Intro

Kindly read the Proposal first to get the idea what I am trying to implement here.

Usage

The `src.py` file contains some of the custom function definitions, which are used in `demo.py`.
Run `demo.py` for a sample run. For now, the whole implementation isn’t complete and currently, only the epipolar lines and all the TERs (Temporal Epipolar Regions) are being printed, in future, we would add lookup table and highlight the valid TERs.

Dataset

We are using a custom dataset in which we artificially move an object along a linear path and take a number of snapshots and try to find the valid TERs in this case.
We are also using a dataset provided by Tali Basha, here.

Requirements

```
python 3.7.0
opencv-python 3.4.2.16
opencv-contrib-python 3.4.2.16
```

```
numpy 1.15.2

scipy 1.1.0

sklearn 0.19.2
```

Idea

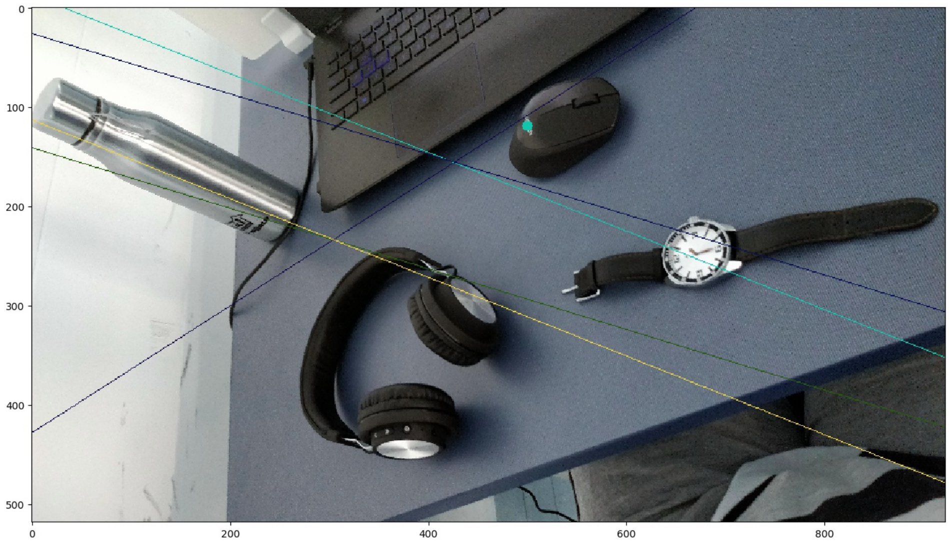
This implementation tries to implement the algorithm to find the valid TERs in an image where we need to find the search space of correspondence of a point of interest, given initial correspondences in a minimum of 3 images that are different from this image.

Methodology

See the proposal `3D_CV_Proposal__Phase_2` for detailed methodology.

Preliminary Results

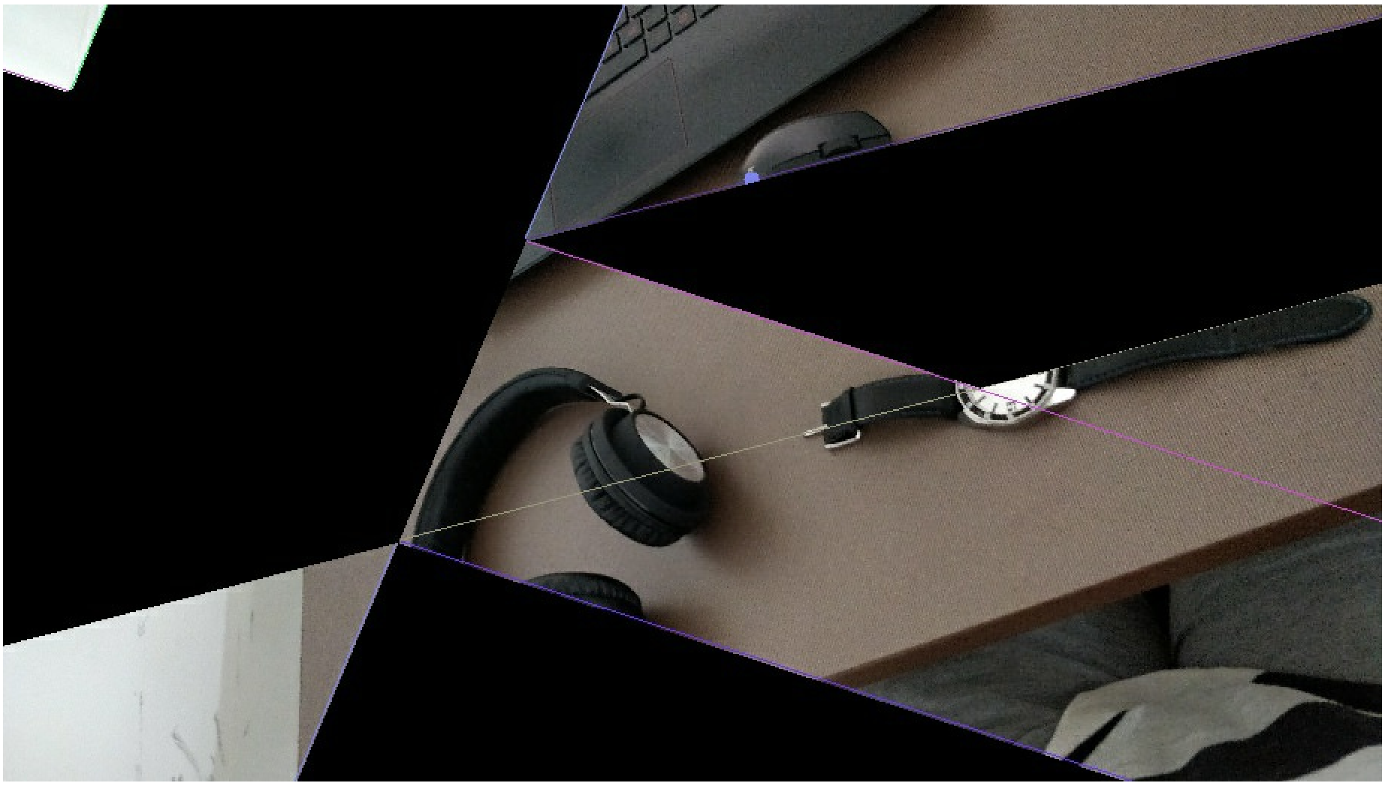
Below are the images we got after dividing the image in which we want to find the correspondence into various temporal epipolar regions.



