

# Spracovanie farebného obrazu

Kvantovanie II.  
Detekcia hrán vo farebnom obraze

8.4.2014

# Mediancut

- Reprezentant
  - priemer
- časová a priestorová náročnosť
- Vylepšenie
  - prestať s delením, keď už oko nerozlišuje susedné farby
  - v modrých prestaneme deliť skôr, než v červených a potom v zelených farbách

# K-means

- Vygeneruj začiatkové rozdelenie
- Nájdí centroid každého zhluku
- Pre každú farbu:
  - Vyrátaj vzdialenosť od každého centroidu
  - Prirad' k najbližšiemu zhluku
- Vyrátaj nové centroidy
- Opakuj, kým nie sú zhluky stabilné ( $MSE < \text{prah}$ )

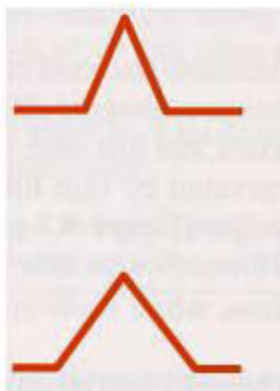


Aké mame typy hrán?

# Aké máme typy hrán?



schod  
rampa

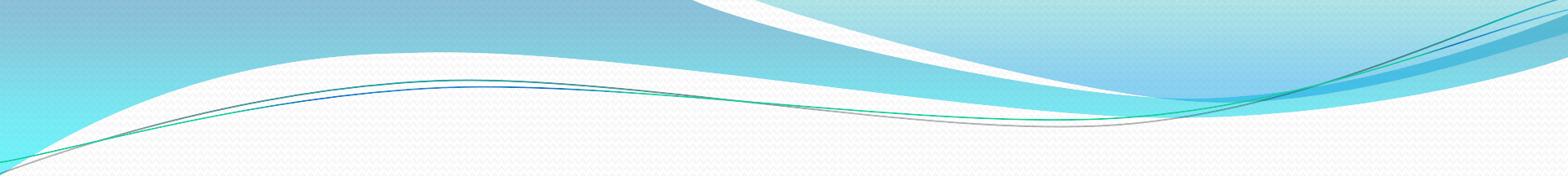


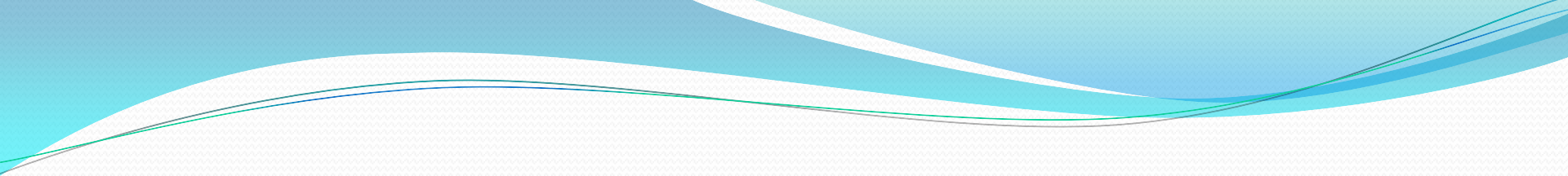
strecha



čiara  
hrebeň

- 
- Aký je veľký problém pri detekcii hrán?

- 
- Aký je veľký problém pri detekcii hrán?
    - šum
  - Ako tento problém môžeme riešiť?

- 
- Aký je veľký problém pri detekcii hrán?
    - šum
  - Ako tento problém môžeme riešiť?
    - Vyhladenie obrázka



# Detekcia hrán

- Aké typy detekcie hrán poznáte?

# Detekcia hrán

- Sobel
- Obrázky  $G_x$  a  $G_y$  konvolúciou z

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$

# Detekcia hrán

- Prewitt filter, Roberts
- Prewitt

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * \mathbf{A}$$

- Roberts

$$\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$$

# IPT

- Metódy
  - Sobel (Sobelova aproximácia derivácie)
  - Canny (Noise red., 4 filters, hister. thres)
  - Roberts (Robertsova aprox. derivácie)
  - Prewitt (Prewitt aprox. derivácie)
  - Log (Laplacian of Gaussian method)
  - Zero crossingge(I)

# Detekcia hrán vo farebnom obraze

- Ako môžeme detekovať hrany vo farebnom obraze?
  - previesť na šedotónový a použiť niektorý z predchádzajúcich metód
- Problém

# Detekcia hrán vo farebnom obraze

- Ako môžeme detekovať hrany vo farebnom obraze?
  - previesť na šedotónový a použiť niektorý z predchádzajúcich metód
- Problém
  - ak je hrana medzi dvomi farbami s rovnakým jasom
- Riešenie

