

使用二维轮廓线的扩展元球造型技术*

刘圣军¹⁺, 金小刚², 林俊聪², 冯结青²

¹(中南大学 数学科学与计算技术学院, 湖南 长沙 410083)

²(浙江大学 CAD&CG 国家重点实验室, 浙江 杭州 310058)

Extended Metaball Modeling Technique with 2D Silhouette Curves

LIU Sheng-Jun¹⁺, JIN Xiao-Gang², LIN Jun-Cong², FENG Jie-Qing²

¹(School of Mathematical Science and Computing Technology, Central South University, Changsha 410083, China)

²(State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou 310058, China)

+ Corresponding author: E-mail: shjliu.cg@gmail.com

Liu SJ, Jin XG, Lin JC, Feng JQ. Extended metaball modeling technique with 2D silhouette curves. *Journal of Software*, 2011, 22(12): 2981-2993. <http://www.jos.org.cn/1000-9825/3943.htm>

Abstract: This paper proposes a simple and efficient method for creating three-dimensional prototypes based on sketching silhouette curves and super-quadric-surface metaballs. Given a silhouette curve, it is first approximated with a set of circles or ellipses. By introducing the third dimensional parameters, a super-ellipsoidal metaball is received for each circle or ellipse. Finally, an analytic implicit surface is constructed by summing the fields of all metaballs and optimizing all parameters. The resulting shape can be conveniently modified by adjusting the positions and shape parameters of metaballs. Different components of a model can be similarly designed by sketching on the different projection planes. This method supports simple solid modeling operations, such as adding, subtracting, semi-sweeping, and is able to generate variants of shapes easily. This method can be applied to designing prototype in the conceptual design stage for computer graphics or computer aided design applications.

Key words: 2D silhouette curve; implicit surface; metaball; modeling; shape approximation

摘要: 提出了一种基于二维轮廓线与超二次曲面型元球模型的造型方法,以实现简单、快速三维模型原型构造. 给定一条二维轮廓线,首先使用圆或椭圆去逼近它.然后对每一个圆或椭圆设置第三个维度的参数,能够获得相应的超椭球体型元球.最后混合所有元球的场,并对所有元球的形状参数进行优化,得到一个解析的元球隐式曲面.通过调整元球的位置或形状参数构造出的曲面能够方便地实现形状的修改.模型的不同组成部分可以在不同的投影平面上勾画轮廓线相似地进行设计.该造型方法支持简单的建模操作,如布尔加、减或准扫成体生成,以生成各种复杂形状.此方法能够广泛地应用在计算机图形学或计算机辅助设计领域中的概念设计阶段的原型设计工作中.

关键词: 二维轮廓线;隐式曲面;元球;建模;形状逼近

中图法分类号: TP391 文献标识码: A

* 基金项目: 国家自然科学基金(60970097, 60933007); 国家重点基础研究发展计划(973)(2009CB320801); 浙江省科技计划(2009C33001); 浙江大学 CAD&CG 国家重点实验室开放课题基金(A0805)

收稿时间: 2009-01-02; 修改时间: 2010-03-03; 定稿时间: 2010-08-27

曲面造型技术是计算机辅助设计(CAD)、计算机辅助制造(CAM)及图形学的基础.其研究始于多项式逼近、插值技术,最终出现了以 Bézier、B 样条等曲线/曲面造型技术为代表的参数曲面造型技术.尽管参数曲面造型技术取得了极大的成功,但由于参数表示的固有缺陷,这一技术仍存在不足.为此,从 20 世纪 80 年代开始,出现了另一种曲面造型技术——隐式曲面造型技术.隐式曲面是使用隐式函数方程来表示物体表面,隐式方程可以写为 $F(x,y,z)-Iso=0$.通过调整 Iso 的值,可以得到不同的曲面.每一个曲面都被称为等势面,而方程 $F(x,y,z)$ 称为场函数.通过定义不同的场函数并且对其进行混合,可以生成各种复杂的曲面.

使用隐式曲面进行造型可以确保生成模型的连续性和光滑性,并且可以很容易地实现对造型结果的变形处理.Blinn 首次在计算机图形学领域提出了一种特殊的隐式曲面造型技术,称其为 Blobby 模型^[1].随后, Nishimura 等人提出了 Metaball 模型^[2],Wyvill 提出 Soft Object 模型^[3].这些模型我们都称之为元球模型,其隐式函数是一些径向对称函数的和,而这些径向对称函数都具有高斯函数的几何形状.元球隐式曲面广泛应用于计算机图形学的各个领域.Tatsumi 给出了元球造型技术在鳕鱼肝脏造型过程中的使用^[4].Muraki 将 Blobby 模型用于三维物体造型(包括人的面部造型)^[5].Max 和 Wyvill 利用 Soft Object 绘制了珊瑚虫^[6].Payne 和 Toga 利用距离场生成了老鼠大脑的模型^[7].Reed 和 Wyvill 利用 Soft Object 来模拟闪电^[8].Tsingos 等人克服文献[5]中方法效率慢的缺点,提出一种准自动的元球模型重构逼近的方法^[9].同年,Bittar 等人提出了一种全自动的元球隐式曲面重构的方法^[10].然而,这种方法不稳定,需要用户指定恰当的中轴分辨率,否则将生成错误的结果.Jin 等人采用中轴球体树的逼近结果,能够稳定地重构出逼近给定模型的隐式曲面,并且用来模拟流体运动特效^[11].在这些应用中,使用的都是各向同性的核函数,如球体函数.由于球体的各向同性的性质,逼近尖锐或者扁平的特征会产生较大的走样.为了减少这种误差,Liu 等人^[12]提出了使用椭球体作为原始体来构造元球隐式曲面.这个算法既减少了元球的个数,又提高了逼近精度,得到了较好的逼近结果.然而,椭球型元球模型无法生成具有尖锐边的模型.因此在本文中,我们扩展椭球型元球模型,采用超二次曲面作为原始体进行建模.

最近几年,研究者在开发直观的快速造型交互平台方面做了大量的工作,使得原型设计工作变得更加容易.在这些基于草图的自由形状造型系统中,采用了各种模型形状表示方法,包括三角网格^[13]、细分曲面^[14]、变分隐式曲面^[15,16]、球型隐式函数^[17]、卷积曲面^[18]、离散的体数据集^[19]和分层隐式体表示(元球树)^[20].Teddy^[13]是首个基于草图概念的自由形状建模系统,它通过用户勾画的二维轮廓线生成封闭的三角网格模型.但 Teddy 系统不支持独立组件的任意融合,不能生成挤压组件,且不能构建亏格大于 0 的物体.Igarashi^[14]在 Teddy 的基础上采用基于局部二次隐式曲面逼近进行重新网格化,可以得到更加光滑的结果.一些研究人员使用变分隐式曲面作为底层的模型表示^[15,16]来改进 Teddy.然而,系统为了求解变分方程,需要根据约束点进行大的矩阵运算,这使得生成的曲面复杂性受到限制.其他基于隐式曲面表示的草图建模系统如使用球型隐式函数^[17]、卷积曲面^[18],则这两个系统不能支持尖锐边的构造,且模型的复杂度低.采用分层隐式体表示^[20]的系统是一个比较好的基于草图的自由形状建模的系统,但是它不能支持模型在轮廓线层次的局部微调.如果要进行形状调整,那么整个结果需要重新计算.Owada 等人在 Teddy 系统中使用体数据来表示模型^[19],构建出具有更多细节的模型,但是它受到体数据分辨率的限制.最近,Li 等人提出一种新的二维代数样条,并使用它进行隐式造型^[21].对任意给定的二维多边形,可以创建分段多项式表示的样条基函数.由于其显式解析解,计算过程简单高效.它们的方法支持自由形状、拉伸面和旋转面造型.然而,该方法与经典的参数样条一样,使用控制多边形或控制多边形网格进行曲线、曲面造型.不同的是,我们直接由模型的外形轮廓线来造型,更加直观.另外,我们的方法对于每一条轮廓线都能独立创建一个实体,而且支持独立两个实体组件间的任意融合,造型过程更加灵活.

