

Projekt do předmětu GMU zaměřený na OpenCL

URYCHLENÍ ZPRACOVÁNÍ OBRAZU ZALOŽENÉ NA MODIFIKACI HISTOGRAMU

Lukáš Piják, xpijak00
Filip Zapletal, xzapple27
Jan Vybíral, xvybir05

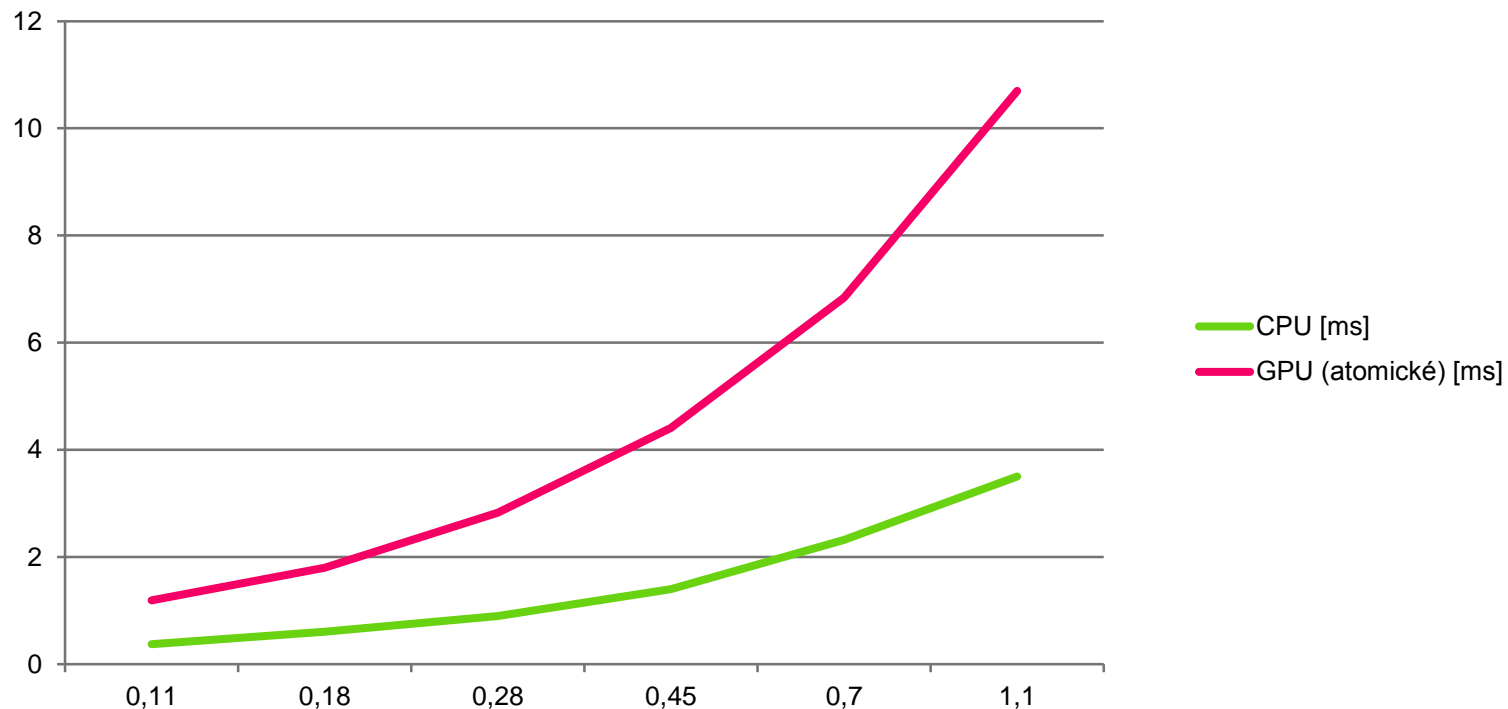
UPŘESNĚNÍ ZADÁNÍ

- Implementovat výpočet histogramu vstupního obrázku pomocí OpenCL.
- Každý člen týmu implementovat jednu metodu využívající histogramu, a to jak s pomocí OpenCL tak CPU implementací.
- Vstup aplikace zpracuje vždy oběma způsoby a vypíše čas každé z nich. Zvolené metody (algoritmy):
 - Ekvalizace histogramu
 - Prahování s pomocí metody Otsu
 - Segmentace obrazu na základě prahování histogramu



