

***Žilinská univerzita v Žiline***

***Fakulta riadenia a informatiky***

***Projekt 1***

***Inteligentná analýza obrazu***

***Spájanie dokumentov získaných kamerou***

*Patrik Hrmo*

*5ZU011*

*2016/2017*

[1. Úvod 3](#_Toc485311685)

[2. Analýza súčasného stavu 3](#_Toc485311686)

[3. Voľba vhodnej začiatočnej implementácie 3](#_Toc485311687)

[3.1. OpenCV 3](#_Toc485311688)

[3.2. BoofCV 5](#_Toc485311689)

[4. Počiatočný dataset 7](#_Toc485311690)

[5. Implementácia 7](#_Toc485311691)

[5.1. Výpočet veľkosti spojeného obrázka na základe homografie 7](#_Toc485311692)

[5.2. Spájanie viacerých snímok 8](#_Toc485311693)

[6. Pokročilý dataset 8](#_Toc485311694)

[6.1. Štruktúra datasetu 9](#_Toc485311695)

[6.2. Popis datasetov 9](#_Toc485311696)

[6.3. Vyhodnocovanie 9](#_Toc485311697)

[7. Testy 9](#_Toc485311698)

[7.1. Zisťovanie vhodných parametrov 9](#_Toc485311699)

[7.2. Rektifikácia vstupných snímok 13](#_Toc485311700)

[8. Bibliografia 14](#_Toc485311701)

# Úvod

V tejto práci sa budem venovať spájaniu snímok textových dokumentov. Tieto snímky môžu byť vytvárané mobilným zariadením, takže budú podliehať perspektívnej transformácií. Výsledkom bude teda obrázok dokumentu poskladaný z týchto snímok tak, aby ho bolo možné prečítať či už ľudským okom, alebo nástrojom OCR.

# Analýza súčasného stavu

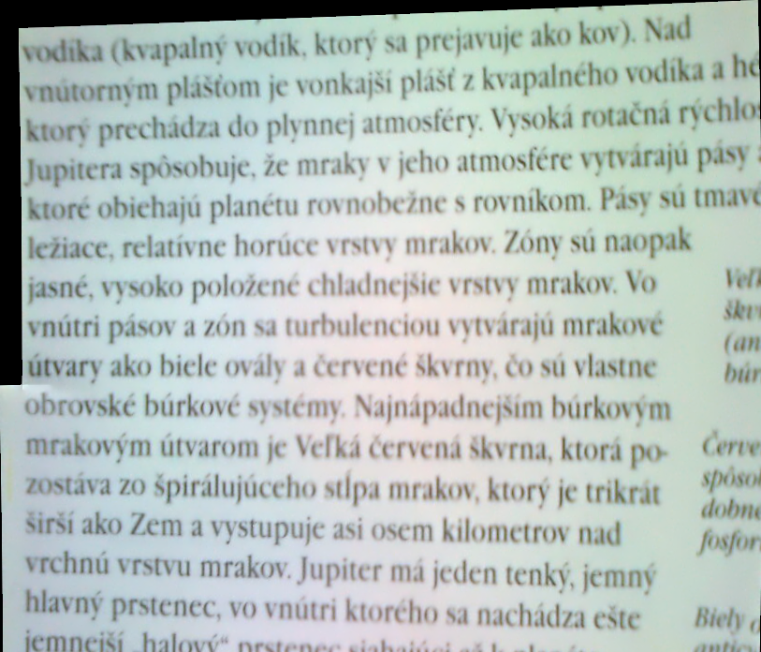
K vyriešeniu tohto problému je napísaných mnoho článkov a kníh. Tieto sa však často venujú spájaniu dokumentov na základe textových vlastností (1) ako napríklad tok textu. V tejto práci sa však sústredím na úpravu algoritmov pre spájanie panorám (2) tak, aby boli funkčné aj na textové dokumenty. Algoritmy na spájanie panorám sú veľmi často využívané, a práve z tohto dôvodu bývajú implementované v rôznych frameworkoch s počítačovým videním.

# Voľba vhodnej začiatočnej implementácie

Ako prvé je teda potrebné nájsť vhodný framework pre prácu s obrazom. Najlepšie taký v ktorom je funkčná implementácia spájania panorám.

## OpenCV

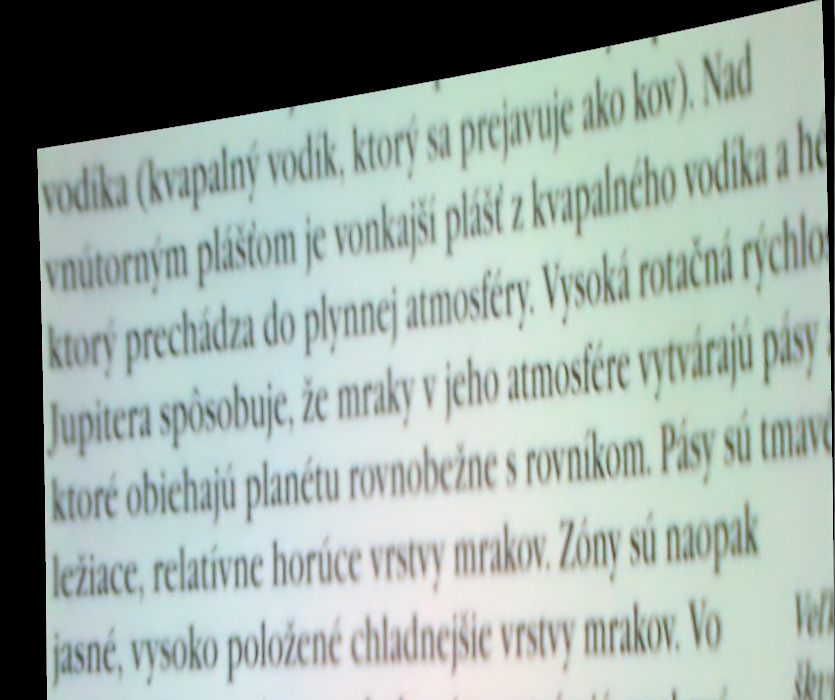
Jeden z najznámejších frameworkov na prácu s počítačovým videním je OpenCV. OpenCV obsahuje triedu Stitcher, ktorá vie spájať obrázky do panorám. Tento algoritmus vie aj dobre spájať snímky textových dokumentov.



