**Testing Vibration Removal from video**

* Aim is to equip camera on drone and remove the vibration from the drone hovering due to wind or etc.
* I will run experimental tests here to show this method works

Video file: “Vibration/l2.mp4” for the left frame and “Vibration/r2.mp4” for the right frame

This video is of me moving a book up and down in the y axis in a somewhat smooth manner, whilst the camera is shaking slightly.

To remove the vibration 2 points are tracked, one on the object and one on a background stationary object. By subtracting the background from the object, we get a smooth y displacement with vibration removed.

Below graphs show the process:

Background point on both frames with KLT tracking output on black image for both frames:

A picture containing text, wall, person, indoor

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Y displacement output from background object (showing movement of the camera):

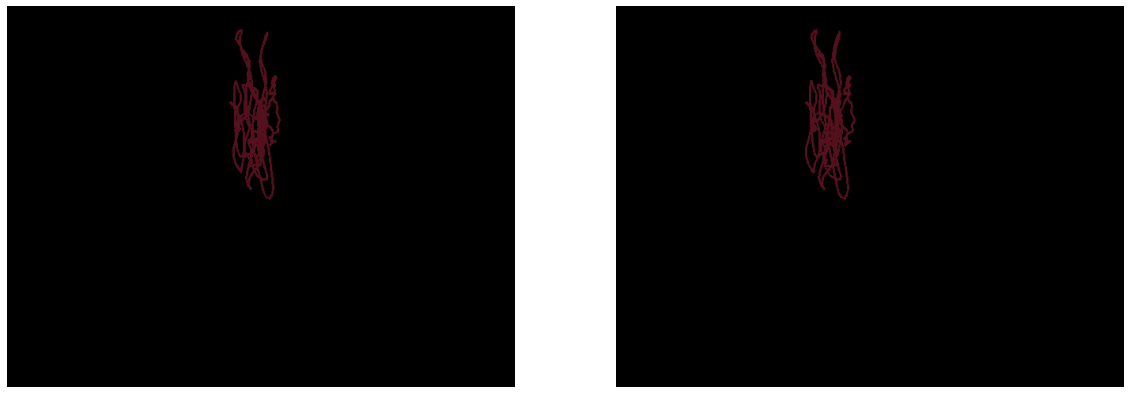
Chart

Description automatically generated

Object point with corresponding tracking path:

A picture containing text, wall, person, indoor

Description automatically generated



Y displacement output from object point (including the camera vibration as seen from time 8-12s):

Chart, line chart

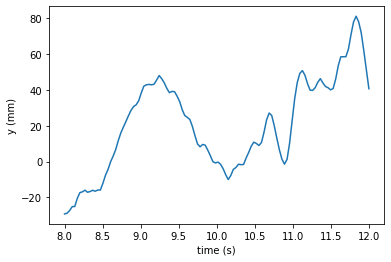
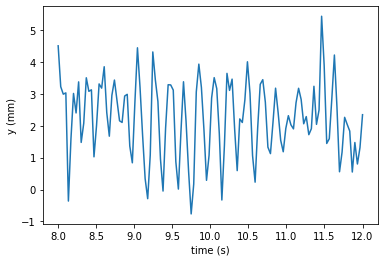
Description automatically generated

Smoothed output when vibration of camera removed from the object displacement:

Chart

Description automatically generated

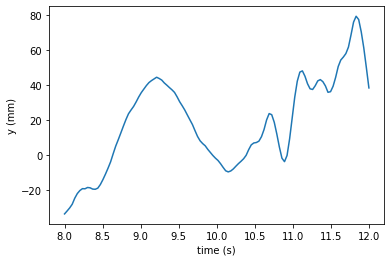
Clear effectiveness in the change from time 8-12s:



Object displacement with vibration

Camera vibration (background displacement)

Object displacement with vibration removed



**Testing on a larger video with movement in x, y and z directions with significant vibration:**

Results are hard to examine but by manually looking through they look promising.

Video file: “Vibration/r5.mp4” and “l5.mp4” for the right and left footage respectively.

Background point vibration (quite extensively done):

Graphical user interface

Description automatically generated

Object point vibration (moving the object in all 3 directions at different times):

A picture containing text

Description automatically generated

Results after removing the camera vibration:

Text, whiteboard

Description automatically generated

Results still look quite jolty and a lot of vibration still in there, will need to do some more calmer attempts to see how the performance is.

Try moving an object very slowly and gently and have a lot of vibration, this experiment I moved the object quite freely and quite a bit, making it hard to assess a good job.