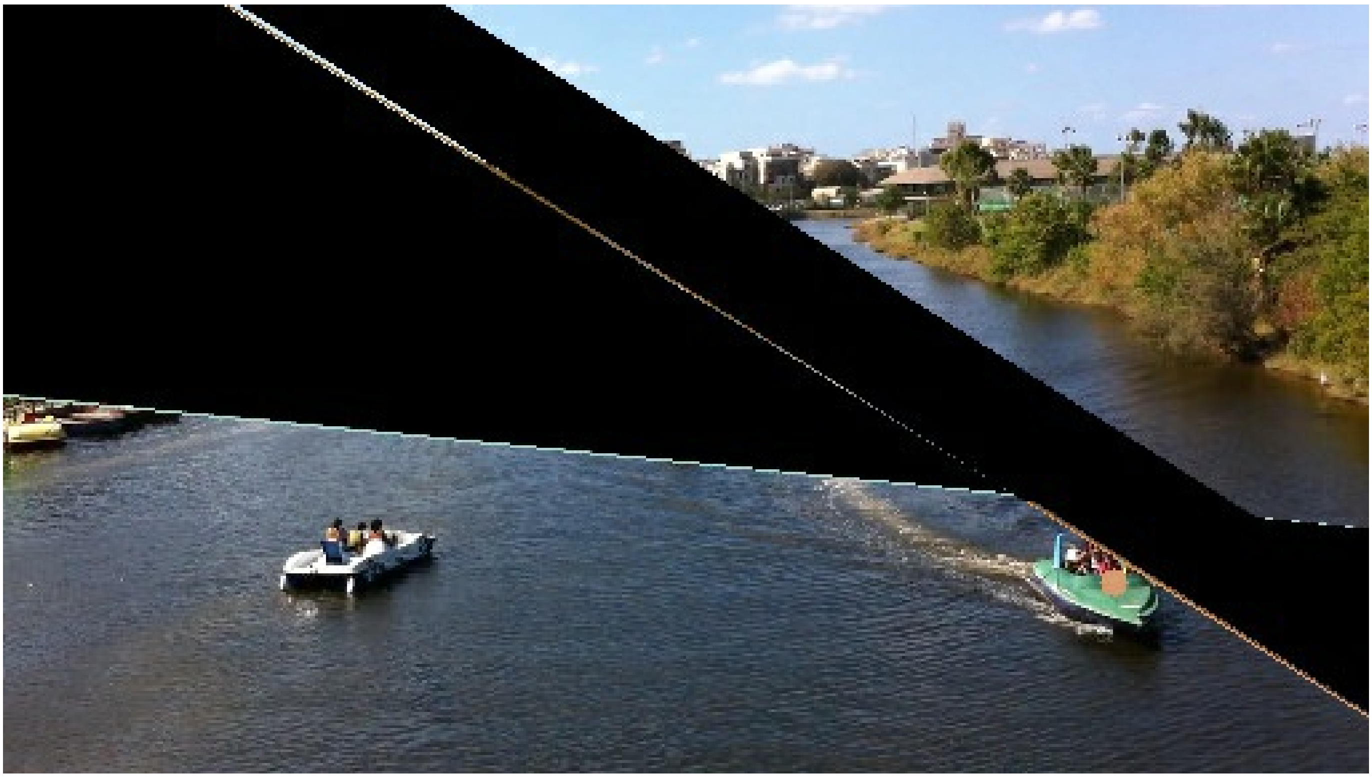
Results

Below are the regions that show that valid search space for the boat moving in approximately linear motion.

