fitxer, '\n');
fprintf(fitxer,['Encerts: ' encerts]);
fprintf(fitxer, '\n');
fprintf(fitxer,['Taxa encerts: ' t_encerts]);
if size(deteccio_lletra,2)~= size(reallletres,2) %en cas que hi hagi un error
de llargada
    fprintf(fitxer, '\n');
    fprintf(fitxer,'ERROR LLARGADA');
end
fprintf(fitxer, '\n');
fprintf(fitxer, '\n');
fprintf(fitxer, 'Text detectat:\n');
fprintf(fitxer,deteccio_lletra);
fprintf(fitxer, '\n');
fprintf(fitxer, '\n');
fprintf(fitxer, 'Text real:\n');
fprintf(fitxer,reallletres);
fclose(fitxer);

%per mostrar en pantalla els resultats
display(['Llargada: ' llargada]);
display(['Encerts: ' encerts]);
display(['Taxa d'encerts: ' t_encerts '%']);
```

```

if size(deteccio_lletra,2)~= size(reallletres,2) %en el cas que hi hagi un
error de llargada
    display('ERROR LLARGADA');
end

```

A.3 Control de les funcions

correr_fitxers.m

```

% FITXER PRINCIPAL

%Jose de la Puente, UPC, Barcelona
% 2017/01/13

clear all;
%Fitxer que cal correr per a realitzar el reconeixement

%
%-----
%%%%%%%%%% NOM PRINCIPAL I DIMENSIO (MODIFICAR) %%%%%%%%%%%

nom = 'text1'; %string
nom_cataleg = 'cataleg_alfanumeric_10.mat';
%
%-----

% NO MODIFICAR

%%% PESOS DELS VALORS I VECTORS SINGULARS
% En els vectors s'indica el pes per cada valor singular (sense comptar el
% primer) per ordre en taxa_d i per a cada vector singular per ordre en
% taxa_v i taxa_u
taxa_d=[1 0.4]; %vector de llargada dimensio
taxa_u=[1 0.4 0.2]; %vector de llargada b (sortida)
taxa_v=[1 0.4 0.2]; %vector de llargada b (arribada)

dimensio=size(taxa_d,2);
base=size(taxa_v,2);

directori=['textos/' nom '/'];
fitxer=[directori,nom '.txt'];
format='tif';
imatge=[directori 'flux_' nom];

[lletres,linies,ll_linia]=segmentacio(imatge,format);
[signatura]=signatures(lletres,dimensio,base);
[deteccio_numero,deteccio_lletra]=OCR(signatura,nom_cataleg,dimensio,base,taxa
_d,taxa_u,taxa_v);
[reallletres, realparaules, reallinees]=lectura(fitxer);
resultats(directori,nom,nom_cataleg,deteccio_lletra,reallletres);

```

A.4 Creació de catalegs

crear_cataleg.m

```
function crear_cataleg()
%Funcio que crea un cataleg a partir d'una plantilla d'imatge en format
%tiff i d'un fitxer de text amb el contingut de la plantilla

%Guarda:
% - cataleg: vector anomenat que conte les signatures de cadascuna de les
%lletres de la plantilla
% - solucio.mat:
%     - solucio: index de les lletres per trobar la lletra corresponent a
%     cada signatura
%     - alfa: string amb totes les lletres del cataleg (ordenades,
%     corresponent a l'index)

%Jose de la Puente, UPC, Barcelona
%2017/01/14

%Provada i funciona

clear all;

%%%%%%%%%%  NOM DE LA PLANTILLA, DIMENSIO I FLUX (MODIFICAR) %%%%%%%%%%%

nom_plantilla = 'plantilla_alfabet';
nom_cataleg = 'cataleg_alfabet';
%
%
% NO MODIFICAR
directori=['catalegs/' nom_plantilla '/'];
fitxer=[directori nom_plantilla '.txt'];
format='tif';
imatge=[directori 'flux_' nom_plantilla];

[lletres,~,~]=segmentacio(imatge,format);
[reallletres,~,~]=lectura(fitxer);
alfa=reallletres;
cataleg=signatures(lletres,2,3);
save (nom_cataleg,'alfa','cataleg');
end
```

A.5 Funcions de flux de tinta

flux_tinta_negra.m

```
function flux_tinta_negra(imatge_flux,format,nfluxes,taxa_cons,taxa_veins)
% Funcio a sobreesciu una imatge donada en format tiff b/n o gris
% aplicant flux de tinta negra.
% Admet creacio neta de tinta, disminucio neta, fluxes negatius
% (=flux de tinta blanca)...
% INPUT: imatge_flux=imatge a ser sobreescrita aplicant el flux,
% format=format del fitxer grafic, nfluxes=nombre d'iteracions del flux,
% taxa_cons=fraccio de la tinta que el pixel conserva,
% taxa_veins=fraccio de la tinta que passa a cadascun dels 4 veins directes,
% taxa_diagonal=fraccio de la tinta que passa a cadascun dels 4 veins en
% diagonal
%
% Jaume Amoros, Jose de la Puente, UPC, Barcelona
% 2017,01,14
%
% provat i funciona

