EE/CprE/SE 4910 WEEKLY REPORT #2

2/26/25 - 3/4/25

Group number 15

Vision Based Camera Motion Tracking

Advisor: Ashraf Gaffar

Team Members:
Andrew Gooding
Eric Wittrock
Isaac Kenyon
Will Ernatt

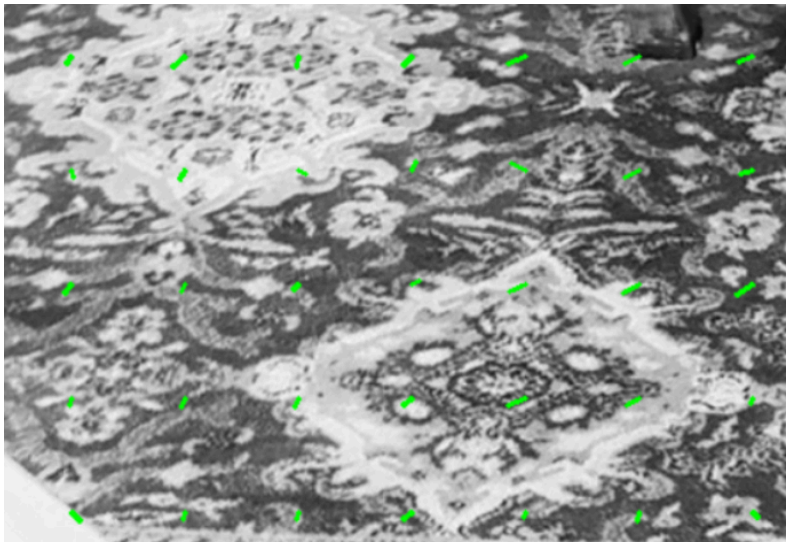
Weekly Summary

This week we continued experimenting with and researching optical flow algorithms and practiced using blender with user-made plugins. These experiments will help us evaluate whether or not these libraries(OpenCV, AliceVision, etc.) could be used for our final product, or whether they contain algorithms or principles that we could reuse for our own custom solution. Our goal as of now is to have our software run within blender as a plug-in(rather than a piece of standalone software) so we have begun experimenting with our own plugins in blender to assess the feasibility of this goal.

Past Weekly Accomplishments

Eric Wittrock

- Create an optical flow algorithm from scratch
    - Using the neural network I built last week, I was able to match similar patches of pixels between two frames.
    - The purpose of the neural network is to mitigate the differences from image comparison that result from the motion-induced perspective distortion of the next frame.
    - The procedure creates a uniform grid of sampling points, then looks at the next frame in the video and estimates the amount of movement at that point



The above image shows a demonstration of this optical flow algorithm. The green lines depict a vector field that shows the estimated motion of the the pixels

- Experimented with the OpenCV built-in optical flow algorithm
    - The OpenCV library has a built-in algorithm that behaves similarly
    - The results of this algorithm were good enough to build off of, but we will probably fall back to a custom solution later on for better results.

- Worked on the mathematics for reconstructing camera motion
    - Now that we have a crude estimation of point motion, we can use the data to calculate camera motion
    - I created a high-level formula for solving the depth of points given parallax, but haven't solved it with non-iterative methods. Currently this is just a linear algebra representation.

Isaac Kenyon

- Worked on the blender plugin to incorporate our python code into the blender software.
    - Looked into having two different types of processing, one for automatic object placement with predefined values set.
    - Another process where the user can access more advanced options to customize the output more.
    - Read and implemented some of the GUI features that blender has for plugins.
- Looked into some of the available hardware that Iowa State has to offer.
    - Would be used for possible machine learning training.

Will Ernatt

- For this week I researched existing frameworks and algorithms for camera tracking, specifically the ones provided by AliceVision (meshroom) and OpenCV.
    - Using AliceVision I was able to create a script to take a video as input to output a point cloud and camera track. This still takes around the same amount of time as Meshroom(several hours for a short video) so it is still a little impractical for quick prototyping.
    - Started looking at OpenCV to create a similar script but was unable to finish last week.
- Began writing blender plugins to practice.
    - Wrote a plugin to import a point cloud and camera track into blender.
    - Wrote a plugin to move an object in blender.

Andrew Gooding

- This week I continued working on blender to gain understanding of the scope of our project and how we can improve the blender plugin that already exists. I have also set up an openCV and python environment to begin scripting.
- I also did some research on other software that may do something similar to our project. I found that matlab has a motion tracking program available to use. Although we are not tracking motion of objects we could maybe see how this was implemented and maybe even see if we can make our program work for matlab as well.

<u>Pending Issues</u>

- **Eric Wittrock:** The Algorithm I created for finding pixel motion by comparing the embeddings of image segments was extremely slow. I will likely have to rewrite this entirely.
- **Isaac Kenyon:** Attempting to get a standalone window/tab in blender. So far I could get buttons on a sidebar but I would like more of a custom look.
- **Will Ernatt:** My AliceVision script frequently ran into issues where the frames obtained from the input video would become corrupted and unable to be used for the SfM algorithm. I'd like to figure out the cause of that problem.
- **Andrew Gooding:** An issue that I had faced this week was setting up the openCV environment on my Macbook. For some reason it initially was not allowing me to download openCV.

Individual contributions

Name	Individual Contributions	Hours This Week	Hours Cumulative
Eric Wittrock	Created an algorithm from scratch that finds the sparse motion of pixels in a video Experimented with the built-in OpenCV optical flow algorithm Worked on mathematics for translating point motion to camera path	6.5	15
Will Ernatt	Wrote an algorithm based on the AliceVision framework to calculate a camera path from a video using the Structure from Motion algorithm. Practiced with blender plugins.	6	13
Andrew Gooding	This week I was not able to work on the project as much as I would like due to planning travel for a convention, but I was able to get essential things needed for our project setup on my computer that I will be able to work on while I am away.	4	9
Isaac Kenyon	Blender plugin implementation for user input. Using the	5	12

	features of blender to make a GUI interface so that the users are able to use the programs by clicking with the mouse.		

Plans for the upcoming week

Andrew Gooding

- Continue using blender to gain a better understanding, continue to test python scripts now that the environment is set up and to start looking into how I can improve how errors are detected when using the camera tracking software.

Eric Wittrock

- I will compute camera motion using the built-in OpenCV optical flow algorithm. I will temporarily ignore things like lens distortion and point error to get a proof of concept as quickly as possible.

Isaac Kenyon

- Get the blender plugin interface done and start looking into the processing of eliminating bad data points from optical flow calculations. Also look at being able to import JSON data files into blender to possibly use.

Will Ernatt

- Debug my AliceVision script. Write an OpenCV script to use the OpenCV SfM algorithm to compare speed and usability to AliceVision/Meshroom
- Write a blender plugin to automatically set up the point cloud and camera track from Meshroom in blender.