Following 3 test are for better testing: 6 is movement in x, 7 for y and 8 for z object displacement:

Video file: “l6.mp4” & “r6.mp4” (x movement)

Results:

Background/camera vibration:

Chart, line chart

Description automatically generated

Object movement:

Chart, line chart

Description automatically generated

Object movement with vibration removed:

A picture containing text, monitor, screen

Description automatically generated

Notes:

Only x movement and the x in image 3 is clean nice movement where it is evident it has removed the vibration/movement of the camera (look a the last second 4-5s)

The y movement has gone from between -50 and 150 to -40 and 30, a much smaller gap and much more likely as there was a slight change in y as my hand isn’t perfectly staying horizontal.

Same goes for the depth, the initial depth movement was between 480 and 600 (120 gap) and the shape was random but after vibration was removed you can see the depth increases at around the same time the object moves to the left meaning I was moving the object at an angle.

Video file: “l7.mp4” & “r7.mp4” (y movement)

Results:

Background/camera vibration:

Text, whiteboard

Description automatically generated with medium confidence

Object movement:

A picture containing text

Description automatically generated

Object movement with vibration removed:

A picture containing text

Description automatically generated

Notes:

Can see the movement in x and z direction looks very similar to the vibration meaning not much movement in those directions.

The movement in the y direction has become more evident and correct, oscillating from -100 to 100 much more smoothly like the true movement should be.

The change in x is relational to the change in y, same timing of peaks, meaning a slight angle in movement as before.

Successful run and can see the vibration/movement of the camera removed.

Video file: “l8.mp4” & “r8.mp4” (z movement)

Results:

Background/camera vibration:

Text, whiteboard

Description automatically generated

Object movement:

Whiteboard

Description automatically generated with low confidence

Object movement with vibration removed:

A picture containing whiteboard

Description automatically generated

Notes:

Depth movement is recovered however probably the least accurate one as it is not the smoothest result as it more should be.

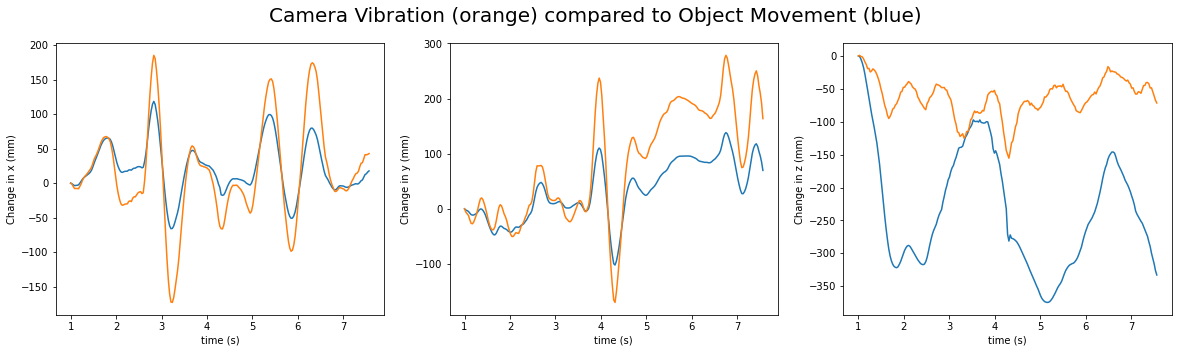
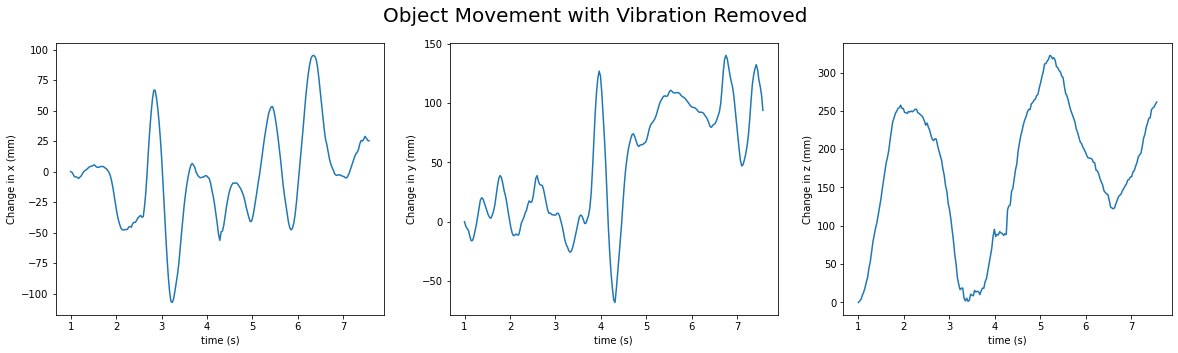
The vibration and movement in x and y are very similar but are producing big changes in the removed vibration outcome, looks like there is some scaling problem and we cannot just take away one measurement from the other.

