Synthetic Data Pipeline for Pose Estimation (Milestone 3)

Group: Nathan Pichette, William Stern, Stephane Baruch, Hanibal Alazar
Faculty Advisor: Dr. Ryan White

Goal and motivation

- Create a program that can generate synthetic training data for training pose estimation for satellites in space where the user can modify different parameters such as lighting, satellite, and motion and create a benchmark neural network model using this data to estimate poses.

Approach

- The user can easily generate a lot of satellite training data for use in neural networks.
- The user can configure things like the type of satellite, the flight path of the satellite, lighting, background, and various other things
- The user can generate a lot of videos for use in machine learning at one time.

Algorithms and tools

- Blender
- Bpy
- Toml
- Ensurepip
- Neural network libraries (Tensorflow or Pytorch)

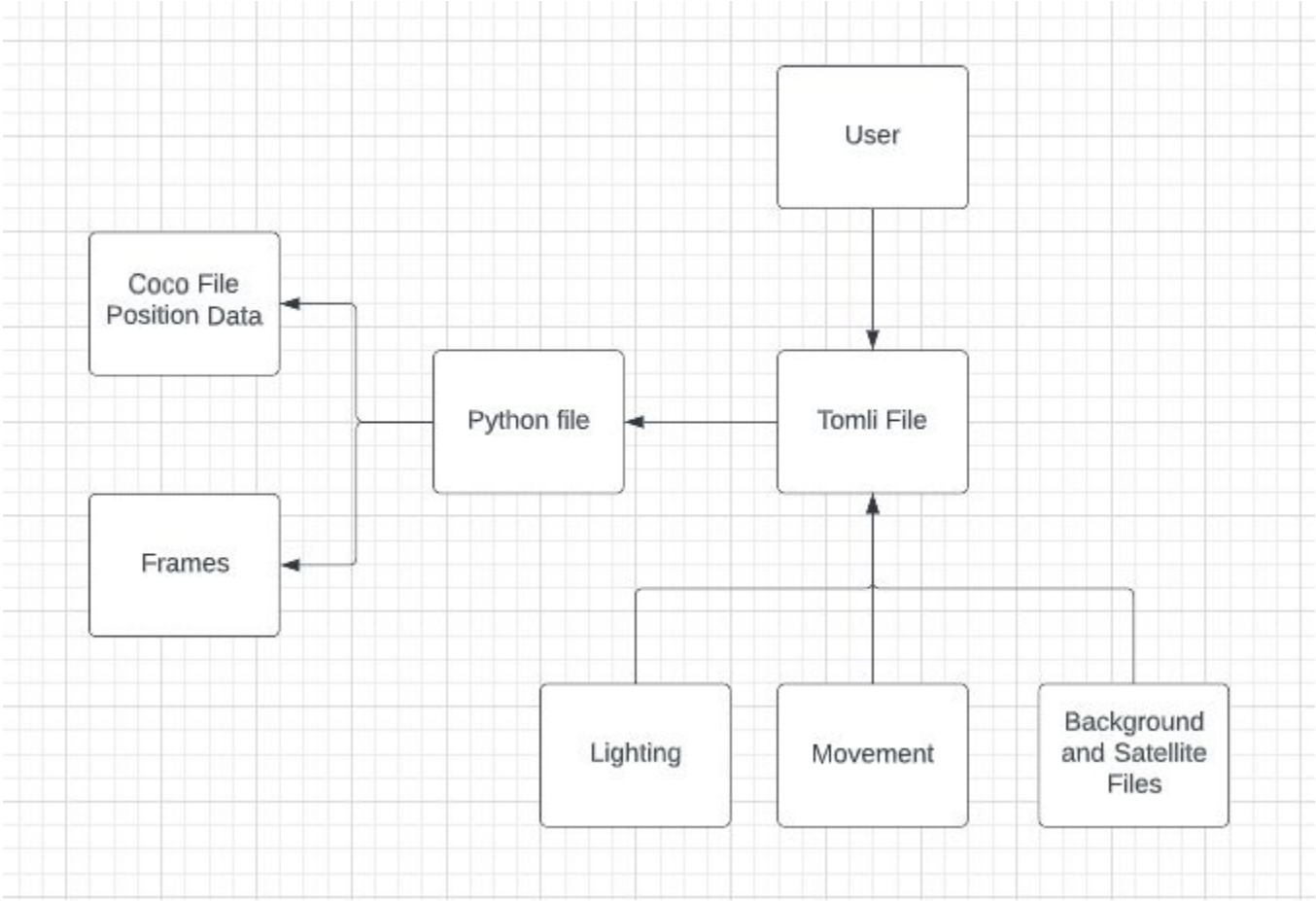
Novel features

- This program will allow for easy and automated creation of synthetic animated satellite data.
- Our system lists satellite position data that can be used to verify the machine learning algorithms output.
- Our program can also be used to test low light conditions previously unworkable with a past iteration of a similar system

Technical Challenges

- Porting to different OS and working with different types of hardware
- Working with blenders custom version of python
- Learning neural networks

Design


User
Coco File
Position Data
Python file
Tomli File
Frames
Lighting
Movement
Background
and Satellite
Files

Evaluation

- Survey Dr. White's machine learning team they will grade the ease of use and quality of output on a scale of 1-5
- We will also evaluate the success of our synthetic data generation system based on the accuracy of how often the neural network that we create is able to predict the pose.

Progress Summary

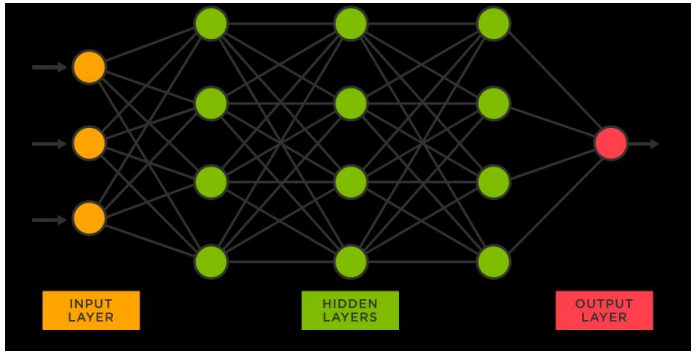
Task	Progress	To Do
1. Complex movement along path	70%	Change method so it is based on time instead of x
2. Enable movement interaction through configuration file	100%	
3. Implement compatibility among os	66%	Add linux compatibility
4. Extract poses as coco type annotations	100%	

Milestone 4

- Finish adding compatibility with linux
- Verify that the rotation and position angles are actually correct.
- Meet with Dr. White’s team to get their opinions and test functionality for users.
- Start learning neural networks
- Add another function for flightpath in TOML file so we can support non-linear changes in x.

Milestone 5

- Implement a neural network that can estimate the pose of the satellite based on the synthetic data that we have been generating.
- Generate a variety of data using different variables for training and testing



Milestone 6

- Improve neural network
- Prepare to hand off the project to Dr. White’s team
- Create poster for Senior Design Showcase
- Implement, test, and demo which features/modules
- Test/demo of the entire system
- Conduct evaluation and analyze results
- Create user/developer manual
- Create demo video

Milestone 4 Matrix

Task	William	Nate	Stephane	Hanibal
Validate that the angles generated are correct	33%	0%	33%	33%
Meet with Dr. White's team to get their opinions and test functionality for users.	25%	25%	25%	25%
3. Start learning neural networks	0%	33%	33%	33%
Finish adding linux compatibility	50%	50%	0%	0%
4. Add another function for flightpath in the TOML file so we can support non-linear changes in x.	0%	0%	50%	50%

Thank You!

Questions?