本文中,我们的目标是将元球隐式曲面应用到基于任意 2D 轮廓线的造型工作中,提供一种直观、快速的造型技术,支持布尔加、减操作,能够进行准扫成体建模和局部修改.在使用圆集逼近算法^[22]的基础上,我们扩展使用椭圆集来逼近轮廓线.基于这种逼近,如果我们要修改模型形状,只需要改其逼近集中的圆或者椭圆即可.

本文第 1 节详细介绍一般的超二次曲面元球模型及其场值计算方法.第 2 节给出详细的造型算法,包括二维轮廓线中轴抽取、圆逼近、椭圆逼近、超椭球型元球曲面的构造和曲面的数值优化.第 3 节给出使用造型方法得到的实验结果.第 4 节是总结,并且提出今后的研究方向.

1 超二次曲面元球模型

给定一个点集 $S \subset \mathcal{R}^3$, 对于每一个点 $c_i \in S, f_i: \mathcal{R}^3 \rightarrow \mathcal{R}$ 是一个核函数, 它表示由点 c_i 生成的势场. 如果 p 是欧氏空间 \mathcal{R}^3 中的一个点, 那么点集 S 中所有的点在点 p 处的总场值可以通过式(1)求得:

$$F(p) = \sum_i \omega_i f_i(c_i, p) \quad (1)$$

$F(p) - t = 0$ 则定义一张阈值为 t 的等势面.

Blinn^[1]在他提出的元球模型中使用全局核函数, 该函数的函数值在无穷远处为 0, 因而计算量非常大. 为了提高计算速度, f_i 通常被定义成一个局部支撑的多项式函数^[2,3,22]. 如 Nishimura^[2]使用了一个二次多项式函数, Wyvill 在文献[3]中给出了一个六次多项式函数, 而 Murakami^[22]定义了一个四次多项式函数. 在我们的算法中采用由 Murakami^[22]提出的四次多项式函数作为核函数:

$$f_i(p) = \begin{cases} \left(1 - \frac{r_i^2}{R_i^2}\right)^2, & \text{当 } r_i \in [0, R_i] \\ 0, & \text{其他} \end{cases} \quad (2)$$

在公式(2)中, $r_i = d(p, c_i)$ 返回的是给定点 p 到第 i 个元球中心 c_i 的欧氏距离, R_i 定义了元球的影响区域. 由一个核函数定义的隐式曲面, 称为原始体. 一般情况下, 元球隐式曲面的原始体都是球体, 称为球型元球模型. 尽管球型元球模型能够逼近表示许多物体的表面, 但是由于球体的各向同性的性质, 不能很好地表示尖锐或者扁平的特征. 为了改善这种情况, Liu 等人^[12]提出了使用椭球体作为原始体来构造元球隐式曲面. 然而, 椭球型元球模型没有办法生成具有尖锐边的模型. 因此, 为了能够进行更加灵活地建模, 本文扩展椭球型元球模型, 采用超二次曲面作为原始体.

由于超二次曲面的各向异性, 空间中点的场值计算不能简单使用公式(2)计算. 为了方便说明, 我们修改公式(2)成如下形式:

$$f_i(p) = \begin{cases} (1 - \varphi_i^2)^2, & \text{当 } \varphi_i \in [0, 1] \\ 0, & \text{其他} \end{cases} \quad (3)$$

这里, φ_i 为空间的点 p 到第 i 个元球中心 c_i 的相对距离. 对于球体, 取 $\varphi_i^2 = r_i^2 / R_i^2$; 对于标准椭球体, 空间中的点在场中的场值可以通过将公式(4)代入公式(3)计算得到:

$$\varphi_i^2 = (x - x_{c_i})^2 / R_{i,x}^2 + (y - y_{c_i})^2 / R_{i,y}^2 + (z - z_{c_i})^2 / R_{i,z}^2 \quad (4)$$

然而在实际应用中, 原始体并不全部都是标准椭球体, 其 3 个轴方向是任意的相互正交的单位向量 $V_{i,a}, V_{i,b}, V_{i,c}$. 为了计算空间点在这种类型椭球原始体所定义的势场中的场值, 我们为每个椭球体建立一个局部坐标系, 椭球体的中心为坐标系原点, 椭球体的 3 个轴作为坐标系的 3 个轴. 这样, 我们就将任意椭球体的场值计算转化成标准椭球体场值的计算. 公式(4)可以修改成如下形式:

$$\varphi_i^2 = (r_i \cos(\alpha))^2 / R_{i,a}^2 + (r_i \cos(\beta))^2 / R_{i,b}^2 + (r_i \cos(\gamma))^2 / R_{i,c}^2 \quad (5)$$

其中, $(R_{i,a}, R_{i,b}, R_{i,c})$ 定义了元球的椭球型影响区域, (α, β, γ) 分别是向量 $c_i p$ 与向量 $V_{i,a}, V_{i,b}, V_{i,c}$ 的夹角. 我们令 $V_{i,p}$ 为向量 $c_i p$ 的单位向量, 则

$$\cos(\alpha) = V_{i,p} \cdot V_{i,a}, \cos(\beta) = V_{i,p} \cdot V_{i,b}, \cos(\gamma) = V_{i,p} \cdot V_{i,c} \quad (6)$$

其中, $(\cos(\alpha))^2 + (\cos(\beta))^2 + (\cos(\gamma))^2 = 1$, 表示点乘. 我们将公式(5)、公式(6)代入公式(3), 即可计算出空间点 p 处的场值.

由上面的结果类推, 我们可以得到空间中任意点 p 到超二次曲面原始体中心的相对距离计算公式:

$$\varphi_i^2 = (r_i |\cos(\alpha)|)^{n_a} / R_{i,a}^{n_a} + (r_i |\cos(\beta)|)^{n_b} / R_{i,b}^{n_b} + (r_i |\cos(\gamma)|)^{n_c} / R_{i,c}^{n_c} \quad (7)$$

其中, (n_a, n_b, n_c) 为超二次曲面的次数. 将公式(6)、公式(7)代入公式(4), 计算得到的值就是空间点 p 在超二次曲面原始体所定义的势场中的场值. 对于每个超二次曲面原始体, 有 11 个自由变量:

$$(x_{c_i}, y_{c_i}, z_{c_i}, R_{i,a}, R_{i,b}, R_{i,c}, n_a, n_b, n_c, \theta, \sigma),$$

其中, (θ, σ) 定义超二次曲面的方向. 用这些自由变量可以调节超二次曲面型元球的形状.

由公式(7)易知: 当 $n_a=n_b=n_c=2$ 时, 它定义了一个椭球原始体; 当 $n_a=n_b=n_c=2$ 且 $R_{i,a}=R_{i,b}=R_{i,c}=R_i$ 时, 定义了一个球原始体.

2 由二维轮廓线构造元球隐式曲面

给定一条二维轮廓线, 使用元球造型系统可以构造出一个三维的模型. 二维轮廓线可以由用户通过鼠标勾画或者通过抽取一幅二维图像的轮廓线, 然后经过用户交互编辑得到. 我们用简单多边形表示二维轮廓线. 为了由这个简单多边形生成一个元球隐式曲面, 我们通过下列步骤确定构成元球隐式曲面的元球及其参数: 首先, 通过带约束的 Delaunay 三角化抽取轮廓线的中轴; 然后, 在中轴上放置二维的原始体(圆或者椭圆), 原始体的大小和个数根据其逼近轮廓线的情况来确定; 最后, 为了生成三维模型, 每个二维的原始体对应一个三维的原始体, 通过数值优化确定每个元球的最优参数值.

