

Optični pretok

David Rubin (david.rubin@student.um.si)

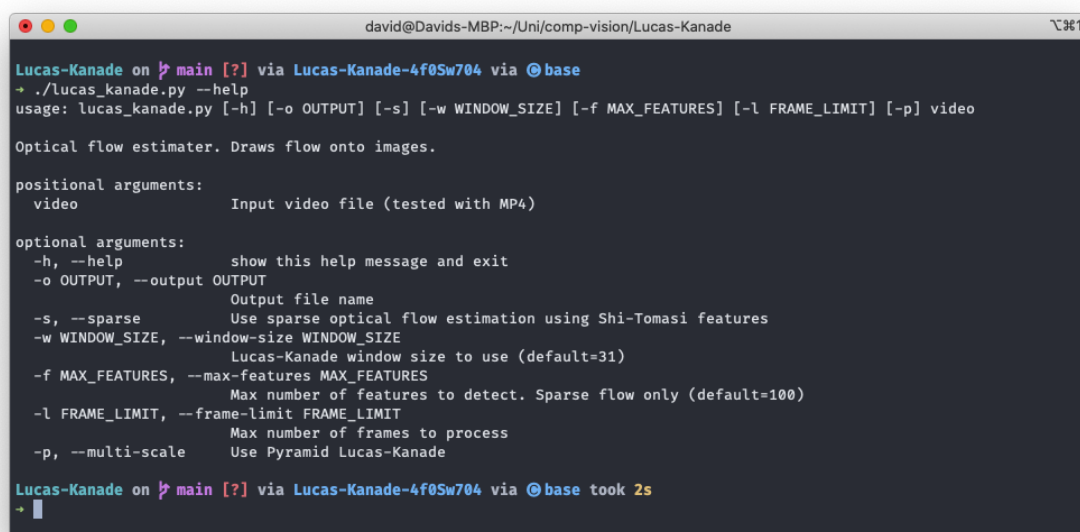
25. avgust 2021

1 Implemetacija

Algoritem sem implementiral v programskem jeziku Python. Za delovanje je potrebna knjižnica `opencv`, `scipy`, `numpy` in pa `matplotlib` (opcijsko za izrisovanje nekaterih informacij). Implementiran program povpraša po video datoteki, z zastavicami pa tudi lahko nastavimo določene parametre. Opisi posameznih parametrov so na voljo na sliki 1.

Rezultati programa so dosegljivi na povezavi <https://imgur.com/a/PDdIXaf>

Za ključne točke pri redkem pretoku sem uporabil Shi-Tomasi detektor na voljo v knjižnici `opencv` (`goodFeaturesToTrack`). Za odstranjevanje vpliva šuma na slikah jih pred zaznavanjem pretoka gladim z Gaussovim jedrom. Prav tako sem odstranil (zakomentiral) pogoje, ki narekujejo kdaj se naj izračuna pretok (lastne vrednosti so dovolj velike in razmerje med največjo in najmanjšo lastno vrednostjo je dovolj veliko). Razlog tišči v tem, da nisem zaznal bistvene izboljšave. Pri risanju gostega pretoka sem se odločil, da ga prikažem kot mrežo, torej pretok se poračuna za vsak posamezen piksel (znotraj okna), prikaže pa se v mreži s črtami. Pri redkem pretoku skušam ohraniti točke na objektu kateremu sledimo, prav tako pa izrisujem črto, ki prikazuje potovanje točke skozi čas. Pri večini primerov sem uporabil velikost okna 31, pri redkem pretoku pa za izbiro ključnih točk največ 100. Ključne točke se v vsakem koraku zmanjšujejo (prenehajo slediti) v kolikor ni pri njih zaznam optični pretok.



```
Lucas-Kanade on main [?] via Lucas-Kanade-4f0Sw704 via @ base  
→ ./lucas_kanade.py --help  
usage: lucas_kanade.py [-h] [-o OUTPUT] [-s] [-w WINDOW_SIZE] [-f MAX_FEATURES] [-l FRAME_LIMIT] [-p] video  
  
Optical flow estimator. Draws flow onto images.  
  
positional arguments:  
  video                Input video file (tested with MP4)  
  
optional arguments:  
  -h, --help            show this help message and exit  
  -o OUTPUT, --output OUTPUT Output file name  
  -s, --sparse          Use sparse optical flow estimation using Shi-Tomasi features  
  -w WINDOW_SIZE, --window-size WINDOW_SIZE Lucas-Kanade window size to use (default=31)  
  -f MAX_FEATURES, --max-features MAX_FEATURES Max number of features to detect. Sparse flow only (default=100)  
  -l FRAME_LIMIT, --frame-limit FRAME_LIMIT Max number of frames to process  
  -p, --multi-scale     Use Pyramid Lucas-Kanade  
  
Lucas-Kanade on main [?] via Lucas-Kanade-4f0Sw704 via @ base took 2s  
→
```

Slika 1: Meni za pomoč ob zagonu programa. Program obvezno potrebuje le video datoteko (preverjeno z MP4 datotekami).