EE/CprE/SE 4910 WEEKLY REPORT #1

1/21/25 - 2/25/25

Group number 15

Vision Based Camera Motion Tracking

Advisor: Ashraf Gaffar

Team Members:
Andrew Gooding
Eric Wittrock
Isaac Kenyon
Will Ernatt

Weekly Summary

Our goal for this week was to familiarize ourselves with the problem at hand, camera motion tracking. We experimented with existing solutions (blender camera tracking tool, Structure from Motion, etc.) in search of inspiration for our own solution. Through this process we were able to gain a general idea of the strengths and weaknesses of each technique. We also used this week to set up our programming environments for the project.

To kick off the research and development process, we focused heavily on experimentation this week. This involved camera tracking experiments with existing software and rapid prototyping of useful computer vision algorithms.

Past Weekly Accomplishments

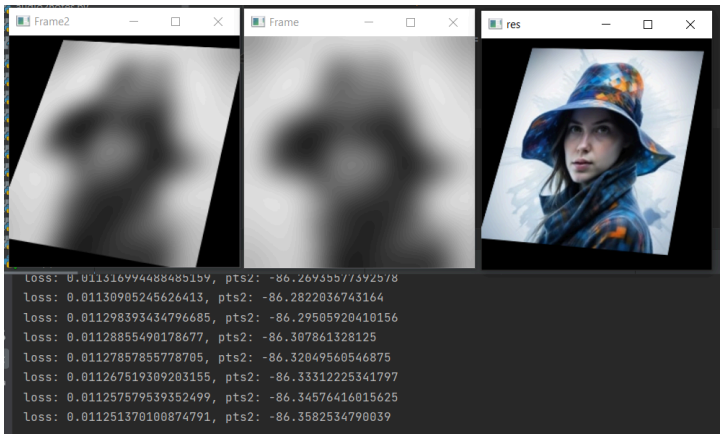
Eric Wittrock

1) Setup python environment and OpenCV
    - Read frame from video our team took as test footage
    - Process video data with canny edge detection

2) Estimate a perspective transformation on a given image.

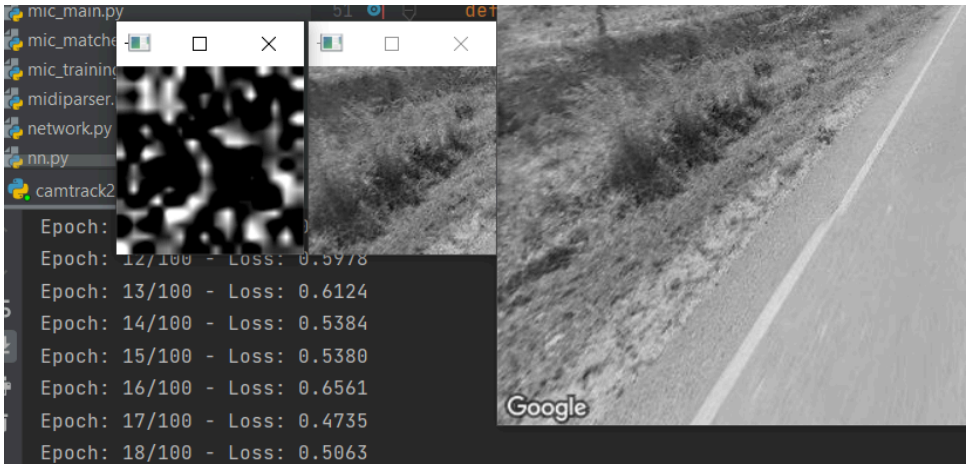
This was an experiment to try to find the transformation matrix used to warp an image given the original image and the deformed one. Using gradient descent, the original image was distorted with many small steps until it resembled the warped (target) image, thus providing us with the transformation in which the warped image underwent in the first step. If we figure out the transformation a region of space undergoes from one frame to the next, we may be able to

reconstruct the camera motion because a movement by the camera corresponds to a perspective warp on its view.