2.1 二维轮廓线的中轴抽取及圆逼近

任何物体都可以被表示成多个球体的并. 中轴变换可以将一个物体表示成包含在它内部的最大球体集的并, 一般这个集合是一个无限集合. 在实际应用中, 通常用一个球体的有限集合去逼近模型. 同理, 在二维情况下, 我们可以用一个圆盘的有限集合去逼近二维形状^[23].

简单多边形的中轴线和 Voronoi 图的边界线段紧密相关. 一个点集的 Voronoi 图是指把平面划分为几个区域, 其中, 区域 I 中包含距离点 I 最近的所有点集. 通过去掉不完全位于多边形内部的 Voronoi 边及每个凹顶点的两条相关 Voronoi 边, 我们可以得到中轴线. 结合元球造型系统参数确定的需求, 我们采用下面一种近似中轴线的计算方法^[24]. 如图 1 所示, 首先, 对简单多边形(如图 1(a)所示)进行带约束的 Delaunay 三角化^[25](如图 1(b)所示), 然后找到每个三角片的外心, 根据三角片的连接关系将其连接起来, 获取一个由三角片的外心构成的骨架(如图 1(c)所示).

获得一个二维轮廓线的近似中轴后, 在每个中轴线节点上放置一个圆, 可以得到轮廓线的圆逼近表示(如图 1(d)所示). 圆的半径根据如下规则计算得到: 找到该节点相关三角片的 3 个顶点, 每个顶点又分别是两条边界边的端点, 分别计算该节点与上面 6 条边的距离, 要求该节点在每条边上的垂足位于该边界内部; 否则, 忽略该距离, 取最小距离为该基元球的半径. 如果我们简单地将这样所得到的圆作为二维轮廓线的圆表示, 会发现其逼近情况很差. 因为在中轴线上, 有些地方相邻节点距离过大. 为了改善二维轮廓线的逼近情况, 需要在距离过大的相邻节点之间插入新的节点, 即增加一个圆, 用来减少中轴圆的逼近误差. 新增中轴圆的位置和半径可以简单地取相邻两节点处中轴圆的插值. 在文献[26]中, 相邻两节点之间的距离是否过大是根据经验值来判断的. 一般认为, 如果两点之间的距离大于两个中轴圆半径之和的 0.6 倍, 则认为距离过大, 需要增加一个新的中轴圆. 详细的近似中轴线抽取算法和圆逼近算法可以参考文献[26].

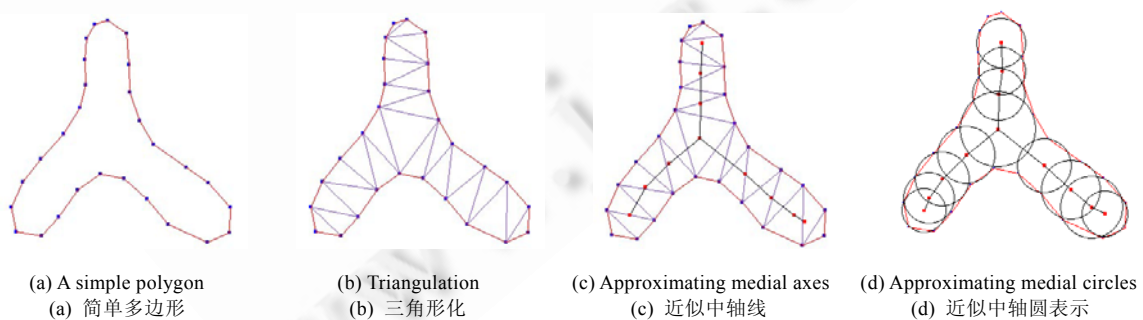


Fig.1 Approximating medial circles from a simple polygon

图 1 获取简单多边形的近似中轴圆表示

2.2 二维轮廓线的椭圆逼近

在第 2.1 节中,我们使用圆为原始体来逼近轮廓线.然而,以圆为原始体进行造型有一定的局限,每个初始的圆原始体仅在一个维度上确保了该原始体接近于我们欲造型物体的轮廓线.在实际使用中,为了使最终得到的模型足够光滑,通常需要利用节点密度比较高的中轴线,并进行适当的插值,从而会导致最终系统的元球个数较多.如图 2(a)所示,仅仅使用原始体 1、原始体 3 是不能达到比较理想的效果的,必须增加新的原始体 2.我们尝试使用椭圆为原始体来进行二维轮廓线的逼近.使用椭圆作为原始体,可以使每个原始体在两个维度上(即椭圆的长、短轴)接近于欲造型的物体,从而可以有效地减少原始体的数量.如图 2(b)所示,对原始体 1、原始体 3 在另一个维度上变化,即可达到比较理想的效果,而无需新增原始体 2,这样能够有效地减少后期的优化及绘制的工作量.

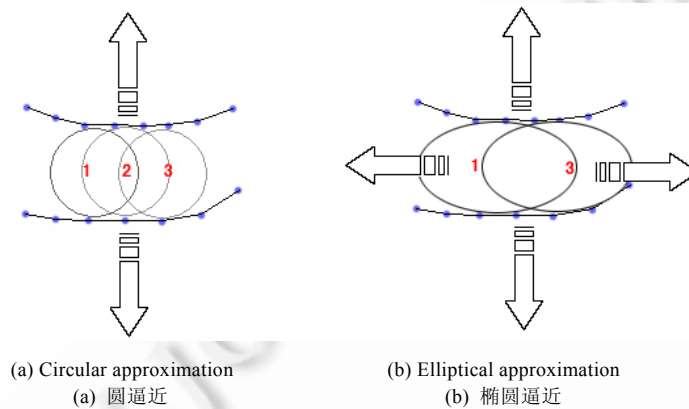


Fig.2 Circular approximation and the elliptical approximation of a silhouette curve

图 2 轮廓线的圆逼近与椭圆逼近

2.2.1 基于中轴线的椭圆生成

基于中轴线的椭圆生成算法,是在圆生成算法基础上改进的.在生成初始的圆集后,我们对圆进行变形以获得椭圆.为了实现这个变形过程,我们要确定椭圆增长的方向,即椭圆长轴方向和增长量.假定待求椭圆的中心为中轴线上的节点 c ,在其处的初始圆半径为 r ,其相邻节点为 r, s, t (中轴线算法决定了每个中轴线节点最多不超过 3 个相邻的节点),短轴长为 $b=r$,对应三角片的 3 个顶点为 p_1, p_2, p_3 .首先,采用节点 c 与节点 $v(v=r, s, t)$ 之间形成的向量作为椭圆长轴的方向向量,其单位向量为 $V_a=(r-v)/|r-v|$.短轴方向为与向量 V_a 垂直的向量方向.然后增大椭圆长轴长度 $a=r+\delta$,直到椭圆碰到除 p_1, p_2, p_3 外的另外一个采样点, δ 是椭圆长轴的增量.为了确定一个最优 δ ,我们计算对于轮廓线上每一个点 x 处在所求椭圆上时,该椭圆的长轴长度增量:

$$\delta(x) = \sqrt{\frac{\|x-c\|^2 - ((x-c) \cdot V_a)^2}{b^2 - ((x-c) \cdot V_a)^2}} \cdot b - r \tag{8}$$

最优 δ 为 $\delta_{opt} = \min\{\delta(x) | (x-c) \cdot V_a > 0\}$.依次遍历中轴线上的有效节点,对每个相邻节点 $v(v=r, s, t)$ 生成一个候选椭圆,最终选取面积最大的那个椭圆作为中心在该节点的椭圆.

