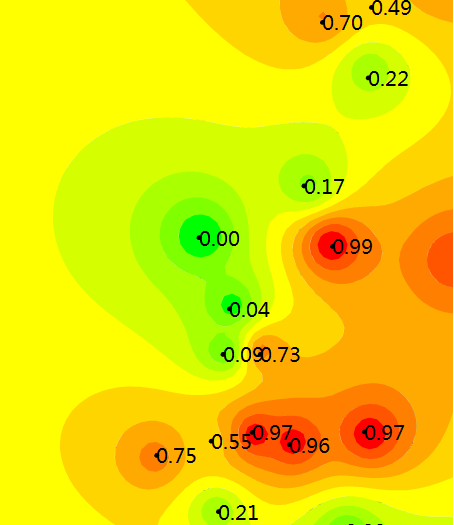
输入：离散的采样点坐标和高度值(x\_0,y\_0,value\_0),(x\_1,y\_1,value\_1)......(x\_n, y\_n, value\_n)

输出：等高线图，如下所示



　　wiki上的[Marching squares](http://en.wikipedia.org/wiki/Marching_squares)算法对此有很好的说明，我也是按照wiki上面的步骤来实现这个算法的，下面对该算法的步骤进行简要说明。

输入参数：

1、点的集合(x\_0,y\_0,value\_0),(x\_1,y\_1,value\_1)......(x\_n, y\_n, value\_n) ;

2、高度值数组，即每条等高线表示的值，以上图为例是[0.0, 0.1, 0.2, .... 1.0]共10个区间，9个等高线值；

3、颜色数组，必须与高度值数组等长。

步骤：

1、构造矩阵

计算出输入点的x,y范围，即找到xmin, ymin, xmax, ymax，形成一个rectangle,然后拟定一个step，形成一个空的矩阵，step越小，等高线越平滑，计算速度也越慢。

2、矩阵插值

由输入点的值确定矩阵上每一个元素的值，这样一来，稀疏的输入点就变成了一个稠密的矩阵。一种简单的插值方式是：

matrix(x,y) = Sum(1/dist[i]^2 \* value[i]) / Sum(1/dist[i]^2),其中dist[i]表示第i个点到(x,y)的欧式距离。

3、画等高线

到这里就可以完全按照wiki上的[Marching squares](http://en.wikipedia.org/wiki/Marching_squares)来进行了。Marching squares算法把等高线绘制分成了两种情况，其一是只画线不上色，即isoLine算法，这种比较简单，不做讨论了；其二是画等高带，即isoBand算法，也就是本文采用的方法，如下说明：

对于每一个等高带，从高度数组中取出相临的两个阈值（如0.0, 0.1），从颜色数组中取一个颜色，用两个阈值把矩阵三值化成为一个只包含0,1,2的矩阵。0表示小于矩阵元素小于这两个阈值，1表示介于其中间，2表示大于这两个阈值。

这时，你的矩阵中就只有(0,1,2）三个值了，然后考虑矩阵中的每四个元素组成的单位正方形的状态，该状态由其四个角的元素值唯一确定，共3^4 = 81种，然后根据这81种情况分类讨论，分别上色了，此处请参考wiki的marching square页面。

如此，第一种颜色的等高带就画完了，把第3步的内容重复10次，画完所有的等高线。

补充一段第3步的actionscript代码，是我按照wiki的步骤实现的

public function isoband(data:qyMatrix, thArray:Array,

dstLayer:GraphicsLayer, extent:Extent):Vector.<Polygon>

{

var x:int, y:int, k:int;

var count:int=0;

var squareWidth:Number = extent.width / (data.width - 1);

var squareHeight:Number = extent.height / (data.height - 1);

var dstPolygonVec:Vector.<Polygon> = new Vector.<Polygon>(thArray.length - 1);

// 等值线的每一个阈值

for (k=0; k<thArray.length - 1; k++){

// 3值化

var stateMat:qyMatrix = new qyMatrix(data.width, data.height);

for (y=0; y<data.height; y++){

for (x=0; x<data.width; x++){

if (data.getData(x, y) < thArray[k]){

stateMat.setData(x, y, 0);

