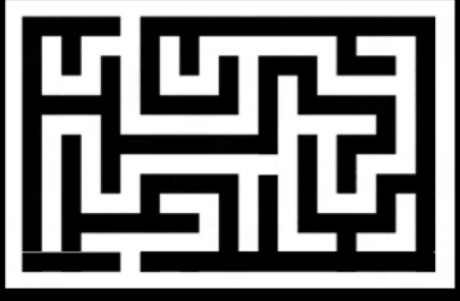




An Optical Maze Solver



Abdulrahman Majash, Malik Al-Basairi, Faisal Al-Fifi | Supervisor Dr. Ahmed Masoud | EE department King Fahd University of Petroleum and Minerals

Project Outline

- To construct a vision artificial intelligence that can solve any maze by connecting a human-specified start point to an end point.



- The main components in this system are:
 1. Data acquisition using webcam and mouse.
 2. Image processing, using edge detection.
 3. artificial intelligence using Harmonic Potential Field.

Data Acquisition

- Maze image is acquired using a webcam



Start and end points acquired using a mouse.

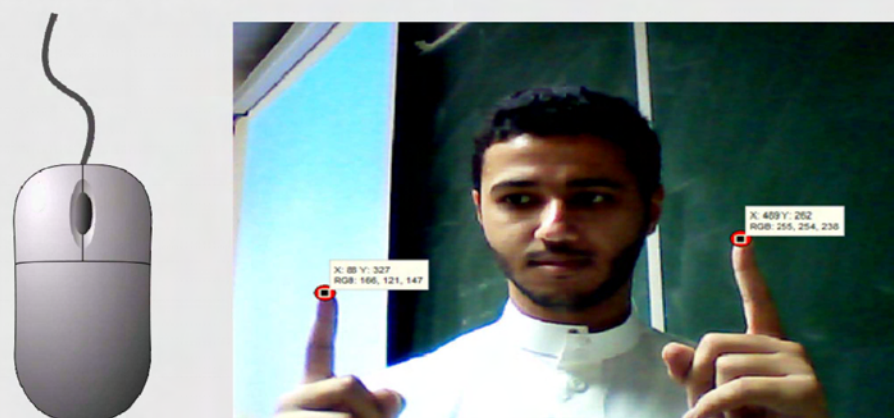
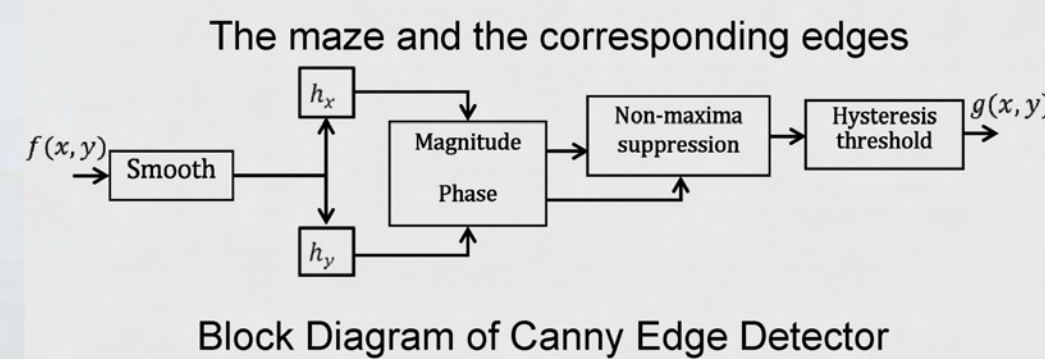
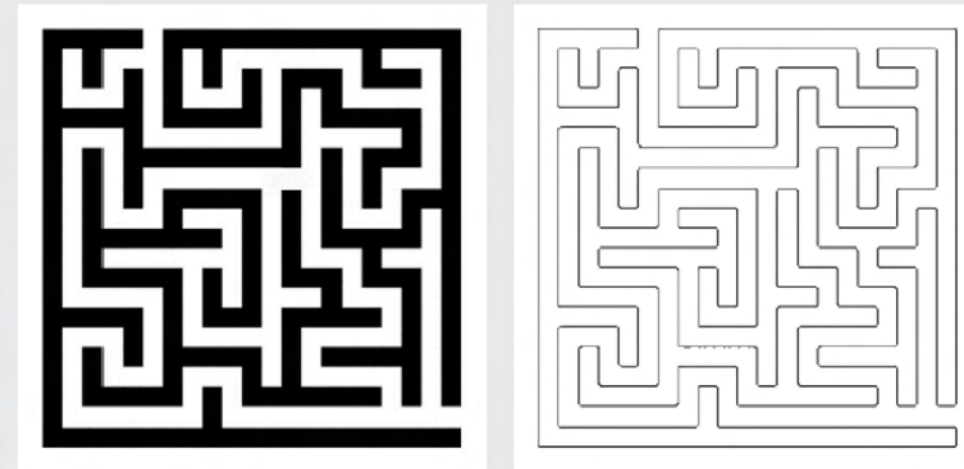


Image Edge Detection

- An edge detector (canny edge detector) is used to segment the image into regions occupied by obstacles and free regions where movement is allowed.



