Video Stabilization using Warping

Gupta, Ankesh
2015CS10435

November 28, 2017

Problem Statement

- Write the code for the recursive Gauss-Newton process yourself.
- Implemente video stabilization using above.

Implementation

1. For *image warping*, work of ***Lucas-Kanade 20 Years On: A Unifying Framework by Simon Baker and Iain Matthews*** was referred. **Inverse Compositional Image Alignment** was implemented.
2. For *video stabilization*, we first ask user for a bounding box/object that would be stabilised. Then all subsequent frames are *warped* to a reference frame(it was the *first frame* in our case).

Tips and Trick

Since the above was implemented in OpenCV using python, following are some tips and tricks for getting Gauss Newton to converge in the aforementioned language.

1. Use the *Inverse Compositional Image Alignment Algorithm* as mentioned above. Simple algorithm mentioned in section 2.2 of paper fails to converge.
2. Remove the *inverse* in the last step of iterative pseudo code.

3. Compute *Hessian Inverse* in pre-computation part for speedy iteration.

4. Avoid *for-loops* and compute all operations using in-built numpy functions for performance(Check reference code).

5. For video-stabilization, keep an *iteration limit* apart from ϵ, so that some frames that does warp(converge) gets handled.

Assumption

1. The stabilizer corrects motion upto a certain waver.

2. The object being stabilized does not move out of scene in subsequent frames.

3. *Inverse* of Hessian exists.

4. The transformation is *affine*.

Results

Epsilon	Iterations
0.05	1
0.01	290
0.005	297
0.001	315
0.0005	581

- Increasing ϵ above a certain value, or decreasing below a threshold is futile. *0.005 appeared optimal.*

- Larger resolution Images take *longer time* to converge.

- Sample warped Images and Videos are shown in respective directories.

- The book video has moderate shiver and algorithm successfully tackles it.

- The kitty video has large camera motions and algorithms produces and *unstable output*.

References

1. Paper mentioned in implementation bullet 1.
2. Video and Picture Credits: *Suyash Agarwal, Saket Dingliwal*