% valors per defecte dels parametres del flux
if nargin<5
    taxa_cons=0.6;
    taxa_veins=0.125;
end;
if nargin<3
    nfluxes=9;
end;

% llegeix imatge com a matriu
Auint = imread(imatge_flux,format);
A = im2double(Auint(:,:,1));
% fa 1=negre, 0=blanc
A=1-A;

% crea matriu ampliada amb veins ficticis a zero
nfil=size(A,1);
ncol=size(A,2);
Aamp=zeros(nfil+2,ncol+2);
Aamp(2:end-1,2:end-1)=A;

% aplica el flux
for k=1:nfluxes
    Aamp(2:end-1,2:end-1)=taxa_cons*Aamp(2:end-1,2:end-1)+taxa_veins*(Aamp(1:end-2,2:end-1)+Aamp(3:end,2:end-1)+Aamp(2:end-1,1:end-2)+Aamp(2:end-1,3:end));
end;
% arrodoneix
Afl=Aamp(2:end-1,2:end-1);
clear Aamp
Afl=round(Afl);
```

```
% escriu la imatge en fitxer tiff b/n (o gris?)
imwrite(1-Afl,imatge_flux,format);
```

aplicar_flux.m

```
function aplicar_flux()
%Crea una copia de la imatge i li aplica el flux de tinta

%Jose de la Puente, UPC, Barcelona
%2017/01/14

nom='ABCD';

directori=['textos/' nom '/'];
format='tif';

original=[directori nom '.tiff']; %fitxer amb la imatge original
imatge_flux=[directori 'flux_' nom '.tiff']; %nou fitxer amb la imatge per
aplicar el flux

copyfile(original,imatge_flux);

iteracions=2; % nombre de vegades que es vol aplicar el flux
nfluxes=9;
taxa_const=0.6;
taxa_veins=0.125;
taxa_diagonal=0;

for i=1:iteracions
    flux_tinta_negra(imatge_flux,format,nfluxes,taxa_const,taxa_veins);
end
```

A.6 Funcions auxiliars

taxa_encerts.m

```
function
taxa_encerts(nom,deteccio_lletra,reallletres,dimensio,base,taxa_d,taxa_u,taxa_
v)
% guarda en un fitxer per a cada linia el nom de la prova i tots els valors
% definits en la variable r

% Jose de la Puente, UPC, Barcelona
% 2017/01/15
n=0;
for i=1:size(deteccio_lletra,2)
```

```

    if i>size(reallletres,2)
        break
    end
    if deteccio_lletra(i)==reallletres(i)
        n=n+1;
    end
end
llargada=size(reallletres,2);
encerts=n;
t_encerts=(n/size(reallletres,2))*100;
r={llargada encerts t_encerts dimensio base taxa_d taxa_v taxa_u}; %resultats

nom_resultats=['resultats/' date '.txt'];
resultats=fopen(nom_resultats,'at');
fprintf(resultats,nom);
fprintf(resultats, ' ');
for i=1:8
    if i==6||i==7||i==8
        fprintf(resultats, '[');
        for j=1:size(r{i},2)
            v=r{i};
            v_j=num2str(v(j));
            fprintf(resultats, v_j);
            if j<size(r{i},2)
                fprintf(resultats, ' ');
            end
        end
        fprintf(resultats, ']');
    else
        r_i=num2str(r{i});
        fprintf(resultats, r_i);
    end
    fprintf(resultats, ' ');
end
fprintf(resultats, '\n');
fclose(resultats);

clear r;

```

crear_cataleg_alfabet.m

```

function crear_cataleg_alfabet()
%crea el cataleg alfabetic

%Jose de la Puente, Barcelona, UPC
%2016/01/10

load('cataleg_majuscules_10.mat')
c_alfa=alfa;
c_cataleg=cataleg;

```

```

load('cataleg_minuscules_10.mat')
alfa=[c_alfa alfa];
c2_cataleg=cataleg;
cataleg={};
cataleg(:,1)={c_cataleg{:,1} c2_cataleg{:,1}};
cataleg(:,2)={c_cataleg{:,2} c2_cataleg{:,2}};
cataleg(:,3)={c_cataleg{:,3} c2_cataleg{:,3}};
save ('cataleg_alfabet_10','alfa','cataleg')

```

crear_cataleg_alfanumeric.m

