OVERVIEW

This project aims at making a bot that can travel between any two points in maze , also the path between the two points is the shortest possible one.

PROCESSES INVOLVED

1.Image acquisition and Processing

The top view of the maze is captured and stored as a jpg image. It is then converted into a binary image. Thereafter it undergoes dilation, erosion and skeletonization to get a clean image of track.

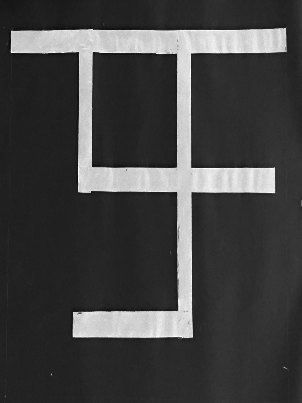


Fig 1. Image of the track

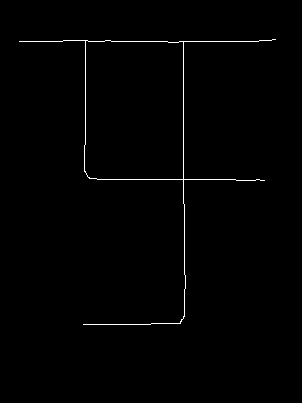


Fig 2. Skeleton of track

2. Finding the shortest path

A-Star algorithm is then applied to the skeleton to get the shortest path. This algorithm finds the shortest path by assigning a value f=g+h to every valid pixel. Here g=successor no. and h=approx. dist. of the neighbor from the destination. The f value of each successor determines whether to include it in shortest path or not. In the end we get the shortest path here.

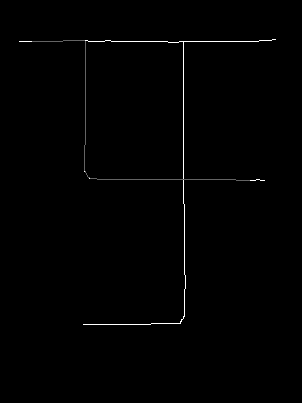


Fig 3. Shortest path (Grey colored)

3. Storing the path as a logical information

The shortest path obtained is then examined and the data is stored as L=Left, R=Right and S=Straight in an array as to what the bot should do whenever it encounters a junction or a turn.

4. Transmitting the path information to the bot.

The array consisting of L, R and S instructions is sent to the bot via HC-05 Bluetooth module.

SOFTWARE USED

Microsoft Visual Studio 2017. (C++) and the opencv library for image processing.

PROBLEMS FACED AND THEIR SOLUTIONS WE IMPLEMENTED

1.Disturbances in image

The image of the track had lot of noise, because of which further processing was very difficult.

Sol : Proper light exposure to be given to the maze while capturing it. Thresholding and blurring of image to be done before processing it.

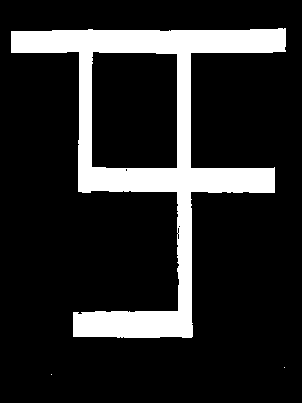
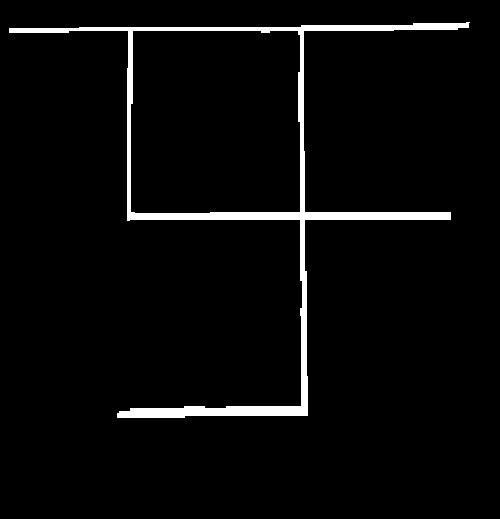
 

Fig 4. Threshold image of track Fig 5. Blurred image of fig 4.

2. Breaks and uneven thickness in skeleton image



FUTURE GOALS

An autonomous bot can be made that can travel through any terrain just by taking a top view image of it.