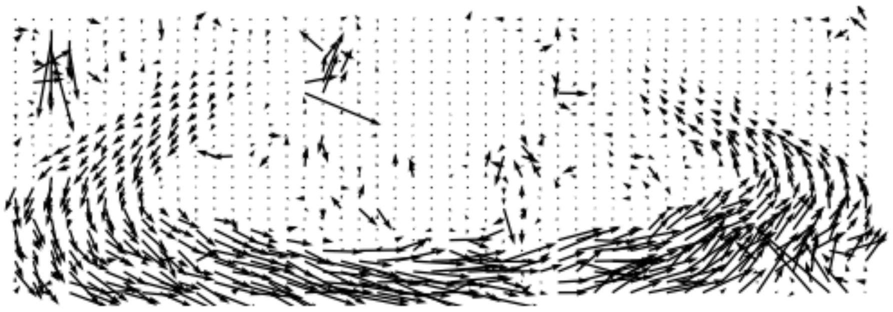
README.md

CrowdFlowDetection



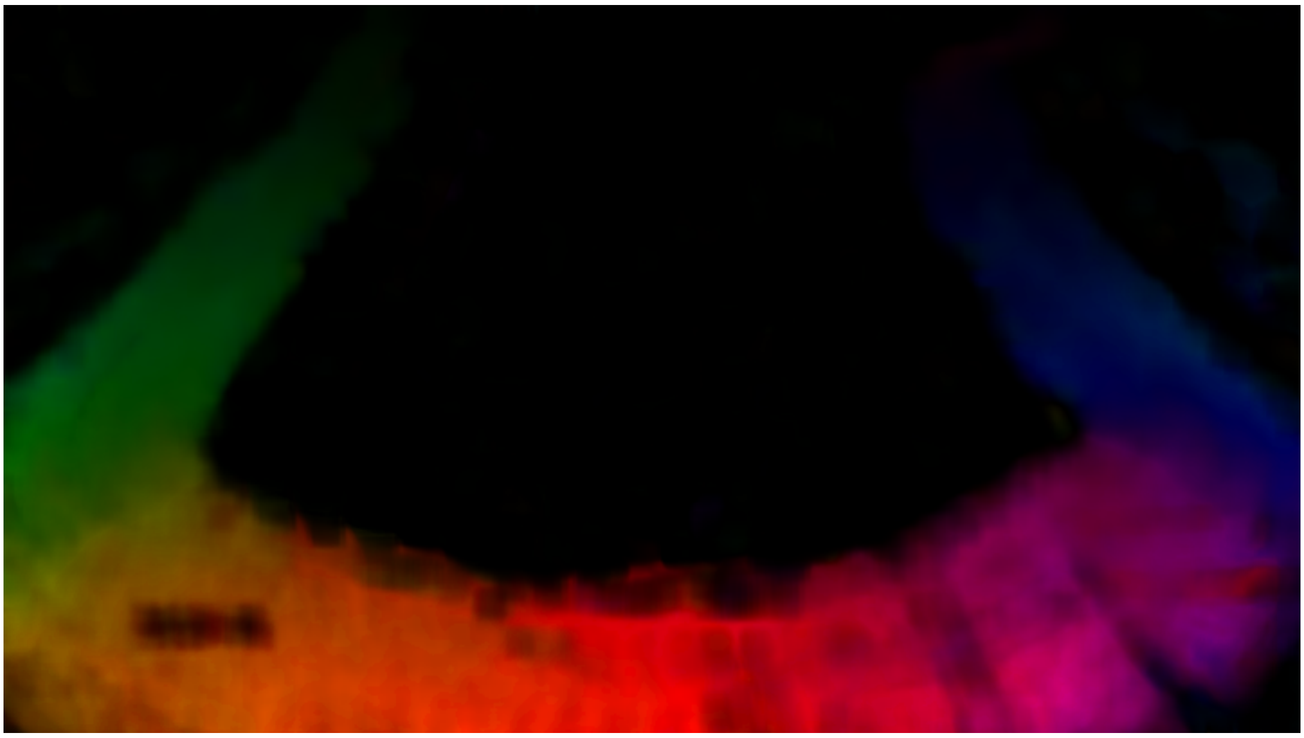


Table of contents

General Info

We segment every frame of the video into regions of different motions based on the similarity of the neighborir
streaklines correspond to similar trajectories of particles passing from neighboring pixels over a period of time.
affinity of current and previous motions at these pixels. First, frame by frame optical flow of the video is comput
set of particles are then moved over the frame to construct the streaklines and the streak flow (not yet impleme
used to compute similarity in a 8-connectivity neighborhood. For every pair of pixels i and j, the similarity is cor
streaklines and streak flow (only streaklines for now).

Team Members

- Ravi Rahar (19/11/EC/014)
- Asad Nizami (19/11/EC/013)
- Umesh (19/11/EC/015)
- Harsh (19/11/EC/016)
- Nitish (19/11/EC/012)

TO-DO

- [ ] Implement Streakflow
- [ ] Give weightage to streakflow in similarity
- [x] Make separate class
- [x] Implement similarity
- [x] Implement watershed
- [x] Implement Streaklines
- [x] Use Optical Flow

Tools Used

- Python
- OpenCV
- Numpy

Contributing

Setting up environment (for ubuntu)

Install Python

```
sudo apt install python3 && sudo apt install python-is-python3
```

Install Pip

```
sudo apt install python3-pip
```

Install OpenCV and Numpy

```
sudo pip install opencv-contrib-python numpy
```

Running for first time

```
git clone https://github.com/RaviRahar/CrowdFlowDetection && cd CrowdFlowDetection
python main.py
```

NOTE: Do not foget to change video file path in `main.py` to video you want to run it on.

References

Implementing crowd flow detection using steaklines. Based on:

```
A Streakline Representation of Flow in Crowded Scenes by Ramin Mehran†, Brian E. Moore‡, Mubara
```