Obrázok spojenie textových dokumentov v Opencv

Obrázok 2 spojenie textových dokumentov v OpenCV

Problém s touto implementáciou nastáva ak sa pokúsime spojiť snímky ktoré sú pod vplyvom rotácií. V tomto prípade algoritmus zlyhá alebo vráti veľmy zlý výsledok.

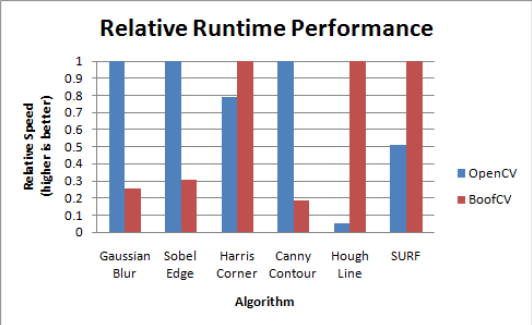


Obrázok 3 zlé spojenie dokumentov v OpenCV

Je teda nutné upraviť tento algoritmus. S týmto sú však spojené problémy ohľadom dokumentácie, ktorá je pri triedach a metódach spájania zanedbaná. Pre lepšiu produktivitu je teda potrebné nájsť iný framework.

## BoofCV

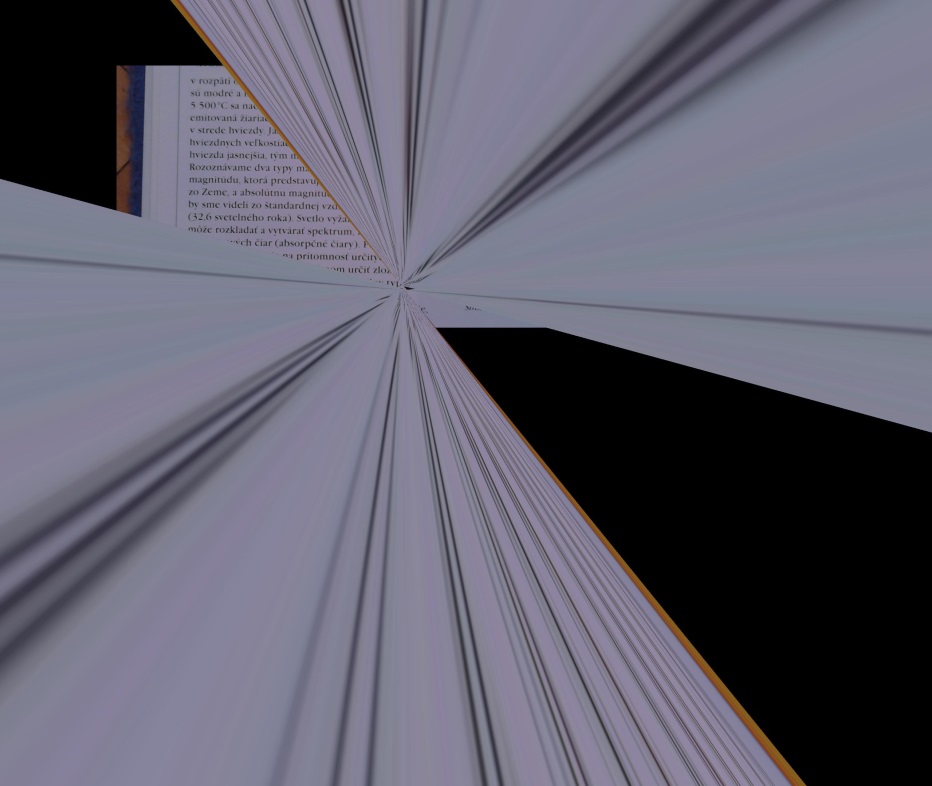
BoofCV je pomerne nový framework ktorý je napísaný čisto v jave. Z toho samozrejme vyplíva že operácie na nízkej úrovni sú pomalšie ako v OpenCV. Pre tento projekt je však dôležitá hlavne rýchlosť detekovania a popisu *feature points* snímok. Tu je však BoofCV približne 2x rýchlejší ako OpenCV.



