

Úvodné cvičenie

Cvičiaci

- Ondrej Straka - D708 - ondrej.straka@stuba.sk - primárne cvičenia v pondelok od 13:00 do 17:00
- Michal Kováč - D708 - michal.kovac1@stuba.sk - primárne cvičenia vo utorok od 13:00 do 19:00

Obsah cvičení

Blok 1

Obsahom bloku je získanie základných vedomostí pre prácu s 2D obrazom. Študenti budú implementovať už existujúce funkcionality na prácu s obrazom a taktiež budú implementovať vlastné algoritmy. Počas cvičení bude úlohou študenta vedieť pracovať s 2D kamerou ukladať a vizualizovať z nich dáta a tie následne vedieť použiť v aplikácií.

Zadania pre BLOK 1:

- **Zad. 1** - Oboznámenie sa s kamerou a Open CV knižnicou [5 bodov].
- **Zad. 2** - Aplikácia metód strojového učenia [8 bodov].
- **Zad. 3** - Implementácia metód strojového učenia [12 bodov].

Blok 2

Obsahom bloku je získanie pokročilých vedomostí a praxe na prácu s 3D obrazom. Študenti budú implementovať pokročilé metódy spracovania mračna bodov (3D priestor) ako napr. zhlukovacie algoritmy, RANSAC.

Zadania pre BLOK 2:

- **Zad. 4** - Spracovanie 3D obrazu [25 bodov].

Bodovanie

- Maximálny počet bodov za zadania je 50 (BLOK 1 + BLOK 2 / 25+25 bodov).
- Minimálny počet bodov za zadania je 25 bodov (BLOK 1 + BLOK 2 / 12,5 + 12,5 bodu).
- Pri neodovzdaní zadania na čas môžu študenti odovzdať v náhradnom termíne pričom maximálny počet bodov sa znižuje na polovicu. Ak neodovzdajú ani v náhradnom termíne za zadanie je automaticky 0 bodov.
- Na získanie zápočtu je potrebné odovzdať **všetky** zadania.
- Podmienka pre získanie bodov za cvičenia - **GIT repozitár**

Vytvorte GIT repozitár `psvo_<priezvisko1>_<priezvisko2>`.

- Pre každé zadanie vytvorte samostatný priečinok s názvom `Zadanie_<číslozadania>`.
- Všetky súbory ukladajte vo vetve `master`.

Všeobecné informácie

- Robí sa v skupinách po dvoch.
- Po vypracovaní bude zadania potrebné nahrat' aj do AIS a prebehne plagiátorská kontrola. V Prípade zistenia plagiátorstva budú mať obe skupiny 0 bodov.
- Každé zadanie bude odovzdávané osobne a bodované podľa miery pochopenia všetkých úloh, na základe zhodnotenia cvičiacim.

Používané knižnice a technológie

- **Jazyk:** preferovaný Python no možné použiť aj C++
- **OS:** Linux preferovaný no na osobnom PC vie fungovať aj pod windows
- **Knižnice:** numpy, opencv, libfreenect etc,
- **GitHub:** očakávame, že v rámci skupiny budete mať založený GitHub, ktorý budete odovzdávať, taktiež bude kontrolovaná aj miera commitov od každého kontribútora -> nemal by programovať len jeden

Harmonogram predmetu

Harmonogram cvičení z predmetu PVSO		
Týždeň	Cvičenie	Odovzdávanie
1.	• Úvodné cvičenie, zadanie zad. 1	odovzdávanie zad. 1
2.	• Zadanie zad. 2	
3.	• Práca na zad. 2	odovzdávanie zad. 2
4.	• Práca na zad. 2	
5.	• Zadanie zad. 3	
6.	• Práca na zad. 3	odovzdávanie zad. 3
7.	• Práca na zad. 3	
8.	• Zadanie zad. 4	
9.	• Práca na zad. 4	odovzdávanie zad. 4
10.	• Práca na zad. 4	
11.	• Práca na zad. 4	
12.	• Záverečné cvičenie	

1 Vytvorenie Python prostredia

Pre prácu s kamerou Ximea odporúčame vytvoriť projekt v Python IDE, napríklad v PyCharm. Pri vytváraní projektu nastavíme virtuálne prostredie (venv), aby sme nepracovali s globálnymi knižnicami.

- V PyCharm zvolte File → New Project.
- Vyberte možnosť Virtual Environment (venv).
- Po vytvorení projektu bude virtuálne prostredie pripravené na použitie.