有时,获取的原始体产生很大的冗余.这时,需要根据每个椭圆与相邻椭圆对物体的贡献进行筛选.具体策略为:假定椭圆 A 、椭圆 B 为相邻的椭圆,若 A, B 的长轴方向相同, $|r_a - r_b| < \alpha \delta$ (α 为一较小的正浮点数, r_a, r_b 为 A, B 的短轴),且 $dis + r'_a < r'_b$ 或 $dis + r'_b < r'_a$ (dis 为 A, B 中心距离, r'_a, r'_b 为 A, B 的长轴),则删除掉对物体贡献小的椭圆.

在实际应用中我们发现,如果对初始圆的半径取一个微缩参数 0.95,即取 $r_{new} = 0.95 r_{old}$,可以使椭圆的长轴得到更加有效的延展,那么每个椭圆原始体对物体的贡献更大,进而得到更加理想的效果.

2.2.2 基于采样点的椭圆生成

Bischoff 等人在对 3D 模型的椭球体分解中,提出了一种基于原始数据点生成有效椭球的算法^[27].我们将这

种算法简化到二维空间中,并做了一些修改,从而更简单地实现了对一个二维轮廓的椭圆分解逼近.任意选取原始数据点 p ,开始增大(椭圆),直到碰到另一个原始数据点 q 为止.取(椭圆)的圆心 c ,沿向量 cp 的垂直方向继续增大(椭圆),直到第3个原始数据点 s 被发现. p, q, s 即确定了一个椭圆原始体.依次对所有的原始数据点重复前面的步骤,即可以得到了所有的椭圆原始体,如图3所示.

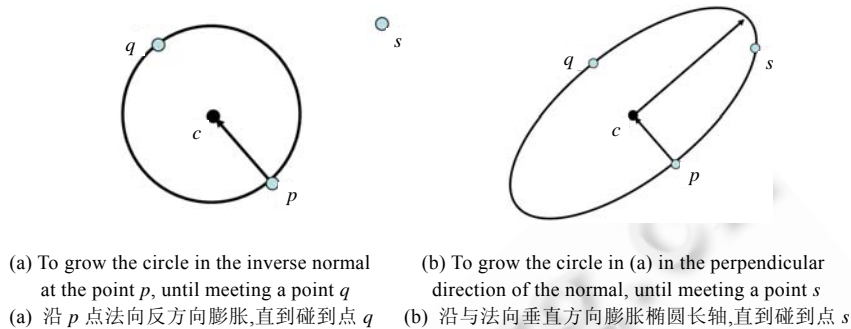


Fig.3 Illustration of an ellipse generation (the points p, q, s determine an elliptical primitive)

图3 椭圆生成示意图(p, q, s 即确定了一个候选椭圆原始体)

为了保证在生成椭圆的过程中避免由于需要判断椭圆是否突出到二维形状的外面而进行的线段同椭圆的求交运算,我们对二维轮廓线进行密集采样.将原始的数据点作为生成椭圆的种子点,从 p 点开始,估计该点的一个法向 n_p ,沿着法向量方向,定义一个半径为 r 的圆.为了使 p 点在圆上,则圆心位置为 $c=p+rn_p$.不断增加半径 r 的值,使得有轮廓线上的第2个点 q 处在圆上.在这里,点 q 可以是原始数据点或者采样点.

对于每个点 x ,由插值点 p, x 和法向 n_p 确定的圆的半径 $r(x)$ 为

$$r(x) = \frac{\|x - p\|^2}{2(x - p)^T n_p} \quad (9)$$

取最优的半径 r 为 $r_{opt} = \min\{r(x) | (x-p) \bullet n_p > 0\}$. 找到一个有两个点处在圆上的最大的内切圆后,我们将继续增大这个圆.按照文献[27]中的方法,这时的增长方向是垂直于向量 pq 的方向.根据他们的三维算法,能够很容易推导出二维的公式.本文使用了更简单快速的方法生成椭圆:将内切圆的中心 c 作为椭圆的中心, n_p 作为短轴方向,沿着 n_p 的垂直方向增长椭圆,直到另外一个点 s 被发现处于椭圆上.此时,我们可以采用基于中轴线的椭圆生成算法来确定椭圆长轴的增长量.对于每一个种子点,我们都生成一个尽可能大的椭圆.

我们发现,通过上面的计算过程得到的椭圆原始体非常多.为了使用尽可能少的原始体来生成物体,采用贪婪算法,从所有的候选椭圆原始体中确定最终的有效椭圆原始体集合.首先从对物体贡献最大的椭圆原始体开始,将其加入到有效椭圆原始体集合中,然后在每一步计算中,我们都选取对未覆盖区域贡献最大的椭圆原始体加入到有效椭圆原始体集合中.如果有多个椭圆的贡献相同,则选取半径最小的椭圆.

计算每个椭圆原始体对物体的贡献是一件比较耗时的工作.在实现中,我们采用了一种近似算法进行计算.首先生成一个包含所有原始数据点的网格,分别计算每个候选椭圆原始体包含的网格个数作为其贡献值.当某个椭圆原始体被确定加入到有效椭圆原始体集合中后,将该椭圆原始体包含的网格置为无效,重新计算剩余候选椭圆原始体的贡献,直到所有候选椭圆原始体的贡献为0时结束.图4给出了由初始所有种子点生成的椭圆原始体及构造出的三维模型和经过筛选后的有效椭圆原始体及其构造的模型比较.我们发现,椭圆筛选后生成的模型没有明显差别.

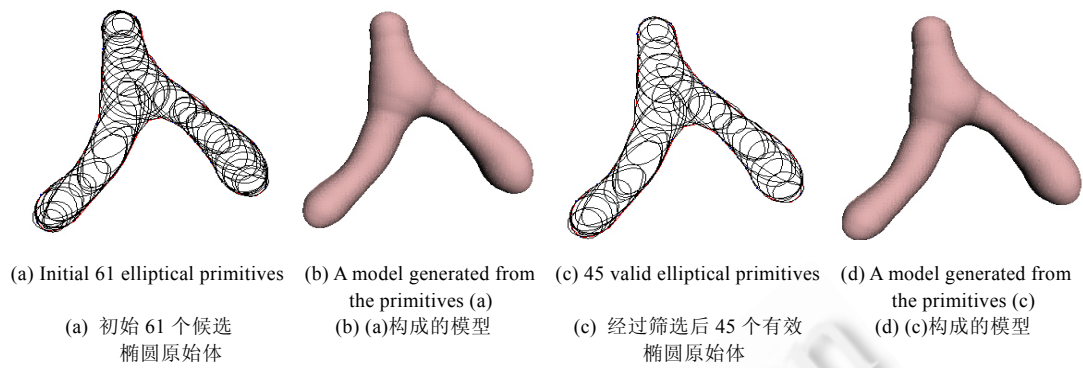


Fig.4 Initial 61 elliptical primitives and the 45 valid elliptical primitives after filtering, the models generated from the primitives respectively

图 4 初始 61 个候选椭圆原始体与经过筛选后 45 个有效椭圆原始体构成的模型

2.3 构造元球隐式曲面

获得二维轮廓线的原始体逼近后,需要设置第三维度的信息以生成三维形状.不失一般性,我们假设用户交互界面为 $x-y$ 平面,二维轮廓线处在该平面上.在这个平面上,我们采用圆或椭圆作为原始体逼近二维轮廓.当我们设定这些圆或者椭圆在 z 轴方向的大小及其相关参数,可以生成球面、椭球面或超二次曲面.每一个原始体对应一个元球,将它们进行混合后可以得到给定轮廓的三维物体.

