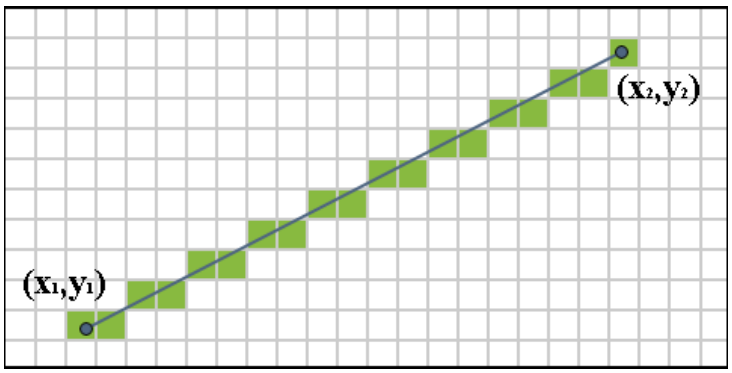
Occupancy Grid Map

1. จงสร้างโปรแกรม Matlab สำหรับสร้าง Bresenham's line

1.1 หลักการทำงานของ Bresenham algorithm มีดังนี้

Bresenham algorithm นั้นใช้สำหรับการหาพิกัดที่อยู่ระหว่างจุด 2 จุด (ดังแสดงในรูปที่ 1 ที่จุด x_1,y_1 และ x_2,y_2) ในที่นี้คือตำแหน่งจากหุ่นยนต์ไปที่สิ่งกีดขวาง ถ้าหากมีการลากเส้นตรง (ในทางปฏิบัติเส้นตรงนี้เปรียบเสมือน Lidar sensor) ระหว่างตำแหน่งดังกล่าวจะสามารถทราบพิกัด x , y ที่เส้นตรงตัดผ่านทั้งหมด (ดังแสดงในรูปที่ 1 ช่องสีเขียว) เพื่อนำไปใช้ประโยชน์ในการอัพเดท Grid map ได้ว่าตำแหน่งใหนว่างหรือมีสิ่งกีดขวาง โดยจะใช้วิธีดังกล่าวในข้อ 1.3



รูปที่ 1 แสดงหลักหาพิกัดระหว่างจุด 2 จุดด้วย Bresenham

1.2 จากหลักการดังกล่าวมาข้างต้นสามารถนำมาสร้าง Function ของ Bresenham ในโปรแกรม Matlab "`bresenham_line(x1,y1,x2,y2)`" ดังแสดงในรูปที่ 2 (จากรูปดังกล่าวแสดงบางส่วน สามารถดูตัวเต็มได้ที่ "`bresenham_line.m`")

```
function B_line = bresenham_line(xs,ys,xe,ye)
%BRESENHAM_LINE Summary of this function goes here


xs = round(xs); ys = round(ys); xe = round(xe); ye = round(ye);
dx = abs(xe-xs); dy = abs(ye-ys); sleep = abs(dy) > abs(dx);

if sleep
    t=dx;dx=dy;dy=t;
end

if dy == 0
    q = zeros(dx+1,1);
else
    q =[0;diff(mod([floor(dx/2):-dy:-dy*dx+floor(dx/2)]',dx))>=0];
end

if sleep
```

