

方程曲面的一种绘制算法

汪国华,于俊清,廖兆存

(华中科技大学 计算机科学与技术学院,湖北 武汉 430074)

(wangguohua0012@126.com)

摘要:提出了显式方程和参数方程表示的曲面的一种绘制算法,该算法可以绘制任意复杂的方程曲面。首先将方程曲面离散成四边形网格,然后采用消隐算法消除曲面的自隐藏线。消隐算法引入遮挡多边形的概念,利用遮挡多边形判断曲面上离散点的隐藏性,同时将潜在隐藏线段与遮挡多边形求交,从而得到该线段的实际隐藏部分。该绘制算法对以网格模型显示三维物体具有通用性。

关键词:曲面;消隐;方程;网格模型

中图分类号: TP391.41 **文献标识码:** A

Algorithm for drawing the surface of equation

WANG Guo-hua, YU Jun-qing, LIAO Zhao-cun

(College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan Hubei 430074, China)

Abstract: An algorithm which can draw any complex surfaces expressed by explicit equation or parametric equation was given. Firstly, the surface was dispersed into quadrangular meshes, then, the self-hidden lines were removed from the surface by the algorithm of removing hidden lines. In the algorithm of removing hidden lines, the conception of hiding-polygon was introduced. By judging whether the dispersed vertexes on the surface should be hidden or not, and getting the intersecting points between the potential hidden lines and the hiding-polygons, the actual hidden lines which were the parts of the potential hidden lines can be obtained. The experimental results have verified the effectiveness of the proposed algorithm, and it is generalizable to 3D objects displayed in meshes model.

Key words: surface; hidden lines removal; equation; meshes model

0 引言

在现实生活中,经常需要根据方程产生曲面,这些曲面是基于三维坐标系的。曲面方程的形式有参数形式和非参数形式两种^[1]。非参数形式有显式表示和隐式表示之分,在实际的工程中,经常使用参数形式和显式形式。为了用计算机来显示这些曲面,需要对曲面进行透视、消隐,并进行颜色填充和光照效果处理,最终得到三维的真实感图形。其中消隐(即消除隐藏线和消除隐藏面)是生成三维真实感图形的关键步骤之一。当用笔绘图仪或其他画线设备绘制图形时,要解决的主要是消除隐藏线问题。曲面根据实际需要有多种不同的显示形式,有以网格形式显示的,也有以轮廓线形式显示的。其中多数曲面是不需要消隐的,绘制方法较简单,但不能解决较复杂的方程曲面的绘制问题。本文提出了基于四边形网格模型的方程曲面的一种绘制算法,适用于任意显式形式和参数形式的方程曲面,能解决任意复杂的方程曲面的绘制问题。

1 根据曲面方程绘制未消隐的曲面

1.1 显式形式

显式形式的曲面方程根据未知变量和第三维的不同有三种不同的形式,分别如下:

$$z = f(x, y), X_1 \leq x \leq X_2, Y_1 \leq y \leq Y_2 \quad (1)$$

$$x = f(y, z), Y_1 \leq y \leq Y_2, Z_1 \leq z \leq Z_2 \quad (2)$$

$$y = f(x, z), X_1 \leq x \leq X_2, Z_1 \leq z \leq Z_2 \quad (3)$$

下面以式(1)为例来说明这种表示方式生成未消隐曲面的方法:

(1) 将 $x(X_1 \leq x \leq X_2)$ 分成 m 等分(离散化),取等分点处的值,分别记为: $x_0, x_1, x_2, \dots, x_i, \dots, x_j, \dots, x_{m-1}$ 。

(2) 将 $y(Y_1 \leq y \leq Y_2)$ 分成 n 等分(离散化),取等分点处的值,分别记为: $y_0, y_1, y_2, \dots, y_i, \dots, y_j, \dots, y_{n-1}$ 。

(3) 对于任意离散化的 x 值(如 x_i),遍取离散化的 y 值,由这些值对,根据方程(1)计算该点处的 z 值。如值对 (x_i, y_i) 处的 z 值为 $z_{i,j} = f(x_i, y_j)$ 。点 $p(x_i, y_j, z_{i,j})$ 即为方程曲面上的一点,所有这些点形成一个空间点阵 $ptOrig[m][n]$,其中点 $ptOrig[i][j]$ 即为点 $p(x_i, y_j, z_{i,j})$ 。

(4) 选择好视点和透视平面,对曲面上的所有点进行透视,使三维的点阵变成二维的点阵 $ptPers[m][n]$,点 $ptPers[i][j]$ 为点 $ptOrig[i][j]$ 透视所得。

