

摘 要

变形,是指从初始物体到目标物体的连续、光滑、自然的过渡(这里的物体可以是数字图像、曲线、曲面、网格等)。变形有着十分广泛的应用,如计算机图形学、动画设计、工业造型、科学计算可视化、电影特技等。本文对同构平面三角网格的变形和平面多边形的变形算法进行了研究,主要的研究结果如下:

1) 平面多边形的相似同构三角剖分。本文提出了基于初始多边形和目标多边形之间相似性的同构三角剖分算法。该方法将初始多边形和目标多边形的相似性进行分析,在不添加额外顶点的情况下,首先进行初始和目标多边形的相似部分三角剖分,得到相应的简化初末多边形,该过程能够起到对初始和目标多边形简化的作用,对此简化初始和目标多边形进行相应的同构三角剖分。这样能够减少在原始的初始和目标多边形之间的同构三角剖分的计算,以达到好的的效果。应用该算法能够减少进行同构三角剖分需要增加的额外顶点的个数,从而能够减少多边形变形的复杂度和计算量。

2) 基于小波的平面多边形变形算法。该算法首先将初始和目标多边形进行适当的小波分解,得到相应的轮廓多边形和细节部分,再分别对轮廓多边形和细节进行变形,最后通过重构算法得到中间多边形。该算法在保持多边形轮廓不自交上有显著的改善,因为该算法能够在很大程度上保证轮廓多边形不自交,从而能够将该算法与同构三角网格变形算法相结合,结合两者的优点。通过实践证明运用了该算法能够减轻同构三角剖分的难度,以及减少进行变形的计算量,进而达到实时的效果,而且变形取得令人满意的效果。

3) 平面凸网格的保凸变形研究。本文提出了具有不同凸边界的同构平面三角网格的保凸变形算法。本文提出了基于角度的保凸变形算法变形网格的凸边界,并给出了该算法保凸的理论证明,且具有凸边界内角按同一速度变化的性质。对网格的内部顶点采用凸组方法。本文方法能够保证网格边界在变形过程中始终保持凸性,且任意时刻的中间网格与初末网格同构,即不产生自交现象。

关键词: 同构三角网格, 相似性剖分, 简化多边形, 小波变换, 多分辨分析, 轮廓多边形, 细节信息, 额外顶点, 避免自交, 保凸变形, 凸组合

Abstract

Morphing, also known as metamorphosis, is the gradual, consecutive, slippery and natural transformation of the source object into the target object (the object maybe digital image, curve surface, gridding and so on). Morphing has wide practical use in areas such as computer graphics, animation design, industrial modeling, science computation visualization, film stunt and so on. This paper makes researches on the morphing of planar compatible triangulations and planar polygons, and the main results are as follows:

1) The similitude compatible triangulations of planar polygons. This paper presents a compatible triangulations arithmetic which is based upon the comparability between the source polygon and the target polygon. This arithmetic takes the comparability into account. Firstly, triangulates the similitude parts between the source polygon and target polygon, needing no extra vertex in this process, this can simplify the two polygons, called the two resulted polygons as simplified source polygon and simplified target polygon. Using the already existing arithmetic, the two simplified polygons can be triangulated. Then the computation can be decreased for triangulating the original polygons and the results of triangulation are satisfied. The number of extra vertexes can be decreased, so the complexity and the computation are decreased for metamorphosizing the polygons.

2) The morph arithmetic of polygons is based upon waveletes. Firstly, the source polygon and the target polygon are decomposed at the same level, as a result, the overall shapes of the polygons and the details of the polygons can be gotten. Secondly, the overall shapes and the details are metamorphosized separately. Lastly, the middle polygon can be reconstructed via reconstructed arithmetic according to the middle overall shapes and the middle details. The intersection-free of the overall shape of the polygons is largely improved, so this arithmetic can be merged with the morph arithmetic of the compatible triangulation, to exert the virtues of the two arithmetics. There are some examples to show that this arithmetic can decreased the complication of compatible triangulation, bringing on the decreased the computation of morph and

the process is real time. The morph results are satisfied.

3) This paper goes along with the research on convexity-preserving when planar convex griddings metamorphosize. When two polygons that have different convex boundaries metamorphosize, the arithmetic presented in this paper can keep the boundary convexity. This arithmetic is base on the angle and that is used to metamorphosize the boundaries of the griddings. The character that this arithmetic is convexity-preserving is proved in this paper and this arithmetic has the character that all the angles in the polygon have the same transformational rate. The inner vertexes can be metamorphosized via convex combination. The arithmetic in this paper can keep the boundary protruding all the time and the middle gridding is compatible with source gridding and target gridding, the phenomena of intersection will not appear in the whole process of metamorphosise.

Key word: compatible triangulations, similitude compatible triangulation, simplified polygons, wavelet transformation, multi-resolution-analysis, overall shape polygon, detail, extra vertex, intersection-free, convexity-preserving, convex combination

第一章 绪论

1.1 变形概述和研究背景

变形(Metamorphosis 或 Morphing),也称为形状调配或融合(Shape blending)、形状平均(Shape averaging)等,是指从初始物体到目标物体的连续、光滑、自然的过渡(这里的物体可以是数字图像、多边形、多面体等)。变形技术在许多领域有着十分广泛的应用,如计算机图形学、动画设计、工业造型设计、虚拟现实、科学计算可视化、电影特技制作等领域。变形应用于计算机动画,可以减轻美工师的手工劳动;在产品造型设计中,应用变形可以把不同造型特征相互交融,产生新的造型系列;变形应用于电影制作,可以产生奇特的电影特技效果,给人以美的视觉享受。随着现代信息社会和变形技术的不断发展,变形将会有更加广泛的应用。变形问题已经引起了广泛的关注和研究,比如二维图像变形研究[8、21、25、32、38、49、50、51、52、60、68],多边形变形及多边形曲线变形研究[12、16、17、24、34、36、42、44、57、58、60、61、64、67],自由曲线变形[54、56],三维变形[27、28、53、55]。

变形通常要解决两个关键问题:第一是建立初末两物体的元素(如顶点,边,角度等)之间的对应关系,称为对应问题(Correspondence problem);第二是通过插值初末两物体的对应元素产生中间状态,称为插值问题(interpolation problem)。目前国内外已有许多有关变形的研究成果,给出了解决变形问题的比较有效的算法,但遗憾的是,变形是一种视觉效果,与人的审美标准密切相关,主观因素很大,因此对于变形的两个问题什么是成功地解决方案,还没有正式明确的定义,但研究者普遍认为一个令人满意的变形应该满足以下三个条件:

1. 变形过程中产生的中间状态的一些特征,如边长、夹角、面积等应保持单调平滑的变换。
2. 变形过程中产生的中间状态没有出现自交、收缩、内部区域发生扭曲等不自然现象。
3. 中间状态要保持初始状态和目标状态的视觉特征。

目前已经有许多的变形算法,其类型及其复杂性在很大程度上取决于物体的表示

方法和相应的数据结构，因此不同的表示方法所采用的变形算法也不尽相同，大致分为以下三类：

1. 基于体元表示

物体的体元表示即一个三维物体表示成定义在整个 3D 空间上的函数的水平集，假设物体表示为点集 p ，使得 $f(p)=C$ ，则 $f(p)=C$ 即为该物体的体元表示。

2. 基于正视图 (elevation map) 表示

该表示方法在地貌造型和图像变形中有着重要的应用。例如 2D 图像可以看作是在像素格上由正视图 (elevation map) 定义的颜色映射。

3. 基于边界表示

物体的边界表示是指物体由多边形 (多面体) 和参数曲线 (曲面) 如样条曲线 (曲面) 这两个主要模型来构成其边界的表示方法，有相当多的物体是以其边界表示的。

1.2 国内外研究综述

目前国内外研究者通常主要研究变形的某一个具体问题：或主要研究变形的对应问题，其插值问题则用简单的顶点线性插值法来解决；或假设对应关系已经给定的前提条件下对插值问题进行研究，也有一些研究者同时研究这两个问题。其中对应问题的研究相对比较少，更多研究成果则是插值问题方面的。

1.2.1 对应问题

在变形对应问题的研究中，每一种算法都各有其适用范围。在文献[58]中 Thomas W.Sederberg 基于一种物理模型，把平面多边形的边看作为金属丝，通过使得初始多边形经过弯曲或者伸缩变为目标多边形时所做的功最小来建立初末多边形顶点的对应关系，该方法适用于两个相似的平面多边形；Kanai 在其文献[55]中利用 Harmonic 映照使得初始和目标多面体网格与相应的平面网格相对应，并将这两个平面网格重叠进而建立初末多面体的对应关系，这种方法适用于两个同胚于 3D 球或 2D 圆盘的可以不相似的多面体；文献[5]中 Gregory 用特征点把多面体网格对应分片，通过对分片的对应，建立两多面体的对应关系，该方法适用于两个

不相似的同胚的简单（或非简单）多面体；在文献[15]中 Decarlo 利用初末表面上的稀疏控制网格建立亏格不同的曲面之间的对应关系，该方法适用于拓扑结构不同的曲面。关于对应问题，将在第二章详细的介绍。

1.2.2 插值问题

对变形插值问题的研究常常是在假设对应关系已经确定的前提条件下进行的。在文献[25]中 Hermann Birkholz 和[52]中 Beier 提出的图像变形的的方法，都是基于一种影响场，通过在初末图像中使用若干条对应的辅助线来完成图像的对应和插值，从而实现图像变形。这是一种较成功的图像变形方法，但在有些情况下会出现病态，即某些时刻的中间图像产生重叠、相交。在解决多边形、多面体、曲线和曲面的插值问题的方法中，一个最简单的方法是顶点线性插值法，这种方法生成速度快，但变形中可能引起边界的非正常萎缩，产生自交，一些简单的几何特征如长度、角度等不能一致的变化，原因是各个顶点在变形过程中缺乏联系，其路径是相互独立的。在文献[1]中 Kaul 和 Rirossignac 同时研究变形的对应问题和插值问题，提出了基于 Minkowski sum 的算法，使得这两个问题同时得到解决，但该方法很容易混淆物体的相似部分，使得变形效果不理想。1993 年 Sederberg 和王国瑾等在[57]中提出了关于平面多边形变形的内在解算法，1998 年建立了用于空间多边形或多面体变形的的新算法 MSI（见[1]）。该方法首先找出初末多边形或多面体的顶点在局部直角坐标中的极坐标或者球面坐标，即内在解，通过线性插值其内在变量重构出中间状态。这种方法能够体现多边形或者多面体的内在结构特征，且生成速度快，取得了较好的变形效果，但是由于边界的各个部分之间缺乏联系，没有明确地考虑到多边形或多面体的内部，所以许多情况下边界容易自交，内部面积（体积）在变形中易产生扭曲。为了克服内在解算法中出现的容易自交的问题，Shapira 和 Rappoport[42]利用多边形的星骨架表示方法提出了一种新的插值方法，该方法首先构造初末多边形的同构的星骨架，然后线性插值相应的元素如边长、角度等得出中间多边形。由于它即考虑到了多边形的边界，也考虑到了多边形的内部，故减少了变形过程中产生自交的可能性。

Alexa[36]将初末物体分解成同构的单纯复形，在两个对应单纯形之间定义仿射变换，并将其表示成旋转变换和伸缩变换的复合，将这两种变换线性插值得到

期望的仿射变换，然后通过使得中间物体在最小二乘的意义下逼近该期望的仿射变换，从而达到扭曲最小的效果。该方法考虑了物体的内部，且变形具有刚性，因而变形过程比较自然，且不易产生自交，但是该方法计算量比较大。Floater 和 Gotsman[19]基于平面三角网格的内顶点的凸组合表示方法，提出了具有相同凸边界的同构三角网格的变形方法，称为凸组合变形，该方法能够使得中间三角网格始终与初末网格同构，不产生自交，且变形过程连续光滑，这给以后可保证不自交的变形算法的研究打下了基础。受[19]方法的启发，Gotsman 和 Surazhsky[60]给出了可避免自交的平面简单多边形的变形方法，它将初末多边形分别嵌入到两个具有相同凸边界的同构平面三角网格中，通过采用文献[19]中的方法对这两个三角网格进行变形，以此实现多边形的变形。该方法首次彻底的解决了平面简单多边形的变形中产生自交的问题，而且也是到目前为止唯一能够保证变形的中间状态不自交的变形方法。并且应用于图像变形，也可以保证中间图像不产生折叠。

Floater 和 Gotsman[19]的凸组合变形方法能够保证初末多边形在变形过程中不产生自交，且变形过程连续光滑。但是在进行变形前，先要对初末多边形进行同构三角剖分，这是一项十分复杂和艰苦的工作。对于同构三角网格的研究将在第三章详细介绍。对插值问题的研究将在第四章详细介绍。

1.3 论文研究内容及结果

在国内外二维图形变形的基础上，本文将在假设初末物体的对应关系已经确立的前提条件下对变形的插值问题进行研究，本文的若干研究结果，有的综合了已有算法的优点，在原有算法的基础上进行了改进，使得原有的算法更加完善。有的结果则是针对现有算法的不足与缺陷，提出了更好的算法，解决了原有算法的缺陷，得到了更加令人满意的结果。本文的研究内容及研究结果如下：

1) 平面多边形的相似同构三角剖分，提出了基于初始多边形和目标多边形的相似性的同构三角剖分。该方法将初始多边形和目标多边形的相似性考虑进来，首先在不添加额外顶点的情况下三角剖分掉初始和目标多边形的相似部分，能够达到对初始和目标多边形简化的效果，在此简化的初始和目标多边形基础上进行同构三角剖分，能够达到简化原始的初始和目标多边形之间的同构三角剖分的效果。应用该方法能够减少进行同构三角剖分增加的额外顶点个数，从而减少了多

边形变形的复杂度。

2) 基于小波的多边形变形算法, 提出了平面简单多边形的简单有效的变形算法。该方法结合了小波算法和多边形的同构三角剖分变形算法, 通过实践证明了运用该方法能够减轻同构三角剖分的难度, 以及减少了进行变形的计算量, 从而达到实时的效果, 而且变形也可以取得令人满意的变形效果。

3) 平面凸网格的保凸变形研究, 提出了具有不同凸边界的同构平面三角网格的保凸变形方法。本文提出了基于角度的保凸变形算法变形网格的凸边界, 并给出了该算法保凸的理论证明, 且具有凸边界内角按同一速度变化的性质, 对网格的内部顶点采用凸组方法。本文方法能够保证网格边界在变形过程中始终保持凸性, 且任意时刻的中间网格与初末网格同构, 即不产生自交现象。

第二章 对应问题

初末物体的对应问题是进行变形的基础，也是影响变形质量的一个关键环节。初末物体的对应问题已经有着广泛的研究[4、9、16、24、30、35、48、55、58]，而且取得了较好的成果，但是每一种对应方法都有其自身的应用范围。对应问题的研究还有更加光明的研究前景，通过对应问题的研究及实现，可以减少应用变形时所需要的手工劳动。

本章的目的在于寻找图形上对应的顶点位置。给定两个初末物体，如果仅仅需要建立的对应关系是物体的边界点对应关系，可以通过使得从初始物体变形到目标物体时所做的功最小或者采用加权的形式等方法决定初末物体之间的顶点对应关系。如果同时需要建立内部顶点的对应关系，可以采用嵌入的方法确定两者的对应关系，现在平图的变形技术不但要求对初末物体的外部顶点，而且需要变形图形的内部性质（比如边长，面积，颜色等），所以对初末物体的内部变形也是至关重要的。则本章重点作网格顶点的对应介绍。

给定两个网格 M_0 和 M_1 ，这一过程的结果是一个重心坐标的集合 B_0 ，使得 M_1 上这些重心坐标的几何体 $W_0 \in \phi_{v_1}(B_0)$ 是 M_0 在 M_1 上的一个嵌入 ϕ_{w_0} ，反之亦然。想法就是将一个网格的顶点映射到另一个网格上，以完成 M_0 和 M_1 之间双射的主要部分，经过这一步，仅有边和面需要做相应的调整。

这一步骤的典型做法是为曲面寻找一个公共的参数域 D 。通过将每一个曲面可逆的映射到参数域上，图形间的映射就被建立起来了。在变形的情况下，网格的典型参数域是球面 S^2 （在网格是一拓扑球面的情况下）或者是一个由分段线性参数域描述的拓扑圆盘的集合。在圆盘情况下，网格必须分解成圆盘的同构结构（这要求原网格是同构的）。一个主要的约束的要考虑用户指定或自动生成的特征的对应（如点-点对应）。依赖于所选的方法，决定是通过重新参数化还是按照特征对应分解网格来完成这一步骤。

在映射到球面的情况下，需要计算嵌入 ϕ_S ， $S = \{s_0, s_1, \dots\}, s_i \in R^3, |s_i| = 1$ 。球面上的嵌入由一个球面到球面的双射 f 按照特征对应对齐。

$$\begin{array}{ccc}
 \{i\} \in K_0 & \xrightarrow{w_0} & \phi_1(B_0) \\
 \phi_{S_0} \downarrow & & \uparrow \phi_{S_1}^{-1} \\
 S^2 & \xrightarrow{f} & S^2
 \end{array}$$

这个方法里的主要问题是计算球面上的顶点坐标 S_0 , S_1 以及重新参数化映射 f 。

分解方法是更普遍也更困难的方法。除了要产生拓扑圆盘的嵌入外, 还必须在考虑可能的特征对应的情况下, 将网格按同构方式分解。形式上, 要用一个包含 K_0 和 K_1 子集的抽象单纯复形 L 来粗略的逼近两个网格: $\phi_{V(0)}(|L|) \approx \phi_{V(0)}(|K_0|)$ 且 $\phi_{V(1)}(|L|) \approx \phi_{V(1)}(|K_1|)$ 。典型的, L 是 K_0 和 K_1 的子式, 如它是网格的一个划分 (partition)。 K_0 和 K_1 的顶点视为 L 中的某一个面, 并且所有属于一个特殊面的顶点嵌入到它的平面图形中。这需要将网格的每一片嵌入到平面上。

$$\begin{array}{ccc}
 \{i\} \in K_0 & \xrightarrow{w_0} & \phi_1(B_0) \\
 \phi_{L_0} \downarrow & & \uparrow \phi_{L_1}^{-1} \\
 |L| & \xrightarrow{f} & |L|
 \end{array}$$

下面就平面图形的顶点对应方法以及将平面上单连通的有界或无界网格嵌入到球面上的技术作介绍。

2.1 图形边界顶点对应

2.1.1 基于最小做功的顶点对应

Sederberg 等人在受到用力拉伸一根金属丝, 该力做功的启发下, 在文献[58]中提出了基于物理模型的二维多边形顶点对应算法。物体的变形主要由伸缩变换、旋转变换和平移变换构成, 做相应的变换都会有相应的功。

对于伸缩变换有, 对一根长度为 L_0 的金属丝施加大小为 P 的一个外力, 则金

属丝会发生一定的形变，其拉伸长度为： $\delta = \frac{PL_0}{AE}$ 。其中 A 为金属丝的横截面的面积， E 为金属丝的拉伸系数，这是一个和金属材料有关的不变的常量（例如：铁

的参数 E 为 29000000psi）。在拉伸金属丝的过程中，外力 P 所作的功为：

$$W = \frac{\delta^2 AE}{2L_0} \quad (2.1)$$

对于一根固定的金属丝，因为 AE 是一个不变的衡量，用一个用户自定义的“固体拉伸常量” k_s 替换式中的 AE 。如果 L_0 是一根金属丝的初始长度， L_1 是该金属丝的最终长度，当金属丝的初末状态交换的话，公式（2.1）将得到不同的值（在初始状态将得到 $\frac{\delta^2 AE}{2L_0}$ ，而在最终状态将得到 $\frac{\delta^2 AE}{2L_1}$ ）。当多边形的一条边缩短成为

一个点时（即： $L_0=0$ ），上面的公式将计算得到一个无穷值。对公式（2.1）改进得到：

$$W_s = k_s \frac{(L_1 - L_0)^2}{(1 - c_s) \min(L_0, L_1) + c_s \max(L_0, L_1)} \quad (2.2)$$

其中 $\delta = L_1 - L_0$ ， c_s 是用户定义的一个常量，通过该常量，用户可以处理当一条边缩短成一个点的情况。公式（2.2）中的指数 2 只有在金属丝为线性弹性的情况下才能适用，即金属丝的伸缩变换不是太剧烈的情况下。如果伸缩变换太剧烈，则实际所用的功比通过公式计算所得到的结果要小，那是因为在这种情况下，金属丝已经发生了弹性形变，用指数 1 替代公式中的指数 2 更能描述拉伸或者压缩金属丝所作的功。因此，对计算拉伸或者压缩金属丝所作的功的公式（2.2）做最终

的改变如下： $W_s = k_s \frac{|L_1 - L_0|^{e_s}}{(1 - c_s) \min(L_0, L_1) + c_s \max(L_0, L_1)}$ ，其中 k_s ， c_s 和 e_s 是

用户定义的常量。

在现实中的物理模型中，上面的公式只有在处理金属丝被拉长的情况下（即： $L_1 > L_0$ 的情况下）才有意义，处理被压缩（ $L_1 < L_0$ ）的情况下没实际意义。但是，

为了变形的问题，必须处理拉伸和压缩两种情况。因此设 $L_0 = \|P_{i_1} - P_{i_0}\|$ ，

$L_1 = \|P_{j_1} - P_{j_0}\|$ ，定义 $W_s([i_0, j_0], [i_1, j_1])$ 如下：

$$W_s([i_0, j_0], [i_1, j_1]) = \frac{k_s |L_1 - L_0|^{e_s}}{(1 - c_s) \min(L_0, L_1) + c_s \max(L_0, L_1)} \quad (2.3)$$

拉伸就相当于将 $P_{i_0} - P_{j_0}$ 映射到 $P_{i_1} - P_{j_1}$ ，其中 $i_1 = i_0$ 或者 $i_1 = i_0 + 1$ ， $j_1 = j_0$ 或者 $j_1 = j_0 + 1$ 。

对于旋转变换，需要解决从初始多边形的一个角度变换成目标多边形相应角度问题。对于从角 $\angle(P_{i_0}^0, P_{i_1}^0, P_{i_2}^0)$ 变换到角 $\angle(P_{j_0}^1, P_{j_1}^1, P_{j_2}^1)$ ，定义其功为：

$$W_b([i_0, j_0], [i_1, j_1], [i_2, j_2]) = \begin{cases} k_b (\Delta\theta + m_b \Delta\theta^*)^{e_b} & \text{如果 } \forall t \in [0, 1], \theta(t) \neq 0 \\ k_b (\Delta\theta + m_b \Delta\theta^*)^{e_b} + p_b & \text{如果 } \exists t_0 \in [0, 1], \theta(t_0) = 0 \end{cases} \quad (2.4)$$

其中 $\Delta\theta$ 和 $\Delta\theta^*$ 在文献[58]的 2.1 节与 2.2 节给出了定义， $\theta(t)$ 是从初始角度变换到目标角度的中间角度， k_b ， m_b ， e_b 和 p_b 是用户定义的常数变量，常量 k_b 是表示物体弯曲的难度系数， m_b 是补偿系数，适合于当角度变换不是单调变换， e_b 是一个和 e_s 扮演相同角色的指数， p_b 是处理当 $\theta(t)$ 经过零的补偿常量。

顶点之间的对应关系通过上面的伸缩变换所做的功 (2.3) 和旋转变换所做的功 (2.4) 之和最小来决定的，当对公式中的常数变量设定不同的值时，所得到的对应关系很可能也不一样。该方法适用于两个相似的平面多边形。

2.1.2 基于最小成本的顶点对应

Henry Johan 等人在文献[24]中提出了一种顶点对应算法，该算法延伸了 Sederberg 等人在文献[58]中提出的最小能量算法和 S.Cohen 等人在文献[48]中提出的顶点对应算法。

Henry 在 Sederberg 的思想提出了一种新的成本算法。在该算法里，首先定义了顶点处的切线方向 $\vec{P}_i^* = \frac{P_{i+1}^* - P_{i-1}^*}{\|P_{i+1}^* - P_{i-1}^*\|}$ (其中 * 代表初始或者目标)，其中各变量如图 2-1 所示，并且定义算子如下：

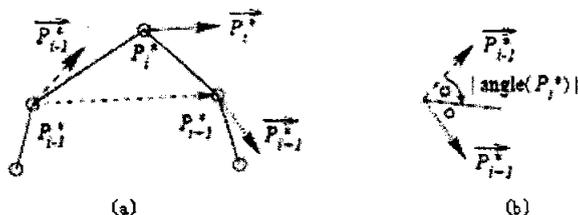


图 2-1 (a)顶点处的切向量定义与(b)角度成本计算

$$\text{sign}(x) = \begin{cases} 1.0 & x \geq 0.0 \\ -1.0 & x < 0.0 \end{cases} \quad Z(\vec{A}, \vec{B}) = A_x B_y - A_y B_x$$

从而得到在顶点 P_i^s 处的角度成本为:

$$\text{angle}(P_i^s) = \frac{1}{2} \arccos(\overline{P_{i-1}^s}, \overline{P_{i+1}^s}) \text{sign}(Z(\overline{P_{i+1}^s}, \overline{P_{i-1}^s}))$$

顶点 P_i^s 处的参数简单的定义为 $\frac{i}{n}$ (n 为初始多边形的顶点个数), 同理可定义 P_j^r 处的参数为 $\frac{j}{m}$ (m 为目标多边形的顶点个数), 则通过角度成本和参数成本表示的初始多边形的第 i 个顶点与目标多边形的第 j 个顶点对应的成本函数为:

$$\text{cost}(i, j) = \omega_1 |\text{angle}(P_i^s) - \text{angle}(P_j^r)| + \omega_2 \left| \frac{i}{n} - \frac{j}{m} \right|$$