}

else{

if (data.getData(x, y) < thArray[k+1]){

stateMat.setData(x, y, 1);

}

else{

stateMat.setData(x, y, 2);

}

}

}

}

var x1:Number, y1:Number, x2:Number, y2:Number, x3:Number, y3:Number, x4:Number, y4:Number;

var polygon:Polygon = new Polygon;

for (y=0; y<data.height - 1; y++){

var ymin:Number = extent.ymin + y / (data.height - 1) \* extent.height;

var ymax:Number = extent.ymin + (y+1) / (data.height - 1) \* extent.height;

for (x=0; x<data.width - 1; x++){

var xmin:Number = extent.xmin + x / (data.width - 1) \* extent.width;

var xmax:Number = extent.xmin + (x+1) / (data.width - 1) \* extent.width;

// square 四角坐标

var p7:MapPoint = new MapPoint (xmin, ymin);

var p9:MapPoint = new MapPoint (xmax, ymin);

var p3:MapPoint = new MapPoint (xmax, ymax);

var p1:MapPoint = new MapPoint (xmin, ymax);

// square 四角数值

var d7:Number = data.getData(x, y);

var d9:Number = data.getData(x+1, y);

var d3:Number = data.getData(x+1, y+1);

var d1:Number = data.getData(x, y+1);

var mid:Number;

// isoband的顶点坐标

var pt1:MapPoint = null;

var pt2:MapPoint = null;

var pt3:MapPoint = null;

var pt4:MapPoint = null;

var pt5:MapPoint = null;

var pt6:MapPoint = null;

var pt7:MapPoint = null;

var pt8:MapPoint = null;

var squareState:String = getSquareState(stateMat, x, y);

switch (squareState) // total 81 cases

{

// no color

case "2222":

case "0000":

break;

// square

case "1111":

polygon.addRing([p7, p9, p3, p1]);

break;

// triangle 8 cases

case "2221":

x1 = p1.x + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

y2 = p1.y - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, p1.y);

pt2 = new MapPoint(p7.x, y2);

polygon.addRing([pt1, p1, pt2]);

break;

case "2212":

y1 = p3.y - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

x2 = p3.x - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

pt1 = new MapPoint(p9.x, y1);

pt2 = new MapPoint(x2, p1.y);

polygon.addRing([pt1, p3, pt2]);

break;

case "2122":

x1 = p9.x - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

y2 = p9.y + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

pt1 = new MapPoint(x1, p7.y);

pt2 = new MapPoint(p9.x, y2)

polygon.addRing([pt1, p9, pt2]);

break;

case "1222":

x1 = p7.x + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y2 = p7.y + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, p7.y);

pt2 = new MapPoint(p7.x, y2);

polygon.addRing([p7, pt1, pt2]);

break;

case "0001":

x1 = p3.x - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

y2 = p7.y + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, p1.y);

pt2 = new MapPoint(p7.x, y2);

polygon.addRing([pt1, p1, pt2]);

break;

case "0010":

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

x2 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

pt1 = new MapPoint(xmax, y1);

pt2 = new MapPoint(x2, ymax);

polygon.addRing([pt1, p3, pt2]);

break;

case "0100":

x1 = p7.x + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

y2 = p3.y - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

pt1 = new MapPoint(x1, p9.y);

pt2 = new MapPoint(p9.x, y2);

polygon.addRing([pt1, p9, pt2]);

break;

case "1000":

x1 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y2 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmin, y2);

polygon.addRing([p7, pt1, pt2]);

break;

