

Akadémia Trojstenu

Ako naučiť robota spoznávať veci nepravidelných tvarov

Andrej Lúčný

Katedra aplikovanej informatiky

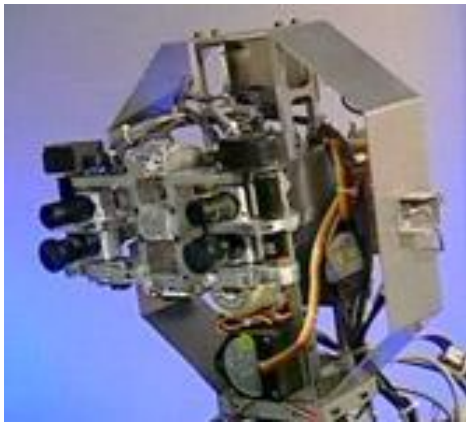
FMFI UK Bratislava

andy@microstep-mis.com

www.microstep-mis.com/~andy

Úloha

Mobilný robot sleduje scénu a má v nej
rozpoznať určitý objekt



COG 1993



iCub 2007

Pravidelné objekty



Objekty, ktoré sa dajú popísať malým počtom parametrov sa dajú rozpoznať spätnou projekciou obrazu do priestoru týchto parametrov (Houghova transformácia)

Pravidelné objekty



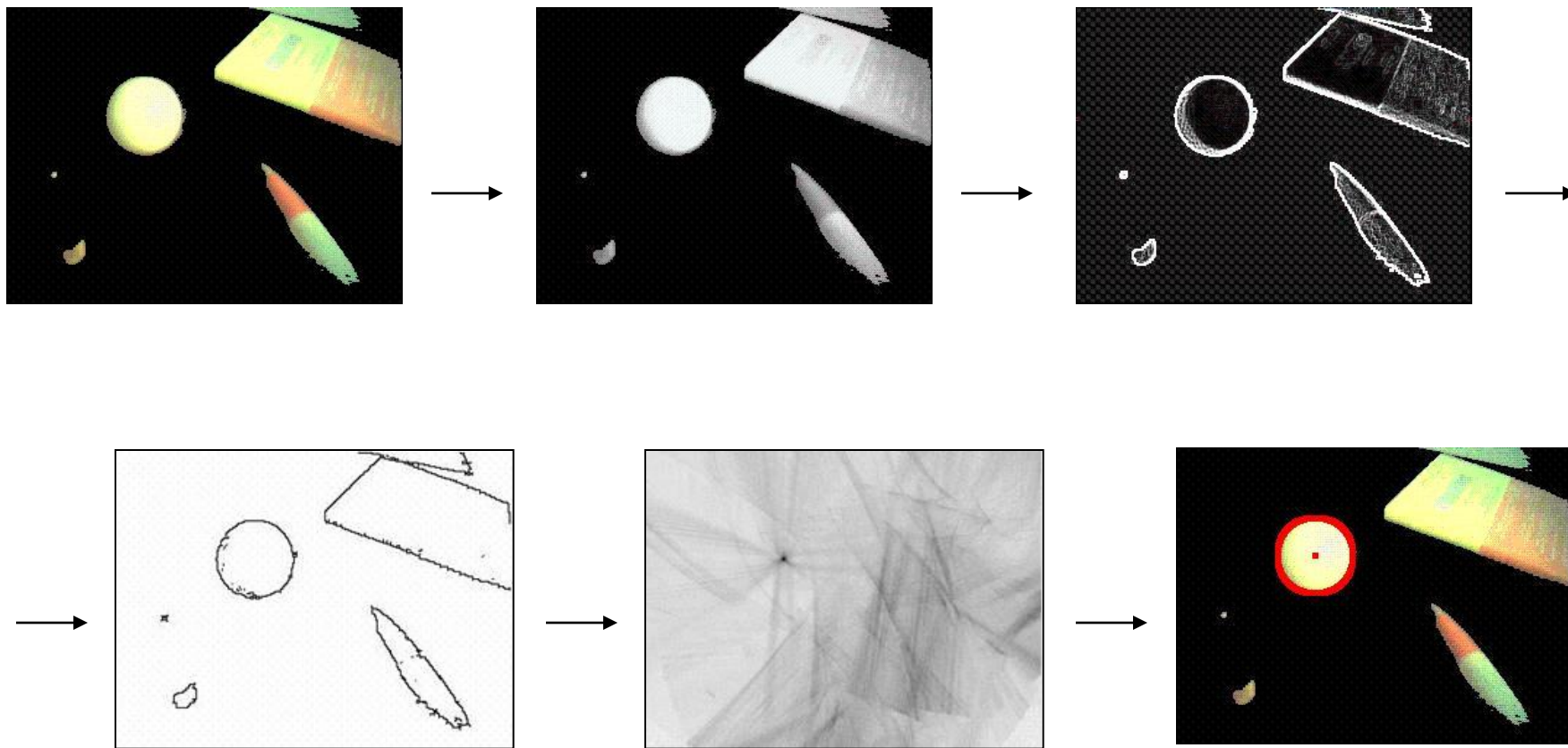
Parametre: (c_x, c_y, r)

