

Contents

- [Hill](#)
- [Another Hill](#)
- [Circular Valley](#)
- [Valley](#)
- [Uniform Grid](#)
- [Random Grid](#)
- [Flat](#)
- [Flat With Lake](#)
- [Helper Function - Normalize](#)

```
% Artificial Point Cloud Test Sets And Results
% Kartik V. Sastry, Team NavX
% ECE 4012 - Senior Design 2

% Generates simple test cases for path planning. Data is normalized such
% that all dimensions are on the range 0 to 1 so that incline angles can be
% visualized.

% Each test case is visualized, and a text file of tuples [X, Y, Z, CLASS]
% is generated for input into the path planner.

% The Path Planner is run on each dataset for a success case and a failure
% case and results are visualized.
```

Hill

```
close all; clear; clc

% Form Dataset
myN = 40;
myX = linspace(-5, 5, myN);
myY = linspace(-5, 5, myN);
[myXX, myYY] = meshgrid(myX, myY);
myZZ = - myXX.^2 + 4.*myXX.^3 - myYY.^4;
myXVec = reshape(myXX, myN*myN, 1);
myYVec = reshape(myYY, myN*myN, 1);
myZVec = reshape(myZZ, myN*myN, 1);

% Surface Plot
figure('units','normalized','outerposition',[0 0 1 1])
subplot(2, 2, 1)
surf(myXX, myYY, myZZ);
colormap winter
xlabel('x'); ylabel('y'); zlabel('z'); title('Hill')

% Plot "Sampled Data"
myXVec = normalize(myXVec);
myYVec = normalize(myYVec);
myZVec = normalize(myZVec);
subplot(2, 2, 2)
scatter3(myXVec, myYVec, myZVec, 'bo', 'filled')
xlabel('x');
ylabel('y');
zlabel('z');
title('Hill Normalized, Sampled')

% All classified as ground
myClasses = 2.*ones(length(myXVec), 1);

% Write to a file
myLongRow = zeros(1, 4*length(myXVec));
myLongRow(1:4:end) = myXVec;
myLongRow(2:4:end) = myYVec;
myLongRow(3:4:end) = myZVec;
myLongRow(4:4:end) = myClasses;
csvwrite('hill_raw.csv', myLongRow);

% Visualize Edge Connectivity
myFilename = 'pseudoTestData/hill_raw-graph.csv';
myEdges = csvread(myFilename);
subplot(2, 2, 3)
[r,c] = size(myEdges);
for ii = 1:r
    plot3(myEdges(ii,1:2), myEdges(ii,3:4), myEdges(ii,5:6), 'b-')
    hold on
end

% Plot Start and End Points
myEndpoints = [ 0.3077 0.359 0.6855;
                0.6923 0.641 0.7205];
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
    'MarkerEdgeColor',[0.3 0.7 1],'MarkerFaceColor',[0.3 0.7 1],'LineWidth',1.5)
```

```

% Overlay Path Generated by A*
myFilename2 = 'pseudoTestData/hill_raw-path.csv';
myPath = csvread(myFilename2);
[r,c] = size(myPath);
plot3(myPath(:, 1), myPath(:,2), myPath(:,3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')

% Overlay Path on the Surface Plot
subplot(2, 2, 4)
surf(myXX, myYY, myZZ);
colormap winter
hold on;
xlabel('x'); ylabel('y'); zlabel('z'); title('Hill')

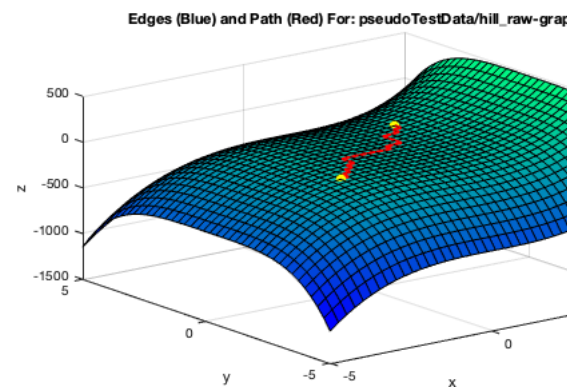
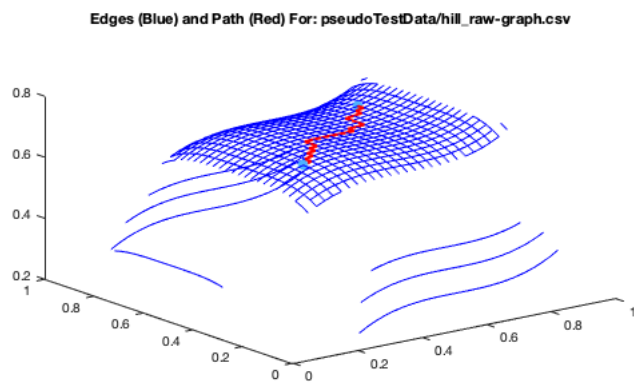
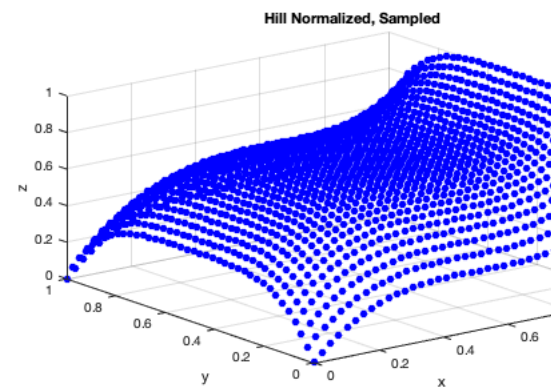
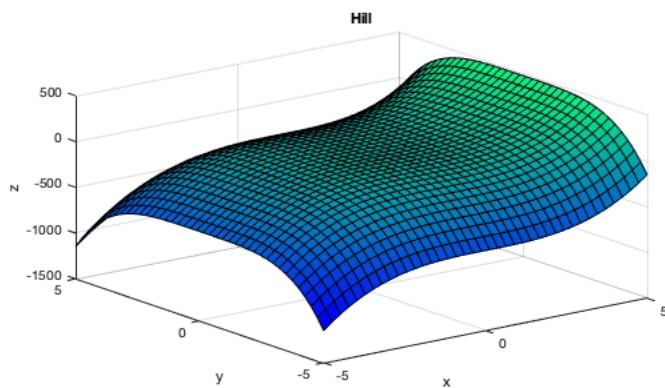
% Endpoints and Path Waypoints Need to be Un-Normalized!
myXVec = reshape(myXX, myN*myN, 1);
myYVec = reshape(myYY, myN*myN, 1);
myZVec = reshape(myZZ, myN*myN, 1);

myEndpoints(:, 1) = (myEndpoints(:, 1) * range(myXVec)) + min(myXVec);
myEndpoints(:, 2) = (myEndpoints(:, 2) * range(myYVec)) + min(myYVec);
myEndpoints(:, 3) = (myEndpoints(:, 3) * range(myZVec)) + min(myZVec);

myPath(:, 1) = (myPath(:, 1) * range(myXVec)) + min(myXVec);
myPath(:, 2) = (myPath(:, 2) * range(myYVec)) + min(myYVec);
myPath(:, 3) = (myPath(:, 3) * range(myZVec)) + min(myZVec);

% Plot Surface and Path
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
    'MarkerEdgeColor',[1 1 0], 'MarkerFaceColor',[1 1 0], 'LineWidth',1.5)
plot3(myPath(:, 1), myPath(:, 2), myPath(:, 3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')

