



***Modifikácia adaptívneho prahovania
pre zlepšenie čitateľnosti nevhodne
skenovaných či fotografovaných
dokumentov***

Autor: Miroslav Kobliška

Školiteľ: RNDr. Róbert Bohdal, PhD.

Cieľ

- Chceme upraviť textový dokument, tak aby ho OCR systém vedel lepšie rozpoznať

platiť plné učiteľské platy.
má byť zriadená 3 členná rozhodčia
pp.dr.Oskar Reiss,ing.V.Holz a riad.

:Rozhodnuté bolo poštou podať m e m o
a potom každá strana si menuje po
ct./Viď tu I 2/g./
rokov.arzákať cestovníar -

platiť plné učiteľské platy.
má byť zriadená 3 členná rozhodčia
pp.dr.Oskar Reiss,ing.V.Holz a riad.

:Rozhodnuté bolo poštou podať m e m o
a potom každá strana si menuje po
ct./Viď tu I 2/g./
rokov.arzákať cestovníar -

Skupiny prahovacích techník

1. Globálne
2. Lokálne

Globálne prahovanie

- Najst' jednu prahovú hodnotu pre celý dokument
- Rýchle

Lokálne prahovanie

- Je založené na vypočítaní prahov samostatne pre každý pixel, tak že použijeme informáciu z lokálneho okolia daného pixelu
- Sú pomalšie – počítame prah pre každý pixel zvlášť

Integrálny obraz

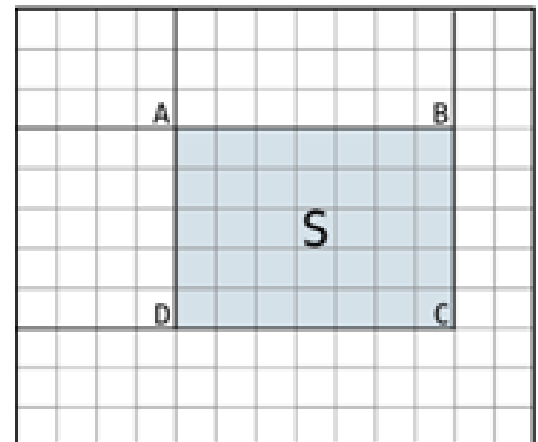
- Urýchľuje rýchlosť prahovania
- Výpočet :

$$I(x,y) = f(x,y) + I(x-1,y) + I(x,y-1) - I(x-1,y-1).$$

$$S = \text{obdĺžnik}(C) - \text{obdĺžnik}(B) - \text{obdĺžnik}(D) + \text{obdĺžnik}(A)$$

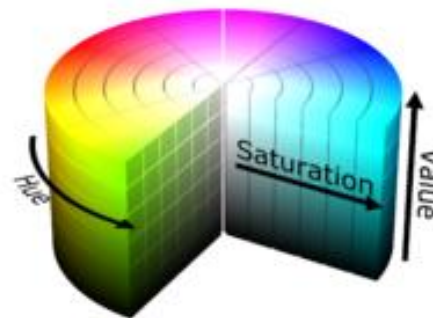
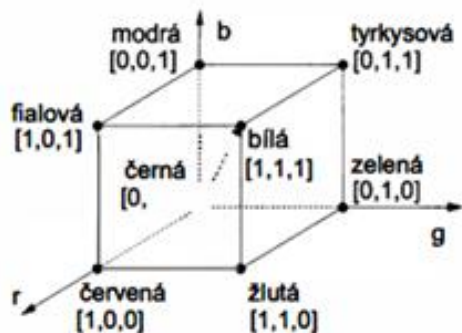
4	1	2	2
0	4	1	3
3	1	0	4
2	1	3	2

4	5	7	9
4	9	12	17
7	13	16	25
9	16	22	33



Farebné modely RGB a HSV

- RGB - červená (R, red), zelená (G, green) a modrá (B, blue)
- Farby vyjadrujeme farebným vektorom, ktorého zložky nadobúdajú hodnoty z $[0, 255]$
- HSV - farebný tón (H, hue), sýtosť (S, saturation) a jasová hodnota (V, value)
- Farebný tón označuje prevládajúcu spektrálnu farbu, sýtosť určuje prímies iných farieb a jas je daný množstvom bieleho svetla



Konverzia z RGB do HSV

- Chceme robiť aj s farebnými dokumentami
- Postupujeme nasledovne
- $R' = R/255$
- $G' = G/255$
- $B' = B/255$
- Následne vypočítame hodnoty:
- $C_{max} = \max(R', G', B')$
- $C_{min} = \min(R', G', B')$
- $\Delta = C_{max} - C_{min}$

Konverzia z RGB do HSV

$$H = \begin{cases} 0^{\circ}, & \Delta = 0 \\ 60^{\circ} * \left(\frac{G' - B'}{\Delta} \bmod 6\right), & c_{max} = R' \\ 60^{\circ} * \left(\frac{B' - R'}{\Delta} + 2\right), & c_{max} = G' \\ 60^{\circ} * \left(\frac{R' - G'}{\Delta} + 4\right), & c_{max} = B' \end{cases}$$

$$S = \begin{cases} 0, & c_{max} = 0 \\ \frac{\Delta}{c_{max}}, & c_{max} \neq 0 \end{cases}$$

$$V = c_{max}$$

Prečo konverzia do HSV?