- x-súradnica stredu c_x
- y-súradnica stredu c_y
- polomer r

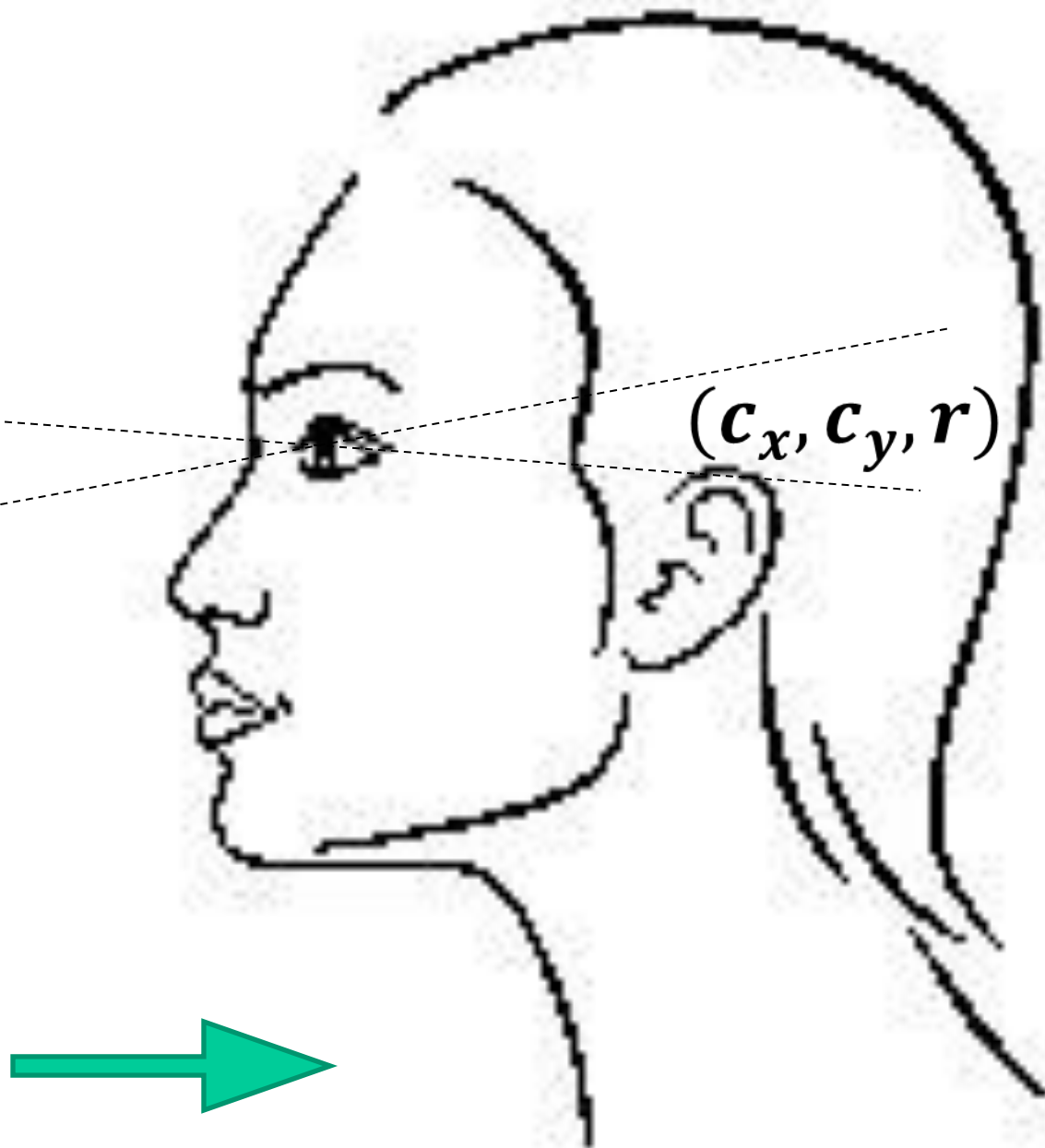
$$(x - c_x)^2 + (y - c_y)^2 = r^2$$

efektívny algoritmus na vykreslenie

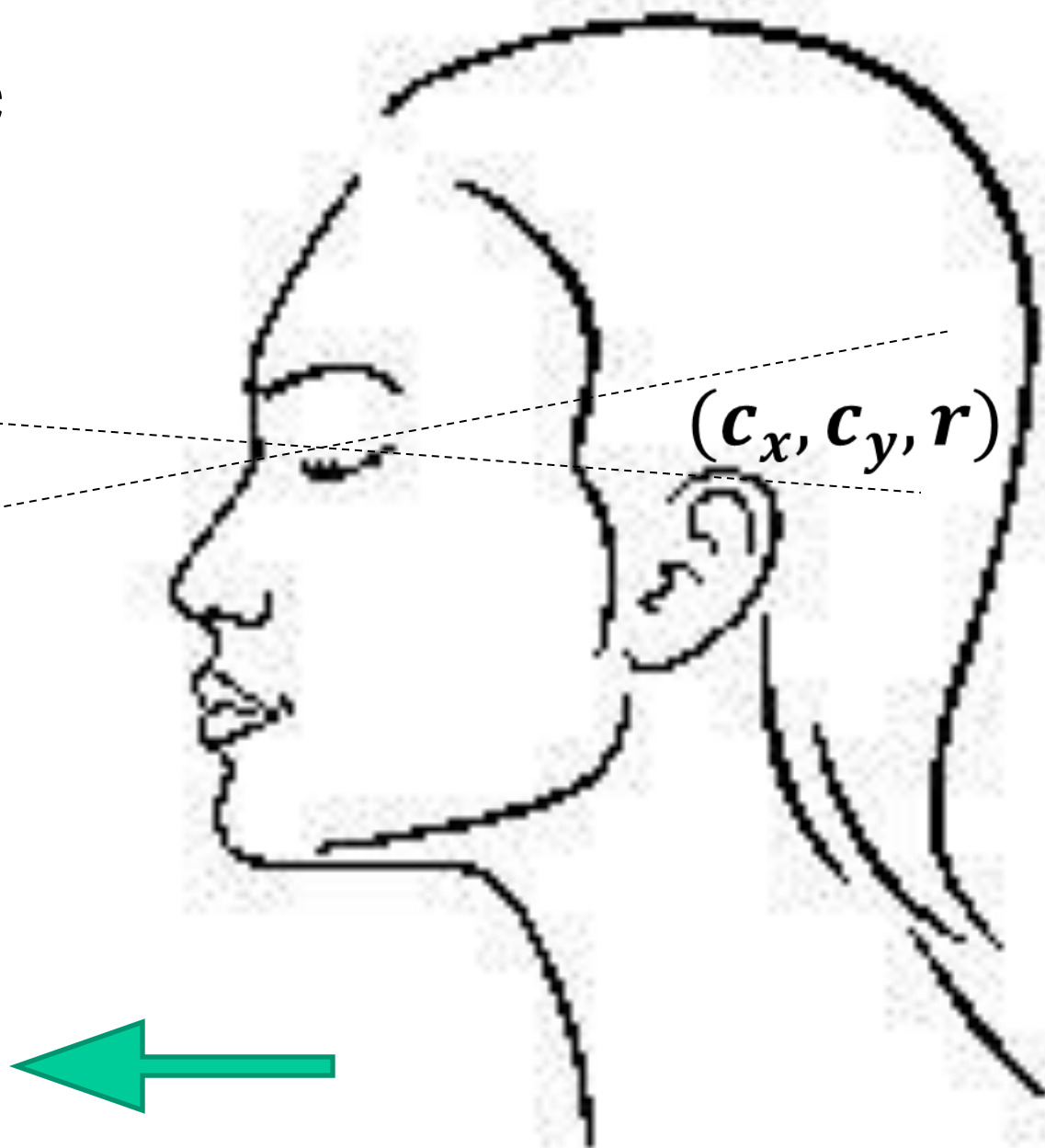
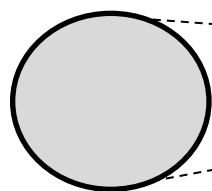
Pravidelné objekty