其中 ω_1 与 ω_2 为相应的权值系数, 则顶点之间的对应关系可以通过使得

$$\sum_{i=1}^n \text{cost}(i, J(i)) \text{ 最小来求得, 即求解 } \min \sum_{i=1}^n \text{cost}(i, J(i)).$$

2.1.3 模糊的顶点对应

Xiaoyi Jiang 等人在文献[65]中提出了新的变形算法, 其具体内容将在第四章作详细介绍。在该文中, 作者先将初末图形用向量串表示, 然后采用[45]中从初始向量串变形到目标向量串所定义的编辑算子, 并且采用该文中计算两向量串距离的算法计算用向量串表示的初末两幅图形的距离, 在计算初末图形距离的同时, 初末向量串中各元素之间的对应关系也就确定了下来。

2.1.4 结论与比较

平面图形边界顶点的对应算法的最大不同点在于定义了不同的成本计算方法。在定义了成本计算方法之后，通过计算顶点之间的最小成本来确定顶点之间的对应关系。该类顶点对应方法在初末图形是相似的情况下，效果比较明显，一旦初末图形有较大的差异，则会导致变形过程出现自交等不良现象。

2.2 基于网格的顶点对应方法

三维图形边界的单连通 (simply-connected) 部分同构于一圆盘，因此称为拓扑圆盘。为了寻找这些单连通片的一个参数化，需要定义一个从有界单连通网格到平面的一个双射。

在实际的应用中，需要寻找片与片之间的双射。这里关注从任意有界单连通网格到单位圆盘上的映射，使得网格边界顶点位于单位圆上。这里仅局限于几种参数化方法，这些方法允许三角化曲面的边界是非凸的或者没有固定是优先获得更光滑的还是更少扭曲的边界映射[11、21、22]。

第一步，将边界顶点固定在单位圆上。首先，从基域 L 取三个点，按等角方式固定在单位圆上，这是取保基域中相邻的面在经过基域边是有一个连续的参数化所必要的。剩余的边界顶点按弧长与原边长成比例的方式固定。剩余的内部顶点是自由的，它们的位置由其与相邻点的关系确定。

大部分已发表的方法是令一个由该顶点相对于其邻点的关系表示的二次误差函数最小来确定内部顶点。这样内部顶点的确定可以简化为一个线性方程组的求解问题。非线性的方法或者是使一高次函数最小化或者利用算法本身的性质 (如 Grego 等[4、5]，用户指定一对对应顶点，通过最短路径确定内顶点的对应关系)。

更明确的说，令 $\{v_i\}$ 是要映射到圆盘上的顶点，其中自由的内部顶点的下标为 $0 \leq i < n$ ，固定的边界点下标为 $n \leq i < N$ ，下面的目标是要寻找平面上点的位置 w_i ， $|w_i| = 1, n \leq i < N$ 。当且仅当没有边相交时这个映射是个双射，或者说

$$\forall (i, j, k) \in K, \det \begin{pmatrix} w_x & w_y & 1 \\ w_{jx} & w_{jy} & 1 \\ w_{kx} & w_{ky} & 1 \end{pmatrix} > 0 \quad (2.5)$$

然而，如果将这个二次表达式作为保证有效的平面嵌入的准则，则是不方便的，这也是为什么多数方法转而更严格但却更有效的线性条件的原因。

接下来，将讨论三种定义线性系统的方法，这些线性系统的解将产生顶点的位置，另外还将说明 Gregory 等[4、5]的 Divid&Conquer 方法。

2.2.1 重心映射

Tutte[66]已经证明了怎样利用重心映射将平面图映射到平面上。在约束假设中，思想很简单，就是将每一个顶点置于它的相邻点的中心：

$$w_i = \sum_{j \in N(i)} \frac{1}{d_i} w_j \quad (2.6)$$

$$\text{设 } \Lambda = \{\lambda_{i,j}\} \quad \lambda_{i,j} = \begin{cases} d_i^{-1} & \{i, j\} \in K \\ 0 & \{i, j\} \notin K \end{cases} \quad (2.7)$$

则(2.6)式可以写成线性方程组的形式：

$$(I - \Lambda) \begin{pmatrix} w_0 \\ w_1 \\ \dots \\ w_{n-1} \end{pmatrix} = \begin{pmatrix} \sum_{i=n}^{N-1} \lambda_{0,i} w_i \\ \sum_{i=n}^{N-1} \lambda_{1,i} w_i \\ \dots \\ \sum_{i=n}^{N-1} \lambda_{n-1,i} w_i \end{pmatrix} \quad (2.8)$$

矩阵 $(I - \Lambda)$ 是满秩的，这样方程组(2.8)就有唯一解。Tutte[66]已经证明这个解是网格的一个有效嵌入。

注意到，网格的形状并不影响顶点在平面上的位置，嵌入中的所有信息均来自于 K ，显然这样的嵌入并不能反映网格的顶点位置 V 所蕴涵的几何性质。下面试着混合原图形的这些信息。

2.2.2 保形参数化

在重心映射中, 权系数 λ 仅包含拓扑信息。Floater[39]定义了反映网格局部形状的权系数, 更准确的说, 就是权系数 λ 的选择考虑了某顶点周围的角度以及边的长度。

为了计算某特定点 v_i 的权值, 这个点先被置于原点, 与它相连的边按原来的长度置于平面上, 相应的角度则与原图形的角度成比例, 这被假定为网格关于 v_i 最理想的参数化 w_i' 。

权值按以下方式进行: 即如果相邻点 w_j 已固定, 那么所求的权值将使 w_i 置于原点, 并且必须求解线性方程组。这样有

$$w_i = \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \sum_{j \in N(i)} \lambda_{i,j} w_j \quad (2.9)$$

$$\text{且} \quad \sum_{j \in N(i)} \lambda_{i,j} = 1 \quad (2.10)$$

如果 v_i 仅有三个相邻点, 则上式将唯一确定一组权值; 如果 v_i 有三个以上的相邻点, 那么正的权值就必须从可能的解空间里选择。权值的正性导致了凸组合, 这是确保一个有效嵌入的必要条件。Floater 给出了一个计算正权值的方法: 取循环有序的相邻点集合 $j_k \in N(i), k \in \mathbb{Z}_{N(i)}$, 对每一个 k , 产生一组非负权值, 平均这些权值以产生最终的权值:

$$\lambda_{i,j} = \frac{1}{|N(i)|} \sum_k \lambda_{i,j}(k) \quad (2.11)$$

有关正权值的计算将在后面章节做详细的说明。确定权值后, w_i 的位置由方程组

(2.8) 给出。最后 Floater[40、41]证明了 Tutte 定理的一个一般化的结论, 其中表明每一个顶点都表示为其相邻点的凸组合是产生有效嵌入的充分条件。

2.2.3 离散调和映射

调和映射 (harmonic mapping) 是几个应用微分的数学领域中的一个概念。调

和映射经常被描述为在所有映射到给定区域 Ω 的函数中,使 Dirichlet 能量

$$E_D(u) = \frac{1}{2} \int_{\Omega} |\nabla u|^2 \quad (2.12)$$

最小的函数 u 。Pinkall 和 polthier[59]说明了如何在三角形上离散这个问题,使得每一点的权值都可以导出,并且可以得出一个形如方程组(2.8)的线性方程组。更详细的推导可参看 polthier[33],在这篇文章里,离散的 Dirichlet 能量用下式表示:

$$E_D(u) = \frac{1}{4} \sum_{i,j \in K} (\cot \alpha_{i,j} + \cot \beta_{i,j}) |v_i - v_j|^2 \quad (2.13)$$

$$\alpha_{i,j} = \angle(i, k_0, j), \beta_{i,j} = \angle(i, k_1, j), \{i, j, k_c\} \in K$$

每一点的最小解为

$$0 = \frac{1}{2} \sum_{j \in N(i)} (\cot \alpha_{i,j} + \cot \beta_{i,j}) (v_i - v_j) \quad (2.14)$$

这样可求得权值

$$\lambda_{i,j} = \begin{cases} \frac{\cot \alpha_{i,j} + \cot \beta_{i,j}}{\sum_{j \in N(i)} (\cot \alpha_{i,j} + \cot \beta_{i,j})} & \{i, j\} \in K \\ 0 & \{i, j\} \notin K \end{cases} \quad (2.15)$$

这些权值代入式(2.8)中可求得一个嵌入。

2.2.4 保面积的 Divid&conquer 映射

Gregory 等[4、5]描述了一个递归算法,其目的是在映射中保持三角形的面积。基本思想就是促使映射递归的将片(patch)分割两部分,每一部分再独立的映射。这些分割的选择使得原网格与嵌入中的面积比大致相同。

特别的,选择直径上的两个顶点。最短路径按 Dijkstra 的算法计算。这条路径映射成连接两顶点的线段。然后改变路径使得在嵌入中与在三角曲面上的面积比差异最小。这些线段将每一片网格分成两半,每一半再按同样的方式进行,直到映射完所有的顶点。

2.2.5 比较与结论

应用上述四种方法,已经将网格的一些部分作了嵌入。注意到矩阵方程(2.8)的解可以利用分级(hierarchical)技术来执行,这等同于利用多栅方法。然而,这

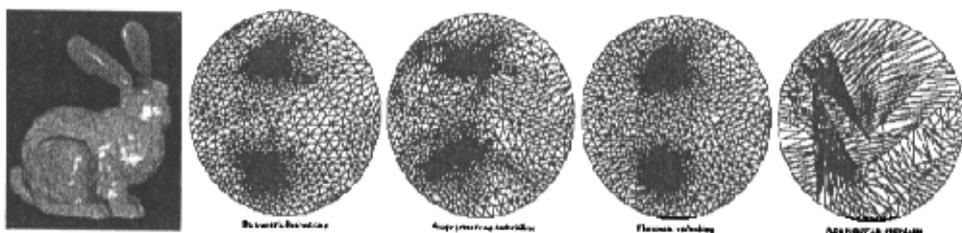


图 2-2 用不同的映射技术将网格的一部分在单位圆盘上做的参数化。原来的图形是图中红色的部分。重心映射不能反映网格的几何性质（见 2.2.1 节）。保形嵌入试图通过局部逼近正投影映射来抓住网格的局部形状（见 2.2.2 节）。离散调和嵌入将矩阵扭曲最小化（见 2.2.3 节）。保面积嵌入是一个递归过程，它的目标是逼近原三角形的面积（见 2.2.4 节）。

个矩阵是稀疏的，可以发现利用它的稀疏性，应用迭代解法也是很有有效的。

图 2-2 说明了这几种方法比较的一些结果。很明显，在所有用线性优化技术生成的嵌入中，较大和较小三角形的一般结构是类似的。这也暗示了连通性是这一类嵌入的主要影响因素。改变用来计算嵌入的权值仅仅改变嵌入的局部行为。它们都具有面积压缩的问题：内部三角形比外部三角形的面积小的多。面积保持算法消除了这个问题，但是它却扭曲了三角形图形。

实际上，由于面积或角度的高比率和浮点数的精度限制，所有这些方法可能都不实用。研究发现，为了获得一个合理的网格参数化，基域必须足够的“外壳 (skin)”。

在局部图形上的细小差异对图形对应的结果似乎并没有多大的影响，尤其是当局部特征通过重新参数化对齐的时候（见 2.2.4 节）。

2.3 参数化拓扑球面

无边界的单连通 2-流形称为拓扑球面，因为它们与球面是同胚的。因此，对这种图形一个自然的参数域就是单位球面。

2.3.1 星图 (star shape)

Kent 等[29、30]首次提出了将某一类零亏格网格映射到球面上的技术。特别简单的一类物体就是凸图形。一个凸图形有以下性质：连接任意两个边界顶点的直

线都完全落在图形内部。这样，所有的点对图形的任意内部点都是可见的，并且从一个内部点到一个封闭球面的投影是个双射。

这一思想的一般化就是将凸图形推广到星图上。在星图中，至少与一个内部点，使得连接该点与边界点的直线完全落在图形内部。具有这一性质的内部点称为星点。显然，通过一个星点将边界点投影到一个封闭球面的映射是个双射。明确的说，如果对所有的网格点，点 O 是可见的，那么变换所有的点使得点 O 与原点重合，然后规范化所有点的坐标。这些坐标就是网格点在单位球面上的参数化。图 2-3 给出了一个说明。

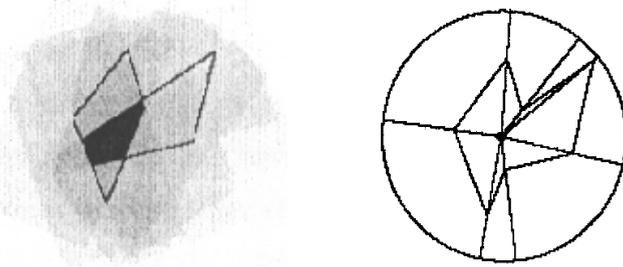


图 2-3 一个多边形星图以及它在圆上的投影。星图的核是所有边（在多面体情况下是面）上的开半空间的交集。每一个核里的点通过投影导出一个到圆上的双射

现在，唯一的问题就是确定一个图形是否为一个星图，如果是，寻找星点。对分段线性图形（网格）来说，可以通过求有网格面元导出的半空间的交来解决这个问题。所有半空间的交集称为核（kernel）。如果核是非空的，那么网格就是星图，并且每一个（凸）核内部的点都可以作为星点。应用标准的技术[9]，计算一个三维网格的核的复杂度为 $O(n \log n)$ 。

2.3.2 简化

Shapiro 和 Tal[7]首次提出了一个将任意零亏格多面体转化为凸图形的可靠算法。他们首先用顶点移除的方法将图形简化，直到简化的图形变成一个四面体为止。仅仅度数为 3, 4, 5 的顶点被移除。既然网格是三角网格，这样的顶点是恒存在的：由 Euler-Poincare 方程易得，任意的三角（曲面）网络的平均度数小于 6。这样，至少存在一个度数严格小于 6 的点。

一旦图形被简化成四面体，顶点被重附着在图形上，并确保图形保持凸性。更明确的，它说明了怎样重附着度数为 3, 4, 5 的顶点使得图形保持为凸的。也就是说，如果一个顶点 $\{i\}$ 必须添加到面 f 上，那么这个点的位置必须在现有网格凸包的外部，但是在 f 的相邻面的核的内部。

2.3.3 弹性嵌入

Alexa[35]将平面嵌入方法的一个变化引入到多面体在球面上的嵌入中去。其基本思想和重心映射是一样的：将每一个顶点置于其相邻点的中心。然而，在球面上，平面情况的两个条件就被破坏了：第一，相邻顶点位置的凸组合并不是参数域（球面）的一部分；第二，没有给定的外部圆环来支持嵌入。

这个方法是用一个松弛算法来计算重心约束的解。初始的外形是通过计算网格表示的实体模型的一个内部点，然后再将所有的点投影到一个以刚计算得到的内部点为中心的一个球面上来产生的。松弛算法重复的将每一个顶点置于它的相邻点的中心。既然这些中心不在球面上，将它们的坐标规范化：

$$w_i^{t+1} = \frac{\sum_{j \in N(i)} w_j^t}{\left\| \sum_{j \in N(i)} w_j^t \right\|} \quad (2.16)$$

这个方法的主要问题是缺少了固定顶点。在平面情况下，固定的顶点避免了所有的顶点塌陷成一个点，这个点是 (2.16) 的一个平凡解。在球面上，存在局部最小使得这些点分布在球面上，然而，松弛算法倾向于寻找全局最小值，也就是说所有顶点一致。

对这个问题，Gumhold[15]报告了一个优秀且简单的解决方案：在每一次松弛循环后使球面回到中心，也就是说

$$w_i^{t+1} = \frac{1}{n} \left(\sum_{j \in K} w_j^t \right) - w_i^t \quad (2.17)$$

如果要保证嵌入的拓扑正确性，任意类型的一个 ε 范围作为唯一的终止准则是不充分的，代替的，这一过程只有当找到一个有效的嵌入时才能结束。一个嵌入是有效的，当且仅当所有的面均有相同的方向。可以通过测试沿着每一个面的边界的三个连续的点的方向来检查这一条件。这里，方向指的是球面上的曲面的三个点是否是一个顺时针的顺序。这可通过计算 $\text{sgn}((v_0 \times v_1) \cdot v_2)$ 的值来确定。图 2-4

描述了一匹马的多面体模型的松弛过程，图 2-5 展示了几个模型的嵌入结果。

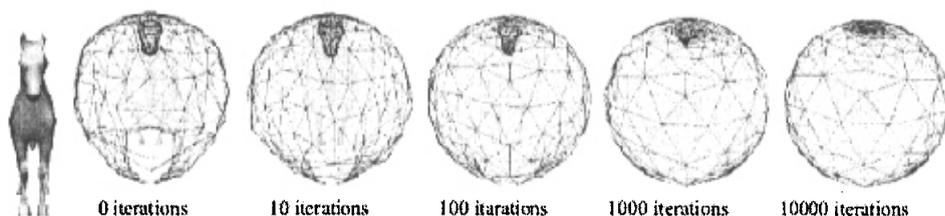


图 2-4 用松弛算法将一多面体物体嵌入到球面上。最初，所有顶点通过模型的一个内部点投影到单位球面上。当所有的面有正确的方向时，松弛过程结束。图中红色边围起来的面为方向不正确的面。

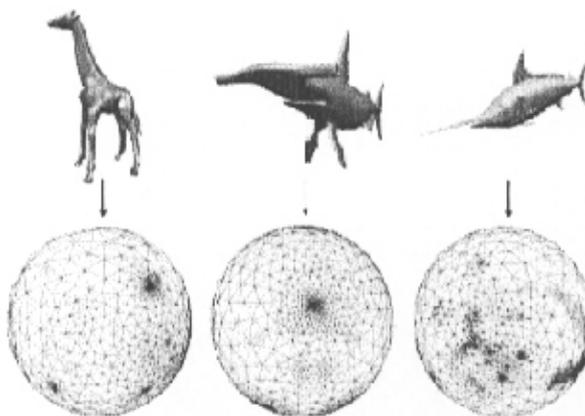


图 2-5 一只长颈鹿、一只锤头状的鲨鱼和一只旗鱼模型的球面嵌入

2.3.4 比较与结论

除了必要的约束（如 2.2.4 节所讨论）以使得嵌入是有用的，上面所讨论的方法没有一个特意的去尝试保持原网格的性质。中心投影也显然只局限于一小类物体。可以发现这两种针对一般零亏格网格的技术具有互补的特征：简化方法更稳定（关于几何计算和网格拓扑检测的敏感性），而松弛方法则生成更光滑的嵌入。这两种方法都有前面所提到的面积压缩的问题。

2.4 本章小结

在本章中介绍了顶点的对应算法，对各种算法作了一定程度的分析。对应算

法的好坏直接影响到了变形的效果，因此在做变形处理的时候，一定要选择好的对应算法。对应算法在一定程度上影响着初末多边形之间的同构三角剖分难度，对应算法重要，多边形之间的同构三角剖分同样也很重要，因此在下一章将介绍平面多边形的同构三角剖分。

第三章 平面多边形的相似同构三角剖分

同构三角网格在各个领域都有着十分重要的作用，有着广泛的应用范围，当一个好的同构三角网格应用于变形时，可以减少变形的计算量以及取得好的变形效果。在计算机图形学中，平面三角网格被用来作为曲面和平面形状的参数和表示方法，因而研究同构平面三角网格的变形方法，对于这一类物体的变形具有重要的意义。基于同构三角网格给出的变形方法有 Alexa[17]给出的方法，Floater 和 Gotsman 给出的凸组合变形方法，Gotsman 和 Surazhsky 给出的可控制的同构平面三角网格的变形方法[56]，同构三角网格的内在变形[64]等。相应的同构三角网格算法研究有[6、26、34、43、61、62]等。然而，应用于平面多边形变形的网格算法中，得到初末多边形同构三角网格所增加的额外顶点是大家所关心的问题，因为所添加的额外顶点个数越多，所需要的计算量也就越大，从而使得额外顶点个数成为衡量计算量大小的一个标准。本章在如何减少额外顶点方面做了进一步的研究，取得了比较好的效果。

3.1 平面同构三角网格预备知识

当两个平面三角网格的拓扑结构是等价的时候，则称这两个平面三角网格是同构的，或者是称为相容的。拓扑结构的引入是为了在一般的集合上建立映射连续的概念。下面介绍两平面三角网格同构的相关知识。

定义 3.1 (拓扑空间) 设 X 是一个非空集合，若 X 的一个子集族 τ 它满足：

- (1) $\{X, \phi\} \subset \tau$;
- (2) τ 中任意多个成员的并集仍在 τ 中;
- (3) τ 中两个成员的交集仍在 τ 中;

则称 τ 为 X 的一个拓扑，称 (X, τ) 为一个拓扑空间。

一个图 G 是由有 p 个顶点的非空有限集合 $V = V(G)$ 和预先给定由 V 中不同顶点的 q 个无序对构成的一个集合 $E = E(G)$ 组成，记作 $G = G(V, E)$ 。 E 中的每个顶点对 (u, v) 称为 G 的边，如果用 e 表示这条边，则记 $e = (u, v)$ 或简记为 $e = uv$ ，称 u ,

v 是边 e 的端点, 且称 u 和 v 是邻接的顶点. 简单图是指一张没有环边且边在边集 E 中不重复出现的图.

在图论中, 两个简单图 $G_0 = G_0(V_0, E_0)$ 和 $G_1 = G_1(V_1, E_1)$ 同构 (isomorphic), 是指 G_0 和 G_1 的顶点集合和边集合之间分别存在一一映射, 使得对应的边连接对应的顶点.

定义 3.2 图 $G = G(V, E)$ 称为平面图, 如果它满足如下的条件:

- (1) 存在一一映射 p 将顶点集 V 映射到平面上的一个互不重合的点集;
- (2) G 的每条边 $(i, j) \in E$ 对应于以 $p(i)$ 和 $p(j)$ 为端点的简单曲线;
- (3) 这些简单曲线除了在公共端点处的交点外, 无其它交点.

此时的图 G 称为平面图 (Planar graph), 其平面表示称为平面图 (Plane graph). 由以上条件可以看出, 一个平面图与其平面图是同构的, 且其平面图不唯一.

平面三角网格 (planar triangulation) $T = T(G, P)$, 是指其每个有界面仅含三条边且每条边均由直线段表示的简单平面图. 两个平面三角网格 $T_0 = T_0(G_0, P_0)$ 和 $T_1 = T_1(G_1, P_1)$ 是否同构 (isomorphic 或 compatible), 关键是看它们的拓扑结构是否是等价的, 其判断规则如下:

- (1) 图 G_0 和图 G_1 同构.
- (2) 平面三角网格 T_0 和 T_1 的有界面之间存在一一映射的关系, 并且两个相对应的有界面之间连接图 G_0 和图 G_1 中相对应的顶点.
- (3) 边界 ∂G_0 和 ∂G_1 具有相同的方向 (orientation), 即 ∂G_0 和 ∂G_1 存在相对应的顶点序列, 它们的顶点相对于其内部有界面或者是都呈现逆时针方向, 或者都呈现顺时针方向.
- (4) T_0 和 T_1 的所有互相对应的有界面都具有相同的方向.