// trapezoid 8 cases

case "2220":

x1 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

x2 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

y1 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

y2 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, p1.y);

pt2 = new MapPoint(x2, p1.y);

pt3 = new MapPoint(p7.x, y1);

pt4 = new MapPoint(p7.x, y2);

polygon.addRing([pt1, pt2, pt3, pt4]);

break;

case "2202":

y1 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

y2 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

x1 = xmax - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

x2 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

pt1 = new MapPoint(xmax, y1);

pt2 = new MapPoint(xmax, y2);

pt3 = new MapPoint(x1, ymax);

pt4 = new MapPoint(x2, ymax);

polygon.addRing([pt1, pt2, pt3, pt4]);

break;

case "2022":

x1 = p9.x - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

x2 = p9.x - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y1 = p9.y + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

y2 = p9.y + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

pt1 = new MapPoint(x1, p9.y);

pt2 = new MapPoint(x2, p9.y);

pt3 = new MapPoint(p9.x ,y1);

pt4 = new MapPoint(p9.x, y2);

polygon.addRing([pt1, pt2, pt3, pt4]);

break;

case "0222":

x1 = p7.x + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

x2 = p7.x + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y1 = p7.y + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

y2 = p7.y + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, p7.y);

pt2 = new MapPoint(x2, p7.y);

pt3 = new MapPoint(p7.x, y1);

pt4 = new MapPoint(p7.x, y2);

polygon.addRing([pt1, pt2, pt3, pt4]);

break;

case "0002":

x1 = p3.x - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

x2 = p3.x - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

y1 = p7.y + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

y2 = p7.y + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, p1.y);

pt2 = new MapPoint(x2, p1.y);

pt3 = new MapPoint(p1.x, y1);

pt4 = new MapPoint(p1.x, y2);

polygon.addRing([pt1, pt2, pt3, pt4]);

break;

case "0020":

y1 = p9.y + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

y2 = p9.y + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

x1 = p1.x + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

x2 = p1.x + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

pt1 = new MapPoint(p3.x, y1);

pt2 = new MapPoint(p3.x, y2);

pt3 = new MapPoint(x1, p3.y);

pt4 = new MapPoint(x2, p3.y);

polygon.addRing([pt1, pt2, pt3, pt4]);

break;

case "0200":

x1 = p7.x + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

x2 = p7.x + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y1 = p3.y - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

y2 = p3.y - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

pt1 = new MapPoint(x1, p9.y);

pt2 = new MapPoint(x2, p9.y);

pt3 = new MapPoint(p9.x, y1);

pt4 = new MapPoint(p9.x, y2);

polygon.addRing([pt1, pt2, pt3, pt4]);

break;

case "2000":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

x2 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y1 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

y2 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(xmin, y1);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([pt1, pt2, pt3, pt4]);

break;

// rectangle 12 cases

case "0011":

y1 = p9.y + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

y2 = p7.y + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(p9.x, y1);

pt2 = new MapPoint(p7.x, y2);

polygon.addRing([pt1, p3, p1, pt2]);

break;

case "0110":

x1 = p7.x + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

x2 = p1.x + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

pt1 = new MapPoint(x1, p7.y);

pt2 = new MapPoint(x2, p1.y);

polygon.addRing([pt1, p9, p3, pt2]);

break;

case "1100":

y1 = p3.y - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

y2 = p1.y - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(p9.x, y1);

pt2 = new MapPoint(p7.x, y2);

polygon.addRing([p7, p9, pt1, pt2]);

break;

case "1001":

x1 = p9.x - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

x2 = p3.x - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

pt1 = new MapPoint(x1, p7.y);

pt2 = new MapPoint(x2, p1.y);

polygon.addRing([p7, pt1, pt2, p1]);

break;

case "2211":

y1 = p3.y - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

y2 = p1.y - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(xmax, y1);

pt2 = new MapPoint(xmin, y2);

polygon.addRing([pt1, p3, p1, pt2]);

break;

case "2112":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

x2 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymax);

polygon.addRing([pt1, p9, p3, pt2]);

break;

case "1122":

y1 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareWidth;

y2 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareWidth;

pt1 = new MapPoint(xmax, y1);

pt2 = new MapPoint(xmin, y2);

polygon.addRing([p7, p9, pt1, pt2]);

break;

case "1221":

x1 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

x2 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymax);

polygon.addRing([p7, pt1, pt2, p1]);

break;

case "2200":

y1 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

y2 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

y3 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

y4 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(xmax, y1);

pt2 = new MapPoint(xmax, y2);

pt3 = new MapPoint(xmin, y3);

pt4 = new MapPoint(xmin, y4);

polygon.addRing([pt1, pt2, pt3, pt4]);

break;

case "2002":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

x2 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

x3 = xmax - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

x4 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(x3, ymax);

pt4 = new MapPoint(x4, ymax);

polygon.addRing([pt1, pt2, pt3, pt4]);

break;

case "0022":