一个标准的超二次曲面的方程为

$$\frac{x^n}{R_a^n} + \frac{y^n}{R_b^n} + \frac{z^n}{R_c^n} = 1 \tag{10}$$

其中, R_a, R_b, R_c 为 3 个方向轴的大小, n 为正实数.在我们的应用中,生成的超二次曲面在平行交互界面的平面上的截线是椭圆,称这类超二次曲面为超椭球.由超椭球作为原始体生成的元球隐式曲面,称为超椭球型元球隐式曲面.一个标准的超椭球的方程为

$$\frac{x^2}{R_a^2} + \frac{y^2}{R_b^2} + \frac{z^n}{R_c^n} = 1 \tag{11}$$

这里, R_a 和 R_b 在对轮廓线进行逼近的时候已经确定,用户可以根据造型需要来确定 n 和 R_c 的值.使用公式(3)为元球的核函数.由公式(7),可以得到空间中的点到第 i 个超椭球原始体的相对距离计算公式:

$$\varphi_i^2 = (r_i |\cos(\alpha)|)^2 / R_{i,a}^2 + (r_i |\cos(\beta)|)^2 / R_{i,b}^2 + (r_i |\cos(\gamma)|)^n / R_{i,c}^n \tag{12}$$

将公式(6)、公式(12)代入公式(3),计算得到的值就是空间点 p 在超椭球原始体所定义的势场中的场值.对于每个超椭球原始体,有 9 个自由变量 $(x_{c_i}, y_{c_i}, z_{c_i}, R_{i,a}, R_{i,b}, R_{i,c}, n, \theta, \sigma)$ 来调节超椭球型元球的形状.在我们的造型方法中,轮廓线的逼近能够获得对变量 $x_{c_i}, y_{c_i}, z_{c_i}, R_{i,a}, R_{i,b}, R_{i,c}, \theta, \sigma$ 较好的估计.元球的形状可以通过改变 $R_{i,c}, n$ 的值来调节,如图 5 所示.

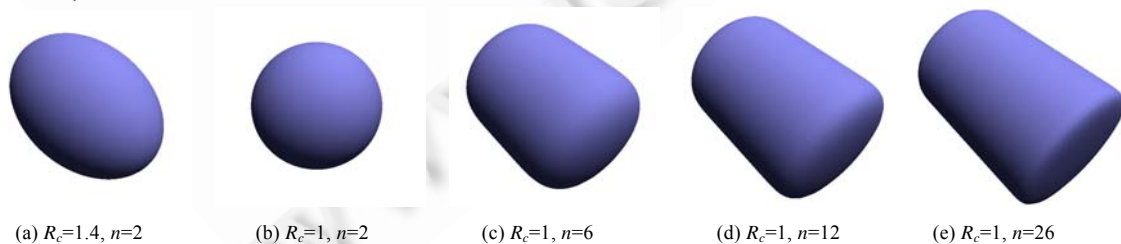


Fig.5 Super-Ellipsoids with different values of R_c, n

图 5 取不同的 R_c, n 值生成的超椭球体

利用超椭球核函数的几何特性,我们可以设计一些比较特殊模型(如较为扁平的模型).图6所示的“求是”鹰模型是取 $R_{i,c}$ 为定长 0.15、取 z 方向指数 n 为 4 生成的结果.

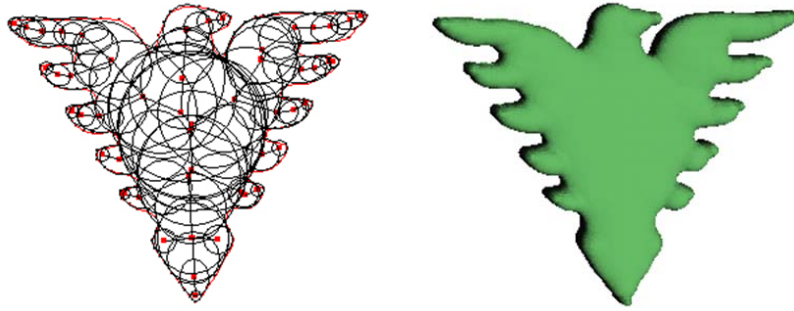


Fig.6 Creation of the “QiuShi” eagle model

图6 “求是”鹰的三维模型的构造

2.4 曲面优化

初始的元球模型建立之后,其定义的等势曲面并不能保证对轮廓线的良好逼近.为了得到比较理想的造型效果,我们需要通过优化算法进一步调整元球系统中每个元球的参数.由于我们的优化目的是为了轮廓线上的采样点集都位于由 $F(p)-t=0$ 定义的元球曲面上,超椭球原始体在 z 轴方向的主轴大小及指数值 $R_{i,c},n$ 由用户设定,将不影响曲面对轮廓线的逼近结果.因此,需要优化的参数为 $R_{i,a},R_{i,b}$ 和 ω_i .

我们取用户的原始输入点集为采样点集 M ,并定义能量方程:

$$E_{eng} = \frac{1}{m} \sum_p (F(p) - t)^2 \quad (13)$$

其中, p 是采样点集 M 中的一个采样点, m 为采样点集 M 中采样点的个数.我们利用下山单纯形法对能量方程(13)求极小值,即可得到优化后的元球参数 $R_{i,a},R_{i,b},\omega_i$.其中, $R_{i,a},R_{i,c}$ 的初始值取为前面得到的椭圆长短轴半径, ω_i 取 1.0.在实验中我们发现,当元球数量较多的时,全局优化是比较费时的工作.因此,我们采用了在每个原始体参数逐个进行优化的方法,通常在优化 3~5 次后,可以得到比较理想的效果.图7中给出分别优化 1 次和 5 次的海豚模型的效果图.不难发现,优化后的模型形状有明显改善.对于图 7(a)给定的海豚轮廓线及生成的中轴,图 7(b)所示为没有经过优化而得到的模型,其能量值为 0.23.优化 1 次后,其能量值为 0.174,如图 7(c)所示.而图 7(d)所示为经过 5 次优化后获得的模型,其能量值为 0.031,用时 0.109s.

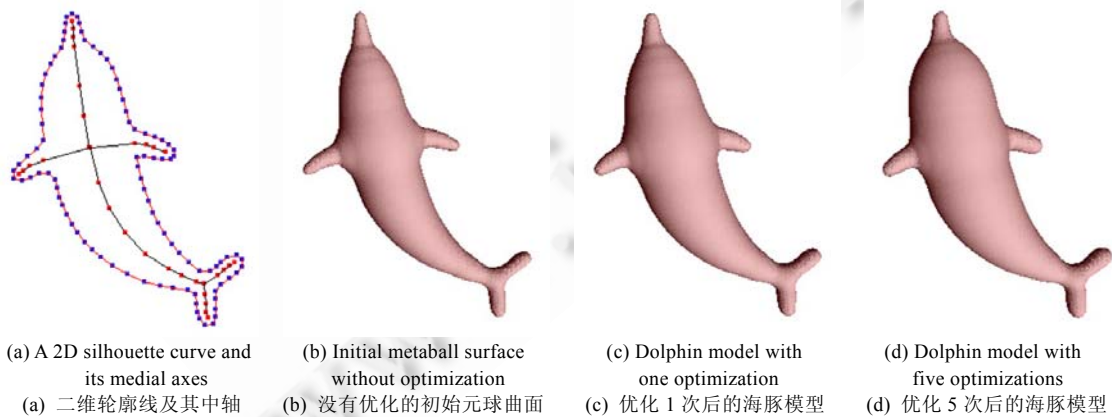


Fig.7 Dolphin model constructed with the spherical metaballs

图7 使用球型元球模型构造的海豚模型

3 实验结果

本文提出了一种基于二维轮廓线的元球造型方法.该方法由一条二维轮廓线创建一个元球系统,从而获得一个三维模型.对于复杂模型,可以通过对模型各个组成部分的轮廓线各自创建其元球系统进行布尔加/减运算获得,如图 11~图 13 所示.由于我们的算法中使用了超椭球型元球表示方法,从而能够产生扁平模型,如图 6、图 12、图 13 所示.我们在配置为 Intel Core Duo CPU T2450 2.0GHz CPU+1G RAM 的笔记本电脑上实现了该造型方法.下面给出一些实例,图片是通过渲染隐式曲面多边形化^[28]后的网格模型得到的,相关的统计数据见表 1.