รูปที่ 2 แสดงการสร้าง Bresenham function

Occupancy Grid Map

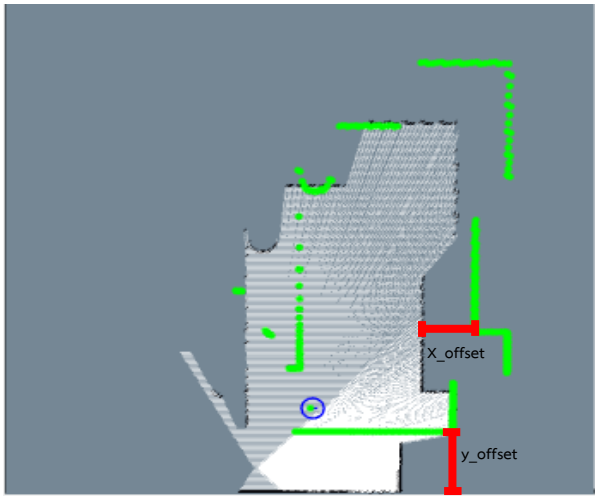
1.3 เมื่อได้ข้อมูลจาก Function ของ Bresenham แล้ว ต้องนำข้อมูลดังกล่าวไปอัพเดทแผนที่โดยสามารถเขียนเป็นโปรแกรมได้ดังแสดงในรูปที่ 3 การทำงานของโปรแกรมส่วนนี้ คือ รับข้อมูลพิกัด x,y ทั้งหมด (ระหว่างจุด 2 จุดที่คำนวนได้จาก Function ของ Bresenham) จากนั้นแทนลงไปในแผนที่ (ในโปรแกรมคือ เมตริกซ์ C) เมื่อเทียบกับข้อมูลตำแหน่งของหุ่นยนต์ ข้อมูลที่ได้จาก Bresenham (ในโปรแกรมคือ เมตริกซ์ B_line) จากโปรแกรมดังแสดงในรูปที่ 4 No.1 ตรวจสอบว่าค่าไหนคือค่าสุดท้ายที่ได้จาก Bresenham line เนื่องด้วยจุดดังกล่าวคือจุดที่มีสิ่งกีดขวาง แทนจุดนั้นด้วย 0 (แสดงผลเป็นสีดำ) และยังทำการตรวจสอบเพิ่มว่าถ้าจุดถัดไปของจุดนั้นเป็นสีดำ ให้เปลี่ยนจุดถัดไปของจุดนั้นเป็น 0.5 (แสดงผลเป็นสีบรอนซ์) และเปลี่ยนจุดนั้นเป็น 1 แทน รูปที่ 4 No.2 ทุกจุดนอกเหนือจากจุดสุดท้ายกำหนดให้เป็น 0 (แสดงผลเป็นสีบรอนซ์) นอกจากว่าจุดนั้นเคยเป็น 1 (สีดำ) มาก่อนให้แทนเป็น 1 (สีดำ) เหมือนเดิม ค่า Offset (x_os, y_os) ใช้แก้ปัญหาในกรณีทำการแปลงหน่วยและปัดเศษข้อมูล ทำให้เกิดระยะ offset สะสม ดังนั้นต้องมีการใส่ระยะชดเฉย ค่าที่ใช้หาได้จากการทดลองรันโปรแกรมจากนั้นวัดระยะที่ offset ออกไปในแนว x,y ดังแสดงในรูปที่ 4

```
%% Upgrade Occupancy gridmap %%
for e = 1:size(B_line,1)
    for i=B_line(e,1)
        for j=B_line(e,2)
            if sum(abs([i,j]-B_line(end,:))) == 0
                if C((j+y_os)+1,(i+x_os)+1) == 0
                    C(j+y_os,i+x_os)= 0.5;
                else
                    C(j+y_os,i+x_os)= 0;
                end
            else
                if C(j+y_os,i+x_os)== 0
                    C(j+y_os,i+x_os)= 0;
                else
                    C(j+y_os,i+x_os) = 1;
                end
            end
        end
    end
end
```

