**Content-Preserving Warps for 3D Video Stabilization Paper findings:**

Used Voodoo 3D camera tracker, didn’t implement their own method -> this is quite an effective method and can be used in blender

Can use 2D stabilization (not as good with larger rotations and moving camera)

This method assumes the camera is moving along a path and filters that which might be a vey good option whilst moving the drone, however when the drone is hovering at a set position 2D stabilization might be very effective for removing the small vibration/movement of the drone.

2D stabilization approach:

1. Estimate projective transform between 2 frames
2. Parameters of this are low pass filtered across time
3. Full frame warp computed between original and filtered motion
4. MATSUSHITA, Y., OFEK, E., GE, W., TANG, X., AND SHUM, H.Y. 2006. Full-frame video stabilization with motion inpainting. IEEETransactions onPattern Analysis and MachineIntelligence 28, 7, 1150–1163.
5. GLEICHER, M. L., AND LIU, F. 2008. Re-cinematography: Improving the camerawork of casual video. ACM Transactions on Multimed. 5, 1, 1–28.

Paper says the 2D approach is quite limited and not as effective as 3D, however 3D more computationally complex.

3D approach:

1. Compute 3D camera motion and 3D point cloud
2. Specify desired 3D camera motion (my case no movement)
3. Warp each frame using 3D point cloud and video frames
   1. rigid transformations for warping best
   2. They use a grid like warping, splitting a frame into nxm grid mesh and compute a warped version of this grid, minimize cost function of 2 terms: sparse displacement of point from input to output frame, similarity transformation term from each grid

**2D Stabilization Methods:**

Paper: <file:///C:/Users/Sean/Downloads/article.pdf>

Interactive results from paper: <https://ipolcore.ipol.im/demo/clientApp/demo.html?id=209&key=8B3EB41CC1CCE972BDA2DC0CA9F8098F>

Using feature-based method for motion estimation with homography for transformation type to find the next frame for smoothed results.

Trying out a python version based on this tutorial, bit more basic as we only translation and rotation in 2D and not in a 3D. Tutorial link: <https://learnopencv.com/video-stabilization-using-point-feature-matching-in-opencv/>

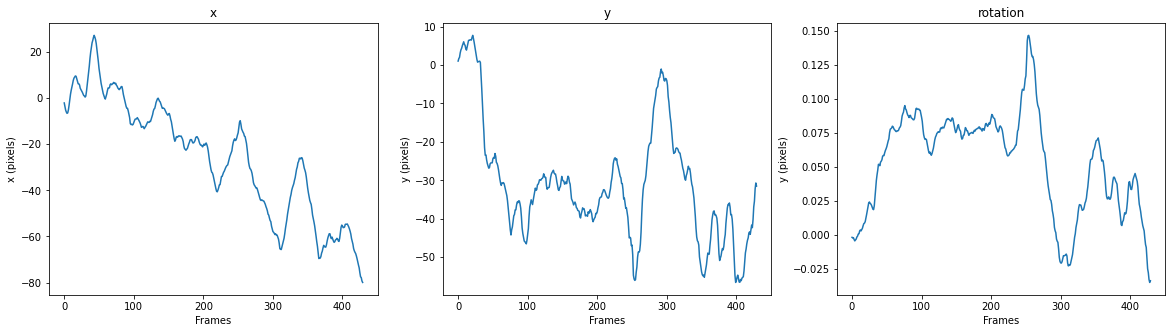
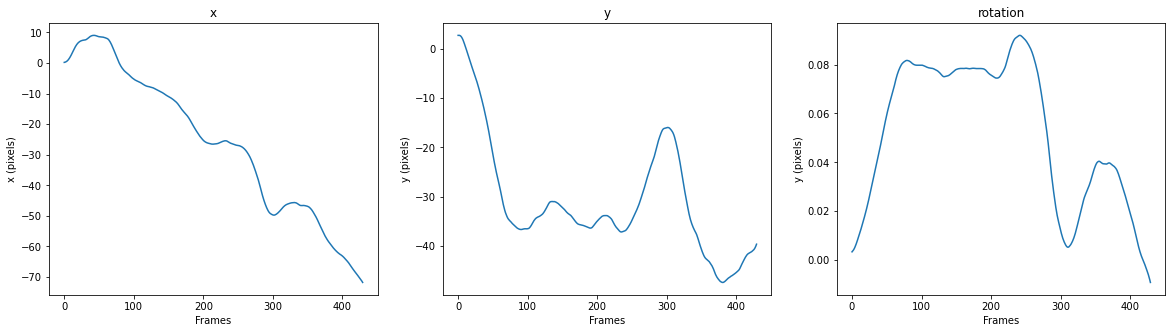
* Applying Euclidean (similarity) transformation -> 2D translation rotation and scale
* Successful with small camera motion movements
* Euclidean requires tracking only 2 points however the more points the more accurate
* Smooth the trajectory of x, y and angle using moving average filter
* Trying out on video under “stabilized videos/vid1.mp4”
  + Stable is using radius 20 for smoothing path
  + Stable2 is failed attempt of zeroing movement of camera (doubled it)
  + Stable3 is successful attempt of making the camera movement zero, most successful one as it eliminates very large movements quite well in the x and y direction as well as rotations. Retry this method on lesser movements to see better results
  + Stable4 is using radius 100 for smoothing path, very similar to 20 radius, will be the more effective choice if the camera is following a path, not set vibrating stationary.

