

*Educafe*

**Vidia roboty tak, ako my, alebo  
len šikovne podvádžajú?**

**Andrej Lúčny**

**Katedra aplikovanej informatiky**

**Fakulta matematiky, fyziky a informatiky**

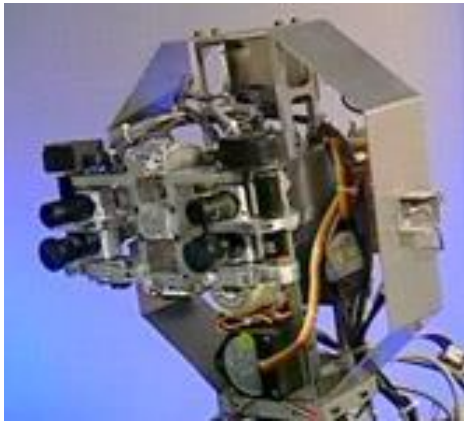
**Univerzita Komenského v Bratislave**

**andy@microstep-mis.com**

**www.microstep-mis.com/~andy**

# Mobilné roboty, ktoré vidia

Mobilný robot sleduje scénu a má v nej  
rozpoznať určitý objekt

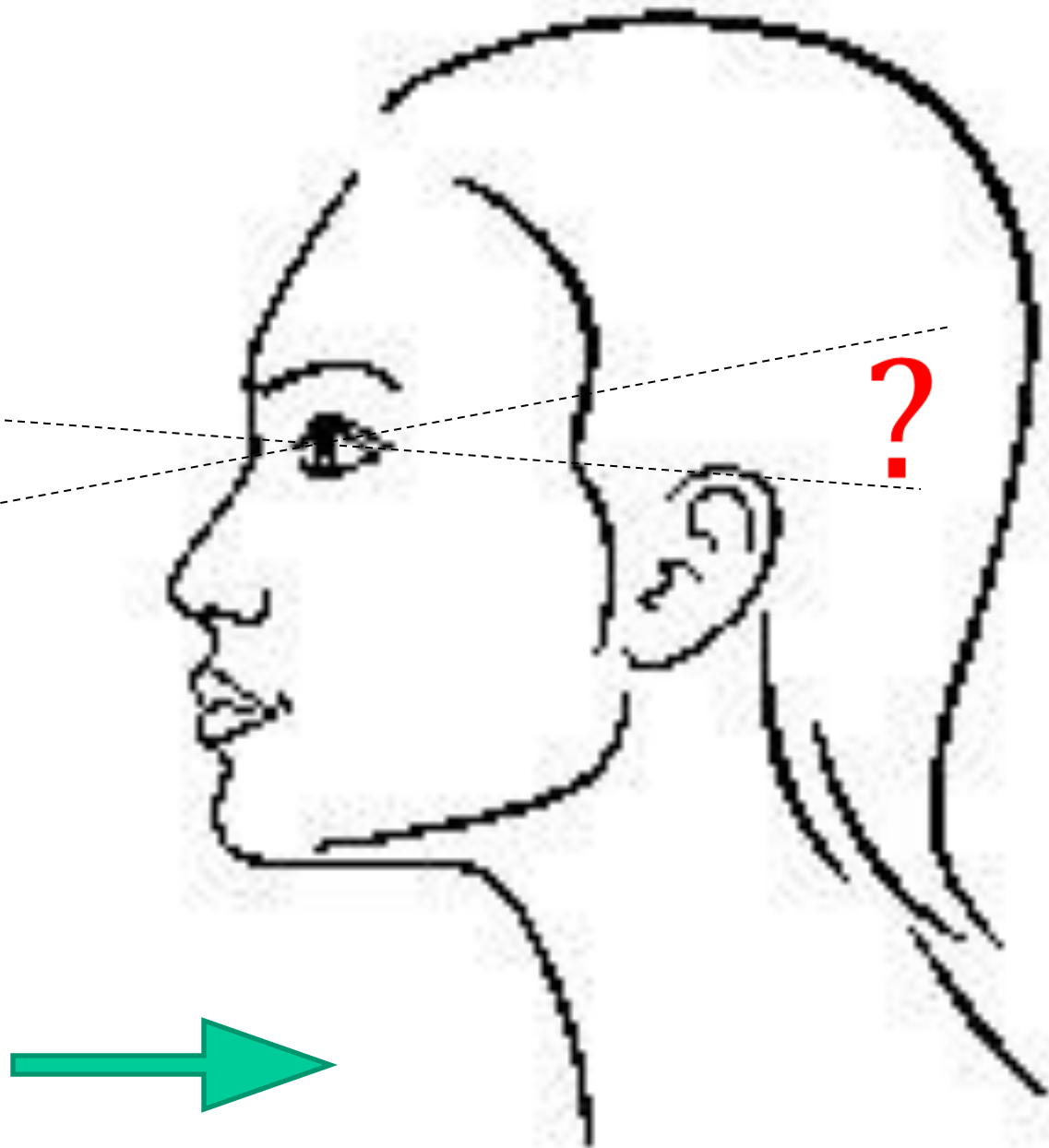


COG 1993

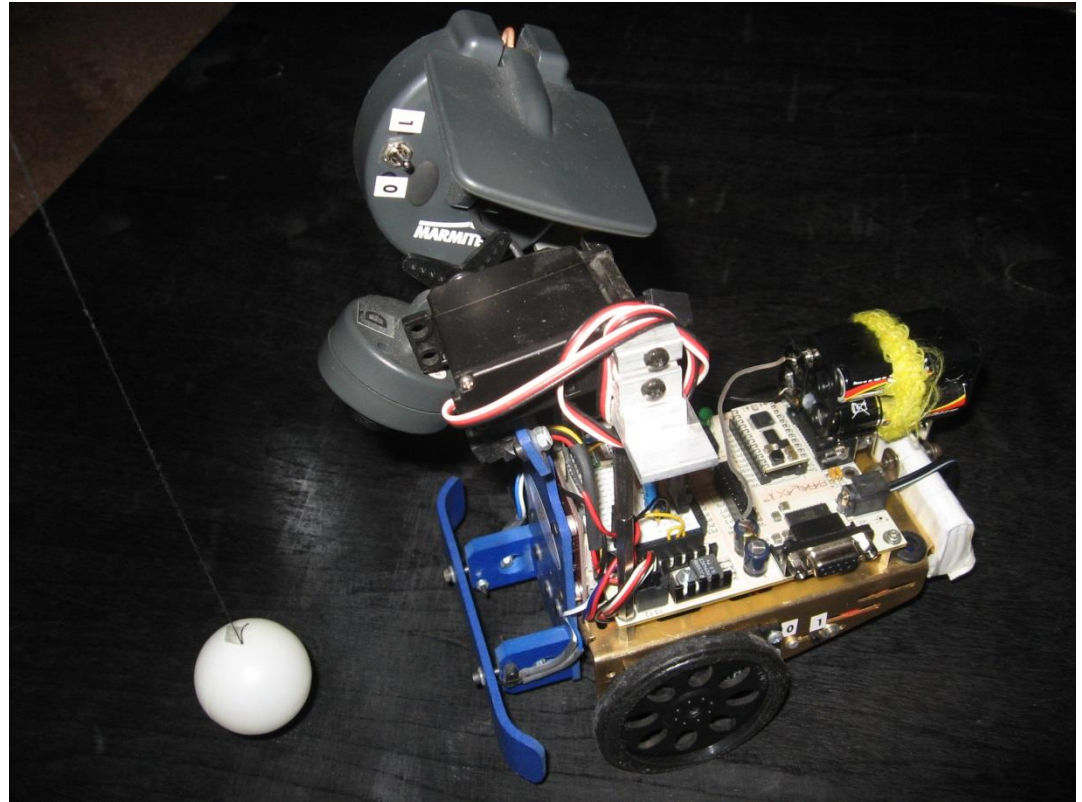


iCub 2007

Čo to  
znamená  
vidieť ?



Vidí tento  
robot  
loptičku ?

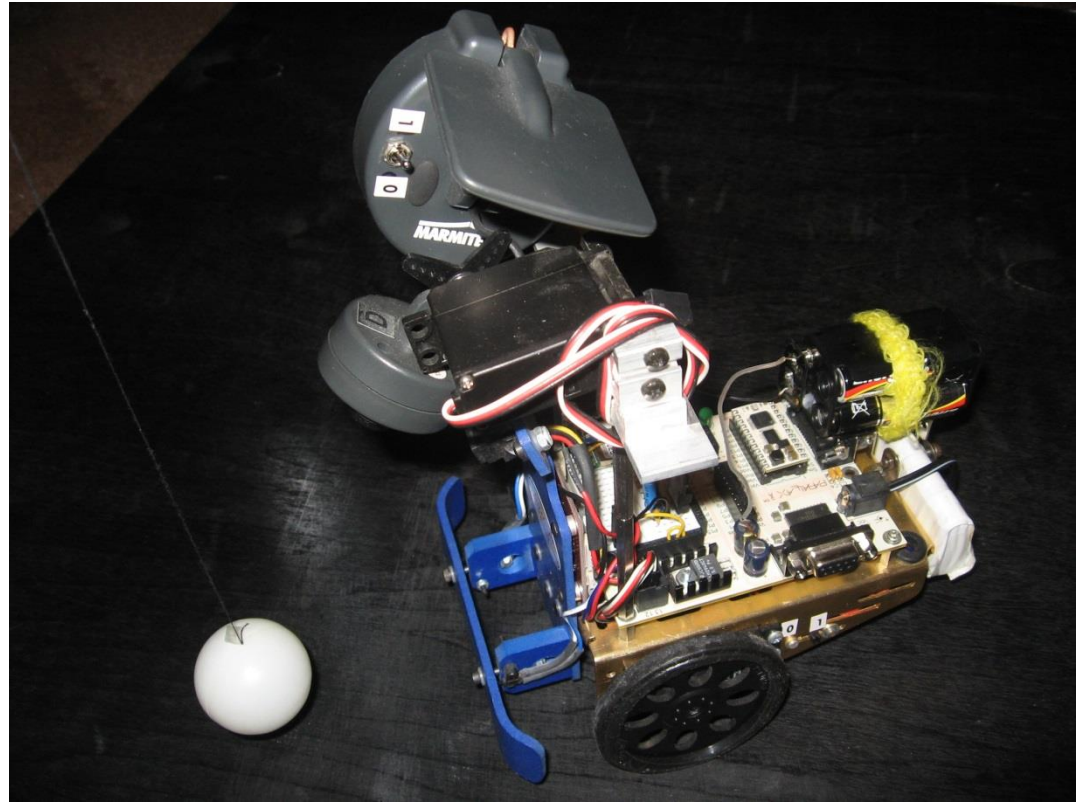


V zmysle Turingovho testu áno!

vstup: *obraz z kamery*

výstup: *správnne pohyby motorov*

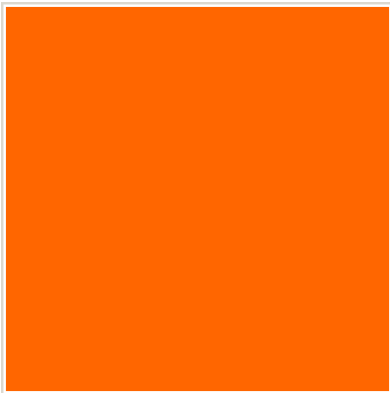
Vidí ju ale  
tak ako ju  
my vidíme?



Žiaľ, my sami nie celkom presne  
vieme ako vidíme!

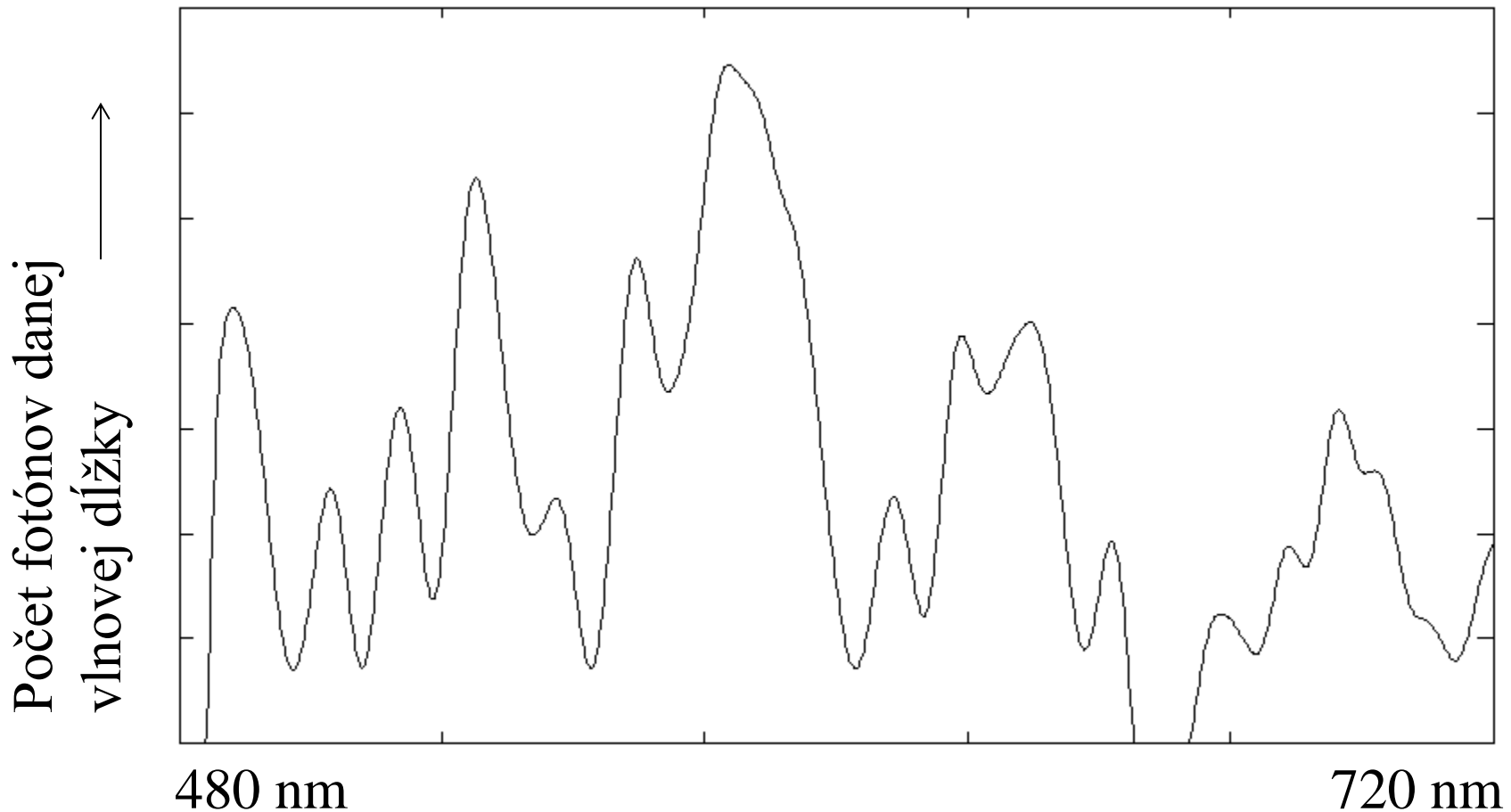
Vieme ale, ako vidí robot

# Ako robot vidí farby ?

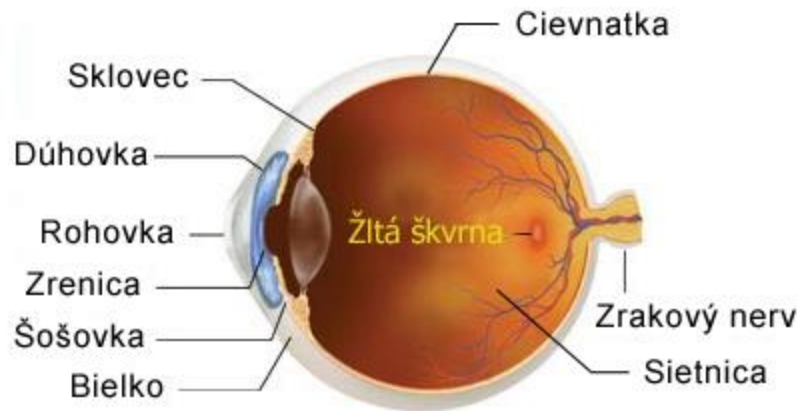


Farba je štatistická vlastnosť svetla ktoré dopadá na bod čipu kamery či sietnice oka

