

```

% Clears variables
clear

% Load dataset
load('reeeee.csv')
x = reeeee(:,1);
y = reeeee(:,2);

% Create data matrix in coordinate form (x,y)
fulldata = [x, y];
% Slims down data matrix by taking out every 25th element
startingdata = fulldata(2:20:end,:);
data = fulldata(2:20:end,:);
% Declaration of variables
bestInliersSoFar = 0;
bestPoint1SoFar = [0 0];
bestPoint2SoFar = [0 0];
bestTestLineSoFar = [];
endpoints = [0 0 0 0];
newpoints = [0 0];
inlierpoints = [0 0];

% RANSAC algorithm
% Number of times a pair of points are randomly selected
n = 4000;
% Distance Threshold to line
d = 50;

while length(data) > 2 % goes through loop until there only two points left in dataset
    % Clear variables after each iteration of loop
    newpoints = [];
    TrueXmax = [];
    TrueXmin = [];

    for i = 1:n
        inlierpoints = [];
        % Randomly selects two points
        point1 = data(randi([1, length(data)]),:);
        point2 = data(randi([1, length(data)]),:);
        % Prevents points from being the same
        if point1 == point2
            point1 = data(randi([1, length(data)]),:);
            point2 = data(randi([1, length(data)]),:);
        end
        % Find the tangent vector
        That = (point2 - point1) ./ vecnorm(point2 - point1);
        % Find the normal vector
        Nhat = ([-That(2) That(1)]);
        inliers = 0;

        % Test every point against the line
        for j = 1:length(data)
            % Finds perpendicular distance from a point to line
            r = data(j,:) - point1;
            dist = dot(r,Nhat);

```

```

        % Counts number of inliers
        if abs(dist) <= d
            inliers = inliers + 1;
            % Creates a matrix of all the inlierpoints
            inlierpoints = cat(1,inlierpoints,[data(j,1),data(j,2)]);
        end
    end

    % Finds which set of points has the largest amount of inliers
    if inliers > bestInliersSoFar
        bestPoint1SoFar = point1;
        bestPoint2SoFar = point2;
    end

end

% Finds true endpoint of line
% Calculates max X and min X point of all the inliers
a = max(inlierpoints);
b = min(inlierpoints);

TrueXMax = a(1);
TrueXMin = b(1);
for t = 1:length(inlierpoints)
    if inlierpoints(t) == TrueXMax
        maxendpoint = inlierpoints(t,:);
    end
end
for u = 1:length(inlierpoints)
    if inlierpoints(u) == TrueXMin
        minendpoint = inlierpoints(u,:);
    end
end

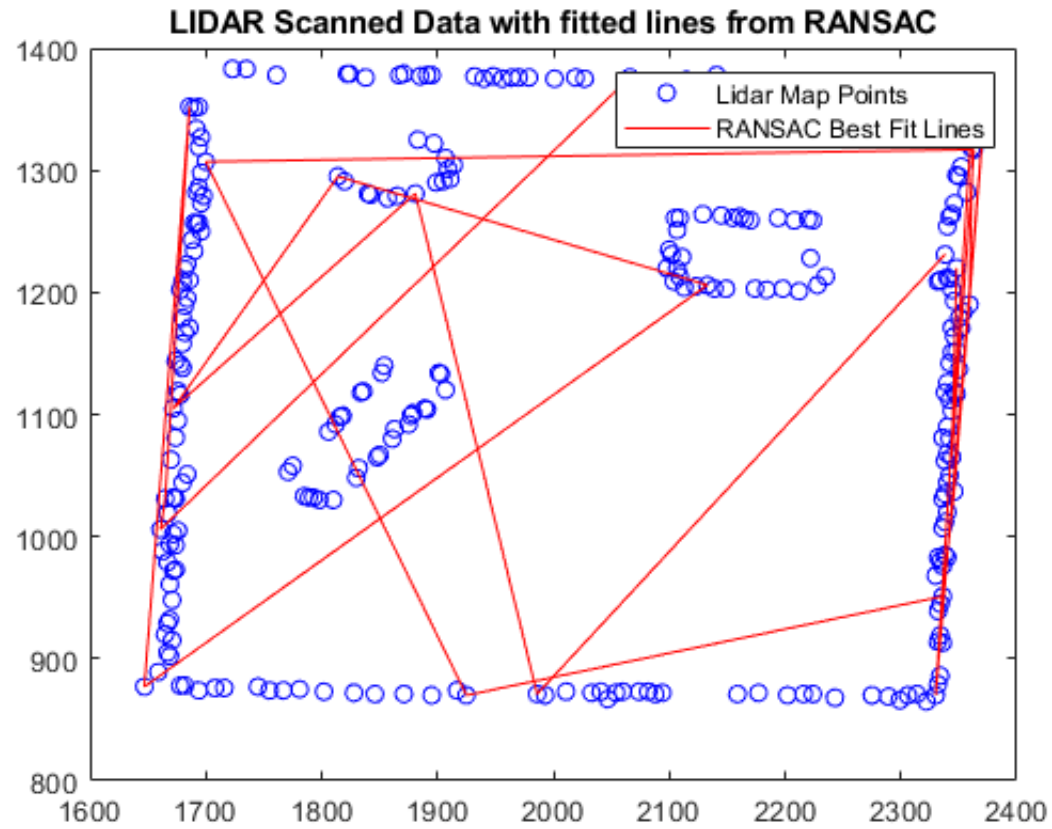
% Stores endpoints
endpoints = cat(1, endpoints, [minendpoint, maxendpoint]);

% Creates a new data matrix with points that are not inliers
for m = 1:length(data)
    r1 = data(m,:) - bestPoint1SoFar;
    That1 = (bestPoint2SoFar - bestPoint1SoFar) ./ vecnorm(bestPoint2SoFar - bestPoint1
SoFar);
    Nhat1 = ([-That1(2) That1(1)]);
    dist1 = dot(r1,Nhat1);
    if abs(dist1) > d
        newpoints = cat(1,newpoints, [data(m,1),data(m,2)]);
    end
end
% Delete inlier points
data = newpoints;
end

% Plot Map and RANSAC line
plot(startingdata(:,1),startingdata(:,2),'bo')
hold on
plot([endpoints(2:end,1),endpoints(2:end,3)], [endpoints(2:end,2),endpoints(2:end,4)], '-r');

```

```
hold off
title("LIDAR Scanned Data with fitted lines from RANSAC")
legend("Lidar Map Points", "RANSAC Best Fit Lines")
```



LIDAR Scanned Data with fitted lines from RANSAC