No.1

No.2

รูปที่ 3 แสดงโปรแกรมในส่วนที่ใช้ในการอัพเดทแผนที่



รูปที่ 4 การหาระยะ offset

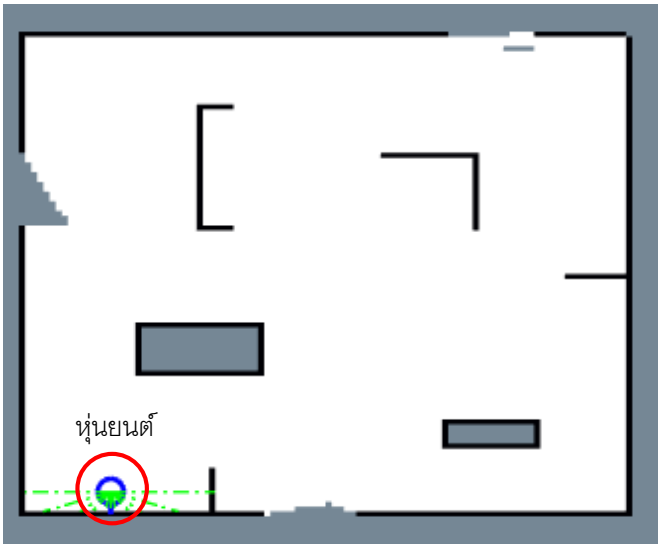
Occupancy Grid Map

2. จงดูตัวอย่าง Code matlab ในไฟล์ “main_ocgm_Xz_only_no_bresenham.m” ที่ให้จากนั้นนำโปรแกรมที่เขียนขึ้นเองในข้อที่ 1 มาใช้สำหรับสร้าง Occupancy map ตามข้อมูลที่ดึงจากไฟล์ “sample_data.mat”

2.1 การสร้าง “Occupancy map” นั้นสามารถสร้างได้โดย การสร้างเมตริกซ์ที่มีขนาดอย่างน้อยเท่ากับแผนที่จริงหรือมากกว่า ทำการสร้างเมตริกซ์ C กำหนดค่าเริ่มต้นของเมตริกซ์ C ให้เป็นสีบรอนซ์ที่มีค่าเป็น 0.5 เมื่อโปรแกรมเริ่มทำงาน หุ่นยนต์จะเริ่มเคลื่อนที่ จากเริ่มต้นไปที่เป้าหมายที่กำหนด โดยใช้พิกัดการเคลื่อนที่ ที่ได้จากไฟล์ “**sample_data.mat**” อยู่ในรูปเมตริกซ์ X ภายในจะมีข้อมูลตำแหน่งและมุมการเลี้ยวของหุ่นยนต์ (x,y,th) และยังมีข้อมูลเกี่ยวกับ Lidar sensor ที่มีระยะจากหุ่นยนต์ไปถึงสิ่งกีดขวางและมุมของระยะเทียบกับหุ่นยนต์จัดเก็บมาให้อยู่ในรุปของเมตริกซ์ z

จากนั้นเมื่อได้ข้อมูลของ Lidar sensor จากเมตริกซ์ z จะสามารถหาพิกัดระหว่างจุดที่หุ่นยนต์อยู่และสิ่งกีดขวางได้โดยใช้ “bresenham_line function” เมื่อได้ตำแหน่งมาแล้ว เขียนโปรแกรมให้อัพเดทค่าในเมตริกซ์ C จากพิกัดที่ได้จาก bresenham_line โดยกำหนดให้ค่าตัวสุดท้ายเป็น 1 (เป็นสีดำ) หรือถ้าจะให้เส้นหนาขึ้นป้องกันหุ่นยนต์ชนกับผนัง ต้องกำหนดให้สามค่าสุดท้ายเป็น 1 (เป็นสีดำ) นอกเหนือจากนั้นกำหนดให้เป็น 0 (สีขาว) เมื่อหุ่นยนต์เคลื่อนที่ตามที่กำหนดแล้วจะได้แผนที่ (ดังแสดงในรูปที่ 5)

สามารถแสดงการทำงานได้ดังวีดีโอแนบ “Occupancy_gridmap_sample.MP4” แสดงการทำงานด้วยโปรแกรม Matlab ดังไฟล์แนบ “occupancy_gridmap_sample.m”