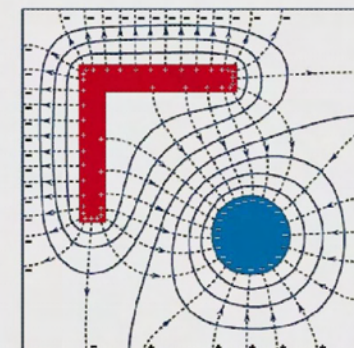
EM-Based Maze Solver

- Electromagnetics is used to build the maze solver. Motion is generated by a current moving in a conducting sheet. The obstacles are considered as insulators with voltage = 1. Free space is considered as a conductor. The target point is set to 0 volts.

Solve: $\nabla^2 V = 0, \quad X, Y \in \Omega(\text{free space})$
 Subject to: $V(X_T) = 0$ (Target point)
 $V(X) = 1 \forall X \in \Gamma$ (Obstacles)

$$\begin{pmatrix} X_{i+1} \\ Y_{i+1} \end{pmatrix} = \begin{pmatrix} X_i \\ Y_i \end{pmatrix} - \begin{pmatrix} \frac{\partial V}{\partial x} \\ \frac{\partial V}{\partial y} \end{pmatrix}$$

With $\begin{pmatrix} X_1 \\ Y_1 \end{pmatrix} = \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix}$