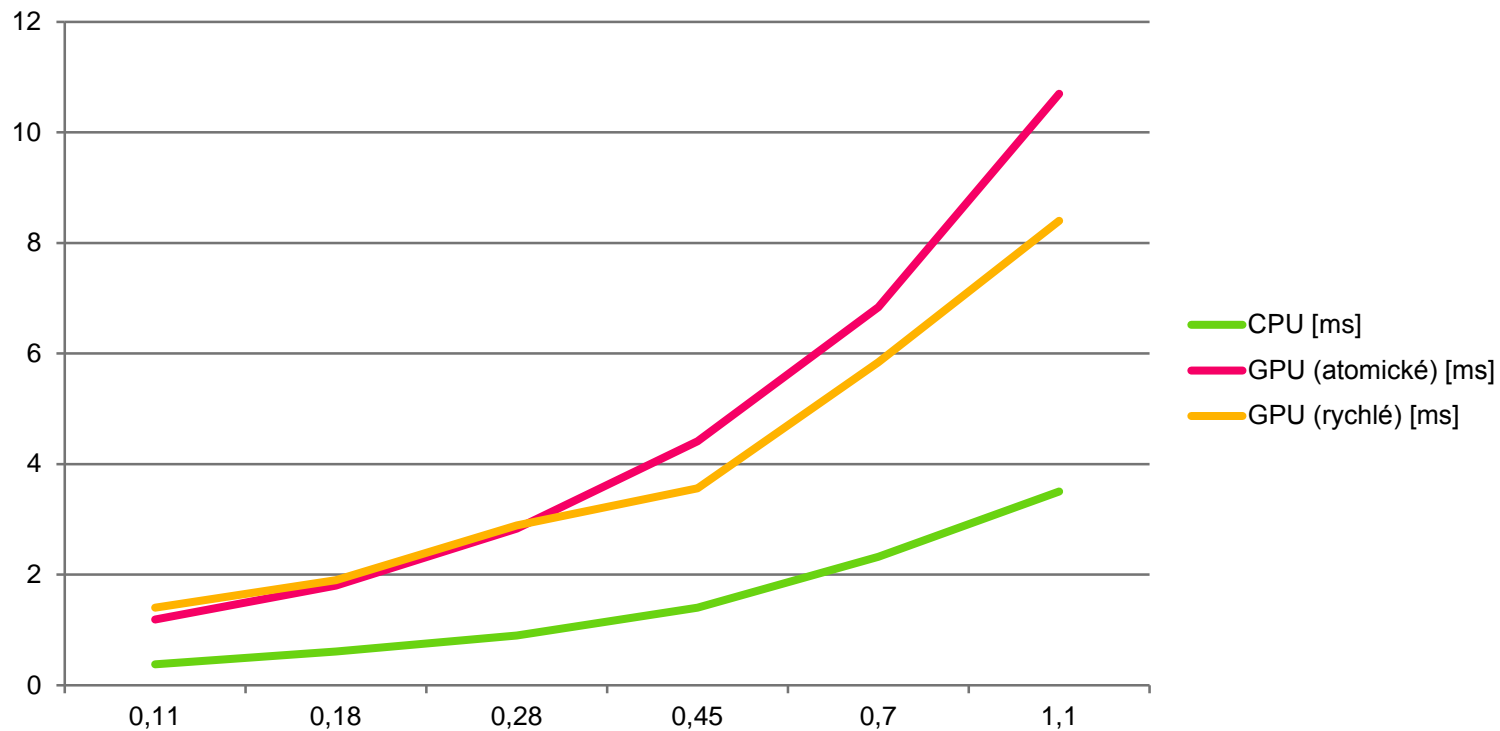
VÝPOČET HISTOGRAMU

- První metoda: počítání lokálních subhistogramů ve workgrupách, jejich následné sečtení, atomické instrukce