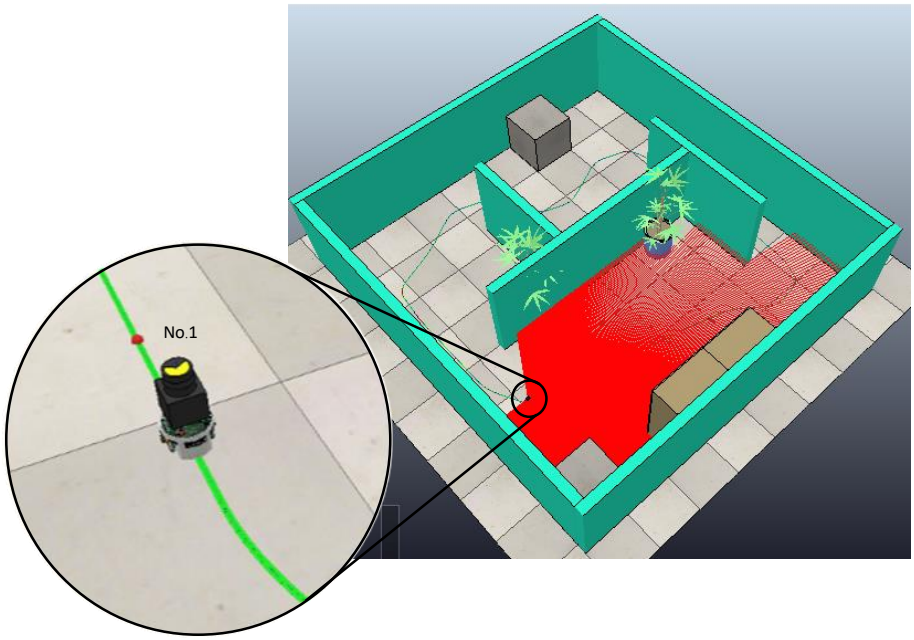


รูปที่ 5 แผนที่ที่ได้จากโปรแกรมดังกล่าวมาข้างต้น

Occupancy Grid Map

3. ดูตัวอย่างไฟล์ vrep-OCGM.ttt ซึ่ง Scene ที่รันในโปรแกรม Coppeliasim ทำการรัน จากนั้นรันไฟล์ Matlab ชื่อ get_data_from_vrepsim.m (ก่อนรันให้ copy ไฟล์ remApi.m, remoteApi.dll และ remoteApi.dylib ไปไว้ใน directory เดียวกัน) ซึ่งโปรแกรมดังกล่าวจะทำการเก็บข้อมูล 2D laser scanner จากหุ่นยนต์ในโปรแกรม Coppeliasim และเซฟเป็นไฟล์ Matlab ชื่อ data_vrep.mat เมื่อเข้าใจหลักการ ให้ทำการสร้าง Scene ของตัวเองขึ้นมาใหม่เองในไฟล์ Coppeliasim จากนั้นนำข้อมูลที่ได้ไปสร้าง Occupancy grid map จากโปรแกรมที่สร้างขึ้นในข้อ 2

3.1 จากโจทย์ทำการสร้าง Scene เพื่อใช้ในการจำลองการสร้างแผ่นที่ โดยใส่ผนังและเฟอร์นิเจอร์ต่างๆ เข้าไปที่สำคัญสร้างหุ่นยนต์ (ดังแสดงในรูปที่ 6 No.1) ติดตั้ง Lidar sensor และกำหนดเป้าหมายการเคลื่อนที่ให้กับหุ่นยนต์ให้เคลื่อนที่ไปให้ทั่วห้องเพื่อให้ได้ข้อมูลที่สมบูรณ์ที่สุด (ดังแสดงในรูปที่ 6)