```



Another Hill

Can pick endpoints on opposite side of the hill

```

close all; clear; clc

% Form Dataset
myN = 40;
myX = linspace(-5, 5, myN);
myY = linspace(-5, 5, myN);

```

```

[myXX, myYY] = meshgrid(myX, myY);
mySigmaX = 1.5;
mySigmaY = 1.5;
myMeanX = 0;
myMeanY = 0;
% Bivariate Gaussian
myZZ = exp( -( ( myXX-myMeanX).^2 ./ (2*mySigmaX.^2) ) + ...
              ( myYY-myMeanY).^2 ./ (2*mySigmaY.^2) ) );
myXVec = reshape(myXX, myN*myN, 1);
myYVec = reshape(myYY, myN*myN, 1);
myZVec = reshape(myZZ, myN*myN, 1);

% Surface Plot
figure('units','normalized','outerposition',[0 0 1 1])
subplot(2, 2, 1)
surf(myXX, myYY, myZZ);
colormap winter
xlabel('x'); ylabel('y'); zlabel('z'); title('Another Hill')

% Plot "Sampled Data"
myXVec = normalize(myXVec);
myYVec = normalize(myYVec);
myZVec = normalize(myZVec);
subplot(2, 2, 2)
scatter3(myXVec, myYVec, myZVec, 'bo', 'filled')
xlabel('x'); ylabel('y'); zlabel('z'); title('Another Hill Normalized, Sampled')

% All classified as ground
myClasses = 2.*ones(length(myXVec), 1);

% Write to a file
myLongRow = zeros(1, 4*length(myXVec));
myLongRow(1:4:end) = myXVec;
myLongRow(2:4:end) = myYVec;
myLongRow(3:4:end) = myZVec;
myLongRow(4:4:end) = myClasses;
csvwrite('anotherHill_raw.csv', myLongRow);

% Visualize Edge Connectivity
myFilename = 'pseudoTestData/anotherHill_raw-graph.csv';
myEdges = csvread(myFilename);
subplot(2, 2, 3)
[r,c] = size(myEdges);
for ii = 1:r
    plot3(myEdges(ii,1:2), myEdges(ii,3:4), myEdges(ii,5:6), 'b-')
    hold on
end

% Plot Start and End Points
myEndpoints = [ 0.3077 0.1026 0.01322;
                0.8205 0.8718 0.004746];
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
        'MarkerEdgeColor',[0.3 0.7 1],'MarkerFaceColor',[0.3 0.7 1],'LineWidth',1.5)

% Overlay Path Generated by A*
myFilename2 = 'pseudoTestData/anotherHill_raw-path.csv';
myPath = csvread(myFilename2);
[r,c] = size(myPath);
plot3(myPath(:, 1), myPath(:,2), myPath(:,3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')

% Overlay Path on the Surface Plot
subplot(2, 2, 4)
surf(myXX, myYY, myZZ);
colormap winter
hold on;
xlabel('x'); ylabel('y'); zlabel('z'); title('Hill')

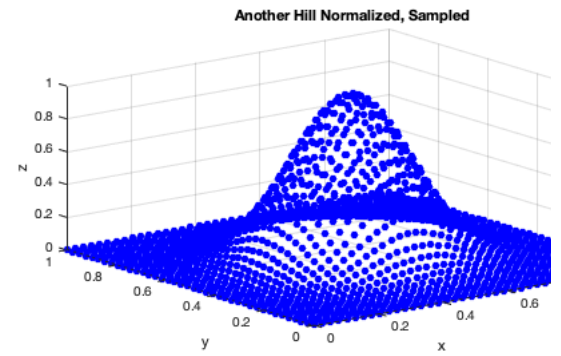
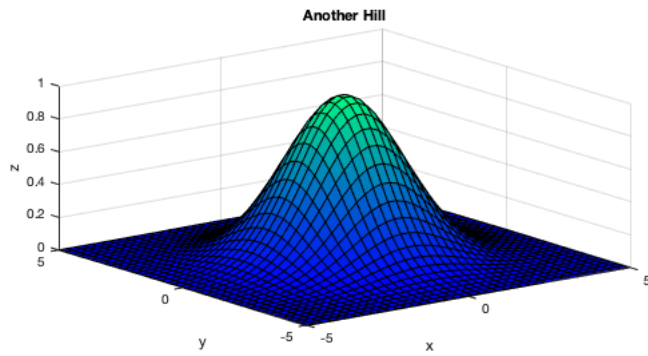
% Endpoints and Path Waypoints Need to be Un-Normalized!
myXVec = reshape(myXX, myN*myN, 1);
myYVec = reshape(myYY, myN*myN, 1);
myZVec = reshape(myZZ, myN*myN, 1);

myEndpoints(:, 1) = (myEndpoints(:, 1) * range(myXVec)) + min(myXVec);
myEndpoints(:, 2) = (myEndpoints(:, 2) * range(myYVec)) + min(myYVec);
myEndpoints(:, 3) = (myEndpoints(:, 3) * range(myZVec)) + min(myZVec);

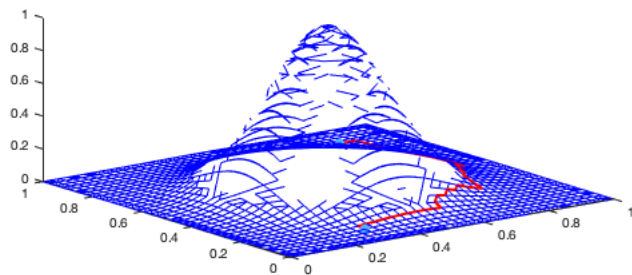
myPath(:, 1) = (myPath(:, 1) * range(myXVec)) + min(myXVec);
myPath(:, 2) = (myPath(:, 2) * range(myYVec)) + min(myYVec);
myPath(:, 3) = (myPath(:, 3) * range(myZVec)) + min(myZVec);

% Plot Surface and Path
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
        'MarkerEdgeColor',[1 1 0],'MarkerFaceColor',[1 1 0],'LineWidth',1.5)
plot3(myPath(:, 1), myPath(:, 2), myPath(:, 3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')

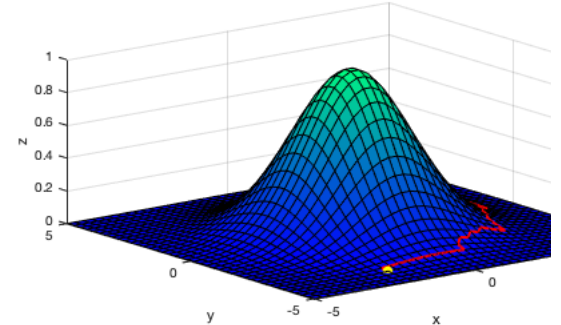
```



