Semnale și sisteme

Lucrare de laborator nr. 10

Descriptori de formă. Descriptori Fourier

Descriptori de formă. Descriptori Fourier

- caracterizează conturul unui obiect printr-un set de numere care reprezintă conținutul de frecvență al unui semnal unidimensional asociat conturului obiectului
- descriptorii Fourier pot fi calculați utilizând diverse reprezentări ale conturului
- primele componente ale seriei Fourier furnizează informații referitoare la forma generală a obiectului, în timp ce componentele de ordin superior, asociate cu frecvențe mai mari, descriu detalii ale conturului

Descriptori de formă. Descriptori Fourier

- asocierea unui obiect grafic cu un set de numere care cuantizează forma va permite introducerea unor măsuri ale similitudinii între obiecte, cu aplicații importante în **clasificarea** și **recunoașterea** formelor
- un astfel de set este folositor îndeosebi dacă valorile numerice se conservă la transformări de translație, rotație și scalare a formei; în acest caz va poseda un atribut important, numit **invarianță**
- atunci când setul de numere conține suficiente informații despre contur, se poate folosi pentru **reconstrucția** conturului

FUNCȚIA DE CURBURĂ DESCRISĂ PRIN SERIA FOURIER

Exprimând conturul unui obiect ca o funcție complexă $c(p) = x(p) + iy(p)$, unde $(x(p), y(p))$ sunt coordonatele pixelilor de pe contur, unghiul determinat de tangenta la contur în punctul p_j poate fi exprimat sub forma:

$$\varphi(s) = \arctan\left(\frac{y(p_j)-y(p_{j-1})}{x(p_j)-x(p_{j-1})}\right), 0 \le s \le L$$

unde L reprezintă lungimea curbei.

Funcția de curbură măsoară direcția unghiulară a curbei în funcție de lungimea curbei și este descrisă de diferența dintre valori successive ale unghiurilor:

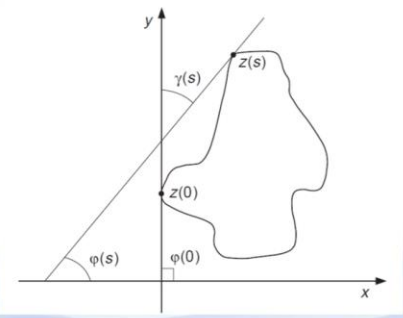
$$\kappa(s) = \varphi(s) - \varphi(s-1)$$

Dezavantaje ale funcției de curbură: sensibilitatea la zgomot (pentru a diminua efectul zgomotului conturul poate fi filtrat) și discontinuitățile.

Pentru a se evita discontinuitățile se poate utiliza *funcția unghiulară cumulativă*:

$$\gamma(s) = \int_0^s \kappa(r)dr - \kappa(0)$$

unde $\kappa(r)$ este curbura în punctul r, $\kappa(0)$ este curbura în punctul de start.



Relația dintre funcția unghiulară $\varphi(s)$ și funcția unghiulară cumulativă $\gamma(s)$.

Valorile inițiale, respectiv finale ale funcției $\gamma(s)$ atunci când conturul este parcurs în sensul acelor de ceas, sunt $\gamma(0)=0$, $\gamma(L)=-2\pi$.

Funcția unghiulară cumulativă normalizată normalizează domeniul de definiție $[0, L]$ al funcției $\gamma(s)$ la intervalul $[0, 2\pi]$ prin parametrizarea $t = 2\pi s/L$:

$$\gamma^*(t) = \gamma\left(\frac{L}{2\pi}t\right) + t$$

unde $\gamma^*(0) = \gamma^*(2\pi) = 0$.

Funcția γ^* este invariantă la translații, rotații și modificări ale lungimii curbei.