Running some tests on 2D stabilization:

**Vid2 -> stationary camera with slight movement**

* Vid2-stable1: Path from moving average filter with radius 30

Path:



* Vid2-stable2: Assuming path of 0 movement
  + Way better stabilization for zero camera movement, the movement was still quite significant as we can see in the graph below. 3 pixel movement isn’t a lot in pixels but when transferred to mm the change is more significant due to depth of the object tracked
  + Graphical user interface, chart

    Description automatically generated
  + Plot of the camera movement or transforms\_smoothed, the transformation needed in x, y and rotation to obtain zero movement. It is basically the flip of the unsmoothed path which makes sense to zero out movement.
  + Text

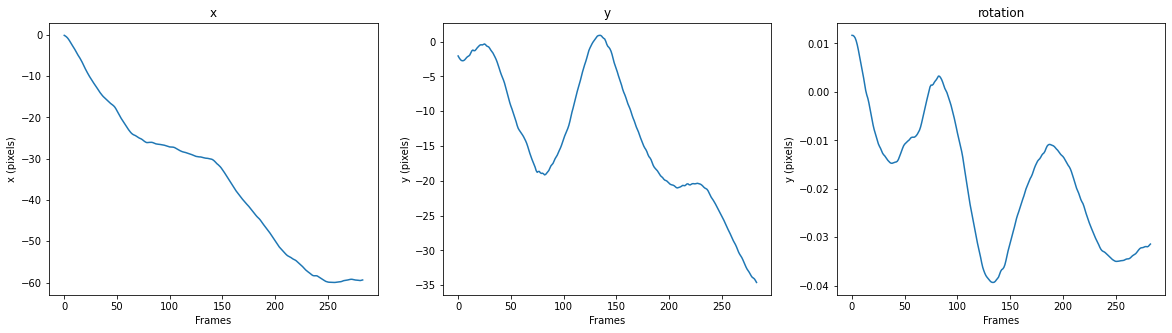
    Description automatically generated with low confidence

**Vid3 -> stationary camera with heavy movement**

* Vid3-stable1: Path from moving average filter with radius 30

Path: Chart

Description automatically generated



* + A lot of movement still will occur in this video due to the large vibrations
* Vid3-stable2: Assuming path of 0 movement using superpoint/glue
  + Great results with basically no movement, very slight movements due to the intensity of the vibrations, smaller vibrations will be eliminated well
* Vid3-stable3: Assuming path of 0 movement using KLT
  + Graphical user interface