Obrázok 4 porovnanie rýchlostí OpenCV a BoofCV (3)

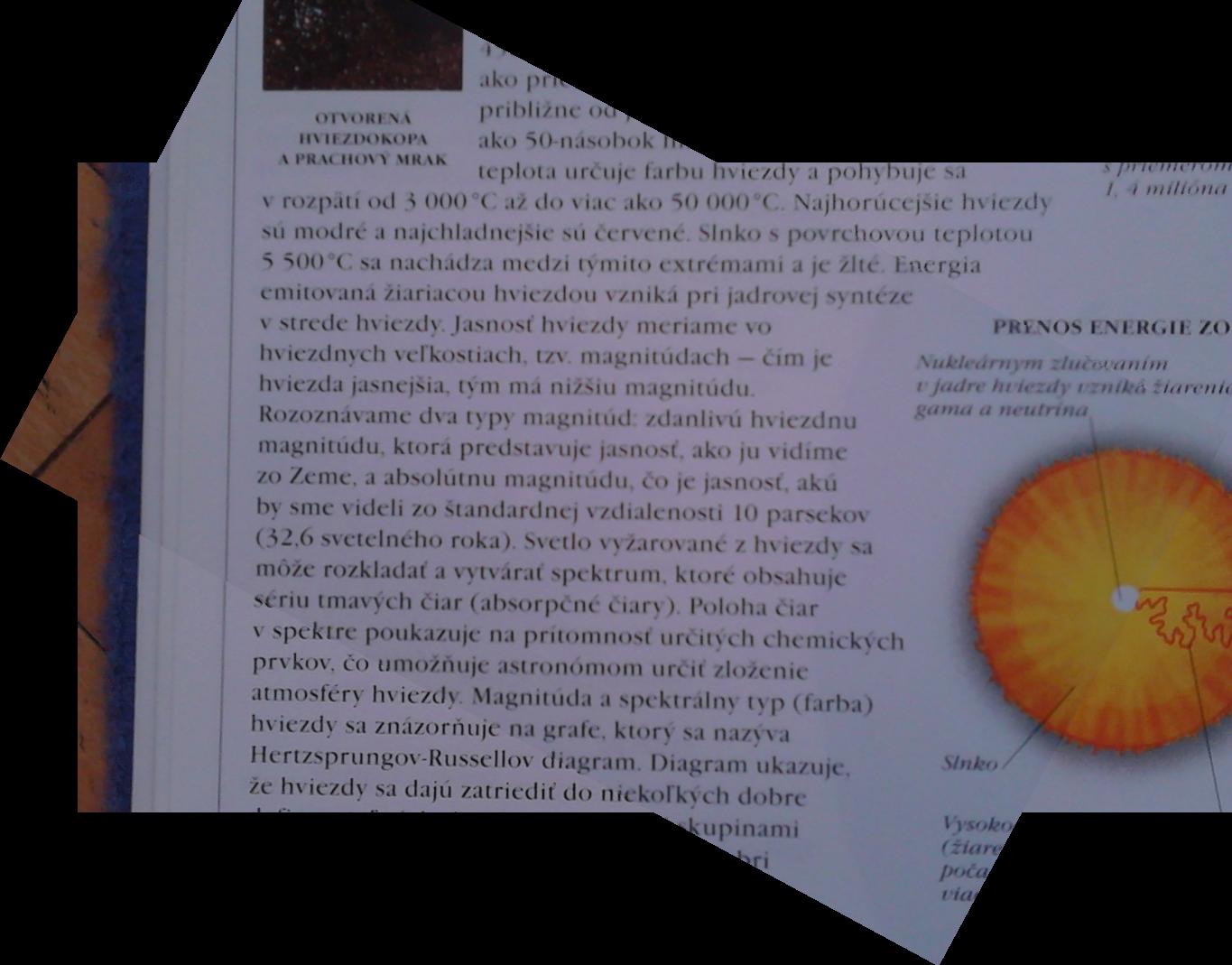
Nakoľko má táto knižnica aj dobrú dokumentáciu, zvolili sme si ju na ďalšie použitie.

BoofCV neposkytuje defaultne žiadnu triedu na spájanie obrázkov, avšak medzi príkladmy využitia je aj jednoduchý algoritmus spájania obrázkov. Tento príkladný algoritmus je však nevhodný pre spájanie textových dokumentov.



Obrázok 5 zlé spojenie textového dokumentu v BoofCV

Analýzov som zistil že toto spojenie zlyhalo kôli nedostatočnému počtu feature pointov. Tak isto som zmenil konfiguráciu RANSACU ktorý sa v tomto algoritme použiva na určenie homografie medzi obrázkami.