V každom okamihu dopadá na každý bod sietnice množstvo fotónov rôznej vlnovej dĺžky. Meranie na jednom bode sietnice za čas  $\Delta t$  dá celú distribúciu početnosti fotónov určitej vlnovej dĺžky



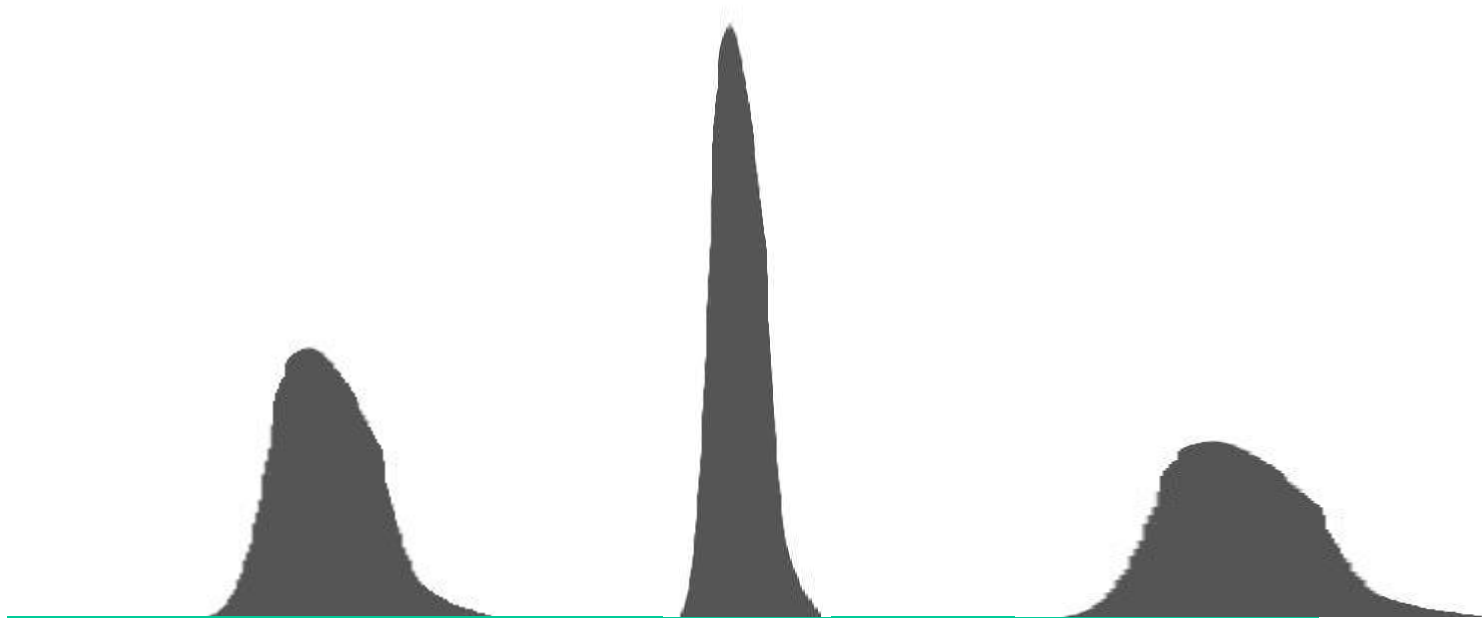
# Tri farebné zložky stačia



Na sietnici oka má človek tri druhy čapíkov založené na troch druhoch proteinov citlivých na svetlo. Každý z nich je citlivý na celý rozsah vlnových dĺžok ale na niektoré viac a iné menej

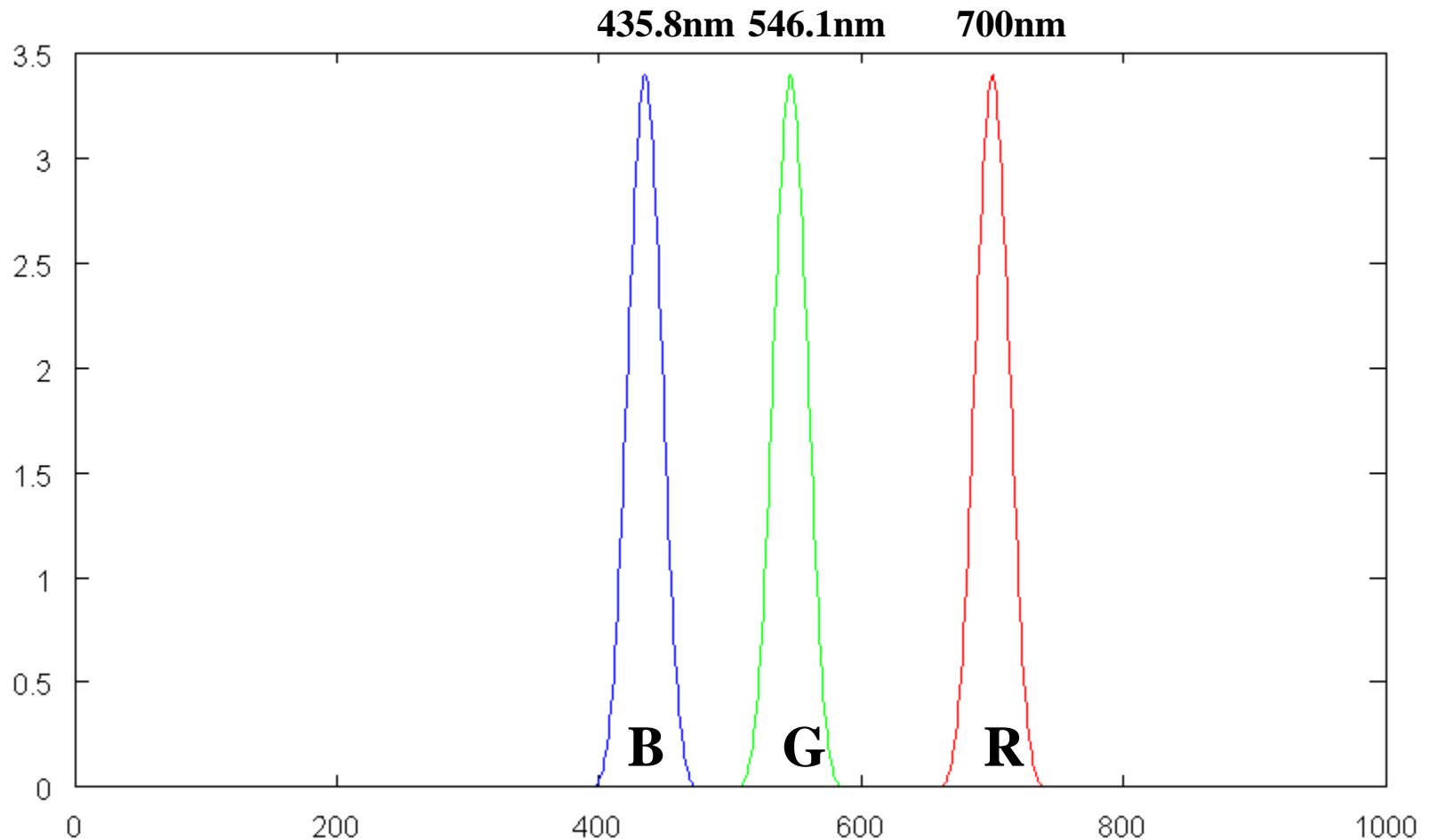


Signál, ktorý oko posiela do mozgu, je rovnaký pre veľké množstvo distribúcií vlnových dĺžok



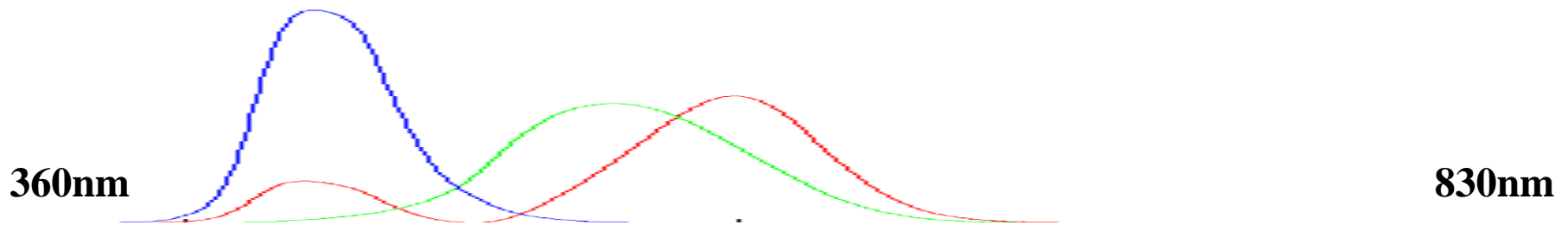
# RGB model

Vďaka tomu je možné dostatočne široko zvoliť tri monochromatické farby a ich miešaním vieme vyvolať v oku dojem (skoro) akékoľvek farby



# RGB model

Pomery R,G,B pre miešanie farieb dúhy

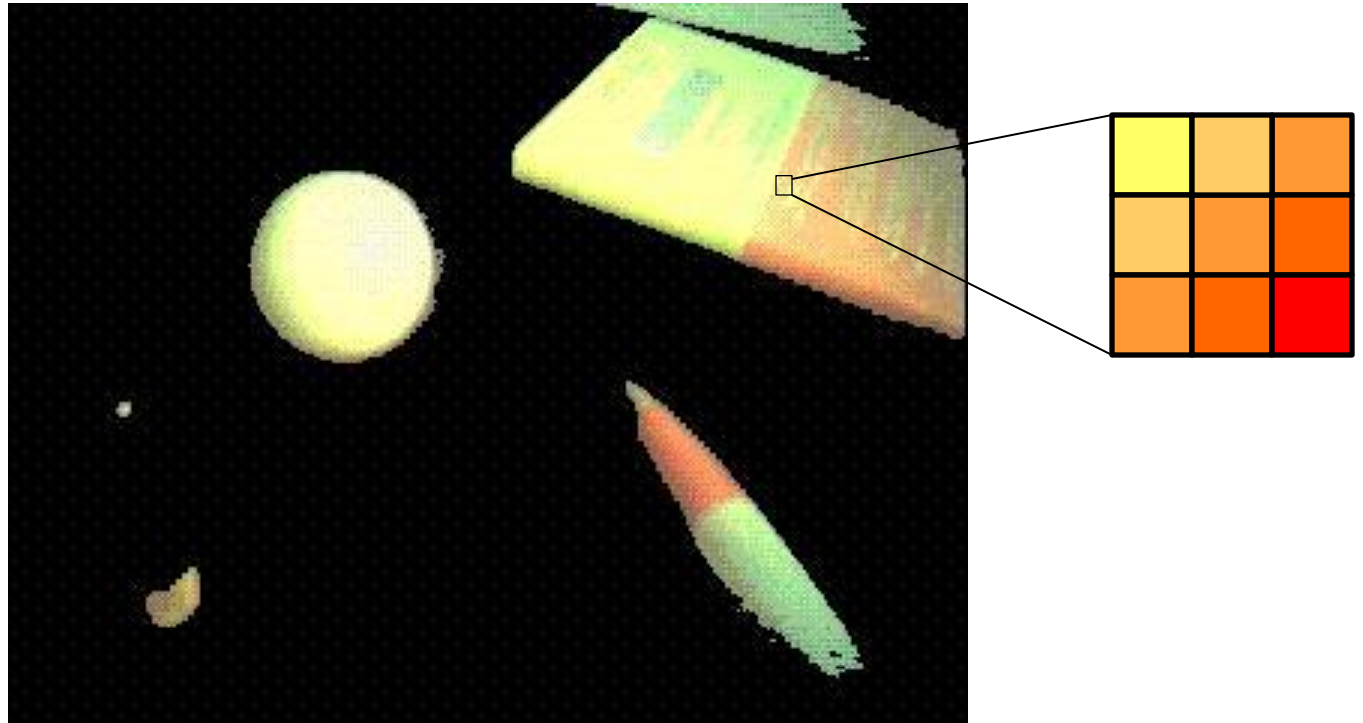


Ultrafialové  
žiarenie

svetlo

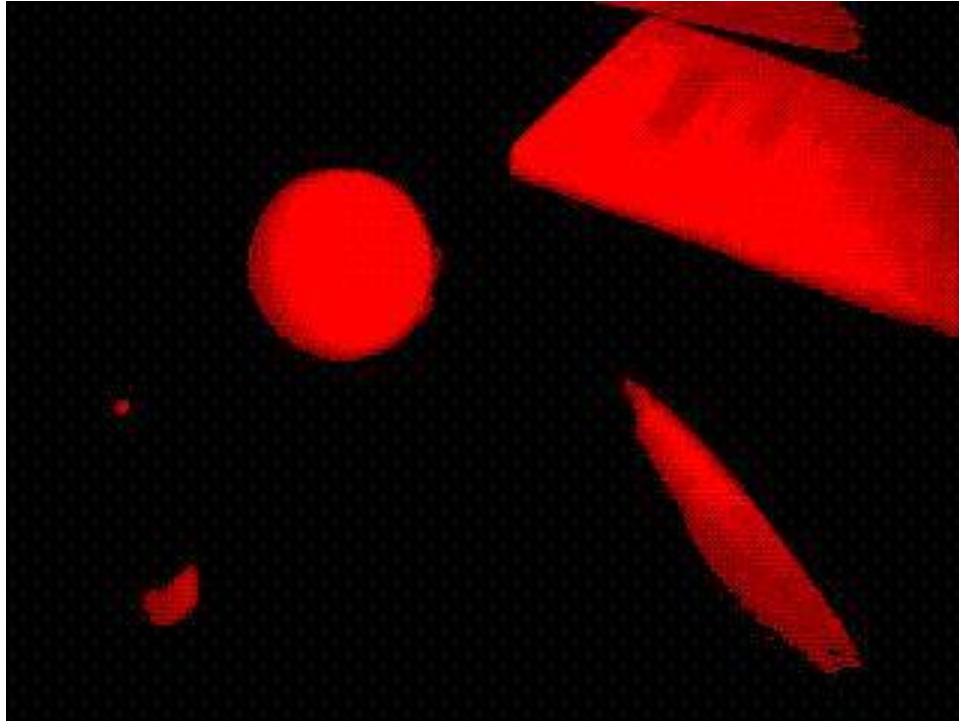
Infračervené  
žiarenie

# Vstup: obraz z kamery

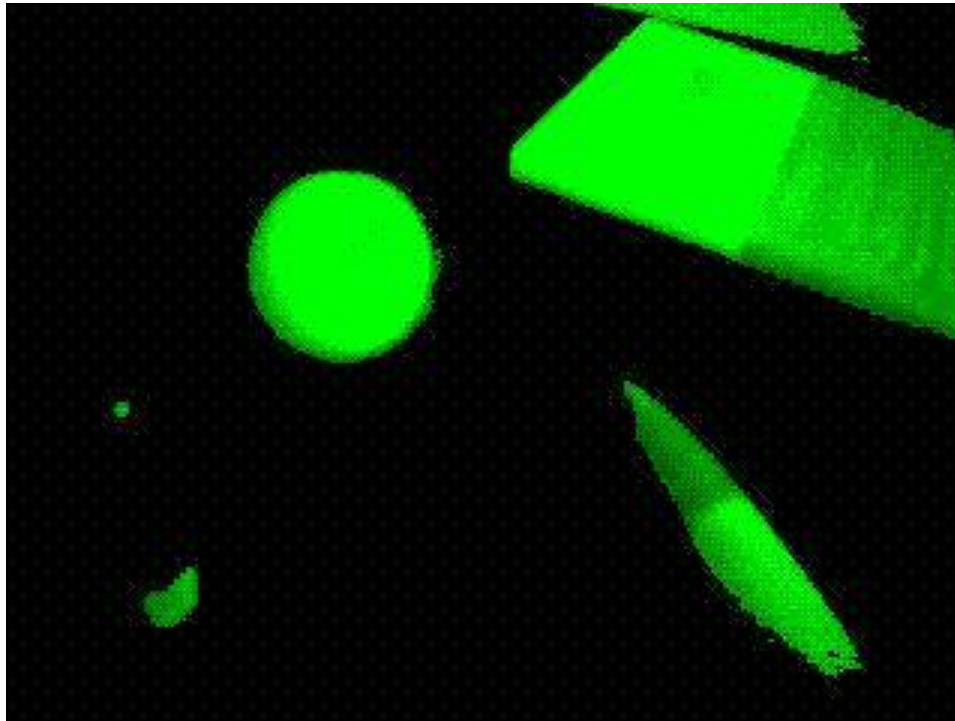


Tri polia  $r[h,w]$ ,  $g[h,w]$ ,  $b[h,w]$ , každý ich prvok je číslo  $0..255$  a predstavujú červenú, zelenú a modrú zložku farby