If the background moves 20mm the object could be moving by 40mm because of depth distortion. Might need to look into how to change this, use pixels for change and then calculate in mm?

Could also be though that this is correct and there was indeed these changes in x and y as moving closer the camera could have gone down/up in y value as I wasn’t coming directly at the camera with the object. (this seems like a more viable answer as the past 2 results showed promise and the found depth here looks correct)

Another thing to note is that the depth is not starting at -300mm here, this is a side effect of subtracting the camera movement which has a much greater depth away and will therefore always result in a negative outcome. To get around this, likewise to x and y, the initial point starts at 0 and finds the change of depth from there, so if we know the first point depth we can recover all the rest.

For a clearer result of last output r8:



Still wary of this depth distortion having effect on how the final x and y outcomes look – look into this more…

Depth is correct however the x and y is not as its based off the depth (proportionally) therefore the further away the object such as the background point the more movement it should have.

Equations for x and y displacement across frames:

x = ((x2-x1)/(f\*210))\*z

y = ((y2-y1)/(f\*210))\*z

As you see they are multiplied by depth, the further away the object the more movement it will have, however the further away in point tracked the less the x or y point changes compared to a closer point. Not sure how to get around this concept, will try out a couple of things and see what comes up.

To check my theory I will look at the camera movement from 2 different background points and see if they have different x and y movements, proving depth distorts the results.

A picture containing text, picture frame

Description automatically generated

A picture containing text

Description automatically generated

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

Very similar path as 2 stationary background points (one with a slightly lesser depth) as evident in the 2 graphs above this, as well as looking at the depth change in the vibration graphs is slightly different.

As seen in the vibration graphs in x and y their results are different due to their change in depth. The resultant vibration removed graph should be a flat zero line as we have tracked 2 background stationary points, however it is very much not zero showing the error in removing camera vibration thus proving my theory.

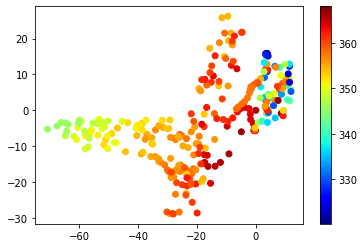
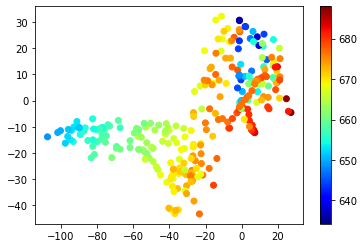
Taking a new video of vibration in the camera and with no moving object but 2 stationary objects at different depths to see if we can replicate the camera movement from these 2 points:

File name: “Vibration/l9.mp4” and “r9.mp4”

Results/Findings:

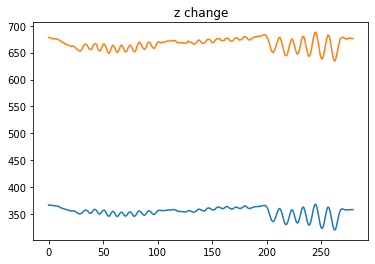
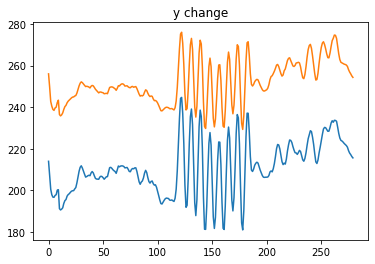
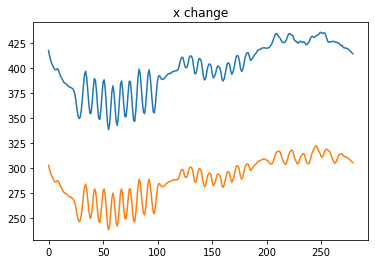
point 1 -> (idx=17) is further away point on deo can

potin 2 -> (idx=16) is closer point on book



Path of points in x, y and z direction. Can see that they are very similar in shape, but one has a much further depth (deo can point).

Comparing the x, y and z pixel coordinates: (look very similar)



What if we subtract the initial point difference from the furth point and apply that number to all frames?

A picture containing text, pencil, writing implement, flag

Description automatically generated

Don’t line up exactly, they have the same shape but a slightly different scale due to the object being at different depths.

Histogram

Description automatically generated

Looking only at the movement in the first few frames where the object changes in x direction. Blue being the book and orange the deo can (further back).