Edges (Blue) and Path (Red) For: pseudoTestData/anotherHill_raw-graph.csv



Edges (Blue) and Path (Red) For: pseudoTestData/anotherHill_raw-



Circular Valley

Can pick endpoints on opposite sides of a circular valley.

```
close all; clear; clc

% Form Dataset
myN = 40;
myX = linspace(-5, 5, myN);
myY = linspace(-5, 5, myN);
[myXX, myYY] = meshgrid(myX, myY);
myZZ = sin(3.*sqrt(myXX.^2+myYY.^2));
myXVec = reshape(myXX, myN*myN, 1);
myYVec = reshape(myYY, myN*myN, 1);
myZVec = reshape(myZZ, myN*myN, 1);

% Surface Plot
figure('units','normalized','outerposition',[0 0 1 1])
subplot(2, 2, 1)
surf(myXX, myYY, myZZ);
colormap winter;
xlabel('x'); ylabel('y'); zlabel('z'); title('Circular Valley')

% Plot "Sampled Data"
myXVec = normalize(myXVec);
myYVec = normalize(myYVec);
myZVec = normalize(myZVec);
subplot(2, 2, 2)
scatter3(myXVec, myYVec, myZVec, 'bo', 'filled')
xlabel('x'); ylabel('y'); zlabel('z'); title('Circular Valley Normalized, Sampled')

% All classified as ground
myClasses = 2.*ones(length(myXVec), 1);

% Write to a file
myLongRow = zeros(1, 4*length(myXVec));
myLongRow(1:4:end) = myXVec;
myLongRow(2:4:end) = myYVec;
myLongRow(3:4:end) = myZVec;
myLongRow(4:4:end) = myClasses;
csvwrite('circularValley_raw.csv', myLongRow);

% Visualize Edge Connectivity
myFilename = 'pseudoTestData/circularValley_raw-graph.csv';
myEdges = csvread(myFilename);
subplot(2, 2, 3)
[r,c] = size(myEdges);
for ii = 1:r
    plot3(myEdges(ii,1:2), myEdges(ii,3:4), myEdges(ii,5:6), 'b-')
```

```

hold on
end

% Plot Start and End Points
myEndpoints = [ 0.1538 0.5641 0.04644;
                0.8718 0.5641 0.02583];
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
'MarkerEdgeColor',[0.3 0.7 1],'MarkerFaceColor',[0.3 0.7 1],'LineWidth',1.5)

% Overlay Path Generated by A*
myFilename2 = 'pseudoTestData/circularValley_raw-path.csv';
myPath = csvread(myFilename2);
[r,c] = size(myPath);
plot3(myPath(:, 1), myPath(:,2), myPath(:,3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')

% Overlay Path on the Surface Plot
subplot(2, 2, 4)
surf(myXX, myYY, myZZ);
colormap winter
hold on;
xlabel('x'); ylabel('y'); zlabel('z'); title('Hill')

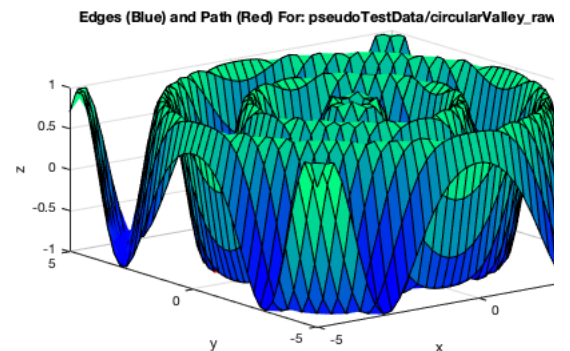
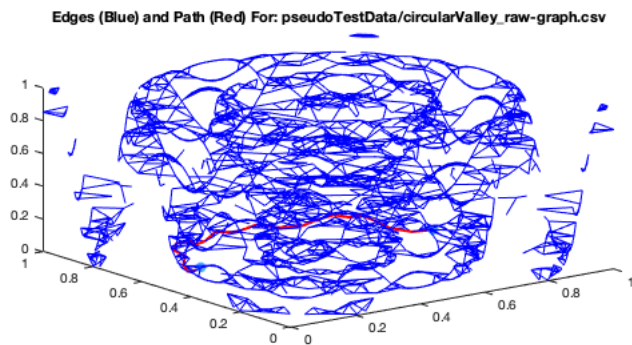
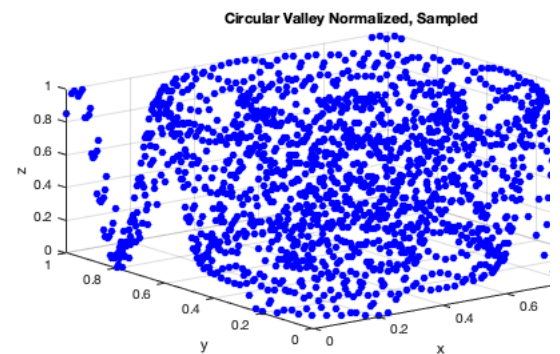
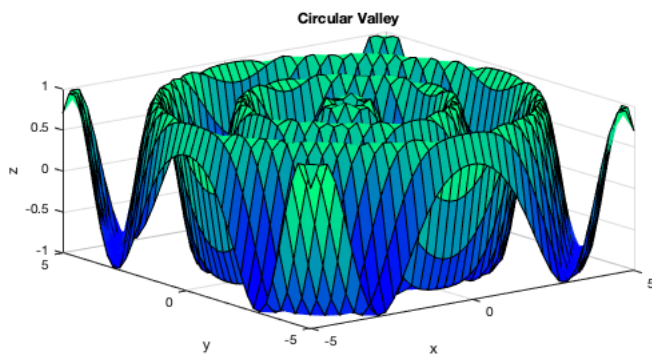
% Endpoints and Path Waypoints Need to be Un-Normalized!
myXVec = reshape(myXX, myN*myN, 1);
myYVec = reshape(myYY, myN*myN, 1);
myZVec = reshape(myZZ, myN*myN, 1);

myEndpoints(:, 1) = (myEndpoints(:, 1) * range(myXVec)) + min(myXVec);
myEndpoints(:, 2) = (myEndpoints(:, 2) * range(myYVec)) + min(myYVec);
myEndpoints(:, 3) = (myEndpoints(:, 3) * range(myZVec)) + min(myZVec);

myPath(:, 1) = (myPath(:, 1) * range(myXVec)) + min(myXVec);
myPath(:, 2) = (myPath(:, 2) * range(myYVec)) + min(myYVec);
myPath(:, 3) = (myPath(:, 3) * range(myZVec)) + min(myZVec);

% Plot Surface and Path
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
'MarkerEdgeColor',[1 1 0],'MarkerFaceColor',[1 1 0],'LineWidth',1.5)
plot3(myPath(:, 1), myPath(:, 2), myPath(:, 3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')