其中 (3) 与 (4) 是等价的. 设有界面的有序顶点依次为 p_1, p_2, \dots, p_n , 其方向有逆时针和顺时针两种: 当面的方向是逆时针时, 是指顶点序列 p_1, p_2, \dots, p_n 是有界面的顶点通过有界面的质心按逆时针方向排列得到的;

当面的方向为顺时针时,是指 p_1, p_2, \dots, p_n 是有界面的顶点通过有界面的质心按顺时针方向排列得到的。图 3-1 给出了平面三角网格同构与不同构的实例。

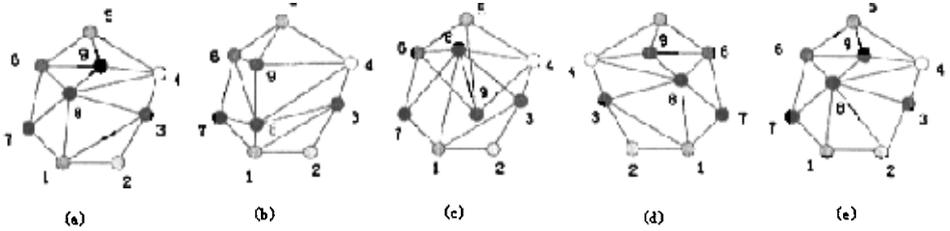


图 3-1 在上面给出的五个平面三角网格中, (b)与(a)是同构的, 其余网格均不与(a)同构, 网格(c)不满足平面图的条件, (d)与(a)不满足判断规则(3)(或者是(4)), (e)与(a)不满足判断规则(2)

给定两个平面三角网格, 检验其是否同构是比较简单的问题, 只要考察它是否符合上面的条件即可得到是否是同构的结果。但是, 对于给定的两组顶点, 要寻找其同构三角网格则比较困难的; 对于给定初末两个多边形, 要寻找其同构三角剖分, 同样不是一件容易的事情, 在不添加额外顶点的情况下, 同构三角剖分有时是不存在的。已经有研究者[47]证明出同构三角网格的存在性是一个 NP 问题, 有时只有添加一定数量的额外顶点才能够构造出同构三角网格剖分。

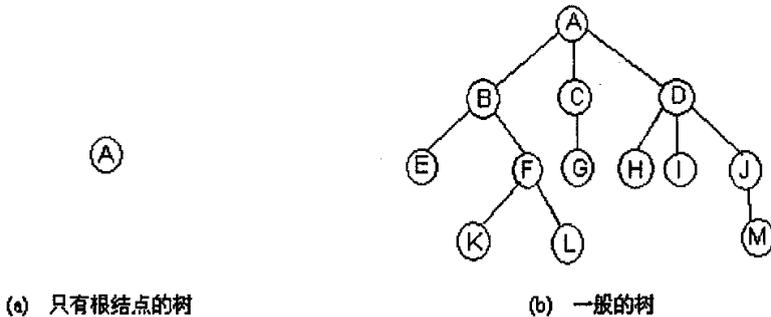


图 3-2 树的示例

定义 3.3 (树) 树 (Tree) 是 $n (n \geq 0)$ 个结点的有限集。

在任意一棵非空树中: (1) 有且仅有一个特定的称为根 (Root) 的结点; (2) 当 $n > 1$ 时, 其余结点可分为 $m (m > 0)$ 个互不相交的有限集 T_1, T_2, \dots, T_m , 其中每一个集合本身又是一棵树, 并且称为根的子树 (SubTree)。如图 3-2 所表示。

图 3-2 (b) 中, 子集{B, E, F, K, L}就是根 A 的子树。

定义 3.4 (叶) 结点拥有的子树数称为结点的度 (Degree), 度为零的结点称为叶。如图 3-2 (b) 中, E、K、L、G、H、I、M 都是树的叶结点。

定义 3.5 (对偶图) 设 G 是有 p 个顶点, q 条边和 f 个面的平面图, G 的 f 个面分别用 S_1, S_2, \dots, S_f 表示。在 G 的每个面中放置一个顶点, 令 S_i 中的顶点为 v_i 。如果两个面 S_i, S_j 相邻 (两个面的边界至少有一条公共边, 则称这两个面是相邻的), 则用边 (v_i, v_j) 连接 v_i 和 v_j , 使它与面 S_i, S_j 的公共边只相交一次 (此时称 (v_i, v_j) 与所相交的边成为对应边), 且与图的其它边界无公共点。这样得到的有 f 个顶点和 q 条边的图 G^* 称为 G 的对偶图。

定义 3.6 (对角线) 设 p_i, p_j 是多边形 P 的任意两个顶点, 如果 $\overline{p_i p_j}$ 位于多边形的内部, 则称 $\overline{p_i p_j}$ 为多边形 P 的一条对角线。

定义 3.7 (耳) 设 p_{i-1}, p_i, p_{i+1} 是多边形 P 的三个连续顶点, 并且 $\overline{p_{i-1} p_{i+1}}$ 是 P 的对角线, 那么 p_{i-1}, p_i, p_{i+1} 构成 P 的一个耳, p_i 称为耳尖。

3.2 平面多边形的同构三角剖分算法介绍

当给定两个都具有 n 条边的简单多边形时, 如果三角剖分的顶点仅仅局限于平面多边形所提供的顶点上, 不允许增加额外的顶点, 在这两个多边形之间很肯能不存在同构三角剖分; 当在两个多边形的内部, 额外顶点被允许使用时, 则在两个多边形之间是否存在同构三角剖分呢? 答案是肯定的。这是在 Boris Aronov 等人在文献[6]中给出的肯定回答, 在此文献中, 作者给出了两种同构三角剖分的算法, 并且这两种算法要使用 $O(n^2)$ 个额外顶点。

方法一: 给定都含有 n 个顶点的两个简单多边形 P_1 和 P_2 , 并且假设两多边形顶点之间的对应关系已经确定, 将 P_1 和 P_2 相应的顶点按逆时针方向或者顺时针方向从 1 到 n 进行编号。多边形 P 是一个含有 n 个顶点的任意凸多边形, 并且 P 的顶点有着相同方向的从 1 到 n 的编号。要对两个多边形进行同构三角剖分, 首先从两

个多边形 P_1 和 P_2 单独三角剖分开始, 在对 P_1 和 P_2 进行三角剖分时, 可以在不添加任意额外的顶点的情况下完成对两个多边形的三角剖分。假设对 P_1 三角剖分得到 T_1 , 对 P_2 进行三角剖分得到 T_2 , 将 T_1 和 T_2 分别映射到 P 上, 则得到多边形 P 的两个三角剖分 T_1' 与 T_2' 和线性同胚映射 $L_1: P \rightarrow P_1, L_2: P \rightarrow P_2$, L_1 将三角剖分 T_1' 的顶点, 角以及三角形分别映射到 T_1 上相应的顶点, 角及边上, L_2 将三角剖分 T_2' 的顶点, 角以及三角形分别映射到 T_2 上相应的顶点, 角及边上。将 P 的三角剖分 T_1' 与 T_2' 重叠在一起, 则得到 P 的一个凸细分, 在不引入额外顶点的情况下, 对 P 的凸细分进行三角剖分, 可以得到 P 的一个三角剖分 S , 该剖分拥有 $O(n^2)$ 个顶点。则 $S_1 = L_1(S)$ 和 $S_2 = L_2(S)$ 便分别为 P_1 和 P_2 的同构三角剖分。剖分实例如图 3-3 所示。

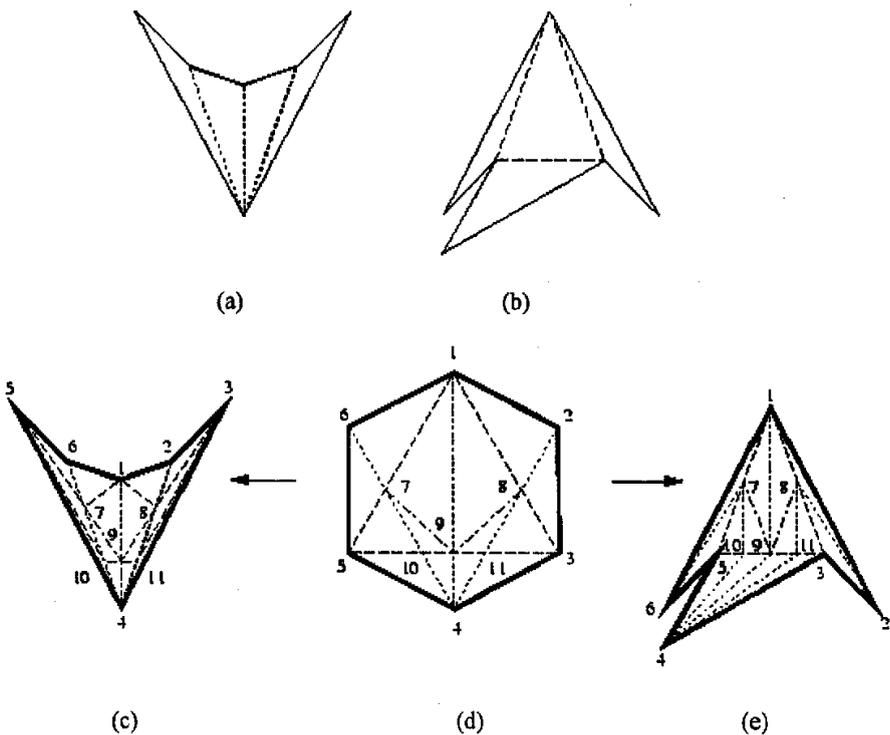


图 3-3 (a)与(b)是两个不存在同构三角剖分的六边形。(c)与(e)为由方法一通过利用凸多边形(d)构造得到的同构三角剖分

方法二：该方法是 Boris Aronov 等人在文献[6]中给出的另外一种多边形的通用 (A Universal Triangulation) 三角剖分方法：将多边形按照“蜘蛛网” (如图 3-4 (a)) 的图案进行三角剖分。由图可见，对于一个拥有 n 条边的多边形，其“蜘蛛网”结构是一个拥有 $n-2$ 层同心层多边形和一个公共的中心顶点，对于最里面一

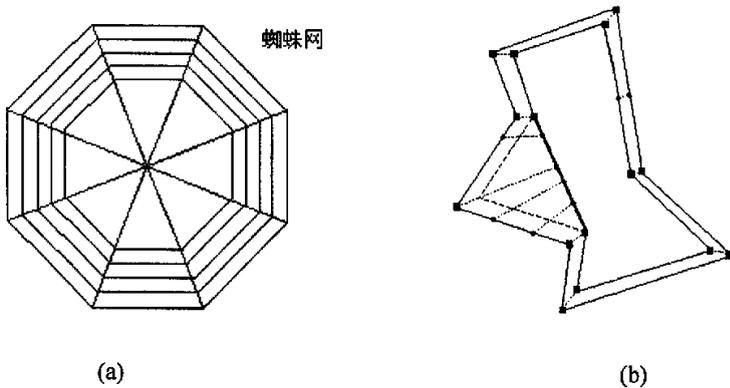


图 3-4 (a)具有 8 个角点的多边形“蜘蛛网”。(b)由 12 个顶点中含有 8 个角点的多边形产生一个新的里层的多边形，该多边形具有 12 个顶点，其中 7 个角点。

层多边形的所有顶点都与中心顶点相连接。对于同心层多边形之间，用线段连接相邻两层多边形对应的顶点。

对任意一个有 n 条边的多边形的“蜘蛛网”三角剖分可以通过如下步骤很容易得到：设多边形 P 是一个拥有 n 个顶点的多边形，其中 n 个顶点中有 m 个角点 (即：剩下的 $n-m$ 个顶点在多边形的边上)，在 P 的内部构造一个比 P 稍小且形状相同的多边形 P' ，这样便在 P 与 P' 之间形成了一条环形带，连接 P 与 P' 之间的对应顶点，则将该环形带分解成了 n 个凸四边形。在 P' 的所有角点中，必定包含有耳，设其中一个耳是 c ，将与 c 相关联的两边替换为一条弦 d ，并在弦 d 上增加一定数量的顶点得到多边形 P'' (如图 3-4 (b))，此时的 P'' 具有 n 个顶点和 $m-1$ 个角点。不断重复上面的步骤，对于拥有 n 个顶点，并且所有顶点均为角点的情况下，处理整个过程只需要 $n-3$ 次迭代便可使得最里面一层的多边形仅仅含有 3 个角点。此时将“蜘蛛网”的中心顶点与多边形的所有顶点相连接，并将前面所产生的凸四边形变成三角形，便得到了 P 的三角剖分。在整个构造过程中，要用到 $n(n-3)+1$ 个额外顶点。但是事实上，当多边形含有两个不重叠的耳时，每次迭代可以同时去掉

两个角点，而且每个含有 5 个角点的多边形都是星形多边形（这就意味着迭代至 5 个顶点就可以了，不需要迭代至 3 个顶点），故迭代次数可以减少，这样所需要的额外顶点就可以更少一些。

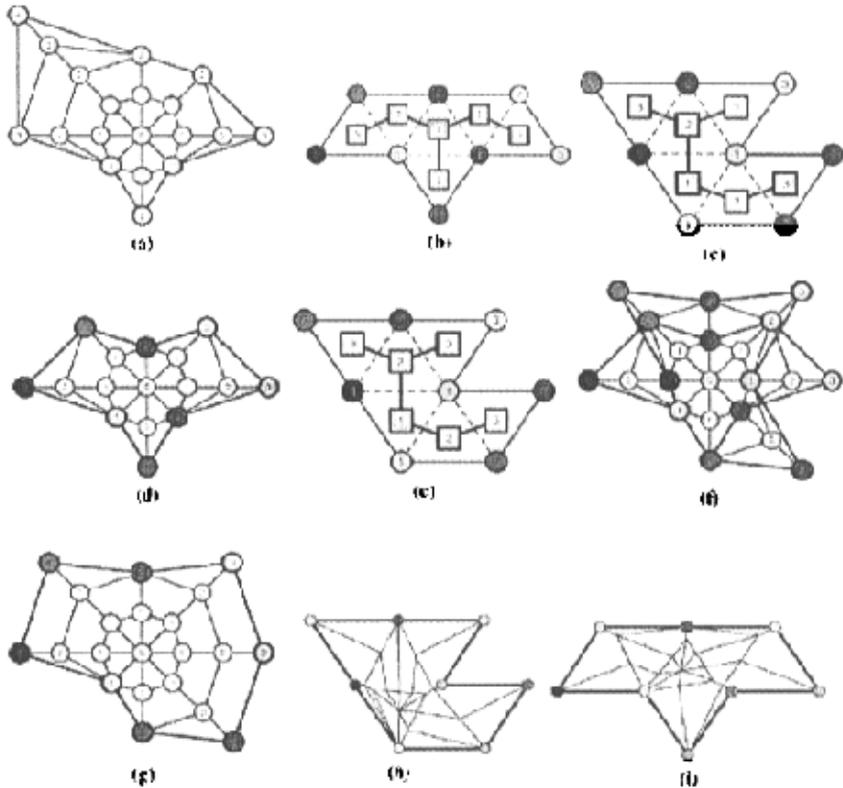


图 3-5 采用方法三构造两个简单多边形的同构三角剖分。(a) 序列为 $\langle 1, 2, 1, 3, 2, 2, 4, 3 \rangle$ 的“蜘蛛网”连结图。(b), (c)给出了两个多边形的三角剖分、标有深度的对偶图以及顶点的里程碑。其中对偶图结点的深度值、顶点的里程碑以及两个多边形顶点的对应关系分别用方框中的数字、圆圈中的数字和顶点旁边的数字来表示。(d), (e)分别为对应于(b), (c)里程碑序列的“蜘蛛网”连结图。(f) 融合后的具有最大深度值的“蜘蛛网”连结图。(g) 具有和(f)的深度值序列的共同的“蜘蛛网”连接图。(h), (i) 基于(g)的“蜘蛛网”连结图的初末多边形的同构三角剖分。注意，其中的四边形面由于为凸，故易将其同构三角化。

方法三: Craig Gotsman 和 Vitaly Surazhsky 在研究如何可避免多边形变形过程中的自交问题[60]时，给出了多边形同构三角剖分的另一种算法，上述方法二的通用三角剖分具有“蜘蛛网”结构，该结构包含一个中心顶点和以它为中心的若干多边形（在此称为同心层），相邻同心层的所对应的顶点用弦相连接。Gotsman 和 Surazhsky 所提出的方法正是对方法二的一种改进和提高，利用多边形的形状和相

似性，简化了上述“蜘蛛网”结构，其中同心层可能是不完整的，因而中心顶点到多边形各顶点的辐射状的边的数目就可能是不相同的（如图 3-5 (a) 所示），这样，改进了上述通用三角剖分算法，减少了额外顶点的数目。

给定一个简单多边形，将其三角剖分，考虑该三角剖分的对偶图，对偶图总是一种树形结构，此时可以将该树的所有叶结点依次同时砍掉（相当于砍掉了多边形的一个耳），直到最终只剩下一个顶点，这就是所要找的中心顶点。得到中心顶点之后，每个三角面的深度（对应于该三角面内顶点在树中的深度）就可以计算出来，从而可以计算出多边形内各顶点的里程标（milestones），这里顶点的里程标定义为与该顶点相关联的三角面的深度的最小值（见图 3-5 (b) 和 (c)）。利用初末多边形的里程标序列构造其相应的“蜘蛛网”连接图（见图 3-5 (d) 和 (e)），将这两个“蜘蛛网”连接图融合在一起，计算出对应顶点的最大深度的新序列（见图 3-5 (f)），在此新的深度序列的基础上，可以构造出公共的“蜘蛛网”连接图（见图 3-5 (g)）。最后构造出两个平图，使得其边界为初末多边形，而内部的额外顶点与公共的“蜘蛛网”连接图具有相同的连接关系（见图 3-5 (h) 和 (i)）。这样构造的两个三角剖分是同构的。该算法通过选择初末多边形所允许的最小深度，减少了额外顶点的数目。

方法四：方法三给出的算法能够比较好的改进方法二，减少了许多额外的顶点。在对初末多边形进行同构三角剖分的时候，一个十分重要的环节就是使用尽量少的额外顶点，额外顶点是不希望出现的，因为每增加一个顶点，就会相应的增加解决变形问题的复杂性，同时增加相应计算量。基于方法三的启发，Vitaly Surazhsky 和 Craig Gotsman 在文献[61]中提出了改进算法，该算法能够减少增加的额外顶点。下面介绍一下该文献中提出的算法：方法二能够为每一个多边形建立简化的“蜘蛛网”结构，但是该方法也从最外层多边形（即初末多边形）那里制造了多余的额外顶点，之所以出现了多余的额外顶点这种情况，是因为外一层的顶点对应里面一层多边形的直线段上的顶点，而且此过程从最外层多边形开始，不断的迭代至最里面的中心顶点，额外顶点的数量就这样不断的增加。最后，当一些相对应的顶点在分别的简化“蜘蛛网”结构当中为额外顶点的情况下，可以去除掉这些顶点。

这个网格的作用是合并两个网格的不同之处，去除掉两个网格的相同之处，

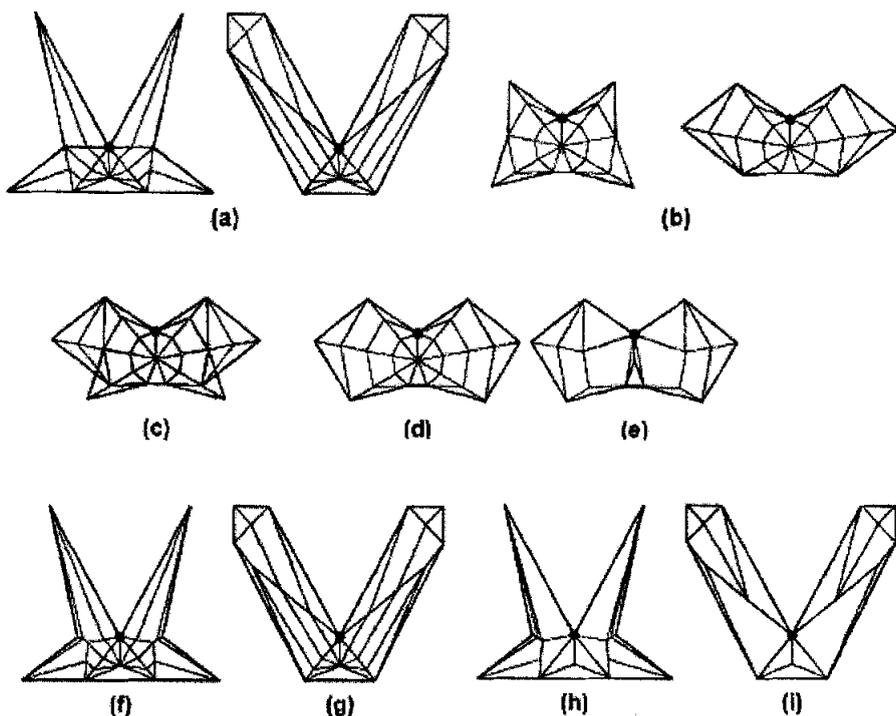


图 3-6 改进性“蜘蛛网”的构造。对应关系是从图中的黑点开始的一个环。(a)对初始和目标多边形分别进行三角剖分，目的是为了得到各个结点的深度；(b)采用“蜘蛛网”结构反映各结点的深度；(c)将(b)中的两个“蜘蛛网”结构融合为一个“蜘蛛网”结构；(d)和(b)中的“蜘蛛网”相同的“蜘蛛网”，即通用“蜘蛛网”结构；(e)改进了的“蜘蛛网”结构算法得到的结果；(f)、(g)为采用(d)的“通用蜘蛛网”结构算法得到的同构三角剖分；(h)、(i)为采用改进的“蜘蛛网”同构三角剖分算法得到的同构三角剖分

这样得到一个新的网格剖分，这个“蜘蛛网”结构的剖分是对应于两个多边形的同构三角剖分。这个剖分减少了尽量多的额外顶点，而且把初末多边形的几何相似性考虑了进来。在此剖分中增加的额外顶点反映了初末多边形之间的几何区别。当用此方法去处理初末多边形是相同的情况时，构造出的同构三角剖分将不会有额外顶点的存在。

图 3-6 用图示表明了该算法的是如何建立初末多边形的同构三角剖分的，而图 3-7 则给出了用此方法得到的结果和以往方法得到的结果的比较。

3.3 相似同构三角剖分算法

前面给出的算法能够达到同构剖分平面多边形的目的，但是增加的顶点个数

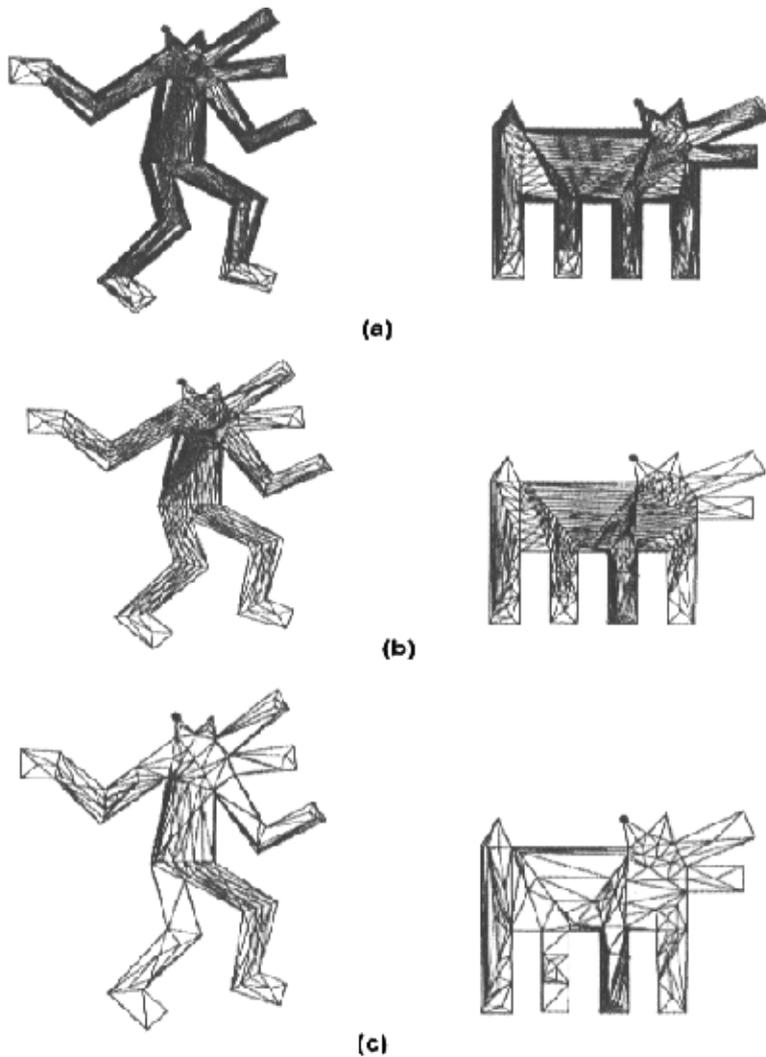


图 3-7 同构三角剖分具有 38 个顶点的两对应的多边形，对应关系是从黑顶点开始的一个环。(a) 采用算法二需要添加 575 个额外顶点；(b) 采用算法三需要添加 334 个额外顶点；(c) 用算法四进行同构三角剖分需要添加 153 个额外顶点

是个需要解决的问题，因为增加的顶点个数越多，对处理增加的难度也就越大。在方法一、方法二中，都没对初末多边形的形状相似性做任何考虑。方法三提出了一个新的同构三角剖分方法，该方法能够较好的减少额外顶点的数目，但是也只是在方法上提出了改进，而对初末多边形形状的相似性没有提出考虑。在方法四中给出了多边形形状相似性的考虑，但是这只是在增加完额外的顶点之后进行，可将部分不需要的额外顶点去掉，但是此方法还有待于提高。

正是鉴于以上方法的不足之处，本文对该算法做了进一步的研究，提出了一种更加有效的同构剖分方法，而且也得到了比较好的结果。本文的算法主要是将需要进行同构三角剖分的多边形之间的相似性考虑进来。利用此算法能够很好的减少额外顶点的数目，从而使得处理初末多边形变得更容易些。

3.3.1 算法基础

定义 3.8 若多边形 P 的顶点序列 p_1, p_2, \dots, p_n 按逆时针方向排列，并且点 p_i 在 $\overline{p_{i-1}p_{i+1}}$ 的左（右）侧，则称顶点 p_i 为凸（凹）点。以凸（凹）点为顶点的内角 $< \pi$ ($> \pi$)。

定理 3.1 对于任意一个简单封闭的多边形，在其内部必定含有凸顶点。

证明 （采用反证法）假设在简单封闭的多边形内部没有含有凸顶点。

假设由 n 个顶点构成多边形 P ，其顶点按逆时针方向排列，定义相邻的两向量 $\overline{p_{i-1}p_i}$ 与 $\overline{p_i p_{i+1}}$ 之间的有向夹角为 θ_i ($-\pi < \theta_i < \pi$)，当 $i=1$ 时， θ_1 为向量 $\overline{p_n p_1}$ 与向量 $\overline{p_1 p_2}$ 之间的有向夹角。各变量的定义如图 3-8 所示。

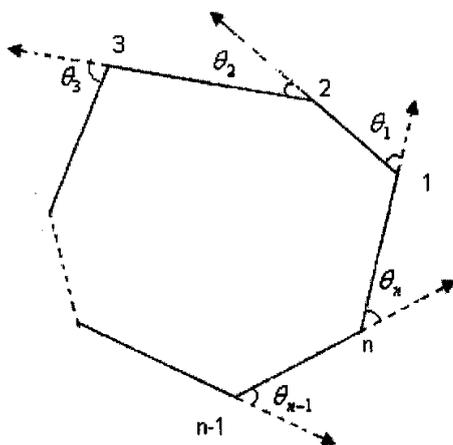


图 3-8 简单封闭多边形

由凸点的定义以及给出的假设，则任意两条向量之间的有向夹

角为一负角，即 $\theta_i < 0$ ($i=1, 2, \dots, n$)，从而有 $\sum_{i=1}^n \theta_i < 0$ ，这与给出的多边形顶点按逆时针方向排列相矛盾。从而存在有有向角 $\theta_j > 0$ ，即对于任意一个简单的封闭的多边形，其顶点序列中必定存在有凸顶点。

