ZPO 2016/2017 – Projekt č. 10
Efektívna implementácia mediánu

Dominik Šimek (xsimek23)
Filip Lištiak (xlisti00)
26.04.2017

1 Zadanie

“Vypracujte funkci, která efektivně implementuje výpočet mediánového filtru o velikosti 3x3 pixely, případně 5x5 nebo 7x7 pixelů nad obrazem. Efektivní implementací se v tomto případě myslí taková implementace, která využije skutečnosti, že se medián počítá pro všechny pozice v obrazu a proto lze využít "klouzajícího okna", které mění pozici v každém kroku výpočtu jen například o jeden pixel vpravo. Proto pro řazení hodnot pixelů ve filtru lze provést tak, že se nejprve seřadí hodnoty ve sloupcích (okénka 3x3, 5x5 nebo 7x7) a poté se například metodou "list merge sort" seřadí hodnoty v celém okně. Pro pohybu okna o jeden pixel vpravo se mohou využít sloupce seřazené pro minulou polohu s tím, že přibude jen jeden sloupec. Implementujte v jazyce C nebo v assembleru. PZ”

2 Popis riešenia

Cieľom bolo je navrhnúť a implementovať funkciu (funkcie), ktorá realizuje mediánový filter. Pre tento účel bolo najskôr nutné implementovať výpočet mediánu z pohyblivého okna (veľkosti 3x3, 5x5, 7x7 pixelov) klasickým, konvenčným spôsobom. To znamená naplnenie pohyblivého okna hodnotami, jeho zoradenie (štandartným algoritmom) a vrátenie mediánu. Táto implementácia bola braná ako referenčná, voči ktorej bola porovnávaná implementácia popisovaná v zadaní (na úrovni korektnosti výpočtu a dosiahnutého zrýchlenia). Implementácia ktorá je cieľom projektu pracuje na princípe “cache” stĺpcov pohyblivého okna. Jednotlivé stĺpce okna sú najskôr zoradené (efektívnym algoritmom), po ich zlúčení je vrátená hodnota mediánu. Stĺpce ktoré už boli zoradené sa nieje potrebné radiť znova, sú použité hodnoty z “cache” stĺpcov (predchádzame tým opakovanému radeniu rovnakých hodnôt).

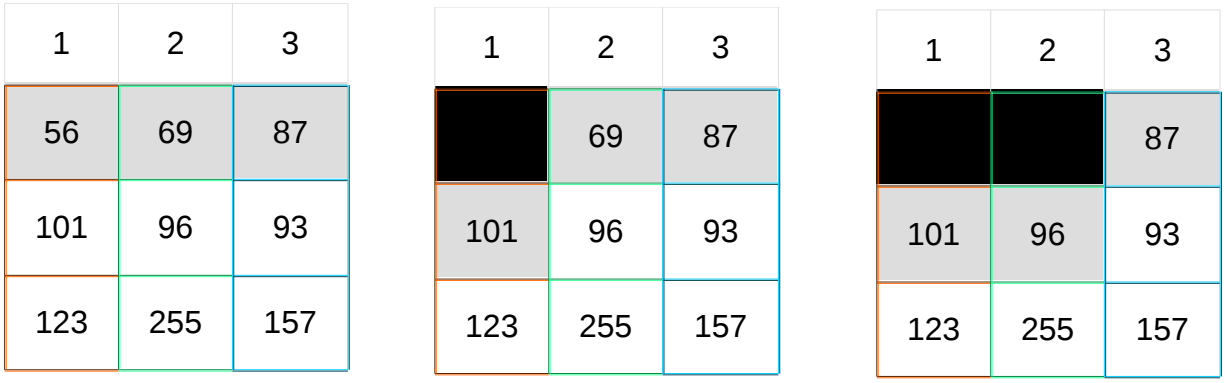
2.1 Efektívna implementácia mediánového filtru

1	2	3
101	69	93
56	255	87
123	96	157

Obrázok predstavuje pohyblivé okno rozdelené do troch stĺpcov. Mediánový filter zmení hodnotu stredného vyznačeného prvku podľa jeho okolia, ktoré sa zoradí a vyberie prostredný prvok – medián. Okno sa následne posunie o 1 krok doprava. Efektívna implementácia spočíva v tom, že sa stĺpce okna zoraďujú samostatne a následne prebehne ich merge. Keď sa okno posunie doprava tak stĺpce 2 a 3 už budú zoradené, stačí teda zoradiť len jeden a vykonať merge sort.

