

Príznaky

Cvičenia z Počítačového Videnia 2

Zuzana Haladová

Príznamy

- Príznamy
- Segmentácia pomocou K-means
- Popis oblastí- Region props
- Freeman code
- Počítanie obvodu

Príznaky

- Globálne príznaky (Nízka, Stredná, Vysoká úroveň)
 - Tvar
 - Farba
 - Textúra
- Lokálne príznaky
 - Príznaky invariantné affiným transformáciám (SIFT,SURF)
- Príznak/príznakový vektor (intenzity, histogram, HOG)

Tvar objektu

- Šedoúrovňový obrázok => treshold => Binárny obrázok
=>Segmentácia => Číslo oblastí
- Vyhľadanie oblastí (Spojité komponenty)
- Vlastnosti oblastí
 - Tvar
 - Veľkosť
 - Orientácia hlavnej osi
 - ...

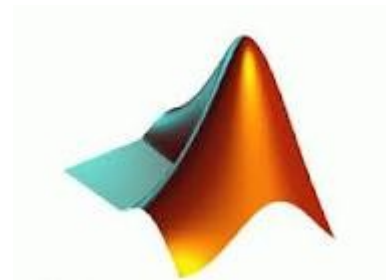
MATLAB

Rýchle opakovanie

Case sensitive
; na konci
% komentár

Matice
zeros(2,3);
K= sum(v);
K(1,2)=0
K(:,2)=0
K(:)=0
b^2 nieje b.^2

K = I (:,:,1)>100;
sum(K(:));



MATLAB

m-files

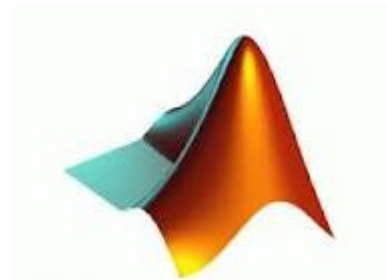
```
function [out] = nazov(in)
out= nazov(in);
```

```
fprintf ('Hello World!');
fprintf ('Hodnota x je %g', x);
```

```
x = linspace(0, 5, 100);
y1 =x;
plot(x,y1); grid;
```

Obrázky sú matice
True color vs. Indexed image

```
I=imread('nazov');
[X,map] = rgb2ind(I, n)
I(:, :, 1)
figure; imshow(I);
```



MATLAB

m-files

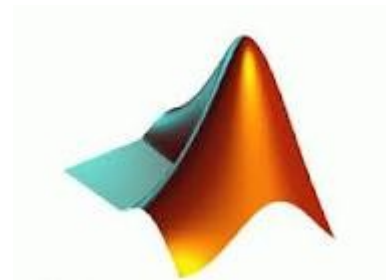
```
function [out] = nazov(in)
out= nazov(in);
```

```
fprintf ('Hello World!');
fprintf ('Hodnota x je %g', x);
```

```
x = linspace(0, 5, 100);
y1 =x;
plot(x,y1); grid;
```

Obrázky sú matice
True color vs. Indexed image

```
I=imread('nazov');
[X,map] = rgb2ind(I, n)
I(:, :, 1)
figure; imshow(I);
```



Segmentácia

- K-means clustering

Minimalizuje vnútrotriednu vzdialenosť, pre K tried (klastrov)

$$W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(j)=k} \|x_i - x_j\|^2 = \sum_{k=1}^K N_k \sum_{C(i)=k} \|x_i - m_k\|^2$$

```
I=imread('1.jpg');
```

```
G=rgb2gray(I);
```

```
IK=double(double(G)/255);
```

```
J = reshape(kmeans(IK(:,3),3),size(IK));    % K=3
```

```
imshow(J,colormap('prism'));
```

Tvar objektu

```
STATS = regionprops(bwlabel(BW),'centroid');  
STATS.Centroid(:)  
figure; imshow(BW); hold on;  
plot(STATS.Centroid(1),STATS.Centroid(2),'*');
```

Vykreslite bounding box

'Area'	'EulerNumber'	'Orientation'
'BoundingBox'	'Extent'	'Perimeter'
'Centroid'	'Extrema'	'PixelIdxList'
'ConvexArea'	'FilledArea'	'PixelList'
'ConvexHull'	'FilledImage'	'Solidity'
'ConvexImage'	'Image'	'SubarrayIdx'
'Eccentricity'	'MajorAxisLength'	
'EquivDiameter'	'MinorAxisLength'	