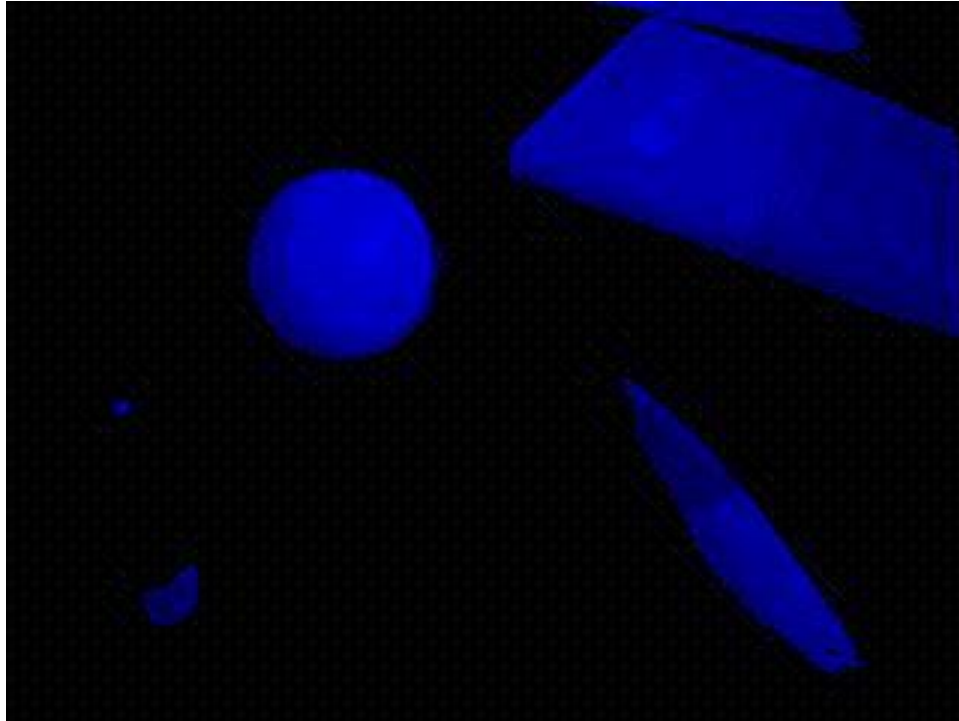
# Červená zložka



# Zelená zložka

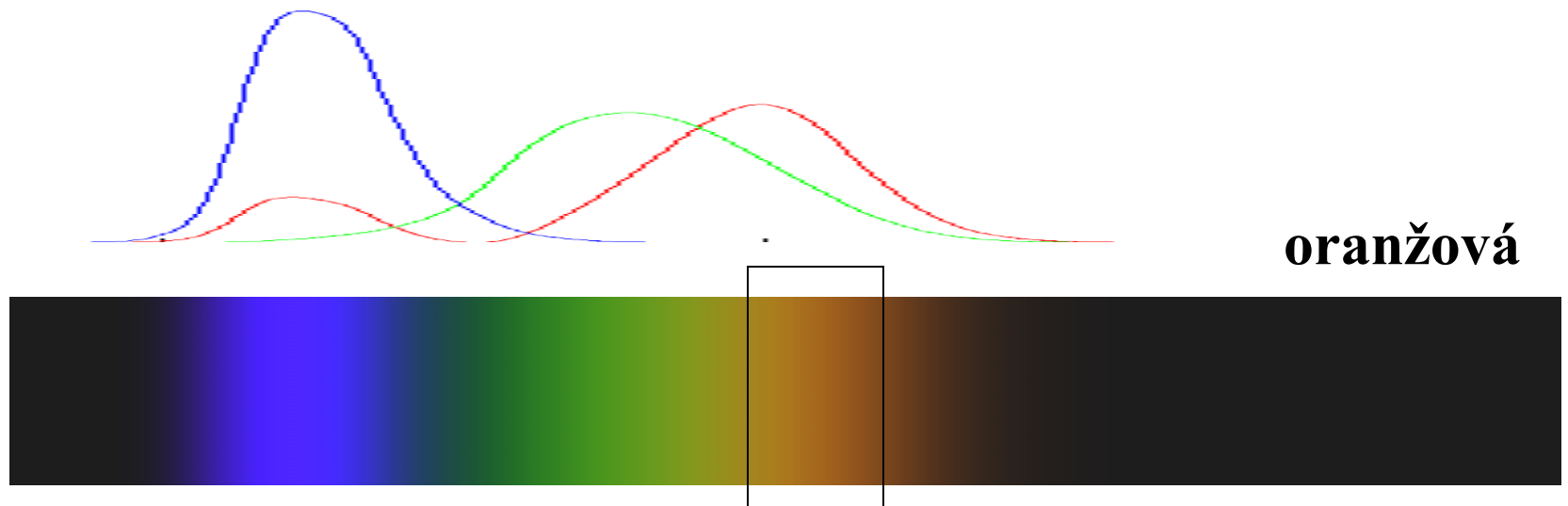


# Modrá zložka



# Rozpoznávanie farieb

- Na dúhe je vidno oranžovú farbu a vidíme:
  - neobsahuje modrú zložku
  - obsahuje proporcionálne viac červenej ako zelenej