Obrázok 6 spojenie v BoofCV po drobných zmenách

# Počiatočný dataset

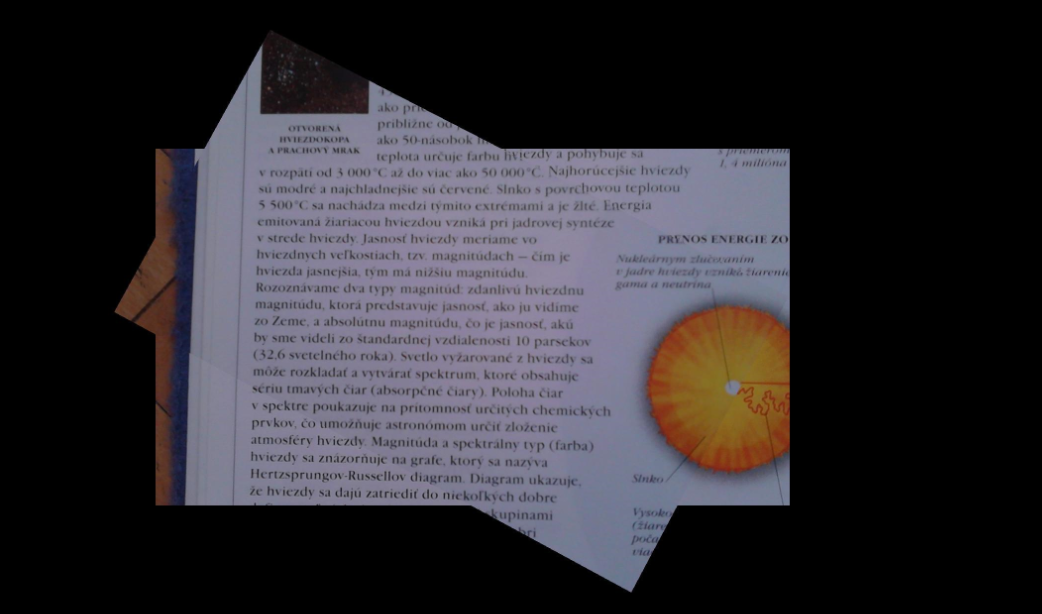
Keď už máme základnú funkčnosť algoritmu implementovanú, je dobré si vytvoriť dataset ktorý bude slúžiť pre budúce testy. Dataset je tvorený priečinkami, kde každý priečinok obsahuje dve snímky – jednu bez transformácie(budú sa k nej pripájať ostatné snímky a teda výsledný obrázok bude tiež bez transformácie. Tým pádom je na výslednom obrázku možné spúšťať OCR ktoré je použité pri vyhodnocovaní testov) a druhú s transformáciou. Takto môžme ľahko sledovať úspešnosť spájania pri rôznych transformáciách. Bol vytvorený aj testovací program, ktorý prejde a pospája všetky snímky v príslušných priečinkoch. To nám urýchli testovanie nakoľko nebude nutné meniť názvy snímok a priečinkov v kóde.

# Implementácia

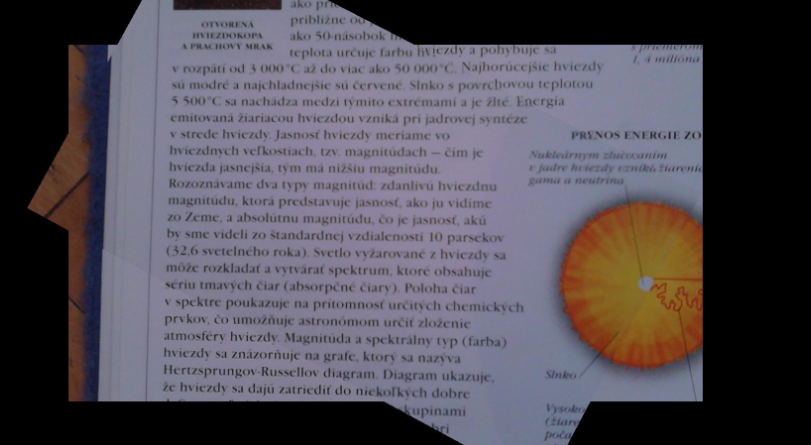
Teraz môžme pristúpiť k pokročelejším úpravám a implementácií chýbajúcich častí algoritmu.

## Výpočet veľkosti spojeného obrázka na základe homografie

Je potrebné upraviť veľkosť obrázka ktorý vznikne po spájaní. Jeho veľkosť je momentálne určená konštantne, čo spôsobuje zbytočné preplnenie pamäte. V horšom prípade by táto vlastnosť mohla zapríčiniť nemožnosť spojenia tohto obrázka s ostatnými, nakoľko sa časti tohto obrázka nemusia nachádzať vo vykreslenej zóne.



Obrázok 7 ukážka voľného vyplitvaného priestoru



Obrázok 8 ukážka orezania časti pospájaného obrázka

Za pomoci homografie je možné vypočítať presnú veľkosť výsledného obrázka. Ako prvé je potrebné určiť si body reprezentujúce rohy obrázkov ktoré spájame. U základného obrázka je to ľahké nakoľko nebude podliehať žiadnej transformácií (jeho body budú teda [0,0],[0,sirka], ...). U druhého obrázka je to zložitejšie – treba vypočítať ich súradnice po transformácií. Keď už máme vypočítaných všetkých osem bodov, je úloha nájdenia veľkosti výsledného obrázka triviálna(odčítanie y súradnice najvrchnejšieho bodu od najspodnejšieho, analogcky pre najľavejší a najpravjší bod).

## Spájanie viacerých snímok

Keď vieme spájať dve snímky s prijateľnými výsledkami, môžme pristúpiť k rozšíreniu algoritmu o viac ako dve snímky. Algoritmus spájania je nasledovný:

1. Zvoľ prvý snímok za hlavný – na tento obrázok sa budú pripájať ostatné obrázky. Ostatné snímky zaraď do *zoznamu doposiaľ nespojených snímkach*.
2. Detekuj feature pointy na všetkých snímkach v *zozname doposiaľ nespojených snímkach*.
3. Ak je *zoznam nespojených snímok* prázdny KONIEC. Výsledný pospájaný obrázok je v hlavnom obrázku
4. Detekuj feature pointy na hlavnom snímku.
5. Zo zoznamu doposiaľ nespojených snímok vyber snímku, ktorá má najviac zhôd s hlavným snímkom.
6. Spoj túto snímku s hlavnou snímkou, choď na krok 3.