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

y2 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

y3 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

y4 = ymin + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(xmax, y1);

pt2 = new MapPoint(xmax, y2);

pt3 = new MapPoint(xmin, y3);

pt4 = new MapPoint(xmin, y4);

polygon.addRing([pt1, pt2, pt3, pt4]);

break;

case "0220":

x1 = xmin + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

x2 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

x3 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

x4 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(x3, ymax);

pt4 = new MapPoint(x4, ymax);

polygon.addRing([pt1, pt2, pt3, pt4]);

break;

// hexagon 12 cases

case "0211":

x1 = xmin + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

x2 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y1 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

y2 = ymin + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(xmax, y1);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([pt1, pt2, pt3, p3, p1, pt4]);

break;

case "2110":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

x2 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

y1 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

y2 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymax);

pt3 = new MapPoint(xmin, y1);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([pt1, p9, p3, pt2, pt3, pt4]);

break;

case "1102":

y1 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

x1 = xmax - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

x2 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

y2 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(xmax, y1);

pt2 = new MapPoint(x1, ymax);

pt3 = new MapPoint(x2, ymax);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([p7, p9, pt1, pt2, pt3, pt4]);

break;

case "1021":

x1 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

y2 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

x2 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(xmax, y2);

pt4 = new MapPoint(x2, ymax);

polygon.addRing([p7, pt1, pt2, pt3, pt4, p1]);

break;

case "2011":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

x2 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

y2 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(xmax, y1);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([pt1, pt2, pt3, p3, p1, pt4]);

break;

case "0112":

x1 = xmin + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

x2 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

y1 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

y2 = ymin + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymax);

pt3 = new MapPoint(xmin, y1);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([pt1, p9, p3, pt2, pt3, pt4]);

break;

case "1120":

y1 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

x1 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

x2 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

y2 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(xmax, y1);

pt2 = new MapPoint(x1, ymax);

pt3 = new MapPoint(x2, ymax);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([p7, p9, pt1, pt2, pt3, pt4]);

break;

case "1201":

x1 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y1 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

y2 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

x2 = xmax - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(xmax, y2);

pt4 = new MapPoint(x2, ymax);

polygon.addRing([p7, pt1, pt2, pt3, pt4, p1]);

break;

case "2101":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

y1 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

x2 = xmax - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

y2 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(x2, ymax);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([pt1, p9, pt2, pt3, p1, pt4]);

break;

case "0121":

x1 = xmin + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

y1 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareWidth;

x2 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

y2 = ymin + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(x2, ymax);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([pt1, p9, pt2, pt3, p1, pt4]);

break;

case "1012":

x1 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

x2 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

y2 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(x2, ymax);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([p7, pt1, pt2, p3, pt3, pt4]);

break;

case "1210":

x1 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y1 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

x2 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

y2 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(x2, ymax);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([p7, pt1, pt2, p3, pt3, pt4]);

break;

// pentagon 24 cases

case "1211":

x1 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y2 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y2);

polygon.addRing([p7, pt1, pt2, p3, p1]);

break;

case "2111":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

y2 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmin, y2);

polygon.addRing([pt1, p9, p3, p1, pt2]);

break;

case "1112":

x1 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

y2 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymax);

pt2 = new MapPoint(xmin, y2);

polygon.addRing([p7, p9, p3, pt1, pt2]);

break;

case "1121":

y1 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

x2 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

pt1 = new MapPoint(xmax, y1);

pt2 = new MapPoint(x2, ymax);

polygon.addRing([p7, p9, pt1, pt2, p1]);