```



Valley

Can pick endpoints in the flat region - verify that the robot doesn't climb the hill.

```
close all; clear; clc
```

```

% Form Dataset
myN = 40;
myX = linspace(-5, 5, myN);
myY = linspace(-5, 5, myN);
[myXX, myYY] = meshgrid(myX, myY);
myZZ = abs(myXX-2 + myYY-3) + abs(myXX+2 + myYY+3);
myXVec = reshape(myXX, myN*myN, 1);
myYVec = reshape(myYY, myN*myN, 1);
myZVec = reshape(myZZ, myN*myN, 1);

% Surface Plot
figure('units','normalized','outerposition',[0 0 1 1])
subplot(2, 2, 1)
surf(myXX, myYY, myZZ);
colormap winter
xlabel('x'); ylabel('y'); zlabel('z'); title('Valley')

% Plot "Sampled Data"
myXVec = normalize(myXVec);
myYVec = normalize(myYVec);
myZVec = normalize(myZVec);
subplot(2, 2, 2)
scatter3(myXVec, myYVec, myZVec, 'bo', 'filled')
xlabel('x');
ylabel('y');
zlabel('z');
title('Valley Normalized, Sampled')

% All classified as ground
myClasses = 2*ones(length(myXVec), 1);

% Write to a file
myLongRow = zeros(1, 4*length(myXVec));
myLongRow(1:4:end) = myXVec;
myLongRow(2:4:end) = myYVec;
myLongRow(3:4:end) = myZVec;
myLongRow(4:4:end) = myClasses;
csvwrite('valley_raw.csv', myLongRow);

% Visualize Edge Connectivity
myFilename = 'pseudoTestData/valley_raw-graph.csv';
myEdges = csvread(myFilename);
subplot(2, 2, 3)
[r,c] = size(myEdges);
for ii = 1:r
    plot3(myEdges(ii,1:2), myEdges(ii,3:4), myEdges(ii,5:6), 'b-')
    hold on
end

% Plot Start and End Points
myEndpoints = [ 0.1282 0.8205 0;
                0.8718 0.4872 0];
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
    'MarkerEdgeColor',[0.3 0.7 1], 'MarkerFaceColor',[0.3 0.7 1], 'LineWidth',1.5)

% Overlay Path Generated by A*
myFilename2 = 'pseudoTestData/valley_raw-path.csv';
myPath = csvread(myFilename2);
[r,c] = size(myPath);
plot3(myPath(:, 1), myPath(:,2), myPath(:,3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')

% Overlay Path on the Surface Plot
subplot(2, 2, 4)
surf(myXX, myYY, myZZ);
colormap winter
hold on;
xlabel('x'); ylabel('y'); zlabel('z'); title('Hill')

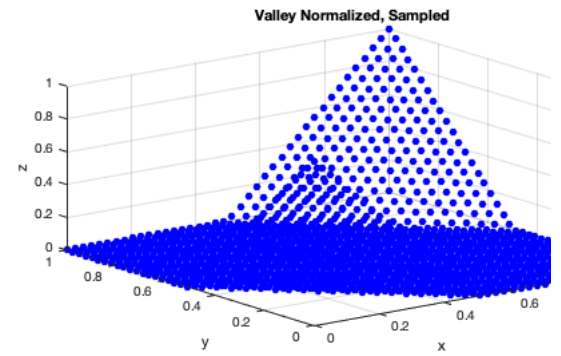
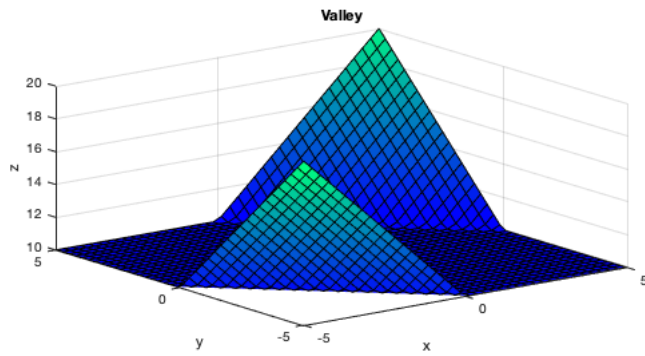
% Endpoints and Path Waypoints Need to be Un-Normalized!
myXVec = reshape(myXX, myN*myN, 1);
myYVec = reshape(myYY, myN*myN, 1);
myZVec = reshape(myZZ, myN*myN, 1);

myEndpoints(:, 1) = (myEndpoints(:, 1) * range(myXVec)) + min(myXVec);
myEndpoints(:, 2) = (myEndpoints(:, 2) * range(myYVec)) + min(myYVec);
myEndpoints(:, 3) = (myEndpoints(:, 3) * range(myZVec)) + min(myZVec);

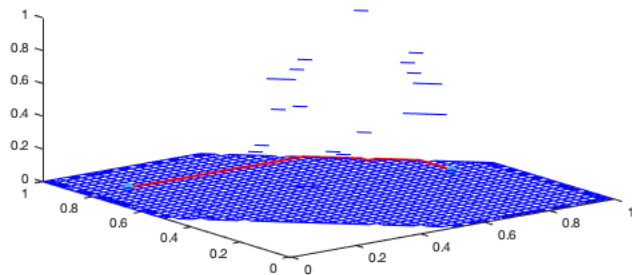
myPath(:, 1) = (myPath(:, 1) * range(myXVec)) + min(myXVec);
myPath(:, 2) = (myPath(:, 2) * range(myYVec)) + min(myYVec);
myPath(:, 3) = (myPath(:, 3) * range(myZVec)) + min(myZVec);

% Plot Surface and Path
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
    'MarkerEdgeColor',[1 1 0], 'MarkerFaceColor',[1 1 0], 'LineWidth',1.5)
plot3(myPath(:, 1), myPath(:, 2), myPath(:, 3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')

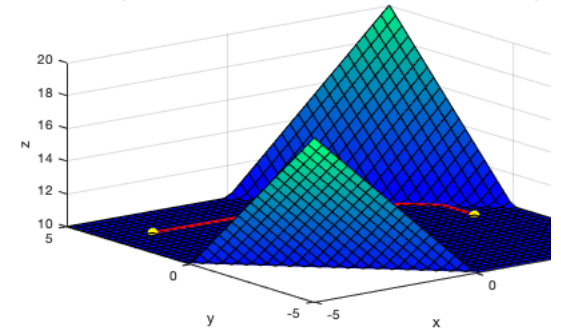
```