56	69	87	93	96	101	123	157	255

2.2 Merge sort



Do merge sortu vstupujú už zoradené stĺpce okna, ktorý z nich spraví jednu zoradenú postupnosť. Algoritmus pracuje na následovnom princípe:

- Vyberie sa prvý prvok z každého stĺpca a označíme ich šedou farbou.
- Z označených prvkov sa vyberie ten najmenší a zapíše sa do výstupného pola.
- V stĺpci, v ktorom sa vybral najmenší prvok sa označí šedou farbou nasledujúci prvok
- Postupne sa vyberá najmenší prvok z označených šedých prvkov až kým sa nezoradí celé okno

2.3 Technológie

Pre implementáciu projektu bol použitý jazyk C/C++. C++ je použité preto, aby bolo možné funkciu jednoducho testovať a porovnávať aj za použitia knižnice OpenCV. Samotná funkcia je imeplementovaná v jazyku C, ale ako vstup príjíma typ cv::Mat.

3 Popis obsluhy programu

Program sa prekladá pomocou makefilu príkazom make. Pre úspešnú kompiláciu je potrebné mať nainštalovaný c++11 kompiler a opencv. Príklady spustenia programu sú nasledovné:

- `./median -i input.png` (východzia veľkosť okna: 3x3, východzí výstup: output_filtered.bmp)
- `./median -i input.png -w 5 -o output.png`
- `./median -t` (spustenie unit testov, viac v sekcii 4.1)
- `./median -b` (spustenie benchmarku)

- `./median -h` (nápoveda)

Prepínač -w umožňuje zadať veľkosť okna mediánového filtru, ktorá ovplyvňuje jemnosť filtru.

4 Testovanie

Najviac času a úsilia bolo venované práve testovaniu. K tomuto účelu bol vytvorený benchmark test, ktorý porovnáva efektívnu a neefektívnu implementáciu mediánového filtru použitím rôznych testovacích dát opísaných v sekcii 4.2. Do testovania boli taktiež zahrnuté aj unit testy.

4.1 Unit testy

Pre kontrolu správnosti jednotlivých funkcií boli vytvorené unit testy. Testujú sa 3 hlavné oblasti:

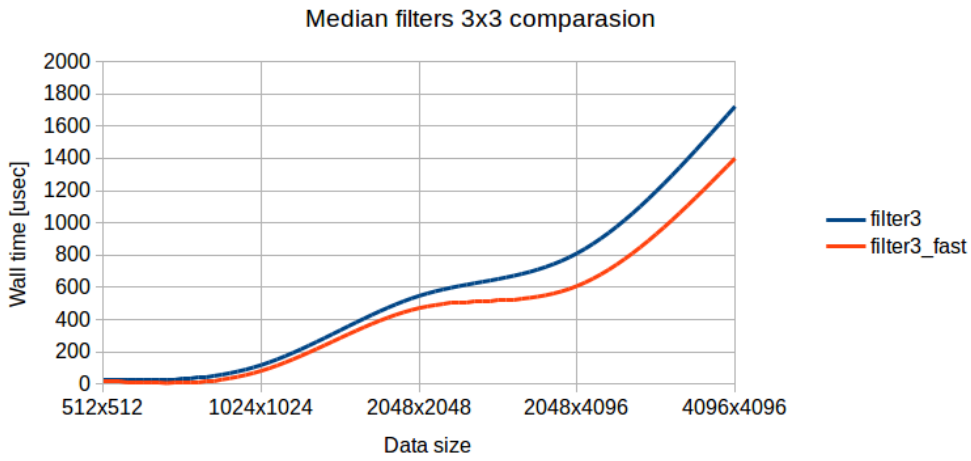
- Radenie
  - Testuje sa správne radenie rôzne dlhých postupností, predovšetkým tie, ktoré sa v programe využívajú najviac – 3, 5, 7.
- Merge
  - Testuje sa merge zoradených postupností o dĺžke 3, 5, 7 prvkov
- Medián
  - Testy celkového výpočtu mediánového filtru s veľkosťou okna 3, 5 alebo 7. Testy na výpočet mediánu sa vykonávajú vždy vo dvojici – jeden test prebieha na obyčajnej, neefektívnej verzii, druhý test využíva efektívnu implementáciu. Týmto spôsobom sa jednoducho vypočíta časový rozdiel medzi efektívnou a neefektívnou implementáciou. Výsledok mediánového filtru sa následne porovnáva s mediánovým filtrom v opencv.