Reprezentarea în serie Fourier a lui $\gamma^*(t)$ (funcție periodică cu perioada 2π, $\omega_0 = 1$):

$$x(t) = c_0 + \sum_{k=1}^{+\infty}\left[a_k \cos(kt) + b_k \sin(kt)\right]$$

unde $c_0 = \frac{1}{2\pi}\int_0^{2\pi} \gamma^*(t)dt$,

$$a_k = \frac{1}{\pi}\int_0^{2\pi} \gamma^*(t)\cos(kt)dt, \quad b_k = \frac{1}{\pi}\int_0^{2\pi} \gamma^*(t)\sin(kt)dt$$

Coeficienții seriei Fourier pot fi rescriși în funcție de $\gamma(t)$:

$$c_0 = \frac{1}{2\pi}\int_0^{2\pi} \gamma\left(\frac{Lt}{2\pi}\right)dt + \frac{1}{2\pi}\int_0^{2\pi} t dt = \frac{1}{2\pi}\int_0^{2\pi} \gamma\left(\frac{Lt}{2\pi}\right)dt + \pi,$$

$$a_k = \frac{1}{\pi}\int_0^{2\pi} \gamma\left(\frac{Lt}{2\pi}\right)\cos(kt)dt + \frac{1}{\pi}\int_0^{2\pi} t\cos(kt)dt = \frac{1}{\pi}\int_0^{2\pi} \gamma\left(\frac{Lt}{2\pi}\right)\cos(kt)dt,$$

$$b_k = \frac{1}{\pi}\int_0^{2\pi} \gamma\left(\frac{Lt}{2\pi}\right)\sin(kt)dt + \frac{1}{\pi}\int_0^{2\pi} t\sin(kt)dt = -\frac{2}{k} + \frac{1}{\pi}\int_0^{2\pi} \gamma\left(\frac{Lt}{2\pi}\right)\sin(kt)dt$$

Exprimând coeficienții seriei Fourier în termeni de distanțe: $s = \frac{Lt}{2\pi}$, $0 \le s \le L$, $dt = \frac{2\pi}{L}ds$,

$$c_0 = \pi + \frac{1}{L}\int_0^L \gamma(s)ds,$$

$$a_k = \frac{2}{L}\int_0^L \gamma(s)\cos\left(\frac{2\pi k}{L}s\right)ds, \quad b_k = -\frac{2}{k} + \frac{2}{L}\int_0^L \gamma(s)\sin\left(\frac{2\pi k}{L}s\right)ds$$

Aproximând integrala $\int_0^L \gamma(s)ds$ prin suma ariilor dreptunghiurilor de lungime τ_i și înălțime γ_i:

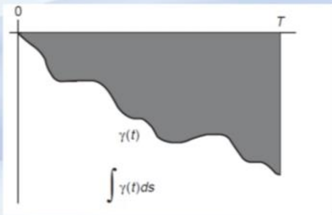
$$c_0 = \pi + \frac{1}{L}\sum_{i=1}^{m} \gamma_i \tau_i,$$

$$a_k = \frac{2}{L}\sum_{i=1}^{m} \gamma_i \tau_i \cos\left(\frac{2\pi k}{L}s_i\right),$$

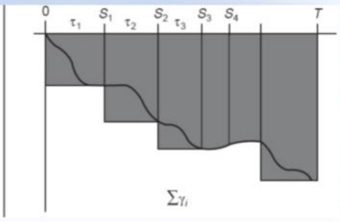
$$b_k = -\frac{2}{k} + \frac{2}{L}\sum_{i=1}^{m} \gamma_i \tau_i \sin\left(\frac{2\pi k}{L}s_i\right)$$

s_i - lungimea curbei din punctul de start în punctul *i* ($s_i = \sum_{r=1}^{i} \tau_r$),