Pravidelné
objekty



Pravidelné
objekty

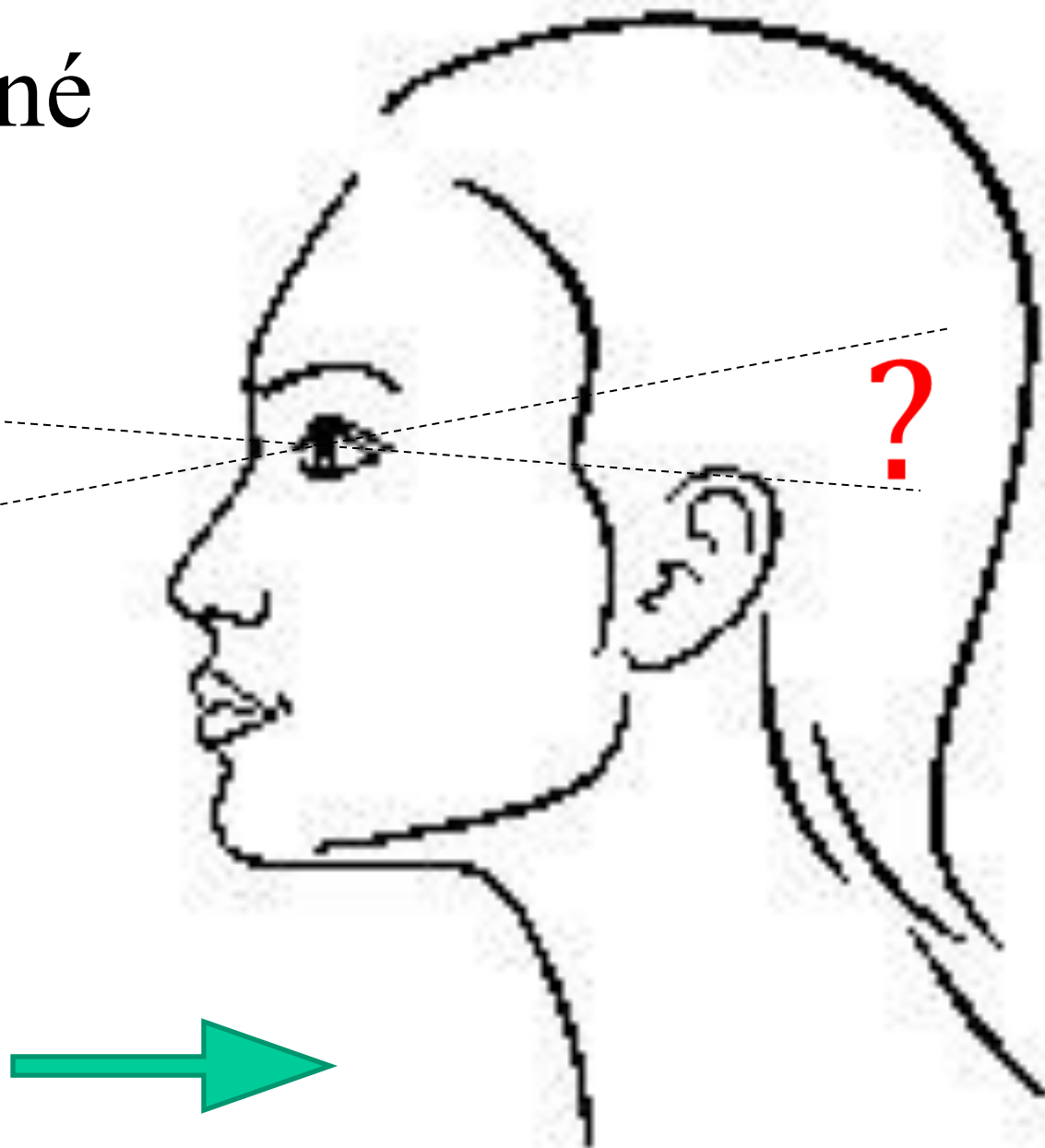


Nepravidelné objekty



Ako ale rozpoznávať objekty, ktoré majú veľa parametrov a nepoznáme ich ?

Nepravidelné objekty



DOT

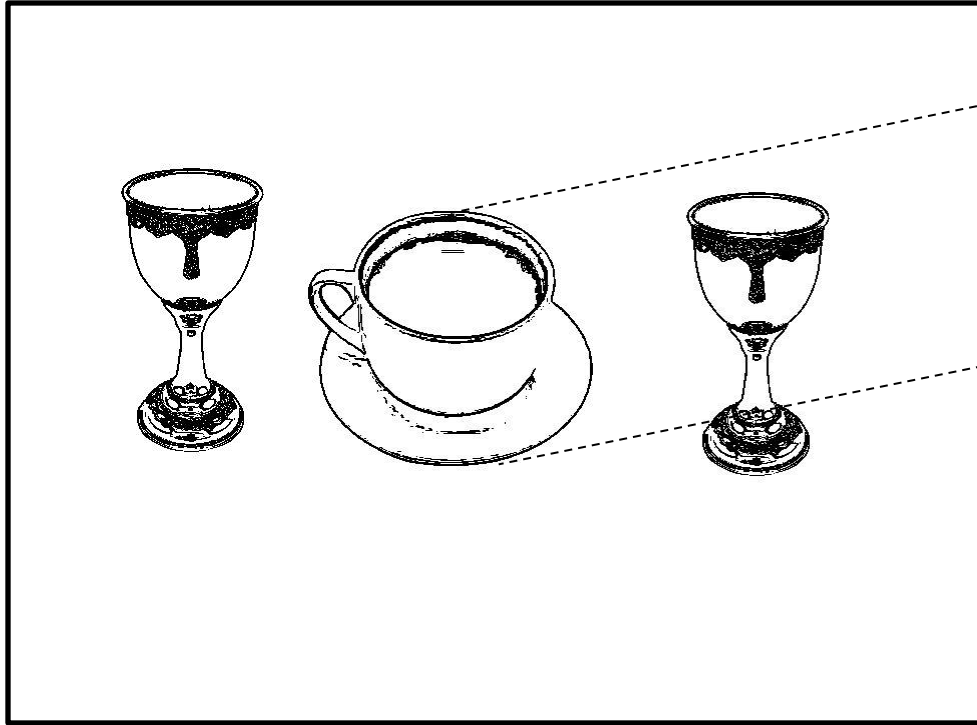
Dominant orientation templates

Šablóny význačných orientácii

- Jedna z najjednoduchších, ale účinných metód na rozpoznávanie objektov nepravidelného tvaru

Motivácia

šablóna



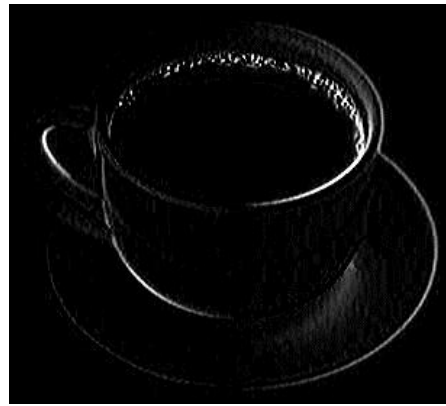
obraz

zamerajme sa
na hrany

Hranový detektor Canny



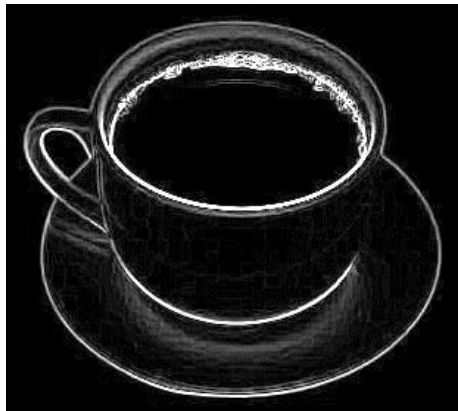
obraz



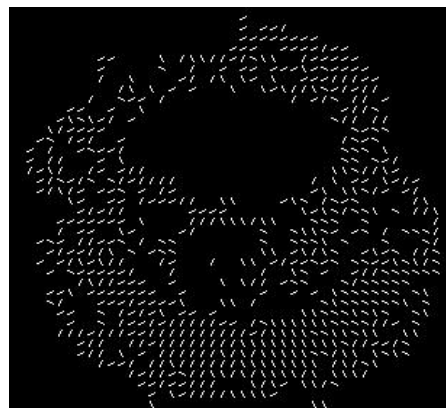
dx



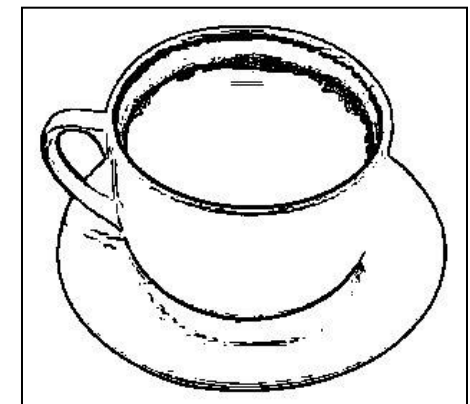
dy



|gradient|



orientácie



hrany

Vstup: obraz z kamery



Tri polia $r[h,w]$, $g[h,w]$, $b[h,w]$, každý ich prvok je číslo $0..255$ a predstavujú červenú, zelenú a modrú zložku farby