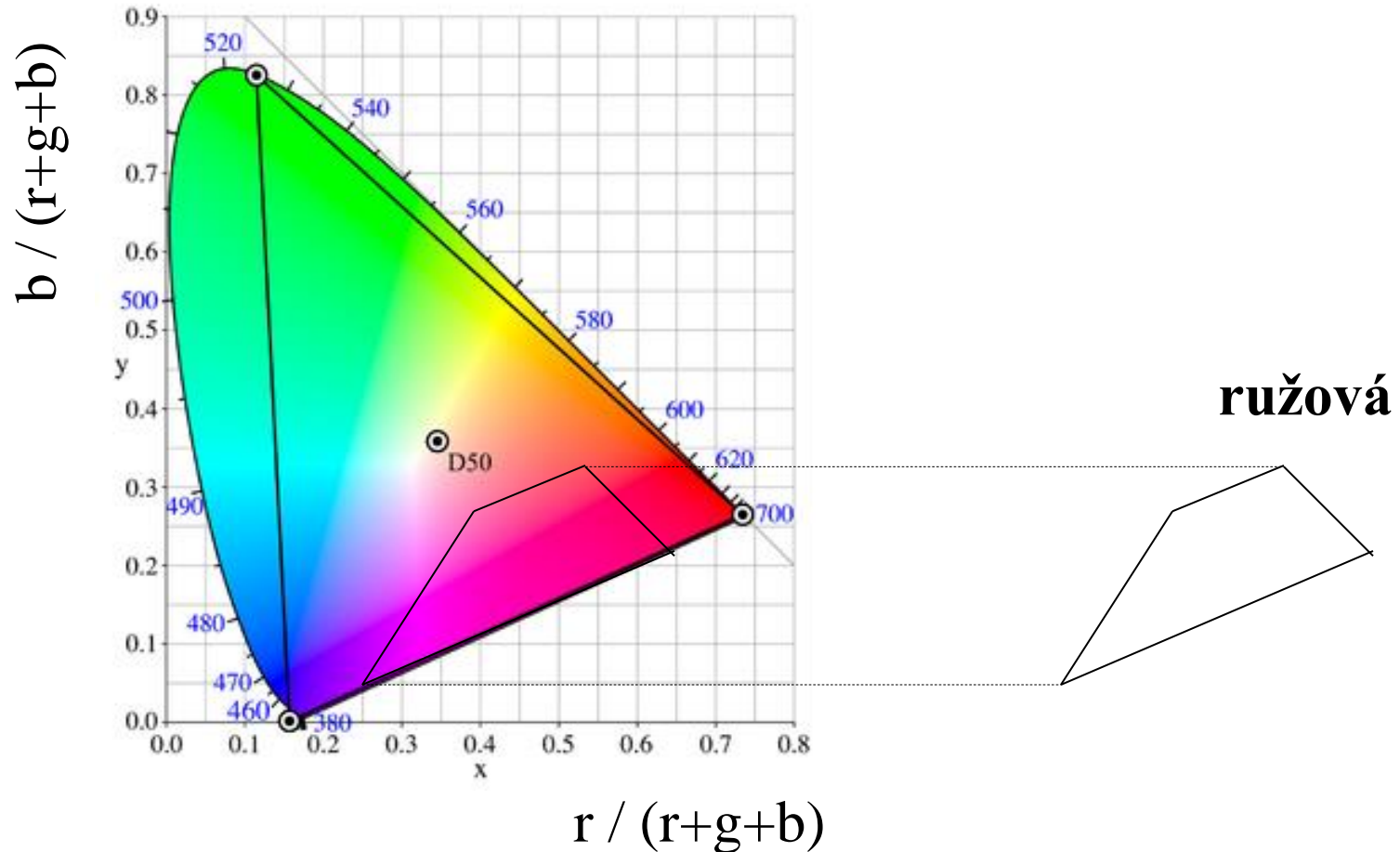


$$r > 50 \ \& \ r/(g+1) > 1.1 \ \& \ r/(g+1) < 3.3 \ \& \ b < 35$$



# Rozpoznávanie farieb

- Farby ktoré nie sú monochromatické



# Ako robot vidí 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  $r$

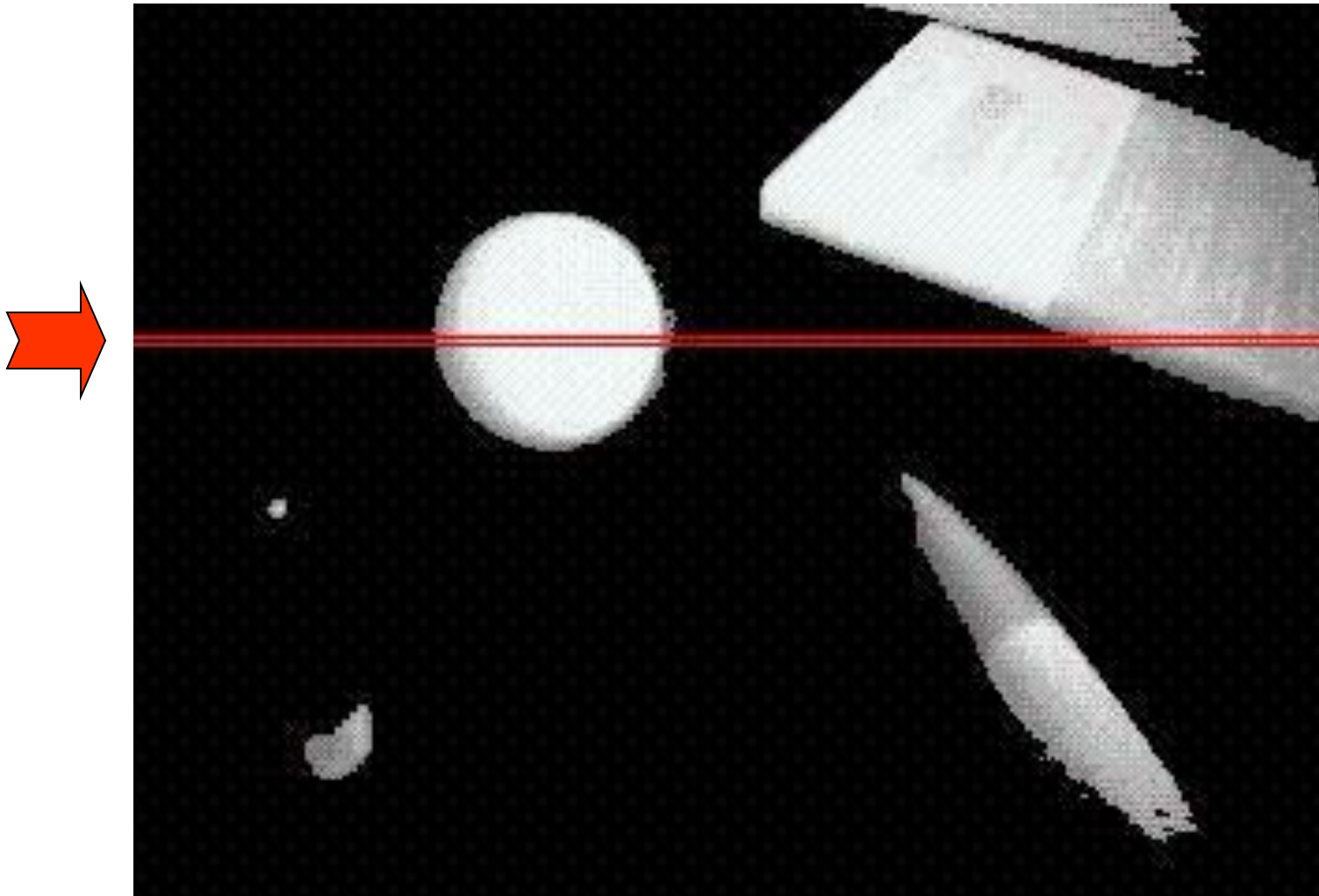
$$(x - c_x)^2 + (y - c_y)^2 = r^2$$

efektívny algoritmus na vykreslenie

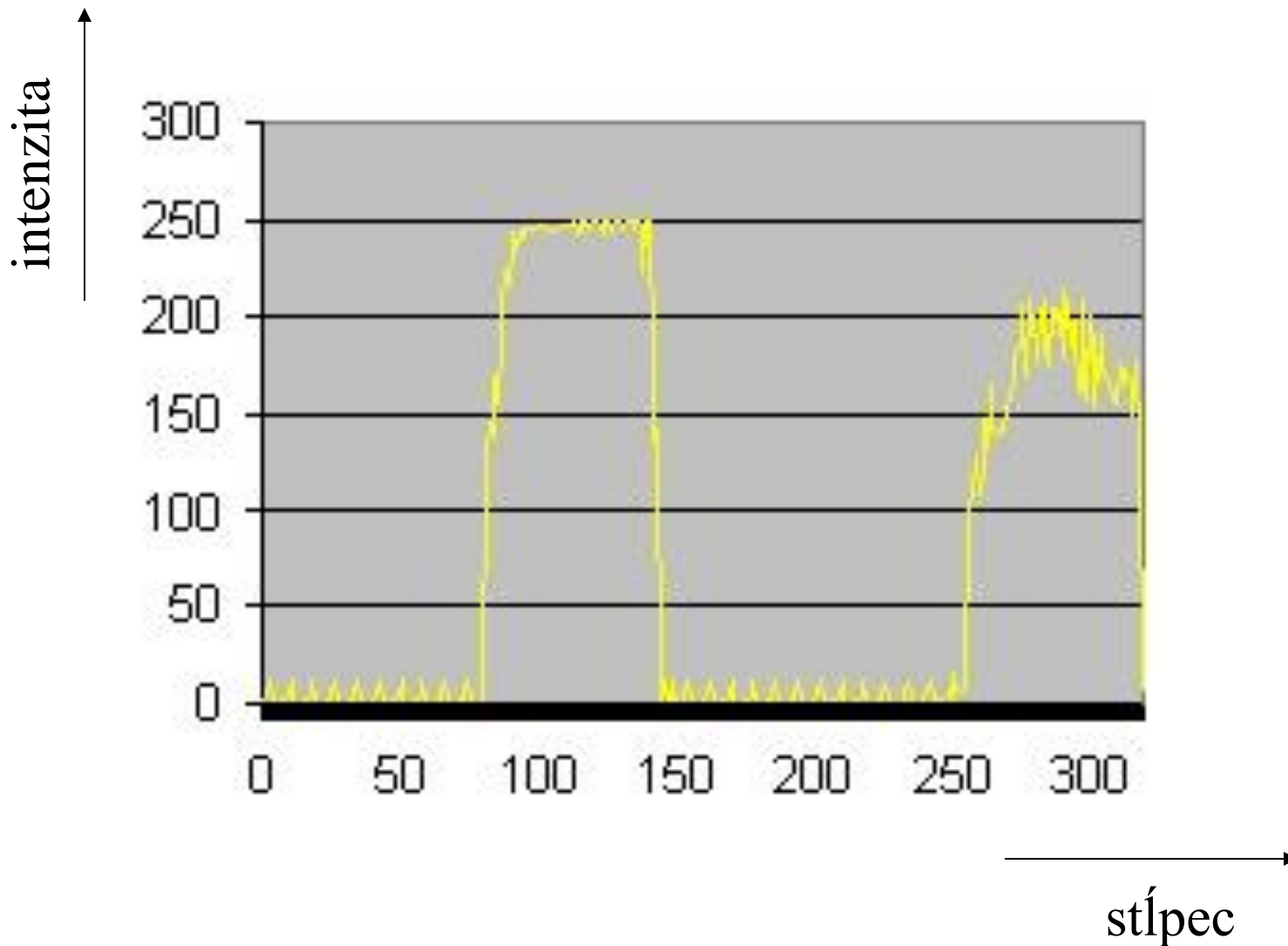
# Čiernobiely obraz



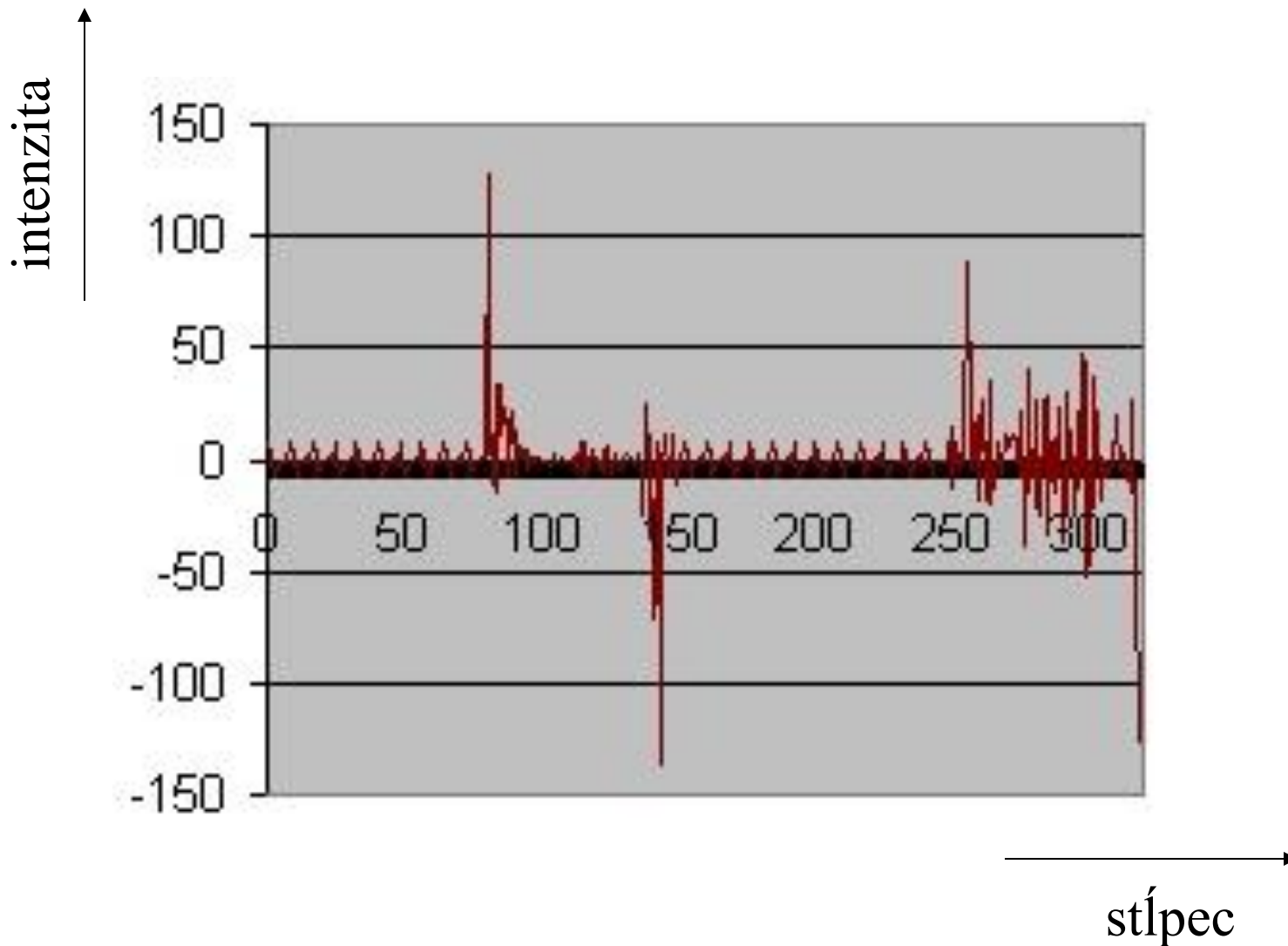
Pole  $bw[h,w]$ , každý jeho prvok je číslo  $0..255$  a predstavuje intenzitu svetla



- Jeden riadok poľa bw si teraz môžeme znázorniť ako funkciu intenzity od čísla stĺpca

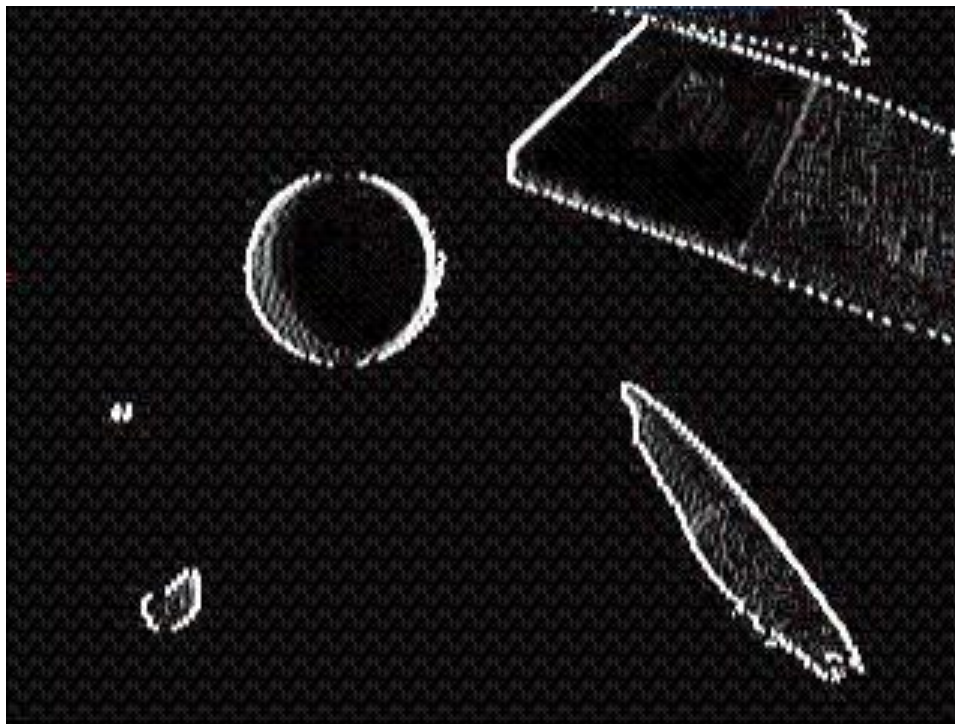


- Hranám zodpovedajú strmé úseky.  
Ako ich odfiltrovať od zvyšku?



- Hrany vyskočia, keď urobíme rozdiel dvoch po sebe idúcich hodnôt

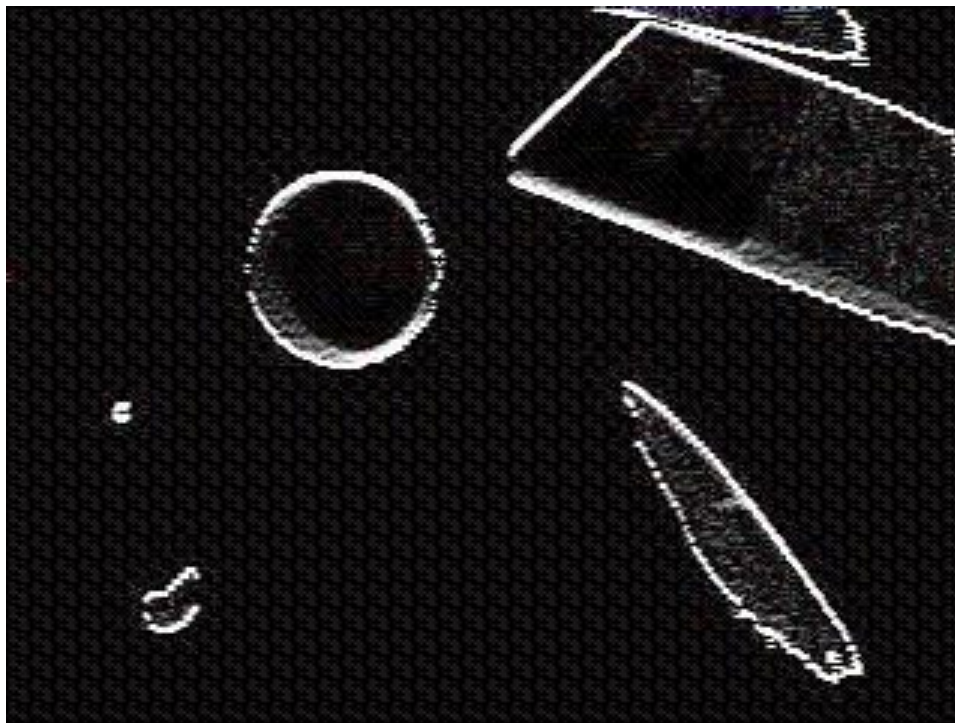
# Sobelov operátor - vodorovne



Máme pekne zvýraznené vertikálne hrany, ale horizontálne sú slabé

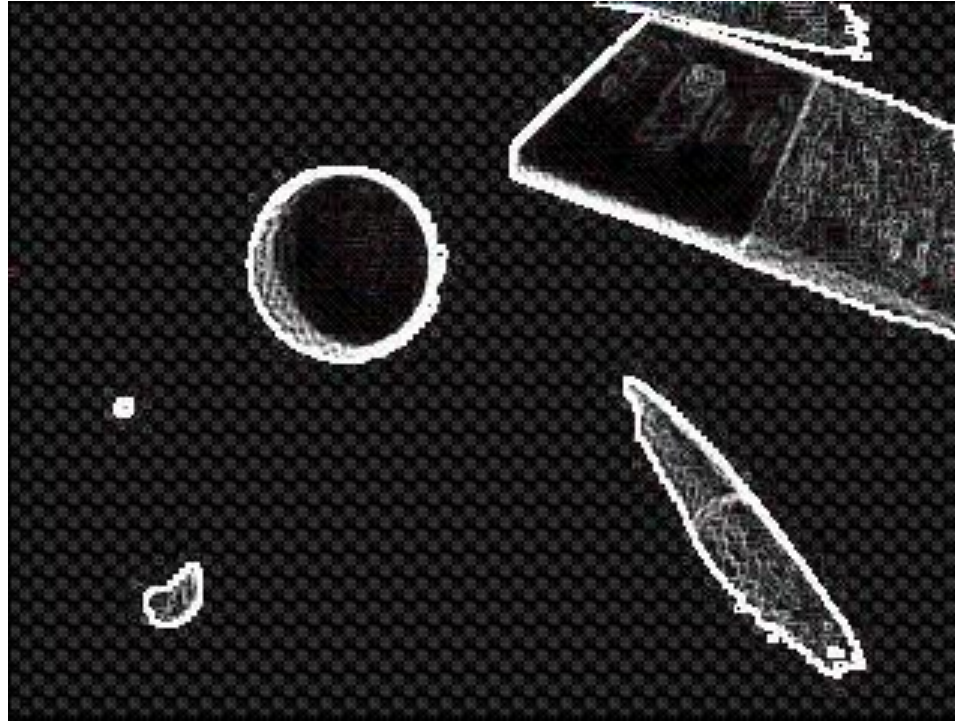


# Sobelov operátor – zvisle



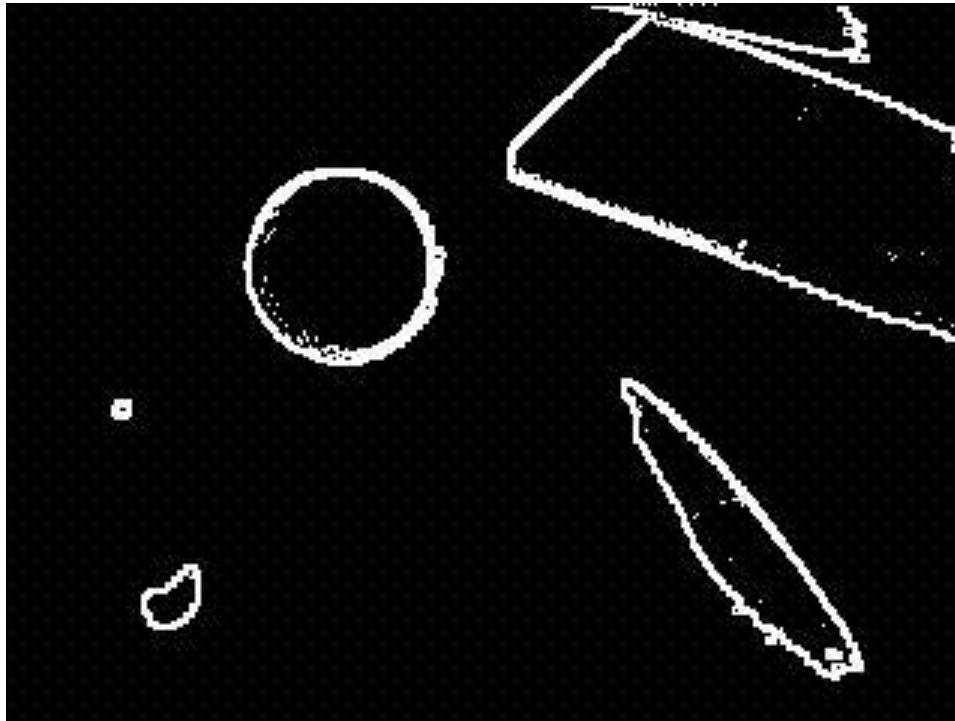
Z toho máme zase zvýraznené horizontálne hrany