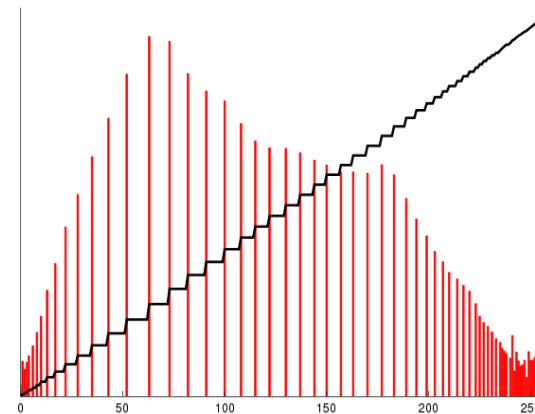
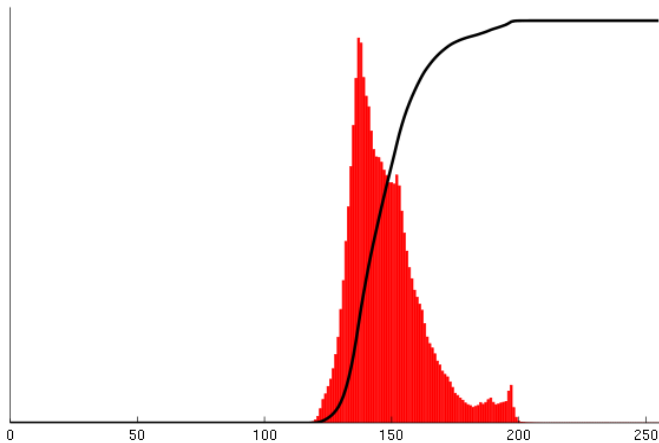
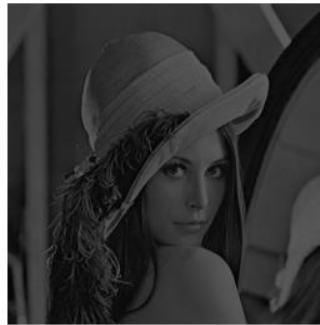
VÝPOČET HISTOGRAMU