- Pri stmavovaní pracujeme iba so zložkou V
- Pri zosvetľovaní pracujeme so zložkami S a V
- Hodnotu S znižujeme
- Hodnotu V zvyšujeme

Metódy nájdenia lokálneho prahu

- V našej práci sme sa rozhodli aplikovať nasledovné metódy:

1. Otsu [Chad16]

2. Bradley [Br16] $T_B = \mu * (100 - k)$

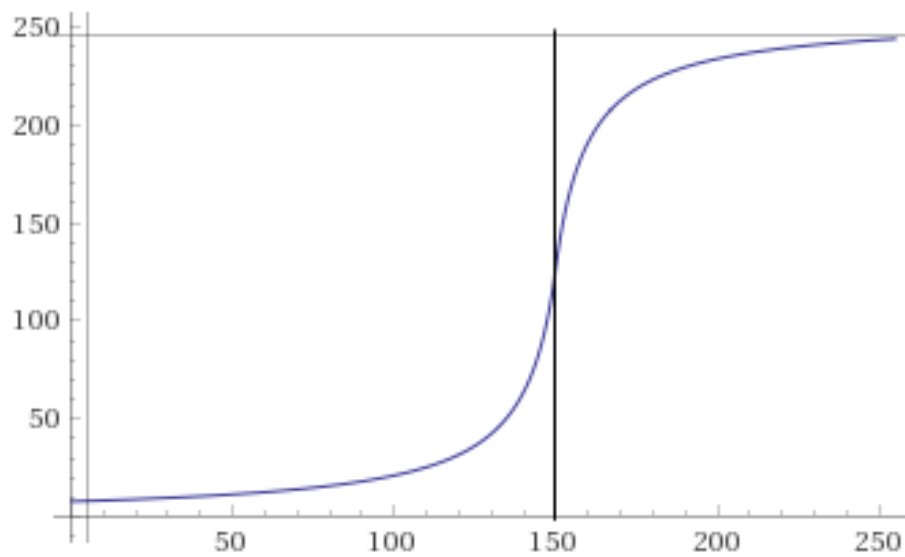
3. Sauvola [Chad16]

$$T_S = \mu * \left(1 + k * \left(\frac{\sum (p_i - \mu)^2}{R} - 1 \right) \right)$$

4. Niblack [Chad16] $T_N = \mu + k \sqrt{\frac{1}{N} \sum (p_i - \mu)^2}$

Hladká prahovacia funkcia

- $f(p) = \frac{255}{2} * \left[\frac{k*(p-t)}{1+k*|p-t|} + 1 \right]$
- Funkcia vyjadruje šedotónové hodnoty