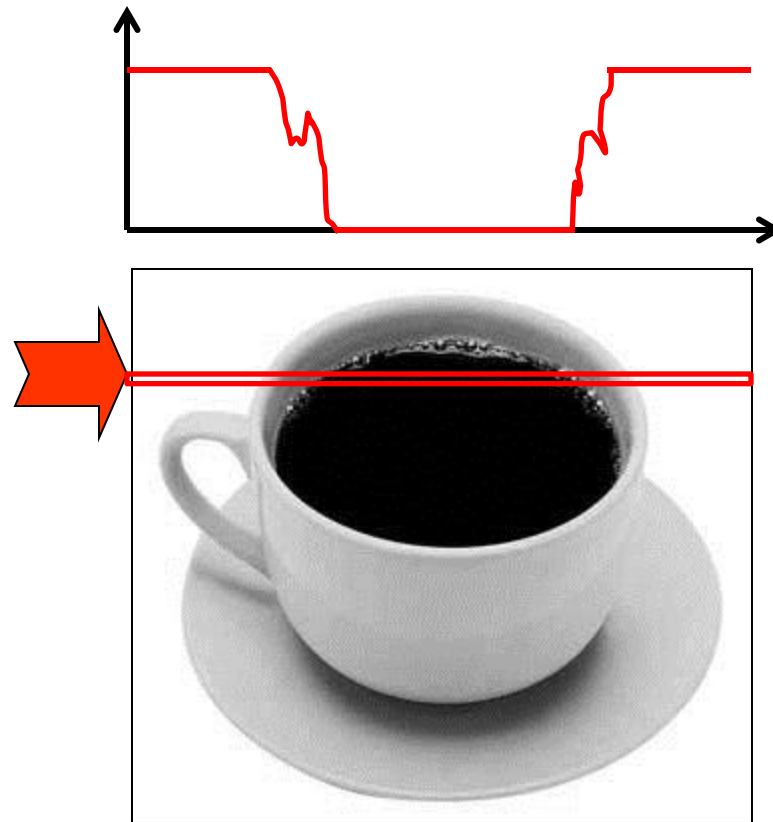
Čiernobiely obraz



Pole $bw[h,w]$, každý jeho prvok je číslo 0..255 a predstavuje intenzitu svetla

$$bw[i,j] = 0.3*r[i,j] + 0.59*g[i,j] + 0.11*b[i,j]$$

Hrany



- Jeden riadok poľa bw si teraz môžeme znázorniť ako funkciu intenzity od čísla stĺpca. Hranám zodpovedajú strmé úseky.

Sobelov operátor

- $dx =$ rozdiely susedných stĺpcov, pričom sa v menšej miere zohľadňujú i susedné riadky

| | | |
|---------------|-------------|---------------|
| $a_{i-1,j-1}$ | $a_{i-1,j}$ | $a_{i-1,j+1}$ |
| $a_{i,j-1}$ | $a_{i,j}$ | $a_{i,j+1}$ |
| $a_{i+1,j-1}$ | $a_{i+1,j}$ | $a_{i+1,j+1}$ |

 \circ

| | | |
|----|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

 $=$

| | | |
|--|-----------|--|
| | | |
| | $b_{i,j}$ | |
| | | |

$$b_{i,j} = | a_{i-1,j+1} + 2a_{i,j+1} + a_{i+1,j+1} - a_{i-1,j-1} - 2a_{i,j-1} - a_{i+1,j-1} |$$

(orežeme na maximálne 255)

Sobelov operátor

- $dy =$ rozdiely susedných riadkov, pričom sa v menšej miere zohľadňujú i susedné stĺpce

| | | |
|---------------|-------------|---------------|
| $a_{i-1,j-1}$ | $a_{i-1,j}$ | $a_{i-1,j+1}$ |
| $a_{i,j-1}$ | $a_{i,j}$ | $a_{i,j+1}$ |
| $a_{i+1,j-1}$ | $a_{i+1,j}$ | $a_{i+1,j+1}$ |

o

| | | |
|----|----|----|
| -1 | -2 | -1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |

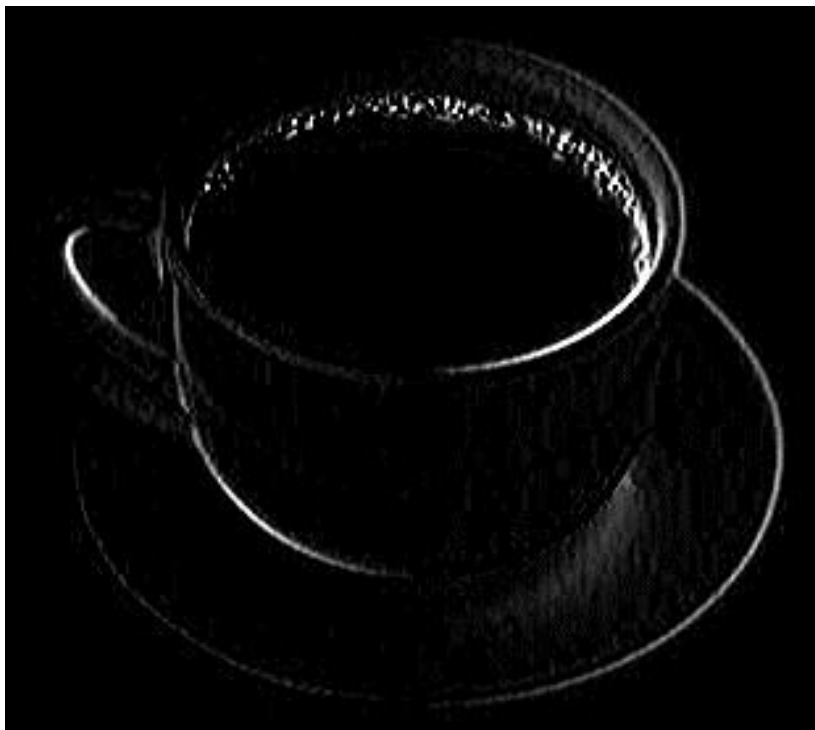
=

| | | |
|--|-----------|--|
| | | |
| | $b_{i,j}$ | |
| | | |

$$b_{i,j} = | a_{i+1,j-1} + 2a_{i+1,j} + a_{i+1,j+1} - a_{i-1,j-1} - 2a_{i-1,j} - a_{i-1,j+1} |$$

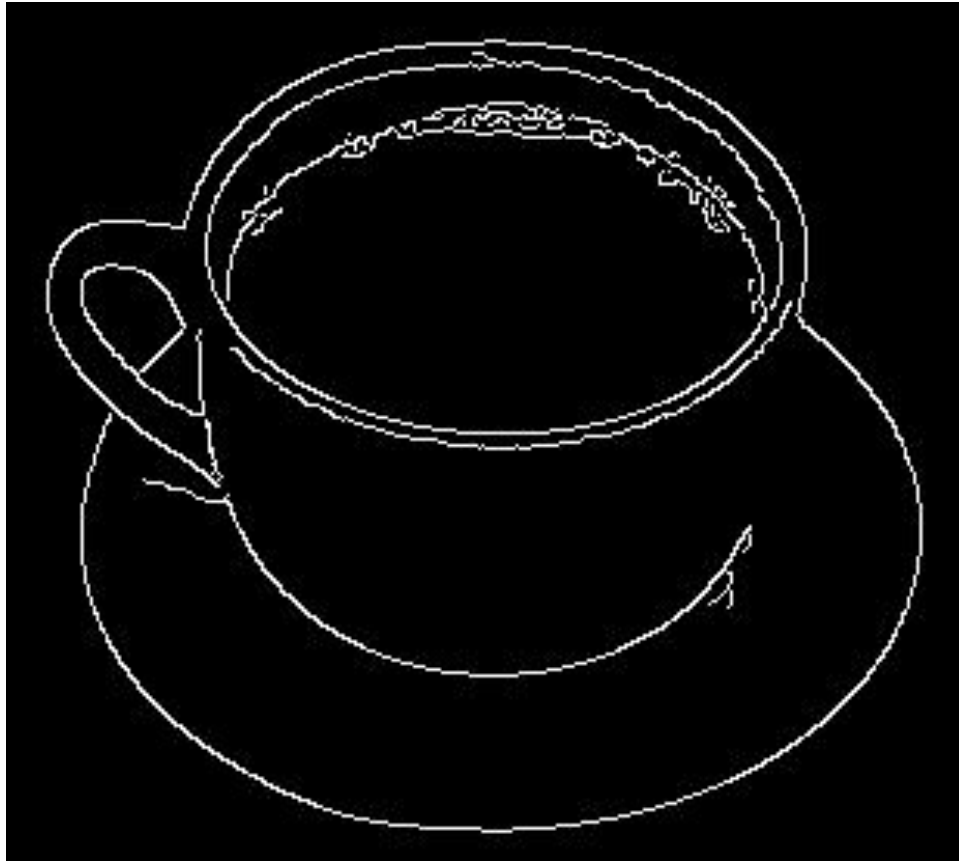
(orežeme na maximálne 255)