In the above figure, the center image is the original, non-warped image (the grayscale and blurring makes it easier for the gradient descent algorithm to re-align with the original while taking small steps). A perspective warp is applied to transform it into the leftmost image. Then the algorithm estimates this transformation by looking at the two left images and displaying the prediction using the rightmost image. Notice the leftmost and rightmost image are skewed in similar ways.

3) Represent a patch of an image in latent space so that the same image rotated different amounts appear identical



In the above figure, the leftmost image shows the latent space representation of the image immediately to the right of it (the input image). If the input image were to rotate, the latent image would remain roughly unchanged, but if the input image were to change locations, a change would be noticeable in its latent space mapping. This was done by training a neural network to recognize one image at different orientations as a single point in space and different images as separate points in space. This will help us track features as they move across a video because a camera may be rotating as it films.

Isaac Kenyon

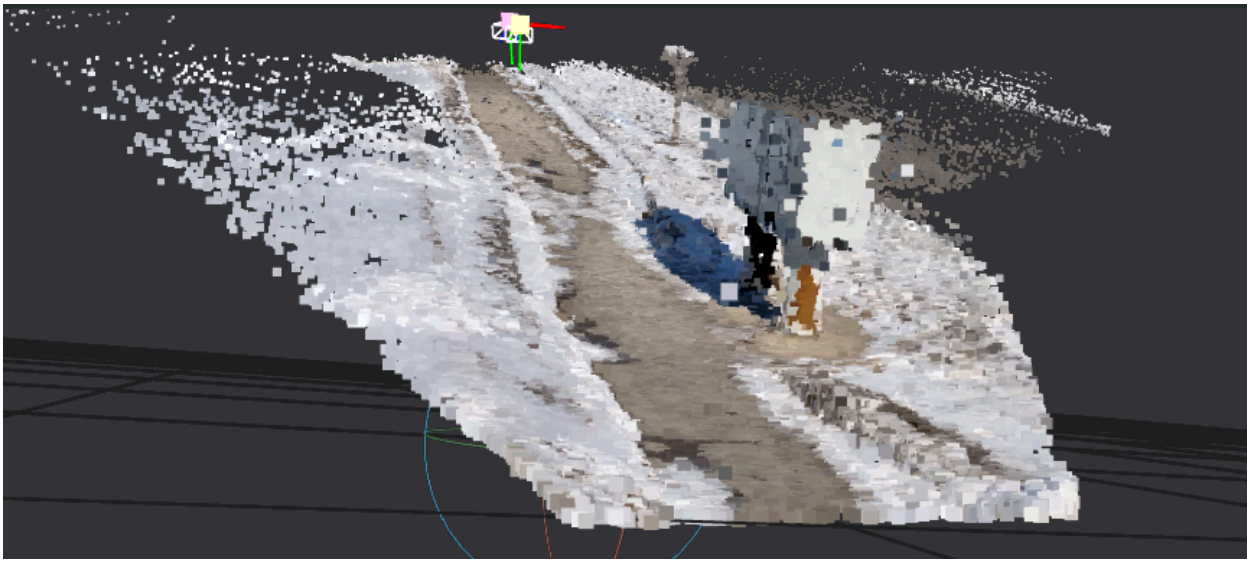
- Setup python environment for my Ubuntu machine using a virtual environment to allow ease of changing versions and not to have each member download all of the required packages on pip.
    - Started up the repository and created the main branch along with my own branch.
    - Worked on the main README to save important information needed for setting up and using the project
- Downloaded and started using blender
    - Did some research and implemented how to use python scripts inside of blender.
    - Learned how to import and export images of a video into blender with python scripts to do processing.
    - Implemented the ability to export the individual frames of a video showing the 3D object in the frame.
- Worked on an optical flow test program in python.
    - Implemented ability to export the optical flow data calculated to JSON and CSV data.