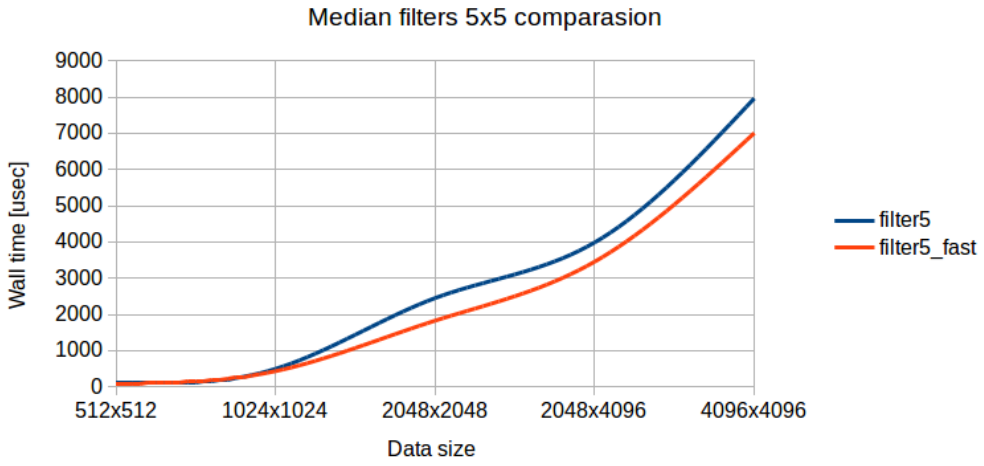
4.2 Testovacie dáta

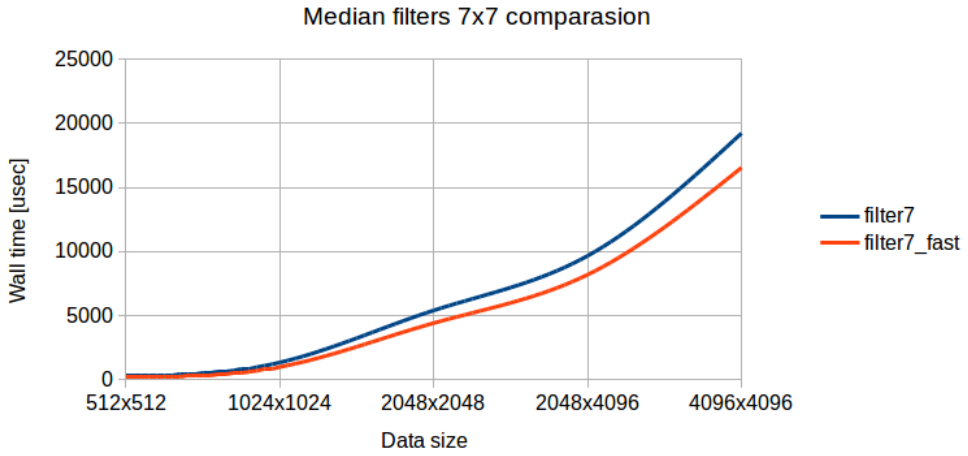
Pre testovacie účely si program vytvára matice rôznych veľkostí naplnené náhodnými číslami. Matice sa generujú s veľkosťami 512x512, 1024x1024 a 2048x4096. Program samozrejme prijíma aj obrázky so šumom zadané v príkazovej riadke. Pre najvýraznejší výsledok sa odporúčajú obrázky so šumom “salt and pepper”, ktorý dokáže mediánový filter perfektne vyhladiť.

5 Výsledky

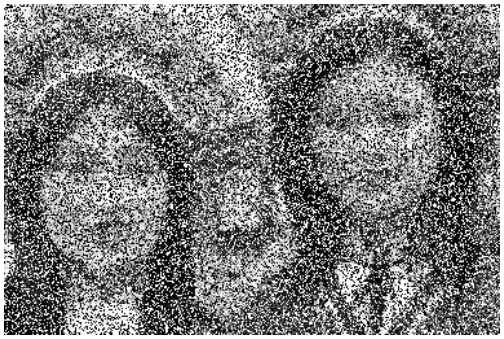
Nasledujúce grafy porovnávajú rýchlosť mediánového filtru medzi efektívnou a klasickou implementáciou.







pôvodný obrázok



okno o velkosti 3x3

okno o velkosti 5x5

okno o velkosti 7x7

6 Záver

Úspešne bol implementovaný efektívny mediánový filter. Na testovanie rýchlosti bol vytvorený benchmark, ktorý porovnáva efektívnu implementáciu a klasickú. Výsledky benchmarku sú znázornené v grafoch v kapitole 5. Pri väčších maticiach sa začína prehlbovať časový rozdiel medzi týmito dvomi implementáciami, čo je možné označiť ako úspech projektu. Zmena veľkosti okna dovoluje meniť jemnosť filtru, čo môžme vidieť na ukážkových obrázkoch v kapitole 5.

7 Literatúra

- Algoritmy radenia: https://en.wikipedia.org/wiki/Sorting_algorithm
- Radiace siete: https://en.wikipedia.org/wiki/Sorting_network
- Merge: http://www.geeksforgeeks.org/merge-k-sorted-arrays/
- Merge: http://stackoverflow.com/questions/39197996/merging-k-sorted-arrays-vectors-complexity
- Prednášky PRL, medián: https://www.fit.vutbr.cz/study/courses/PDA/private/www/h003.pdf
- Medián: http://www.stat.cmu.edu/~ryantibs/median/