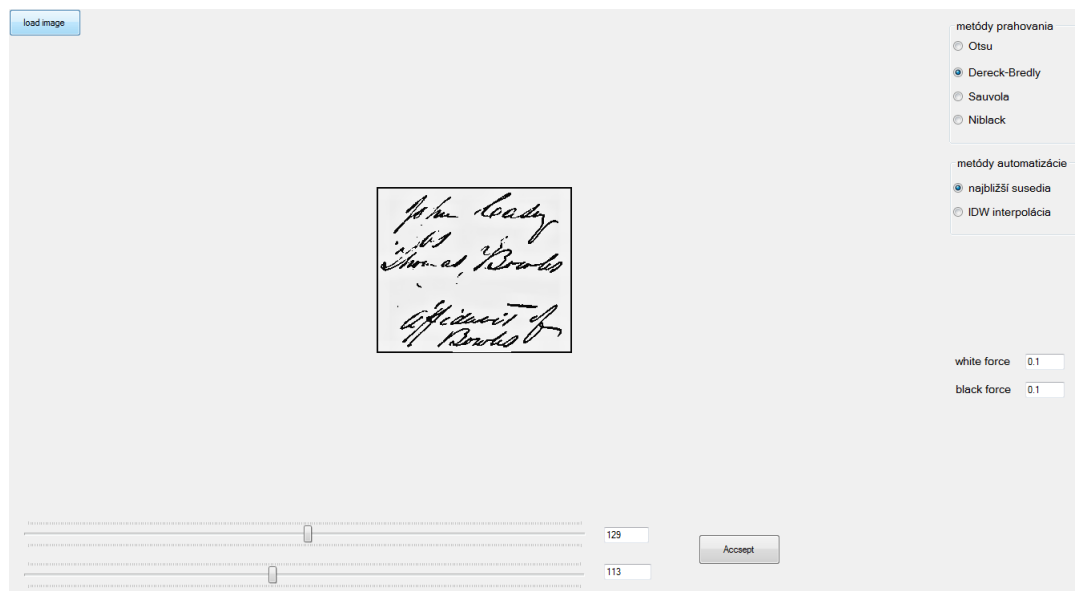


Implementácia

- Na implementáciu používame programovací jazyk C#
- Používame OpenCV knižnicu

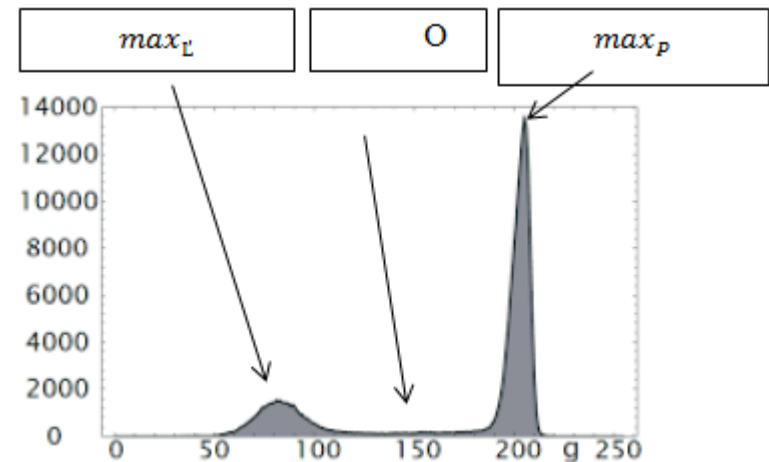
Posuvníky

- Posuvníky nám slúžia ako prah, ich ťahaním posúvame hladkú prahovaciu funkciu
- Posuvník pre stmavovanie
- Posuvník pre zosvetlovanie



Prahovanie neznámeho dokumentu 1

- Predpokladajme, že už máme spracovaných niekoľko dokumentov
- Pri načítaní dokumentu vytvoríme histogram, vyhladíme ho a zistíme nasledovné hodnoty:
 1. Otsu globálny prah (O)
 2. Maximum v histograme naľavo od prahu (max_L)
 3. Maximum v histograme napravo od prahu (max_P)



Vyhladenie histogramu

- Pomocou konvolúcie
- Používame Gaussovské jadro
- $1/273 * [1, 4, 7, 4, 1]$
- Po vypočítate vhodného jadra, môžeme aplikovať Gaussovské vyhladenie s použitím štandardnej konvolučnej metódy
- $g(x) = \sum_{(m) \in O} h(x - m) f(m)$

Prahovanie neznámeho dokumentu 2

- Následne hľadáme dokument, ktorý je najpodobnejší nášmu
- To dosiahneme pomocou **metódy najbližšieho suseda**
- Použijeme euklidovskú vzdialenosť
- $\min_i d(P_1, P_i) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$
- Zistíme, aké mal nájdený dokument hodnoty posuvníkov a priradíme ich nášmu dokumentu

Hypotéza

- Očakávame že takto spracovaný dokument bude pre OCR systém lepšie rozpoznateľný a vhodnejší pre archiváciu „textovej informácie“
- Hypotézu si budeme vedieť overiť po ukončení experimentov

Výber literatúry

- [Br16] Bradley, D., Roth, G. 2016. *Adaptive Thresholding Using the Integral Image*. [online] Dátum platnosti 24. 10. 2019.
- [DA12] DAVIES, E. R. 2012. *Computer and Machine Vision: Theory, Algorithms, Practicalities* (Fourth Edition). [online] Dátum platnosti 12. 2. 2020.
- [KLE14] KLETTE, R. 2014. *Concise Computer Vision An Introduction into Theory and Algorithms*. [online] Dátum platnosti 12. 2. 2020.
- [Fi03] Fisher, R., Perkins, S., Walker, R., Wolfart, E. 2003. *Gaussian Smoothing* [online] <https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm> Dátum platnosti 25. 10. 2019.
- [Cha16] Chandrakala, M. 2016. *Quantitative analysis of local adaptive thresholding techniques*.
- [Chad16] Chandrakala, M. 2016. *Comparative study and image analysis of local adaptive thresholding techniques*
- [SKB] Shafait, F., Keysers, D., Breuel, T. M. rok. *Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images*. [online] Dátum platnosti 24. 10. 2019.

Ďakujem za pozornosť