Sobelov operátor



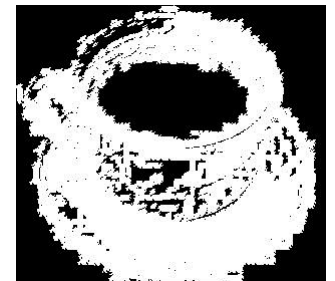
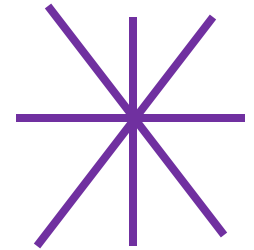
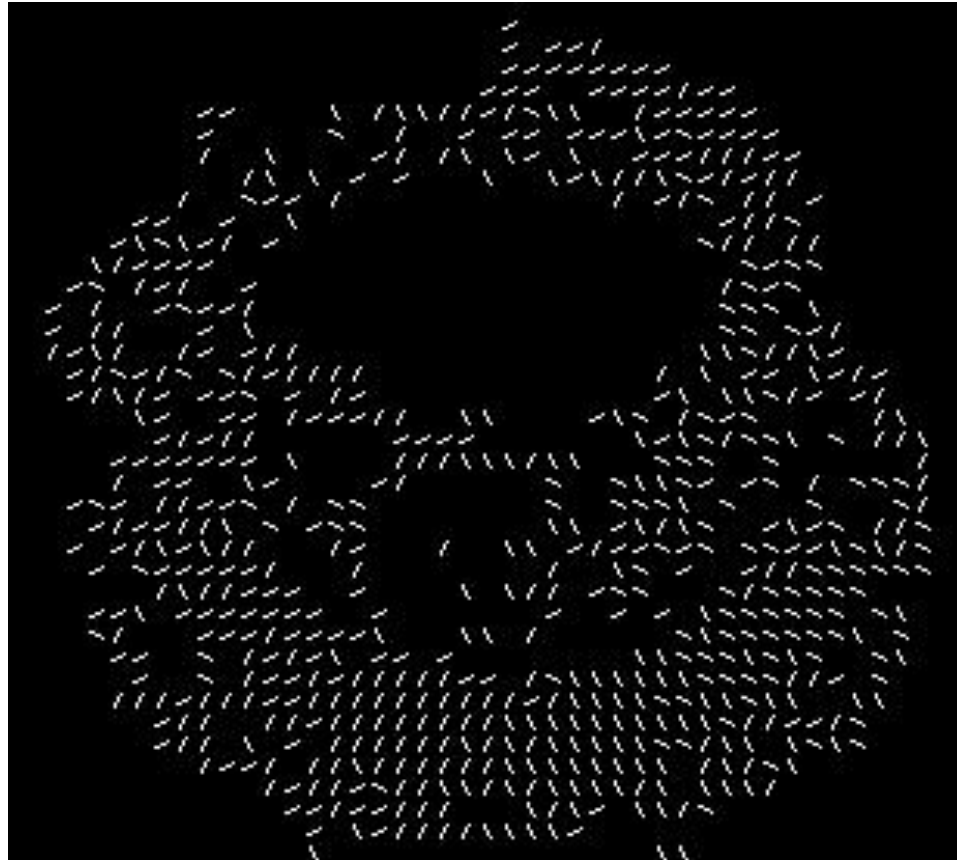
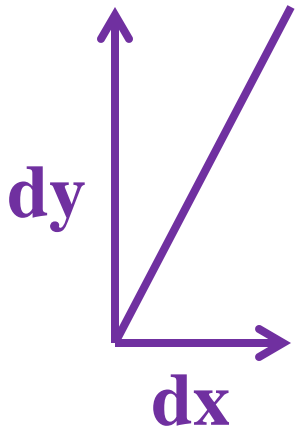
dx pekne zvýrazňuje vertikálne hrany a dy horizontálne

Hrany



dx a dy môžeme zlúčiť a pomocou vhodného prahu premeniť obraz hrán na binárny. Stenčovaním čiar dostaneme výsledné hrany

Orientácie



z dx a dy môžeme určiť orientácie, v každom bode smer v ktorom sa mení jas, pričom prechody z tmavého do svetlého a opačne považujeme za rovnaké