# Sobelov operátor



a nakoniec to sčítame. Teraz sú hrany krásne, ale stále tam máme veľa bodiek čo do hrán nepatria

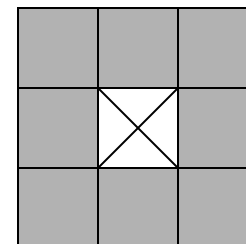
# Prahovanie dá binárny obraz



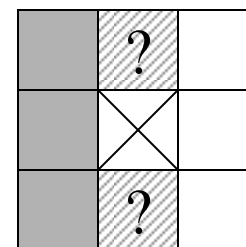
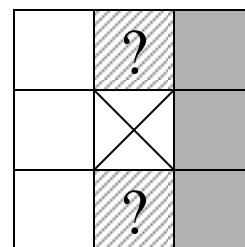
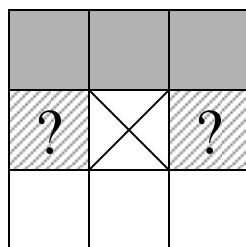
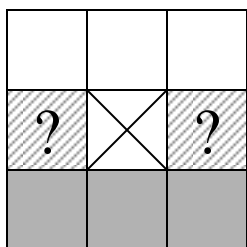
- Radi by sme doterajších 0-255 zmenili na 0/1 a mali len hrany. Dobré výsledky tu dáva keď zvolíme určitý prah, napríklad 200

# Stenčovanie

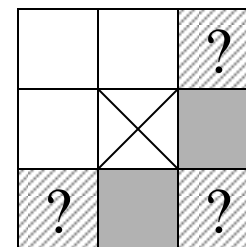
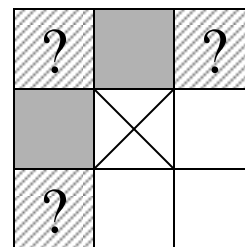
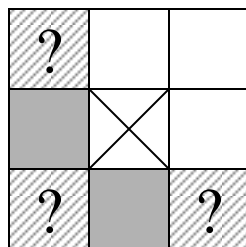
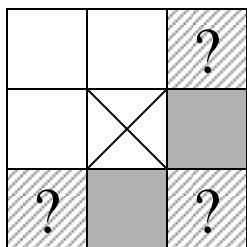
- izolované body:



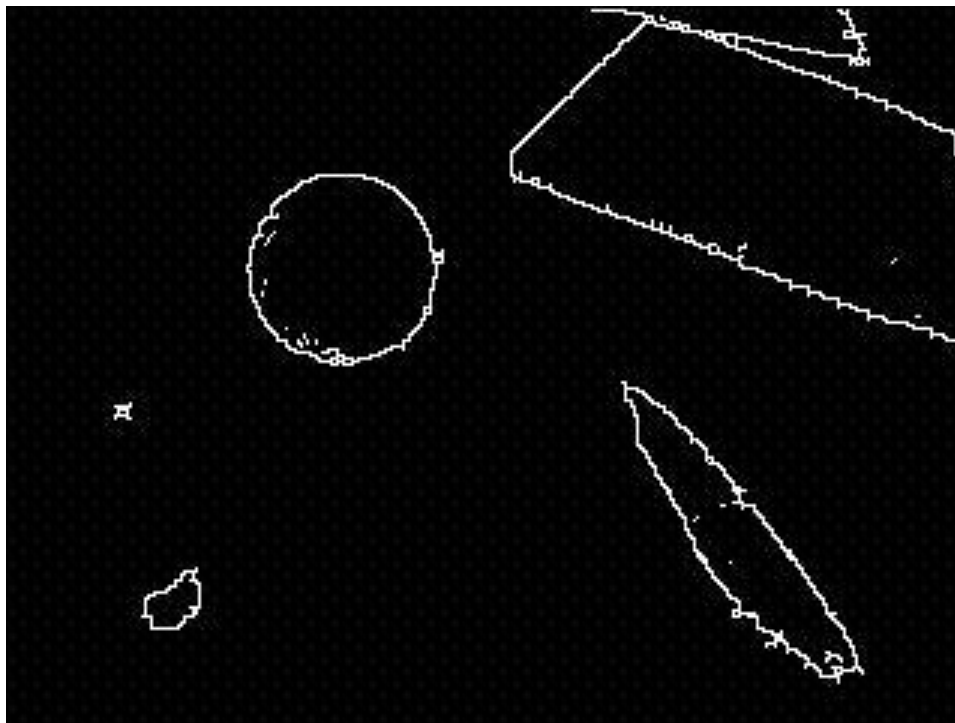
- vodorovné a zvislé hrany:



- rohy:



# Stenčené hrany



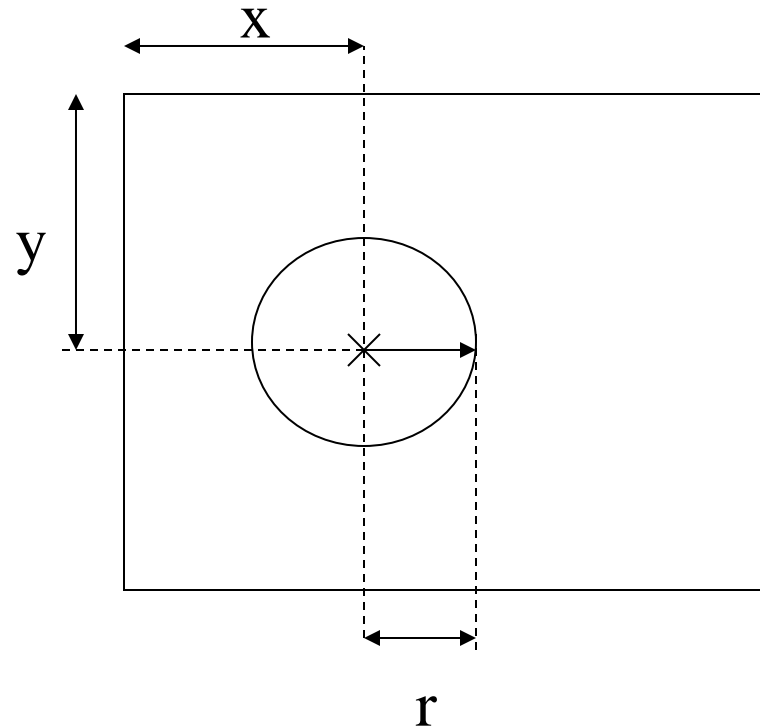
# Rozpoznávanie (pravidelného) tvaru

- Ako teraz zistíme ktoré hrany tvoria kružnicu ? Existuje tu viacero efektívnych ad-hoc metód, ale použijeme jednu všeobecnejšiu známu ako *Houghova transformácia*.
- Je založená na transformácii obrazu do priestoru hodnôt parametrov objektu, ktorý hľadáme

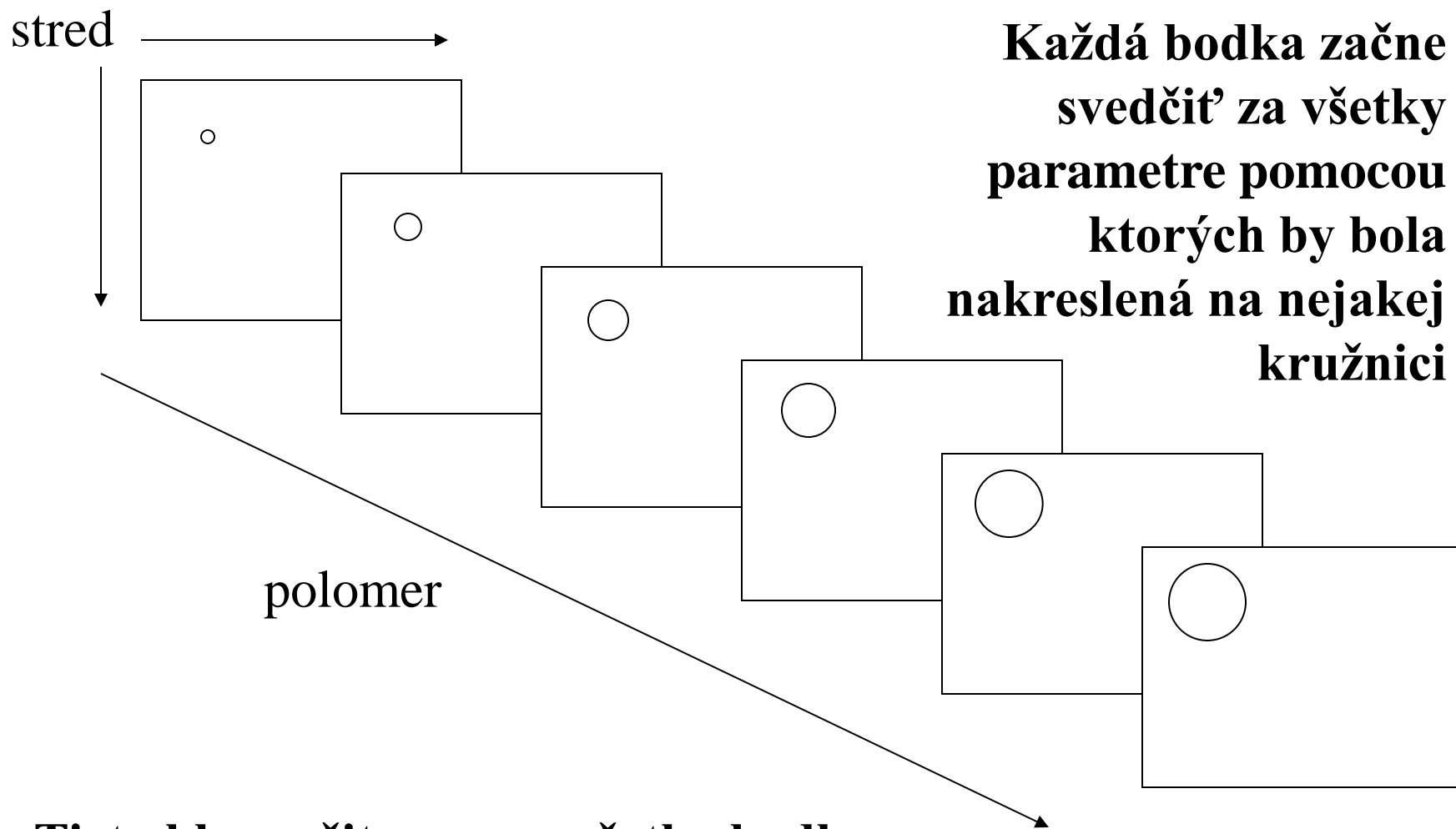
# Houghova transformácia

Vieme, že kružnicu môžeme reprezentovať tromi parametrami:

- $x$ -súradnica stredu
- $y$ -súradnica stredu
- polomer



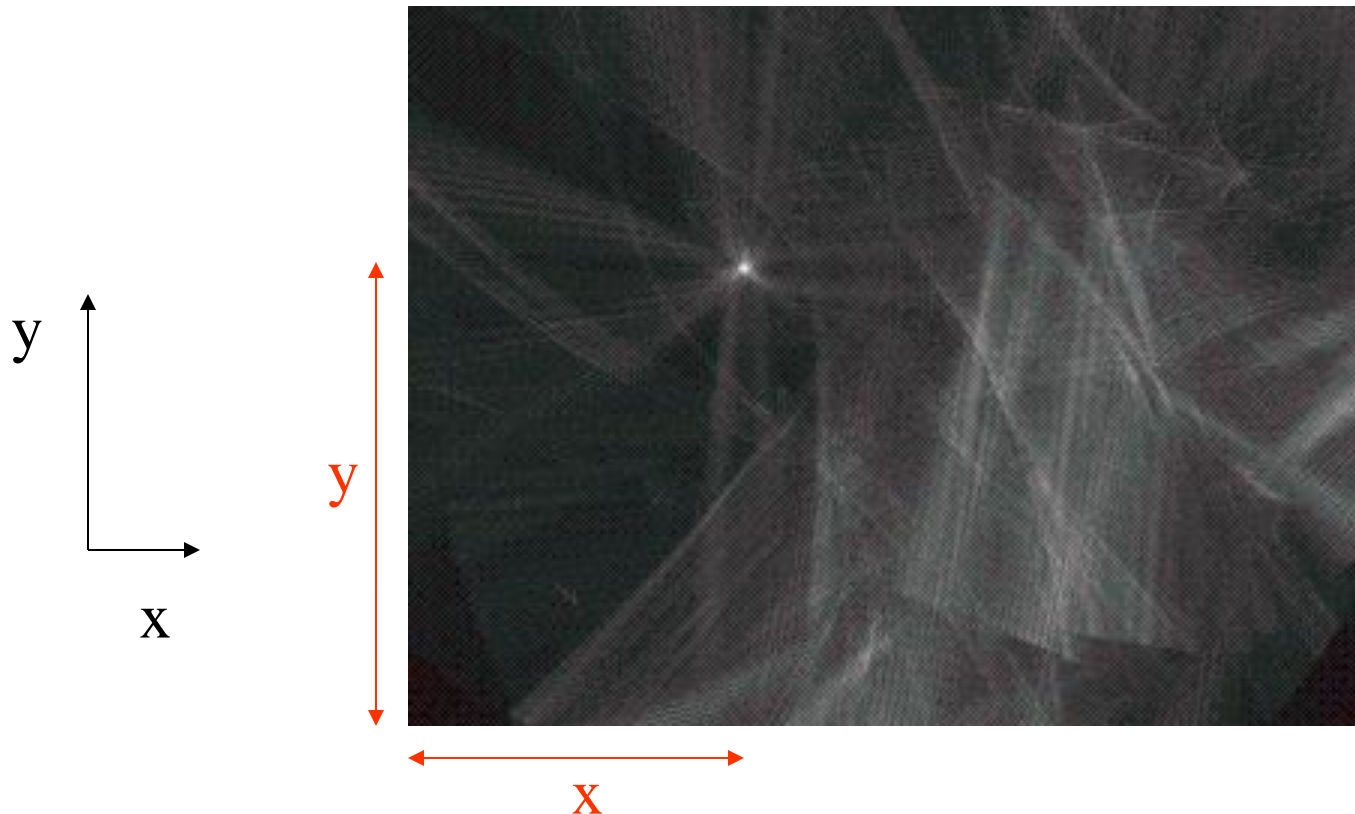
# Houghova transformácia



**Tieto hlasy sčítame pre všetky bodky**

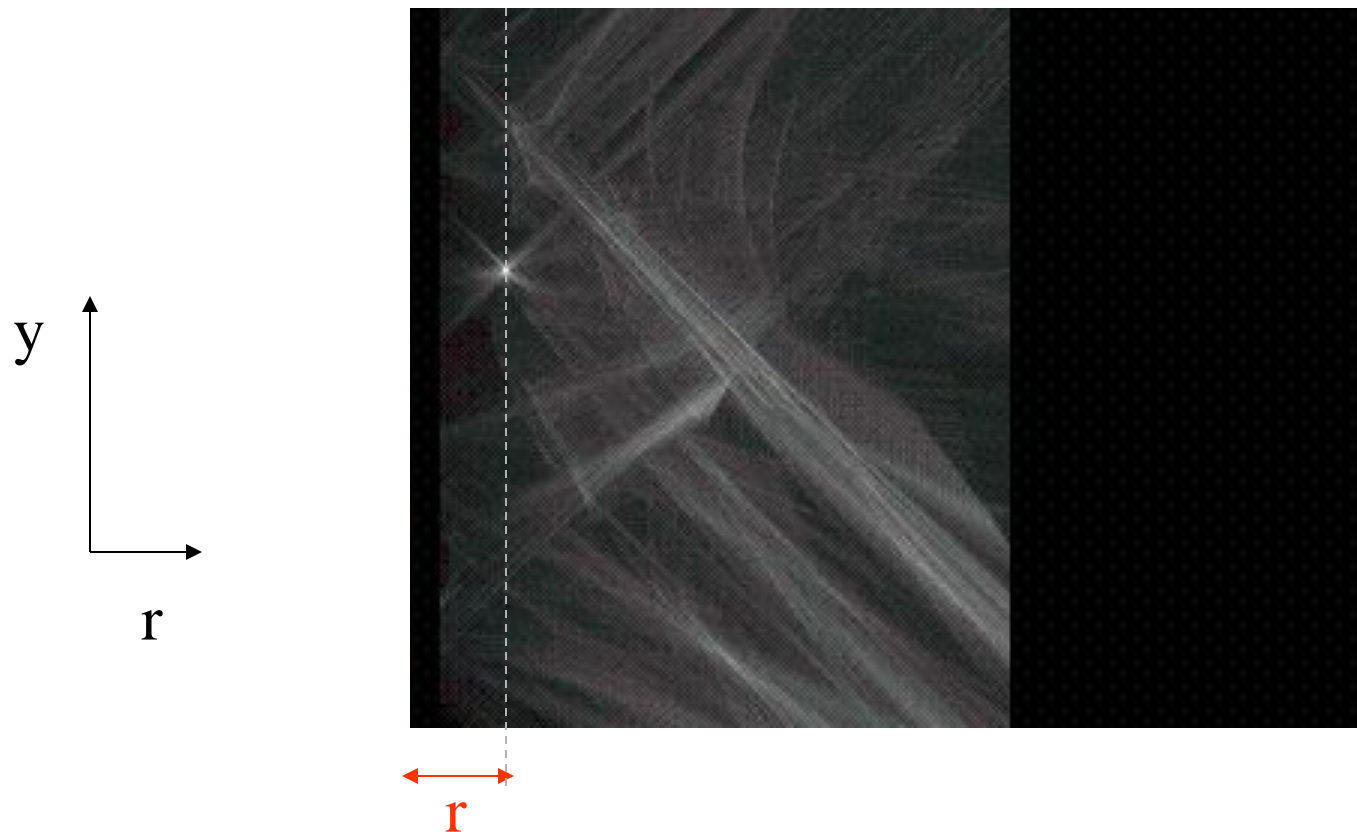


# Houghova transformácia



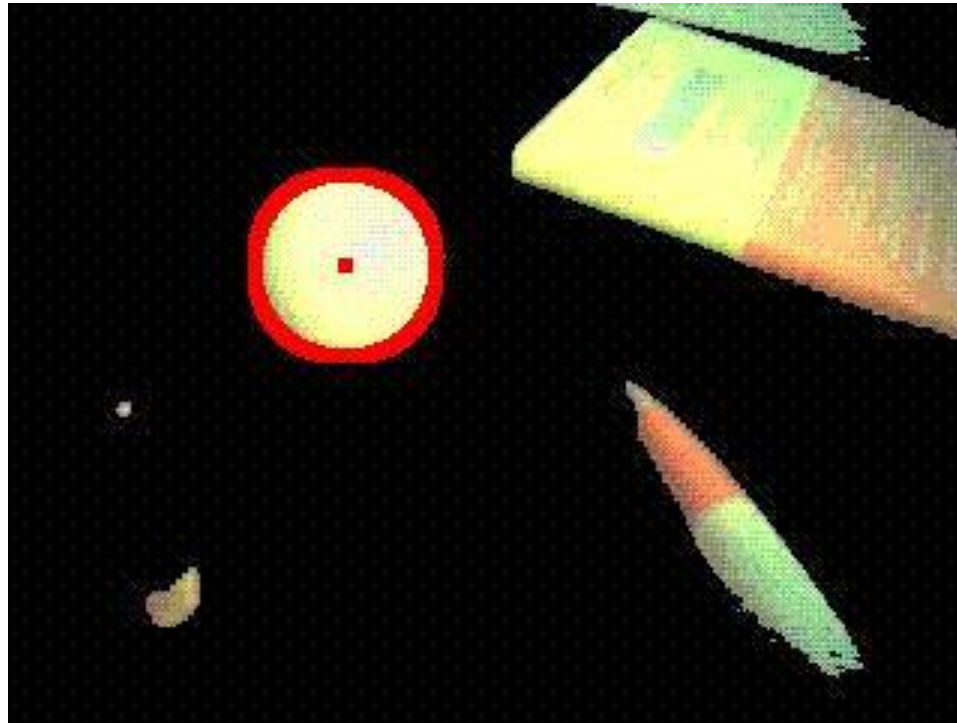
Keď sa na to pozrieme v smere polomeru  
vidíme stred nájdenej guľičky ako hviezdu na oblohe

# Houghova transformácia



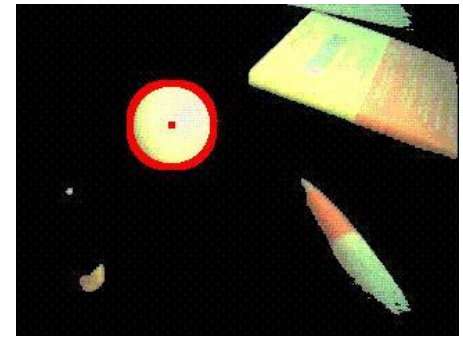
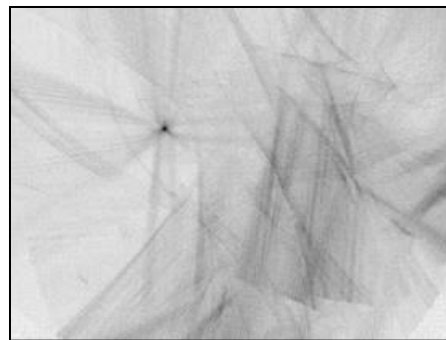
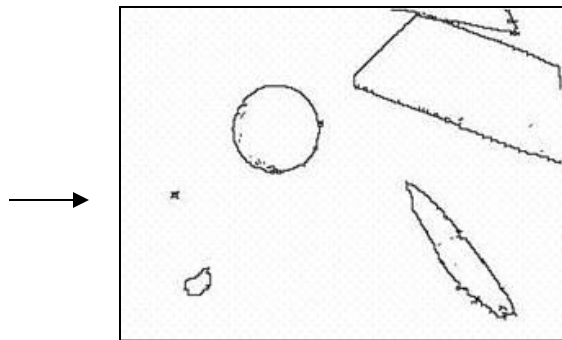
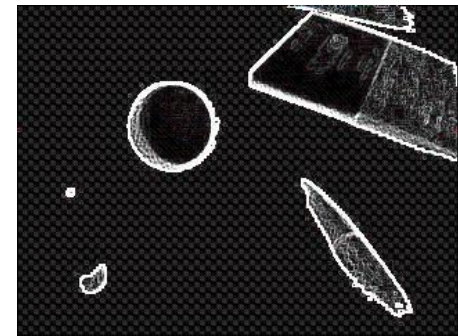
Polomer guľičky vidíme z iného pohľadu ...