# Pokročilý dataset

Pre nové testy je potrebné vytvoriť dataset ktorý bude pokrývať širšiu škálu rôznych vzťahov medzi snímkami. Potrebné je pozorovať ako sa bude bude kvalita spojenia správať pod rôznymi rotáciami a sklonmy kameri.

## Štruktúra datasetu

Štruktúra datasetu je podobná ako pri klasických súboroch – v stromovej štruktúre. Je tu však zopár obmedzení. Samotné snímky ktoré budeme zlučovať sa nachádazjú iba v listoch tejto štruktúry. V každom liste sa teda musia nachádzať snímky ktorých spojenie ideme testovať. Okrem nich sa tu musí nachádzať aj súbor s názvom grandTruth.jpg(ideálna snímka) alebo grandTruth.jpg.txt(textový súbor s grandTruth textom).

## Popis datasetov

1 – veľké prekrytie žiadna rotácia malý sklon

2 – veľké prekrytie veľká rotácia malý sklon

3- stredné prekrytie žiadna rotácia žiadny sklon

5 – strednéprekerytie žiadna rotácia malý sklon

6 – stredné prekrytie stredná rotácia žiadny sklon

7 – stredné prekerytie stredná rotacia malý sklon

8 – stredné prekerytie malá rotácia žiadny sklon

9 – malé prekrytie žiadna rotácia žiadny sklon

10 – malé prekrytie žiadna rotácia malý sklon

11 – malé prekrytie stredná rotácia žiadny sklon

12 - malé prekrytie malá rotácia žiadny sklon

## Vyhodnocovanie

Kvalitu spojenia posudzujeme porovnaním textu, ktorý získame pomocou nástroja OCR Tesseract (4) z popájaných snímok. Za grandTruth text budeme považovať text získaný zo súboru grandTruth.jpg.txt. V tomto súbore sa nachádza text ktorý sa nachádza v dokumente na ktorom sú testované spoje.

S týmto prístupom si však do testov zavádzame istú mieru chyby, nakoľko nástroj OCR nie je dokonalý – aj pri dokonalom spojení sa môže stať že poronávané texty sa nebudú zhodovať(Napr. OCR prečíta namiesto písmenka o číslo 0). Z tohto dôvodu je pri každej tabuľke uvedený údaj v ľavom hornom rohu reprezentujúci percentuálnu zhodu skutočného textu s textom získaným OCR na ideálnej snímke.

### Porovnávanie textov

Jednotlivé texty sú porovnávané na základe Levenštejnovej vzdialenosti. Levenštejnova vzdialenosť medzi textom t1 a textom t2 je počet vkladaní, mazaní a výmen znakov ktoré musíme urobiť na texte t1 aby sme dostali text t2. (Napríklad vzdialenonsť medzi textami „strom“ a „dom“ je rovná 3 – vymažeme „s“ následne „t“ a vymeníme „r“ za „d“).

Avšak vzdialenosť sama o sebe nie je vhodným ukazovateľom ako sú texty zhodné ( levenštejnova vzdialenost rovná 3 je výborná ak porovnávame romány, zlá ak porovnávame iba slová ). Preto je potreba ju normalizovať. Normalizácia je jednoduchá – stačí vydeliť vzdialenosť dĺžkou dlhšieho porovnávaného textu a výsledkok odčítať od 1. Touto normalizáciou dostaneme číslo z intervalu <0,1> kde 0 znamená že texty sú úplne odlišné a 1 - texty sú zhodné. Toto číslo nazveme nromalizovanou podobnosťou.

V tabuľkách sú uvedené percetnuálne hodnoty zhody textov – normalizovaná podobnosť vynásobená číslom 100.

# Testy

Parametrov algoritmu spájania viacerých snímok je 11, a nebude možné nájsť optimálne parametre kombinovaním rôznych hodnôt týchto parametrov.