break;

case "1011":

x1 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y2 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y2);

polygon.addRing([p7, pt1, pt2, p3, p1]);

break;

case "0111":

x1 = xmin + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

y2 = ymin + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmin, y2);

polygon.addRing([pt1, p9, p3, p1, pt2]);

break;

case "1110":

x1 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

y2 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymax);

pt2 = new MapPoint(xmin, y2);

polygon.addRing([p7, p9, p3, pt1, pt2]);

break;

case "1101":

y1 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

x2 = xmax - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

pt1 = new MapPoint(xmax, y1);

pt2 = new MapPoint(x2, ymax);

polygon.addRing([p7, p9, pt1, pt2, p1]);

break;

case "1200":

x1 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y1 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

y2 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

y3 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(xmax, y2);

pt4 = new MapPoint(xmin, y3);

polygon.addRing([p7, pt1, pt2, pt3, pt4]);

break;

case "0120":

x1 = xmin + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

y1 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

x2 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

x3 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(x2, ymax);

pt4 = new MapPoint(x3, ymax);

polygon.addRing([pt1, p9, pt2, pt3, pt4]);

break;

case "0012":

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

x1 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

y2 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

y3 = ymin + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(xmax, y1);

pt2 = new MapPoint(x1, ymax);

pt3 = new MapPoint(xmin, y2);

pt4 = new MapPoint(xmin, y3);

polygon.addRing([pt1, p3, pt2, pt3, pt4]);

break;

case "2001":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

x2 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

x3 = xmax - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

y1 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(x3, ymax);

pt4 = new MapPoint(xmin, y1);

polygon.addRing([pt1, pt2, pt3, p1, pt4]);

break;

case "1022":

x1 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

y2 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

y3 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(xmax, y2);

pt4 = new MapPoint(xmin, y3);

polygon.addRing([p7, pt1, pt2, pt3, pt4]);

break;

case "2102":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

y1 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

x2 = xmax - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

x3 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(x2, ymax);

pt4 = new MapPoint(x3, ymax);

polygon.addRing([pt1, p9, pt2, pt3, pt4]);

break;

case "2210":

y1 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

x1 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

y2 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

y3 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(xmax, y1);

pt2 = new MapPoint(x1, ymax);

pt3 = new MapPoint(xmin, y2);

pt4 = new MapPoint(xmin, y3);

polygon.addRing([pt1, p3, pt2, pt3, pt4]);

break;

case "0221":

x1 = xmin + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

x2 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

x3 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

y1 = ymin + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(x3, ymax);

pt4 = new MapPoint(xmin, y1);

polygon.addRing([pt1, pt2, pt3, p1, pt4]);

break;

case "1002":

x1 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

x2 = xmax - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

x3 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

y1 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymax);

pt3 = new MapPoint(x3, ymax);

pt4 = new MapPoint(xmin, y1);

polygon.addRing([p7, pt1, pt2, pt3, pt4]);

break;

case "2100":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

y1 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

y2 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

y3 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(xmin, y2);

pt4 = new MapPoint(xmin, y3);

polygon.addRing([pt1, p9, pt2, pt3, pt4]);

break;

case "0210":

x1 = xmin + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

x2 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y1 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

x3 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(xmax, y1);

pt4 = new MapPoint(x3, ymax);

polygon.addRing([pt1, pt2, pt3, p3, pt4]);

break;

case "0021":

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

y2 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

x1 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

y3 = ymin + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(xmax, y1);

pt2 = new MapPoint(xmax, y2);

pt3 = new MapPoint(x1, ymax);

pt4 = new MapPoint(xmin, y3);

polygon.addRing([pt1, pt2, pt3, p1, pt4]);

break;

case "1220":

x1 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

x2 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

x3 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

y1 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymax);

pt3 = new MapPoint(x3, ymax);

pt4 = new MapPoint(xmin, y1);

polygon.addRing([p7, pt1, pt2, pt3, pt4]);

break;

case "0122":

x1 = xmin + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

y1 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

y2 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

y3 = ymin + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(xmin, y2);