# Houghova transformácia



... a určíme tak parametre kružnice na obraze

# Pravidelné objekty - sumár

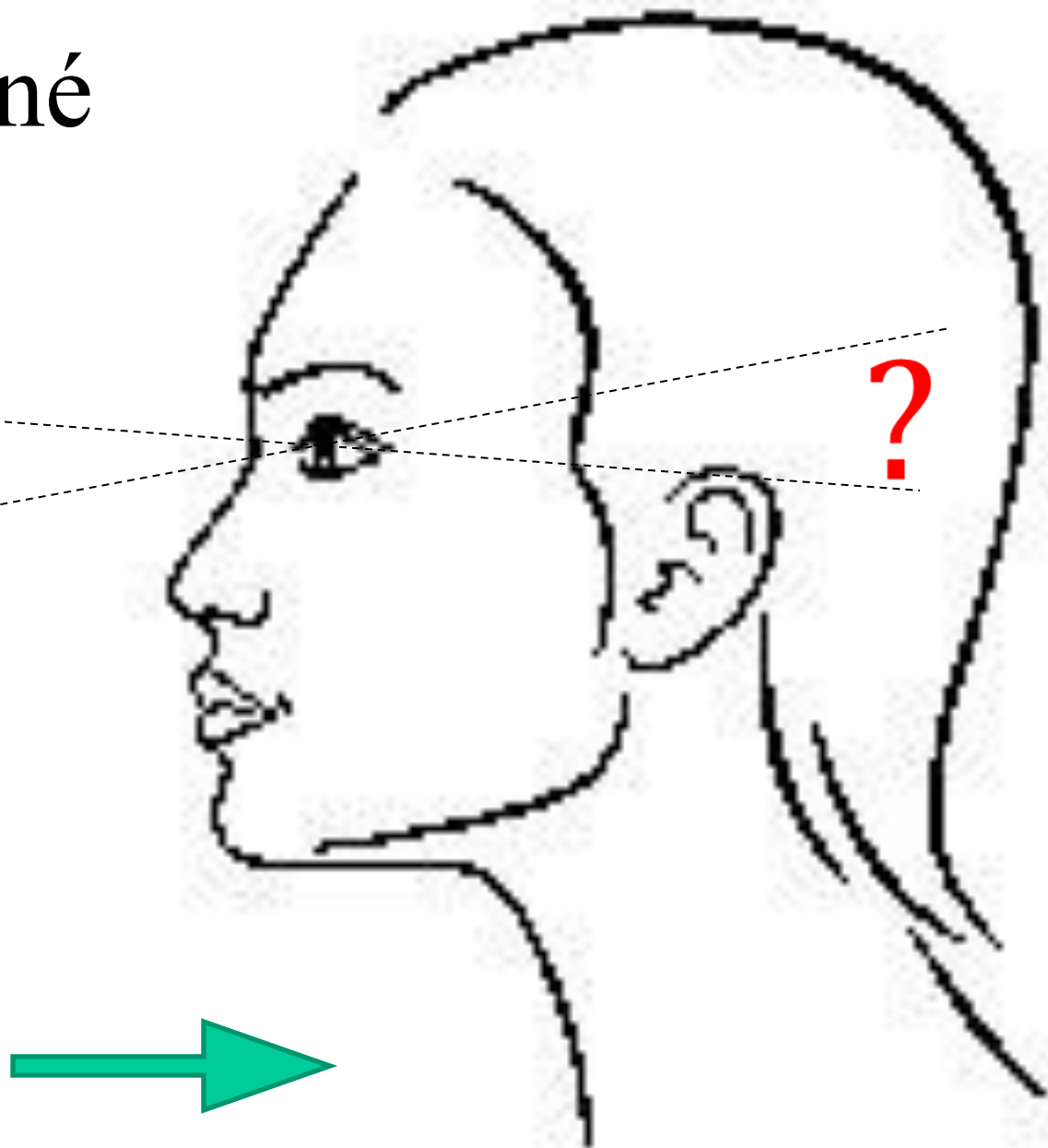


# Ako vidí robot nepravidelné objekty ?



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

# Nepravidelné objekty



# DOT

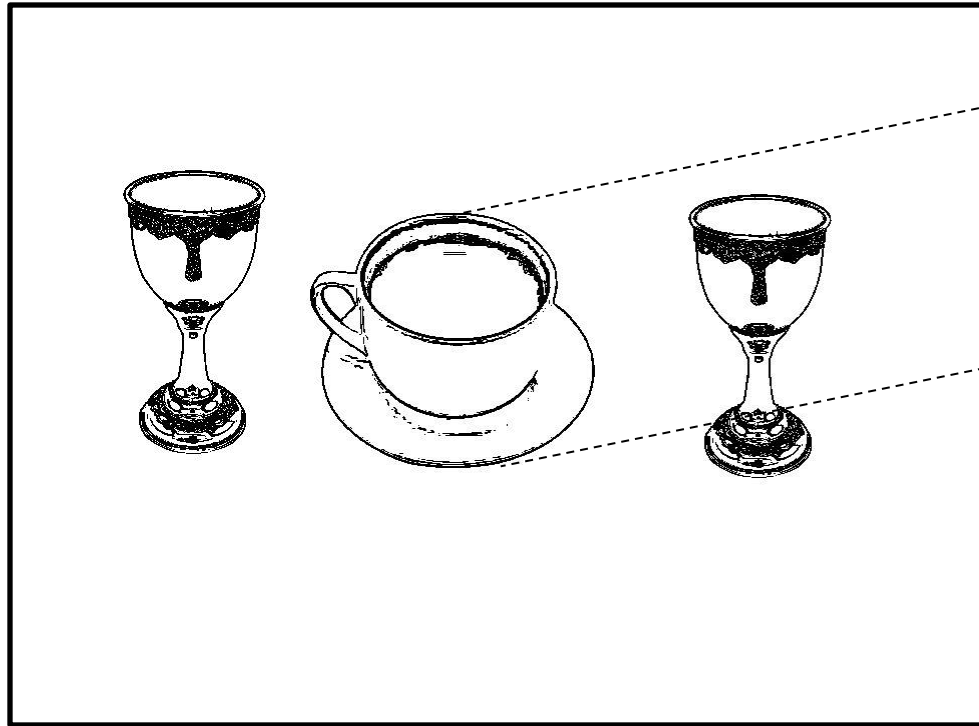
Dominant orientation templates

Šablóny význačných orientácii

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

# Motivácia

šablóna



obraz

zamerajme sa  
na hrany



# 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

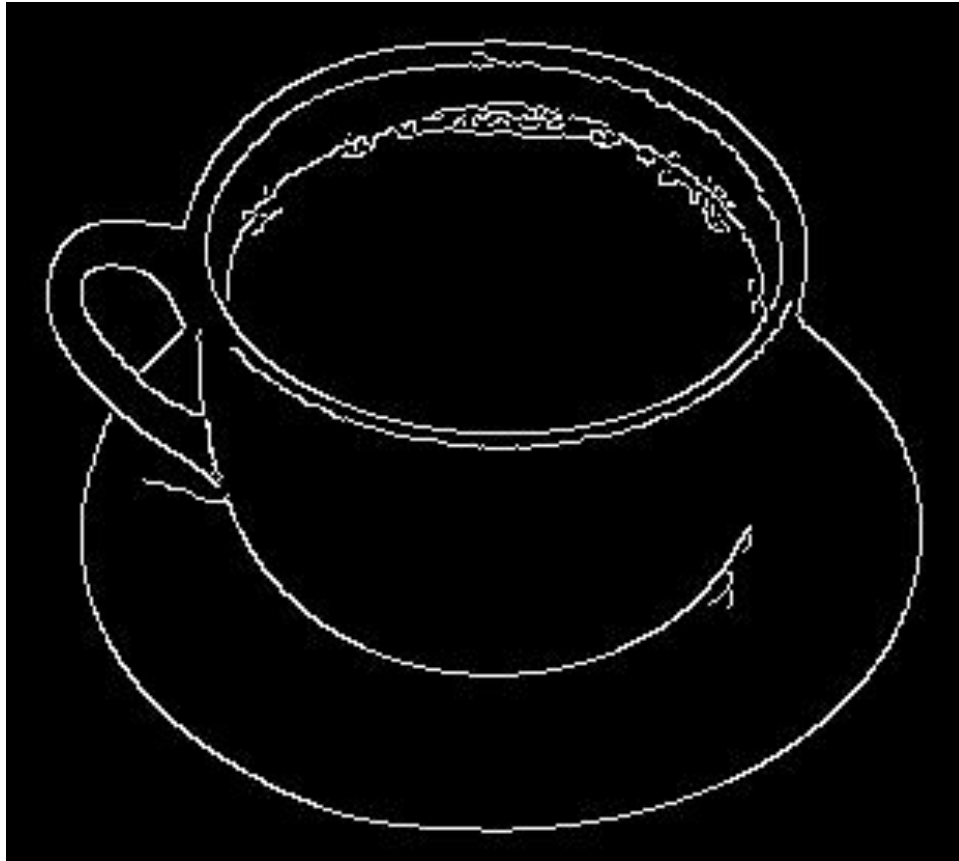
# Čiernobiely obraz



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

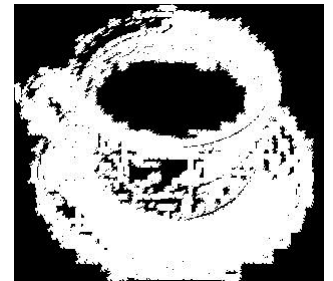
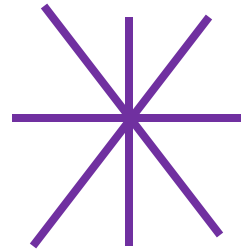
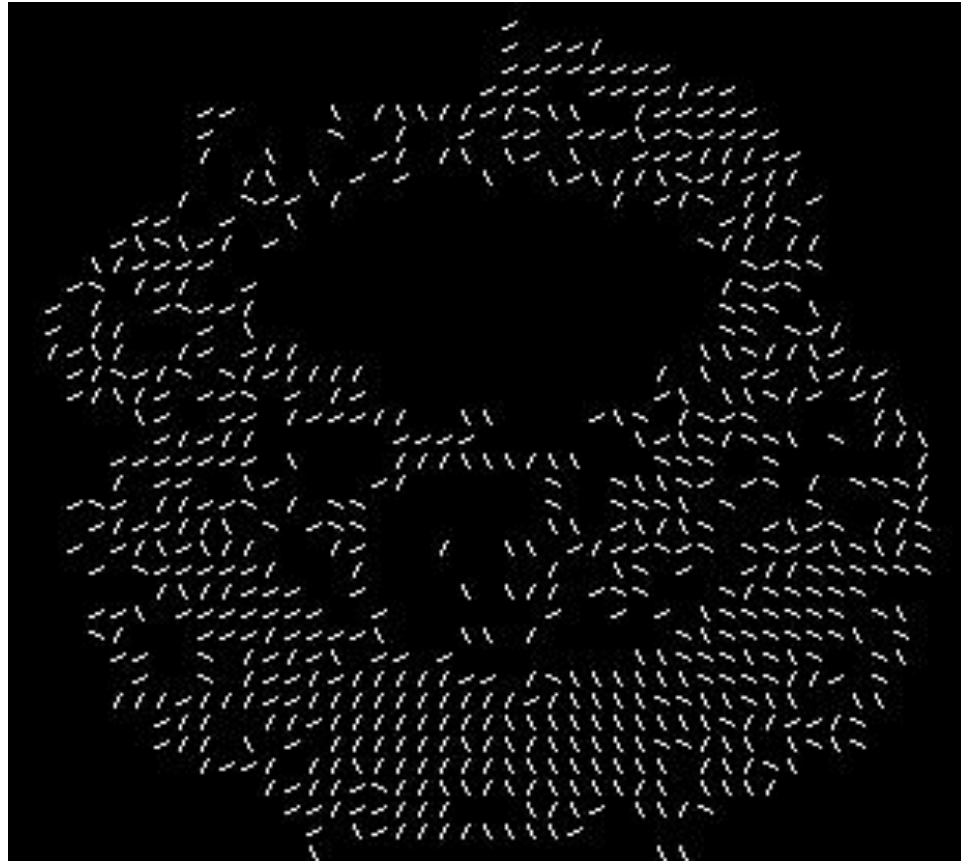
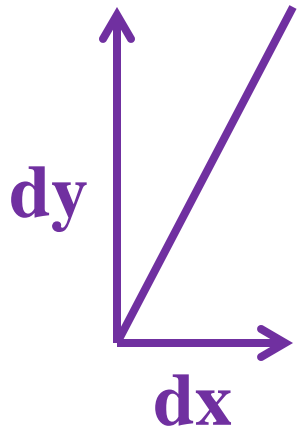
$$bw[i,j] = 0.3*r[i,j] + 0.59*g[i,j] + 0.11*b[i,j]$$

# Hrany



$dx$  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

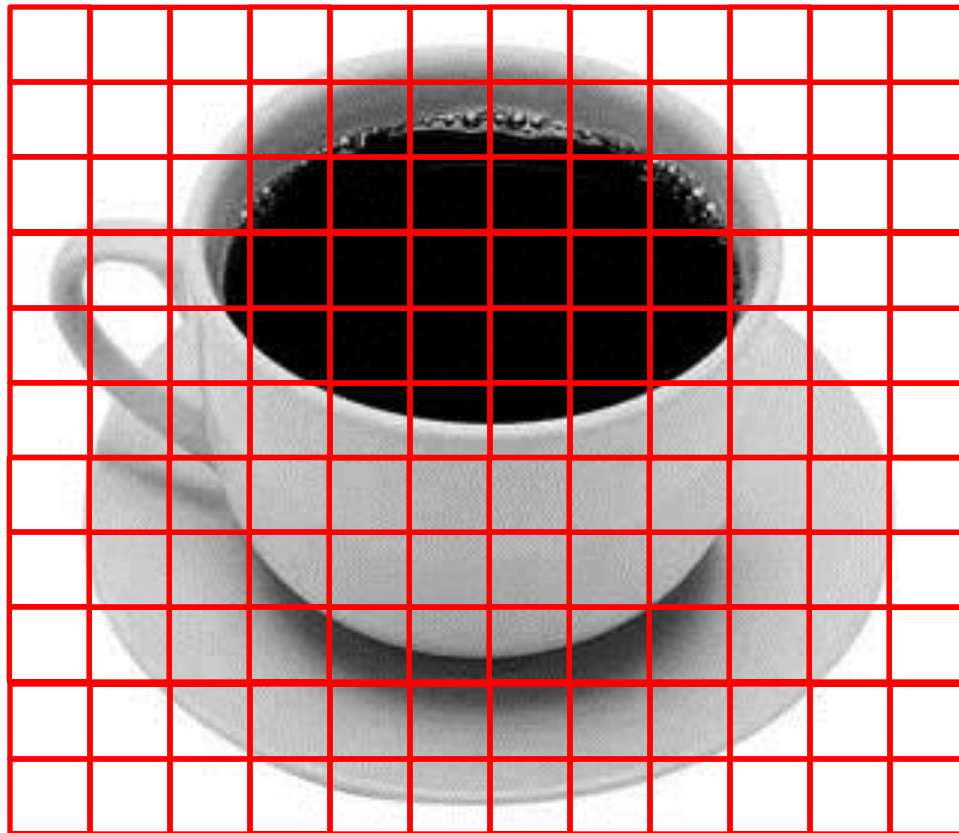
# Šablóna

- Práve orientácie použijeme na zostrojenie šablóny rozpoznávaného objektu



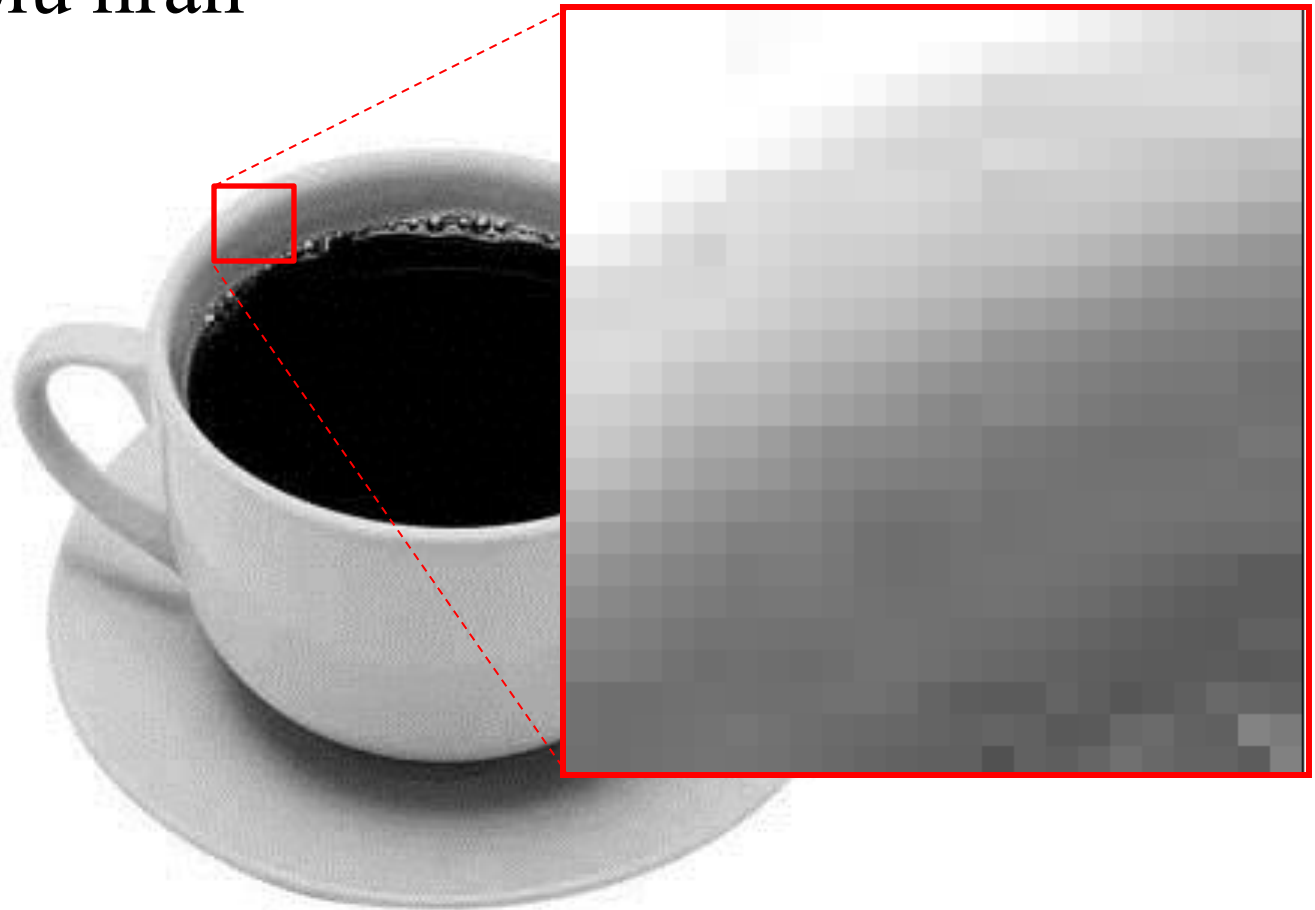
# Šablóna

- objekt pokryjeme neprekrývajúcimi sa regiónmi



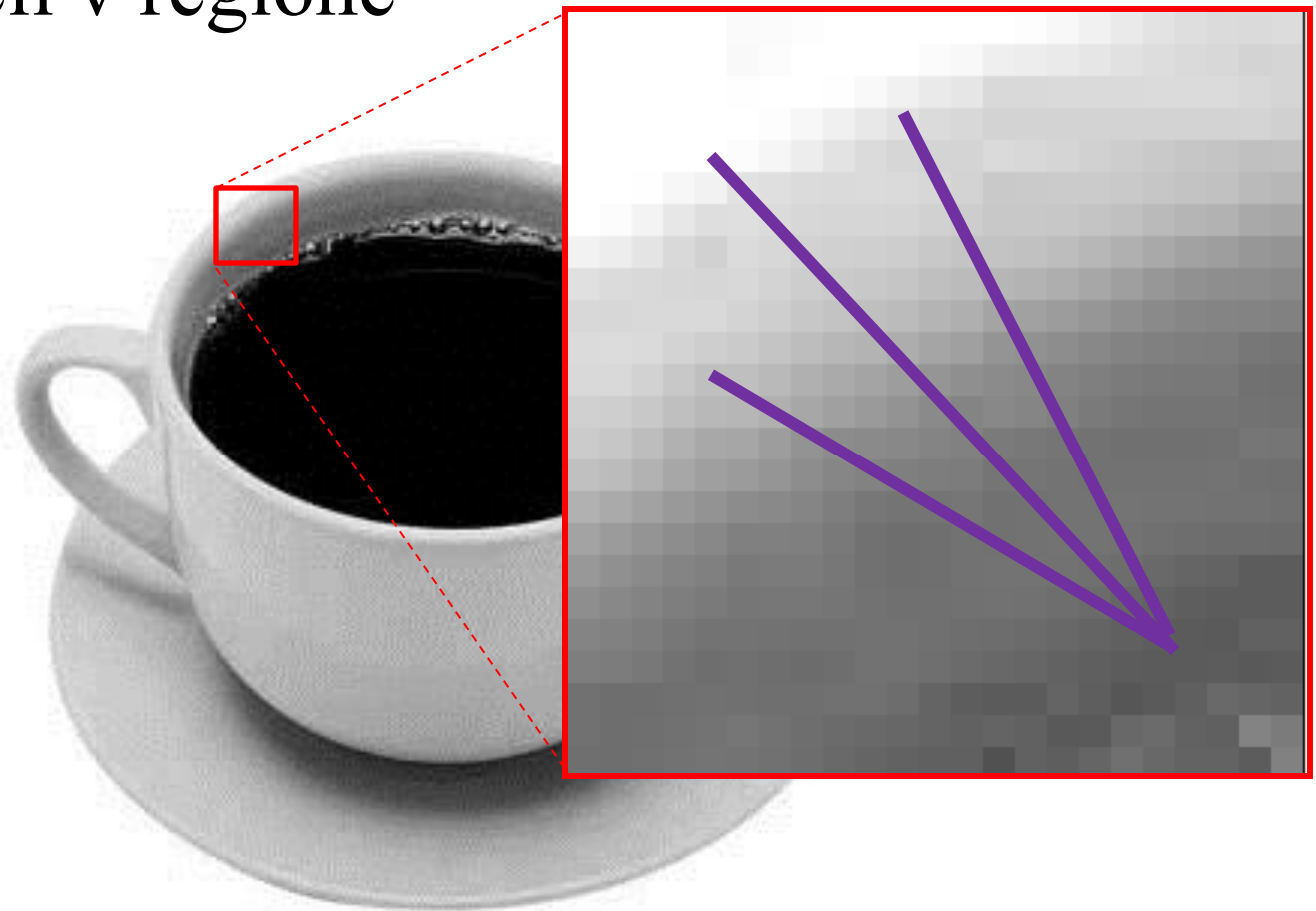
# Šablóna

- Ku každému pixelu v regióne vieme orientáciu hrán



# Šablóna

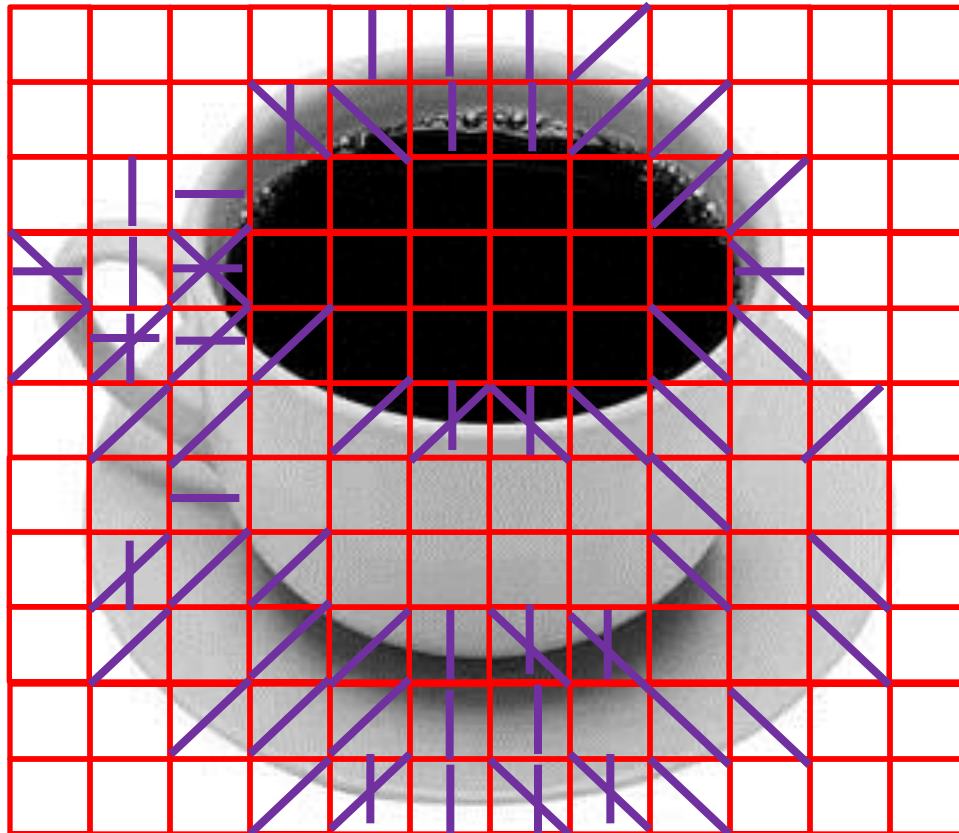
- Určíme z nich sadu prevládajúcich orientácií v regióne