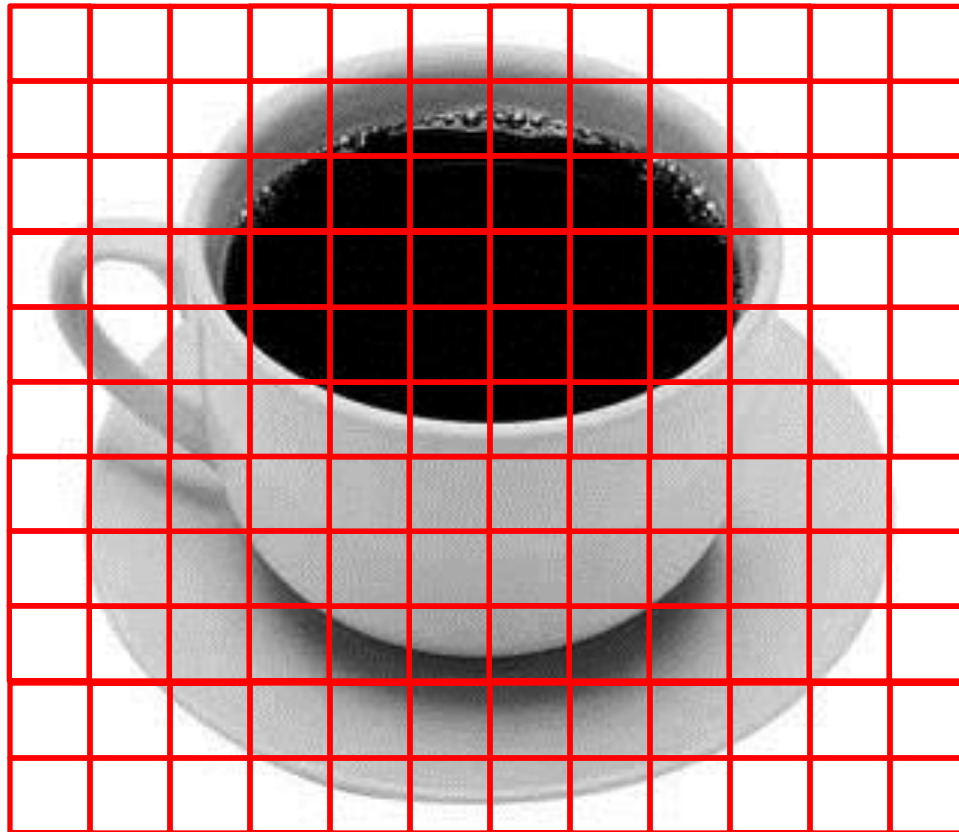
Šablóna

- Práve orientácie použijeme na zostrojenie šablóny rozpoznávaného objektu



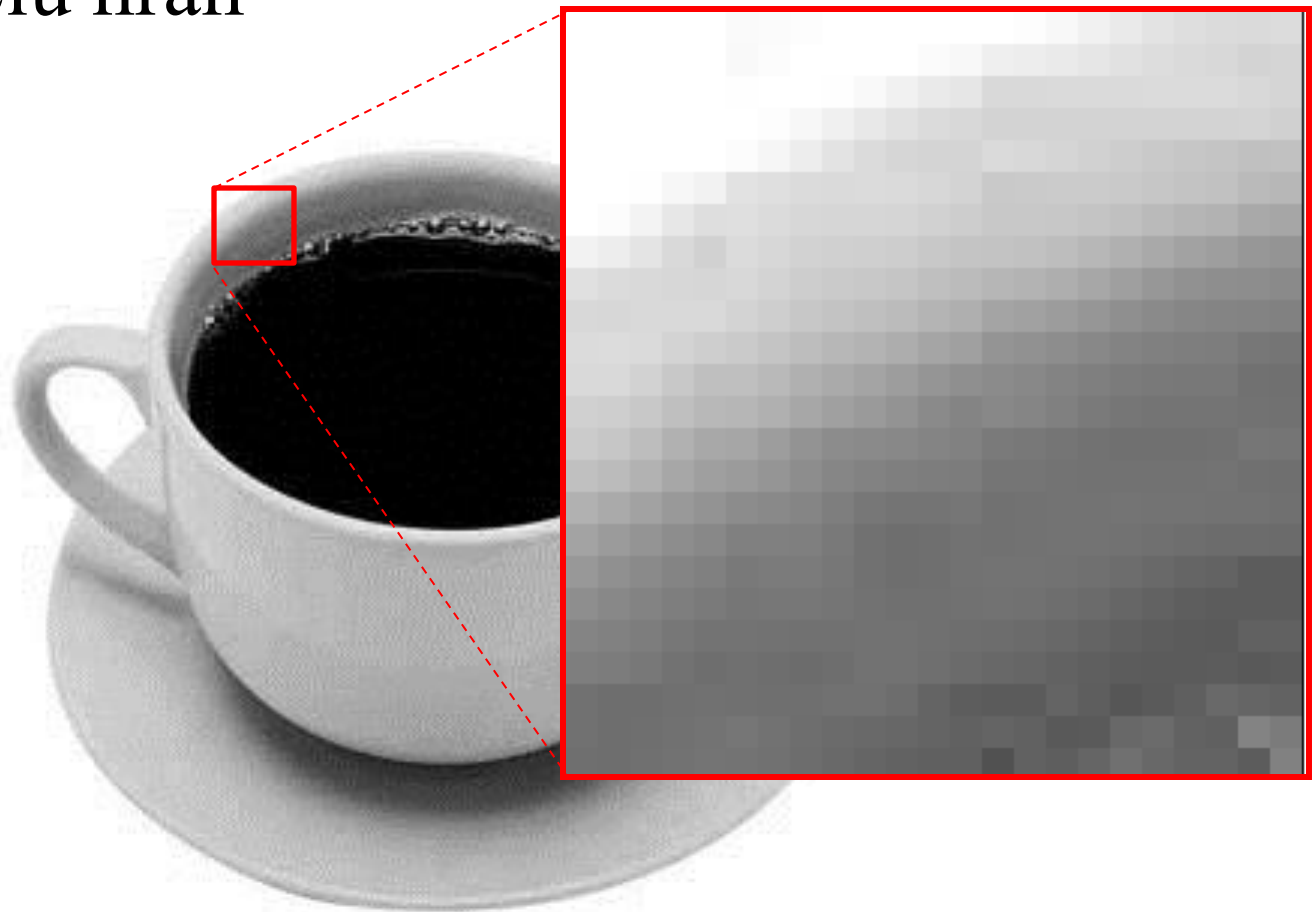
Šablóna

- objekt pokryjeme neprekrývajúcimi sa regiónmi



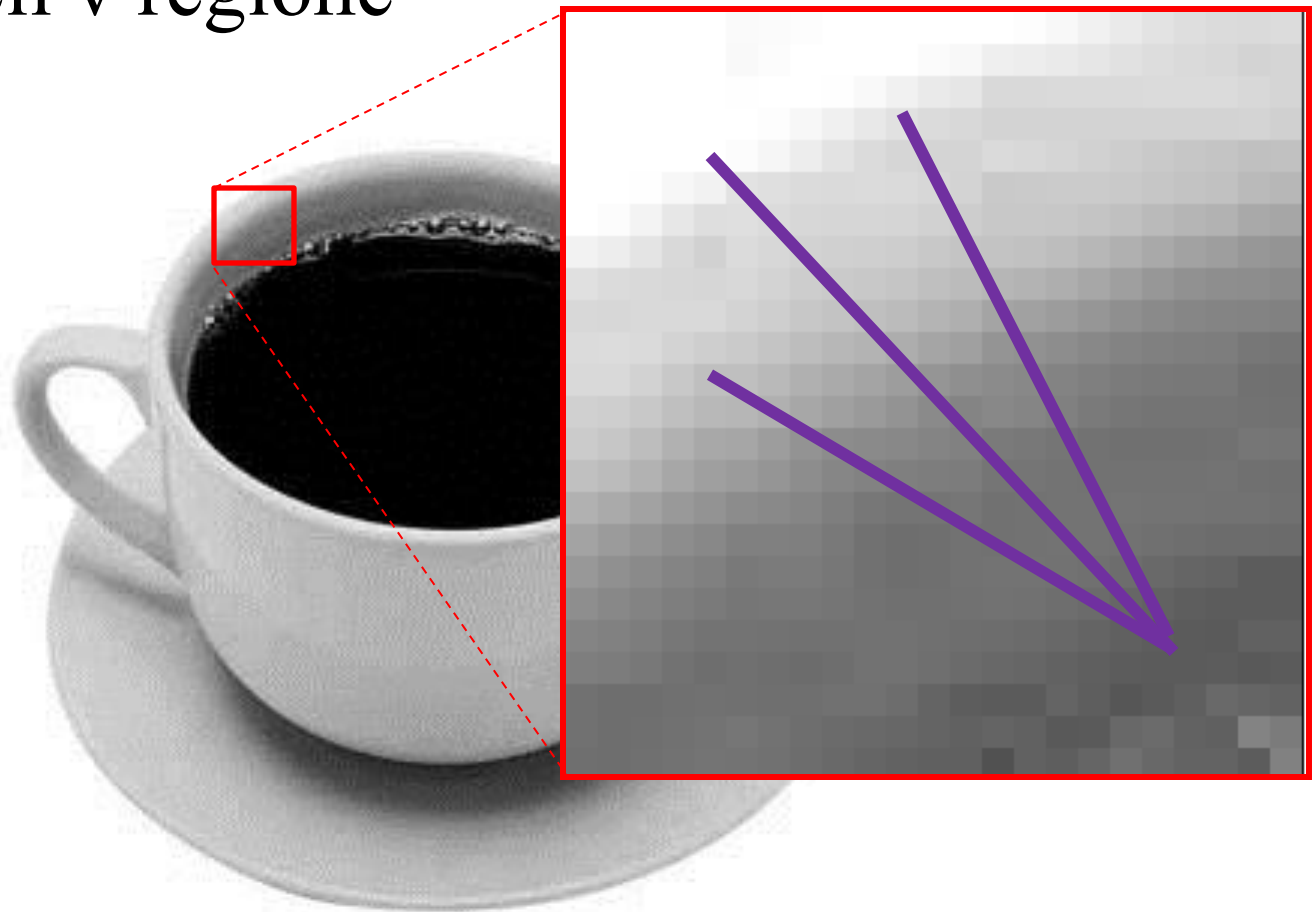
Šablóna

- Ku každému pixelu v regióne vieme orientáciu hrán



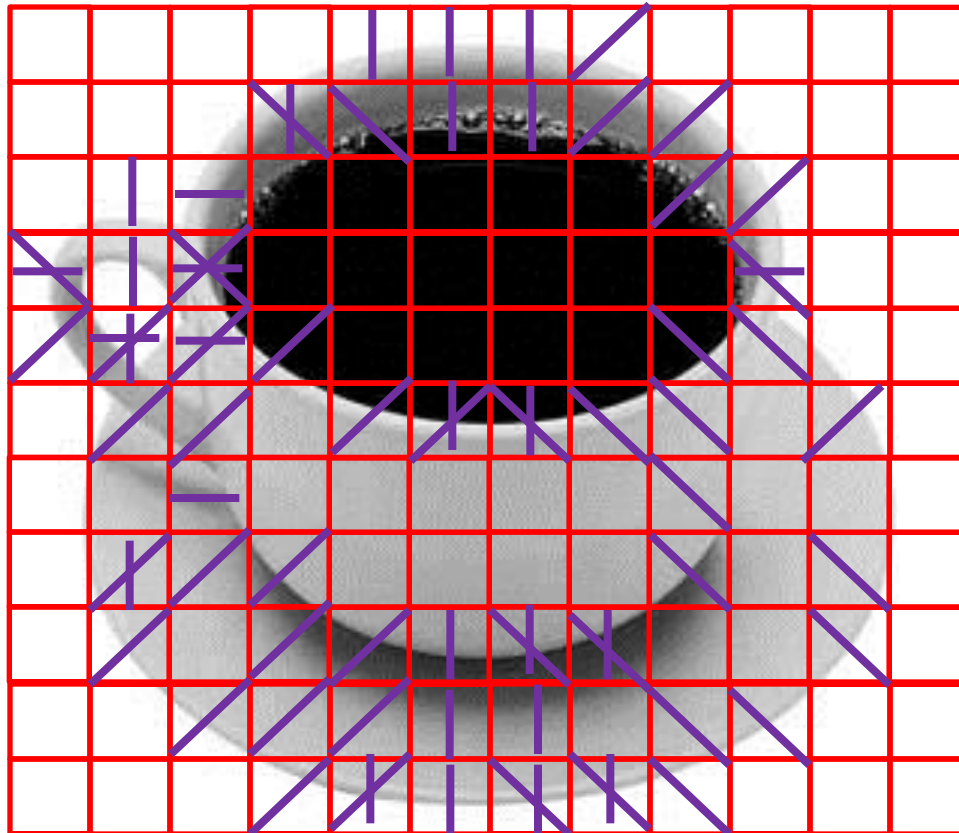
Šablóna