## Zisťovanie vhodných parametrov

Pre začiatok sa zameriame na počet iterácií RANSACu.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Gt: 96.27% | Ransac 1500 | | Ransac 1250 | | Ransac 1000 | | Ransac 750 | | Ransac 500 | | Ransac 250 | |
| dataset | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas [ms] |
| celkom | 84.64 | 86909 | 84.38 | 87469 | 84.17 | 86955 | 83.04 | 86940 | 82.54 | 86830 | 81.07 | 87346 |
| 1 | 92.21 | 9157 | 92.21 | 9235 | 92.21 | 9095 | 92.21 | 9017 | 92.21 | 9048 | 92.21 | 8985 |
| 2 | 85.11 | 8377 | 85.11 | 8408 | 85.11 | 8377 | 85.11 | 8424 | 82.76 | 8377 | 82.76 | 8393 |
| 3 | 95.68 | 7722 | 95.68 | 7784 | 94.41 | 7707 | 94.41 | 7722 | 94.41 | 7722 | 95.51 | 7675 |
| 4 | 80.95 | 7925 | 80.95 | 7894 | 80.95 | 7956 | 80.95 | 7863 | 80.95 | 7878 | 81.50 | 7862 |
| 5 | 82.46 | 8471 | 82.46 | 8643 | 81.42 | 8548 | 62.81 | 8549 | 61.63 | 8502 | 75.70 | 8643 |
| 6 | 69.88 | 7878 | 67.08 | 7847 | 67.08 | 7800 | 71.18 | 7831 | 71.18 | 7753 | 71.18 | 7784 |
| 7 | 93.24 | 8144 | 94.15 | 8236 | 94.15 | 8206 | 94.15 | 8206 | 92.15 | 8190 | 92.15 | 8144 |
| 8 | 84.27 | 7114 | 83.28 | 7161 | 83.28 | 7098 | 83.28 | 7145 | 83.28 | 7098 | 84.31 | 7052 |
| 9 | 85.87 | 6864 | 85.87 | 6973 | 85.87 | 6958 | 87.93 | 6942 | 87.93 | 7052 | 84.26 | 6973 |
| 10 | 85.02 | 7769 | 85.02 | 7831 | 85.02 | 7831 | 85.02 | 7769 | 85.02 | 7815 | 68.49 | 8222 |
| 11 | 76.38 | 7488 | 76.38 | 7457 | 76.38 | 7379 | 76.38 | 7472 | 76.38 | 7395 | 63.70 | 7613 |

Tabuľka 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Gt: 96.27% | Ransac 3000 | | Ransac 5000 | | Ransac 10 000 | | Ransac 1000 000 | |
| dataset | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] |
| celkom | 84.84 | 90778 | 81.51 | 90708 | 81.31 | 93324 | 81.16 | 571861 |
| 1 | 92.21 | 10012 | 92.21 | 9532 | 92.21 | 9801 | 91.67 | 60854 |
| 2 | 85.08 | 8880 | 85.08 | 9124 | 84.45 | 8863 | 84.96 | 41652 |
| 3 | 96.27 | 8018 | 96.27 | 8127 | 94.74 | 8414 | 95.03 | 60323 |
| 4 | 80.92 | 8125 | 82.01 | 8204 | 81.24 | 8462 | 78.75 | 58547 |
| 5 | 82.37 | 8796 | 83.63 | 8826 | 83.24 | 9056 | 82.55 | 44218 |
| 6 | 69.70 | 8109 | 66.91 | 8103 | 66.91 | 8428 | 68.23 | 48829 |
| 7 | 93.88 | 8476 | 93.32 | 8625 | 93.32 | 8827 | 92.86 | 72497 |
| 8 | 85.59 | 7315 | 84.24 | 7402 | 84.47 | 7499 | 85.22 | 54799 |
| 9 | 85.85 | 7209 | 84.42 | 7257 | 84.47 | 7399 | 86.78 | 50124 |
| 10 | 85.02 | 8113 | 52.16 | 7869 | 53.96 | 7955 | 52.26 | 38493 |
| 11 | 76.35 | 7725 | 76.35 | 7639 | 75.36 | 8622 | 74.51 | 41398 |

Tabuľka

Z testov (Tabuľka 1 a Tabuľka 2) je zrejmé že čím väčší je počet iterácií RANSACu tým lepšie bude výsledné spojenie. Avšak pri extrémnych hodnotách (nad 5000) už nezískavame žiadne významné zlepšenie. V ďalších testoch budeme využívať počet iterácií 3000, čo je dobrý kompromis medzi kvalitou spojenia a časom potrebným na spojenie.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Gt: 96.27% | Ransac 3000  Feature 5000  Inlier treshold 1 - default | | Ransac 3000  Feature 5000  Inlier treshold 2 | | Ransac 3000  Feature 5000  Inlier treshold 3 | |
| dataset | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] |
| celkom | 84.84 | 90778 | 84.71 | 88187 | 81.68 | 88906 |
| 1 | 92.21 | 10012 | 92.70 | 9251 | 93.08 | 9937 |
| 2 | 85.08 | 8880 | 82.16 | 8487 | 85.08 | 8518 |
| 3 | 96.27 | 8018 | 95.65 | 8174 | 95.68 | 7925 |
| 4 | 80.92 | 8125 | 82.68 | 7971 | 82.68 | 8034 |
| 5 | 82.37 | 8796 | 82.37 | 8783 | 82.37 | 8674 |
| 6 | 69.70 | 8109 | 69.70 | 7878 | 69.70 | 7894 |
| 7 | 93.88 | 8476 | 94.53 | 8269 | 94.53 | 8424 |
| 8 | 85.59 | 7315 | 84.74 | 7129 | 84.74 | 7301 |
| 9 | 85.85 | 7209 | 85.02 | 7831 | 84.97 | 6989 |
| 10 | 85.02 | 8113 | 76.35 | 7441 | 49.29 | 7675 |
| 11 | 76.35 | 7725 | 75.36 | 8622 | 76.35 | 7535 |

