Software and Perception Team Tasks

Deadline: Midnight June 7, 2021

1 GENERAL INSTRUCTIONS

The following tasks are intended to check your skills in the field of Computer Vision and Path Planning. You would have been added to the MS Teams for the task round. You're encouraged to figure out the problems on your own, however, feel free to contact Agni Keyoor Purani, Neha Dalmia, Rishab Agarwal, Rohit Raj, Rohit Sutradhar or Satwik Chappidi, message anyone on MS Teams or just ask it in the MS Teams channel. The final code must be submitted on a github repo in C++ or Python along with appropriate documentation. The documentation format will also be provided to you. We had fun in creating these tasks and we hope that you'll enjoy them too. Do not copy code from friends as we will use automated plagiarism checkers. Also, if you use any library functions, you might be asked to explain in detail how it works. Ping us for more problems if you don't find this challenging enough. Good luck!

2 BASIC TASKS

2.1 OPTIMIZE ME

In the first task you have to use optimisation techniques to reduce the running time of the code provided to you. First download the zip file here. Before proceeding further, read the Instructions.md file which provides a lengthier explanation of the problem statement along with hints. Markdown(md) files can be viewed on Typora or on your code editor. A bash script to time your code has also been provided to you along with instructions on how to run it. Optimisation techniques can range from basic time complexity reduction to more advanced optimisations using parallel programming, locality of references and the like. Initially the code provided to you will not work for larger values of n, we will slide n from 4 to 1000 to test your code. Code working on large values of n in less time will fetch maximum points. Start with basic time complexity reduction and ease into more advanced methods to see how your running time reduces and have fun trying to achieve the minimum running time possible!

2.2 FACE DETECTION GAME

Have a look at the animation provided here. You have to make a game environment where you can play a similar game. Your face will replace Tom's face. For that you have to detect and track your face, assume your face to be a perfect circle. You'll have to keep a track of the ball as well. The ball should follow the laws of physics and the collisions can be assumed to be perfectly elastic. The game ends if the ball drops to the floor. **Note**:

1. You do not have to consider gravity, i.e., the ball comes down only after hitting the upper wall.
2. Not using built-in functions for face detection and/or tracking will fetch you extra points.

3 Intermediate Tasks

3.1 Path Planning and Computer Vision Puzzle

Hey adventurer! I heard you qualified for the task round for ARK - the best research group in IIT KGP. We have located a treasure but its hidden behind puzzles. You are the chosen one! Start off with the first step which will guide you till the treasure. Good luck!

How do I start the adventure?

Your first encounter is with an image called Level1.png. As you might know in a normal RGB scale the colour is represented by a number from 0 to 255 inclusive for each channel. We also know that an ASCII character can be represented as a binary number from 0 to 255 inclusive. So could the thin initial greyscale pixels of the image contain some useful information? Could it help us on what to do with the rest of the pixels in the image? There are a total of 4 files you would need for this task and are provided here. If builtin functions are used, you must be able to tell how they work, its limitations and the different parameters it takes up and how it affects its working.

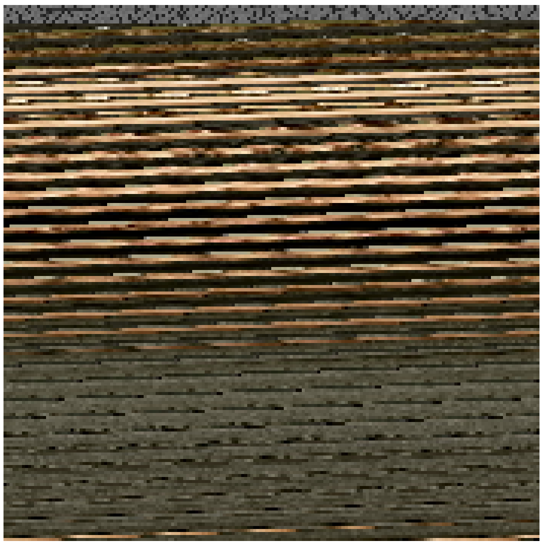

Figure 3.1: Level1.png file

4 Advanced Tasks

4.1 Tracking Multiple Objects

Consider a drone flying in an area. There are 6 ground stations in the area and on seeing the drone they provided you with the coordinates of the drone with respect to them. The exact coordinates of ground stations are also unknown. Your job is to localise the drone and predict its trajectory given that measurements provided by ground stations are noisy due to instruments. Although from experiments we know that noise is distributed according to gaussian.
Problem set 1: This is an easier version of the problem where you are provided with 3 CSV files (CSV1, CSV2, CSV3) with varying magnitude of noise and you have to predict the trajectory of the drone in all the three cases.
Problem set 2: This is a slightly difficult version of the problem where due to server issues some of the values reported by ground stations were modified and were erroneous. Finally you are again supposed to predict the correct trajectory of drone. For this also you are provided with a CSV file (CSV4).
For solving this problem, you need to learn about kalman filters (kalman filter) which are used to track an object from noisy data provided by some sensor like ground station in this case. Bonus points will be given to those who delve deep into the problem and try something fancier like a particle filter(particle filter), etc.
If you want to understand kalman filters you can try reading Kalman.

4.2 3D Tic Tac Toe Agent

Part 1: You have to write an agent to play the 2-D Tic-Tac-Toe game using minimax algorithm. Minimax algorithm basically is just an exhaustive search of your game space. You'll be using the gym-tictactoe environment. A skeleton for player-vs-player agent is provided at here.
Part 2: You will now have write an agent to play 3-D Tic-Tac-Toe. It will be harder to use the Minimax algorithm for this part as the search space is much larger. You will have to look into ways of optimising Minimax algorithm. If you want, you may implement more advanced methods like Q-Value Learning and Genetic Algorithms. Feel free to contact us if you have any trouble in getting the environment setup.

5 Helpful Links

- Install Ubuntu
- Basic Terminal Tutorial
- Git Online Practice