    Description automatically generated with medium confidence
  + Vibration movement is very similar to above, proving results are comparable
  + Video results are very similar, hard to find much difference at all. Therefore, using KLT is a better option as superpoint/glue takes a lot time to find transforms and computational power

**Vid4 -> moving camera with slight movement**

* Vid4-stable1: Path from moving average filter with radius 30

Path: A picture containing text, screenshot, picture frame

Description automatically generated

Chart

Description automatically generated

* + Can see a clear path moving along the x and y direction due to camera motion hence this method will be better than zeroing out the movement.
* Vid4-stable2: Assuming path of 0 movement
  + Yep, this technique is shocking when movement is involved.
  + However, I will be working with footage from a hovering drone and therefore this method will be the go-to, however interesting to learn how the other method also works.

Success!

**Stabilization with attention on movement and rotation in the z direction:**

Video: “Vid5.mp4”

Results:

Vid5-stable1: Path from moving average filter with radius 30

* Text

  Description automatically generated with medium confidence
* Chart, line chart

  Description automatically generated
* Can’t draw many conclusions from here

Vid5-stable2: Assuming path of 0 movement

* Can see the rotations in the z direction and movement in z direction is still largely there so we must be able to move to 3D stabilization to remove this.
* The object was close to the camera also which can have a big effect on the stabilization as points are predominantly found from the background.
* Hard to draw big conclusions from here but a 3D stabilizing system will perform better in this scenario.

**Stabilization with moving object on screen:**

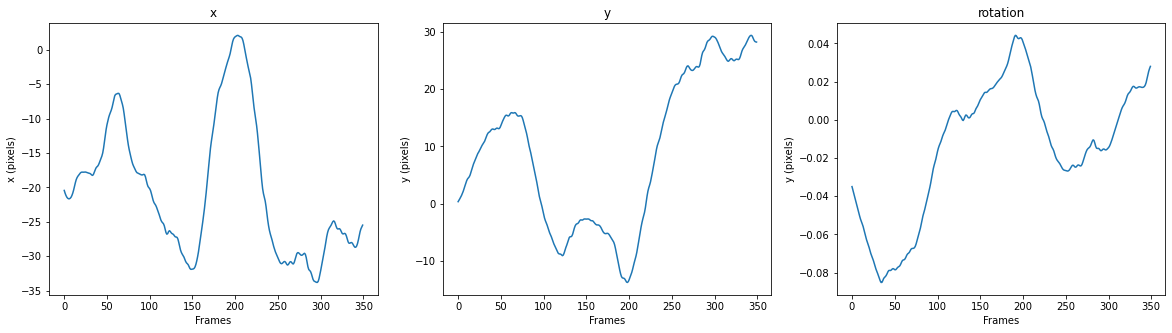
Video: “Vid6.mp4”

Results:

* If majority points found in background the transformations will be completed for zero movement in the background which doesn’t include the movement of the object
* Also, possible to stabilize the object movement by specifically taking the points from the moving object
* This could result in problems with large moving objects entering the frame causing incorrect transformations, would need to specifically focus on background points and eliminate the rest

Vid6-stable1: Path from moving average filter with radius 30

* Graphical user interface

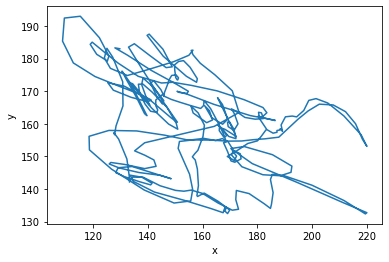
  Description automatically generated
* 
* Not a great method in this situation