证毕

定理 3.2 顶点数 $n \geq 4$ 的多边形至少有一条对角线。

证明 由定理 3.1，多边形至少含有一个凸顶点，设 p_i 是一个凸顶点， p_{i-1} 和

p_{i+1} 是与 p_i 相邻的顶点。如果 $\overline{p_{i-1}p_{i+1}}$ 是一条对角线,则定理成立。否则,假定 $\overline{p_{i-1}p_{i+1}}$ 不是一条对角线,则或者 $\overline{p_{i-1}p_{i+1}}$ 在 P 的外部,或者它与 ∂P 相交。在这两种情况的任一情况下,既然 $n > 3$,三角形 $p_{i-1}p_i p_{i+1}$ 至少包含 P 的一个顶点(除了 p_{i-1}, p_i, p_{i+1} 之外),设 p 是距 p_i 最近的、并在三角形 $p_{i-1}p_i p_{i+1}$ 内部的 P 的顶点,

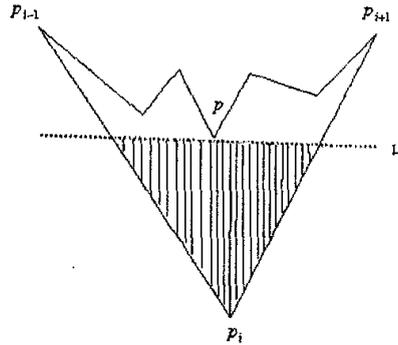


图 3-9

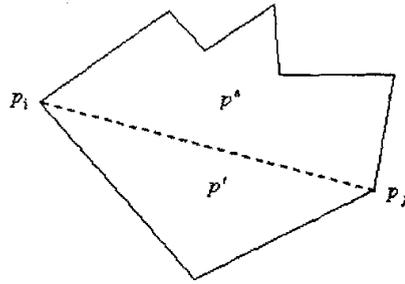


图 3-10

该顶点是从 p_i 向 $\overline{p_{i-1}p_{i+1}}$ 移动并平行于 $\overline{p_{i-1}p_{i+1}}$ 的线 L 所碰到的三角形 $p_{i-1}p_i p_{i+1}$ 的第一个顶点。如图 3-9 所示。

显然,由于包含 p_i (上图中的阴影域)的 L 所限界的半平面相交的三角形 $p_{i-1}p_i p_{i+1}$ 的内部不再包含 ∂P 的顶点,因此,除了在 p_i 和 p , $\overline{p_i p}$ 不可能与 ∂P 相交,即 $\overline{p_i p}$ 是一条不与 ∂P 相交的对角线。证毕

定理 3.3 设多边形 P 具有 n 个顶点,对 P 通过添加零条或者多条对角线可以剖分成三角形。

证明 对多边形顶点数 n 施归纳法。如果 $n=3$,多边形 P 是一个三角形,结论显然成立。设 $n \geq 4$,由定理 3.2,假定 $d = p_i p_j$ 是 P 的一条对角线,另由对角线的定义, d 仅在其端点与 ∂P 相交,并把 P 划分成两个多边形 P' 与 P'' ,这两个多边形以 d 作为一条公共边,而且每个多边形的顶点数少于 n ,如图 3-10 所示,这是由于该剖分过程中没有增加新的顶点。显然,在 P' 和 P'' 中除了 p_i 与 p_j 之外至

少有一个顶点。对 P' 与 P'' 应用归纳法假设便可以完成此证明。

证毕

定理 3.4 无孔简单多边形三角剖分的对偶 T 是一棵树。

证明 假设 T 不是树，则它必有一条回路，组成该回路的结点所对应的三角形必形成多边形内的一个孔，这与已知条件矛盾，因此 T 是一棵树。 证毕

定理 3.5 顶点数 $n \geq 4$ 的任意多边形至少有两个不重叠的耳。

证明 三角剖分对偶 T 中的一个叶结点对应于一个耳。具有两个结点的数，有两个不重叠的耳。具有多个结点的树，有 $(n-2) > 2$ 个结点，必至少有两个叶结点。

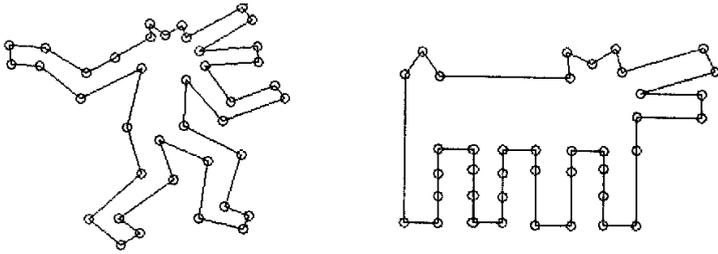
证毕

定义 3.9 (相似顶点) 在初末多边形的顶点对应关系已经确定的情况下，当一对对应顶点都为耳尖，则称该对顶点为相似顶点。

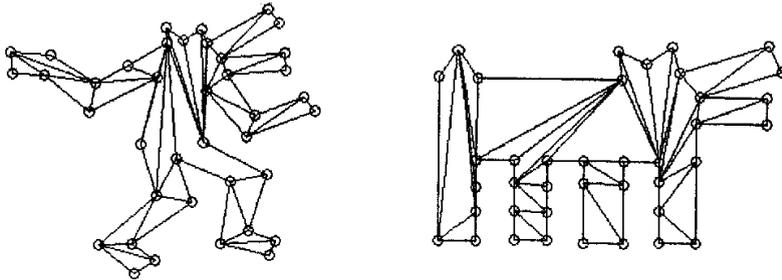
3.3.2 剖分算法

从 3.2.1 节中知道，任意一个简单多边形都含有耳，与耳尖相邻的两个多边形顶点的连线与耳一起构成一个三角形，这三条线段为该三角形的三条边。当用该对角线替换掉该凸顶点时，则构成一个新的多边形，该新多边形比原来的多边形少了一个顶点和一条边，该新多边形仍然存在耳，重复前面的步骤，则可以将该多边形三角剖分。对于同构三角剖分，在初末多边形的顶点对应关系已经确定的情况下，可以先将两多边形的相似顶点去掉，当去掉一对相似顶点之后，便形成了一对新的初末多边形，继续去掉它们之间的相似顶点，最终使得初末多边形达到最简，即在此最简初末多边形之间不存在相似顶点。然后采用前面介绍的方法一对最简多边形进行同构三角剖分，再和前面去掉相似顶点时所得到的三角网格组合起来，便得到了初末多边形的同构三角剖分。整个过程可描述为：总——分——总——分。下面给出本文算法的主要步骤。

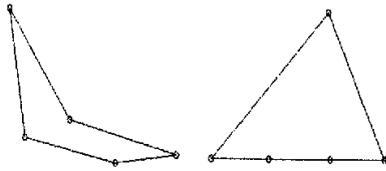
设 P_s 为由 n 个顶点 $p_s^i (i=1, 2, \dots, n)$ 构成的初始多边形， P_t 为由 n 个顶点 $p_t^i (i=1, 2, \dots, n)$ 构成的目标多边形，且初末多边形顶点之间的对应关系已经确定，其对应关系假定为 p_s^i 对应 p_t^i 。则算法步骤如下：



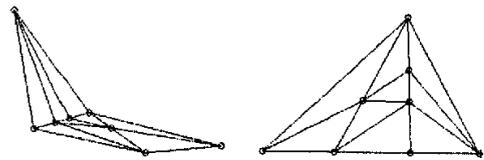
(a) 原始初末多边形



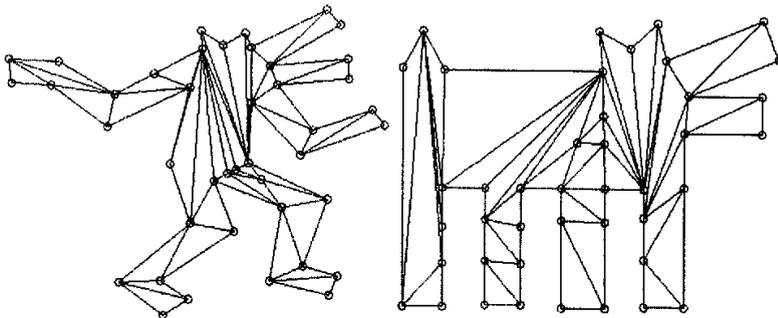
(b) 采用本文算法三角剖分原始初末多边形的相似顶点所得到的结果，内含一对由 5 个顶点的组成的最简多边形



(c) 由本文算法得到的最简初末多边形



(d) 采用方法 1 得到的最简单初末多边形之间的同构三角网格



(e) 将 (c) 与 (d) 组合起来便得到了原始初末多边形之间的同构三角网格

图 3-11 该图所示为机械人和机械狗的同构三角剖分，用本文算法只添加了 3 个额外顶点就可以达到同构三角剖分。

- 步骤 1. 令 $m = 0$, $num = n$, $P_{sm} = P_s$, $P_{tm} = P_t$ 。
- 步骤 2. 从 1 到 num 线性搜索初末多边形的相似顶点, 如果在初末多边形之间存在相似顶点, 则转步骤 3; 如果多边形的顶点个数小于 4, 则停止; 否则转步骤 4。
- 步骤 3. 假设搜索到的相似顶点为顶点 p_s^k 与 p_t^k (k 为大于等于 1 且小于等于多边形的顶点个数 num 的整数), 当 $k = 1$ 时, 连接顶点 2 与顶点 num ; 当 $k = num$ 时, 连接多边形的顶点 $num - 1$ 与顶点 1; 对于其余的情况, 连接多边形的顶点 $k - 1$ 与 $k + 1$; 将由顶点 $p_{sm-1}^1, \dots, p_{sm-1}^{k-1}, p_{sm-1}^{k+1}, \dots, p_{sm-1}^{num}$ 构成的多边形记作新初始多边形 P_{sm} , 由顶点 $p_{tm-1}^1, \dots, p_{tm-1}^{k-1}, p_{tm-1}^{k+1}, \dots, p_{tm-1}^{num}$ 构成的多边形记作新目标多边形 P_{tm} , $num = num - 1$ 。转步骤 2。
- 步骤 4. 对最终的多边形 P_{sm} 与 P_{tm} 进行同构三角剖分, 将在此得到的同构三角剖分和前面得到三角剖分结合起来, 便得到原始多边形的同构三角剖分。

3.3.3 算法实例

下面将参考文献[61]中的实例引用在此说明本文的算法过程。该算法的整个过程如图 3-11 所示, 该算法只需要 3 个额外顶点, 从算法结果得出, 和其他几种算法 (如图 3-6) 相比, 该算法取得了较好的效果。图 3-12 是引用于参考文献[61]的一个实例, 用本文算法得到的同构三角剖分所使用的额外顶点少了, 计算难度也降低了。针对图例 3-11, 表 3-1 给出了本文算法和以往算法的比较。

表 3-1 同构剖分算法比较表

比较项	方法一	方法二	方法三	方法四	本文算法
额外顶点数 (个)	417	575	334	153	3
计算复杂度	大	大	较大	一般	小

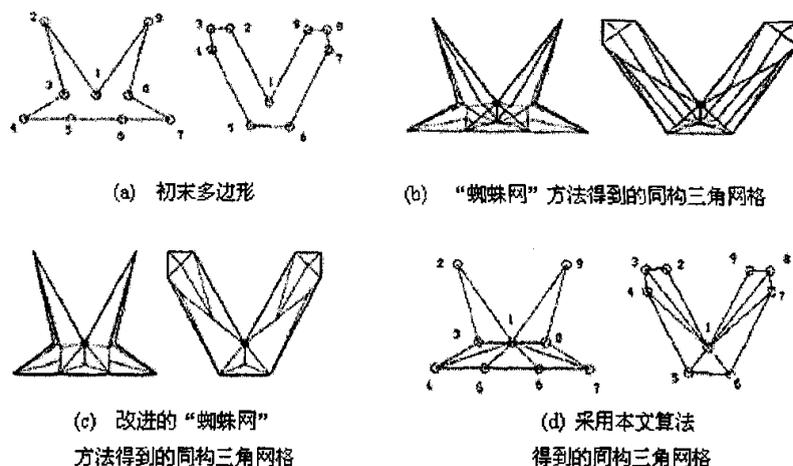


图 3-12 (b)中需要添加 15 个额外顶点；(c) 中需要添加 7 个额外顶点；
(d) 不需要添加额外顶点就可以得到同构三角剖分

3.4 算法改进

上面介绍的方法没有能够很好的考虑到多边形同构三角剖分的质量，可以通过设置剖分变量来提高剖分质量，比如设置剖分出的同构三角网格的最小角度不小于多少度，最大角度不大于多少度等约束条件。理想的三角剖分应该是剖分出的三角形都是正三角形，通过增加的约束条件，本文算法能够得到很好的剖分结果。

3.5 本章小结

对给定对应关系的初末多边形进行同构三角剖分，本章提出了多边形的相似性同构三角剖分，它更全面的考虑到了初末多边形之间的相似性，能够使得对多边形进行同构三角剖分时增加尽量少的额外顶点。当给定的初末多边形相同时，得到的同构三角剖分不需要添加额外的顶点就能够达到同构三角剖分。得到初末多边形的同构三角网格之后，接下来的工作是对初末多边形进行变形，变形算法将在下一章作详细介绍，并提出了基于小波的多边形变形算法。

第四章 基于小波的平面多边形变形算法

平面多边形的变形研究具有很重要的意义，平面多边形变形是平面曲线及其他形式的变形的的基础，因而平面曲线可以用离散的多边形来逼近，平面曲线的变形，往往转化为平面多边形的变形来实现。目前有许多研究者对这方面进行了探索研究，如 Sederberg 和王国瑾等的内在解算法，Shapira 和 Rapport 的星骨架 (Star skeleton) 方法，Gotsman 和 Surazhsky 的可避免自交的变形方法等。

4.1 已有的平面多边形变形方法介绍

4.1.1 线性插值变形

给定两个具有相同顶点个数的多边形 P^0 和 P^1 ，外形融合可以通过线性插值两个多边形的对应顶点来完成，线性插值只能处理初末多边形的顶点个数相同的情况。当初末多边形的顶点个数不同时，可以在多边形上增加额外的顶点使它们的顶点个数达到相同。分别定义 P^0 和 P^1 如下：

$$P^0 = [P_0^0, P_1^0, \dots, P_n^0]; \quad P^1 = [P_0^1, P_1^1, \dots, P_n^1]$$

其中 P_i^k 表示变形过程中多边形的顶点，当 $k=0$ 时，表示的是初始多边形的顶点；当 $k=1$ 时，表示的是目标多边形的顶点，则变形过程如下定义：

$$\begin{aligned} P(t) &= uP^0 + tP^1 \\ &= [uP_0^0 + tP_0^1, uP_1^0 + tP_1^1, \dots, uP_n^0 + tP_n^1] \\ &= [P_0(t), P_1(t), \dots, P_n(t)] \end{aligned}$$

其中 $u+t=1$ 。用此算法可以得到如图 4-1、4-2 与 4-3 所示的结果，该算法计算速度快，能够很快的得到变形序列，但是该算法只考虑到顶点的对应关系，根本就没有考虑到同一个多边形内部顶点之间的关系，从而导致该方法在变形过程中容易产生自交，如图 4-4 所示。所以导致该方法不太适用。

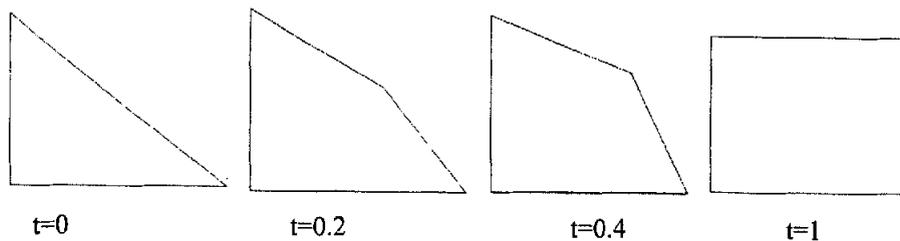


图 4-1 该图表示的是从一个三角形变形到四边形的变形序列

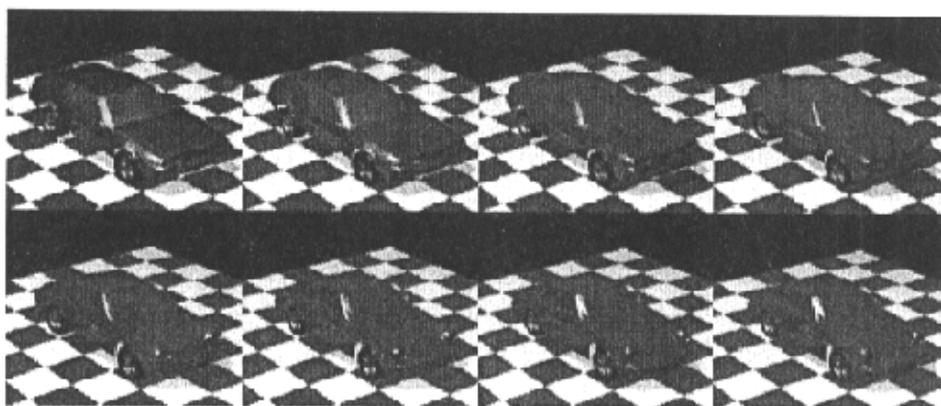


图 4-2 利用基域，通过线性插值得到的变形序列



图 4-3 一个亏格大于零的物体之间的变形序列

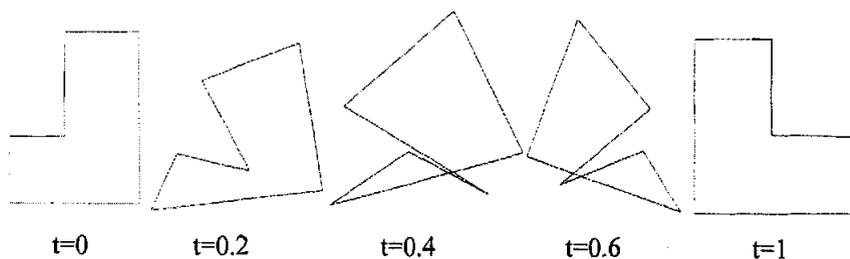


图 4-4 从初始多边形变形到目标多边形的过程，由图可见，在变形的过程中，发生了自交现象

4.1.2 平图变形算法

C.Erten 等人在文献[12、13]中提出了保持不自交的变形算法为：将准备变形的初末平图嵌入到两个相同凸边界里，从初末平图中选出一对对应的顶点，并且该对顶点对于凸边界的顶点是可见的，相应的连接平图上的顶点与凸边界上的顶点，然后对两幅平图进行同构三角剖分，得到对应的同构三角网格，作者将变形过程分为两个过程：其一，用二维的仿射变换将初末平图之间的距离拉近，该仿

射变换被考虑为一个 2×2 的矩阵加上一行表示平移变量的行：
$$\begin{pmatrix} c_{11} & c_{12} & 0 \\ c_{21} & c_{22} & 0 \\ t_x & t_y & 1 \end{pmatrix}$$
，则

对于初始平图中的一点 (x, y) 经此仿射变换后，变换为：

$$x' = c_{11}x + c_{21}y + t_x \qquad y' = c_{12}x + c_{22}y + t_y$$

假设初始平图中点 $p_s = (x_s, y_s)$ 对应于目标平图中点 $p_t = (x_t, y_t)$ ，则上面仿射变换的各参数为满足下述条件的量：这些参数使得所有对应顶点的距离的平方和，即：

$\sum_{p_i \in D_s} dist^2(p_i, p_t)$ 最小，其中 $dist$ 定义为平面空间上两顶点的欧式距离。在得到仿

射变换矩阵之后，将该矩阵分解为旋转矩阵与伸缩矩阵之积。分别对该两矩阵进行插值后再组合，则得到中间时刻的仿射变换矩阵，从而得到该时刻的图形。其

二，是对平图的内部顶点进行变形，该文所采用的方法是基于文献[21]中的思想：给定一个平图，将该平图的最外层映射到一个凸多边形上，则平图的内部的每一个顶点都是该顶点的相邻位置上顶点的一个凸组合，而且其可以表示为：

$$u_i = \sum_{j \in N(i)} \lambda_{ij} \times u_j, \quad \sum_{j \in N(i)} \lambda_{ij} = 1, \quad \text{其中 } \lambda_{ij} \text{ 是顶点 } u_i \text{ 相对于顶点 } u_j \text{ 的凸组合系数。 } N(i)$$

是顶点 u_i 的邻接顶点坐标集。作者将此凸组合表示为一 $n \times n$ 矩阵 (λ_{ij} 对应于该矩

阵中第 i 行第 j 列的元素值)。然后线性插值初末平图的邻接矩阵，得到中间时刻内

顶点的邻接矩阵，结合步骤一则得到了在 t 时刻的整个平图。

4.1.3 基于凸组合的内在解算法

上一小节介绍的保持非自交的变形算法，当对邻接矩阵进行线性插值时，其

实是对组合系数 λ_j 进行线性插值，多边形内部的边与角度不是成线性变换的。正是基于此，VITALY SURAZHSKY 等人在文献[64]中提出了线性插值多边形的边与角度的算法。在计算组合系数 λ_j 时，是采用文献[37]中的方法：

$$\lambda_i = \frac{\omega_i}{\sum_{j=1}^k \omega_j} \quad \omega_i = \frac{\tan\left(\frac{\alpha_i}{2}\right) + \tan\left(\frac{\alpha_{i-1}}{2}\right)}{l_i} \quad (4.1)$$

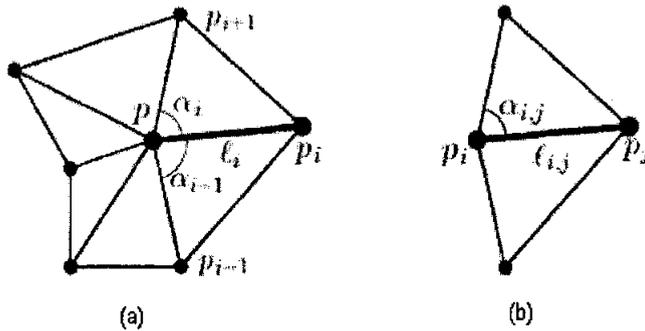


图 4-5 公式(4.1)中各变量的定义

其中各变量如图 4-5 所示。本文的方法线性插值角度与边长，即： $\alpha'_i = (1-t)\alpha_i + t\alpha'_i$ ，

$l'_i = (1-t)l_i + tl'_i$ 。从而得到 t 时刻的组合系数为： $\lambda'_i = \frac{\omega'_i}{\sum_{j=1}^k \omega'_j}$ ，

$$\omega'_i = \frac{\tan\left(\frac{\alpha'_i}{2}\right) + \tan\left(\frac{\alpha'_{i-1}}{2}\right)}{l'_i}$$

此变形方法能够使得初末多边形三角网格的边以及角度之间连续的变化。但是，该方法也容易让内部区域产生扭曲。

4.1.4 加权曲线变形算法

在文献[65]中，作者提出了加权曲线变形算法。对于平面上对应的顶点 p_1 与 p_2 ，从顶点 p_1 映射到顶点 p_2 的路径的最简便方法是线性插值： $p(t) = (1-t)p_1 + tp_2$ ， $t \in [0,1]$ ，则每一个中间顶点 $p(t)$ 可看作是对顶点 p_1 与 p_2 的加权组合。且 $p(t)$ 满足

方程:

$$d_E(p(t), p_1) = t \cdot d_E(p_1, p_2) \quad d_E(p(t), p_2) = (1-t) \cdot d_E(p_1, p_2)$$

其中 d_E 表示的是两点间的欧式距离。正是基于此加权组合的思想, 作者通过采用某距离函数 $d(x, y)$, $x, y \in U$, 其中 U 为一任意的空间。使得变形过程为从任意初始物体 P_1 变形到目标物体 P_2 的中间物体 $P(t)$, $t \in [0, 1]$, 并且保持性质:

$$d(p(t), p_1) = t \cdot d_E(p_1, p_2) \quad d(p(t), p_2) = (1-t) \cdot d_E(p_1, p_2)$$

通过一系列 xy 平面上的顶点 $(x_1, y_1), \dots, (x_n, y_n)$ 表示的二维曲线, 对此曲线变形的首要步骤是对此曲线进行采样, 得到采样点 $(\bar{x}_1, \bar{y}_1), \dots, (\bar{x}_m, \bar{y}_m)$, 通过这些采样点, 二维曲线可用一串向量表示, z_1, \dots, z_{m-1} , 其中 z_i 是从顶点 (\bar{x}_i, \bar{y}_i) 到顶点 $(\bar{x}_{i+1}, \bar{y}_{i+1})$ 的向量。

作者采用著名的 Levensht 串距离函数作为两个串之间距离的测试函数, 该测试函数包含三种算子: 删除 ($a \mapsto \xi$), 插入 ($\xi \mapsto a$) 和减 ($a \mapsto b$), 相应的代价为:

$$C(a \mapsto \xi) = C(\xi \mapsto a) = \|a\|, \quad C(a \mapsto b) = \|a - b\|。$$

给定两条用向量串表示的初末曲线后, 采用文献[23]中提供的算法计算在 t 时刻的图形所对应的初始图形与目标图形的权值。在采用文献[45]中的算法计算两向量串所表示的曲线的距离时, 能够得到如何采用上面介绍的算子将初始图形变形到目标图形。

4.1.5 定角插值变形算法

Ping-Hsien Lin 等在文献[44]中提出的算法也是在同构三角网格的基础上提出来的。首先将三角网格中的三角形用三条线段来表示, 这三条线段的公共端点位于三角形的内部, 而另一端则分别连在三角形的三个顶点上, 从而一改过去插值三角形的方法, 而插值对应三角形内的对应线段。