# Detekcia hrán vo farebnom obraze

- Ako môžeme detekovať hrany vo farebnom obraze?
  - previesť na šedotónový a použiť niektorý z predchádzajúcich metód
- Problém
  - ak je hrana medzi dvomi farbami s rovnakým jasom
- Riešenie
  - vo farebnom obraze vieme určiť 90% hrán z šedotónového obrazu
  - zvyšných 10% hrán z farebného obrazu

# Detekcia hrán vo farebnom obraze

- Sekvenčný prístup:
  - Jednotlivé kanály samostatne

$$G(x, y) = \sqrt{(G^2_{red} + G^2_{green} + G^2_{blue})}$$



# Vektorový prístup

- Difference vector operators
- vypočítame gradient v 4 smeroch

$$|\nabla f|_{0^\circ} = \|Y_{0^\circ} - X_{0^\circ}\|$$

$$|\nabla f|_{90^\circ} = \|Y_{90^\circ} - X_{90^\circ}\|$$

$$|\nabla f|_{45^\circ} = \|Y_{45^\circ} - X_{45^\circ}\|$$

$$|\nabla f|_{135^\circ} = \|Y_{135^\circ} - X_{135^\circ}\|$$

$$DV = \max(|\nabla f|_{0^\circ}, |\nabla f|_{45^\circ}, |\nabla f|_{90^\circ}, |\nabla f|_{135^\circ})$$



Fig. 11. DV detector: edges of the noised image 'Lena'

- X, Y sú 3D vektorové konvolučné masky

- Základná maska pre okno 3x3
- $v(x,y)$  – pixel
- $v(x_0,y_0)$  – stredný pixel

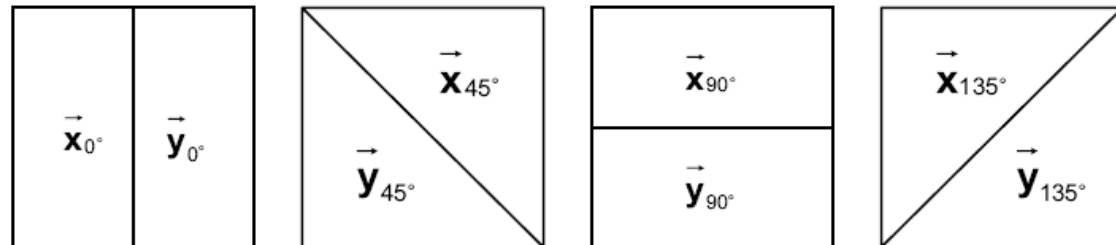
$$X_{0^\circ} = v(x_{-1}, y_0), \quad Y_{0^\circ} = v(x_1, y_0)$$

$$X_{45^\circ} = v(x_{-1}, y_1), \quad Y_{45^\circ} = v(x_1, y_{-1})$$

$$X_{90^\circ} = v(x_0, y_1), \quad Y_{90^\circ} = v(x_0, y_{-1})$$

$$X_{135^\circ} = v(x_1, y_1), \quad Y_{135^\circ} = v(x_{-1}, y_{-1})$$

- Pred detekciou môžeme obraz filtrovať – treba použiť väčšiu masku
- Ak okno  $W$  je veľkosti  $n \times n$  ( $n=2k+1$ ) vytvoríme sub-okno veľkosti  $N = (n^2-1)/2$



Configuration of sub-windows for VD filters

# Zašumený obraz

- Vector median filter

- Efektívny pri redukovaní impulsného šumu

$$f_{\text{VM}}(v_1, v_2, \dots, v_N) = v^{(1)}$$

- Vector mean filter

- Efektívny pri redukovaní Gaussovho šumu

$$f_{\text{VM}}(v_1, v_2, \dots, v_N) = \frac{1}{N} \sum_{i=1}^N v_i$$

- $\alpha$ -trimmed mean filter

$$f_{\alpha\text{-trim}}(v_1, v_2, \dots, v_N) = \frac{1}{N(1-2\alpha)} \sum_{i=1}^{N(1-2\alpha)} v^{(i)}$$

where:  $\alpha$  is within  $[0, 0,5)$  interval.

- Adaptive nearest neighbour filter

$$f_{\text{adap}}(v_1, v_2, \dots, v_N) = \sum_{i=1}^N w_i v_i$$

# Úloha

- Pre každý farebný kanál R, G a B
  - Vytvoriť obrázky  $G_x$  a  $G_y$  konvolúciou
    - Konvolúciou masiek Sobela a jednotlivých kanálov (conv2)

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A}$$

- Následne určiť hrany
    - $G_{\text{red}} = \sqrt{(G_x^2 + G_y^2)}, \dots$
- Detekovať hrany vo farebnom obraze:

$$G(x, y) = \sqrt{(G_{\text{red}}^2 + G_{\text{green}}^2 + G_{\text{blue}}^2)}$$