pt4 = new MapPoint(xmin, y3);

polygon.addRing([pt1, p9, pt2, pt3, pt4]);

break;

case "2012":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

x2 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

x3 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(xmax, y1);

pt4 = new MapPoint(x3, ymax);

polygon.addRing([pt1, pt2, pt3, p3, pt4]);

break;

case "2201":

y1 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

y2 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

x1 = xmax - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

y3 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(xmax, y1);

pt2 = new MapPoint(xmax, y2);

pt3 = new MapPoint(x1, ymax);

pt4 = new MapPoint(xmin, y3);

polygon.addRing([pt1, pt2, pt3, p1, pt4]);

break;

// saddles - 8 sided 2 cases

case "2020":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

x2 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

y2 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

x3 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

x4 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

y3 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

y4 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(xmax, y1);

pt4 = new MapPoint(xmax, y2);

pt5 = new MapPoint(x3, ymax);

pt6 = new MapPoint(x4, ymax);

pt7 = new MapPoint(xmin, y3);

pt8 = new MapPoint(xmin, y4);

mid = (d7 + d9 + d3 + d1) / 4;

if (mid < thArray[k]){

polygon.addRing([pt1, pt2, pt7, pt8]);

polygon.addRing([pt3, pt4, pt5, pt6]);

}

else if (mid < thArray[k+1]){

polygon.addRing([pt1, pt2, pt3, pt4, pt5, pt6, pt7, pt8]);

}

else{

polygon.addRing([pt1, pt2, pt3, pt4]);

polygon.addRing([pt5, pt6, pt7, pt8]);

}

break;

case "0202":

x1 = xmin + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

x2 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y1 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

y2 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

x3 = xmax - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

x4 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

y3 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

y4 = ymin + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(xmax, y1);

pt4 = new MapPoint(xmax, y2);

pt5 = new MapPoint(x3, ymax);

pt6 = new MapPoint(x4, ymax);

pt7 = new MapPoint(xmin, y3);

pt8 = new MapPoint(xmin, y4);

mid = (d7 + d9 + d3 + d1) / 4;

if (mid < thArray[k]){

polygon.addRing([pt1, pt2, pt3, pt4]);

polygon.addRing([pt5, pt6, pt7, pt8]);

}

else if (mid < thArray[k+1]){

polygon.addRing([pt1, pt2, pt3, pt4, pt5, pt6, pt7, pt8]);

}

else{

polygon.addRing([pt1, pt2, pt7, pt8]);

polygon.addRing([pt3, pt4, pt5, pt6]);

}

break;

// saddles: 6 sided 4 cases

case "0101":

mid = (d7 + d9 + d3 + d1) / 4;

if (mid < thArray[k]){

x1 = p7.x + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

y2 = p3.y - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

pt1 = new MapPoint(x1, p9.y);

pt2 = new MapPoint(p9.x, y2);

polygon.addRing([pt1, p9, pt2]);

x1 = p3.x - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

y2 = p7.y + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, p1.y);

pt2 = new MapPoint(p7.x, y2);

polygon.addRing([pt1, p1, pt2]);

}

else{

x1 = xmin + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

y1 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

x2 = xmax - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

y2 = ymin + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(x2, ymax);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([pt1, p9, pt2, pt3, p1, pt4]);

}

break;

case "1010":

mid = (d7 + d9 + d3 + d1) / 4;

if (mid < thArray[k]){

x1 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y2 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmin, y2);

polygon.addRing([p7, pt1, pt2]);

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

x2 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

pt1 = new MapPoint(p9.x, y1);

pt2 = new MapPoint(x2, p1.y);

polygon.addRing([pt1, p3, pt2]);

}

else{

x1 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

x2 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

y2 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(x2, ymax);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([p7, pt1, pt2, p3, pt3, pt4]);

}

break;

case "2121":

mid = (d7 + d9 + d3 + d1) / 4;

if (mid < thArray[k+1]){

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

y1 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

x2 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

y2 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(x2, ymax);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([pt1, p9, pt2, pt3, p1, pt4]);