Freemanov kód

4,8 susednosť

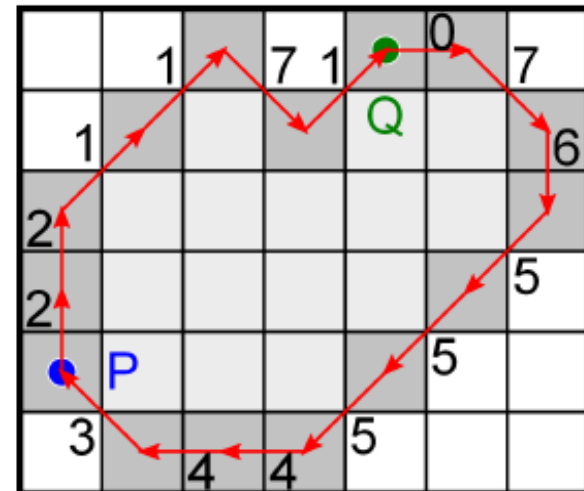
P: 221171076555443

Shift na najmenšie číslo

Q: 076555443221171

Invariantnosť

P: 070627777007077



$$c'_i = \begin{cases} (c_{i+1} - c_i) \bmod 8 & \text{pre } 1 \leq i < N, \\ (c_1 - c_N) \bmod 8 & \text{pre } i = N. \end{cases}$$

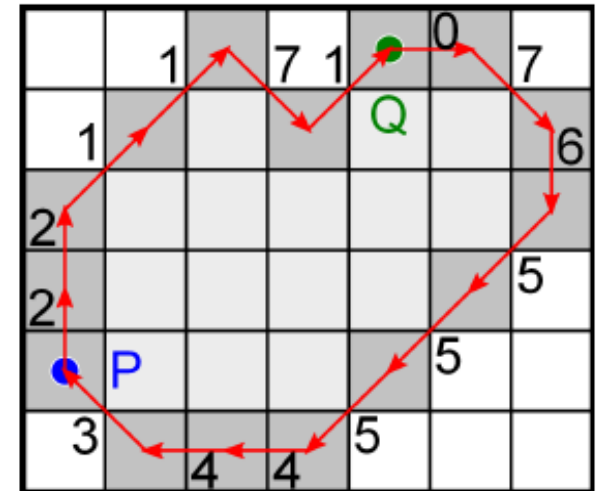
Freemanov kód

Ako na to

Šedoúrovňový, binárny, hranica, kód

```
B = bwtraceboundary(G,[118,178], 'NW');  
figure; imshow(G); hold on; line(B(:,1),B(:,2));
```

Ako by ste našli bod kontúry automaticky?



Freemanov kód

Obvod - Perimeter

$$P = N_p + N_n$$

počet párnych resp. nepárnych číslíc v kóde

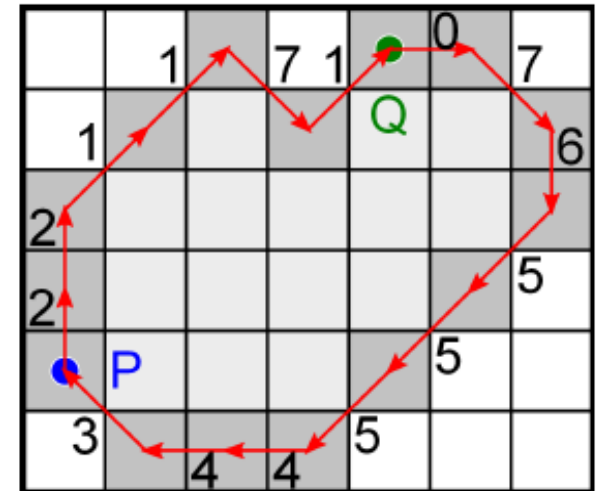
Dĺžka nie je invariantná voči otočeniu

$$P = 0,948 * N_p + 1,340 * N_n$$

Váhovanie

$$P = 0,980 * N_p + 1,406 * N_n - 0,091 * N_r$$

Počet rohov N_r (miest, kde sa mení kód)



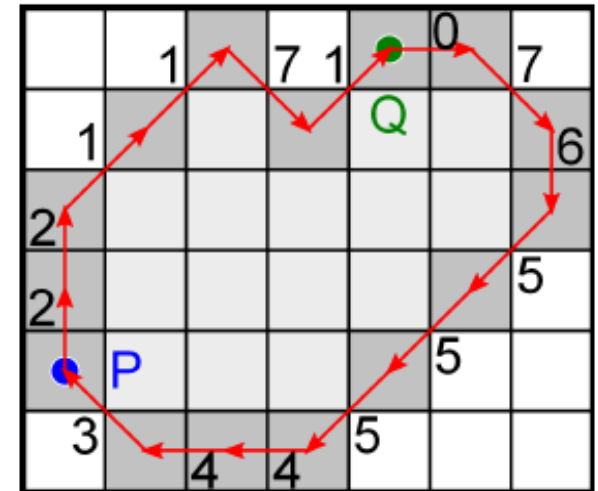
Freemanov kód

Obvod - Perimeter

Spočítajte pre rotáciu o 30° , 45° , 60°

`J = imrotate(I,-30,'bilinear','crop');`

Porovnajte rozne metody pocitania obvodu



Priznaky

Ako by ste postupovali ak by ste mali spraviť vlastné OCR pre rukou písané znaky?