τ_r - distanța dintre punctul *r*-1 și punctul *r*

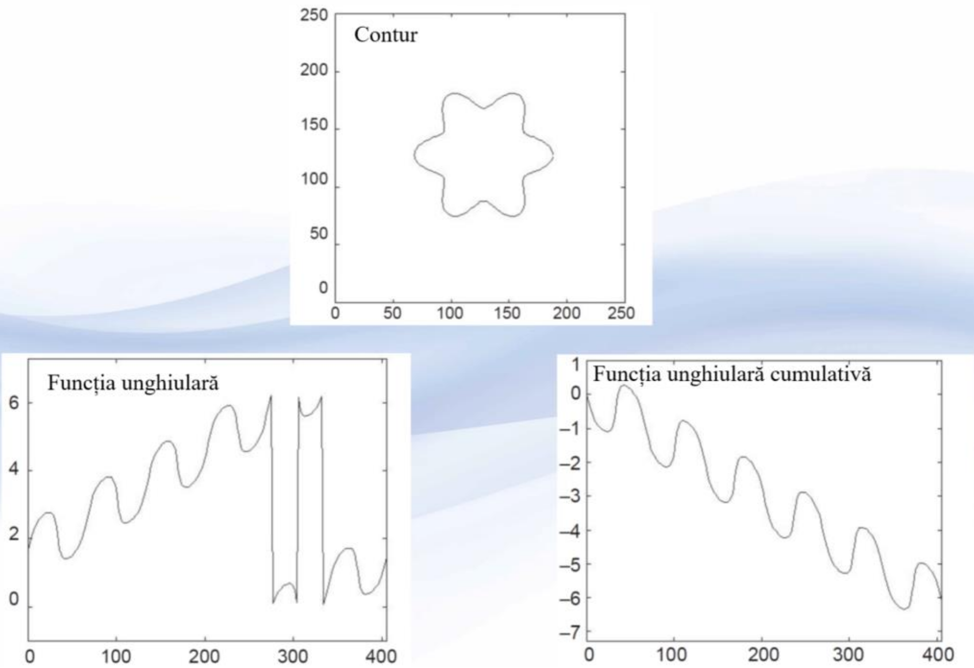


Curba continua



Suma Riemman

```
function angularFunctionDescriptors(curve, nc)

X=curve(1,:); Y=curve(2,:); m=length(X);
figure, subplot(2,3,1);plot(X,Y), title('Contur');

% calcul distante S_i
S=zeros(1,m); S(1) = sqrt( (X(1)-X(m))^2 + (Y(1)-Y(m))^2 );
for i=2:m, S(i) = S(i-1)+sqrt( (X(i)-X(i-1))^2 + (Y(i)-Y(i-1))^2 ); end
L=S(m); % L - lungime perimetru

% calculul functiei de curbura
wnd = 10; A = zeros(1,m);
for i=1:m
   % medierea coordonatelor pixelilor de pe contur pe o fereastra de dimensiune wnd pentru atenuarea zgomotului
   x1=0; x2=0; y1=0; y2=0;
   for j=1:wnd
      pa=i-j; pb=i+j; if(pa<1), pa=m+pa; end, if(pb>m), pb=pb-m; end
      x1=x1+X(pa); y1=y1+Y(pa); x2=x2+X(pb); y2=y2+Y(pb);
   end
   x1=x1/wnd; y1=y1/wnd;    x2=x2/wnd; y2=y2/wnd;

   % calculul valorii functiei de curbura in punctul i de pe contur
   dx=x2-x1;    dy=y2-y1;
   if(dx==0) dx=.00001; end
   if ((dx>0) && (dy>0)), A(i)=atan(dy/dx);
   elseif ((dx>0) && (dy<0)), A(i)=atan(dy/dx)+2*pi;
   else A(i)=atan(dy/dx)+pi; end
end
subplot(2,3,2);plot(S,A); axis([0,S(m),-1,2*pi+1]); title('\phi(s)')

% calcul functie unghiulara cumulativa
G = zeros(1,m);
for i = 2:m
    dA = A(i)-A(i-1); d = min(abs(dA),abs(abs(dA)-2*pi));
    if (d > 0.5),       G(i) = G(i-1);
    elseif (dA < -pi),  G(i) = G(i-1) - (dA + 2*pi);
    elseif (dA > pi),   G(i) = G(i-1) - (dA - 2*pi);
    else,               G(i) = G(i-1) - dA; end
end
subplot(2,3,3); plot(S,G); axis([0,S(m),-2*pi-1,1]); title('Functie unghiulara cumulativa \gamma(s)')
t=(2*pi*S)/L;  F = G + t;
subplot(2,3,4);
plot(t,F); axis([0,2*pi,-2*pi,2*pi]); title('Functie unghiulara cumulativa normalizata \gamma^*(t)')

% calcul descriptori Fourier
a = zeros(1,nc); b = zeros(1,nc);
for k = 1:nc
    a(k) = a(k) + G(1)*(S(1))*cos(2*pi*k*S(1)/L);
    b(k) = b(k) + G(1)*(S(1))*sin(2*pi*k*S(1)/L);
    for i = 2:m
        a(k) = a(k) + G(i)*(S(i)-S(i-1))*cos(2*pi*k*S(i)/L);
        b(k) = b(k) + G(i)*(S(i)-S(i-1))*sin(2*pi*k*S(i)/L);
    end
    a(k) = a(k)*(2/L); b(k) = b(k)*(2/L)-2/k;
end
subplot(2,3,5); bar(a); title('Descriptori Fourier a_k'), subplot(2,3,6); bar(b); title('Descriptori Fourier b_k')
```