Edges (Blue) and Path (Red) For: pseudoTestData/valley_raw-graph.csv



Edges (Blue) and Path (Red) For: pseudoTestData/valley_raw-gr



Uniform Grid

Can pick endpoints that are diagonally opposite. Snake through obstacles

```
close all; clear; clc

% Form Dataset
myN = 40;
myX = linspace(-5, 5, myN);
myY = linspace(-5, 5, myN);
[myXX, myYY] = meshgrid(myX, myY);
myMask = zeros(size(myXX));
UniformDim = 4;
for i = 1+UniformDim:2*UniformDim:myN-UniformDim
    for j = 1+UniformDim:2*UniformDim:myN-UniformDim
        myMask(i:i+UniformDim, j:j+UniformDim) = 1;
    end
end
myZZ = ones(size(myXX)) + 5.*ones(size(myXX)).*myMask;
myXVec = reshape(myXX, myN*myN, 1);
myYVec = reshape(myYY, myN*myN, 1);
myZVec = reshape(myZZ, myN*myN, 1);

% Surface Plot
figure('units','normalized','outerposition',[0 0 1 1])
subplot(2, 2, 1)
surf(myXX, myYY, myZZ);
colormap winter
xlabel('x'); ylabel('y'); zlabel('z'); title('Uniform Grid')
```

```
% Plot "Sampled Data"
myXVec = normalize(myXVec);
myYVec = normalize(myYVec);
myZVec = normalize(myZVec);
subplot(2, 2, 2)
scatter3(myXVec, myYVec, myZVec, 'bo', 'filled')
xlabel('x'); ylabel('y'); zlabel('z'); title('Uniform Grid Normalized, Sampled')
```

```
% All classified as ground
myClasses = 2.*ones(length(myXVec), 1);

% Write to a file
myLongRow = zeros(1, 4*length(myXVec));
myLongRow(1:4:end) = myXVec;
myLongRow(2:4:end) = myYVec;
myLongRow(3:4:end) = myZVec;
myLongRow(4:4:end) = myClasses;
csvwrite('uniformGrid_raw.csv', myLongRow);
```

```

% Visualize Edge Connectivity
myFilename = 'pseudoTestData/uniformGrid_raw-graph.csv';
myEdges = csvread(myFilename);
subplot(2, 2, 3)
[r,c] = size(myEdges);
for ii = 1:r
    plot3(myEdges(ii,1:2), myEdges(ii,3:4), myEdges(ii,5:6), 'b-')
    hold on
end

% Plot Start and End Points
myEndpoints = [ 0.2564 0.1795 0; 0.6667 0.7436 0];
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
    'MarkerEdgeColor',[0.3 0.7 1],'MarkerFaceColor',[0.3 0.7 1],'LineWidth',1.5)

% Overlay Path Generated by A*
myFilename2 = 'pseudoTestData/uniformGrid_raw-path.csv';
myPath = csvread(myFilename2);
[r,c] = size(myPath);
plot3(myPath(:, 1), myPath(:,2), myPath(:,3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')

% Overlay Path on the Surface Plot
subplot(2, 2, 4)
surf(myXX, myYY, myZZ);
colormap winter
hold on;
xlabel('x'); ylabel('y'); zlabel('z'); title('Hill')

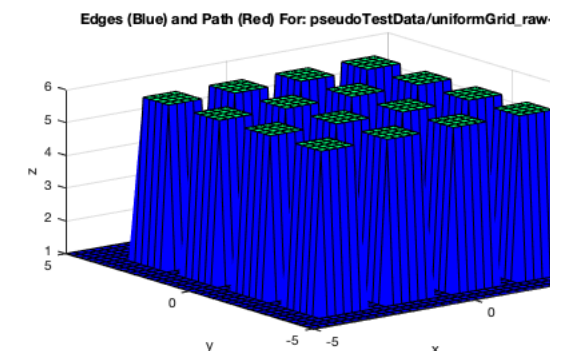
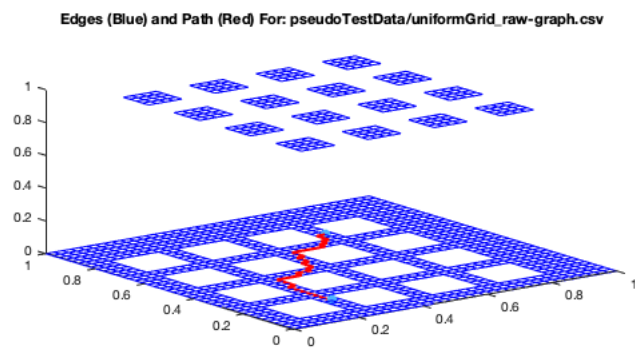
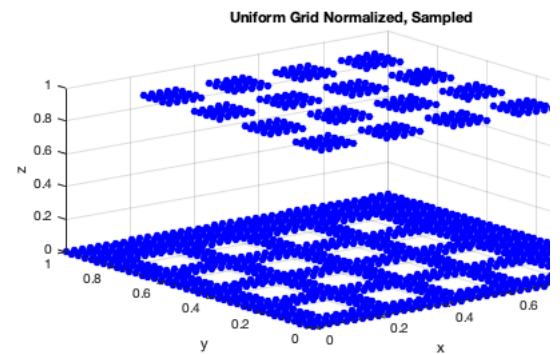
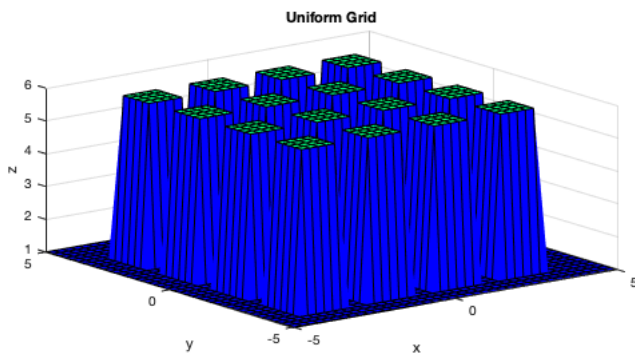
% Endpoints and Path Waypoints Need to be Un-Normalized!
myXVec = reshape(myXX, myN*myN, 1);
myYVec = reshape(myYY, myN*myN, 1);
myZVec = reshape(myZZ, myN*myN, 1);

myEndpoints(:, 1) = (myEndpoints(:, 1) * range(myXVec)) + min(myXVec);
myEndpoints(:, 2) = (myEndpoints(:, 2) * range(myYVec)) + min(myYVec);
myEndpoints(:, 3) = (myEndpoints(:, 3) * range(myZVec)) + min(myZVec);

myPath(:, 1) = (myPath(:, 1) * range(myXVec)) + min(myXVec);
myPath(:, 2) = (myPath(:, 2) * range(myYVec)) + min(myYVec);
myPath(:, 3) = (myPath(:, 3) * range(myZVec)) + min(myZVec);

% Plot Surface and Path
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
    'MarkerEdgeColor',[1 1 0],'MarkerFaceColor',[1 1 0],'LineWidth',1.5)
plot3(myPath(:, 1), myPath(:, 2), myPath(:, 3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')