Table 1 Statistics for the test data

表 1 测试数据列表

Model	Samples number on the silhouette	Medial circular or elliptical approximation		Time(s)			Energy
		Type	Number	Time for medial circles and ellipses generation	Time for parameters optimization	Total	
Fig.7(c)	95	Circle	92	0.016	0.109	0.125	0.031
Fig.8(a)	95	Circle	40	0.016	0.062	0.1	0.044
Fig.8(b)	95	Circle	62	0.031	0.078	0.11	0.032
Fig.8(c)	95	Ellipse	39	0.016	0.188	0.21	0.032
Fig.9(a)	110	Ellipse	41	0.063	0.235	0.3	0.048
Fig.9(b)	110	Ellipse	61	3.75	0.3	4.1	0.04
Fig.10	188	Ellipse	48	0.095	0.344	0.44	0.037
Fig.11	Body	Circle	67	-	0.094	0.109	0.033
	Tail	Circle	43	-	0.016	0.016	0.033
	Right back leg	Circle	80	-	0.063	0.078	0.032
	Left back leg	Circle	76	-	0.063	0.07	0.039
	Right front leg	Circle	64	-	0.046	0.047	0.036
	Left front leg	Circle	62	-	0.047	0.062	0.042
Fig.12	Board	Ellipse	34	0.031	0.219	0.25	0.009
	Human model	Ellipse	111	0.062	0.532	0.6	0.019

图 8 中的例子分别使用球型元球曲面和椭球型元球曲面构造出海豚模型的结果.其中,图 8(a)中的左图是对海豚模型的轮廓线进行中轴抽取和 40 个圆逼近的结果,图 8(b)中的左图是对相同的轮廓线在图 8(a)中左图的基础上使用 62 个圆进行逼近的结果,图 8(c)图中的左图是在图 8(a)的圆逼近的基础上使用 39 个椭圆进行逼近的结果.这个例子中,曲面参数设置为 $R_{i,c} \leq 0.4, n=2$.从图 8 构造的模型结果可以看出,使用椭球型元球模型尽管原始体数量更少,但构造结果更好,曲面更光滑.如果要使球型元球曲面达到类似于椭球型元球曲面的逼近效果,大约需要 62 个圆去逼近轮廓线.

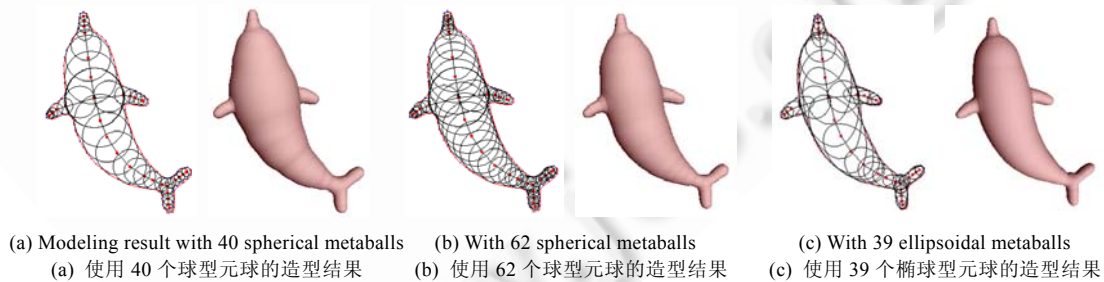


Fig.8 Comparison between the models using the spherical and ellipsoidal metaballs

图 8 使用球与椭球型元球模型进行造型的结果比较

图 9 所示为分别采用不同的椭圆生成算法获得的轮廓线的椭圆逼近及最终的椭球型元球模型建模的结果(曲面参数 $R_{i,c} \leq 0.4, n=2$).由实验结果我们发现,采用基于原始输入点和基于中轴线算法生成的模型效果相差不大,而基于中轴线的算法在效率上大大优于基于原始输入点的算法.图 9(a)中基于中轴线的椭圆生成算法生成

41个椭圆原始体,耗时63ms;而图9(b)中基于采样点的椭圆生成算法生成61个椭圆原始体,耗时3750ms.因而在后面的造型过程中,我们采用基于中轴线的算法生成轮廓线的椭圆逼近.图10给出的是手模型的构造结果,其中,轮廓线上有188个采样点,使用基于中轴线的算法生成48个椭圆,耗时32ms.其超椭球型元球模型的参数取值为 $R_{i,c} \leq 0.3, n=2$.

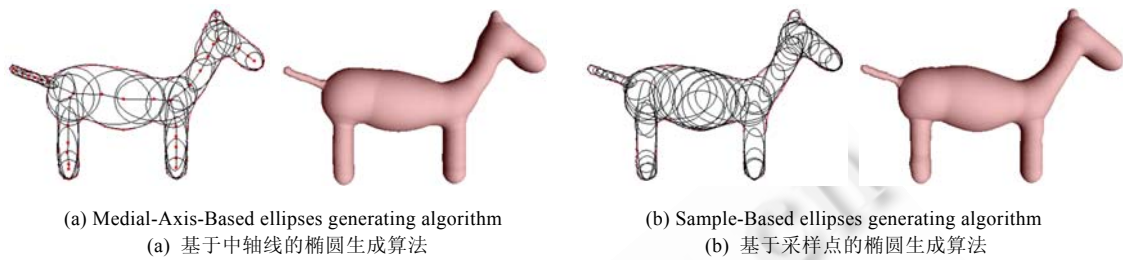


Fig.9 Modeling results using different algorithms to approximate the given 2D silhouette curve
图9 采用不同的椭圆生成算法逼近二维轮廓线进行元球模型造型的结果比较

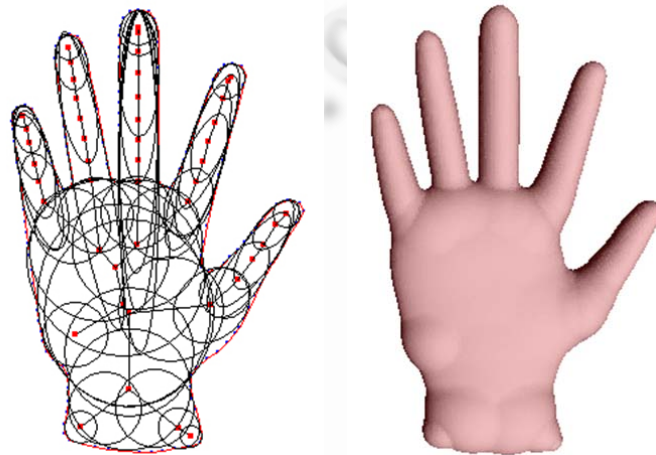


Fig.10 Hand model constructed using the ellipsoidal metaballs
图10 使用椭球型元球模型构建出的手的模型

由于元球隐式曲面具有光滑拼接的性质,因而对于具有复杂形状模型,我们将其分成多个组成部分.在不同的视线方向勾画其不同组成部分的二维轮廓线,并逐步地将这些由二维轮廓线生成的组件添加到已完成的模型中,最终组合成一个具有光滑表面的复杂模型.对于每一条二维轮廓线,独立生成一张元球隐式曲面,用户可以修改参数调节曲面的形状或者进行一些基于组件的编辑.我们利用造型方法的这种特性,一个具有比较复杂结构的马的模型构造出来了,如图11所示.图11(a)给出了马的各个组成部分的轮廓线,图11(b)是从不同视角看到的将几个组成部分组合在一起后构造出的马模型.模型的所有组件的元球曲面都是使用球型元球曲面构造的,即元球曲面参数为 $R_{i,c}=R_{i,a}=R_{i,b}, n=2$.