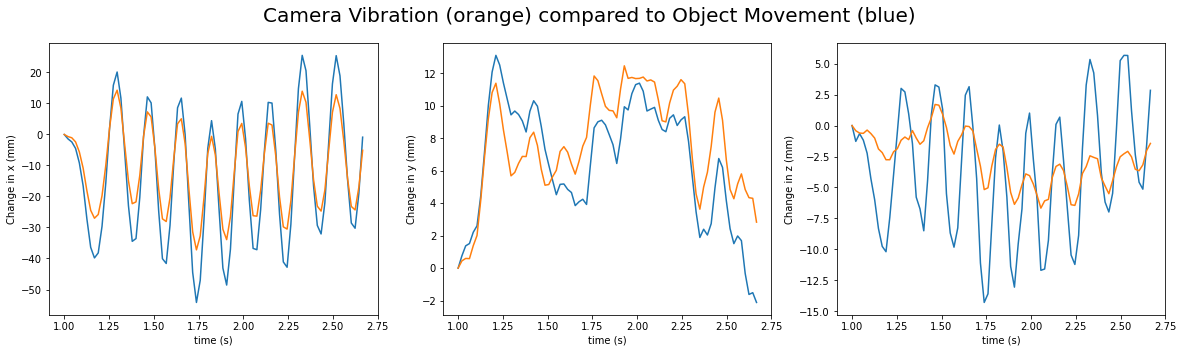
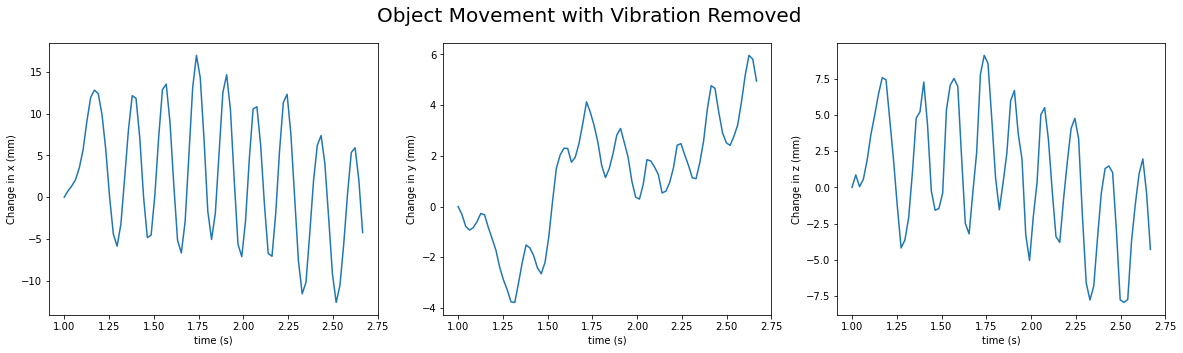
Shape, arrow

Description automatically generated

Scaled by their depth does not bring good results either. x1[0] \* z1[0] … etc.

However the scale is much neater now, before the change in x1 and x2 was not consistent the whole way through but now the results of the 2 are quite consistent. Before there were parts where orange was larger and lesser than blue but here the orange is always lesser than blue which makes more sense as the object closer being the blue (book) will move much more due to depth distortion. So this result makes sense and from this point we must find some sort of scale factor that makes the vibration similar.

This same idea happens when we convert from pixel to mm as this process takes depth into account:



Therefore, the measurements shouldn’t be affected by depth distortion as that is accounted for when converting to real measurements.

Maybe the change in x for the background point must be somehow converted back into the true change in x of the camera?

Can try some set experiments with all known dimensions: move the camera between a certain length in 1 direction and see how 2 background points change.

After running some quick experiments, if your camera stays parallel to a flat surface the distance in mm that the object travels will be the same as how much the camera moves. However, any slight rotation/tilt in the camera can quickly change the result.

3D camera tracking is much more complicated as we need to account for rotation.

Restart experiments but with very basic camera movement, only move the camera in the x direction and track camera movement, make sure objects are parallel to the camera.

A picture containing text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Using new video named l10.mp4 and r10.mp4 which was a controlled movement only on the x axis with no rotation involved

The results speak for themselves as the changes in the 2 x points are near perfect. And the change in y and z are very minimal. The error is points being at different heights will effect the depth and y coordinates changing slightly.

Try for z direction on video l11 and r11:

Chart, histogram

Description automatically generated

A picture containing text

Description automatically generated

Z displacement is slightly out due to incorrectly calculated depth at both points, depth will be the worst performing out of the 3 directions. X change is very similar again this time due to the x axis being controlled again. However, y will change very differently due to the angle of movement and the different heights of the points.

Can see the mappings of the points of each frame are very similar, just differing in its depths (but at the same scale) which makes sense.