}

else{

x1 = p9.x - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

y2 = p9.y + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

pt1 = new MapPoint(x1, p7.y);

pt2 = new MapPoint(p9.x, y2)

polygon.addRing([pt1, p9, pt2]);

x1 = p1.x + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

y2 = p1.y - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, p1.y);

pt2 = new MapPoint(p7.x, y2);

polygon.addRing([pt1, p1, pt2]);

}

break;

case "1212":

mid = (d7 + d9 + d3 + d1) / 4;

if (mid < thArray[k+1]){

x1 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y1 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

x2 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

y2 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(x2, ymax);

pt4 = new MapPoint(xmin, y2);

polygon.addRing([p7, pt1, pt2, p3, pt3, pt4]);

}

else{

x1 = p7.x + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y2 = p7.y + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, p7.y);

pt2 = new MapPoint(p7.x, y2);

polygon.addRing([p7, pt1, pt2]);

y1 = p3.y - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

x2 = p3.x - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

pt1 = new MapPoint(p9.x, y1);

pt2 = new MapPoint(x2, p1.y);

polygon.addRing([pt1, p3, pt2]);

}

break;

// saddles 7 sided 8 cases

case "2120":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

y1 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

x2 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

x3 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

y2 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

y3 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(x2, ymax);

pt4 = new MapPoint(x3, ymax);

pt5 = new MapPoint(xmin, y2);

pt6 = new MapPoint(xmin, y3);

mid = (d7 + d9 + d3 + d1) / 4;

if (mid < thArray[k+1]){

polygon.addRing([pt1, p9, pt2, pt3, pt4, pt5, pt6]);

}

else{

polygon.addRing([pt1, p9, pt2]);

polygon.addRing([pt3, pt4, pt5, pt6]);

}

break;

case "2021":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

x2 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

y2 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

x3 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

y3 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(xmax, y1);

pt4 = new MapPoint(xmax, y2);

pt5 = new MapPoint(x3, ymax);

pt6 = new MapPoint(xmin, y3);

mid = (d7 + d9 + d3 + d1) / 4;

if (mid < thArray[k+1]){

polygon.addRing([pt1, pt2, pt3, pt4, pt5, p1, pt6]);

}

else{

polygon.addRing([pt1, pt2, pt3, pt4]);

polygon.addRing([pt5, p1, pt6]);

}

break;

case "1202":

x1 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y1 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

y2 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

x2 = xmax - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

x3 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

y3 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(xmax, y2);

pt4 = new MapPoint(x2, ymax);

pt5 = new MapPoint(x3, ymax);

pt6 = new MapPoint(xmin, y3);

mid = (d7 + d9 + d3 + d1) / 4;

if (mid < thArray[k+1]){

polygon.addRing([p7, pt1, pt2, pt3, pt4, pt5, pt6]);

}

else{

polygon.addRing([p7, pt1, pt6]);

polygon.addRing([pt2, pt3, pt4, pt5]);

}

break;

case "0212":

x1 = xmin + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

x2 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y1 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

x3 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

y2 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

y3 = ymin + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(xmax, y1);

pt4 = new MapPoint(x3, ymax);

pt5 = new MapPoint(xmin, y2);

pt6 = new MapPoint(xmin, y3);

mid = (d7 + d9 + d3 + d1) / 4;

if (mid < thArray[k+1]){

polygon.addRing([pt1, pt2, pt3, p3, pt4, pt5, pt6]);

}

else{

polygon.addRing([pt1, pt2, pt5, pt6]);

polygon.addRing([pt3, p3, pt4]);

}

break;

case "0102":

x1 = xmin + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

y1 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

x2 = xmax - (thArray[k] - d3) / (d1 - d3) \* squareWidth;

x3 = xmax - (thArray[k+1] - d3) / (d1 - d3) \* squareWidth;

y2 = ymin + (thArray[k+1] - d7) / (d1 - d7) \* squareHeight;

y3 = ymin + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(x2, ymax);

pt4 = new MapPoint(x3, ymax);

pt5 = new MapPoint(xmin, y2);

pt6 = new MapPoint(xmin, y3);

mid = (d7 + d9 + d3 + d1) / 4;

if (mid < thArray[k]){

polygon.addRing([pt1, p9, pt2]);

polygon.addRing([pt3, pt4, pt5, pt6]);

