

Počítačové videnie - Prízny

Ing. Viktor Kocur
viktor.kocur@fmph.uniba.sk

DAI FMFI UK

25.2.2020

Hľadanie objektov

- Binarizácia

- Označenie komponentov

Príznačky

- Príznačky segmentovaných oblastí

- Využitie príznakov na lepšiu segmentáciu

Vyhľadanie vzorového objektu v obrázku

- Úloha

- Postup

Invariancia príznakov?

- Obvod

Pôvodný obrázok



Binarizácia

Binarizácia

```
I = imread('spendliky.png')  
BW = imbinarize(I)  
imshow(BW)
```

Binarizácia

Binarizácia

```
I = imread('spendliky.png')  
BW = imbinarize(I)  
imshow(BW)
```

Zlý prah

Zvolený prah oddeluje špendlíky na príliš veľa kusov

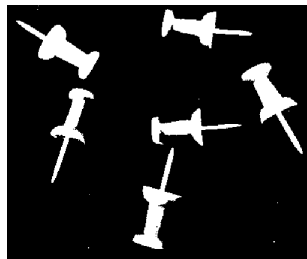
Úloha

Použite metódu `imbinarize(I, threshold)` vo for cykle a nájdite ideálny prah. Zíde sa vám funkcia `waitforbuttonpress`.

Hľadanie prahu

Riešenie

```
I = imread('spendliky.png')  
for level = 0.4:0.02:0.6  
    disp(level);  
    BW = imbinarize(I, level);  
    imshow(BW);  
    waitforbuttonpress;  
end
```



Vhodný prah 0.48

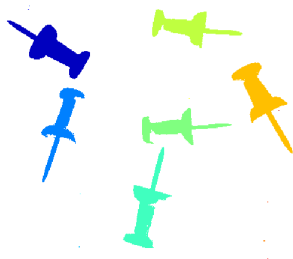
Označenie komponentov

bwlabel

$[L, \text{num}] = \text{bwlabel}(\text{BW})$ - v L vráti tzv. label maticu veľkosti obrázka, ktorá má na pozíciách samostatných objektov číslo daného objektu, do num vráti počet objektov

label2rgb

$\text{RGB} = \text{label2rgb}(L)$ - vráti RGB obrázok, kde sú samostatné objekty nakreslené inou farbou



Príznyky v matlabe

regionprops

`s = regionprops(L, 'property')` - vráti štruktúru obsahujúcu pre pole `property` výstup pre danú vlastnosť. Pozor formát tohto výstupu môže byť rôzny! Treba čítať `help`.

Properties

'Area', 'Centroid', 'BoundingBox', 'SubarrayIdx',
'MajorAxisLength', 'MinorAxisLength', 'Eccentricity', 'Orientation',
'ConvexHull', 'ConvexImage', 'ConvexArea', 'Image', 'FilledImage',
'FilledArea', 'EulerNumber', 'Extrema', 'EquivDiameter', 'Solidity',
'Extent', 'PixelIdxList', 'PixelList', 'Perimeter', 'PerimeterOld',
'PixelValues', 'WeightedCentroid', 'MeanIntensity', 'MinIntensity',
'MaxIntensity'

Prízny v matlabe - príklad

Príklad

```
s = regionprops(L,'Centroid')
centers = reshape([s.Centroid],2,[])
imshow(I);
hold on;
plot(centers(1,:),centers(2,:), 'r*');
hold off;
```

Prízny v matlabe - príklad

Príklad

```
s = regionprops(L,'Centroid')
centers = reshape([s.Centroid],2,[])
imshow(I);
hold on;
plot(centers(1,:),centers(2,:), 'r*');
hold off;
```

Pozor!

Máme až 15 objektov aj na zlých miestach

Príznamy v matlabe - obsah

Príklad

```
s = regionprops(L,'Area')  
areas = [s.Area];  
idx = find(areas > 10)  
BW = ismember(L, idx)
```

ismember

`C = ismember(A,B)` - vráti logickú maticu `C` s rozmermy rovnakými ako `A`, kde 1 je na každom mieste kde sa v `A` nachádza hodnota, ktorá sa nachádza aj niekde v `B` a 0 všade inde

Úloha



Úloha

V obrázku motyle.png identifikujte motýla z motyl3.png. (vyfarbením, hviezdičkou)



Postup

Výber príznakov

Treba brať do úvahy invarianciu voči rotácii! Použite nejaké príznaky z regionprops. Môžete ich aj skombinovať napr. pomer ôs.

Metrika v príznakovom priestore

V prípade že máme definovaný vektor príznakov \vec{a} pre objekt A a \vec{b} pre objekt B . Tak môžeme vytvoriť metriku $\rho(A, B) = \rho(\vec{a}, \vec{b})$. Ak sme dobre vybrali príznaky a metriku, tak náš objekt nájdeme pomocou $\operatorname{argmin}_{K \in \mathbb{K}} (\rho(V, K))$, kde V je náš vzor a \mathbb{K} je množina kandidátov.

$$\text{Napri. : } \rho(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^{\dim(\vec{a})} (a_i - b_i)^2}$$

Obvod

Freeman Code

```
I = imread('jeden.jpg');  
BW = imbinarize(I);  
B = bwboundaries(BW,'noholes');  
F = Freeman_code(B{1});
```

Definície obvodu

N_p, N_n je počet párnych resp. nepárnych čísel v kóde a N_r je počet rohov (2 po sebe idúce čísla sú rôzne):

$$P_s = N_p + N_n$$

$$P_d = N_p + \sqrt{2}N_n$$

$$P_v = 0.948N_p + 1.340N_n$$

$$P_c = 0.980N_p + 1.406N_n - 0.091N_r$$

Obvod - Úloha

Zadanie

Vykreslite plot toho ako sa menia obvody podľa rôznych definícií ak budeme obrázok pred operáciami otáčať pomocou `imrotate(I, uhol, 'bilinear', 'crop')`;

Obvod - Úloha

Zadanie

Vykreslite plot toho ako sa menia obvody podľa rôznych definícií ak budeme obrázok pred operáciami otáčať pomocou `imrotate(I, uhol, 'bilinear', 'crop')`;

Hint

```
Nn = sum(mod(F,2));  
Np = numel(F) - Nn;  
Nr = sum(abs(diff(F)) > 0) + abs(sign(F(1) - F(end)));
```


Obvod - Riešenie

```
angles = -180:2:180;
P = zeros(4, numel(angles));
O = imread('jeden.jpg');
for i = 1:numel(angles)
    I = imrotate(O, angles(i), 'bilinear', 'crop');
    B = bwboundaries(imbinarize(I));
    F = Freeman_code(B{1});
    Nn = sum(mod(F,2));
    Np = numel(F) - Nn;
    Nr = sum(abs(diff(F)) > 0) + abs(sign(F(1) - F(end)));
    P(1,i) = Np + Nn;
    P(2,i) = Np + sqrt(2)*Nn;
    P(3,i) = 0.948*Np + 1.340*Nn;
    P(4,i) = 0.980*Np + 1.406*Nn - 0.091*Nr;
end
plot(angles,P)
```