- Určíme z nich sadu prevládajúcich orientácií v regióne



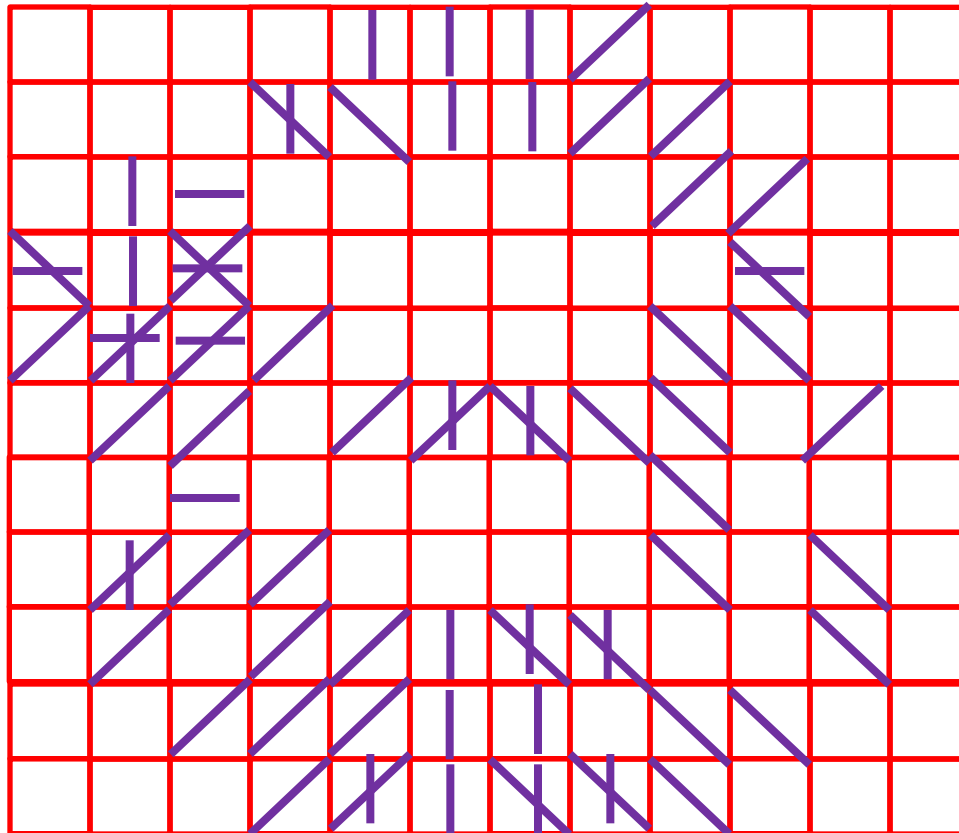
Šablóna

- Pre každý štvorček tak dostaneme sadu orientácií



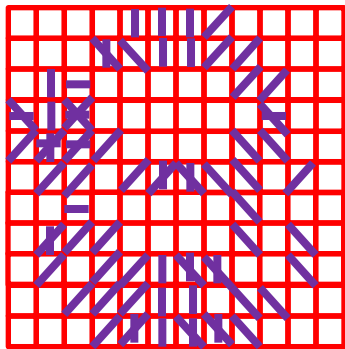
Šablóna

- A sada týchto orientácií bude tvoriť šablónu, tj. reprezentovať objekt

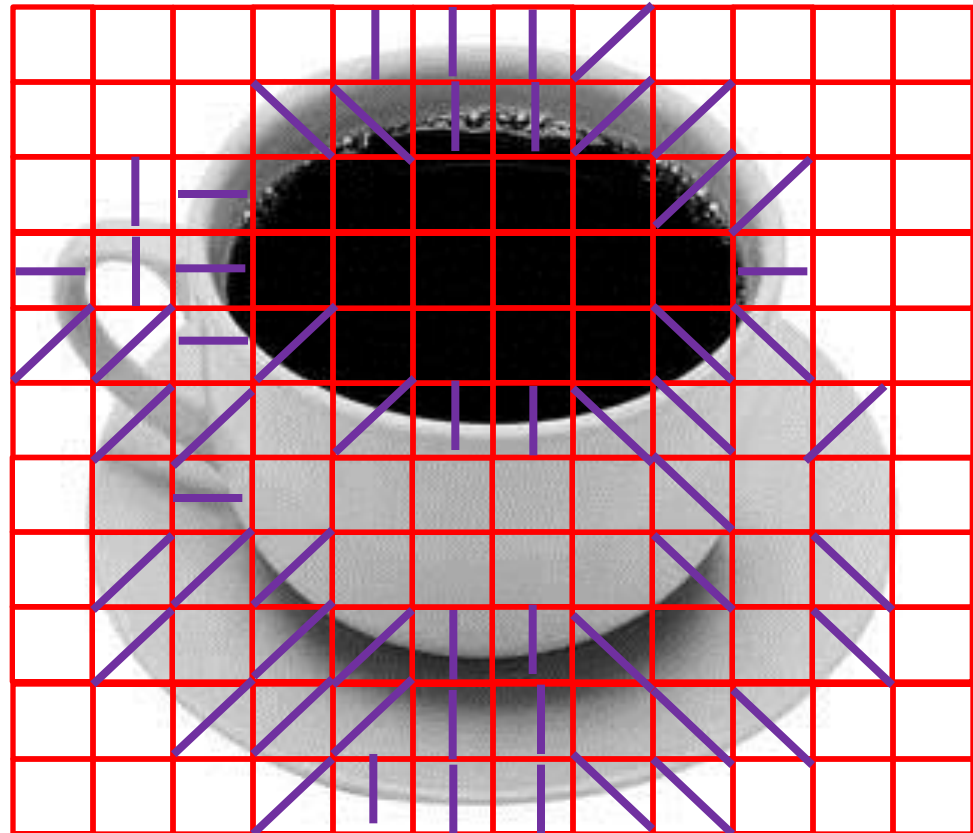


Hľadanie podľa šablóny

- Obrázok rozdelíme na regióny a v každom určíme jedinú, tzv. dominantnú orientáciu



