文件夹中有一个**add\_path.m**函数，每次尽量把**add\_path.m**这个函数和**CoordsTransfrom**，**icosahedron2sphere**，**Projection**，**BasicProcessing**，**visualization**，这五个文件夹放在同一目录下。

1. **imageprocess.m**函数：

此函数将一张全景图进行拆分，其中：

**输入**为一张全景图（读.mat文件），**输出**为多视角的拆分图（保存成.mat文件）。

1. 函数里面**fov = pi/2**;是需要改变的地方，若视角为90度，则fov = pi/2;

若视角为60度，则fov = pi/3;（在第27行）

1. 若视角为90度，则**image\_number= 1:12**；

若视角为60度，则**image\_number= 1:24**；（在第38行）

样例1（90度）：

输入：



输出：



样例1（60度）：

输入：



输出：



1. **Reconstruct\_box.m**函数：

此函数绘制全景图上八个点的坐标，并将八个点进行连线

函数**输入**：一张全景图和这张全景图对应的点；

函数**输出**：这八个点在全景图上的连线

此函数需要改动的地方为paintParameterLine函数里面的**panoEdgeC(drawId) = 0.8**;

最后的值不同画出来线的颜色也不一样。

样例：

输入：



输出：



1. **point2point.m**函数：

此函数根据全景图上的八个点来计算对应拆分图的点的坐标位置。

函数**输入**：全景图上的把各点的坐标

函数**输出**：对应拆分图的坐标位置

1. 函数里面**fov = pi/2**;是需要改变的地方，若视角为90度，则fov = pi/2;

若视角为60度，则fov = pi/3;（在第33行）

1. 若视角为90度，则**image\_number= 1:12**；

若视角为60度，则**image\_number= 1:24**；（在第44行）

1. **perspective2pano.m**函数：

此函数将拆分图拼接为全景图

函数**输入**：一系列拆分图

函数**输出**：一张全景图

1. 函数里面**fov = pi/2**;是需要改变的地方，若视角为90度，则fov = pi/2;

若视角为60度，则fov = pi/3;（在第30行）

1. 若视角为90度，则**image\_number= 12**；

若视角为60度，则**image\_number= 24**；（在第14行）

输出为2 x image\_number x 512 x1024 大小的数组

1. **pano\_algin.m**函数:

此函数将通过**perspective2pano.m**函数出来的图像进行拼接

函数**输入**：2 x image\_number x 512 x1024 大小的数组

函数**输出**：一张拼接好的全景图像

1. **GT\_Prediction.m**函数：

此函数把test数据集里面的原始点与经过网络的得到的点进行比较，并画在一张图上，最后输出到文件中。其中红色点代表预测得到的点，绿色点代表GT

函数**输入**：一张全景图，原始点坐标，预测出来的点的坐标

函数**输出**：分别画着预测出来以及原始点坐标位置的一张全景图

样例：

输入：



输出：



1. **GT\_Prediction\_lines.m**函数：

此函数把test数据集里面的原始点与经过网络的得到的点进行比较，做出线段，并画在一张图上，最后输出到文件中其中红色点代表预测得到的点，绿色点代表GT

函数**输入**：一张全景图，原始点坐标，预测出来的点的坐标

函数**输出**：分别画着预测出来以及原始点坐标位置以及连线的一张全景图

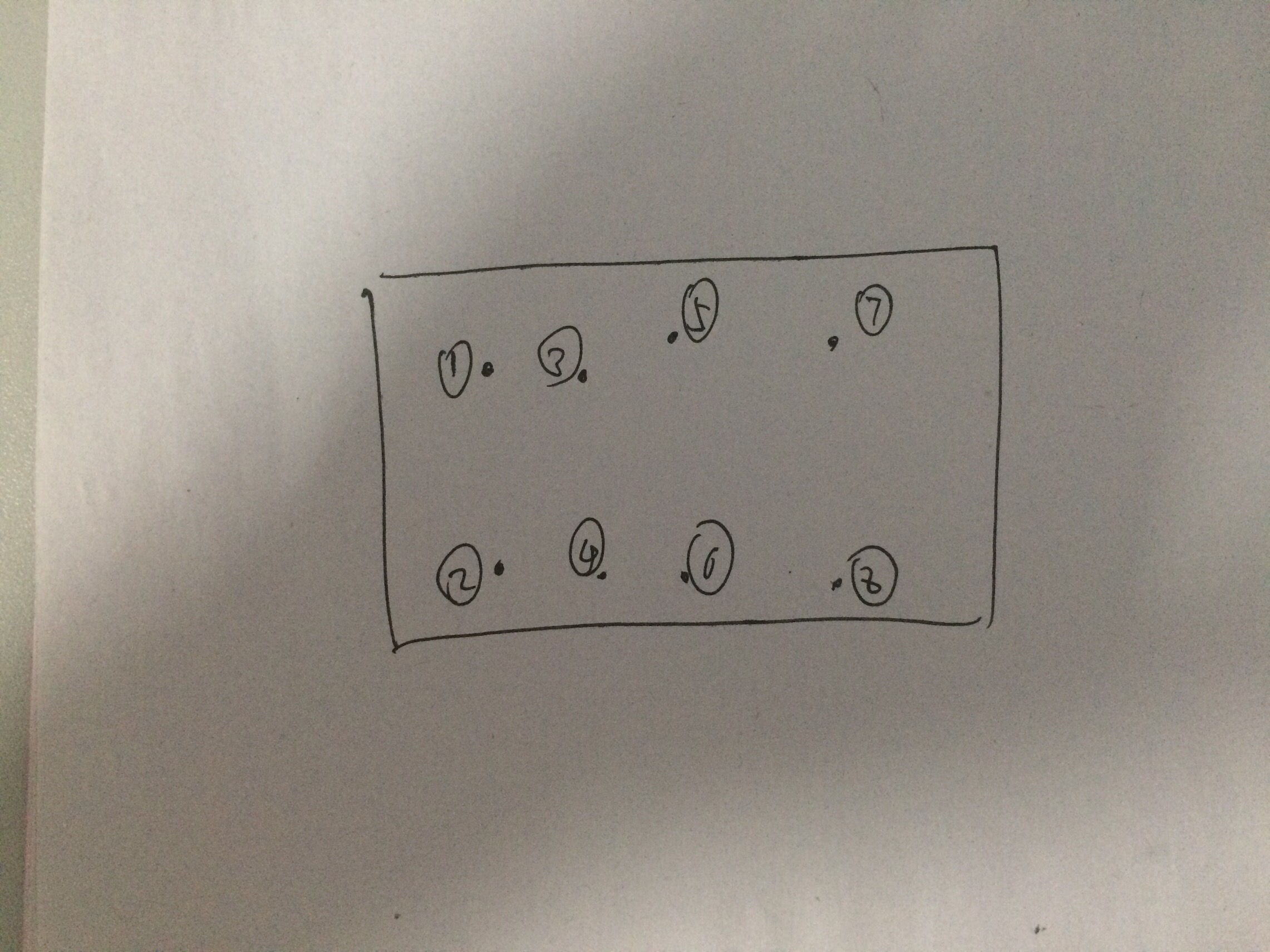
需要注意的是

firstID = [2 4 6 8 1 3 5 7 2 4 6 8];

secndID = [4 6 8 2 3 5 7 1 1 3 5 7];

这两行代码，与点的位置有关系，firstID(i)与secondID(i)相对应的点代表要连接的线段，一共12条线段。

如果八个点的坐标标号在图上为如图所示情况，那么firstID与secndID保持不变。



样例：

输入：



输出：