Tabuľka 3

V tomto teste (Tabuľka 3) sme pozorovali ako vplýva inlier treshold v RANSACU na spájanie. Zistili sme že že bude najlepšie ponechať defautlnú hodnotu 1 pixel.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Gt: 96.27% | Scorer 0.1 | | Scorer 0.3 | | Scorer 0.5 | | Scorer 0.7 | | Scorer 2 | | Scorer max | |
| dataset | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas [ms] |
| celkom | 81.93 | 88684 | 84.84 | 87490 | 84.84 | 87562 | 84.84 | 87171 | 84.84 | 86830 | 84.84 | 87346 |
| 1 | 93.05 | 10030 | 92.21 | 9303 | 92.21 | 9408 | 92.21 | 9266 | 92.21 | 9235 | 92.21 | 8985 |
| 2 | 85.01 | 8486 | 85.08 | 8611 | 85.08 | 8408 | 85.08 | 8392 | 82.76 | 8377 | 82.76 | 8393 |
| 3 | 94.66 | 7800 | 96.27 | 7707 | 96.27 | 7894 | 96.27 | 7768 | 94.41 | 7722 | 95.51 | 7675 |
| 4 | 84.83 | 9094 | 80.92 | 7800 | 80.92 | 7831 | 80.92 | 7863 | 80.95 | 7878 | 81.50 | 7862 |
| 5 | 71.11 | 7831 | 82.37 | 8611 | 82.37 | 8611 | 82.37 | 8596 | 61.63 | 8502 | 75.70 | 8643 |
| 6 | 93.71 | 8253 | 69.70 | 7846 | 69.70 | 7846 | 69.70 | 7706 | 71.18 | 7753 | 71.18 | 7784 |
| 7 | 84.33 | 7223 | 93.88 | 8252 | 93.88 | 8252 | 93.88 | 8221 | 92.15 | 8190 | 92.15 | 8144 |
| 8 | 86.05 | 7036 | 85.59 | 7191 | 85.59 | 7176 | 85.59 | 7114 | 83.28 | 7098 | 84.31 | 7052 |
| 9 | 54.91 | 7456 | 85.85 | 6942 | 85.85 | 6942 | 85.85 | 6973 | 87.93 | 7052 | 84.26 | 6973 |
| 10 | 74.41 | 7581 | 85.02 | 7941 | 85.02 | 7862 | 85.02 | 7893 | 85.02 | 7815 | 68.49 | 8222 |
| 11 | 79.81 | 7485 | 76.35 | 7286 | 76.35 | 7332 | 76.35 | 7379 | 76.38 | 7395 | 63.70 | 7613 |

Tabuľka 4

V tomto teste (Tabuľka 4) sme pozorovali aký vpliv má výška tolerancie chyby medzi dvoma feature bodmy na výsledok spájania. Príliš nízka tolerancia ma za následok veľkú prísnosť, a teda sa v mnohých prípadoch stáva že sa pre zdrojový feature bod nenájde žiadny potenciálny bod pre match. Pri malej prísnosti – veľkom čísle – sa nájde bodov pre match mnoho, a z nich sa vyberá najlepší. Pre spájanie je najlepšie vypnúť tento filter feature bodov (hodnota Double.MAX\_VALUE).

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Gt: 96.27% | Feature 5000  Ransac 3000  Inlier treshold 1  Scorer max | | Feature 4000  Ransac 3000  Inlier treshold 1  Scorer max | | Feature 3000  Ransac 3000  Inlier treshold 1  Scorer max | | Feature 2000  Ransac 3000  Inlier treshold 1  Scorer max | | Feature 1000  Ransac 3000  Inlier treshold 1  Scorer max | | Feature 500  Ransac 3000  Inlier treshold 1  Scorer max | |
| dataset | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas [ms] |
| celkom | 84.84 | 88977 | 80.89 | 72464 | 80.31 | 54649 | 80.82 | 36816 | 80.86 | 24024 | 78.21 | 19093 |
| 1 | 92.21 | 10328 | 92.86 | 7785 | 93.23 | 5663 | 93.23 | 4009 | 92.00 | 2902 | 92.68 | 2386 |
| 2 | 85.08 | 8767 | 83.79 | 7083 | 82.33 | 5335 | 82.74 | 3557 | 84.58 | 2309 | 84.90 | 1731 |
| 3 | 96.27 | 8234 | 94.70 | 6506 | 95.85 | 5086 | 95.17 | 3260 | 94.80 | 2075 | 95.82 | 1451 |
| 4 | 80.92 | 7815 | 81.05 | 6603 | 80.39 | 4774 | 78.72 | 3120 | 80.37 | 1965 | 79.25 | 1498 |
| 5 | 82.37 | 8424 | 81.33 | 7145 | 82.36 | 5336 | 82.41 | 3526 | 81.62 | 2309 | 80.65 | 1794 |
| 6 | 69.70 | 7676 | 67.48 | 6552 | 70.09 | 4930 | 66.87 | 3229 | 69.59 | 2059 | 69.25 | 1544 |
| 7 | 93.88 | 8206 | 94.19 | 6646 | 92.66 | 4914 | 92.97 | 3230 | 93.87 | 2012 | 92.99 | 1529 |
| 8 | 85.59 | 7160 | 85.72 | 5772 | 83.62 | 4431 | 84.04 | 3042 | 86.11 | 1903 | 82.95 | 1482 |
| 9 | 85.85 | 6989 | 86.65 | 5818 | 86.48 | 4415 | 85.50 | 2964 | 85.66 | 1887 | 85.38 | 1450 |
| 10 | 85.02 | 8015 | 49.82 | 6318 | 51.99 | 4836 | 51.87 | 3432 | 50.24 | 2403 | 27.04 | 2512 |
| 11 | 76.35 | 7363 | 72.15 | 6225 | 64.38 | 4929 | 75.52 | 3447 | 70.62 | 2200 | 69.40 | 1716 |

Tabuľka 5

V tomto teste (Tabuľka 5) sme pozorovali aký má počet feature pointov vplyv na čas a kvalitu spojenia snímok. Z výsledkov testov vyplíva, že práve tento parameter má najväčší vpliv na kvalitu a rýchlosť spájania. Najlepším kompromisom sa zdá byť 1000 feature pointov.