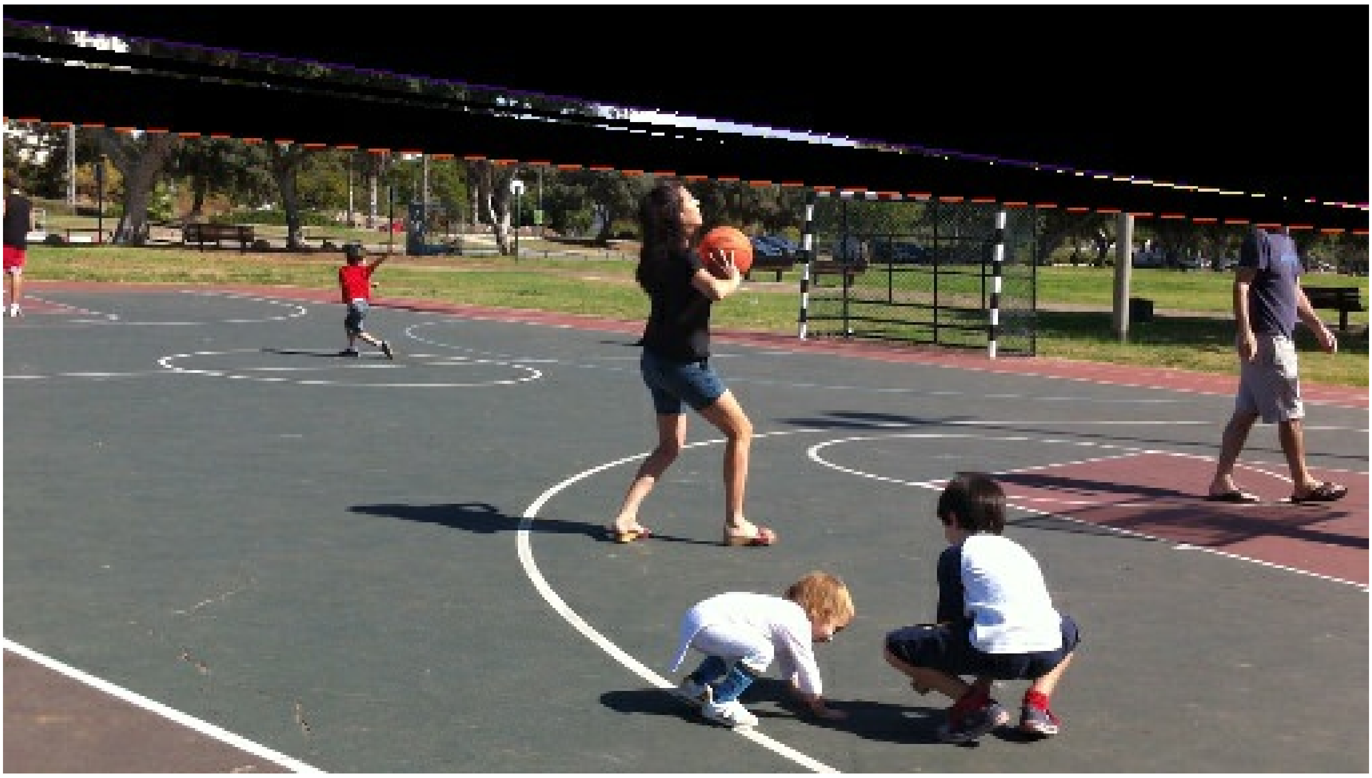
In the below image the point of interest is the mouse top, if you look close enough you can see a blue dot depicting the point of interest.



In the below image the point of interest is the green boat, if you look close enough you can see a brown dot depicting the point of interest.



In the below image the point of interest is the nose top of the person in blue tshirt. We are not able to see the point of interest here as the TER is very narrow.



Observations:

- We see above the results look good, as the point of interest is within the valid regions, in the case of images, `I1` and `I2`.
- In the case of image `I3`, since the valid TER is really narrow we are not able to see the point of interest lies within it.

Conclusions

- We can add a forgiveness metric to the algorithms to allow for slightly non-linear trajectory of a point of interest. This would take `f_pixels`, which would accept any point even on the wrong side of a line until its distance crosses `f_pixels`.
- This `f_pixels` approach could also be used to solve the issue of very narrow TERs as seen in the case of `I3`.

Future Work

- Forgiveness matric to make the valid regions more robust to a nonlinear motion of the point of interest.
- Forgiveness metric to make the valid regions more wide in the case of narrow TERs.
- Convert the code to be more modular to accept any custom fundamental matrices that user might want to give.