}

else{

polygon.addRing([pt1, p9, pt2, pt3, pt4, pt5, pt6]);

}

break;

case "0201":

x1 = xmin + (thArray[k] - d7) / (d9 - d7) \* squareWidth;

x2 = xmin + (thArray[k+1] - d7) / (d9 - d7) \* squareWidth;

y1 = ymax - (thArray[k+1] - d3) / (d9 - d3) \* squareHeight;

y2 = ymax - (thArray[k] - d3) / (d9 - d3) \* squareHeight;

x3 = xmax - (thArray[k] - d3) / (d1- d3) \* squareWidth;

y3 = ymin + (thArray[k] - d7) / (d1 - d7) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(xmax, y1);

pt4 = new MapPoint(xmax, y2);

pt5 = new MapPoint(x3, ymax);

pt6 = new MapPoint(xmin, y3);

mid = (d7 + d9 + d3 + d1) / 4;

if (mid < thArray[k]){

polygon.addRing([pt1, pt2, pt3, pt4]);

polygon.addRing([pt5, p1, pt6]);

}

else{

polygon.addRing([pt1, pt2, pt3, pt4, pt5, p1, pt6]);

}

break;

case "1020":

x1 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

y2 = ymin + (thArray[k+1] - d9) / (d3 - d9) \* squareHeight;

x2 = xmin + (thArray[k+1] - d1) / (d3 - d1) \* squareWidth;

x3 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

y3 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(xmax, y1);

pt3 = new MapPoint(xmax, y2);

pt4 = new MapPoint(x2, ymax);

pt5 = new MapPoint(x3, ymax);

pt6 = new MapPoint(xmin, y3);

mid = (d7 + d9 + d3 + d1) / 4;

if (mid < thArray[k]){

polygon.addRing([p7, pt1, pt6]);

polygon.addRing([pt2, pt3, pt4, pt5,]);

}

else{

polygon.addRing([p7, pt1, pt2, pt3, pt4, pt5, pt6]);

}

break;

case "2010":

x1 = xmax - (thArray[k+1] - d9) / (d7 - d9) \* squareWidth;

x2 = xmax - (thArray[k] - d9) / (d7 - d9) \* squareWidth;

y1 = ymin + (thArray[k] - d9) / (d3 - d9) \* squareHeight;

x3 = xmin + (thArray[k] - d1) / (d3 - d1) \* squareWidth;

y2 = ymax - (thArray[k] - d1) / (d7 - d1) \* squareHeight;

y3 = ymax - (thArray[k+1] - d1) / (d7 - d1) \* squareHeight;

pt1 = new MapPoint(x1, ymin);

pt2 = new MapPoint(x2, ymin);

pt3 = new MapPoint(xmax, y1);

pt4 = new MapPoint(x3, ymax);

pt5 = new MapPoint(xmin, y2);

pt6 = new MapPoint(xmin, y3);

mid = (d7 + d9 + d3 + d1) / 4;

if (mid < thArray[k]){

polygon.addRing([pt1, pt2, pt5, pt6]);

polygon.addRing([pt3, p3, pt4]);

}

else{

polygon.addRing([pt1, pt2, pt3, p3, pt4, pt5, pt6]);

}

break;

default:

break;

}

}

}

dstPolygonVec[k] = polygon;

//DrawToolBox.qyPolygon(dstLayer, polygon, 0, 0, 0.8, colorArray[k]);

}

return dstPolygonVec;

function getSquareState(mat:qyMatrix, x:int, y:int):String

{

var str:String = '';

str += String(mat.getData(x, y));

str += String(mat.getData(x + 1, y));

str += String(mat.getData(x + 1, y + 1));

str += String(mat.getData(x, y + 1));

return str;

}

}