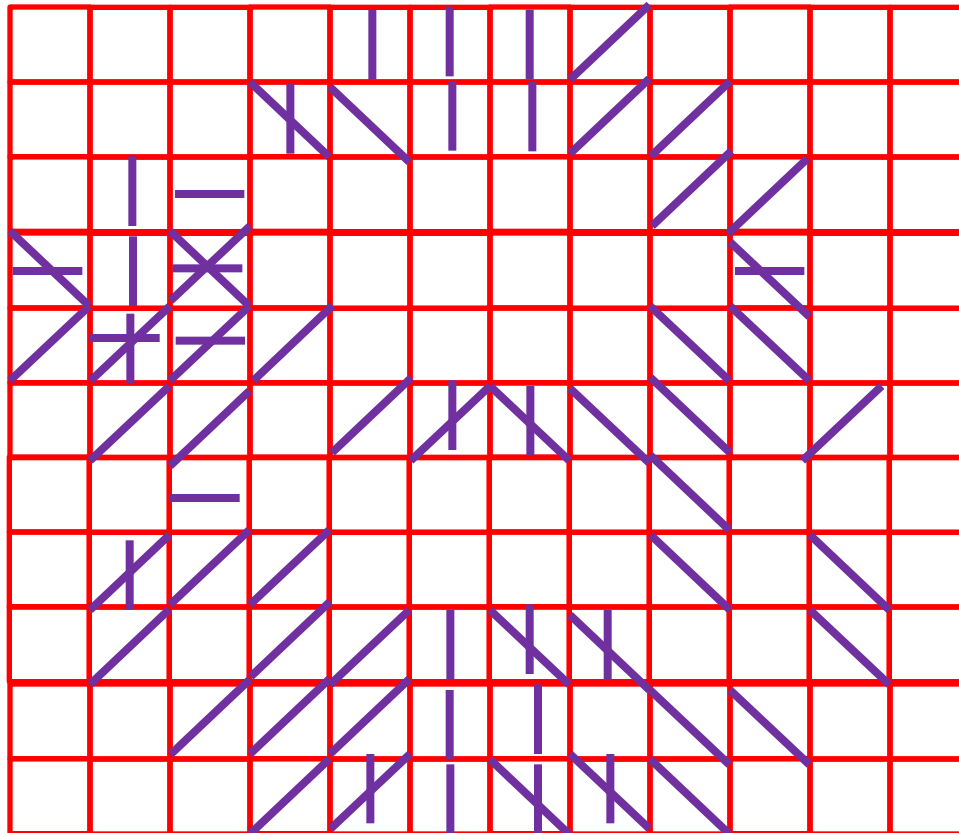
# Šablóna

- Pre každý štvorček tak dostaneme sadu orientácií



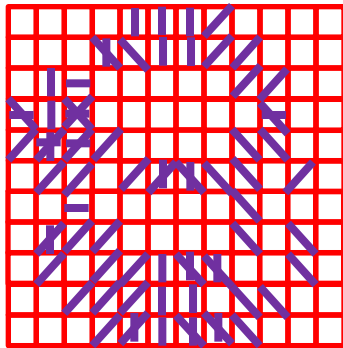
# Šablóna

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

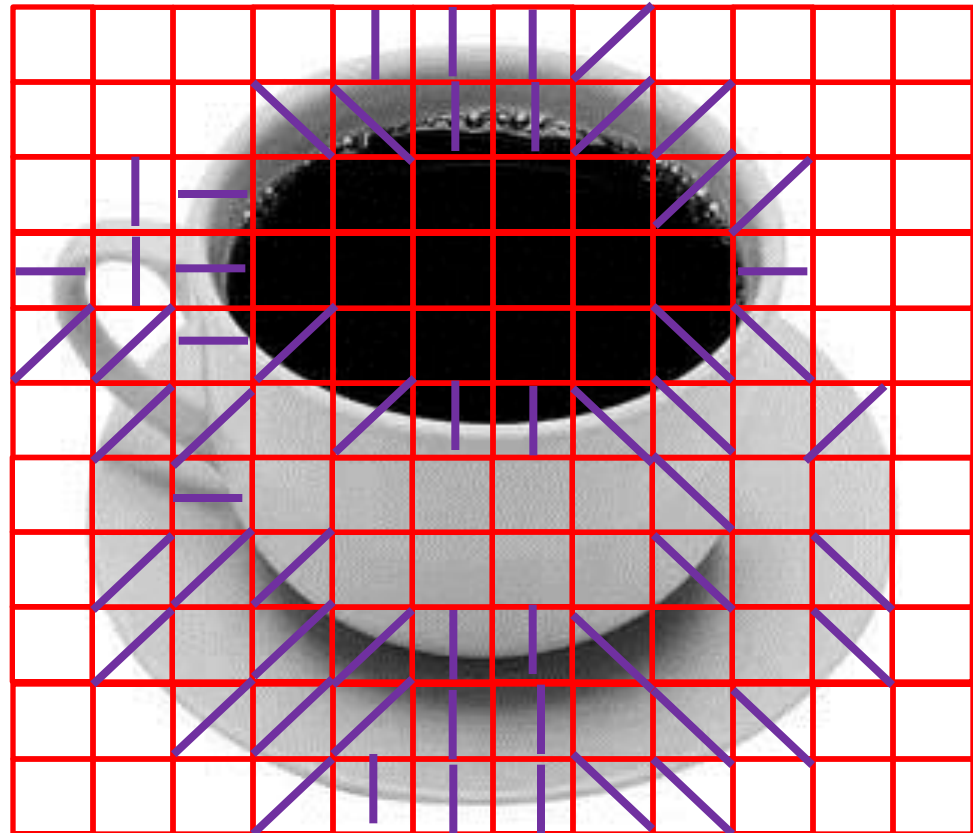


# Hľadanie podľa šablóny

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



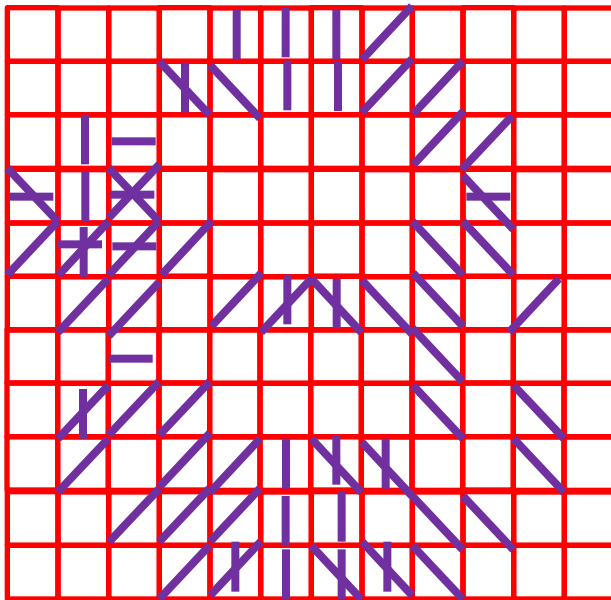
šablóna



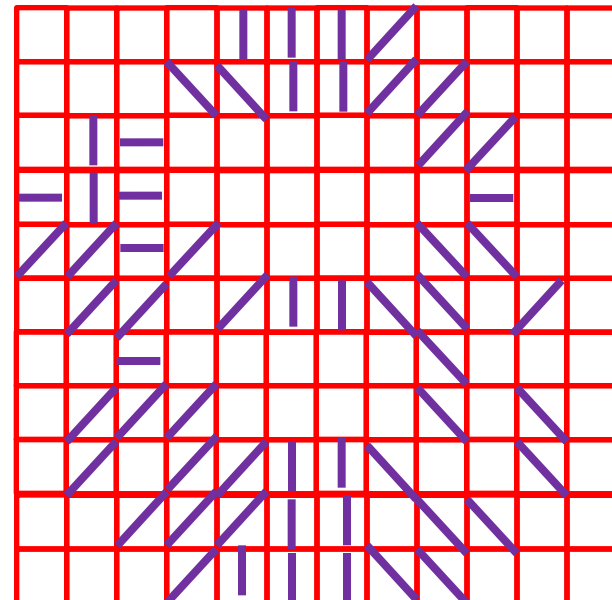
obraz

# Hľadanie podľa šablóny

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



šablóna

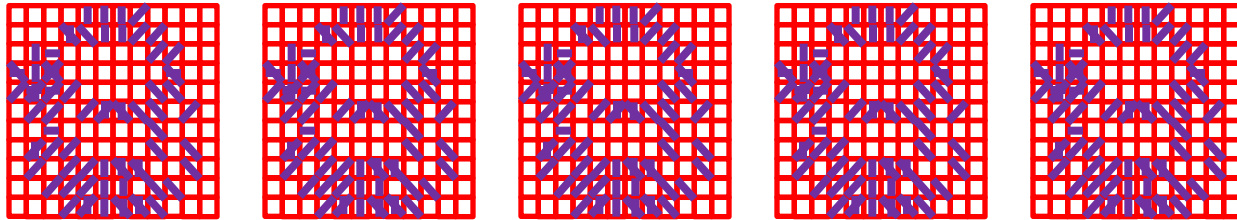


obraz

# Nefunguje to ?

- Nemáme záruku, že sme trafili región na región. Ani že sa dívame z rovnakého pohľahu. Nevadí, narobíme si veľa šablón pre rôzne posuny a pohľady

šablóny



Sada takýchto šablón bude reprezentáciou objektu

# Nepravidelné objekty - sumár



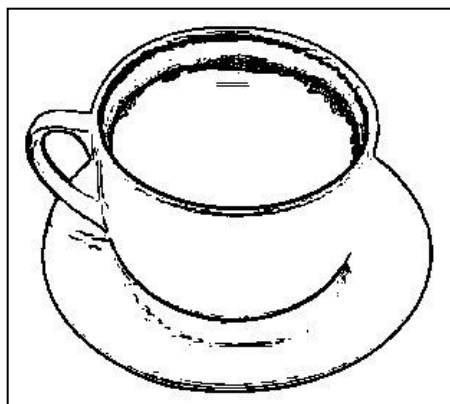
obraz



dx



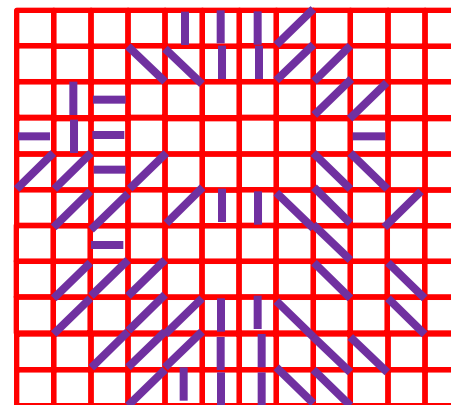
dy



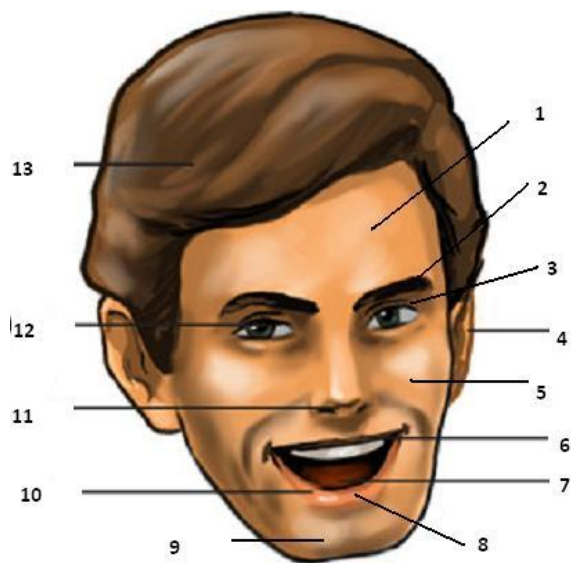
hrany



orientácie



# Vidí robot alebo podvádza ?

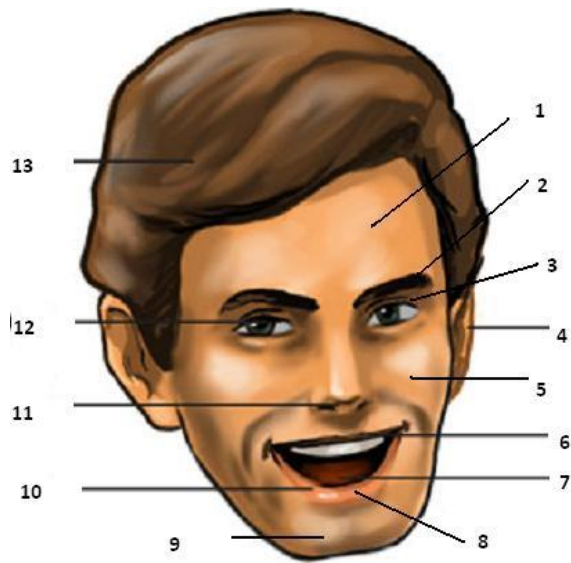


Podvádza!



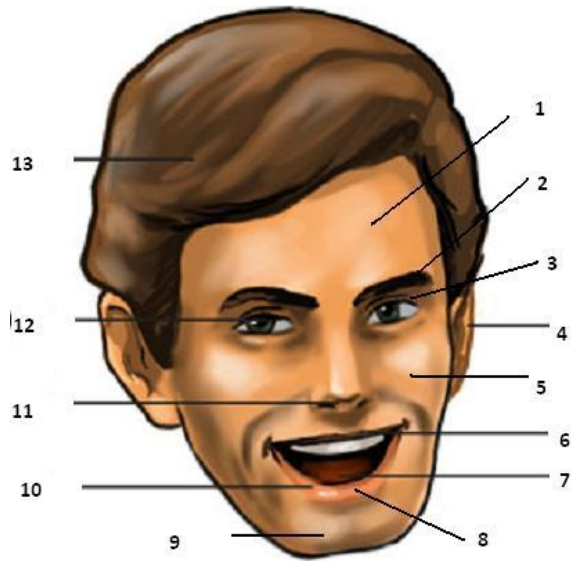


# Lenže, vidí člověk alebo podvádza ?



**FINISHED FILES ARE THE RE-  
SULT OF YEARS OF SCIENTIF-  
IC STUDY COMBINED WITH THE  
EXPERIENCE OF YEARS**

# Podvádzajú obaja !



**Ďakujem za pozornosť !**

**Andrej Lúčny**

**Katedra aplikovanej informatiky**

**FMFI UK Bratislava**

**[andy@microstep-mis.com](mailto:andy@microstep-mis.com)**

**[www.microstep-mis.com/~andy](http://www.microstep-mis.com/~andy)**