- Druhá metoda: bez použití atomických instrukcí



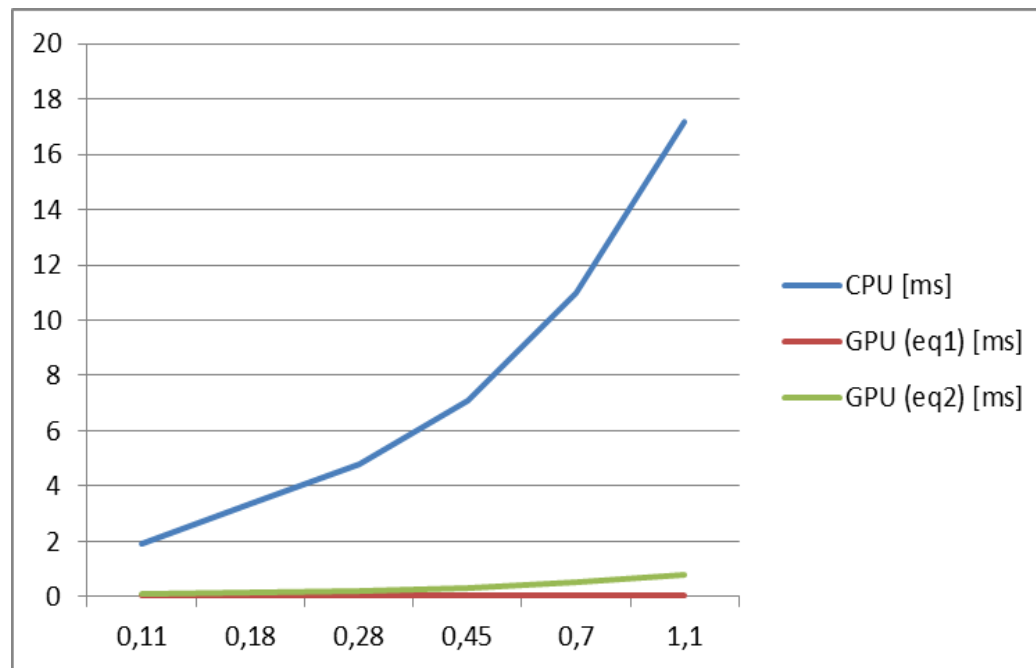
EKVALIZACE HISTOGRAMU

- Slouží k zvýšení kontrastu rovnoměrnějším rozložením intenzit



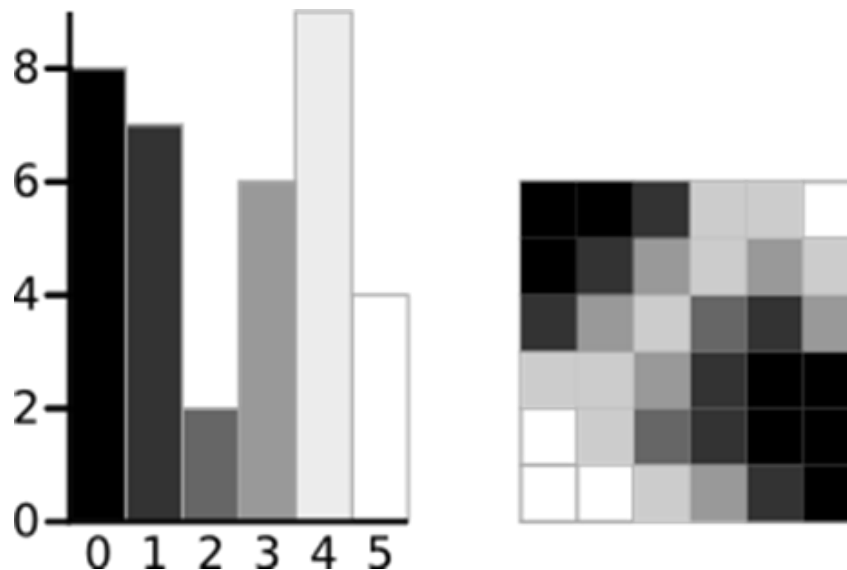
EKVALIZACE HISTOGRAMU

- Výpočet kumulativního histogramu
- Vynásobení jeho hodnot hodnotou (velikost histogramu / počet pixelů) – paralelně
- Tím získáme nové hodnoty pro jednotlivé původní hodnoty pixelů
- Přiřazení nových hodnot pixelům - paralelně

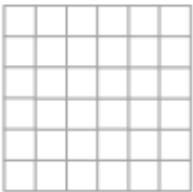
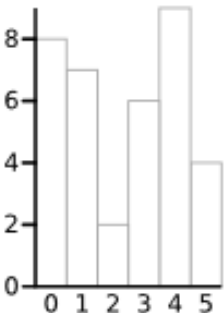

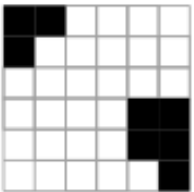
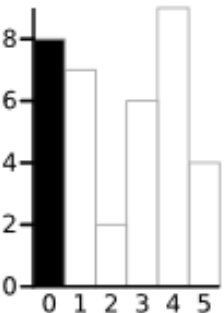

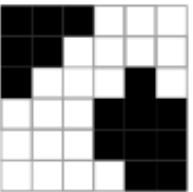
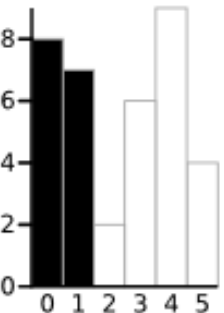

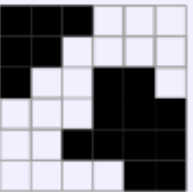
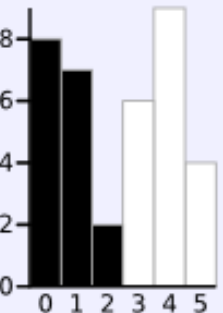

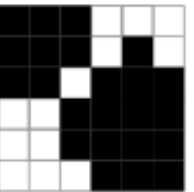
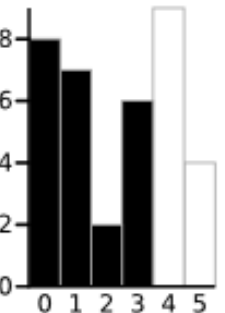

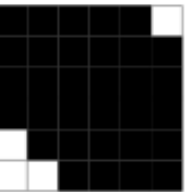
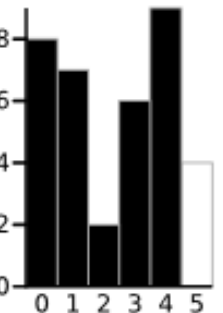



PRAHOVÁNÍ S POMOCÍ METODY OTSU

- Hledání optimálního prahu na základě výpočtu rozptylu.
- Cílem je najít takový práh, který maximalizuje mezirozptyl, nebo alternativně minimalizuje vnitřní rozptyl.