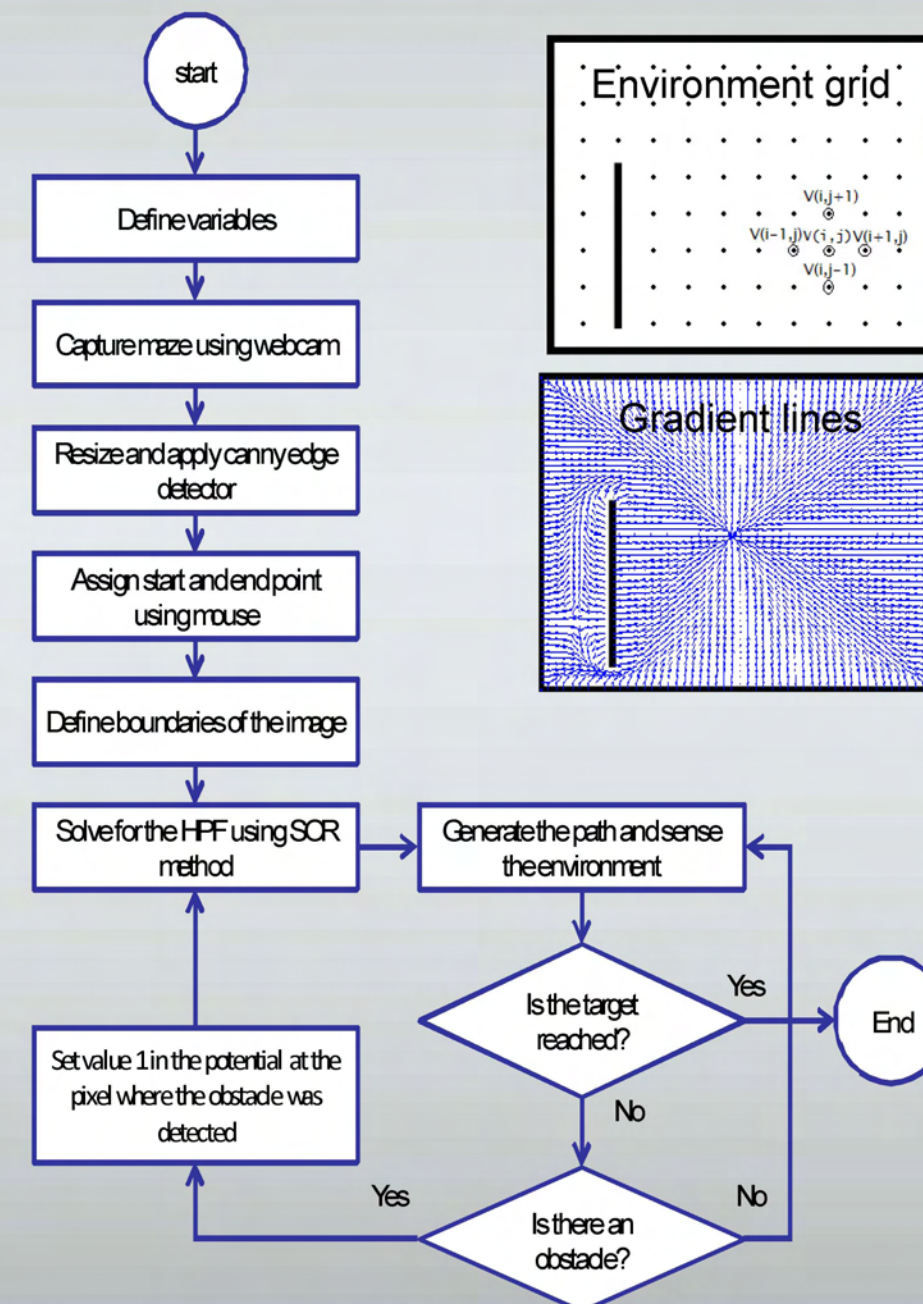


Numerical Solver

- The FDM method is used to solve the Laplace equations using MATLAB®

$$\nabla^2 V = 0$$

$$V_{i,j} = \frac{1}{4} (V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1})$$



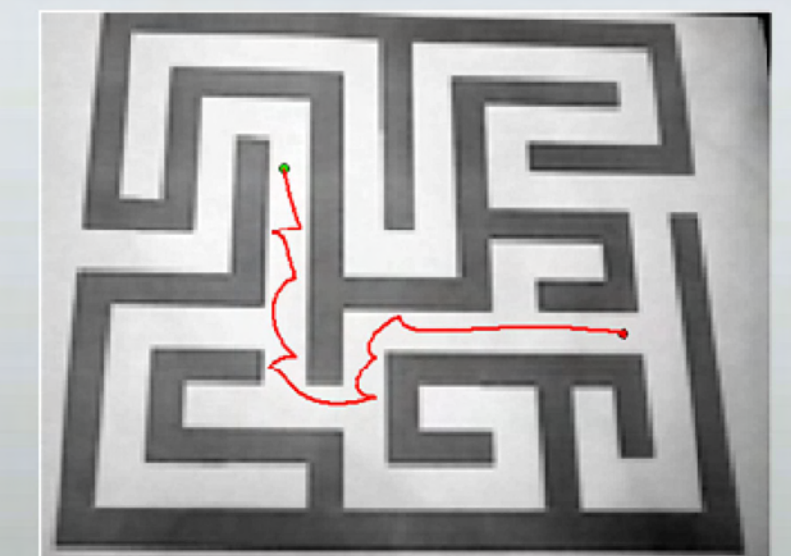
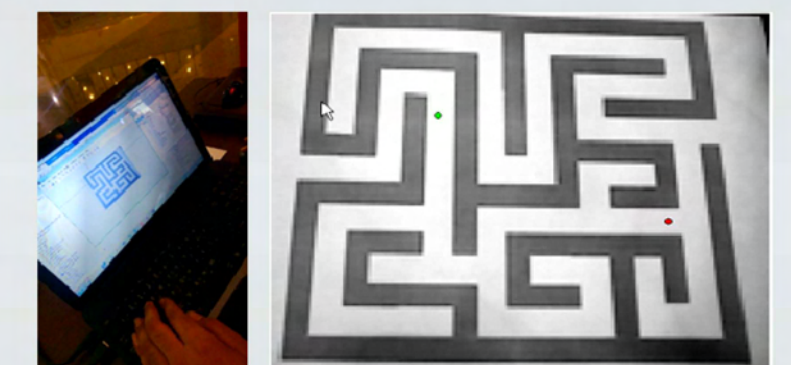
Flowchart of the solver

The Vision AI System

The above three components are combined to produce the system that will acquire a maze using a webcam. Allow a human operator to choose start and end points. Then, generate an obstacle-free path from start to end.



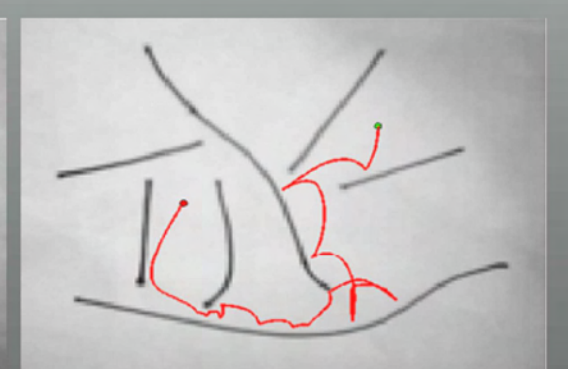
- The system acquires the maze image then generates a path between the selected points using only locally sensed obstacle boundaries



- If the system uses full information about the environment, the path generated will be much better



Maze solving using full information



Maze solving using partial information