对线段的插值方法一般采用线性插值对应的边 (l) 和对应的角度 (θ):

$$\begin{cases} l(t) = (1-t)l_s + tl_t \\ \theta(t) = (1-t)\theta_s + t\theta_t \end{cases} \text{ 其中 } S \text{ 代表初始状态, } T \text{ 代表目标状态}$$

而在该文中提出的算法是在变形的过程中, 线段的非中心顶点的轨迹上任意一点的切线与该点处的向量夹角应该是固定的 (如图 4-6 所示), 即 $\frac{dr}{rd\theta} = a$, 其中 a 是一个常量, 两边积分得到: $r = Ce^{a\theta}$, 从而提出插值公式:

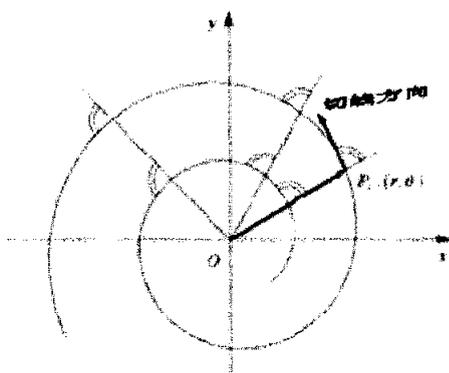


图 4-6 $r = e^{a\theta}$ 的图示

$$\begin{cases} \theta(t) = (1-t)\theta_s + t\theta_t \\ l(t) = l_s e^{a(\theta(t)-\theta_s)} \end{cases}$$

其中 $a = \frac{1}{\theta_t - \theta_s}$, 且 $\theta(t)$ 是个单调函数。

数。

对整个初末同构三角网格中每一个三角形采用此插值方法时, 由于其中的某些顶点同时属于不同的三角形, 单独变形后的顶点可能出现不重合的情况, 则在变形前, 将初末多边形中的三角形一个接一个的按一定的序列组合起来, 即选定其中的某个三角形为根三角形, 按照深度遍历或者广度遍历将三角形排列起来。首先对根三角形进行插值, 然后按序列依次对三角形进行插值, 当现在处理的三角形的顶点与已经处理过的三角形的某些顶点本应该重合而未重合时, 选择他们公共边上的一个顶点为基点, 利用表示三角形的三条线段中的两条线段确定不属于公共边的那个顶点。连接相应的顶点则求得这个三角形。如此迭代至所有的三角形处理完, 则得到了初末多边形对应于此时刻的图形。

4.1.6 内在解算法变形

Thomas W. Sederberg 等人在其文献[57]中提出了内在解算法, 并于 1998 年建立了用于空间多边形或多面体变形的算法 MSI [1], 就平面内在解算法, 其变形方法如下:

当给定一个多边形时, 计算该多边形的内在集 $\Gamma = \{(L_i, \theta_i) | i = 1, 2, \dots, n\}$, 其中 n 为多边形的顶点个数, L_i 为相邻两顶点 p_{i+1} 与顶点 p_i 之间的欧氏距离, 当 $i = 1$ 时,

θ_i 为顶点向量 $\overline{p_1 p_2}$ 与 x 轴之间的有向夹角, 否则 θ_i 为后两顶点所成向量 $\overline{p_{i+1} p_i}$ 与前两顶点所成向量 $\overline{p_i p_{i-1}}$ 的一个夹角。内在集的计算可表示为如下的公式:

$L_i = |p_{i+1} - p_i|$, $\theta_i = \text{angle}(\overline{p_{i+1} p_i}) - \text{angle}(\overline{p_i p_{i-1}}) - m\pi$, 其中这里的 *angle* 表示的是向量的角度, m 为使得不等式 $-\pi < \theta_i < \pi$ 成立的偶数。则通过顶点表示的多边形

可以重新用多边形的首顶点与该内在集重新表示。通过首顶点与内在集, 多边形可以通过如下方式得到:

$$p_{i+1} = p_i + L_i \times \left(\cos\left(\sum_{k=1}^i \theta_k\right), \sin\left(\sum_{k=1}^i \theta_k\right) \right) \quad i = 1, \dots, n \quad (4.2)$$

当 $i = n$ 时, p_{n+1} 所表示的顶点就是 p_1 。

当给定初末多边形 P_0 与 P_1 时, 首先计算出初末多边形对应的内在集 Γ^0 与 Γ^1 , 对初末多边形的变形处理即对初末多边形的内在集进行线性插值: $\Gamma^t = (1-t) \cdot \Gamma^0 + t \cdot \Gamma^1$ ($0 \leq t \leq 1$) 得到对应于 t 时刻多边形的内在集表示 Γ^t , 通过 (4.2) 即可得到 t 时刻的多边形。

在用此算法变形封闭图形的时候, 所得到的中间多边形往往是不封闭的, Thomas W. Sederberg 等人给出了相应的补偿常量, 使得中间图形在变形过程中保持封闭, 补偿常量的大小可参见文献[57]。该方法考虑到了局部顶点之间关系, 而没有考虑到整体顶点之间的联系, 从而无法避免在变形过程中出现自交的现象。

4.1.7 方向插值变形

有几种现存的方法可以用来求两个图形的相对方向。由于插值两个以上三维图形的方向比较困难, 所以接下来讨论的方向插值将严格限制在两个图形之间。

第一步, 通常对图形做变换使得它们的重心与原点一致。然后通过将刚性/仿射部分从变形的弹性部分分离出来, 计算一个旋转[14、16]或仿射变换。定义刚性/仿射部分的一种方法是最小化应用相应变换后对应顶点距离的平方。寻找仿射变换的最小化问题可以利用坐标向量的广义逆来求解。令顶点向量排列成一个 $n \times 3$

的矩阵 $V = \begin{pmatrix} v_{1x} & v_{1y} & v_{1z} \\ v_{2x} & v_{2y} & v_{2z} \\ \dots & \dots & \dots \end{pmatrix}$ ，那么在一个仿射变换 A 作用下，坐标距离的平方为

$(V(0)A - V(1))^2$ ，并使其最小化，这是一个线性方程组，可以利用广义逆 $V(0)^+$ 求解：

$$A = V(0)^+ V(1) = (V(0)^T V(0))^{-1} V(0)^T V(1) \quad (4.3)$$

作为选择，最小二乘解（或者说广义逆）可以利用 SVD（奇异值分解）来计算，这种分解允许对近秩缺失的敏感性做显式的控制[12]。

中间图形 $V(t) = \{V_1(t), V_2(t), \dots\}$ 可表示为 $V(t) = A(t)V(0)$ ，问题是怎样定义 $A(t)$ 才合理。最简单的解决方案是： $A(t) = (1-t)I + tA$ ，然而，我们需要更详细的方法以得到所期望的 $A(t)$ 的一些性质：1、变换应该是对称的；2、旋转角和面积应该单调的变化；3、变换不能带来不良影响；4、得到的路径应该简单。

基本的思想是将 A 分解成旋转（正交矩阵）部分和具有正的放缩比例分量的比例剪切部分。Alexa 等[13]检验了几种分解，通过试验，他们发现一种分解为一个旋转矩阵和一个对称矩阵的方法（如极分解），该方法能产生在视觉上最好的变换。这个结论被 Shoemake 以数学和心理的理由加以支持。分解可以从 SVD 推导而来，如下：

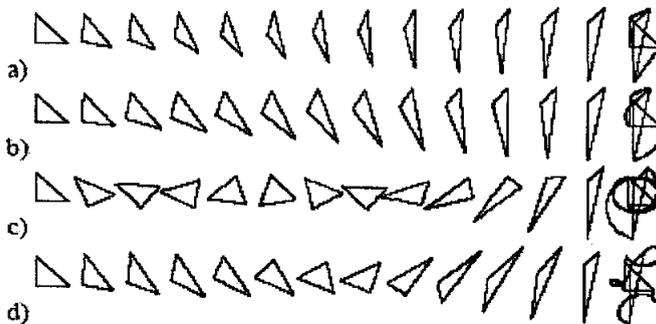


图 4-7 变换单个三角形。(a)线性插值；(b-d) 计算从源三角形到目标三角形的一个仿射映射，并将其分解为旋转部分和比例-剪切部分。中间三角形是通过线性插值旋转角、比例因子和剪切参数建立的。(b)是利用 SVD 建立的；(c)是通过将(b)中所有的角中的某一角减去 2π 所推导得到的结果；(d)对应于方程(4.5)的结果；所有行的最后一列画出了顶点的路径。

$$A = R_\alpha D R_\beta = R_\alpha (R_\beta R_\beta^T) D R_\beta = (R_\alpha R_\beta) (R_\beta^T D R_\beta) = R_\gamma S \quad (4.4)$$

然而, [74]中也指出, 有在计算上比这个分解更节约的选择。基于这种分解, $A(t)$ 可以通过线性插值 (4.4) 中因子里的自由参数来计算得到, 也就是说,

$$A_\gamma(t) = R_\gamma ((1-t)I + tS) \quad (4.5)$$

图 4-6 是一个三角形的变换结果。为了比较, 图 4-7 (a) 演示的是顶点坐标的线性插值; 图 4-7 (b) 是通过奇异值分解和线性插值得到的变换 $A_{\alpha,\beta}(t)$ 的演示结果, 注意到结果虽然是对称的且旋转角是线性变化的, 但结果仍不令人满意, 因为一个超过 π 的旋转是不必要的。当从一个角减去 2π , 如图 4-7 (c), 结果就会更好一点。我们发现将 A 分解为一个旋转矩阵和一个对称阵并利用 $A_\gamma(t)$ 产生了最好的效果, 如图 4-7 (d), 与 SVD 相比, 它避免了不必要的旋转或剪切, 并且通常比基于 QR 分解的方法更对称。

4.2 基于小波的多边形变形算法

小波函数是一个逐层分解函数的数学工具, 它允许一个函数被表示为这个函数的粗糙的大致结构加上细节, 使得该函数逐步精确。不管所要处理的函数所表示的是一幅图像、一条曲线或者是一个曲面, 小波为处理要处理的目标对象的细节提供了很好的技术支持。

哈尔小波是最简单的小波基。为了对小波基函数是如何起到作用有一个初步的认识, 首先从一个简单的例子开始。假定这里有一幅只有四个像素的图像, 且它们的值为 [11 9 5 6], 可以采用哈尔小波基的形式重新表示这幅图像, 其值通过小波变换得到。首先成双成对的计算两个像素值在一起的平均值, 这个平均值就可以作为下一层图像像素的值, 可以明显的看到, 经过处理过后的图像之后有两个像素点 [10 6]。很清楚的可以看到, 在采用上面的方法处理的时候, 一些信息已经被丢失掉了, 为了能够从两个像素点的图像恢复得到原始的四点图像, 所以需要加上一些细节系数, 通过这些细节系数, 就能够抓住通过上面方法处理时丢失的信息。在上面给出的例子中, 应该选择 1 做为第一个细节系数, 那是因为做对 11 和 9 做平均得到 10 比 11 少 1, 比 9 多 1。通过这个数就可以恢复出原始图像的前两个像素, 同样的道理, 第二个细节系数为 -1。所以原始图像的四个象

素点通过分解得到下一层的结果为：两个象素点和相对应的两个细节系数。重复这一过程，直到所需要的结果。

分解层次	平均	细节系数
4	[11 9 5 6]	
2	[10 6]	[1 -1]
1	[8]	[2]

最终，定义小波变换（或者叫小波分解）分解原始的四点图像得到的一个点为原始图像的总轮廓，和一些重构原始图像所需要的细节系数。所以，上面给出的一维图像[11 9 5 6]可以表示为下面的方式： $[8 \ 2 \ 1 \ -1]$ 。

下面给出了图形说明该方法的工作过程，如图 4-8 所示：

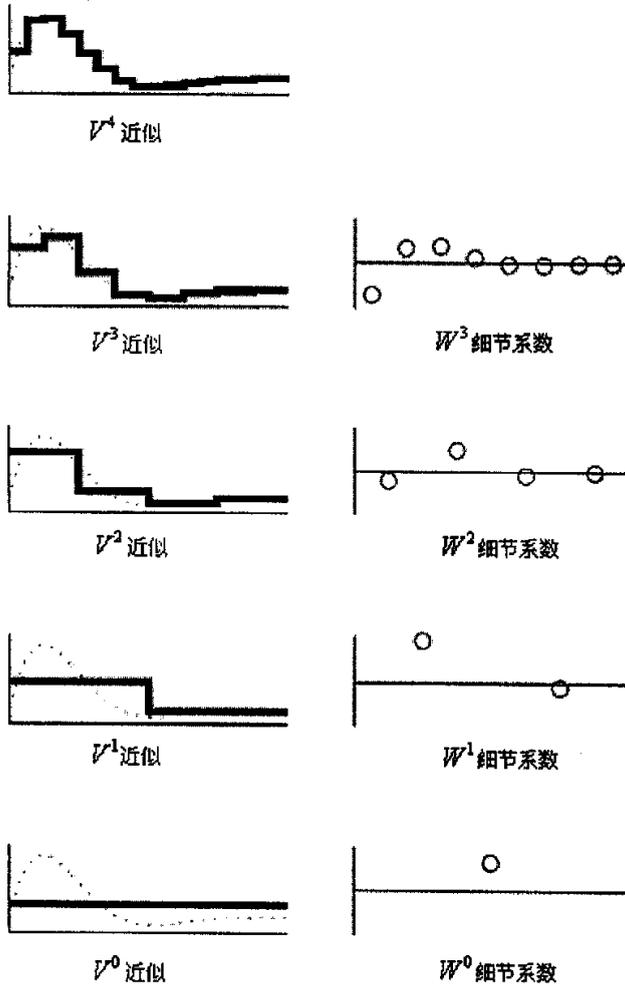


图 4-8 小波分解的分解示意图

在文献[67]中作者给出了用小波分解和重构方法来进行多边形变形的办法。通过应用小波变换，一个具有 2^j 个顶点的多边形 P^j 可以被分解为多分辨多边形，即一个有结构组织的一系列轮廓多边形 P^0, P^1, \dots, P^{j-1} 和相应的细节信息 D^0, D^1, \dots, D^{j-1} 。在小波变换里， P^i 和 $D^i, i=0, 1, \dots, j$ 分别被看作是向量空间 V^i 和 W^i 中的向量。在这里，向量就是多边形顶点所在的坐标所定义的向量。对于向量空间 V^i 的基函数 $\Phi^i = \{\phi_k^i, k=0, 1, \dots, 2^i-1\}$ 被称为尺度函数，向量空间 W^i 的基函数 $\Psi^i = \{\psi_k^i, k=0, 1, \dots, 2^i-1\}$ 被称为小波函数。

给定两个多边形 P_1^j 和 P_2^j ，用前面介绍的小波分解算法，可以将两个多边形分别分解为 $P_1^0, P_1^1, \dots, P_1^{j-1}$ 和 $P_2^0, P_2^1, \dots, P_2^{j-1}$ 以及相应的细节信息 $D_1^0, D_1^1, \dots, D_1^{j-1}$ 和 $D_2^0, D_2^1, \dots, D_2^{j-1}$ ，这里轮廓多边形反应了原始多边形的轮廓，而细节信息则反映了原始多边形的细节部分。在一个适当分解层次上，对多边形的轮廓部分采用前面介绍过的内在解变形方法进行变形，设得到在 t 时刻的多边形轮廓为 $P^k(t)$ 。对于多边形的细节部分则采用线性插值。在对多边形轮廓进行变形的时候，设选择了适当的分解层次 k ，则只须对相对应层次及以上的细节部分进行插值即可得到变形的中间细节：

$$D^i(t) = (1-t)D_1^i + tD_2^i, \quad i = k+1, k+2, \dots, j-1$$

通过上面得到的 $P^k(t)$ 和 $D^i(t)$ ，利用前面介绍的重构方法重构出在 t 时刻的中间图形。

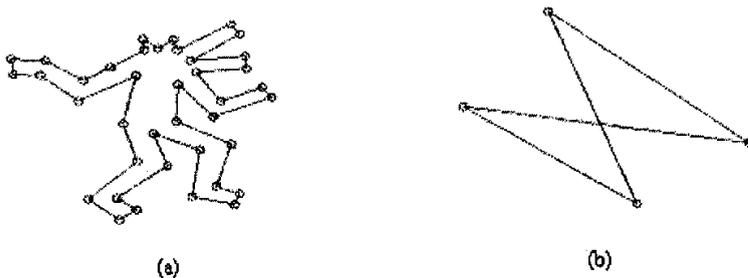


图 4-9 多边形轮廓自交的一个实例

但是该方法不是可靠的, 因为原始多边形的某些细节信息在进行多边形细节变形的时候已经被丢失了, 所以得到多边形不能十分准确的反应多边形变形的中间形状; 其次, 当轮廓多边形拥有不相同个数的顶点时, 则需要增加额外的顶点, 而在细节信息中却没有与之相对应的细节信息, 因而使得重构过程无法进行, 所以对于顶点个数不同的情况, 用此方法是不能处理的; 再次, 算法要求多边形顶点个数必须是 2 的 n 次方个; 最后, 用该方法得到的轮廓多边形在很大程度上是自交的, 如图 4-9 所示。导致该多边形轮廓自交的主要原因是当顶点数大于 2 的 $n-1$ 次方且小于 2 的 n 次方时, 需要添加顶点导致的。因此本章提出了适用性能更强的一种基于小波的多边形变形算法。该方法能够更大程度的保证多边形轮廓不自交, 从而将此小波变形方法与同构三角网格方法结合起来, 可以减少增加额外定点数, 从而使变形算法的计算量大大的得到减小。

4.2.1 小波与小波变换

先回顾一下傅立叶变换, 它使用的是正弦曲线波作为它的正交基函数。之所以称为波, 是因为它们类似于大海的波涛和在其他媒体中传递的波。对于积分变换来说, 这些函数都在两个方向无限扩展。离散傅立叶变换的基向量也在它们的整个域中非零。也就是说它们并不是紧支集 (compact support)。

定义 4.1 如果函数 $\psi(x) \in L^2(R)$ 满足容许条件

$$C_\psi = \int_{\mathbb{R}} \frac{|\hat{\psi}(w)|^2}{|w|} dw < +\infty$$

且满足规范化条件 $\|\psi\| = 1$, 则称 $\psi(x)$ 为基本小波。

定义 4.2 设 $f(x) \in L^2(R)$, $\psi(x)$ 为基本小波, $a, b \in R$, 定义 $f(x)$ 的连续小波变换为:

$$W_f(a, b) = \int_{\mathbb{R}} f(x) |a|^{1/2} \overline{\psi(a(x-b))} dx$$

或者又可以写成

$$W_f(a, b) = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(w) |a|^{-1/2} \overline{\psi\left(\frac{w}{a}\right)} e^{iwb} dw$$

一般来说, 满足容许条件的 $\psi(x)$ 便可以作为基本小波了。但是实际上一般还

要求 $\psi(x)$ 具有一定的正则条件, 以便 $\hat{\psi}(x)$ 在频域上表现出较好的局部性质。

定理 4.1 连续小波变换具有如下性质:

(1) 设 $f(x) = f_1(x) + f_2(x)$, 则

$$W_f(a, b) = W_{f_1}(a, b) + W_{f_2}(a, b)$$

(2) 设 $g(x) = f(x - c)$, 则

$$W_g(a, b) = W_f(a, b - c)$$

(3) 设 $g(x) = f(cx)$, 则

$$W_g(a, b) = |c|^{-1/2} W_f(a/c, bc)$$

(4) 对 $f(x) \in L^2(\mathbb{R})$, 有 $W_f(a, b) \in L^2(\mathbb{R}^2)$, 并且对 $f(x), g(x) \in L^2(\mathbb{R})$, 有

$$\langle W_f(a, b), W_g(a, b) \rangle = C_\psi \langle f(x), g(x) \rangle$$

证明

$$W_f(a, b) = \int_{\mathbb{R}} f(x) |a|^{1/2} \overline{\psi(a(x-b))} dx = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(w) |a|^{-1/2} \overline{\hat{\psi}\left(\frac{w}{a}\right)} e^{iwb} dw$$

而

$$\begin{aligned} \iint_{\mathbb{R}^2} \left| \hat{f}(w) |a|^{-1/2} \overline{\hat{\psi}\left(\frac{w}{a}\right)} \right|^2 dw da &= \int_{\mathbb{R}} |\hat{f}(w)|^2 \left(\int_{\mathbb{R}} \left| |a|^{-1/2} \overline{\hat{\psi}\left(\frac{w}{a}\right)} \right|^2 da \right) dw \\ &= \int_{\mathbb{R}} |\hat{f}(w)|^2 \left(\int_{\mathbb{R}} \frac{|\hat{\psi}(y)|^2}{|y|} dy \right) dw = C_\psi \int_{\mathbb{R}} |\hat{f}(w)|^2 dw < +\infty \end{aligned}$$

由 Fubini 定理, 对几乎所有的 $a \in \mathbb{R}$, $\hat{f}(w) \frac{|a|^{1/2}}{a} \overline{\hat{\psi}\left(\frac{w}{a}\right)}$ 关于 w 是平方可积的, 从而

$W_f(a, b)$ 关于 b 是平方可积的。故

$$\iint_{\mathbb{R}^2} |W_f(a, b)|^2 da db = \frac{1}{2\pi} \iint_{\mathbb{R}^2} \left| \hat{f}(w) |a|^{-1/2} \overline{\hat{\psi}\left(\frac{w}{a}\right)} \right|^2 dw da < +\infty$$

即 $W_f(a, b) \in L^2(\mathbb{R}^2)$, 进而

$$\langle W_f(a, b), W_g(a, b) \rangle = \frac{1}{2\pi} \left\langle \hat{f}(w) |a|^{-1/2} \overline{\hat{\psi}\left(\frac{w}{a}\right)}, \hat{g}(w) |a|^{-1/2} \overline{\hat{\psi}\left(\frac{w}{a}\right)} \right\rangle$$

$$= \frac{C_\psi}{2\pi} \langle \hat{f}(w), \hat{g}(w) \rangle = C_\psi \langle f(x), g(x) \rangle \quad \text{证毕}$$

定理 4.2 对 $\forall f(x) \in L^2(R)$, 当 $\hat{f}(w) \in L^1(R)$ 时有

$$f(x) = \frac{1}{C_\psi} \iint_{\mathbb{R}^2} W_f(a, b) |a|^{1/2} \psi(a(x-b)) da db$$

证明

由于 $\overline{(|a|^{1/2} \psi(a(x-b)))^\wedge(w)} = e^{-iwx} |a|^{-1/2} \hat{\psi}\left(\frac{w}{a}\right)$, 则

$$\begin{aligned} \iint_{\mathbb{R}^2} W_f(a, b) |a|^{1/2} \psi(a(x-b)) da db &= \\ \frac{1}{2\pi} \int_{\mathbb{R}} \left(\int_{\mathbb{R}} \hat{f}(w) |a|^{-1/2} \overline{\hat{\psi}\left(\frac{w}{a}\right)} |a|^{-1/2} \hat{\psi}\left(\frac{w}{a}\right) e^{iwx} dw \right) da &= \\ \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(w) e^{iwx} \left(\int_{\mathbb{R}} \left| |a|^{-1/2} \hat{\psi}\left(\frac{w}{a}\right) \right|^2 da \right) dw &= \frac{C_\psi}{2\pi} \int_{\mathbb{R}} \hat{f}(w) e^{iwx} dw = C_\psi f(x) \end{aligned}$$

证毕

4.2.2 小波变换的多分辨率分析

1988年, Mallat 与 Meyer 合作提出了多分辨率 (Multi-resolution-analysis, 简称MRA) 的框架, 其主要思想是将 $L^2(R)$ 分解成一串具有不同分辨率的子空间序列, 该子空间序列的极限就是 $L^2(R)$, 然后将 $L^2(R)$ 中的 f 描述为具有一系列近似函数的逼近极限, 其中的每一个近似函数都是 f 的不同分辨率子空间上的投影。通过这些投影可以将分析和研究 f 在不同分辨率子空间上的状态和特征。

定义 4.3 设 $\{V_j\}_{j \in \mathbb{Z}}$ 是 $L^2(R)$ 的一串闭子空间序列, 如果满足一下五条, 则称 $\{V_j\}_{j \in \mathbb{Z}}$ 为 $L^2(R)$ 的一个 MRA。

(1) 单调性: $V_j \subset V_{j+1}$

(2) 平移不变性: 若

$$u(x) \in V_j \Leftrightarrow u(x-k) \in V_j$$

(3) 二进制伸缩相关性质: 若

$$u(x) \in V_j \Leftrightarrow u(2x) \in V_{j+1}, u(2^j x) \in V_{2j}$$

(4) 逼近性:

$$\bigcup_{j \in \mathbb{Z}} V_j = L^2(\mathbb{R}), \quad \bigcap_{j \in \mathbb{Z}} V_j = \{0\}$$

(5) Riesz 基的存在性: 存在 $g \in V_0$, 使 $\{g(x-k)\}_{k \in \mathbb{Z}}$ 是 V_0 的 Riesz 基。

从定义中的 (1) 可以看出, 闭子空间序列是一个包含一个, 由 (4) 可知它们的极限位置:

$$L^2(\mathbb{R}) \supset \dots \supset V_j \supset V_{j-1} \supset \dots \supset \{0\}$$

4.2.3 小波变形算法

4.2.3.1 小波分解和原形重构

多分辨率分析的框架, 其主要思想是将 $L^2(\mathbb{R})$ 分解为一串具有不同分辨率的子空间序列, 该子空间序列的极限就是 $L^2(\mathbb{R})$, 然后将 $L^2(\mathbb{R})$ 中的 f 描述为具有一系列近似函数的逼近极限, 其中每一个近似函数都是 f 在不同分辨率子空间上的投影。