```



Random Grid

Can pick endpoints that are diagonally opposite. Snake through obstacles

```

close all; clear; clc

% Form Dataset
myN = 40;
myX = linspace(-5, 5, myN);
myY = linspace(-5, 5, myN);
[myXX, myYY] = meshgrid(myX, myY);
myMask = zeros(size(myXX));
rng(22) % Seed for repeatability
UniformDim = 2;
for i = round((myN-UniformDim-(1+UniformDim))*rand(1, 4) + 1+UniformDim)
    for j = round((myN-UniformDim-(1+UniformDim))*rand(1, 6) + 1+UniformDim)
        myMask(i:i+UniformDim, j:j+UniformDim) = 1;
    end
end
myZZ = ones(size(myXX)) + 5.*ones(size(myXX)).*myMask;
myXVec = reshape(myXX, myN*myN, 1);
myYVec = reshape(myYY, myN*myN, 1);
myZVec = reshape(myZZ, myN*myN, 1);

% Surface Plot
figure('units','normalized','outerposition',[0 0 1 1])
subplot(2, 2, 1)
surf(myXX, myYY, myZZ);
colormap winter
xlabel('x'); ylabel('y'); zlabel('z'); title('Random Grid')

% Plot "Sampled Data"
myXVec = normalize(myXVec);
myYVec = normalize(myYVec);
myZVec = normalize(myZVec);
subplot(2, 2, 2)
scatter3(myXVec, myYVec, myZVec, 'bo', 'filled')
xlabel('x'); ylabel('y'); zlabel('z'); title('Random Grid Normalized, Sampled')

% All classified as ground
myClasses = 2.*ones(length(myXVec), 1);

% Write to a file
myLongRow = zeros(1, 4*length(myXVec));
myLongRow(1:4:end) = myXVec;
myLongRow(2:4:end) = myYVec;
myLongRow(3:4:end) = myZVec;
myLongRow(4:4:end) = myClasses;
csvwrite('randomGrid_raw.csv', myLongRow);

% Visualize Edge Connectivity
myFilename = 'pseudoTestData/randomGrid_raw-graph.csv';
myEdges = csvread(myFilename);
subplot(2, 2, 3)
[r,c] = size(myEdges);
for ii = 1:r
    plot3(myEdges(ii,1:2), myEdges(ii,3:4), myEdges(ii,5:6), 'b-')
    hold on
end

% Plot Start and End Points
myEndpoints = [ 0.3333 0.8718 0 0.9231 0.3077 0];
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
    'MarkerEdgeColor',[0.3 0.7 1],'MarkerFaceColor',[0.3 0.7 1],'LineWidth',1.5)

% Overlay Path Generated by A*
myFilename2 = 'pseudoTestData/randomGrid_raw-path.csv';
myPath = csvread(myFilename2);
[r,c] = size(myPath);
plot3(myPath(:, 1), myPath(:,2), myPath(:,3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')
% Overlay Path on the Surface Plot
subplot(2, 2, 4)
surf(myXX, myYY, myZZ);
colormap winter
hold on;
xlabel('x'); ylabel('y'); zlabel('z'); title('Hill')

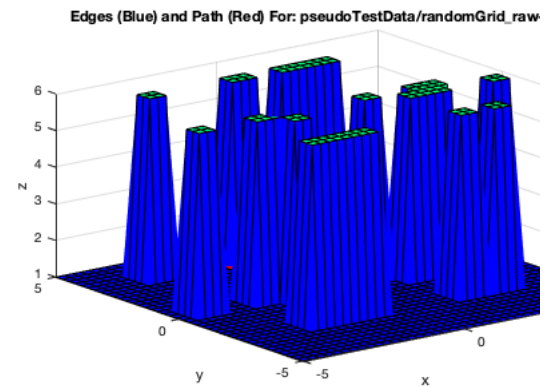
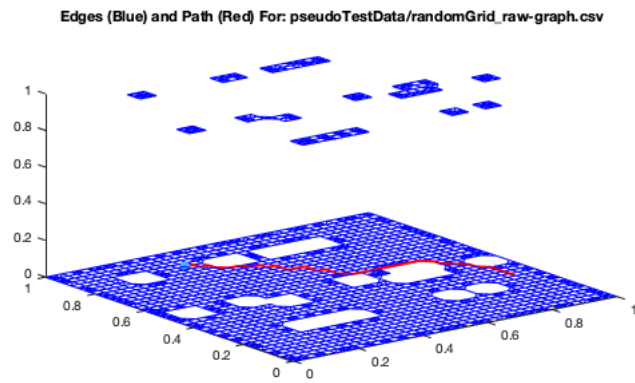
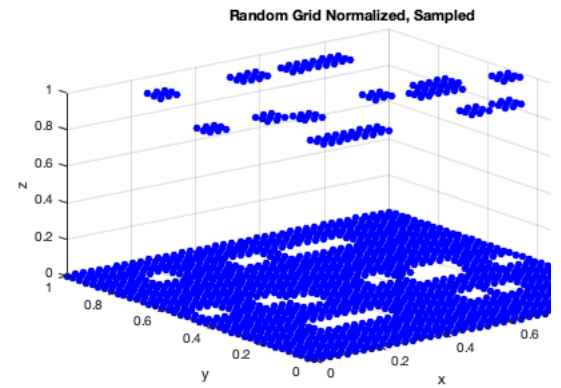
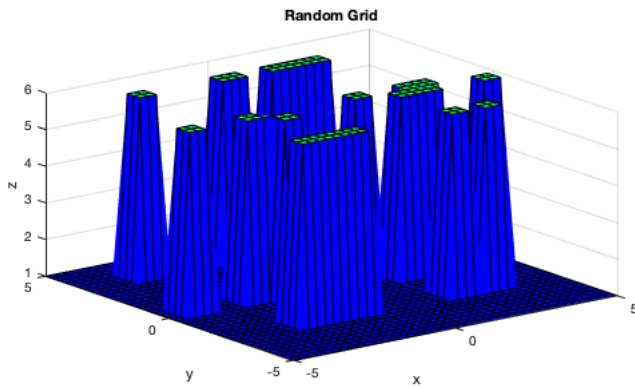
% Endpoints and Path Waypoints Need to be Un-Normalized!
myXVec = reshape(myXX, myN*myN, 1);
myYVec = reshape(myYY, myN*myN, 1);
myZVec = reshape(myZZ, myN*myN, 1);

myEndpoints(:, 1) = (myEndpoints(:, 1) * range(myXVec)) + min(myXVec);
myEndpoints(:, 2) = (myEndpoints(:, 2) * range(myYVec)) + min(myYVec);
myEndpoints(:, 3) = (myEndpoints(:, 3) * range(myZVec)) + min(myZVec);

myPath(:, 1) = (myPath(:, 1) * range(myXVec)) + min(myXVec);
myPath(:, 2) = (myPath(:, 2) * range(myYVec)) + min(myYVec);
myPath(:, 3) = (myPath(:, 3) * range(myZVec)) + min(myZVec);