Will Ernatt

- Researched existing methods for Camera Motion Tracking
    - Found this method that uses photo scanning techniques to generate an accurate camera track: https://www.youtube.com/watch?v=1dhdEmGLZhY

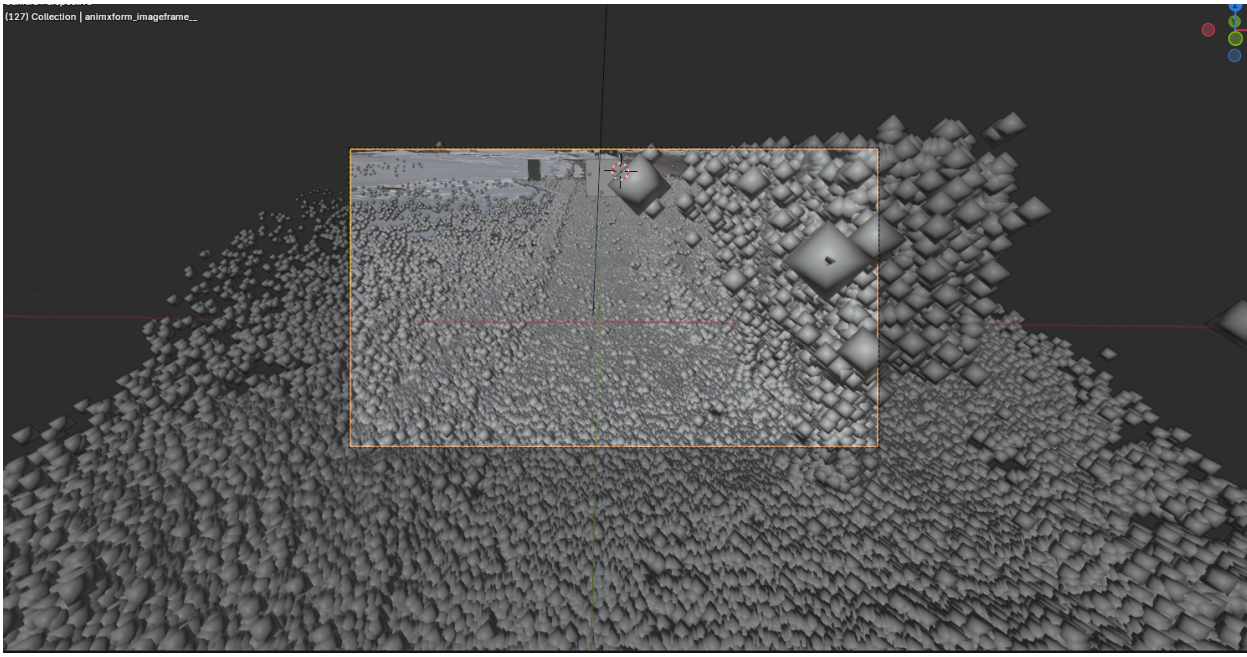
- Attempted the technique with my own recorded video.

- First frame of the video submitted to Meshroom (photogrammetry software).



- Colored point cloud generated by meshroom using Structure from Motion (estimates 3d space from 2d motion). This process also generates a camera

track



- Point cloud and camera pov when imported to Blender

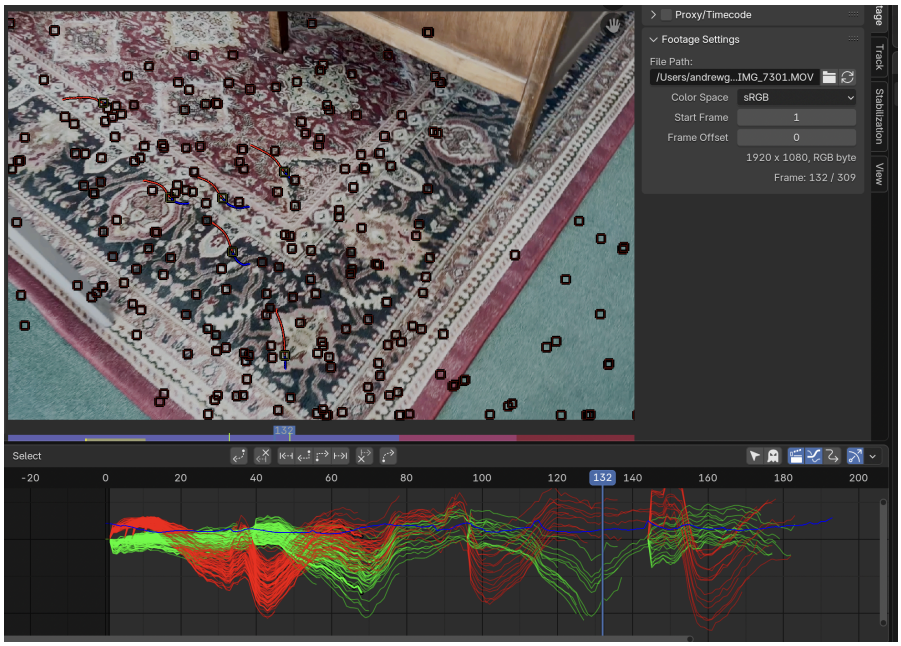
- First frame of the final composition, the 3d box seen on the sidewalk appears to match up exactly with my video. I did not need to manually adjust the camera track myself.
- This process took approximately 4 hours to generate a camera track for a ~500 frame video (19 seconds @ 30 fps). While it can be performed successfully by