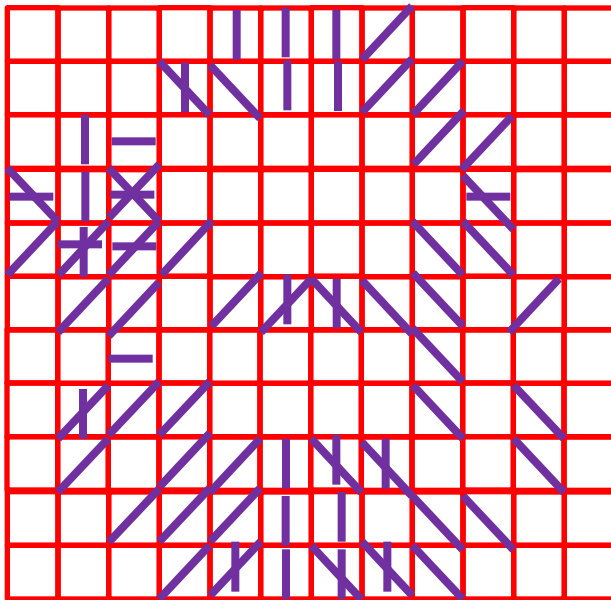
šablóna



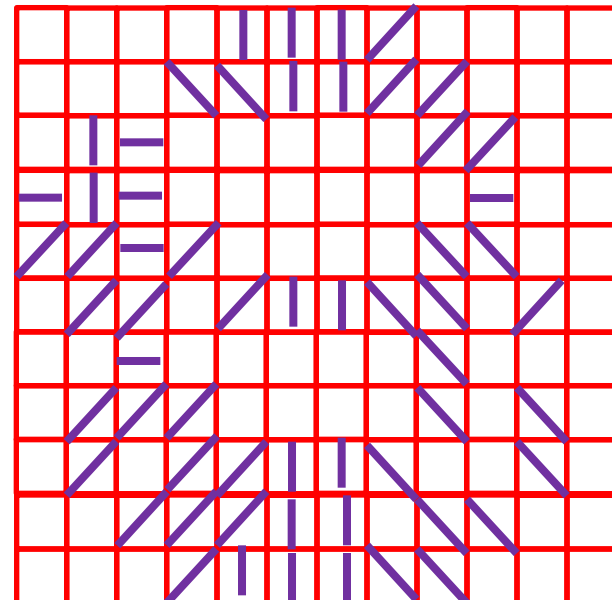
obraz

Hľadanie podľa šablóny

- Ak sa pre väčšinu regiónov nachádza dominantná orientácia v šablóne, tak OK.



šablóna

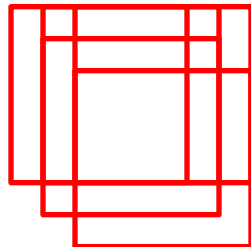
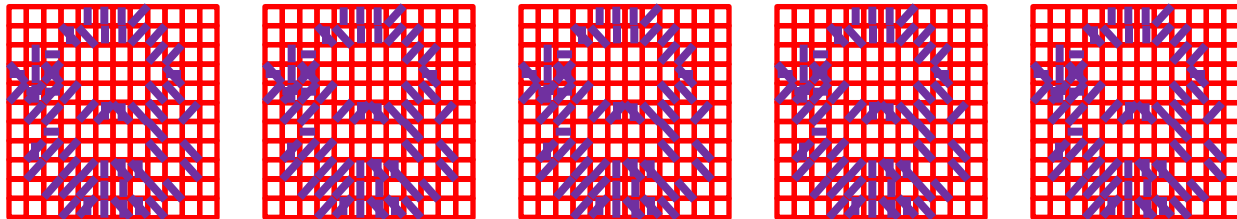


obraz

Nefunguje to ?

- Nemáme záruku, že sme trafili región na región. Nevadí, narobíme si veľa šablón posunutých o pár pixlov

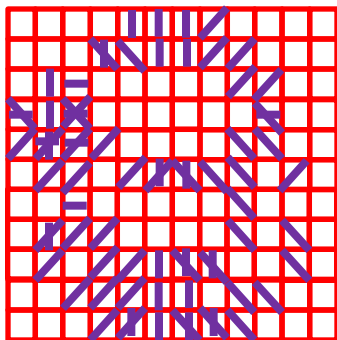
šablóny



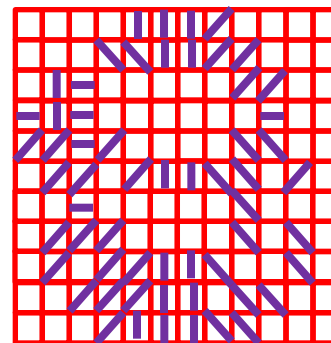
Zefektívnenie (pri strate presnosti)

- Môžeme potom tieto prekrývajúce sa šablóny zosumarizovať do jednej tak, že v regióne šablóny budú všetky orientácie, y okolitých regiónov. Takú šablónu nie je potrebné posúvať po pixeloch, hoci už tak presne nevystihuje reprezentovaný objekt

šablóna



obraz



Pohl'ad z rôznych strán

- Jedna šablóna nám umožní rozpoznať objekt len z určitého pohľadu
- Preto je nevyhnutné aby sme objekt reprezentovali množstvom šablón urobených z rôznych pohľadov
- Stačí kamerou objekt obísť a snímať šablóny. Z nich program vyberá dostatočne od seba odlišné, aby ich nebolo moc a mali dobrú výpovednú hodnotu

Bitová reprezentácia

- Prítomnosť orientácii v regióne šablóny možno reprezentovať bitovo a celé priloženie šablóny vypočítat' pomocou bitových operácií, ktoré sú veľmi efektívne
- DOT je preto veľmi rýchla a funguje v reálnom čase

Hranica objektu

- DOT umožňuje nájsť aj približnú hranicu objektu. Tvoria ju tie hrany, ktorých orientácie sme našli v šablóne.

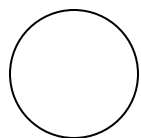


Ako získať šablónu?

- Šablónu nie je možné vytvoriť priamo z objektu, ktorý sa nachádza v scéne.
- Musíme ho nasnímať voči kontrastnému pozadiu
- Alebo môžeme použiť nejakú metódu separácie objektu z obrazu, napríklad na základe pohybu objektu



Omylnosť rozpoznania (DOT)



VZOR



Omyl či kreativita ?

Ďakujem za pozornosť !

Andrej Lúčný

Katedra aplikovanej informatiky

FMFI UK Bratislava

andy@microstep-mis.com

www.microstep-mis.com/~andy