图12给出的例子体现出我们的造型方法在生成准扫成体、不同模型间进行布尔加/减操作的能力.我们使用图12(a)给出的两条轮廓线 L_1 和 L_2 构造出两个独立的元球曲面,并在两个元球曲面之间进行布尔加(如图12(b)所示)和布尔减(如图12(c)、图12(d)所示)运算.对于轮廓线 L_1 ,图12(b)~图12(d)中,设置所有元球 z 方向半径为 $R_{i,c}=0.1, z_{c_i}=0, n=4$,从而生成一块厚度为0.2的圆角矩形的薄板.对于 L_2 ,在图12(b)中,我们设置所有元球 z 方向半径为 $R_{i,c}=0.05, z_{c_i}=0.1, n=6$;图12(c)中, $R_{i,c}=0.05, z_{c_i}=0.05, n=6$;图12(d)中, $R_{i,c}=0.1, z_{c_i}=0, n=6$.

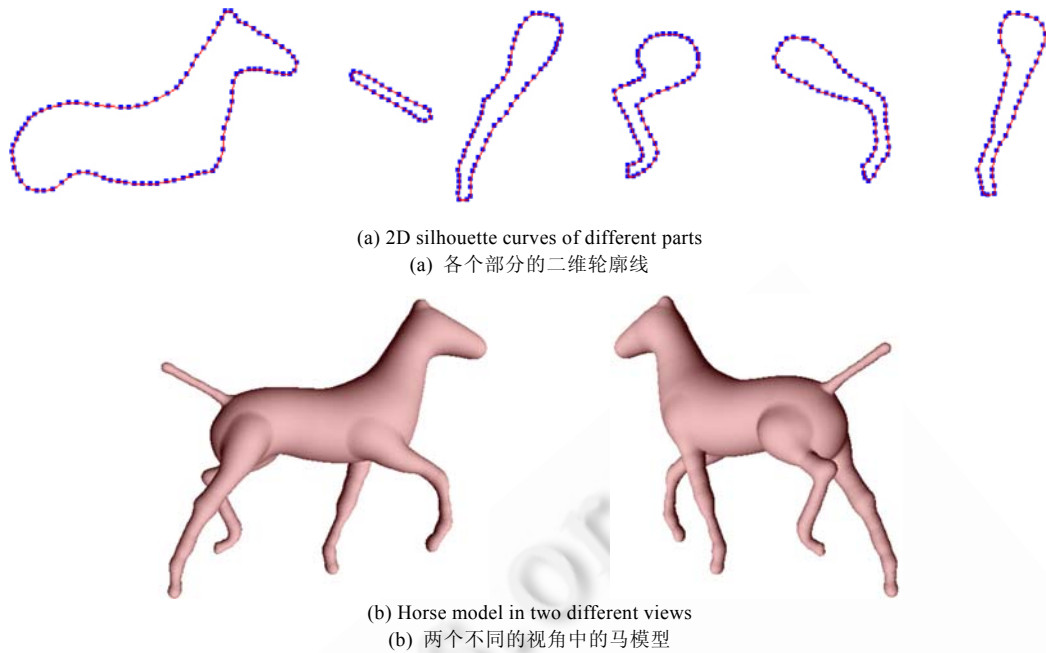


Fig.11 A horse model constructed with multiple silhouette curves

图 11 使用多个二维轮廓线构造出一个马的模型

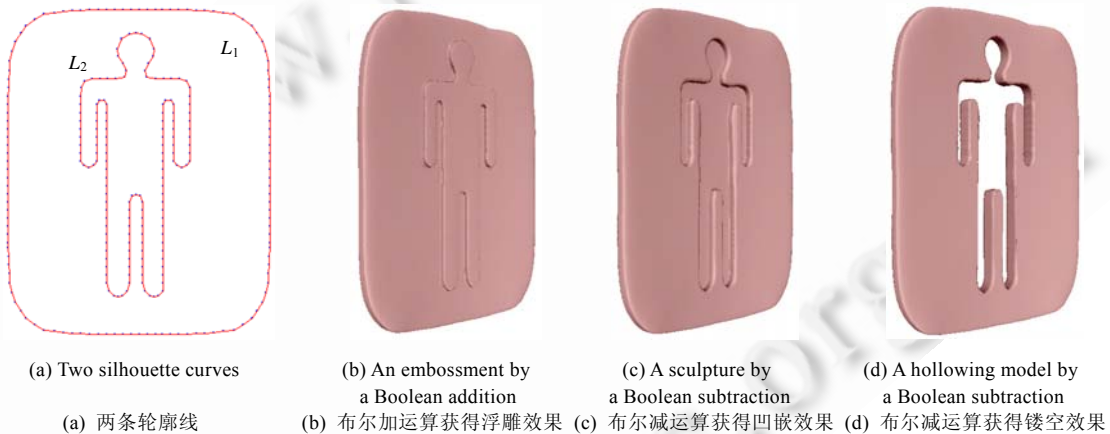


Fig.12 Different results generated from two silhouette curves by Boolean operations

图 12 由两条轮廓线生成的元球模型进行布尔运算得到的效果

4 结 论

本文提出了一种基于二维轮廓线和元球隐式曲面的造型方法.首先,由用户在交互界面上勾画或者由一幅图像中抽取模型或复杂模型组件的外形轮廓线;然后,使用二维原始体(圆或者椭圆)逼近二维轮廓线;最后,由用户对每个二维的原始体,设定他们在空间中第三个维度的信息,形成元球系统的三维原始体,对系统中元球的形状参数进行优化后,构造出轮廓逼近给定二维轮廓线的三维模型.通过实验我们发现,相比于轮廓线的球逼近,使用椭圆来逼近二维轮廓线可以减少原始体的个数,并能够提供更好的逼近精度.文中给出了两个生成轮廓线的椭圆表示的方法:一种是在文献[24]中的基于中轴线的圆表示生成算法的基础上扩展的算法,一种是基于

二维轮廓线原始数据的算法.为了生成三维模型,我们使用超二次曲面作为原始体来进行模型构造,并且提供了空间点场值计算公式.在实际造型过程中,我们采用超椭球型元球隐式曲面.由元球隐式曲面和超二次曲面的性质,使用我们的方法进行建模,可以生成一些扁平,准扫成体模型如图 6 所示,并能实现布尔加、减操作如图 11、图 12 所示.

然而,我们的造型方法还具有一定的局限:(1) 使用超椭球型元球隐式曲面依然很难精确表示模型的尖锐和扁平的特征;(2) 由一条轮廓线构造出的模型是平面对称的,尽管能通过布尔加/减运算实现复杂形状的造型;(3) 不能进行旋转造型.克服或改善这些局限,将是我们未来工作的一个方向.另外,由于元球模型能很好地表示流体的形状和模拟流体的运动,我们将在以后的工作中采用本文提出的造型方法构造流体模型(如水、云、烟雾等),深入挖掘元球模型在游戏、动画等领域中的新的应用.