```

function crear_cataleg_alfanumeric()
%crea el cataleg alfanumeric

%Jose de la Puente, Barcelona, UPC
%2016/01/10

load('cataleg_majuscules_10.mat')
c_alfa=alfa;
c_cataleg=cataleg;

load('cataleg_minuscules_10.mat')
c_alfa=[c_alfa alfa];
c2_cataleg=cataleg;

load('cataleg_nombres_10.mat')
alfa=[c_alfa alfa];
c3_cataleg=cataleg;
cataleg={};
cataleg(:,1)={c_cataleg{:,1} c2_cataleg{:,1} c3_cataleg{:,1}};
cataleg(:,2)={c_cataleg{:,2} c2_cataleg{:,2} c3_cataleg{:,2}};
cataleg(:,3)={c_cataleg{:,3} c2_cataleg{:,3} c3_cataleg{:,3}};
save ('cataleg_alfanumeric_10','alfa','cataleg')

```

crear_cataleg_ABCD.m

```

function crear_cataleg_ABCD()
%crea el cataleg ABCD

%Jose de la Puente, Barcelona, UPC
%2016/01/10

load('cataleg_majuscules_10.mat')
alfa=alfa(1:40);
cataleg=cataleg(1:40,:);
save ('cataleg_ABCD_10','alfa','cataleg')

```

B. Catàlegs

En aquest apartat es presenten las imatges que han servit com a plantilla per a realitzar els catàlegs. Aquest és el model de catàleg definitiu, en el qual cada caràcter està repetit 10 vegades. Ja han sigut tractades amb el programa de flux de tinta negra.

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 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 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 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 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 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 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 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 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 Y Z

Figura B.1 Plantilla de majúscules per al catàleg



a a a a a a a a a a
b b b b b b b b b b
c c c c c c c c c c
d d d d d d d d d d
e e e e e e e e e e
f f f f f f f f f f
g g g g g g g g g g
h h h h h h h h h h
i i i i i i i i i i
j j j j j j j j j j
k k k k k k k k k k
l l l l l l l l l l
m m m m m m m m m m
n n n n n n n n n n
o o o o o o o o o o
p p p p p p p p p p
q q q q q q q q q q
r r r r r r r r r r
s s s s s s s s s s
t t t t t t t t t t
u u u u u u u u u u
v v v v v v v v v v
w w w w w w w w w w
x x x x x x x x x x
y y y y y y y y y y
z z z z z z z z z z

Figura B.2 Plantilla de minúscules per al catàleg

0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9

Figura B.3 Plantilla de nombres per al catàleg

C. Textos

En aquest apartat es mostren les imatges que contenen els textos a ser reconeguts en l'últim apartat de l'estudi (8.6). Ja han sigut tractades amb el programa de flux de tinta negra.

Do you read the Bible
Brett Well there's
this passage I got
memorized sort
of fits this occas
ion Ezekiel twen
ty five seven tee
n the path of the
righteous man is
beset on all sides
by the iniquities
of the selfish an
d the tyranny of
evil men Blessed
is he who in the

Figura C.1 text1

n a m e o f c h a r i t y
a n d g o o d w i l l s h
e p h e r d s t h e w e a
k t h r o u g h t h e v a l
l e y o f d a r k n e s s
f o r h e i s t r u l y h
i s b r o t h e r s k e e
p e r a n d t h e f i n d
e n o f l o s t c h i l d
r e n A n d I w i l l s
t r i k e d o w n u p o n
t h e e w i t h g r e a t
v e n g e a n c e a n d
f u r i o u s a n g e r

Figura C.2 text2

t h o s e w h o w o u
l d a t t e m p t t o
p o i s o n a n d d e
s t r o y m y b r o t h
e r s A n d y o u w i l
l k n o w m y n a m e
i s t h e L o r d w h
e n I l a y m y v e
n g e a n c e u p o n
t h e e

Figura C.3 text3

A B C D C B A C B B D
C B A B C D A D B C C
A B C A D D D A C B C
D A D B D C C D B B A
B B D B A D C D C A A
A C D C A A B B C C C
D D D D A A A B B B B

Figura C.4 ABCD

3 1 4 1 5 9 2 6 5 3 5
2 7 1 8 2 8 1 8 2 8 4
1 6 1 8 0 3 3 9 8 8 7
6 0 2 2 1 0 2 3
6 6 7 4 1 0 1 1
1 3 8 0 6 4 8 8 1 0 2 3
4 1 3 5 6 6 7 3 1 0 1 5
2 9 9 7 9 2 4 5 8
1 4 1 4 2 1 3 5 6 2 3

Figura C.5 nombres1