通过小波变换, 一个 N 个顶点的多边形 P^N 能够被分解成一系列的轮廓多边形 (即多边形的轮廓) P^i ($i = N-1, \dots, 3$) (P^i 是拥有 i 个顶点的多边形) 和高频的细节信息 D^i ($i = N-1, \dots, 3$)。

要对多边形进行分解与重构处理, 首先定义如下算子。对于数据序列 $\{a_0, a_1, \dots, a_{n-1}, a_n\}$, 定义上采样符号 $\uparrow 2$ 为: $\{a_0, a_1, a_1, \dots, a_{n-1}, a_{n-1}, a_n\}$; $\uparrow 2$ 为: $\{a_0, 0, a_1, 0, \dots, 0, a_{n-1}, 0, a_n, 0\}$ 。对于数据序列 $\{a_0, a_1, \dots, a_{2n-1}, a_{2n}, a_{2n+1}\}$, 定义下采样符号 $\downarrow 2$ 为: $\left\{a_0, \frac{a_1+a_2}{2}, \frac{a_3+a_4}{2}, \dots, \frac{a_{2n-1}+a_{2n}}{2}, a_{2n+1}\right\}$, $\downarrow 2$ 为 $\{a_0, a_2, \dots, a_{2n}\}$ 。

在本文中, 继续采用哈尔小波函数, 因为哈尔小波函数具有正交性质和有限支撑的性质。且是比较容易的小波函数。之所以要求小波变换的基是正交的, 这样做是为了保证通过小波分解后的轮廓多边形更相似于原始多边形。这一点对于多边形变形 (特征融合) 是非常重要的, 这是因为当应用融合方法作用于相应的

轮廓多边形时，它可以使得得到的变形序列更加自然、流畅。

有限支撑集可以保证准确的重构。换句话说，当应用具有有限支撑集的小波变换函数作用于多边形，分解该多边形成轮廓多边形。当重构该多边形时，重构出的多边形能够和原始多边形是一模一样的。假设要应用小波变形方法在 n 步将一个给定的初始多边形变换到一个给定的目标多边形，则期望在第一步得到的结果和初始多边形是一样的，而在第 n 步得到的多边形和目标多边形是一样的，这就要求所用的小波基函数具有有限支撑集这个属性来保证重构的准确性。

哈尔尺度函数：

$$\phi_k^j(x) = 2^{-j/2} \phi(2^{-j}x - k) \quad \phi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{其他} \end{cases}$$

和哈尔小波基函数：

$$\varphi_k^j = 2^{-j/2} \varphi(2^{-j}x - k) \quad \varphi(x) = \begin{cases} 1 & 0 \leq x < 1/2 \\ -1 & 1/2 \leq x < 1 \\ 0 & \text{其他} \end{cases}$$

利用哈尔尺度函数和小波函数很容易得到滤波器系数 $\hat{h} = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ 和

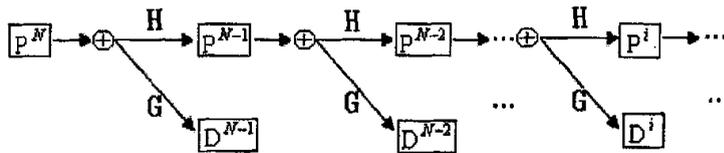
$\hat{g} = (\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$ 。定义算子 H 和 G 分别为与滤波器系数 \hat{h} 和 \hat{g} 的卷积：

$$H: (Ha)_k = \sum_n h_{n-k} a_n \quad G: (Ga)_k = \sum_n g_{n-k} a_n$$

从而有小波分解算子为：

$$H = [\downarrow 2] H [\uparrow 2] \quad G = [\downarrow 2] G [\uparrow 2]$$

有如下的分解示意图：



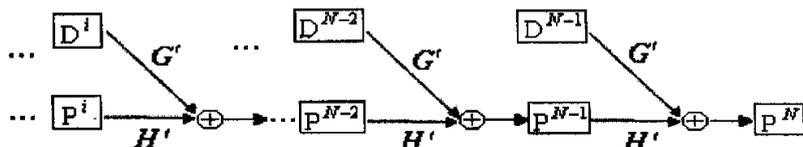
根据分解算子的构造，可以得到重构算子如下：

$$H' = [\uparrow 2] H [\downarrow 2] \quad G' = [\uparrow 2] G [\downarrow 2]$$

其中：

$$H': (H'a)_n = \sum_k h_{n-2k} a_{2k} \quad G': (G'a)_n = \sum_k g_{n-2k} a_{2k}$$

重构示意图如下：



按照本文的思想, 多边形顶点将分成两个点序列: 多边形顶点的 x 坐标组成的序列和 y 坐标组成的序列。上面的分解算法和重构算法都是分别对这两个点序列进行分解和重构的。

要对初末多边形进行同构三角剖分是十分重要的一步, 它直接影响到变形过程的复杂度。构造此算法是为了通过此算法得到多边形的轮廓。将对多边形的同构三角剖分转化为对多边形轮廓进行同构三角剖分。从而减少了多边形细节部分对同构剖分的影响。

4.2.3.2 变形算法

对平面图形进行同构三角剖分的方法比较多, 在文献[6、13、61]中给出了同构三角剖分的算法, 其中 Aronov 在文献[6]中提出了两种对多边形进行同构三角剖分的算法: 一种是“蜘蛛网”同构三角剖分算法; 另一种如下: 先分别在初始和目标图形上进行三角剖分, 将得到的三角剖分影射到用同样顶点数目构成的凸多边形上, 且进行迭加, 在该凸多边形上再进行三角剖分, 最后再把该凸多边形上的剖分分别影射到初始和目标图形上, 则得到两者的同构三角剖分。还有的算法[13、61]对方法二进行了改进, 在第三章详细介绍了相应的改进算法。详细方法可以参见第三章。本文所采用的剖分算法是第三章所介绍的剖分算法。

给定都由 n 个顶点组成的初始多边形 P_s 和目标多边形 P_t 后, 用分解算法对初末多边形进行分解, 得到结果 P_s^i 、 P_t^i (P_s^i 与 P_t^i 都是拥有 i 个顶点的多边形) ($i = n-1, \dots, 3$) 和相应的细节 X_s^i 、 X_t^i , 其中 ($i = n-1, \dots, 3$)。假如分解成最终的多边形轮廓的顶点数目为 k , 现在将 P_s^k 变形到 P_t^k : 先将初末多边形轮廓 P_s^k 与 P_t^k 分别嵌入到相同的凸网格中, 再用第三章中的同构三角剖分算法对它们进行同构三角剖分, 得到 T_s 和 T_t , 再采用凸组方法对它们进行变形。

从原始多边形变形到目标多边形的中间多边形的轮廓可以采用上面的算法,

对原始多边形的轮廓和目标多边形的轮廓进行处理得到。但是要将原始多边形自然、连续、光滑的变形到目标多边形，还必须同时对多边形的细节进行处理，因为多边形的细节信息对轮廓影响很小，从而本文采用线性插值原始多边形细节与目标多边形细节：

$$X^i(t) = (1-t)X_s^i + tX_t^i \quad i = k, k+1, \dots, n-1$$

其中 $X^i(t)(i = k, k+1, \dots, n-1)$ 是从原始多边形变形到目标多边形在 t 时刻的细节。

通过本文中描述的，可以得到从初始多边形变换到目标多边形在 t 时刻的中间多边形的轮廓 $P^k(t)$ 和细节部分 $X^i(t)(i = k, k+1, \dots, n-1)$ ，通过该轮廓和细节，可以采用重构方法重构出在 t 时刻的中间多边形。

4.2.3.3 多边形变形的步骤

该变形是先利用 4.2.3.2 中的分解算法，得到初末多边形的轮廓和细节。然后对多边形的轮廓和细节分别进行变形。该算法的变形可按如下步骤进行：

(1) 利用本文 4.2.3.1 节中的小波分解算法分别对初始多边形和目标多边形进行分解，分别得到对应的轮廓和细节部分。

(2) 利用第三章给出的算法，对初始多边形的轮廓和目标多边形的轮廓进行同构三角剖分，然后再利用凸组合方法将初始多边形的轮廓变换到目标多边形的轮廓。

(3) 对多边形的细节部分，因为对轮廓影响不大，所以采用线性插值。

(4) 利用 4.2.3.2 节中的原形重构算法对用第(2)步中得到的结果和第(3)步中得到的结果进行重构，从而得到原始多边形变形的中间多边形。

4.3 算法实例

本章引用了文献[61]中的实例，对原始多边形进行同构三角剖分，需要添加较多的额外顶点（如图 4-10，其中用圆圈标出了所使用的额外顶点），采用本章算法对原始多边形采用小波分解后，得到剩余 35 个顶点的多边形轮廓，在此多边形轮廓上进行同构三角剖分，所需要的额外顶点明显减少（如图 4-11，在此多边形轮

廓中，不需要添加额外顶点就能得到初末多边形轮廓的同构三角网格)，额外顶点的较少，使得变形速度也得到了提高。在 Inter (R) Pentium (R) 4, CPU 3.00GHz, 2.99GHz, 504MB 的内存微机上用 Matlab7.0.1 对原始多边形采用嵌入凸网格的凸

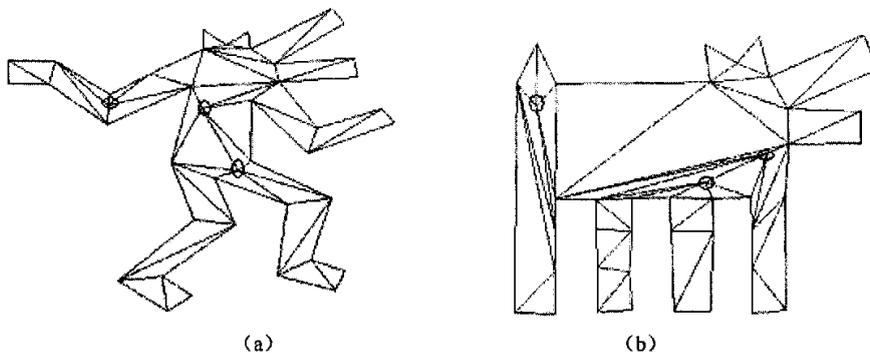


图 4-10 初末多边形必须增加额外的顶点才能得到两者的同构三角剖分，其中用圈标出了额外顶点

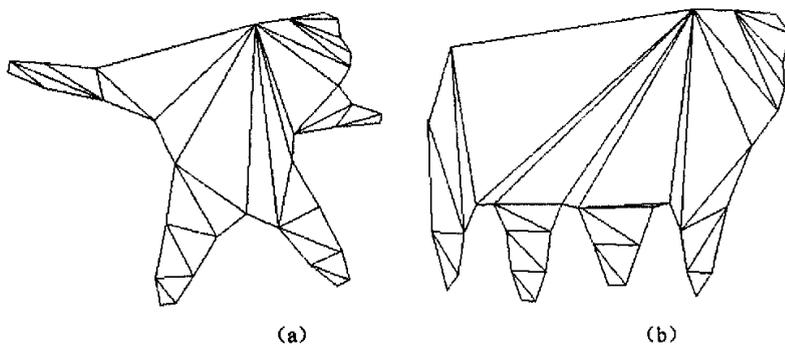


图 4-11 利用小波分解算法，分别对初末多边形进行分解，得到剩余 35 个顶点的初末多边形的轮廓，在不需要增加额外顶点的情况下能够得到同构三角剖分

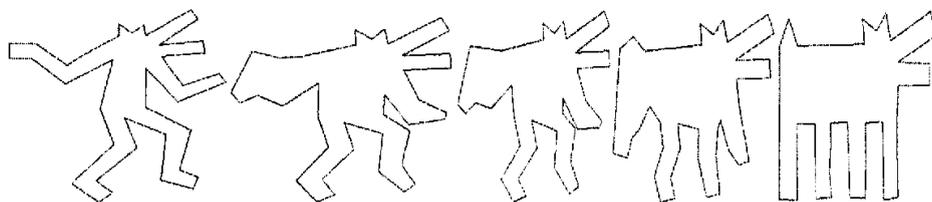


图 4-12 在原始多边形上进行同构三角剖分，再利用变形算法得到的变形序列

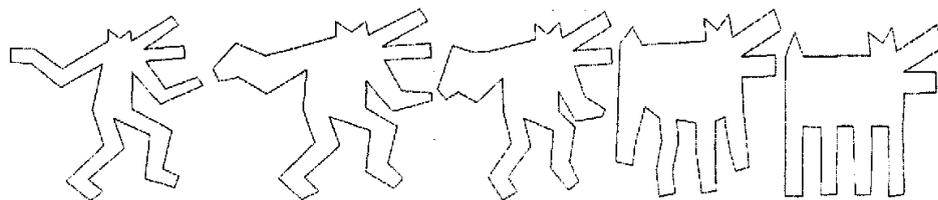


图 4-13 利用本文给出的方法得到的变形序列，该方法能够得到和图 4-11 一致的变形效果，但是在变形计算量方面得到了大大的改善

组合变形，得到变形的中间序列（如图 4-12）需要 12.5940 秒。采用本章算法，能够得到很好的效果（如图 4-13），且只需要 10.7460 秒。

在原始多边形上进行同构三角剖分，需要增加 $O(n^2)$ 的额外顶点。用小波分解时，每分解一次就相应的减少一个顶点，假设选择了适当的分解层次 k ，则剩余 $n-k$ 个顶点，而且小波分解时，去除了多边形细节对剖分的影响，从而使得增加的额外顶点数不会超过 $O((n-k)^2)$ 。从而降低了计算量。

4.4 本章小结

在本文中，给出了一种新的基于小波的多边形变形算法，这种算法能够将原始多边形自然、连续、光滑的变形到目标多边形。

利用本文算法，先利用小波分解将给定的原始多边形和目标多边形分别分解成相应的轮廓和细节。再通过分别对轮廓和细节进行变形，用得到的中间轮廓和中间细节进而重构出中间多边形。首先，利用分解算法，得到了多边形的轮廓，减少了多边形细节对同构三角剖分的影响，从而减轻了同构三角剖分以及后面工作的计算量，提高了变形速度。

在变形过程中，中间状态要保持初始状态和目标状态的视觉特征。且变形过程中产生的中间状态的一些特征，如边长、夹角、面积等应保持单调平滑的变换。当给定两个边界为凸的同构凸网格，对其变形时，期望得到的中间网格的边界也是凸的，在第五章将对此进行研究。

第五章 平面凸网格的保凸变形研究

对平面多边形变形研究已经比较成熟,其中能保证多边形在变形过程中不自交的算法是基于凸组合的变形算法。当初末多边形具有相同的凸边界时,可直接采用凸组合变形算法,即可得到避免自交的变形算法。当初末多边形不是凸多边形或者初末多边形的凸边界不不同时,Gotsman 和 Surazhsky 在文献[60]中给出了可避免自交的平面简单多边形变形方法,即将初末多边形分别嵌入两个具有相同凸边界的同构平面三角网格中。后 Floater MS 和 Gotsman[19]、宋伟杰[2]又提出嵌入凸多边形的凸组合变形算法,对凸边界进行保凸变形算法,对内部顶点采用凸组合的变形方法,该方法有利于对凸多边形的要求降低,可减少额外顶点的产生,减少计算量。因此研究平面凸网格的保凸变形具有十分重要的意义。

5.1 相同凸边界的同构平面三角网格变形

任意一个平面三角网格,采用凸组合映射方法可以构造出与之同构且边界为凸多边形的平面三角网格。以此为理论依据,文献[19、63、64]等对相同凸边界的同构平面三角网格的变形进行了研究,其中 Floater 和 Gotsman[19]的研究是开创性的,Gotsman 和 Surazhsky[63、64]则在此基础上,对这种变形的可控制性和内在性质的保留等进行了一些研究工作。

定义 5.1 (凸组合表示) 设 $v_0, v_1, \dots, v_k (k \geq 3)$ 是平面上 $k+1$ 个点,且 v_1, v_2, \dots, v_k 将点 v_0 围在中心,如果存在 $\lambda_1, \lambda_2, \dots, \lambda_k$ 满足条件:

$$\lambda_1, \lambda_2, \dots, \lambda_k > 0, \quad \sum_{i=1}^k \lambda_i = 1 \quad (5.1)$$

$$\sum_{i=1}^k \lambda_i v_i = v_0 \quad (5.2)$$

则称 v_0 可通过 v_1, v_2, \dots, v_k 凸组合表示,称 $\lambda_1, \lambda_2, \dots, \lambda_k$ 为凸组合系数。

Floater 和 Gotsman[19]的变形方法也称为凸组合变形。他们研究了 tiling 的变形,由于本章主要研究平面三角网格,故这里仅介绍平面三角网格的凸组合变

形。设 T^0, T^1 为具有相同凸边界的同构平面三角网格, $\lambda_{i,j}^k$ 为 T^k 的内顶点 p_i^k 相对于其相邻顶点 p_j^k 的严格为正的凸组合系数, $k=0,1, i=1,2,\dots,n, j=1,2,\dots,N$, $\lambda_{i,j}^k$ 显然满足 (5.1) 和 (5.2)。在变形过程中, 网格边界保持不变, 中间网格内顶点的位置确定如下: 对于任意 $t \in [0,1]$ 时刻, 令

$$\lambda_{i,j}(t) = (1-t)\lambda_{i,j}^0 + t\lambda_{i,j}^1, \quad (5.3)$$

显然 $\lambda_{i,j}(t)$ 满足 (5.1), 则下列线性方程组的解就为 t 时刻中间网格内顶点 $p_i(t), i=1,2,\dots,n$ 的坐标值:

$$\sum_{j=1}^N \lambda_{i,j}(t) \cdot p_j(t) = p_i(t), \quad i=1,2,\dots,n \quad (5.4)$$

如果采用相邻矩阵来描述上述凸组合变形, 则形式更为简洁[63]。设 A_0, A_1 分别为上述 T^0, T^1 的相邻矩阵, 令 $A(t) = (1-t)A_0 + tA_1, t \in [0,1]$, 解方程组 $A(t) \cdot x(t) = x(t), A(t) \cdot y(t) = y(t)$, 便得到了中间网格的内顶点, 而网格边界同样保持不变。

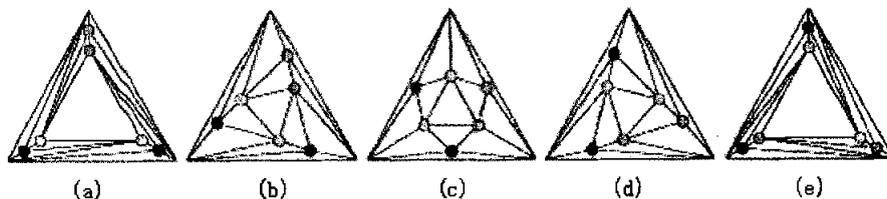


图 5-1 对具有相同凸边界的同构平面三角网格进行凸组合变形。(a) 初始三角网格, (e) 目标三角网格, 顶点对应关系采用不同的颜色来表示, (b) - (d) 中间三角网格。

采用凸组合变形方法对两个具有相同凸边界的同构平面三角网格进行变形时, 能够保证任意时刻的中间平面三角网格始终与初末网格同构, 即不产生自交现象, 且变形过程连续光滑 (如图 5-1)。这是由于 $\lambda_{i,j}(t)$ 关于 t 是连续光滑的, 故线性方程组 (5.4) 的解即网格的内顶点 $p_i(t), i=1,2,\dots,n$ 关于 t 也是连续光滑的, 而任意 $t \in [0,1]$ 时刻网格边界的顶点不变, 即 $p_i(t) = p_i^0 = p_i^1, i=n+1, n+2, \dots, N$,

所以该变形过程连续光滑；又 $\lambda_{j,j}(t)$ 满足 (5.1)，且中间网格边界保持不变，为凸多边形，任意时刻的中间状态为与初末网格同构的平面三角网格，不会产生自交。

在运用凸组合变形方法时，初末网格内顶点的凸组合系数是需要事先求出的。设 $P = (p_1, p_2, \dots, p_d)$ 为一星形多边形，点 p 为 P 的核内部的一点， p, p_1, p_2, \dots, p_d 互不相同。我们希望能够找到一组严格为正的数 $\lambda_1, \lambda_2, \dots, \lambda_d$ ，满足

$$\sum_{j=1}^d \lambda_j p_j = p, \quad \sum_{j=1}^d \lambda_j = 1 \quad (5.5)$$

如果 $d=3$ ，则 p 关于 $\Delta p_1 p_2 p_3$ 的凸组合系数 $\lambda_1, \lambda_2, \lambda_3$ 是唯一确定的，且

$$\lambda_1 = \frac{\text{area}(p, p_2, p_3)}{\text{area}(p_1, p_2, p_3)}, \quad \lambda_2 = \frac{\text{area}(p_1, p, p_3)}{\text{area}(p_1, p_2, p_3)}, \quad \lambda_3 = \frac{\text{area}(p_1, p_2, p)}{\text{area}(p_1, p_2, p_3)} \quad (5.6)$$

如果 $d > 3$ ，则 $\lambda_1, \lambda_2, \dots, \lambda_d$ 不唯一。

Floater 和 Gotsman 在 [19] 中采用了 [18] 的方法来求得内顶点的凸组合系数。如图 5-2，对每一个 $k=1, 2, \dots, d$ ，连接 p_k 和 p 并延长，与 P 的一边相交于点 q_k ，显然点 q_k 是唯一存在的，且唯一存在下标 i ，使得 q_k 在 P 的边

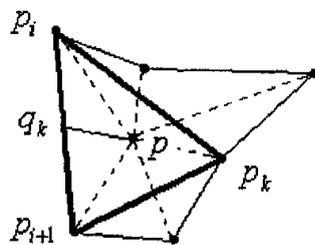


图 5-2

$[p_i, p_{i+1}]$ (当 $i=d$ 时，取 $i+1$ 为 1) 上， $q_k \neq p_{i+1}$ ，设 τ_1, τ_2, τ_3 为 p 关于 $\Delta p_k p_i p_{i+1}$ 的凸组合系数，则有 $p = \tau_1 p_k + \tau_2 p_i + \tau_3 p_{i+1}$ ，其中 $\tau_1, \tau_2 > 0$ ， $\tau_3 \geq 0$ ， $\tau_1 + \tau_2 + \tau_3 = 1$ 。令 $\mu_{k,k} = \tau_1$ ， $\mu_{i,k} = \tau_2$ ， $\mu_{i+1,k} = \tau_3$ ，对于剩余的 $j \in \{1, 2, \dots, d\}$ ，令 $\mu_{j,k} = 0$ ，故对于每一个 $k=1, 2, \dots, d$ ，有

$$\sum_{j=1}^d \mu_{j,k} p_j = p, \quad \sum_{j=1}^d \mu_{j,k} = 1, \quad \mu_{j,k} \geq 0$$

令
$$\lambda_j = \frac{1}{d} \sum_{k=1}^d \mu_{j,k} \quad (5.7)$$

由于 $\lambda_j \geq \mu_{j,j} / d > 0$ ，故 $\lambda_1, \lambda_2, \dots, \lambda_d$ 严格为正，且可以证明满足条件 (5.5)。容易

看出, 公式 (5.7) 包含了 $d=3$ 时的情况。

上述[18]的方法所求得的凸组合系数 λ_j 关于 p 和 p_1, p_2, \dots, p_d 是连续的, 但不是光滑的, 在进行凸组合变形时, 为了达到自然光滑的变形效果, 有必要提高其连续阶。但是, 寻找关于 p 和 p_1, p_2, \dots, p_d 连续光滑且严格为正的凸组合系数一直是一个富有挑战性的问题, [18、63]等方法均不能完全满足上述要求, 目前只有 Floater[37]解决了这一问题, 给出了求凸组合系数的如下公式:

$$\lambda_i = \frac{w_i}{\sum_{j=1}^k w_j}, \quad w_i = \frac{\tan(\alpha_i/2) + \tan(\alpha_{i-1}/2)}{l_i} \quad (5.8)$$

如图 4-5 所示, $\alpha_i = \angle p_i p p_{i+1}$, $l_i = \|p - p_i\|$ 。由于 α_i 为 $\Delta p p_i p_{i+1}$ 的一个内角, 故有 $0 < \alpha_i < \pi$, 容易看出, 由 (5.8) 求得的凸组合系数 $\lambda_i > 0$ 。(5.8) 定义的凸组合系数称为均值坐标 (mean value coordinates), 它关于 p 和 p_1, p_2, \dots, p_d 是连续光滑的。

在 Floater 和 Gotsman[19]的凸组合变形基础上, Gotsman 和 Surazhsky[63]研究了其控制性 (其中采用至少 C^1 连续的凸组合系数), 提出了两种 linear-reducible 方案 (一种为全局的, 另一种为局部的) 和一种使网格中的三角形面积呈自然变化的方案, 并且研究了如何寻找一光滑变形, 使得在给定的时刻插值给定的一个三角网格。后来他们研究了同构三角网格的内在变形[64], 其中凸组合系数采用均值坐标 (5.8), 通过线性插值均值坐标的元素即角度和边长, 而不是直接线性插值均值坐标, 局部的保留了初末网格的内在性质, 取得了更加光滑自然的变形效果。

5.2 同构平面三角网格的保凸变形

在 5.1 节中介绍了 Floater 和 Gotsman[19]的凸组合变形方法, 该方法对具有相同凸边界的同构平面三角网格进行变形时, 网格边界保持不变, 所得的中间平面三角网格始终与初末网格同构。事实上, 由 Floater[18]可以得出结论 (*): 对边界为凸多边形的同构平面三角网格进行凸组合变形时, 只要使得中间网格的边界始终为凸多边形, 便可以保证中间网格始终与初末网格同构, 即不产生自交现象。本文正是以此为理论依据, 设法寻找出一种使得两个凸多边形在变形过程中始终