Vid6-stable2: Assuming path of 0 movement

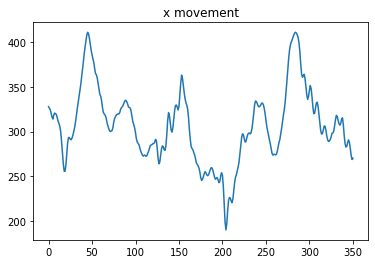
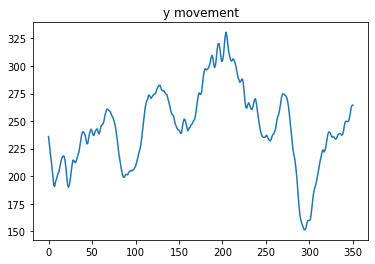
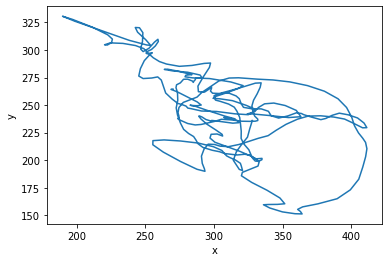
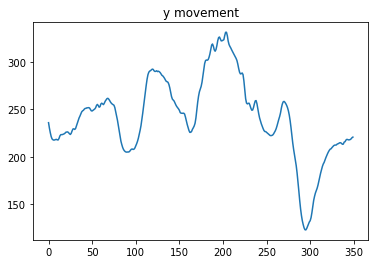
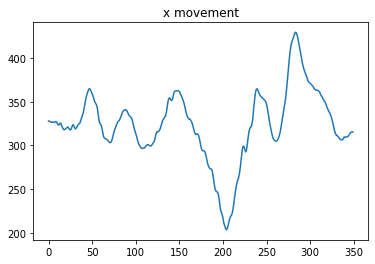
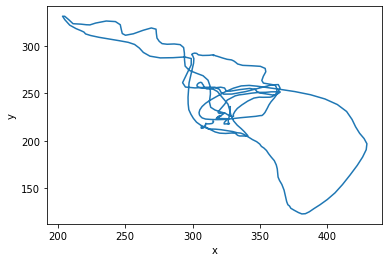
* Very successful, movement of object is kept, and vibration of camera/background is removed
* This is for small object, when larger objects used results may differ due to theory above

**Video 6: Comparing movement of stationary and moving object from (un)stabilized video**

* Movement of selected stationary point in normal and stabilized frames for stabilization accuracy:

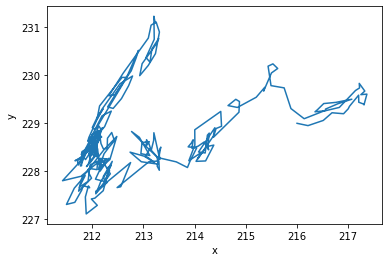


* Way improved stabilization with large vibration, gone from a change in x and y of 120 and 70 to 20 and 15 pixels
* With smaller vibrations this method will basically zero out the movement
* Tracking the movement of the moving object (un-stabilized -> stabilized video)

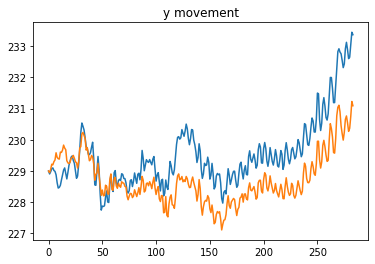
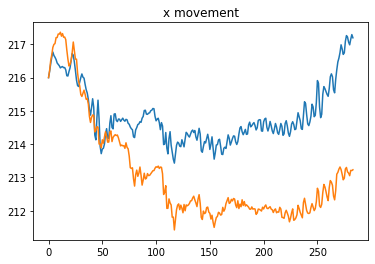


**Video3: Checking stabilization difference on Superpoint and SIFT + KLT**

* Unstabilized video vibration -> stabilized video vibration (Superpoint/glue, KLT)



