|  |
| --- |
| SLOVENSKÁ TECHNICKÁ UNIVERZITA, FAKULTA ELEKTROTECHNIKY A INFORMATIKY |
| VIZS – úloha 1 |
| Návrh optickej odometrie pomocou optického toku |

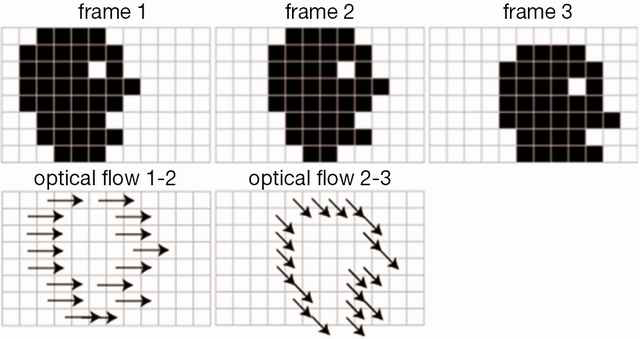
|  |
| --- |
| Adam Sojka 72515, Filip Štec 72520  9.4.2017 |

# Úloha

# Teoretický postup

## Optický tok

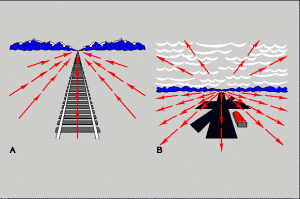
Pod pojmom optický tok rozumieme vizuálny pohyb predmetov na obrázkoch zachytených práve za sebou pomocou kamery. Pohyb pritom môže byť vyvolaný pohybom objektov vo videu alebo aj samotným pohybom kamery. Tento pohyb je zaznamenaný pomocou vektorov, ktoré spájajú určitý bod z predchádzajúceho záberu s tým istým bodom v aktuálnom zábere.



Výpočet optického toku po jednotlivých pixeloch nie je možný, lebo sa dostávame do jednej rovnice s dvomi neznámymi, preto na výpočet použijeme Lucas-Kanade metódu. Táto metóda pozostáva z predpokladu, že susedné body sa pohybujú podobne ako vybratý, tým dostávame preurčenú sústavu rovníc. Metóda rieši aj problém s malými a veľkými pohybmi, lebo používa pyramídu. Pohybom hore po pyramíde sa odstraňujú malé pohyby a veľké pohyby sa stávajú malými, čím dostávame aj mierku.

## Vizuálna odometria

Najprv sa budeme zaoberať jednoduchšou úlohou, a to rozpoznanie smeru pohybu v osiach x, y a z. Pre os x, horizontálny pohyb s kamerou do strany a os z, vertikálny pohyb, je identifikácia smeru pohybu jednoduchá, keďže vektory sa pohybujú vždy iba jedným smerom, horizontálne alebo vertikálne. Pre pohyb v smere osi y, to je pohyb s kamerou dopredu a dozadu, je optický tok zobrazený na nasledujúcom obrázku. Ako vidno, pri pohybe dopredu vektory optického toku smerujú akoby z fotky a zas pri cúvaní idú do fotky. To by sa malo dať zistiť porovnávaním uhla natočenia vektora s jeho absolútnou pozíciou na fotke, keďže vieme predpokladať, že napr. v pravom dolnom rohu fotky pri pohybe dopredu by mal vektor mať uhol približne 45 stupňov a smerovať vpravo dole.



Iným spôsobom určenia pohybu kamery v osi y je pomocou veľkosti detegovaného objektu na obrázku. Je potrebné najprv určiť z dvoch obrázkov, kde vieme v akej vzdialenosti sa objekt nachádza od kamery.

# Programové vybavenie

Ku zdrojovým kódom ako aj tejto dokumentácii je možné sa dostať na umiestnení: <https://github.com/Smadas/VIZSul1>

Všetky zdrojové kódy sú písané v C++ s použitím openCv 3.1, aj staršie verzie by mali byť kompatibilné.

## Funkcia main

1. Deklarácia obrazových matíc
2. Načítanie obrázku
3. Overenie, či bol obrázok načítaný, ak nie skončenie programu
4. Vytvorenie okien pre originálny obrázok, openCv laplace obrázok a naprogramovaný laplace obrázok
5. Aplikácia openCv laplace - applyLaplace(imageOrig)
6. Aplikácia naprogramovaného laplace - applyLaplaceProg(imageOrig)
7. Vykreslenie obrázkov do okien
8. Zapísanie spracovaného obrázka do súboru vystup.png
9. Čakanie, kým užívateľ stlačí klávesu

## Funkcia applyLaplace

1. Deklarácia obrazových matíc
2. Deklarácia parametrov laplaceového operátora
3. Transformácia obrazu na monochromatický model
4. Aplikácia laplaceovho operátora na obraz openCv funkciou
5. Zmena mierky hodnôt jasu výsledného obrazu
6. Vrátenie upraveného obrazu

## Funkcia applyLaplaceProg

1. Deklarácia obrazových matíc
2. Deklarácia parametrov laplaceového operátora
3. Transformácia obrazu na monochromatický model
4. Naklonovanie vstupného obrazu do novej matice
5. Prejdenie každého pixelu obrazu
   1. Výpočet novej hodnoty pixelu - computeOnePixel(src, changed, i, j)
6. Vrátenie upraveného obrazu

### Funkcia computeOnePixel

1. Deklarácia premennej pre pixel
2. Pripočítanie váhovaného pôvodnéhopixelu váhou -4
3. Pripočítaj horný pixel ak nie je mimo obrazu
4. Pripočítaj lavý pixel ak nie je mimo obrazu
5. Pripočítaj dolný pixel ak nie je mimo obrazu
6. Pripočítaj pravý pixel ak nie je mimo obrazu
7. Vráť hodnotu nového pixelu v novej mierke - scalePixelValSymmetric(pixelLaplaceVal)

#### Funkcia scalePixelValSymmetric

1. Vynásob hodnotu pixelu konštantou 3
2. Absolútna hodnota pixelu
3. Orezanie hodnoty pixelu na hodnotu 255 ak je jeho hodnota väčšia ako 255
4. Vrátenie novej hodnoty pixelu

### Funkcia writeImageToFile

1. Deklarácia parametrov pre zapísanie obrázka do súboru
2. Pokus o zapísanie obrázka do súboru vystup.png
3. Ak sa nepodarilo zapísať do súboru, vypíše chybovú hlášku a vráti nepravdu
4. Inak vráti pravdu

# Výsledok odometrie

## Posun kamery od pozadia v osi Y





## Posun kamery v osi X

## Posun kamery v osi Z

## Posun kamery v osi X a súčasne Z

# Zdroje

<http://docs.opencv.org/2.4/doc/tutorials/introduction/windows_install/windows_install.html>

<http://docs.opencv.org/2.4/doc/tutorials/introduction/windows_visual_studio_Opencv/windows_visual_studio_Opencv.html>

<https://github.com/opencv/opencv/blob/master/samples/cpp/lkdemo.cpp>