DESCRIPTORI FOURIER ELIPTICI

Exprimând conturul ca o funcție complexă $c(t) = x(t) + iy(t)$, reprezentarea în serie Fourier a acestuia poate fi scrisă sub forma

$$c(t) = \frac{a_{x0}}{2} + \sum_{k=1}^{\infty}\left(a_{xk}\cos(k\omega t) + b_{xk}\sin(k\omega t)\right) + j\left(\frac{a_{y0}}{2} + \sum_{k=1}^{\infty}\left(a_{yk}\cos(k\omega t) + b_{yk}\sin(k\omega t)\right)\right)$$

sau în formă matriceală:

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \frac{1}{2}\begin{bmatrix} a_{x0} \\ a_{y0} \end{bmatrix} + \sum_{k=1}^{\infty}\begin{bmatrix} a_{xk} & b_{xk} \\ a_{yk} & b_{yk} \end{bmatrix}\begin{bmatrix} \cos(k\omega t) \\ \sin(k\omega t) \end{bmatrix}$$

Componenta de ordinul zero a seriei Fourier, $\frac{1}{2}\left(a_{x0}, a_{y0}\right)$, reprezintă centrul de masă al formei.

Punctele de eșantionare sunt definite în $t = i\tau$, unde *i* este un întreg $1 \leq \ i \leq m$ (m – numărul de puncte egal distanțate de pe contur). x_i și y_i definesc valoarea funcțiilor în punctul de eșantionare *i*: $x_i = x(i\tau)$, $y_i = y(i\tau)$. Coeficienții seriei Fourier sunt:

$$a_{xk} = \frac{2}{m}\sum_{i=1}^{m} x_i\cos(k\omega i\tau), \quad b_{xk} = \frac{2}{m}\sum_{i=1}^{m} x_i\sin(k\omega i\tau)$$

$$a_{yk} = \frac{2}{m}\sum_{i=1}^{m} y_i\cos(k\omega i\tau), \quad b_{yk} = \frac{2}{m}\sum_{i=1}^{m} y_i\sin(k\omega i\tau)$$

```
function ellipticFourierDescriptors(c, nc)
X = c(1,:); Y = c(2,:);
m = length(X);
T=m; w = 2*pi/T; tau = T/m;
% calculul coeficientilor seriei Fourier
ax=[]; bx=[];
for k=1:nc
    ax(k) = 0; bx(k) = 0;
    for i=1:m
        ax(k) = ax(k) + X(i)*cos(k*w*i*tau);
        bx(k) = bx(k) + X(i)*sin(k*w*i*tau);
    end
end
ax = (2/m)*ax; bx = (2/m)*bx; ax0 = 2/m*sum(X); bx0 = 0;
ay=[]; by=[];
for k=1:nc
    ay(k) = 0; by(k) = 0;
    for i=1:m
        ay(k) = ay(k) + Y(i)*cos(k*w*i*tau);
        by(k) = by(k) + Y(i)*sin(k*w*i*tau);
    end
end
ay = (2/m)*ay; by = (2/m)*by; ay0 = 2/m*sum(Y); by0 = 0;
```

```
% reconstructia conturului pe baza coeficientilor Fourier
for n = 1:m
    X2(n) = ax0/2;
    Y2(n) = ay0/2;
    for k = 1:nc
        X2(n) = X2(n) + ax(k)*cos(k*w*n)+bx(k)*sin(k*w*n);
        Y2(n) = Y2(n) + ay(k)*cos(k*w*n)+by(k)*sin(k*w*n);
    end
end
c2 = [X2; Y2];
figure, subplot(3,2,1), plot(c(1,:),c(2,:)); title('Conturul obiectului')
subplot(3,2,2), plot(c2(1,:),c2(2,:)); hold on; plot(ax0/2,ay0/2,'*'); title('Contur reconstruit')
subplot(3,2,3), bar(ax), title('a_x'), subplot(3,2,4), bar(bx), title('b_x')
subplot(3,2,5), bar(ay), title('a_y'), subplot(3,2,6), bar(by), title('b_y')
```