* Stabilized methods did very well to stabilize most the vibration with slight error
* Super has error/uncertainty in (x, y) of (6, 4)
* KLT has error/uncertainty in (x, y) of (4, 5)
* Hard to tell which is better (both perform equally well)
* Comparing x and y movement individually (blue = KLT, orange = Super)



Implementing homography:

* Used cv2’s estimateAffine3D using 2 sets of 3D points
  + Under rot\_trans\_testing.ipynb
  + Successfully finds the R and T between 2 frames
  + However, can’t apply this matrix onto an image as it is only in 2D need to convert matrix to a 3x3 from 4x4, does this include all 3 rotation and translation?
* Videos on math behind methods.
  + <https://www.youtube.com/watch?v=rHLEWRxRGiM>
  + <https://www.youtube.com/watch?v=vlb3P7arbkU&t=23s>
  + <https://www.youtube.com/watch?v=4srS0s1d9Yw> (vid 1 - 2D rotation matrix)
  + <https://www.youtube.com/watch?v=wg9bI8-Qx2Q> (vid 2 – 3D rotation matrix)
  + <https://www.youtube.com/watch?v=09I15RO49vg> (vid 3 – homography matrix)

Other:

* Can look into Meshflow for more effective stabilization: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://www.liushuaicheng.org/eccv2016/meshflow.pdf (paper) <https://abhitronix.github.io/2018/11/30/humanoid-AEAM-3/> (shorter description)

**Drone footage:** (from Yanda)

Filename: “drone/drone\_footage.mp4” or “drone\_footage\_stabilized.mp4”

Normal footage: (point on chess board - 2000)

A person working in a factory

Description automatically generated with low confidence

A picture containing text, screenshot, picture frame

Description automatically generated

(Point on beam - 3080)

A picture containing line chart

Description automatically generated

Similarity graph: (orange=beam, blue=chess)

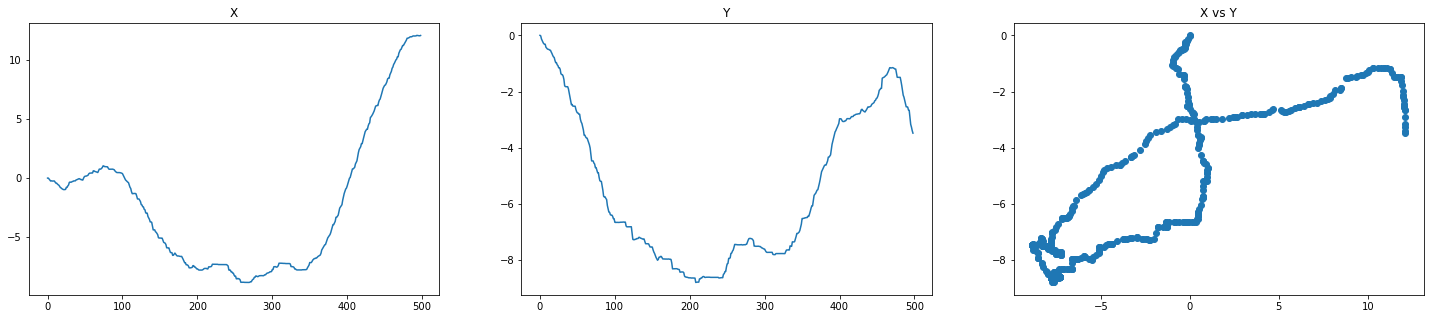
Chart, line chart

Description automatically generated

This is just pixel difference (no depth considered) therefore the incorrect scaling, largely evident in the 3rd similarity graph

Stabilized footage:

(Chess point – 2009)



(Point on beam – 3044)

A picture containing icon

Description automatically generated

Similarity graph:

Chart, line chart

Description automatically generated

Unstablized vs stabilized (beam) legit exactly the same…

Chart, line chart

Description automatically generated