someone with little to no experience in camera tracking it is very impractical for someone who wants to quickly experiment with different camera shots.

- Set up Python environment
- Updated blender
    - Watched tutorials on object tracking within blender.

Andrew Gooding

- Used blender camera tracking tool to gain a better understanding how it is used and how our product can be implemented in a better way that is more user friendly.
- Using the test footage we recorded, we ran a test on how error tracking works.



- This image shows how a camera is being tracked when moving around a carpet.
    - The goal of this would be to insert a 3-d object in the center of this video so that it looks like the camera was recording the image.
- Doing this I learned how to clean up how the camera is being tracked in the video.
- The waveforms on the bottom of the image correlate with the points on the carpet.
- I also watched a few videos so that I can understand how camera tracking works on other softwares as well.
- Here are the links to the videos:
    - Blender - https://www.youtube.com/watch?v=ui0JUHE12k8
    - Adobe - https://www.youtube.com/watch?v=UaWjvi3PvS8

Pending Issues

- **Eric Wittrock:** While experimenting with neural network architectures, my convolutional neural network (CNN) implementation didn't converge very well, but a dense neural network did. I will therefore be working with the latter from now on, but I may need to fix this problem later if I ever return to the CNN for its ability to handle large images efficiently.
- **Isaac Kenyon:** Blender script is really slow and tends to stall/crash the entire blender program. The optical flow program has errors when initial points of analysis either get covered by an object or fall off of the screen. This often causes a line to be placed across the screen that is incorrect.
- **Will Ernatt:** Several of my camera tracking experiments completely failed(using Meshroom, the photogrammetry software), which required me to experiment and understand the limitations of the software and the tracking method(mainly that video input must be more than a simple pan for the algorithm to work).
- **Andrew Gooding:** When inserting an object into blender the function kind of breaks. I believe this is because the waveform may have been cleaned a little bit too much so that there are not enough points to track the camera correctly.

Individual contributions

Name	**Individual Contributions**	**Hours This Week**	**Hours Cumulative**
Eric Wittrock	Built an algorithm to estimate perspective changes Built an algorithm to match patches of an image even if they have been rotated	8.5	8.5
Will Ernatt	Researched existing camera tracking techniques and ran camera tracking experiments of my own.	7	7
Andrew Gooding	Practiced with blender on using its camera tracking method to better	5	5

	understand how it is used and how our project will get rid of those problems.		
Isaac Kenyon	Implemented an optical flow program in python. Wrote a script that interfaces with blender in python. Setup the repository and started the initial commits and pushes.	7	7

Plans for the upcoming week

Andrew Gooding

- Continue using blender to gain a better understanding and start a python test environment to start implementing scripts into projects.

Eric Wittrock

- Use algorithm I built last week to estimate motion of ground at every pixel in a video

Isaac Kenyon

- Keep working on the optical flow and what other algorithms can use the output data of that program.
- Update the blender script so that it runs faster and hopefully make a template script plugin that can be used to implement parameters that can be inputted.

Will Ernatt

- Research the algorithms used in meshroom (an open source) project, to see if it is feasible to implement such a process within blender.