Doterajšie výsledky boli pozorované na po spojení snímok tak, aby na výslednom obrázku bola veľkosť písma v pixloch 12. Veľkosť písmen na vstupných snímkach je 20 pixlov.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ransac 3000  Feature 1000  Velkost 20 | | Ransac 3000  Feature 1000  Velkost 18 | | Ransac 3000  Feature 1000  Velkost 16 | | Ransac 3000  Feature 1000  Velkost 14 | | Ransac 3000  Feature 1000  Velkost 12 | | Ransac 250  Feature 1000  Velkost 10 | |
| dataset | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas  [ms] | [%] | Cas [ms] |
| celkom | 92.82 | 53140 | 90.80 | 48229 | 84.70 | 49535 | ? | ? | 80.86 | 24024 | 64.14 | 20324 |
| optimum | 98.58 |  | 98.29 |  | 98.10 |  | ? |  | 94.80 |  | 82.46 |  |
| uspesnost | 0.94 |  | 0,92 |  | 0,86 |  |  |  | 0,85 |  | 0,75 |  |
| 1 | 98.17 | 5196 | 97.79 | 5496 | 96.57 | 6384 | 94.35 | 4175 | 92.00 | 3058 | 83.32 | 2574 |
| 2 | 95.75 | 5282 | 87.62 | 4511 | 93.57 | 4898 | 91.07 | 3582 | 84.58 | 2324 | 60.26 | 1856 |
| 3 | 98.58 | 4437 | 98.29 | 4042 | 98.10 | 4216 | 97.15 | 3332 | 94.80 | 2059 | 82.46 | 1731 |
| 4 | 94.73 | 4412 | 91.20 | 4034 | 88.85 | 3979 | 86.49 | 3108 | 80.37 | 1965 | 62.06 | 1716 |
| 5 | 93.16 | 5807 | 93.76 | 4731 | 92.49 | 4546 | 91.35 | 3710 | 81.62 | 2309 | 52.33 | 1887 |
| 6 | 94.87 | 4610 | 85.42 | 4273 | 94,89 | 4422 | 74.04 | 3325 | 69.59 | 2059 | 48.18 | 1788 |
| 7 | 97.21 | 4325 | 96.36 | 3882 | 95.11 | 4182 | 93.91 | 3218 | 93.87 | 2012 | 81.34 | 1763 |
| 8 | 94.5 | 4212 | 91.21 | 3831 | 89.71 | 4135 | 91.68 | 2993 | 86.11 | 1903 | 71.32 | 1638 |
| 9 | 71.35 | 5245 | 69.89 | 4641 | 12.30 | 3460 | 88.64 | 3071 | 85.66 | 1887 | 73.52 | 1622 |
| 10 | 91.17 | 5141 | 90.59 | 4406 | 51.99 | 4836 | 3.45 | 2246 | 50.24 | 2403 | 39.95 | 1934 |
| 11 | 72.15 | 6225 | 72.15 | 6225 | 88.02 | 4929 | 75.52 | 3447 | 70.62 | 2200 | 50.77 | 1825 |

Tabuľka 6

V tomto teste (Tabuľka 6) sme pozorovali aký má vplyv zníženie rozlíšenia spojených snímok pre OCR Tesseract. Výsledkom je že čím väčšie rozlíšenie použijeme tým lepšie bude OCR fungovať, a to aj napriek tomu že snímky budú pospájané menej presne.

**Trvanie algoritmov :**

|  |  |  |
| --- | --- | --- |
| velkosť | 20px | 10px |
| resizovanie | 47 | 81 |
| inicializacia | 79 | 74 |
| Popis obrazkov | 2071 | 1018 |
| Hladanie najlepsieho obrazka na spoj | 754 | 461 |
| Tvorba homografie | 249 | 247 |
| vykreslenie | 1967 | 701 |

## Rektifikácia vstupných snímok

V rámci projektovej výučby sme skúšali aj spojenie výsledkov prác – algorimtus spájania a rektifikácie textových dokumentov. Výsledky algoritmu spájania sú značne lepšie s rektifikovanými snímkami. Hlavný dôvodom je že už nie je potrebné zachytiť prvú snímku s čo najmenšou tranformáciou. Algoritmus rektifikovania „vyrovná “ všetky vstupné snímky.

# Bibliografia

1. http://www.cfar.umd.edu/~daniel/daniel\_papersfordownload/LiangICPR2006.pdf. [Online]

2. http://matthewalunbrown.com/papers/ijcv2007.pdf. [Online]

3. *Relative Speed of BoofCV and OpenCV.* [Online] https://boofcv.org/index.php?title=Performance:OpenCV:BoofCV.

4. *Tesseract OCR .* [Online] https://github.com/tesseract-ocr/tesseract.