保持其凸性的方法，从而实现了具有不同凸边界的同构平面三角网格的变形，本文将该方法称为平面凸网格的保凸变形。

Floater 和 Gotsman 在[19]中也研究了不同凸边界情形下的网格变形，作者采用了 Shapira 和 Rappoport[42]的基于极坐标的方法对网格边界进行变形，网格的内顶点则仍采用其凸组合变形方法。设初末网格边界的顶点按逆时针方向分别为 $u_1^k, u_2^k, \dots, u_n^k$, $k = 0, 1$ ，设边界多边形 $(u_1^k, u_2^k, \dots, u_n^k)$ 的重心为 u_0^k , $k = 0, 1$ ，此时 $u_1^k, u_2^k, \dots, u_n^k$ 用极坐标表示为： $u_i^k = u_0^k + r_i^k (\cos \theta_i^k, \sin \theta_i^k)$, $k = 0, 1$ ，其中 $0 \leq \theta_1^k < 2\pi$ ，且 $\theta_1^k < \theta_2^k < \dots < \theta_n^k < \theta_1^k + 2\pi$ 。对于任一 $t \in [0, 1]$ ，令

$$u_0(t) = (1-t)u_0^0 + tu_0^1, \quad r_i(t) = (1-t)r_i^0 + tr_i^1, \quad \theta_i(t) = (1-t)\theta_i^0 + t\theta_i^1$$

则网格边界变形定义为： $u_i(t) = u_0(t) + r_i(t)(\cos \theta_i(t), \sin \theta_i(t))$, $i = 1, 2, \dots, n$ 。

Floater 和 Gotsman[19]的这种基于极坐标的方法，尽管在其文章的所有例子中都能保持网格边界的凸性，但是，由于该方法是基于极坐标的，容易证明它只能保证中间网格边界的星形形状，而无法保证其凸性，对此宋伟杰举出一反例，说明了该方法对于两个凸多边形的变形无法保证中间多边形始终为凸多边形（如图 5-3）。由此看出，该方法对于不同凸边界情形下的网格变形不符合（*），因而无法保证中间网格始终与初末网格同构，有时会导致中间网格出现自交现象。

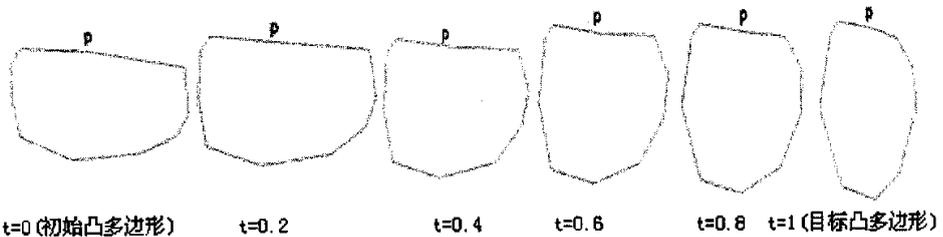


图 5-3 采用[19]的基于极坐标的方法对两个凸多边形的变形。最左边为初始凸多边形，最右边为目标凸多边形，其顶点 p 当 $t = 0.4, 0.6, 0.8$ 时变为凹点。

宋伟杰提出了一个保凸的变形算法，该方法结合了内在解方法[1]和凸组合方法[19]，其中内在解方法用来变形网格的凸边界，在此基础上，由凸组合方法确定出中间网格的内顶点的位置。该方法较 Floater 和 Gotsman[19]有明显的改善效果。由于[1]本身的复杂度很低，故宋伟杰方法的复杂度也比较低。下面是内在解算法

保凸证明:

定理 5.1 给定初末两个凸多边形, 假设其顶点的对应关系已经建立, 对其采用内在解算法进行变形, 则:

- (1)、任意时刻的中间多边形为简单多边形, 即边之间除顶点外不相交;
- (2)、任意时刻的中间多边形为凸多边形。

证明 设 T^0 、 T^1 分别为初末凸多边形, $T^k = [p_0^k p_1^k \cdots p_n^k]$, 其中 $p_n^k = p_0^k$, $k = 0, 1$, 顶点的对应关系为: $p_i^0 \mapsto p_i^1$, $i = 0, 1, \dots, n$, 且顶点按逆时针方向排列。设 $p_{i-2}^k p_{i-1}^k$ 到 $p_{i-1}^k p_i^k$ 的有向转角为 θ_i^k , $i = 2, 3, \dots, n+1$, 这里 $p_{n+1}^k = p_1^k$, 规定 θ_i^k 为逆时针方向时取正值, 否则取负值。由于凸多边形的所有顶点均为凸点, 所以有 $0 \leq \theta_i^k < \pi$, $k = 0, 1$, $i = 2, 3, \dots, n+1$ 。

设 t 时刻的中间多边形 $T^t = [p'_0 p'_1 \cdots p'_n]$ 。由内在解算法可知, $p'_{i-2} p'_{i-1}$ 到 $p'_{i-1} p'_i$ 的有向转角为 $\theta'_i = (1-t)\theta_i^0 + t\theta_i^1$, $t \in [0, 1]$, $i = 2, 3, \dots, n$, 设 $p'_{n-1} p'_0$ 到 $p'_0 p'_1$ 的有向转角为 θ'_{n+1} (由于该算法对中间多边形的边长 p'_i 的调整, 保证了中间多边形 T^t 的封闭性, 即有 $p'_n = p'_0$)。

注意: 在任意多边形 $T = [p_0 p_1 \cdots p_n]$, $p_n = p_0$ 中, 若顶点按逆时针方向排列, 则 $p_{i-2} p_{i-1}$ 到 $p_{i-1} p_i$ 的有向转角 θ_i 恰为内角 $\angle p_{i-2} p_{i-1} p_i$ 的外角。

- (1)、反证: 假设某 t_0 时刻的中间多边形 T^{t_0} 不是简单多边形, 即产生自交。

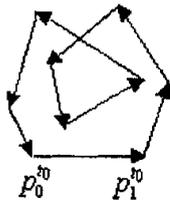


图 5-4 T^{t_0} 的相交情形为“卷形”, 这里“卷”了两圈

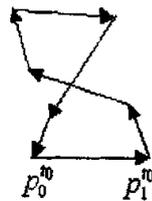


图 5-5 T^{t_0} 的相交情形为“8形”或含有“8形”, 这里为“8形”

由于 $\theta'_i = (1-t_0)\theta_i^0 + t_0\theta_i^1 \geq 0$, $i = 2, 3, \dots, n$, 故 T^{t_0} 的相交情形只可能为“卷形” (如图 5-4), 而不可能是“8形”或含有“8形” (如图 5-5)。这里把多边形 T^{t_0}

看作是初始边向量 $p_0^0 p_1^0$ 旋转有向角度 θ_2^0 ，适当调整其长度后与边向量 $p_1^0 p_2^0$ 重合，然后再旋转有向角度 θ_3^0 ，调整长度后与边向量 $p_2^0 p_3^0$ 重合，依次类推，直到旋转有向角度 θ_{n+1}^0 ，调整长度后回到原位置。由于 $\theta_i^0 \geq 0$ ， $i = 2, 3, \dots, n$ ，故初始边向量总是向逆时针方向旋转，直到旋转 θ_{n+1}^0 (θ_{n+1}^0 的正负现在未知) 后回到原位置，所以 T^0 的相交情形只可能为“卷形”。

设 T^0 “卷”了 n 圈 ($n \geq 2$)，由于多边形的外角和可以看作其初始边向量所转过的有向角度之和，而该向量最终要回到原位置，故此时 T^0 的外角和为 $2n\pi$ ，有

$$2n\pi = \sum_{i=2}^{n+1} \theta_i^0 = \sum_{i=2}^n \theta_i^0 + \theta_{n+1}^0$$

$$\text{而 } \sum_{i=2}^n \theta_i^0 = \sum_{i=2}^n [(1-t_0)\theta_i^0 + t_0\theta_i^1]$$

$$= (1-t_0) \sum_{i=2}^n \theta_i^0 + t_0 \sum_{i=2}^n \theta_i^1$$

$$= (1-t_0)(2\pi - \theta_{n+1}^0) + t_0(2\pi - \theta_{n+1}^1)$$

$$\because \theta_{n+1}^0, \theta_{n+1}^1 \in [0, \pi)$$

$$\therefore (1-t_0)(2\pi - \theta_{n+1}^0) + t_0(2\pi - \theta_{n+1}^1) \in (\pi, 2\pi]$$

$$\therefore \theta_{n+1}^0 = 2n\pi - [(1-t_0)(2\pi - \theta_{n+1}^0) + t_0(2\pi - \theta_{n+1}^1)] \in [2(n-1)\pi, (2n-1)\pi)$$

此时，由于 $n \geq 2$ ，且 θ_{n+1}^0 是外角，而外角的取值范围为 $(-\pi, \pi)$ ，故与上式矛盾。

所以任意时刻的中间多边形为简单多边形。

(2)、由 (1) 结论知，任意 t 时刻的中间多边形为简单多边形，故其外角和为 2π 。所以有

$$2\pi = \sum_{i=2}^{n+1} \theta_i^t = \sum_{i=2}^n \theta_i^t + \theta_{n+1}^t$$

类似于 (1) 中可得

$$\theta'_{n+1} = 2\pi - [(1-t)(2\pi - \theta^0_{n+1}) + t(2\pi - \theta^1_{n+1})] \in [0, \pi)$$

而对于 $i = 2, 3, \dots, n$, 由于 $\theta^k \in [0, \pi)$, $k = 0, 1$, 故有 $\theta'_i = (1-t)\theta^0_i + t\theta^1_i \in [0, \pi)$ 。

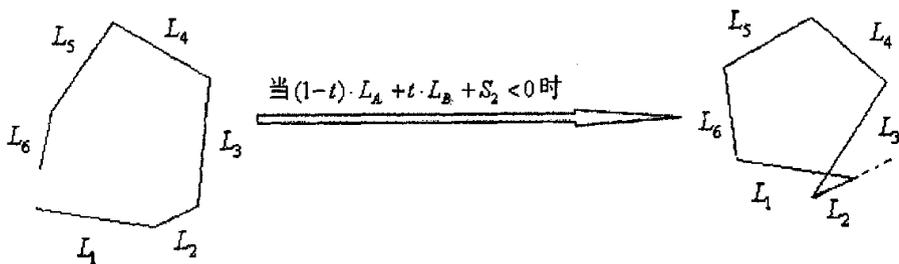
综上所述, 对于 $i = 2, 3, \dots, n+1$, 都有 $\theta'_i \in [0, \pi)$, 即 t 时刻中间多边形的所有顶点都为凸点, 所以任意 t 时刻的中间多边形为凸多边形。

证毕

该证明在绝大部分情况下都是成立的。首先凸多边形应该是一个封闭的多边形, 则在用内在解算法变形凸多边形时, 为了保证变形过程中的多边形是一个封闭的多边形, 从而需要给每一条边增加一定的补偿常量, 即:

$$L_i = (1-t) \cdot L_A + t \cdot L_B + S_i, \quad (i = 0, 1, \dots, m) \quad (5.9)$$

对补偿常量 S_i 的求解可参看文献[57]。(5.9) 中 S_i 的值只满足文献[57]中的一个约束条件: S_i 与伸缩比例的平方和最小。根据内在解变形算法, 当出现 $L_i = (1-t) \cdot L_A + t \cdot L_B + S_i < 0$ 时, 变形的中间序列就会出现自交, 如图 5-6 所示:



(a) 未经调整的 t 时刻的多边形不封闭

(b) 经调整后的 t 时刻凸多边形

图 5-6 不封闭的多边形(a)经调整后产生自交的多边形

虽然出现图 5-6 所示的自交情况的可能性较小, 但是还是有可能。为了杜绝这种可能性的存在, 下面本文给出了初末凸边界的保凸变形算法, 并给出了理论上的证明, 对于内部顶点仍采用凸组合法。

5.3 基于角度的保凸变形算法

该方法是利用三角形的一条边及与该边相连的两个内角确定三角形的第三个

顶点。为了说明的需要，本文用实线将凸多边形划分成多个三角形。

算法 5.1 由 n 个顶点 $p_i^k (i=1,2,\dots,n, k=0,1)$ 构成的初始和目标凸多边形 P^k ，可分别连接顶点 p_1^k 与 $p_i^k (i=2,3,\dots,n-1)$ ，从而构成 $n-2$ 个三角面 $\Delta p_1^k p_2^k p_3^k$ ， $\Delta p_1^k p_3^k p_4^k, \dots, \Delta p_1^k p_{n-1}^k p_n^k$ ，如图 5-7 (a) 和图 5-7 (c) 所示。

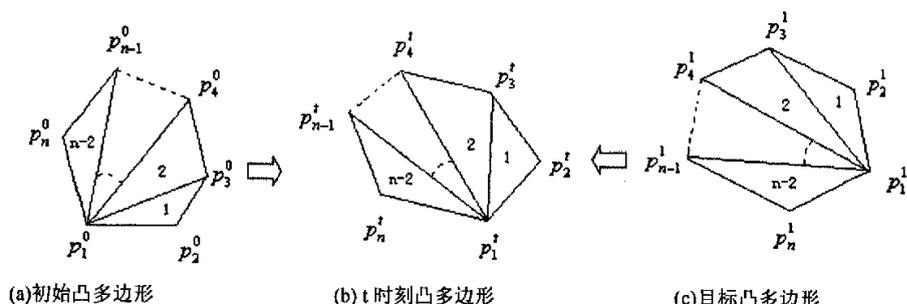


图 5-7

采用算法 5.1 时产生了三角网格，在本文中采用 $\Lambda = (Tes, V)$ 表示三角网格，三角网格中的每一个三角形称为三角网格的一个面；其中 V 表示三角网格各顶点坐标； Tes 为一 $(n-2) \times 3$ 的矩阵，称之为网格邻接关系矩阵，其中 $n-2$ 为该三角网格所包含的三角形数目， Tes 每一行表示一个面，其中的三个元素为构成该面的三个顶点的编号； $Tes(m, j)$ 表示 Tes 的第 m 行第 j 列的元素； V_m 表示第 m 个顶点； T_m 表示第 m 个面，其三个顶点编号即 Tes 的第 m 行的三个元素。

定义 5.2 给定平面三角网格 $\Lambda = (Tes, V)$ ，令 $\theta_{m,j}$ 为 T_m 中顶点为 $V_{Tes(m,j)}$ 的角 $m=1,2,\dots,n-2, j=1,2,3$ ，则称集合 $\Omega = \{\theta_{m,j}\}_{m=1, j=1}^{n-2, 3}$ 为三角网格 $\Lambda = (Tes, V)$ 的角度集， $\theta_{m,j} (m=1,2,\dots,n-2, j=1,2,3)$ 称为角度变量。显然，任意给定一平面网格 Λ ，可唯一的确定一角度集 Ω ；反之，若给定一角度集 $\Omega = \{\theta_{m,j}\}_{m=1, j=1}^{n-2, 3}$ ，为重建三角网格，需要固定一条三角网格的一条边或两个初始顶点，不妨设为 $V_{Tes(1,1)}, V_{Tes(1,2)}$ ，这样可从由 $\{V_{Tes(1,1)}, V_{Tes(1,2)}, \theta_{1,1}, \theta_{1,2}, \theta_{1,3}\}$ 确定的三个点开始，逐步求得 Λ 的其他顶点，具体的说就是先求得第一个面 T_1 的第三个顶点 $V_{Tes(1,3)}$ ，由 $\{\theta_{m,j}\}$ 及 $\{V_{Tes(1,1)}, V_{Tes(1,2)}, V_{Tes(1,3)}\}$ ，通过向量和三角函数运算进行递推，可求得其他面

$T_m (m \neq 1)$, 下述算法 5.2 详细的描述了上述的网格重建过程。

算法 5.2 给定角度集 $\Omega = \{\theta_{m,j}\}_{m=1, j=1}^{n-2, 3}$, 网格邻接关系矩阵 Tes , 设初始边的两顶点为 $V_{Tes(1,1)}, V_{Tes(1,2)}$, 则三角网格 $\Lambda = (Tes, V)$ 可由以下唯一确定:

步骤 1. 由 $\{V_{Tes(1,1)}, V_{Tes(1,2)}, \theta_{1,1}, \theta_{1,2}, \theta_{1,3}\}$ 确定 $V_{Tes(1,3)}$, 使得

$\angle V_{Tes(1,3)}V_{Tes(1,1)}V_{Tes(1,2)} = \theta_{1,1}$ $\angle V_{Tes(1,1)}V_{Tes(1,2)}V_{Tes(1,3)} = \theta_{1,2}$ 的 $V_{Tes(1,3)}$ 由下式确定:

$$V_{Tes(1,3)} = V_{Tes(1,1)} + (\bar{e}_1 \cot \theta_{1,1} + \bar{e}_2) \frac{\|V_{Tes(1,2)} - V_{Tes(1,1)}\|}{\cot \theta_{1,1} + \cot \theta_{1,2}}$$

其中 $\bar{e}_1 = \frac{V_{Tes(1,2)} - V_{Tes(1,1)}}{\|V_{Tes(1,2)} - V_{Tes(1,1)}\|}$, $\bar{e}_2 = \pm \bar{e}_1 \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, 正负号按如下规则取, 当 $V_{Tes(1,1)},$

$V_{Tes(1,2)}, V_{Tes(1,3)}$ 按逆时针排列时取 "+", 否则取 "-" (如图 5-8 所示)。

步骤 2. 若 $\Lambda = (Tes, V)$ 仍有未知的网格顶点, 不妨设 T_m 仅有一个未知点, 且未知点为 $V_{Tes(m,3)}$, 则使得

$\angle V_{Tes(m,3)}V_{Tes(m,1)}V_{Tes(m,2)} = \theta_{m,1}$ $\angle V_{Tes(m,1)}V_{Tes(m,2)}V_{Tes(m,3)} = \theta_{m,2}$ 的 $V_{Tes(m,3)}$ 由下

式确定:

$$V_{Tes(m,3)} = V_{Tes(m,1)} + (\bar{e}_1 \cot \theta_{m,1} + \bar{e}_2) \frac{\|V_{Tes(m,2)} - V_{Tes(m,1)}\|}{\cot \theta_{m,1} + \cot \theta_{m,2}}$$

其中 $\bar{e}_1 = \frac{V_{Tes(m,2)} - V_{Tes(m,1)}}{\|V_{Tes(m,2)} - V_{Tes(m,1)}\|}$, $\bar{e}_2 = \pm \bar{e}_1 \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ 正负号按如下规则取, 当 $V_{Tes(m,1)},$

$V_{Tes(m,2)}, V_{Tes(m,3)}$ 按逆时针排列时取 "+", 否则取 "-"。

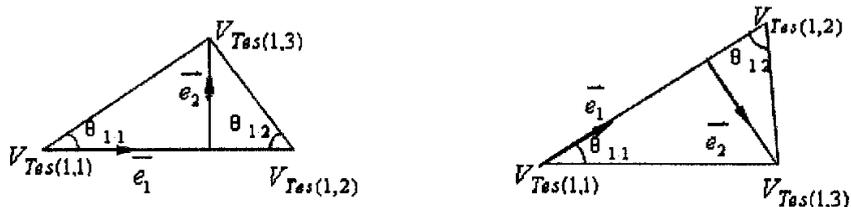


图 5-8 左图为取 "+" 的情况, 右图为取 "-" 的情况

这样，三角网格中的任意顶点均可由与其处于同一个面的其他两个已知点求出，如此循环下去，可重建三角网格。则整个初末凸多边形的保凸变形算法如下：

算法 5.3 给定 $t=0$ 时刻的初始凸多边形和 $t=1$ 时刻的目标凸多边形 P^0 和 P^1 ，则 $t(0 < t < 1)$ 时刻的凸多边形由以下步骤求得：

步骤 1. 采用算法 5.1，建立起初末凸多边形的三角网格 $\Lambda^0 = (T_{es}, V^0)$ 和 $\Lambda^1 = (T_{es}, V^1)$ 。

步骤 2. 计算三角网格 Λ^0 和 Λ^1 的角度集 Ω^0 和 Ω^1

$$\Omega^0 = \{\theta_{m,j}^0\} \quad \Omega^1 = \{\theta_{m,j}^1\}。$$

步骤 3. 由以下各式计算 t 时刻的三角网格 Λ^t 的角度集 $\Omega^t = \{\theta_{m,j}^t\}$

$$\theta_{m,j}^t = (1-f(t))\theta_{m,j}^0 + f(t)\theta_{m,j}^1$$

其中 $f(t)$ 为满足 $f(0)=0$ ， $f(1)=1$ 的单调递增函数，为简便，取 $f(t)=t$ 。

步骤 4. 由 $\Omega^t = \{\theta_{m,j}^t\}$ 和初始两顶点 $V_{T_{es}(1,1)}^t$ ， $V_{T_{es}(1,2)}^t$ ，应用算法 5.2，重建中间过渡网格 Λ^t 。

步骤 5. 由 Λ^t 按步骤 1 相反的过程得到中间过渡凸多边形 P^t 。

整个凸网格的凸变形可描述如下。

算法 5.4 给定 $t=0$ 时刻的初始凸网格和 $t=1$ 时刻的目标凸网格 P^0 和 P^1 ，则 $t(0 < t < 1)$ 时刻的网格由以下步骤求得：

步骤 1. 提取凸网格的凸边界，计算内部各顶点个凸组合系数。

步骤 2. 对凸边界采用算法 5.3 得到 $t(0 < t < 1)$ 时刻的凸边界，线性插值内部顶点的凸组合系数。

步骤 3. 通过步骤 2，可以得到凸边界顶点坐标以及内部顶点坐标，从而 $t(0 < t < 1)$ 时刻的凸网格确定。

5.4 基于角度的保凸变形算法证明

为了证明的需要,不妨假设各个面的邻接关系 Tes 如图 5-7 所示,其中 Tes 每行的元素排列为从顶点 1 开始,按逆时针方向排列。比如 Tes 第一行的元素为 $\{1,2,3\}$ 。

由算法过程可以看出,要证明基于角度的保凸变形能够保证 t 时刻的边界是凸的,只需要证明如下问题:

- (1) $0 < \theta'_{1,2} < \pi$; (2) $0 < \angle V'_{Tes(m,2)} V'_{Tes(m,3)} V'_{Tes(m+1,2)} < \pi$ ($1 \leq m < n-2$); (3) $0 < \theta'_{n-2,3} < \pi$;
 (4) $0 < \angle V'_{Tes(1,2)} V'_{Tes(n-2,1)} V'_{Tes(n-2,3)} < \pi$ 。

定理 5.2 采用算法 5.3 得到凸网格边界是一个凸的多边形。

证明 (1) $0 < \theta'_{1,2} < \pi$

\because 初末多边形都是凸多边形,且 $f(t)$ 为满足 $f(0)=0$, $f(1)=1$ 的单调递增函数

$$\therefore 0 < (1-f(t))\theta^0_{1,2} + f(t)\theta^1_{1,2} < \pi$$

$$\text{即: } 0 < \theta'_{1,2} = (1-f(t))\theta^0_{1,2} + f(t)\theta^1_{1,2} < \pi$$

$$(2) 0 < \angle V'_{Tes(m,2)} V'_{Tes(m,3)} V'_{Tes(m+1,2)} < \pi \quad (1 \leq m < n-2)$$

$$\begin{cases} \angle V'_{Tes(m,2)} V'_{Tes(m,3)} V'_{Tes(m+1,2)} = \theta'_{m,3} + \theta'_{m+1,2} \\ \theta'_{m,3} = (1-f(t))\theta^0_{m,3} + f(t)\theta^1_{m,3} \\ \theta'_{m+1,2} = (1-f(t))\theta^0_{m+1,2} + f(t)\theta^1_{m+1,2} \\ 0 < \theta^0_{m,3} + \theta^0_{m+1,2} < \pi \\ 0 < \theta^1_{m,3} + \theta^1_{m+1,2} < \pi \end{cases} \quad (1 \leq m < n-2)$$

由上面的不等式方程组可得 $0 < \angle V'_{Tes(m,2)} V'_{Tes(m,3)} V'_{Tes(m+1,2)} < \pi$

$$(3) 0 < \theta'_{n-2,3} < \pi$$

利用三角形的内角和有: $\theta'_{n-2,3} = \pi - \theta'_{n-2,1} - \theta'_{n-2,2}$

$$\because 0 < \theta^0_{n-2,1} + \theta^0_{n-2,2} < \pi \text{ 且 } 0 < \theta^1_{n-2,1} + \theta^1_{n-2,2} < \pi$$

$$\therefore 0 < \theta'_{n-2,1} + \theta'_{n-2,2} = (1-f(t))(\theta^0_{n-2,1} + \theta^0_{n-2,2}) + f(t)(\theta^1_{n-2,1} + \theta^1_{n-2,2}) < \pi$$

$$\therefore 0 < \theta'_{n-2,3} = \pi - \theta'_{n-2,1} - \theta'_{n-2,2} < \pi$$

$$(4) \quad 0 < \angle V'_{T_{es}(1,2)} V'_{T_{es}(n-2,1)} V'_{T_{es}(n-2,3)} < \pi$$

$$\angle V'_{T_{es}(1,2)} V'_{T_{es}(n-2,1)} V'_{T_{es}(n-2,3)} = \sum_{m=1}^{n-2} \theta'_{m,1} \quad \text{且} \quad \theta'_{m,1} = (1-f(t))\theta^0_{m,1} + f(t)\theta^1_{m,1} \quad (m=1,2,\dots,n-2)$$

$$\because \quad 0 < \angle V^0_{T_{es}(1,2)} V^0_{T_{es}(n-2,1)} V^0_{T_{es}(n-2,3)} = \sum_{m=1}^{n-2} \theta^0_{m,1} < \pi$$

$$0 < \angle V^1_{T_{es}(1,2)} V^1_{T_{es}(n-2,1)} V^1_{T_{es}(n-2,3)} = \sum_{m=1}^{n-2} \theta^1_{m,1} < \pi$$

$$\therefore \quad 0 < \angle V^t_{T_{es}(1,2)} V^t_{T_{es}(n-2,1)} V^t_{T_{es}(n-2,3)} = (1-f(t))\sum_{m=1}^{n-2} \theta^0_{m,1} + f(t)\sum_{m=1}^{n-2} \theta^1_{m,1} < \pi$$