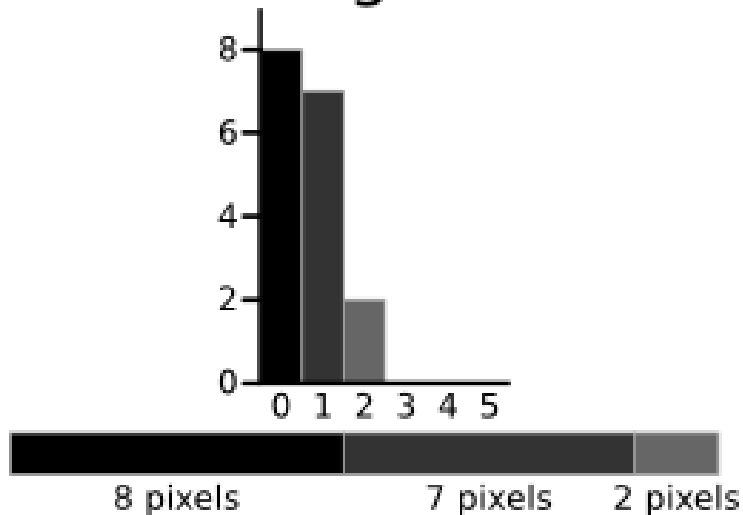


PRAHOVÁNÍ S POMOCÍ METODY OTSU

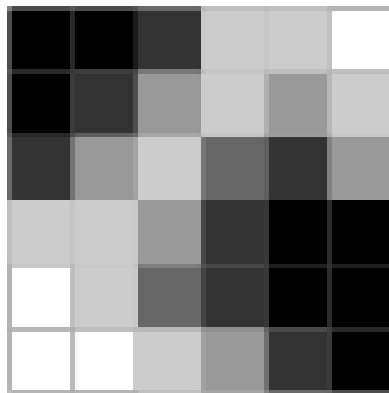
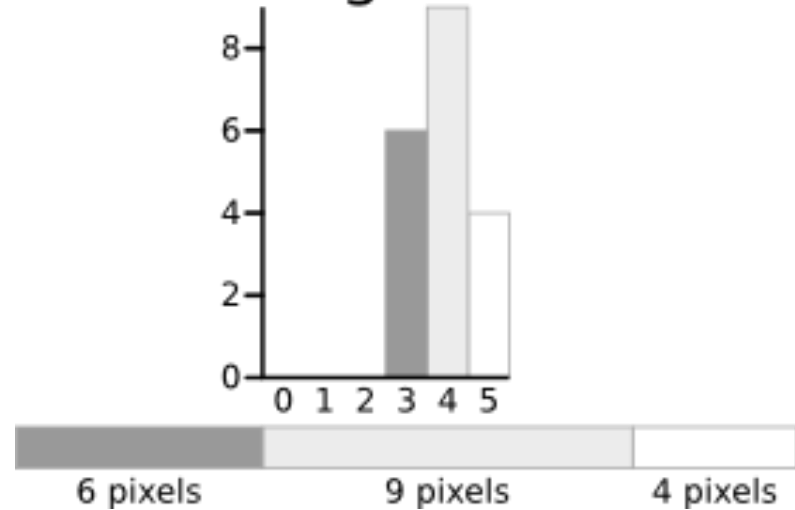
Threshold	T=0	T=1	T=2	T=3	T=4	T=5
	  	  	  	  	  	  
Weight, Background	$W_b = 0$	$W_b = 0.222$	$W_b = 0.4167$	$W_b = 0.4722$	$W_b = 0.6389$	$W_b = 0.8889$
Mean, Background	$M_b = 0$	$M_b = 0$	$M_b = 0.4667$	$M_b = 0.6471$	$M_b = 1.2609$	$M_b = 2.0313$
Variance, Background	$\sigma_b^2 = 0$	$\sigma_b^2 = 0$	$\sigma_b^2 = 0.2489$	$\sigma_b^2 = 0.4637$	$\sigma_b^2 = 1.4102$	$\sigma_b^2 = 2.5303$
Weight, Foreground	$W_f = 1$	$W_f = 0.7778$	$W_f = 0.5833$	$W_f = 0.5278$	$W_f = 0.3611$	$W_f = 0.1111$
Mean, Foreground	$M_f = 2.3611$	$M_f = 3.0357$	$M_f = 3.7143$	$M_f = 3.8947$	$M_f = 4.3077$	$M_f = 5.000$
Variance, Foreground	$\sigma_f^2 = 3.1196$	$\sigma_f^2 = 1.9639$	$\sigma_f^2 = 0.7755$	$\sigma_f^2 = 0.5152$	$\sigma_f^2 = 0.2130$	$\sigma_f^2 = 0$
Within Class Variance	$\sigma_W^2 = 3.1196$	$\sigma_W^2 = 1.5268$	$\sigma_W^2 = 0.5561$	$\sigma_W^2 = 0.4909$	$\sigma_W^2 = 0.9779$	$\sigma_W^2 = 2.2491$