Extragerea conturului obiectelor dintr-o imagine în Matlab

```
im = imread('circles.png'); % incarcare imagine in Matlab
imshow(im) % afisare imagine

% binarizare imagine (in urma acestei operatii pixelii apartinand obiectelor trebuie sa fie albi, iar pixelii
apartinand fundalului negri)
if (~islogical(im))
   if (ndims(im)>2), im = rgb2gray(im);    end
   level=graythresh(im);  BW = im2bw(im,level);
else
   BW = im;
end
figure, imshow(BW)
% daca in urma binarizarii fundalul este alb si obiectele negre, se va complementa imaginea
% BW = ~BW;
% figure, imshow(BW)
```

Extragerea conturului obiectelor dintr-o imagine in Matlab

```
% segmentare imagine si extragere contur obiecte
[B,L,N] = bwboundaries(BW,'noholes');
% N - numarul de obiecte gasite in imagine;
% B - structura continand conturul fiecarui obiect;
% L – matrice 2D in care sunt etichetate regiunile gasite

% afisarea conturului fiecarui obiect din imagine
for (k=1:N)
    X = B{k}(:,2);
    Y = B{k}(:,1);
    contur = [X'; Y'];
    figure, plot(X, Y)
end
```

Exerciții

1. a. Utilizati un program, precum utilitarul Paint, pentru a crea o imagine cu fundal negru conținând 3 obiecte gri pline (un cerc, un dreptunghi și un romb) și salvați imaginea.

b. Încărcați și pregătiți imaginea pentru aplicarea descriptorilor de formă (binarizare și segmentare).

c. Calculați și afișati descriptorii Fourier pentru cele trei obiecte din imagine.

d. Schimbați numărul de coeficienți ai seriei Fourier. Ce observați?

Exerciții

2. a. Creați o imagine cu un dreptunghi plin și salvati imaginea.

b. Incarcati si pregatiti imaginea în Matlab pentru aplicarea descriptorilor de forma.

c. Aflati coordonatele centrului de masa al dreptunghiului utilizand functia *regionprops* (functie care calculeaza proprietati precum aria, dreptunghiul de incadrare, perimetrul, centrul de masa pentru fiecare regiune etichetata din matricea L returnata de functia *bwboundaries*)

```
stats = regionprops(L,'Centroid');
coordinates = stats(1).Centroid;
xc = coordinates(1);
yc = coordinates(2);
```

d. Pentru fiecare punct de pe conturul dreptunghiului calculati distanta dintre acest punct si centrul de masa: $r(j) = \sqrt{\left(x(p_j) - x_c\right)^2 + \left(y(p_j) - y_c\right)^2}$. Reprezentati grafic vectorul r.

e. Ce observati referitor la periodicitatea functiei r?

f. Aratati cum poate fi folosit semnalul r pentru a verifica similitudinea a doua forme.

Exerciții

3. Sa se reconstruiasca conturul obiectului (se vor considera 327 de puncte pe contur) descris de coeficientii Fourier eliptici:

ax(1) = -44.2, ax(2) = 0.6, ax(3) = 0.7, ax(4) = -0.4, ax(5) = -4.8, ax(6) = -0.01, ax(7) = -3.7,

bx(1) = 1.2, bx(2) = 0.1, bx(3) = -0.03, bx(4) = -0.2, bx(5) = 0.5, bx(6) = 0.1, bx(7) = 0.9,

ay(1) = -1.2, ay(2) = -0.1, ay(3) = 0.1, ay(4) = 0.2, ay(5) = 0.6, ay(6) = -0.3, ay(7) = -0.6,

by(1) = -43.8, by(2) = 0.8, by(3) = 0.1, by(4) = 0.4, by(5) = 3.7, by(6) = -0.6, by(7) = -3.6,

ax0 = 141.6, ay0 = 121.3

Sa se afiseze conturul obtinut si centrul de masa.