证毕

5.5 算法实例

本章采用文献[2]中给出的凸多边形进行变形。图 5-9 是采用[19]中的算法得到凸网格变形，可以发现在 $t=0.5$ 这个时刻内顶点与外部顶点发生了重合的现象，而采用本文的保凸变形算法得到的凸网格边界能够很好的保持凸性，凸边界的变形如图 5-10，凸网格的变形如图 5-11 所示，很好的克服了[19]中出现的内顶点与外部顶点重合这个现象。

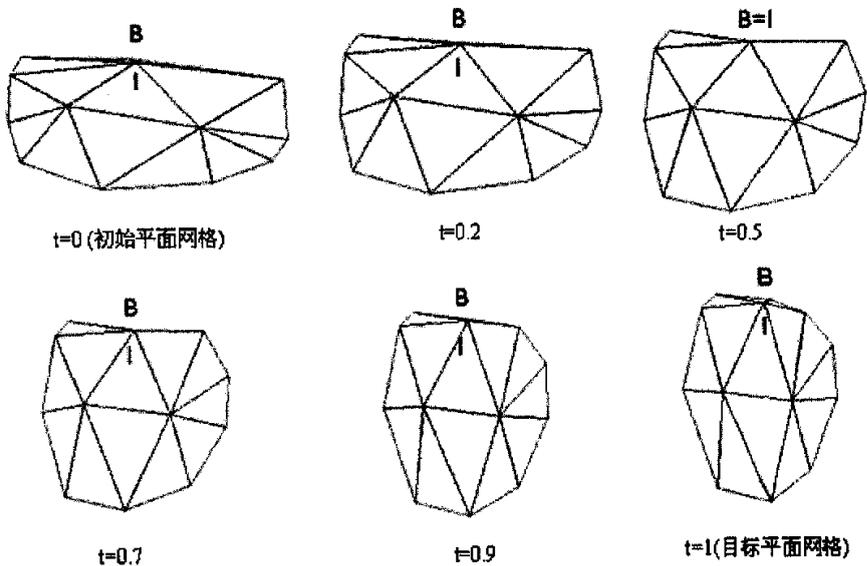


图 5-9 采用[19]的方法对该网格进行变形，当 $t=0.5$ 时，内顶点 I 与边界顶点 B 重合。

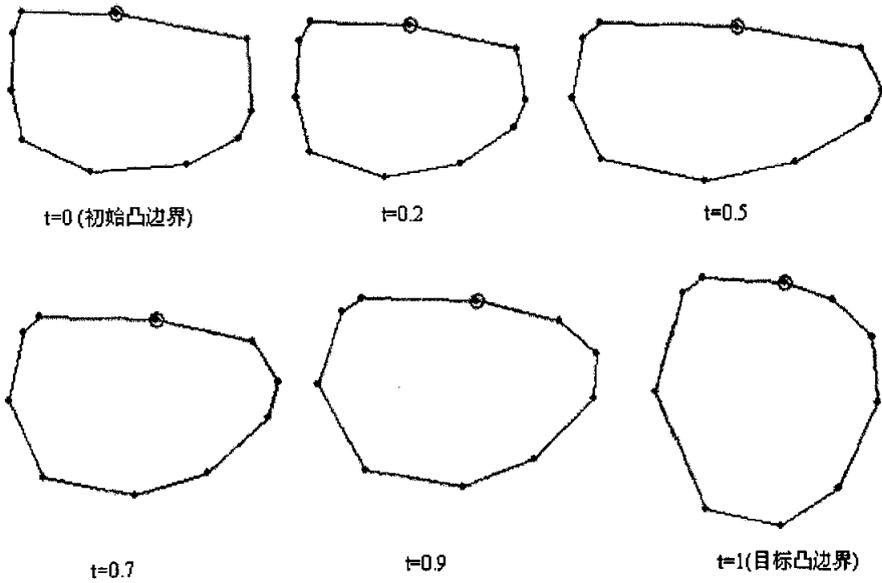


图 5-10 采用算法 5.3 得到网格的凸边界保凸变形

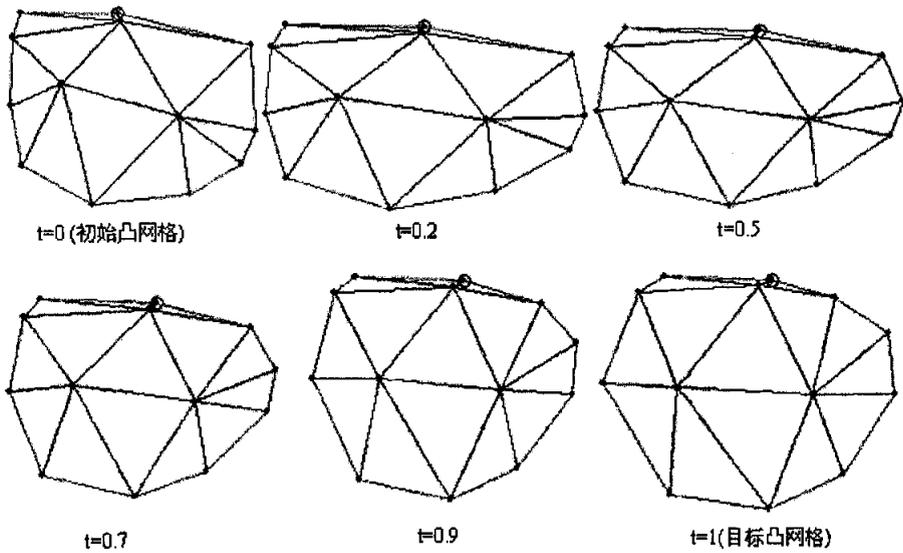


图 5-11 采用算法 5.4 得到的凸网格变形序列

5.6 本章小结

本章对已有的凸网格保凸变形算法进行了介绍，文献[19]中的算法不能够保证凸网格在变形过程中凸边界为凸，从而产生了如图 5-9 所示的内顶点与外顶点重合

的现象，宋在[2]给出的内在解与凸组合集合的算法中，在绝大多数情况下能够保证凸网格的保凸变形，本文对该算法产生自交的可能性做了分析。最后提出了基于角度的保凸变形算法，该算法克服了[19]中的内部顶点与外部顶点重合的现象，同时即保持了内在解算法保持凸边界内角度按同一速度变换的性质，同时保证了凸网格在任何情况下都不会产生自交，给出了理论证明。

第六章 研究结论和展望

6.1 研究结论

本文对平面多边形的变形算法进行了研究，研究主要集中在三个方面：一是平面多边形的同构三角网格的剖分研究；二是基于小波的平面多边形变形算法的研究；三是平面凸网格的保凸变形算法研究。以下是本文在这三个方面取得的主要结论：

- 平面多边形的同构三角网格的剖分研究

对于平面多边形的同构三角网格的剖分，本文第三章提出了基于初末多边形之间的相似性同构三角网格剖分方法。该方法首先考虑对所要进行同构三角剖分的初末多边形之间的相似结构，在不添加额外顶点的情况下三角剖分掉初始和目标多边形之间的相似结构。由此可以得到初末多边形的简化多边形，这两个简化多边形在顶点的个数上已少于原来的初末多边形。再在此简化了的初末多边形之间进行同构三角剖分，这时的同构三角剖分比起在原始初末多边形之间的同构三角剖分来说简单了许多。并且，应用该方法能够减少进行同构三角剖分时不必要的额外顶点，从而减少了多边形变形的复杂度。在第三章中已经给出了该算法的过程以及相应的实例。

- 基于小波的平面多边形变形算法研究

针对现有小波算法的不足与缺点，比如轮廓多边形自交、要求多边形顶点个数必须是2的 n 次方个等不良现象，在本文中，提出了平面简单多边形的简单有效的变形算法，即基于小波的平面多边形变形算法。该方法结合了小波算法和多边形的同构三角网格变形算法，该方法采用小波变换处理初末多边形，得到初末多边形的轮廓多边形以及相应的细节信息，对初末多边形轮廓部分采用内在解算法，对细节信息采用线性插值，再通过重构算法重构出中间时刻的多边形。该方法能够在很大程度上保证初末多边形的轮廓不会自交，且适用范围更加广泛。且通过实践证明了运用该方法能够减轻同构三角剖分的难度，以及减少了进行变形的计算量，从而达到实时的效果，而且变形也可以取得令人满意的变形效果。

- 平面凸网格的保凸变形研究

提出了具有不同凸边界的同构平面三角网格的保凸变形方法。本文提出了基于角度的保凸变形算法变形网格的凸边界，并给出了该算法保凸的理论证明，且具有凸边界内角按同一速度变化的性质，对网格的内部顶点采用凸组方法。本文方法能够保证网格边界在变形过程中始终保持凸性，且任意时刻的中间网格与初末网格同构，即不产生自交现象。

6.2 研究展望

本文在对平面多边形变形的研究过程中，发现其中仍有一些值得进一步探讨的问题，概括如下：

1. 现有的成熟的变形方法已经很多，但如何保证鲁棒的实现这些技术仍是一个值得探讨的问题，如，凸组合变形算法，在理论上它能够完全避免自交、退化等现象的产生，但在网格划分不均匀或者网格包含接近退化的三角形时，虽然理论上仍是可行的，但在实际的计算中，数值计算的稳定性却无法保证，如何避免这些变形中的细节问题，需要进一步的探讨。

2. 在本文的所有算法均是整体的变形算法，即不能进行部分的变形，那么这些算法能否实现部分的变形，以实现更特殊的效果呢？这些将在以后进行更深入的研究。

3. 在对平面多边形进行变形时，不但要求处理多边形的外界顶点，而且更应处理多边形的内部顶点，即多边形的内部属性。现有些比较好的算法停留在处理多边形的外部顶点，如何将这些算法与处理内部属性的算法结合起来值得研究。

4. 目前的嵌入凸边界的凸组合变形算法能够保证初末多边形在变形过程中不产生自交，但是当初始和目标多边形差别较大时，由该方法生成的中间多边形不可避免的会有扭曲的现象。如何克服扭曲现象的产生是值得研究，且很有研究意义。

参考文献

- [1] 王国瑾, 汪国昭, 郑建民. 《计算机辅助几何设计》, 施普林格出版社, 高等教育出版社, 2001.
- [2] 宋伟杰. 同构平面三角网格和平面多边形变形的研究, 西北工业大学硕士论文, 2004.3.
- [3] 张则剑. 同构三角网格的变形研究, 西北工业大学硕士论文, 2005.3.
- [4] A.Gregory, A.State, M.Lin, D.Manocha, and M.Livingston. Feature-based surface decomposition for correspondence and morphing between polyhedra. Proceedings of Computer Animation '98, 64-71, 1998.
- [5] Arthur Gregory, Andrei State, Ming C.Lin, Dinesh Manocha, Mark A.Livingston. Interactive surface decomposition for polyhedral morphing. The Visual Computer, 15(9):453-470, 1999.
- [6] Aronov B, Seidel R, Souvaine DL. On Compatible Triangulations of Simple Polygons. Computational Geometry: Theory and Applications, 3:27-35, 1992.
- [7] A.Shapiro and A. Tal. Polyhedron realization for shape transformation. The Visual Computer, 14(8-9):429-444, ISSN 0178-2789, 1998.
- [8] A.Tal and G.Elber. Image morphing with feature preserving texture. Computer Graphics forum (Eurographics '99 Proceedings), 18(3):339-348, 1999.
- [9] A.W.F.LEE, D.Dobkin, W.Sweldens, and P.Schroder. Multiresolution Mesh Morphing. SIGGRAPH '99 Proceedings, 343-350, 1999.
- [10] Babikov M, Souvaine DL, Wenger R. Constructing piecewise linear homeomorphisms of polygons with holes. Proceedings of Ninth Canadian Conference on Computational Geometry, 1997.
- [11] B. L'evy and J.-L. Mallet. Non-distorted texture mapping for sheared triangulated meshes. Proceedings of SIGGRAPH 98, pages 343-352, July 1998.
- [12] C.Erten, S.G.Kobourov and C.Pitta. Morphing Planar Graphs. Symposium on Computational Geometry : 451-452, 2004.

-
- [13] Cesim Erten, Stephen G.Kobourov, and Chandan Pitta. Intersection-Free Morphing of Planar Graphs, LNCS 2912, Proc. 11th International Symposium on Graph Drawing(GD):320-332, 2003.
- [14] D.Cohen-Or, D.Levin, and A.Solomovici. Three dimensional distance field metamorphosis. ACM Transactions on Graphics, 1998.
- [15] DeCarlo, D., Gallier, J. Topological evolution of surfaces. Proceedings of Graphics Interface'96, Toronto, Canada, 194-203, 1996.
- [16] E.Carmel and D.Cohen-Or. Warp-guided object-space morphing. The Visual Computer, 13:465-478, 1997.
- [17] E.Goldstein and C.Gotsman. Polygon morphing using a multiresolution representation. Proceeding of Graphics interface, 1995.
- [18] Floater, M.S. Parameterization and smooth approximation of surface triangulation. Computer Aided Geometric Design 14,231-250, 1997.
- [19] Floater, M.S. Gotsman C. How to morph tilings injectively. Computational and Applied Mathematics, 101:117-129, 1999.
- [20] Floater, M.S., Pham-Trong, V., Convex combination maps over triangulations, tilings,and tetrahedral meshes. Adv. Comput. Math, 2005
http://heim.ifi.uio.no/~michaelf/papers/cc_maps_short.pdf
- [21] G. Karypis and V. Kumar. Multilevel k-way hypergraph partitioning. Technical Report 98-036, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, 1998.
- [22] G.Zigelmann, R.Kimmel, and N.Kiryati. Texture mapping using surface flattening via multi-dimensional scaling. IEEE Transactions on Visualization and Computer Graphics, to appear, 2002.
- [23] H.Bunke, X.Jiang, K. Abegglen, and A.Kandel, On the weighted mean of a pair of strings, Pattern Analysis and Applications, 2001.
- [24] Henry Johan, Yuichi Koiso, Tomoyuki Nishita. Morphing Using Curves and Shape Interpolation Techniques. In Proceedings of Pacific Graphics 2000, 8th Pacific Conference on Computer Graphics and Application, 348-358, 2000.

-
- [25] Hermann Birkholz, Dietmar Jackel. Image Warping with Feature Curves. Spring Conference on Computer Graphics. Modeling and visualization: 199 - 202, 2003.
- [26] JEFF DANCIGER, SATYAN L.DEVADOSS, AND DON SHEEHY.
COMPATIBLE TRIANGULATIONS AND POINT PARTITIONS BY
SERIES-TRIANGULAR GRAPHS. eprint arXiv:cs/0502043, 02/2005.
- [27] J.F.Hughes. Scheduled Fourier Volume Morphing. SIGGRAPH'92 Proceedings, 26, 2, 43-46, 1992.
- [28] Jindrich Parus, Ivana Kolingerová. Morphing of Meshes with Attributes In Spring conference on computer graphics SCCG 2004. Bratislava: Comenius University, 2004. s.69-78. ISBN80-223-1918-X.
- [29] J. Kent, R. Parent, and W. E. Carlson. Establishing correspondences by topological merging: A new approach to 3-d shape transformation. Graphics Interface '91, pages 271-278, June 1991.
- [30] J.R. Kent, W.E. Carlson, and R.E. Parent. Shape Transformation for Polyhedral Objects. SIGGRAPH '92 Proceedings, 26(2):47-54, 1992.
- [31] Kaul, A., Riossignac, J.. Solid interpolating deformations: construction and animation of PIPs. In: Post, F.H. and Barth, W., eds., Proceedings of Eurographics'91. Elsevier Science Publishers, B.V., 1991.
- [32] K.Fujimura and M.Makarov. Foldover-free image warping. Graphical Models and Image Processing, 60(2):100-111, Mar, 1998.
- [33] K.Polthier. Conjugate harmonic maps and minimal surfaces. Technical Report Preprint No. 446, TU Berlin, SFB 288, 2000.
- [34] M.Alexa, D.Cohen-Or, and D.Levin. As-rigid-as possible polygon morphing. Proceedings of SIGGRAPH '2000, pages 157-164, 2000.
- [35] M.Alexa. Merging polyhedral shapes with scattered features. The Visual Computer, 16(1):26-37, 2000.
- [36] Marc Alexa, Daniel Cohen-Or, David Levin. As-Rigid-As-Possible Shape Interpolation. Proceedings of SIGGRAPH'2000. Computer Graphics, 157-163, 2000.

- [37] Michael S. Floater. Mean Value Coordinates, *Computer Aided Geometric Design* (20): 19- 27, 2003.
- [38] Mikael Persson. Automatic Image Metamorphosis.
http://www.yodel.net/members/mikael/data/morph/Automatic%20Image%20Metamorphosis%20-%20paper_revised.pdf
- [39] M. S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(3):231-250, 1997. ISSN 0167-8396.
- [40] M. S. Floater. Convex combination maps. *Algorithms for Approximation IV*, 2001.
- [41] M. S. Floater. One-to-one piecewise linear mappings over triangulations. *Math. Comp.*, 2002.
- [42] M. Shapira and A. Rappoport. Shape blending using the star-skeleton representation. *IEEE Trans. on Computer Graphics and Application*, 15(2): 44-51, 1995.
- [43] Oswin Aichholzer, Franz Aurenhammer, Ferran Hurtado, Hannes Krasser. Towards Compatible Triangulations. *Theoretical Computer Science*. 296:3-13, Special Issue, 2003.
- [44] Ping-Hsien Lin and Tong-Yee Lee. A Fast 2D Shape Interpolation Technique. *Lecture Notes on Computer Science (LNCS 3482) volume 3482*, Springer-Verlag, pp. 1050-1059, 2005.
- [45] R.A. Wagner and M.J. Fischer. The String-to-String correction problem. *Journal of the ACM*, 21(1): 168-173, 1974.
- [46] R. Singh, R.M. Voyles, D. Littau, and N.P. Papanikolopoulos. Graspling real objects using virtual images, *Proc. of IEEE Conf. On Decision & Control*, 3269-3274, 1998.
- [47] Saalfeld A. Joint triangulations and triangulation maps. *Proceedings of Third Annual ACM Symposium on computational Geometry*, 195-204, 1987.
- [48] S. Cohen, G. Elber, and R. Yehuda. Matching of freeform curves. In *Computer Aided Design*, Vol.29, No.5:369-378, 1997.

-
- [49] Seung-Yong Lee, Kyung-Yong Chwa, James Hahn, Sung Yong Shin. Image Morphing Using Deformation Techniques. *The Journal of Visualization and Computer Animation*, 6(3), 1995.
- [50] Seungyong Lee, George Wolberg, Kyung-Yong Chwa, and Sung Yong Shin. Image Metamorphosis with Scattered Feature Constraints. *IEEE Transactions on Visualization and Computer Graphics*, Volum2(4):337-354, 1996.
- [51] S.Y.Lee, K.Y.Chwa, S.T.Shin, and G. Wolberg. Image Metamorphosis Using Snakes and Free-Form Deformations. *SIGGRAPH '95 Proceedings*. 439-444, 1995.
- [52] T.Beizer and S.Neely. Feature-based image metamorphosis. *Computer Graphics (SIGGRAPH '92)*, 26(2):35-42,1992.
- [53] T.He, S.Wang, and A.Kaufman. Wavelet-Based Volume Morphing. *Proceedings of Visu-alization '94*, 85-92, 1994.
- [54] Thomas W.Sederberg and Eugene Greenwood. Shape Blending of 2-D Piecewise Curves. In *Mathematical methods for curves and surfaces*, Vanderbilt University Press, pages 497-506, 1999.
- [55] T.Kanai, H.Suzuki, and F.Kimura. 3D geometric metamorphosis based on harmonic maps. In *Proceedings of Pacific Graphics*, Tokyo, October, 2001.
- [56] T.Samoilov and G. Elber. Self-intersection elimination in metamorphosis of two-dimensional curvers. *The Visial Computer*, 14:415-428, 1998.
- [57] T.W. Sederberg, P. Gao, G. Wang, and H.Mu. 2-D shape blending: an intrinsic solution to the vertex path problem. *Computer Graphics(SIGGRAPH '93)*,27:15-18, 1993.
- [58] T.W.Sederberg and E.Greenwood. A physically based approach to 2-D shape blending. *Computer Graphics(SIGGRAPH '92)*, 26:25-34, 1992.
- [59] U.Pinkall, K.Polthier. Computing discrete minimal surfaces and their conjugates. *Experimanetal Mathematics*, 2(1):15-36, 1993.
- [60] V.Surazhsky and C.Gotsman. Guaranteed intersection-free polygon morphing. *Computers and Graphics*, 25(1):67-75, 2001.

- [61] Vitaly Surazhsky, Craig Gotsman. Morphing Stick Figure Using Optimized Compatible Triangulations. Proceedings Ninth Pacific Conference on Computer Graphics and Applications, 40-49, 2001.
- [62] Vitaly Surazhsky, Craig Gotsman. High Quality Compatible Triangulations Proceedings of 11th International Meshing Roundtable, pages 183-192, Sept. 2002.
- [63] V. Surazhsky and C. Gotsman, Controllable morphing of compatible planar triangulations, ACM Transactions on Graphics, 20(4):203-231, 2001.
- [64] VITALY SURAZHSKY, CRAIG GOTSMAN. Intrinsic Morphing of Compatible Triangulations. International Journal of Shape Modeling. Vol.9, No.2 :191-201, 2003.
- [65] Xiaoyi Jiang, Horst Bunke, Karin Abegglen, Abraham Kandel. Curve Morphing by Weighted Mean of strings. In Proceedings of the Sixteenth Conference on Pattern Recognition (ICPR 2002), volume 4, pages 192-195. IEEE Press, 2002.
- [66] W. T. Tutte. How to draw a graph. Proc. London Mathematical Society, 13:743-768, 1963.
- [67] Yuefeng Zhang, Yu Huang. Wavelet Shape blending. The Visual Computer. Volume: 16. Issue: 2. p. 106 -115, 2000.
- [68] Ziqiang Liu, Ce Liu, Heung-Yeung Shum, Yizhou-Yu. Pattern-based Texture Metamorphosis. Pacific Conference on Computer Graphics and Applications, pp. 184-193, 2002.

致 谢

值此论文完稿之际，心中百感交集。三年的研究生生活，犹如白驹过隙，往事依然历历在目。在此，我要感谢在这三年里给过我帮助的所有老师和同学、朋友，以及培养我的母校——西北工业大学。

首要一桩，感谢我的导师蒋大为教授，导师治学严谨，学识渊博，品德高尚，平易近人，在论文的撰写全过程中不仅传授了做学问的秘诀，还传授了做人的准则。这些都将使我终生受益。无论是在理论学习阶段，还是在论文的选题、资料查询、开题、研究和撰写的每一个环节，无不得到导师的悉心指导和帮助。他严肃求实的敬业精神和一丝不苟的工作作风将永远是我学习的楷模！我在学习中取得的进步和获得的成果，都离不开他的悉心的指导和耐心指正，使我少走许多弯路。我愿借此机会向我的导师表示衷心的感谢！

这里我要向所有支持、关心和帮助我的人表示诚挚的谢意！感谢他们对我的关心、关注和支持！感谢图形图象教研室的各位老师以及王润霞老师给予我的热情帮助、指导，感谢张正贤和张莹同学在生活上给予我的关心和帮助以及学业上的切磋和指点，感谢 03 级数硕班全体同学的帮助和勉励。同窗之谊和手足之情，我将终生难忘！

最后，我愿在未来的学习和工作过程中，以更加丰厚的成果来答谢曾经关心、帮助和支持过我的所有领导、老师、同学和朋友，以及西北工业大学对我的栽培之恩。由于本人水平有限，此论文中仍会有一些不足之处，尚望各位老师多予批评指正。

攻读学位期间论文发表情况

- 1、 第一作者，《一种基于小波的多边形变形算法》，计算机工程与应用，定于 Vol.42, No.3 上发表。
- 2、 第三作者，《基于迭代函数系统的分形变形方法研究》，计算机工程与应用，定于 Vol.42, No.10 上发表。
- 3、 第四作者，《一种快速稳健的超分辨率图像复原算法》，计算机工程与应用，定于 Vol.42, No.4 上发表。
- 4、 第四作者，《二维迭代函数系统分形吸引子自适应对应变形算法》，计算机辅助设计与图形学学报，已录用。

西北工业大学

学位论文知识产权声明书

本人完全了解学校有关保护知识产权的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属于西北工业大学。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版。本人允许论文被查阅和借阅。学校可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律注明作者单位为西北工业大学。

保密论文解密后适用本声明。

学位论文作者签名：危才华

2006年4月4日

指导教师签名：李大为

2006年4月4日

西北工业大学

学位论文原创性声明

秉承学校严谨的学风和优良的科学道德，本人郑重声明：所呈交的学位论文，是本人在导师的指导下进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容和致谢的地方外，本论文不包含任何其他个人或集体已经公开发表或撰写过的研究成果，不包含本人或其他已申请学位或其他用途使用过的成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式表明。

本人学位论文与资料若有不实，愿意承担一切相关的法律责任。

学位论文作者签名：危才华

2006年4月4日