```

```
% Plot Surface and Path
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
'MarkerEdgeColor',[1 1 0],'MarkerFaceColor',[1 1 0],'LineWidth',1.5)
plot3(myPath(:, 1), myPath(:, 2), myPath(:, 3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')
```



Flat

```
close all; clear; clc

% Form Dataset
myN = 40;
myX = linspace(-4, 4, myN);
myY = linspace(-4, 4, myN);
[myXX, myYY] = meshgrid(myX, myY);
myZZ = ones(size(myXX));
myXVec = reshape(myXX, myN*myN, 1);
myYVec = reshape(myYY, myN*myN, 1);
myZVec = reshape(myZZ, myN*myN, 1);

% Surface Plot
figure('units','normalized','outerposition',[0 0 1 1])
subplot(2, 2, 1)
surf(myXX, myYY, myZZ);
colormap winter
xlabel('x'); ylabel('y'); zlabel('z'); title('Flat (No Need to Normalize)')

% Plot "Sampled Data"
subplot(2, 2, 2)
scatter3(myXVec, myYVec, myZVec, 'bo', 'filled')
xlabel('x'); ylabel('y'); zlabel('z'); title('Flat (No Need to Normalize)')

% All classified as ground
myClasses = 2.*ones(length(myXVec), 1);

% Write to a file
myLongRow = zeros(1, 4*length(myXVec));
myLongRow(1:4:end) = myXVec;
myLongRow(2:4:end) = myYVec;
myLongRow(3:4:end) = myZVec;
myLongRow(4:4:end) = myClasses;
csvwrite('flat_raw.csv', myLongRow);
```

```

myFilename = 'pseudoTestData/flat_raw-graph.csv';
myEdges = csvread(myFilename);

% Visualize Edge Connectivity
subplot(2, 2, 3)
[r,c] = size(myEdges);
for ii = 1:r
    plot3(myEdges(ii,1:2), myEdges(ii,3:4), myEdges(ii,5:6), 'b-')
    hold on
end

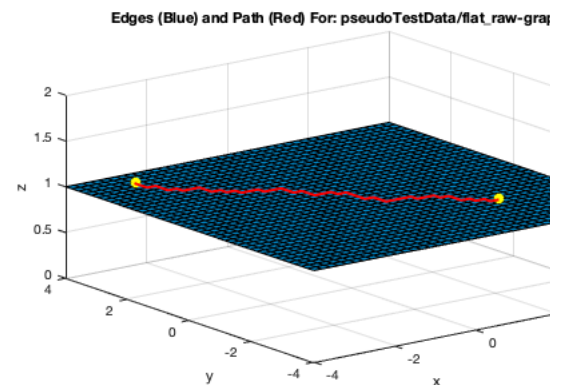
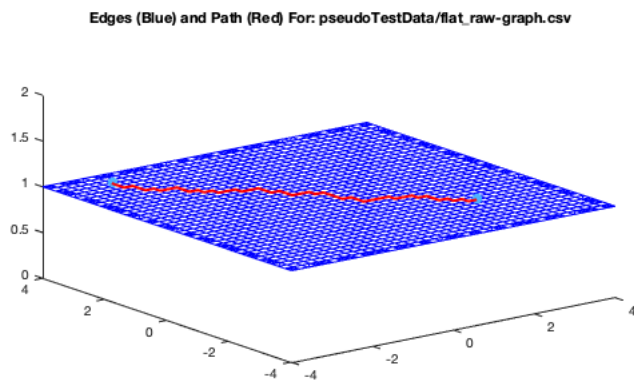
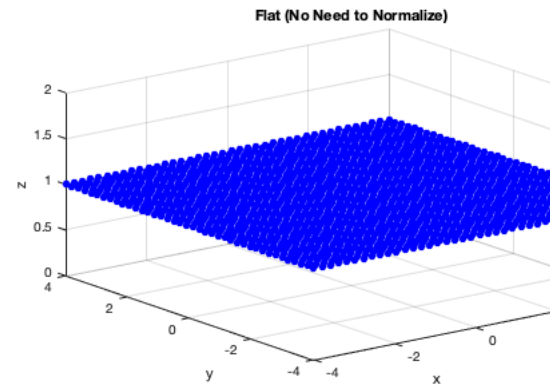
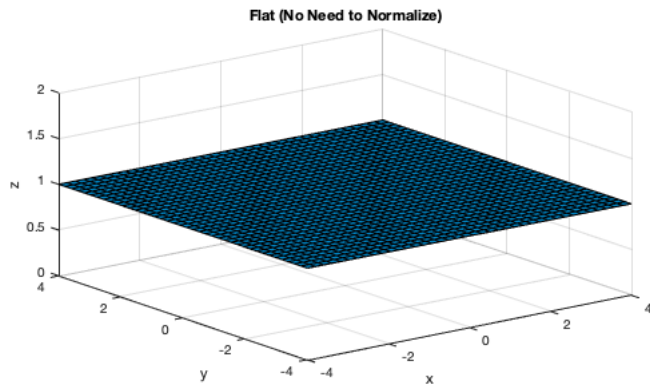
% Plot Start and End Points
myEndpoints = [ -2.692 3.462 1; 2.179 -1.923 1];
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
    'MarkerEdgeColor',[0.3 0.7 1], 'MarkerFaceColor',[0.3 0.7 1], 'LineWidth',1.5)

% Overlay Path Generated by A*
myFilename2 = 'pseudoTestData/flat_raw-path.csv';
myPath = csvread(myFilename2);
[r,c] = size(myPath);
plot3(myPath(:, 1), myPath(:, 2), myPath(:, 3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')

% Overlay Path on the Surface Plot
subplot(2, 2, 4)
surf(myXX, myYY, myZZ);
colormap winter
hold on;
xlabel('x'); ylabel('y'); zlabel('z'); title('Hill')

% Plot Surface and Path
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
    'MarkerEdgeColor',[1 1 0], 'MarkerFaceColor',[1 1 0], 'LineWidth',1.5)
plot3(myPath(:, 1), myPath(:, 2), myPath(:, 3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')