PRAHOVÁNÍ S POMOCÍ METODY OTSU

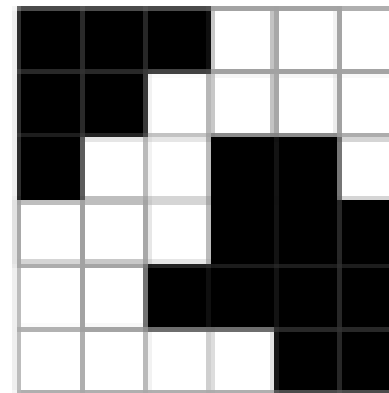
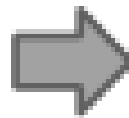
Background



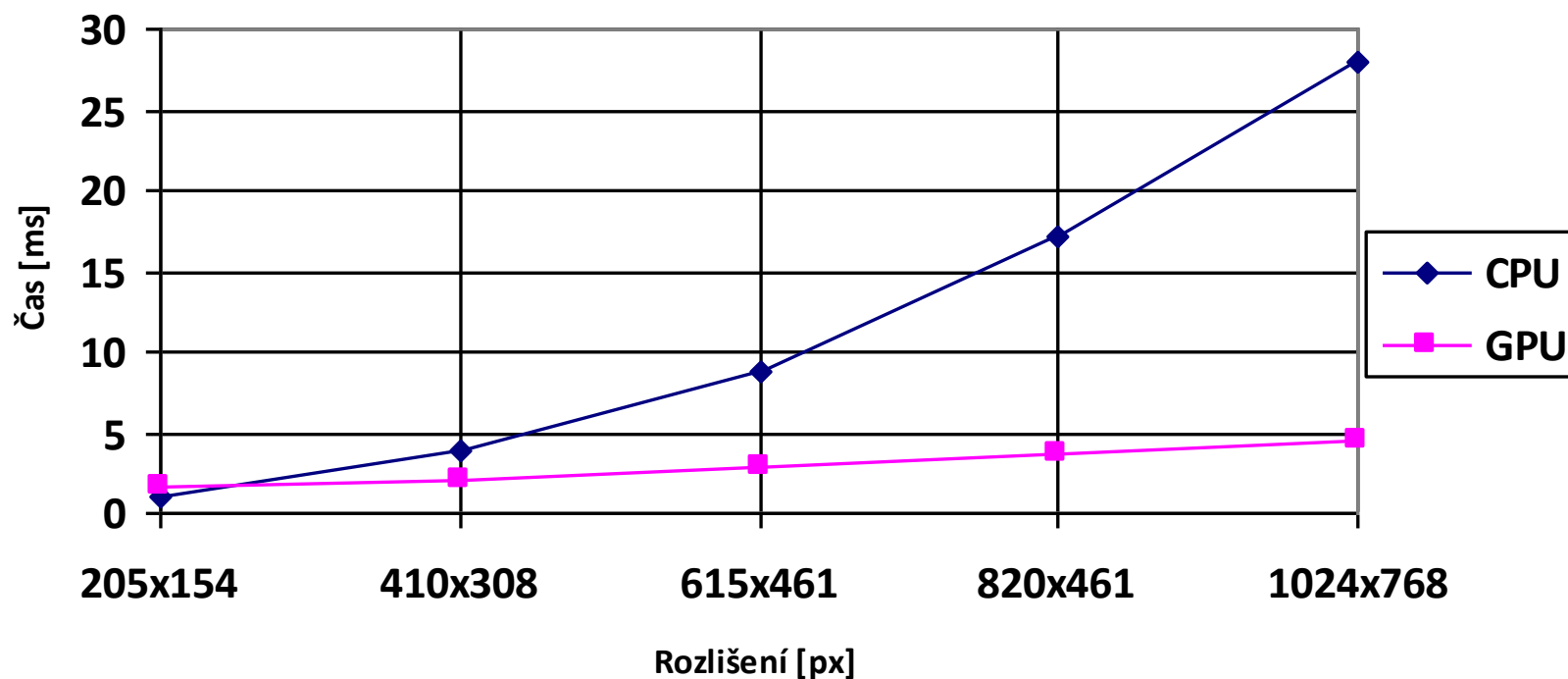
Foreground



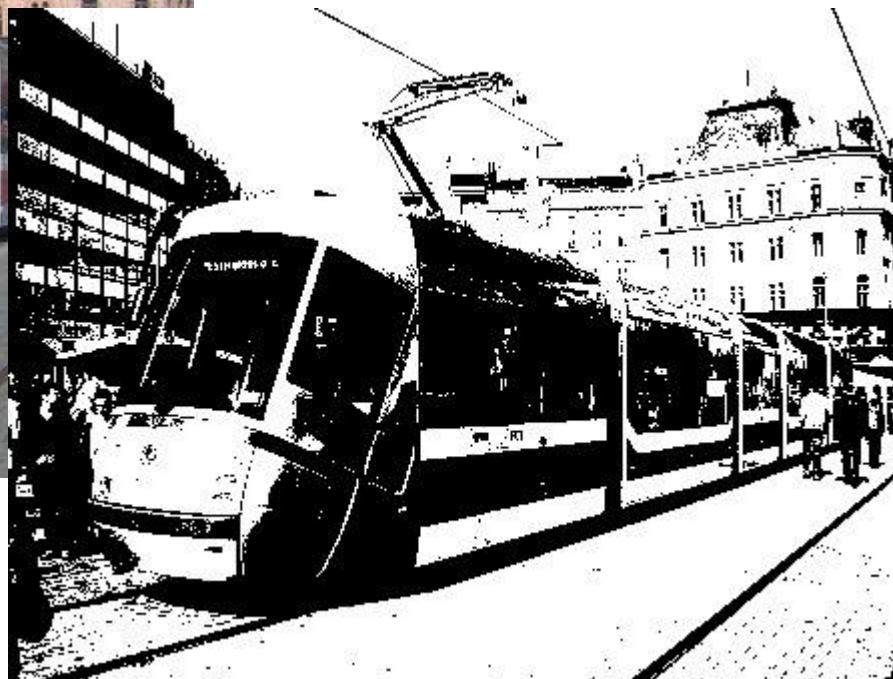
$T=3$



PRAHOVÁNÍ S POMOCÍ METODY OTSU



PRAHOVÁNÍ S POMOCÍ METODY OTSU



SEGMENTACE OBRAZU NA ZÁKLADĚ PRAHOVÁNÍ HISTOGRAMU

- rozdělení jednotlivých částí obrazu do několika podmnožin (popředí x pozadí)
- V histogramu hledáme práh
 - Leží mezi shluky podobných bodů
 - Hledáme jej iterativně – dělíme histogram na části podle aktuálního prahu a spočítáme střed mezi průměry hodnot – nový práh
- Zvolena adaptivní metoda
 - Práh je ovlivněn pouze malým (např, 30 x 30 px) okolím bodu



SEGMENTACE:

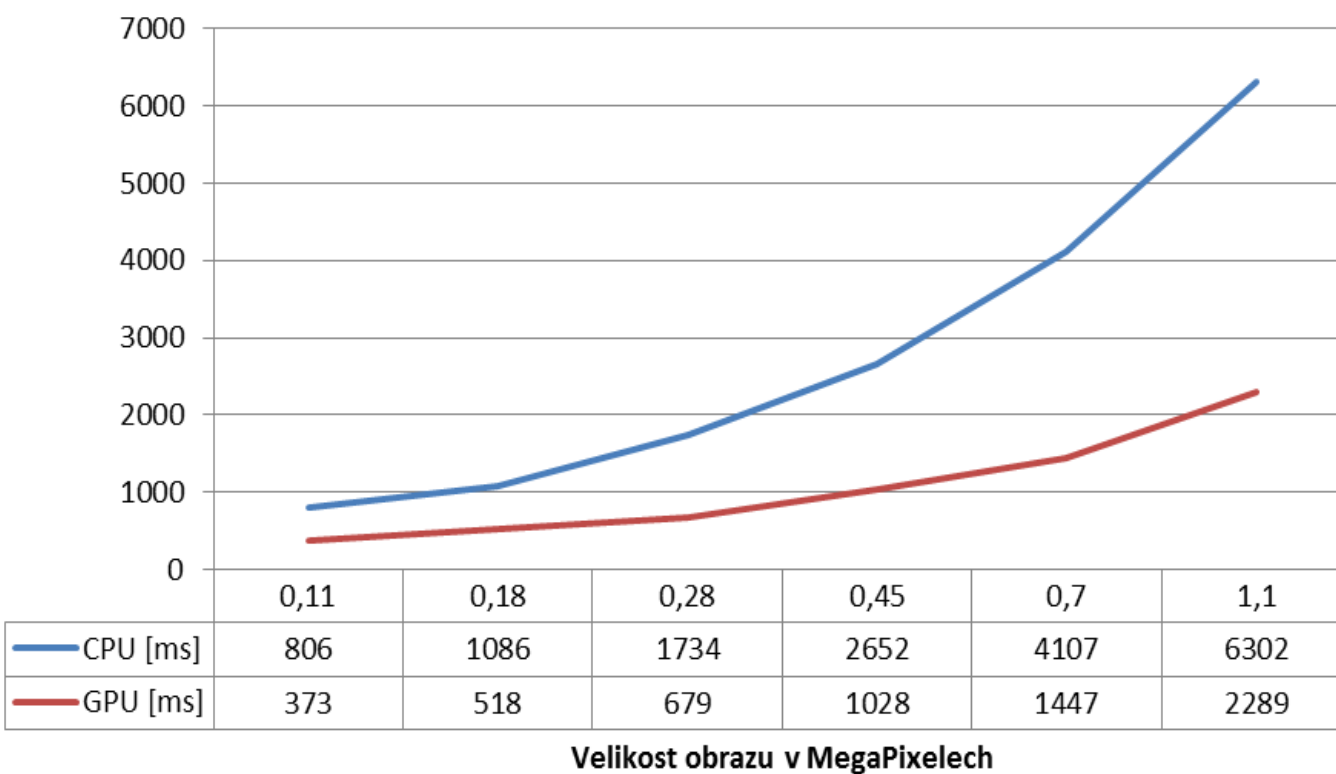
PŘIZPŮSOBENÍ ALGORITMU PRO BĚH NA GPU

- Práh se počítá pouze z malého okolí pro každý bod nezávisle
- Lze tedy výpočet spustit pro velké množství bodů zároveň
- Výsledná akcelpace na testovací stroji
 - CPU – Intel Core i7 2.44GHz; GPU NVIDIA GeForce GT620M
 - 2 – 3 násobná
 - Není tak výrazná
 - Výkonný CPU x Nevýkonná GPU (nízký výkon a počet shaderů)
 - Algoritmus obsahuje poměrně hodně řídicí logiky (cykly, podmínky), které GPU zvládá hůře

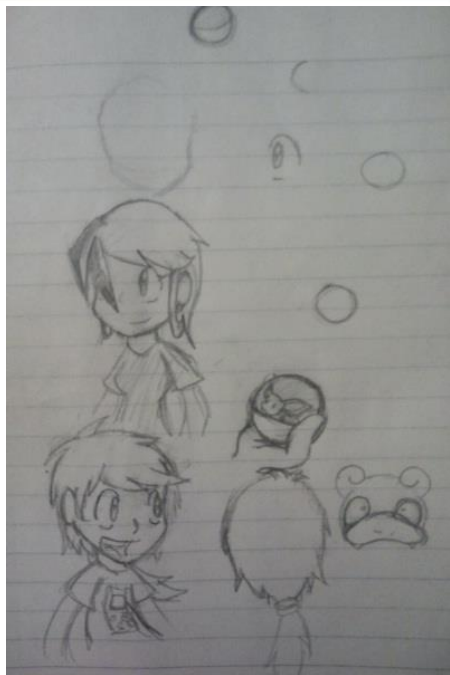


SEGMENTACE: VÝSLEDKY

Graf závislosti doby výpočtu na velikosti obrazu



SEGMENTACE: VÝSLEDKY



Zdroj:
avoision.com
paper-bob-omb.deviantart.com



POUŽITÁ LITERATURA

- L. Lucas, Image Segmentation. München : Technische Universität München, 2010.
- F. Kurugollu, B. Sankur, A.E. Harmanci. Color image segmentation using histogram multithresholding and fusion. 2001.
- Rajagopal, Venugopal. Image Segmentation by Histogram Thresholding. 2002.
- http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf
- <http://www.codeproject.com/Articles/38319/Famous-Otsu-Thresholding-in-C>
- <http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>
- Inspirace z projektu z HSC (FIT VUT Brno)
- <http://www.khronos.org/files/opencvcl-1-2-quick-reference-card.pdf>

