User Manual – Computer-Vision Stabilization Software

Table of Contents

THE UNIVERSITY OF ILLINOIS COLLEGE OF MEDICINE
PEORIA CHICAGO ROCKFORD URBANA

Overview

Computer-Vision Stabilization software is a LabVIEW program to correct motion displacement and deformation with computer-vision image registration algorithms, which is a part of Computer-vision Assisted STabilized intravital imaging (CASTii) method. Image processing of Computer-Vision Stabilization consists of four distinct steps: (i) feature points detection, (ii) motion tracking, (iii) motion correction and (iv) image averaging. Using the Shi and Tomasi method[1], the software detects feature points of images, which are the 'corners' changing rapidly in intensity in the reference image. The software tracks the feature points from the reference image to the paired image using Lucas Kanade optical flow analysis[2], the algorithm estimating the motion vector between frames. These feature point pairs are then used to calculate a homography matrix, which defines the geometric transformation of the paired images. This perspective transformation matrix is used for motion correction. For further motion correction, the software averages specified frames.

System Requirements

We recommend 32-bit LabVIEW 2017 17.0f2 on Windows 7 to use the Computer-Vision Stabilization software.

The following LabVIEW add-on modules and python modules are required:

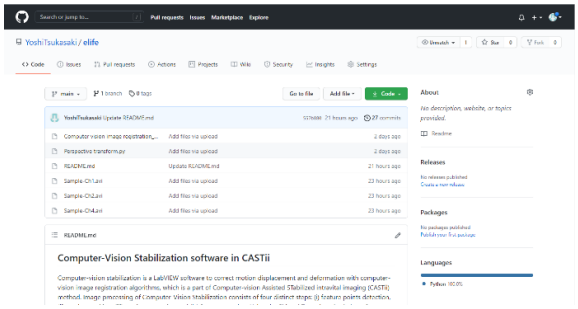
Vision Development Module (required by image processing in LabVIEW)

Enthought Python Integration Toolkit for LabVIEW 1.2.0 (required by Python-OpenCV funcion in LabVIEW)

Python 2.7.13 (required by Python-OpenCV function in LabVIEW)

Python package: mkl 2017.0.1-2 or newer, numpy 1.11.3-2, opencv 2.4.9-5 (required by Python-OpenCV funcion in LabVIEW).

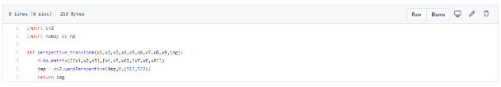
How-to-use



https://github.com/YoshiTsukasaki/elife

1. Go to the GitHub releases page, download 'Computer-Vision Stabilization_Open Source.llb' and 'Perspective trasform.py'. If you need, please download sample movies (‘Sample-Ch1.avi’, ‘Sample-Ch2.avi’, ‘Sample-Ch4.avi’).

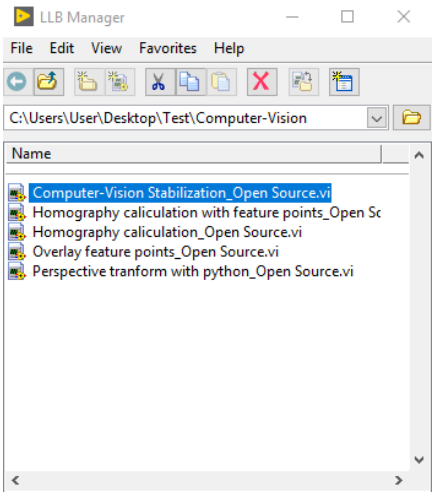
2. To download filed go to each file page, and click ‘View raw’ to download.



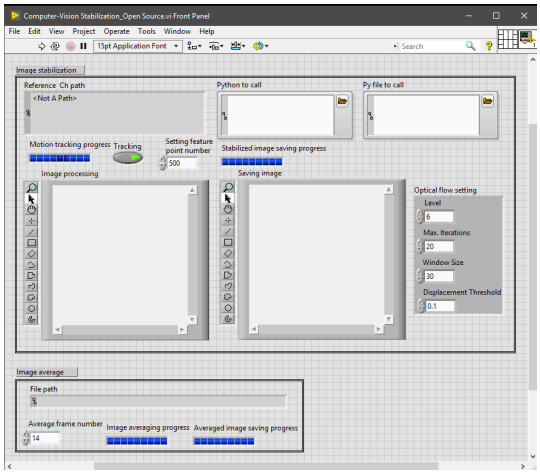
```
import cv2
import numpy as np

def perspective_transform(x1,x2,x3,x4,x5,x6,x7,x8,x9,img):
    M=np.matrix([[x1,x2,x3],[x4,x5,x6],[x7,x8,x9]])
    img = cv2.warpPerspective(img,M,(512,512))
    return img
```

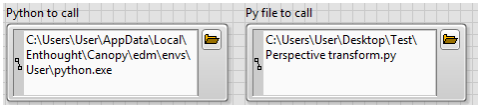
3. For py file, click ‘Raw’ button. From text information, make ‘Perspective transform.py’ file.