```



Flat With Lake

Can pick any two endpoints

```

close all; clear; clc

% Form Dataset

```

```

myN = 40;
myX = linspace(-5, 5, myN);
myY = linspace(-5, 5, myN);
[myXX, myYY] = meshgrid(myX, myY);
myZZ = ones(size(myXX));
myXVec = reshape(myXX, myN*myN, 1);
myYVec = reshape(myYY, myN*myN, 1);
myZVec = reshape(myZZ, myN*myN, 1);

% Make a 'Pond' In the Middle
% Ground is 2, Water is 9.
myClasses = 7.*(myXX.^2 + myYY.^2) < 4);
myClasses = myClasses + 2;
myClassesVec = reshape(myClasses, myN*myN, 1);

myColor = zeros(myN, myN, 3);
myColor(:, :, 2) = (111/255).*(myClasses == 2);
myColor(:, :, 3) = (myClasses == 9) + (49/255).*(myClasses == 2);

% Surface Plot
figure('units','normalized','outerposition',[0 0 1 1])
subplot(2, 2, 1)
surf(myXX, myYY, myZZ, myColor);
xlabel('x'); ylabel('y'); zlabel('z'); title('Flat (No Need to Normalize)')

% Plot "Sampled Data"
myColorVec = zeros(myN*myN, 3);
myColorVec(:, 2) = (111/255).*(myClassesVec == 2);
myColorVec(:, 3) = (myClassesVec == 9) + (49/255).*(myClassesVec == 2);
subplot(2, 2, 2)
scatter3(myXVec, myYVec, myZVec, 10, myColorVec, 'filled')
xlabel('x'); ylabel('y'); zlabel('z'); title('Flat (No Need to Normalize)')

% Write to a file
myLongRow = zeros(1, 4*length(myXVec));
myLongRow(1:4:end) = myXVec;
myLongRow(2:4:end) = myYVec;
myLongRow(3:4:end) = myZVec;
myLongRow(4:4:end) = myClassesVec;
csvwrite('flatWithLake_raw.csv', myLongRow);

% Visualize Edge Connectivity
myFilename = 'pseudoTestData/flatWithLake_raw-graph.csv';
myEdges = csvread(myFilename);
subplot(2, 2, 3)
[r,c] = size(myEdges);
for ii = 1:r
    plot3(myEdges(ii,1:2), myEdges(ii,3:4), myEdges(ii,5:6), 'b-')
    hold on
end

% Plot Start and End Points
myEndpoints = [ -2.692 3.462 1; 2.179 -1.923 1];
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
    'MarkerEdgeColor',[0.3 0.7 1],'MarkerFaceColor',[0.3 0.7 1],'LineWidth',1.5)

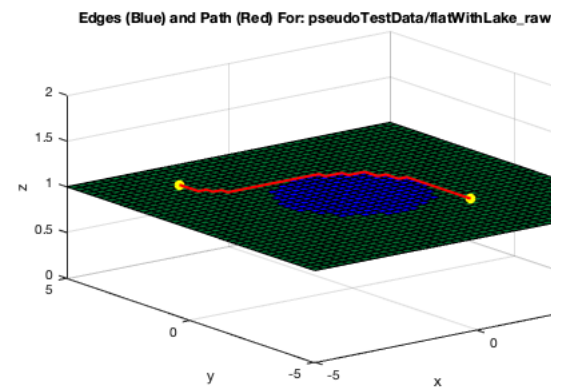
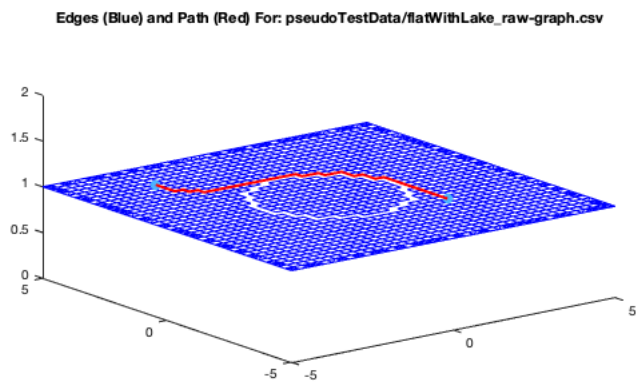
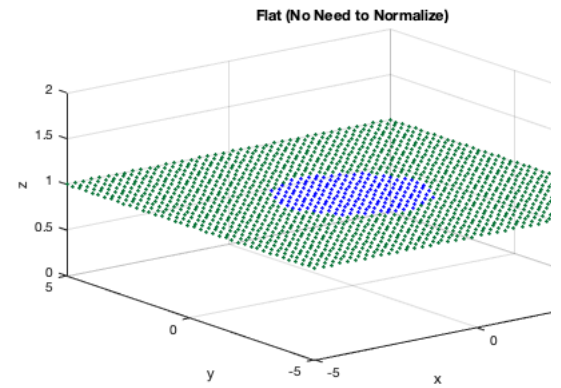
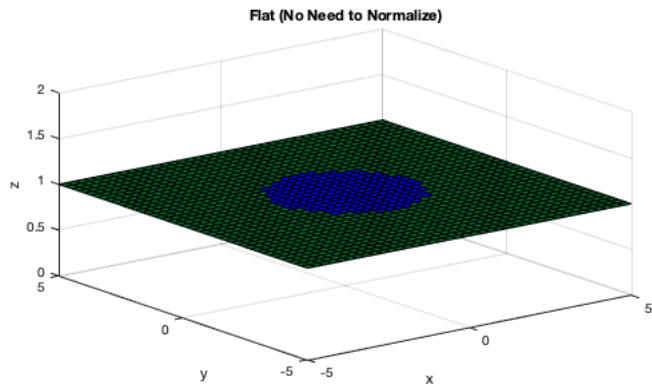
% Overlay Path Generated by A*
myFilename2 = 'pseudoTestData/flatWithLake_raw-path.csv';
myPath = csvread(myFilename2);

[r,c] = size(myPath);
plot3(myPath(:, 1), myPath(:,2), myPath(:,3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')

% Overlay Path on the Surface Plot
subplot(2, 2, 4)
surf(myXX, myYY, myZZ, myColor);
hold on;
xlabel('x'); ylabel('y'); zlabel('z'); title('Hill')

% Plot Surface and Path
scatter3(myEndpoints(:, 1), myEndpoints(:, 2), myEndpoints(:, 3), 40, ...
    'MarkerEdgeColor',[1 1 0],'MarkerFaceColor',[1 1 0],'LineWidth',1.5)
plot3(myPath(:, 1), myPath(:, 2), myPath(:, 3), 'r-', 'LineWidth', 2);
title(['Edges (Blue) and Path (Red) For: ', myFilename], 'Interpreter', 'none')

```



Helper Function - Normalize

```
function myNormedVec = normalize(myVec)
    myNormedVec = myVec - min(myVec);
    myNormedVec = myNormedVec ./ max(myNormedVec);
end
```

Published with MATLAB® R2018b