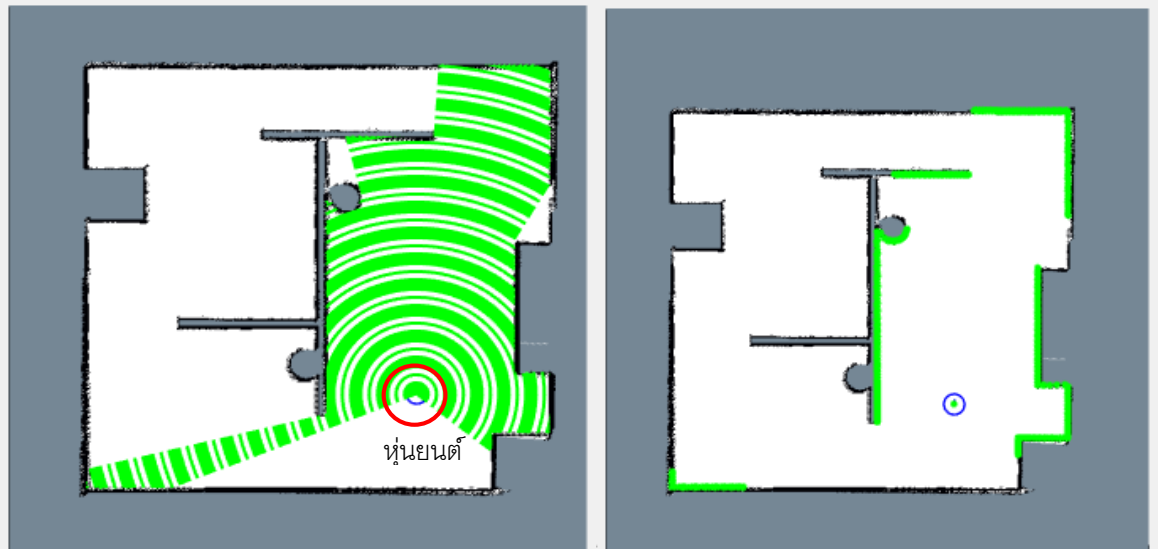


รูปที่ 6 แสดงการสร้าง Scene

3.2 รันโปรแกรม Coppeliasim (ดังไฟล์แนบ “map_vrep.ttt”) ที่สร้างขึ้นจากนั้นรันโปรแกรม Matlab (ดังไฟล์แนบ “get_data_from_vrepsim.m”) เพื่อใช้ในการดึงข้อมูลที่ได้จาก Lidar sensor จากโปรแกรม Coppeliasim มาเก็บไว้ในไฟล์ “data_vrep.mat” เพื่อเป็นข้อมูลในการสร้าง Occupancy grid map ในลับดับถัดไป ควรกำหนดระยะเวลาให้เหมาะสมในโปรแกรม Matlab เพื่อกำหนดขนาดของข้อมูล และให้หุ่นยนต์เดินครบรอบเป็นอย่างน้อย

Occupancy Grid Map

3.3 สร้าง Occupancy grid map จากข้อมูลที่เก็บไว้ในไฟล์ “data_vrep.mat” ซึ่งข้อมูลที่ถูกเก็บไว้นั้นได้อธิบายแล้วในข้อ 2.1 เมื่อโปรแกรมทำงานจะสามารถสร้างแผนที่ได้ดังแสดงในรูปที่ 7 No.1 และจากรูปดังกล่าวสีบรอนซ์คือพื้นที่ ที่ไม่สามารถตรวจจับได้ สีดำสิ่งกีดขวาง สีขาวพื้นที่ว่าง จากรูปที่ 7 No.1 พบว่าสามารถลดภาระแสดงผลของ Lidar sensor เพื่อให้ทำงานได้เร็วขึ้นดังแสดงในรูปที่ 7 No.2



รูปที่ 7 แสดง “Occupancy grid map”

การทำงานของโปรแกรมสามารถแสดงได้ดังไฟล์แนบ “occupancy_gridmap_vrep.m” และวีดีโอแสดงการทำงานของโปรแกรมแสดงดังไปแนบ “occupancy_gridmap_vrep.MP4”