2 Inštalácia Ximea SDK

2.1 Windows

Oficiálny návod

1. Stiahnite a nainštalujte [XIMEA Windows Software Package](#).
2. Po inštalácii sa vytvorí priečinok C:\XIMEA\API\Python\v3\ximea.
3. Skopírujte priečinok ximea do adresára site-packages v rámci vášho venv:

```
.venv/lib/python{3.x}/site-packages
```

2.2 Linux

Oficiálny návod

1. Stiahnite inštalačný balík pomocou príkazu:

```
wget https://kb.ximea.com/downloads/recent/XIMEA_Linux_SP.tgz
```

2. Rozbaľte a nainštalujte:

```
tar xzf XIMEA_Linux_SP.tgz
cd package
./install
```

3. V priečinku API/Python/v3/ximea sa nachádza Ximea Python API.
4. Skopírujte priečinok ximea do adresára site-packages vo vašom venv:

```
.venv/lib/python{3.x}/site-packages
```

3 Inštalácia OpenCV

Na správne fungovanie kamery je potrebné do virtuálneho prostredia doinštalovať knižnicu OpenCV:

```
pip install opencv-python
```

Zadanie 1 (5 b.) - odovzdanie v druhom týždni semestra

Pripojte sa na kameru na vašom PC a vytvorte aplikáciu, ktorá s využitím OpenCV:

1. Zosníma 4 snímky po stlačení medzerníka a následne snímky uloží do vami definovaného priečinku [0,5 b.].
2. Vytvorí mozaiku zo 4 snímok vo formáte 2 x 2 a zobrazí ju pomocou príkazu *imshow* a uloží ju na vami definované miesto – pozor nie použitie funkcie "*subplot*", ale vytvorenie vlastného výsledného obrazu, do ktorého budú jednotlivé snímky vložené pomocou indexovania matice [2 b.].

Dôležité: Najprv vytvorte kompletnú mozaiku (2x2) a až následne pomocou selektorov (indexovania oblasti obrazu) vyberajte jednotlivé časti mozaiky a aplikujte na ne požadované úpravy. Úpravy sa teda budú vykonávať priamo na už existujúcej mozaike.

Ilustrácia rozloženia jednotlivých častí mozaiky:

Časť 1	Časť 2
Časť 3	Časť 4

Príklad použitia selektora v Pythone (výrez oblasti obrazu):

```
cast = mozaika[y_start:y_end, x_start:x_end]
```

3. Na prvej časti mozaiky aplikujte funkciu kernel masky na každý pixel (v rozmere 3x3). Operáciu realizujte už v existujúcej mozaike pomocou selektora. Konvolúciu realizujte pomocou funkcie *filter2D*. Pri spracovaní okrajov použite vhodný padding (napr. *BORDER_DEFAULT*, *BORDER_REPLICATE* alebo iný vhodný typ).

Môžete použiť ľubovoľný 3x3 kernel, napríklad:

- vyhladzovací (priemerovací) filter,
- Gaussov filter,
- detekcia hrán (napr. Laplace, Sobel),
- sharpening filter.

Zvolený kernel stručne okomentujte v kóde. [1 b.].

4. Druhú časť mozaiky otočte o 90° pomocou využitia **for cyklu**. Operáciu realizujte nad príslušnou časťou vybranou selektorom z mozaiky.

Nie je dovolené použiť funkcie `cv2.rotate` ani `np.rot90`. [0,5 b.].

5. Na tretej časti mozaiky zobrazte z RGB len červený kanál [0,5 b.]. Výsledný obraz musí zostať 3-kanálový. (Poznámka: zamyslite sa nad tým, čo musí byť obsahom ostatných farebných kanálov, aby bol zobrazený iba červený kanál.)

6. Do terminálu vypíšte základné informácie o obraze (dátový typ, rozmer, veľkosť). [0,5 b.].

Metodické poznámky

- Cieľom zadania je pochopiť, že obraz je reprezentovaný ako matica (resp. 3D pole pri farebnom obraze) a že jednotlivé operácie je možné realizovať pomocou indexovania.
- Všetky úpravy (konvolúcia, rotácia, práca s kanálmi) realizujte nad konkrétnym výrezom (časťou mozaiky) už vytvorenej mozaiky.

- Dbajte na zachovanie dátového typu obrazu (napr. uint8) po vykonaní operácií.
- Pri práci s farebným obrazom si overte poradie farebných kanálov používané knižnicou OpenCV.

Pomôcky:

- Pre pripojenie kamery využite ukážkový program v PyCharme – kamera už má nainštalovaný driver
- Potrebne návody na vypracovanie jednotlivých úloh nájdete na stránke
 - [Open CV doc - tutorial](#) pre prácu s ukladaním obrázku, zobrazovaním a nahrávaním
 - [Open CV doc - Python](#) pre získanie špecifického farebného kanála z obrázku a získanie vlastností obrázku a vytvorenie subplotu
 - [Open CV - mask operations](#) pre maskovacie operácie na obraze
 - Vysvetlenie formátu MAT v C++ [Open CV C++ MAT](#)