Matching Slides to Video

(NOISY BOIS + 1 GAL)

ROLL NO - 20171184 20171118 20171037

Overall idea

- - - - - - - X

We are given homographically reduced frames from the video. We hence apply techniques learnt in DSAA course to find how this task can be done. The basic idea is to derive a fast technique to match slides and frames.

First Approach

- - - - - - - -X

The basic and foremost approach to this problem by us was to consider using normalized cross correlation. using this an accuracy of 100% was obtained on the sample data. Rationale behind using this:

- As the slides and frames will be somilar the correlation should be high

We took correlation in windows of 50x50

Drawbacks:

- Time complexity was high.
- Images are not centered hence using this might not yield good results.

Second Approach

- - - - - - - - -X

Now since the images are not centered, the idea used was is the the point at which correlation is maximum is where they match the most. Hence the correlation values are smoothened using a gaussian curve since the correlation values at edges should be contributing less. using this an accuracy of 100% was obtained on the sample data. Rationale behind using this:

- Correlation should be maximum where they match the most. They would not probably match at edges hence gaussian smoothing i.e. exponential reduction seems rational.

We again took correlation in windows of 50x50

Drawbacks:

- Time complexity was high.
- Gaussian smoothing doesn't work well

Third Approach

- - - - - - - -X

Time taken for all the above processes is very high. We observed that most of the images had text in them. Hence we used OCR based method using tessarect. But it did not work well on images without text hence this option was eliminated. using this an accuracy of 100% was obtained on the sample data and an accuracy of 67% was obtained on complete dataset.

Final Approach

- - - - - - - -X

Time taken for all the above processes was high. The crux lies in correlating to only those images which might be similar. Hence we use ORB measure to compare images and then take the top 5 images

with the max ORB matching. We then apply correlation on them as done in first approach and output the prediction.

Paper ref: http://www.willowgarage.com/sites/default/files/orb_final.pdf

ORB is basically a fusion of FAST keypoint detector and BRIEF descriptor with many modifications to enhance the performance. First it use FAST to find keypoints, then apply Harris corner measure to find top N points among them. It also use pyramid to produce multiscale-features. But one problem is that, FAST doesn't compute the orientation.

Since here rotational invariance is not needed to be too accurate hence ORB is a good measure. It computes the intensity weighted centroid of the patch with located corner at center. The direction of the vector from this corner point to centroid gives the orientation. To improve the rotation invariance, moments are computed with x and y which should be in a circular region of radius r, where r is the size of the patch.

ORB uses brief descriptors which don't work very well on rotation. But as mentioned before it is not much of a concern, this works well.

As the time is reduced even compared to SIFT and quite less compared to correlation it is helpful.

SAMPLE

CODING OPTIMIZATIONS

1. **Caching**: Images and the descriptors obtained from ORB are cached hence they do not need to be computed and read again and again
2. **Top 5 Approach**: Since only top5 matching entries are correlated it is quite optimized compared to all pairwise correlations

NOTE: The gaussian smoothing based approach was eliminated because ORB already looks into scaling.

REFERENCES

[1] http://www.willowgarage.com/sites/default/files/orb_final.pdf

[2] https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_orb/py_orb.html