

HW5——Image warping

几何变换

全局变换

控制点：控制点对的坐标决定了变换函数

变换类型：

前向变换和后向变换

最近邻：用欧氏距离最相近的原像素点替代

双线性插值：

Image **Morphing**

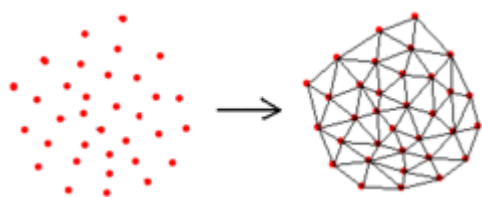
1. 输入控制点对
2. 用Delaunay算法定义三角形mesh
3. 变换每个三角形：仿射变换

控制点对

三角剖分算法

定义

【定义】Delaunay边：假设 E 中的一条边 e （两个端点为 a, b ）， e



若满足下列条件，则称之为Delaunay边：存在一个圆经过 a, b 两点，圆内(注意是圆内，圆上最多三点共圆)不含点集 V 中任何其他的点，这一特性又称空圆特性。

【定义】Delaunay三角剖分：如果点集 V 的一个三角剖分 T 只包含Delaunay边，那么该三角剖分称为Delaunay三角剖分。

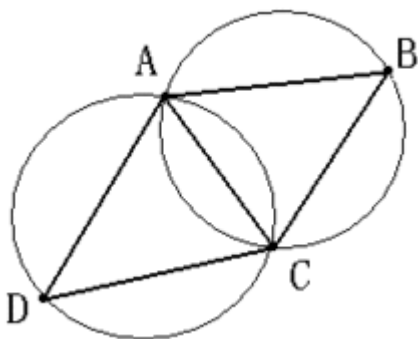
优化处理:

理论上为了构造Delaunay三角网 [2]，Lawson提出的局部优化过程LOP(Local Optimization Procedure)，一般三角网经过LOP处理，即可确保成为Delaunay三角网，其基本做法如下所示：

1. 将两个具有共同边的三角形合成一个多边形。
2. 以最大空圆准则作检查，看其第四个顶点是否在三角形的**外接圆**之内。

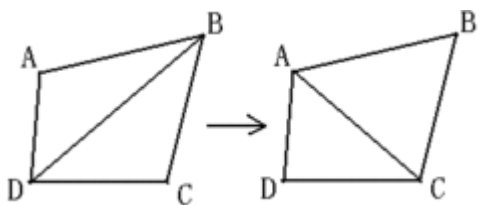
3.如果在，修正对角线即将对角线对调，即完成局部优化过程的处理。

准则:



要满足Delaunay三角剖分的定义，必须符合两个重要的准则：

- 1、空圆特性：Delaunay三角网是唯一的（任意四点不能共圆），在Delaunay三角形网中任一三角形的外接圆范围内不会有其它点存在。如下图所示：
- 2、最大化最小角特性：在散点集可能形成的三角剖分中，Delaunay三角剖分所形成的三角形的最小角最大。从这个意义上讲，Delaunay三角网是“最接近于规则化的”三角网。具体的说是指在两个相邻的三角形构成凸四边形的对角线，在相互交换后，六个内角的最小角不再增大。如下图所示：



特性:

以下是Delaunay剖分所具备的优异特性：

- 1.最接近：以最近的三点形成三角形，且各线段(三角形的边)皆不相交。
- 2.唯一性：不论从区域何处开始构建，最终都将得到一致的结果。
- 3.最优性：任意两个相邻三角形形成的凸四边形的对角线如果可以互换的话，那么两个三角形六个内角中最小的角度不会变大。
- 4.最规则：如果将三角网中的每个三角形的最小角进行升序排列，则Delaunay三角网的排列得到的数值最大。
- 5.区域性：新增、删除、移动某一个顶点时只会影响临近的三角形。
- 6.具有凸多边形的外壳：三角网最外层的边界形成一个凸多边形的外壳。

算法

1. 逐点插入

基本原理：首先建立一个大的三角形或者多边形使其能够吧所有数据点包围起来。向其中插入一点，该点与包含它的三角形三个顶点相连，形成三个新三角形；对四个点逐个进行空外接圆检测，并LOP局部优化。

2. Watson算法基本步骤

1. 构造一个包含所有散点的超级三角形，放入三角形列表；

2. 插入一个散点，在三角形链表中找出外接圆包含插入点的三角形（影响三角形），删除影响三角形的公共边，将插入点同影响三角形的全部定点连接起来。

3. LOP局部优化

仿射变换

对于每一个三角形计算：

以原图像三点得到的U1

$$U1 = \begin{bmatrix} pset1[k].x & pset1[k].x & pset1[k].x \\ pset1[k].y & pset1[k].y & pset1[k].y \\ 1 & 1 & 1 \end{bmatrix}$$

以目标图像得到的U2

$$U2 = \begin{bmatrix} pset2[k].x & pset2[k].x & pset2[k].x \\ pset2[k].y & pset2[k].y & pset2[k].y \\ 1 & 1 & 1 \end{bmatrix}$$

以中间图像得到的A

$$A = \begin{bmatrix} p1.x & p2.x & p3.x \\ p1.y & p2.y & p3.y \\ 1 & 1 & 1 \end{bmatrix}$$

A的逆

$$A^{-1}$$

映射矩阵

$$m1 = U1 * A^{-1}$$

$$m2 = U2 * A^{-1}$$

后向映射到原图像

$$u1 = m1[0][0] * x + m1[0][1] * y + m1[0][2];$$

$$v1 = m1[1][0] * x + m1[1][1] * y + m1[1][2];$$

后向映射到目标图像

$$u2 = m2[1][0] * x + m2[1][1] * y + m2[1][2];$$

$$v2 = m2[1][0] * x + m2[1][1] * y + m2[1][2];$$

分别做双线性插值得到后向映射的点在两图中的rgb值，用 $(1 - \alpha) * rgb1 + \alpha * rgb2$ 得到中间图像某位置的像素值。

测试结果