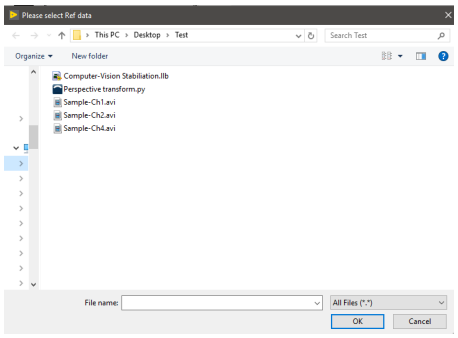


4. Open the downloaded LLB file in LabVIEW and then choose 'Computer-Vision Stabilization_Open Source.vi'.



5. Choose 'python.exe' in Python to call 'perspective transorm.py' in Py file to call on front panel in LabVIEW.

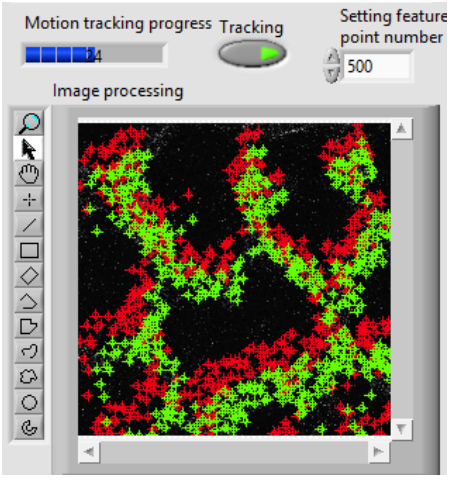




6. Click white arrow buttom in upper left of application and open reference movies (i.e. Sample-Ch1.avi) in 'Please select Ref data' dialog.

   Note: File name should include 'Ch1-4' and all data should be in same folder. The only AVI file is available for this software. To test software, please use sample movies in the

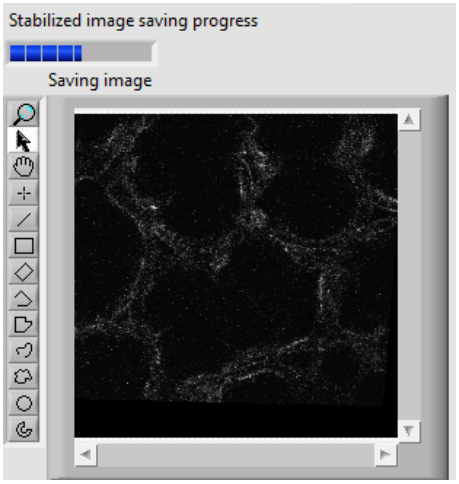
GitHub releases pages ('Sample-Ch1.avi', 'Sample-Ch2.avi', 'Sample-Ch4.avi').



7. The software starts motion tracking over time frames.

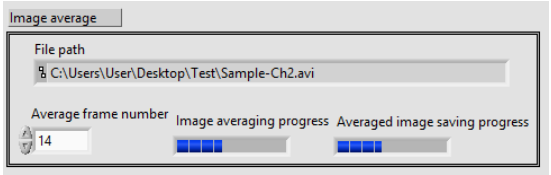
   Note1: Cross circle cursor will appear with tracking button on. Red cross-circle cursor indicates feature points of a reference image. Green cross-circle cursor indicates tracked feature points of paired image.

   Note2: Feature points number threshold can be set by users.

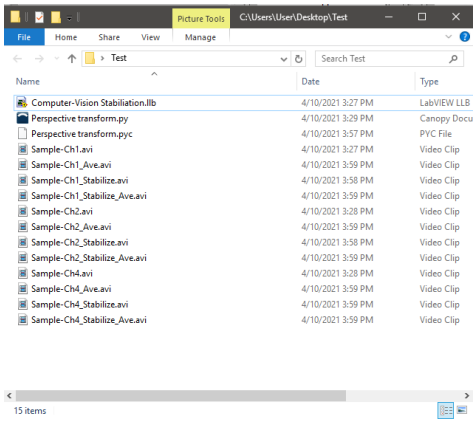


8. After finishing motion tracking over all-frames, the software starts motion correction for all of Channel over all-frames.

   Note: Other channel movies should exist in same folder as reference movies. And movie length should be equal.



9. After finishing motion correction, the software starts image averaging with specified 'Average frame number'.



10. Stabilization and averaging results will appear into the folder with reference movie.

Reference

1. Shi J, Tomasi C. Good features to track. *IEEE Computer Vision and Pattern Recognition*, 593–600 (1994).

2. Lucas BD, Kanade T. An iterative image registration technique with an application to stereo vision. *Proceeding of Imaging Understanding Workshop* **2**, 674-679 (1981).