References:

- [1] Blinn JF. A generalization of algebraic surface drawing. *ACM Trans. on Graphics*, 1982,1(3):235–256. [doi: <http://doi.acm.org/10.1145/357306.357310>]
- [2] Nishimura H, Hirai M, Kawai T, Kawata T, Shirakawa I, Omura K. Object modelling by distribution function and a method of image generation. *The Trans. of the Institute of Electronics and Communication Engineers of Japan*, 1985,J68-D(4):718–725.
- [3] Wyvill G, McPheeters C, Wyvill B. Data structure for soft objects. *The Visual Computer*, 1986,2(4):227–234. [doi: 10.1007/BF01900346]
- [4] Tatsumi H, Takaoki E, Omura K, Fujito H. A new method for 3D reconstruction from serial sections by computer graphics using meta-balls: Reconstruction of hepatoskeletal system formed by ito cells in the cod liver. *Computers and Biomedical Research*, 1990, 23(1):37–45. [[http://dx.doi.org/10.1016/0010-4809\(90\)90005-W](http://dx.doi.org/10.1016/0010-4809(90)90005-W)]
- [5] Muraki S. Volumetric shape description of range data using blobby model. *Computer Graphics*, 1991,25(4):227–235. [doi: <http://doi.acm.org/10.1145/122718.122743>]
- [6] Max NL, Wyvill B. Shapes and textures for rendering coral. In: Patrikalakis NM, ed. *Proc. of the Scientific Visualisation of Physical Phenomena*. Springer-Verlag, 1991. 333–343.
- [7] Payne BA, Toga AW. Distance field manipulation of surface models. *IEEE Computer Graphics and Applications*, 1992,12(1): 65–71. [doi: <http://dx.doi.org/10.1109/38.135885>]
- [8] Reed T, Wyvill B. Visual simulation of lightning. In: *Proc. of the 21st Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH'94)*. New York: ACM, 1994. 359–364. [doi: <http://doi.acm.org/10.1145/192161.192256>]
- [9] Tsingos N, Bittar E, Gascuel MP. Implicit surfaces for semi-automatic medical organ reconstruction. In: Earnshaw R, Vince J, eds. *Proc. of the Computer Graphics: Developments in Virtual Environments*. London: Academic Press Ltd., 1995. 3–15.
- [10] Bittar E, Tsingos N, Gascuel MP. Automatic reconstruction of unstructured 3D data: Combining a medial axis and implicit surfaces. *Computer Graphics Forum*, 1995,14(3):457–468. [doi: 10.1111/j.1467-8659.1995.cgf143_0457.x]
- [11] Jin XG, Liu SJ, Wang CCL, Feng JQ, Sun HQ. Blob-Based liquid morphing. *Computer Animation and Virtual Worlds*, 2005, 16(3-4):391–403. [doi: <http://dx.doi.org/10.1002/cav.v16:3/4>]
- [12] Liu SJ, Jin XG, Wang CCL, Hui KC. Ellipsoidal-Blob approximation of 3D models and its applications. *Computers & Graphics*, 2007,31(2):243–251. [doi: 10.1016/j.cag.2006.12.004]
- [13] Igarashi T, Matsuoka S, Tanaka H. Teddy: A sketching interface for 3D free-form design. In: *Proc. of the 26th Annual Conf. on Computer Graphics and Interactive Techniques*. New York: ACM Press/Addison-Wesley Publishing Co., 1999. 409–416. [doi: <http://doi.acm.org/10.1145/311535.311602>]
- [14] Igarashi T, Hughes JF. Smooth meshes for sketch-based freeform modeling. In: *Proc. of the 2003 Symp. on Interactive 3D Graphics*. New York: ACM, 2003. 139–142. [doi: <http://doi.acm.org/10.1145/641480.641507>]
- [15] Karpenko O, Hughes J, Raskar R. Free-Form sketching with variational implicit surfaces. *Computer Graphics Forum*, 2002,21(3): 585–594. [doi: 10.1111/1467-8659.t01-1-00709]
- [16] Araújo B, Jorge J. Blobmaker: Freeform modelling with variational implicit surfaces. In: *Proc. of the 12th Portuguese Computer Graphics Meeting*. 2003. 17–26.

- [17] Alexe A, Gaildrat V, Barthe L. Interactive modelling from sketches using spherical implicit functions. In: Proc. of the 3rd Int'l Conf. on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa. New York: ACM, 2004. 25–34. [doi: <http://doi.acm.org/10.1145/1029949.1029953>]
- [18] Tai CL, Zhang HX, Fong J. Prototype modeling from sketched silhouettes based on convolution surfaces. Computer Graphics Forum, 2004,23(1):71–84. [doi: 10.1111/j.1467-8659.2004.00006.x]
- [19] Owada S, Nielsen F, Nakazawa K, Igarashi T. A sketching interface for modeling the internal structures of 3D shapes. In: Butz A, Krüger A, Olivier P, eds. Proc. of the 3rd Int'l Conf. on Smart Graphics. LNCS, Berlin, Heidelberg: Springer-Verlag, 2003. 49–57. [<http://dx.doi.org/10.1145/1185657.1185773>]
- [20] Schmidt R, Wyvill B, Sousa MC, Jorge JA. Shapeshop: Sketch-Based solid modeling with blobtrees. In: Proc. of the Eurographics Workshop on Sketch-Based Interfaces and Modeling. 2005. 53–62.
- [21] Li QD, Tian J. 2D piecewise algebraic splines for implicit modeling. ACM Trans. on Graphics, 2009,28(2):1–19. [doi: <http://doi.acm.org/10.1145/1516522.1516524>]
- [22] Murakami S, Ichihara H. On a 3D display method by metaball technique. Journal of the Electronics Communication, 1987,J70-D(8):1607–1615.
- [23] Ranjan V, Fournier A. Matching and interpolation of shapes using unions of circles. Computer Graphics Forum, 1996,15(3):129–142. [doi: 10.1111/1467-8659.1530129]
- [24] Lee DT. Medial axis transformation of a planar shape. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1982,4(4):363–369. [doi: 10.1109/TPAMI.1982.4767267]
- [25] Shewchuk JR. Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. In: Proc. of the 1st Workshop on Applied Computational Geometry. 1996. 124–133.
- [26] Sun L, Jin XG, Feng JQ, Peng QS. 3D metaball modeling technique based on 2D silhouette curves. Journal of System Simulation, 2003,15(Suppl.):96–98 (in Chinese with English abstract).
- [27] Bischoff S, Kobbelt L. Ellipsoid decomposition of 3D-models. In: Proc. of the Int'l Symp. on 3D Data Processing, Visualization, and Transmission. New York: IEEE Computer Society Press, 2002. 480–488. [doi: <http://doi.ieeecomputersociety.org/10.1109/TDPVT.2002.1024103>]
- [28] Bloomenthal J. Polygonization of implicit surfaces. Computer Aided Geometric Design, 1988,5(4):341–355. [doi: [http://dx.doi.org/10.1016/0167-8396\(88\)90013-1](http://dx.doi.org/10.1016/0167-8396(88)90013-1)]

附中文参考文献:

- [26] 孙雷,金小刚,冯结青,彭群生.基于二维轮廓线的三维元球造型技术.系统仿真学报,2003,15(增刊):96–98.



刘圣军(1979—),男,江西都昌人,博士,讲师,主要研究领域为几何造型,隐式曲面建模,计算机动画.



林俊聪(1981—),男,博士,主要研究领域为数字几何处理,人机交互.



金小刚(1968—),男,博士,研究员,博士生导师,CCF会员,主要研究领域为计算机动画,计算机图形,隐式曲面计算.



冯结青(1969—),男,博士,研究员,博士生导师,CCF会员,主要研究领域为计算机图形学,几何造型,科学数据可视化.