(5) 按如下方式连接所有点,形成网格中的横线。

```
for(i=0; i<m; i++)
for(j=0; j<n-1; j++)
{ Line(ptPers[i][j], ptPers[i][j+1]); }
```

(6) 按如下方式连接所有点,形成网格中的竖线。

```
for(j=0; j<n; j++)
for(i=0; i<m-1; i++)
{ Line(ptPers[i][j], ptPers[i+1][j]); }
```

至此,一个基于四边形网格模型的未消隐的方程曲面就生成了。

1.2 参数形式

参数形式是曲面方程最常用的表示形式,如 Coons 曲面、

收稿日期:2006-06-12;修订日期:2006-09-06

作者简介:汪国华(1979-),男,湖北红安人,硕士研究生,主要研究方向:文档信息处理、图形和图像处理; 于俊清(1975-),男,内蒙赤峰人,副教授,博士,主要研究方向:多媒体信息处理与检索; 廖兆存(1967-),男,湖北恩施人,教授,主要研究方向:文字信息处理。

Bézier 曲面、B 样条曲面和 NURBS 曲面等,都是用参数表示的曲面。参数曲面的一般形式如下:

$$\begin{cases} x = f(u, v) \\ y = g(u, v) \\ z = h(u, v) \end{cases} \quad U_1 \leq u \leq U_2, V_1 \leq v \leq V_2 \quad (4)$$

由其绘制未消隐曲面的方法与显式方程形式绘制曲面的方法一样。首先将 $u (U_1 \leq u \leq U_2)$ 离散为 m 等分 $(u_0, u_1, \dots, u_i, \dots, u_{m-1})$, 将 $v (V_1 \leq v \leq V_2)$ 离散为 n 等分 $(v_0, v_1, \dots, v_j, \dots, v_{n-1})$ 。对离散化的 u 值,遍取离散化的 v 值,由这些值对,根据方程(4) 分别计算该点处的 x, y, z 值。如由值对 (u_i, v_j) 即可得到曲面上一点 $p(f(u_i, v_j), g(u_i, v_j), h(u_i, v_j))$ 。然后选择适当的视点和透视平面,将三维点阵透视为二维点阵。最后按照前述方法连接透视后的二维点阵形成横线和竖线。

2 对曲面进行消隐

由上述方法生成的曲面没有经过消隐处理,会有许多隐藏的线和面被画出来,看起来很不真实。因此,在透视后必须经过消隐。凸多面体和凹多面体已有比较成熟的消隐算法。而对于曲面的消隐,通常是将曲面离散表示为多边形(如三角形或者四边形),然后采用多面体的消隐算法进行消隐。这种方法没有考虑曲面自身的特点,计算量大,效率低。下面介绍一种新的消隐算法,其基本思想是判断透视后的点阵中被遮挡的点(称为隐藏点),并求出曲面上相邻两点之间的线段与曲面的交点,最后将所有非隐藏点和交点按照其在曲面上的顺序连接起来,就形成了消隐之后的方程曲面。消隐算法的实现步骤为:

(1)对透视后的点阵(这里选择 YOZ 平面作为透视平面)按照其在原始点阵中的 X 方向分量由大到小进行排序(即透视前的点的 X 方向分量的大小为透视后点阵的排序条件)。排序是基于这样的事实,即曲面上离透视中心越近的点越容易被曲面遮挡。排序后,前面的点在视线方向上总是比后面的点离透视中心远。因此,按这种顺序依次处理所有的点,后处理的点如果落在前面的点形成的多边形内,则后面的点必然被前面的点所在的部分曲面遮挡。这里 X 轴负向是视线方向。对于排序,当 x 为自变量时(式(1)和式(3)情况),按前述方法计算的透视点阵逆序,可以改变原始点阵的计算顺序使透视点阵顺序。当 x 不为自变量时(式(2)和式(4)情况),为提高计算速度,可以采用快速排序^[4,5]。

(2)对排序后的所有点,依顺序依次进行下列处理:

第一步:判断该点(当前点)的隐藏性。如果当前处理的点被某遮挡多边形(本算法中采用四边形)遮挡,则该点是隐藏的,否则该点是非隐藏的。曲面上已处理的相邻点所形成的最小四边形网格在透视平面上的透视图就是遮挡四边形。很显然,表面上的四边形网格是三维的,而透视平面上的遮挡四边形却是二维的。

在这一步中,为提高判断当前点是否为某四边形遮挡的计算效率,可以采用包围盒检验。这里的包围盒检验,指的是包围四边形的最小矩形。先判断当前点是否在包围所有遮挡四边形的最小矩形内,如果在矩形之外,则该点在所有四边形之外,即可断定该点非隐藏。只有当前点在该矩形内才进一步判断其是否在某四边形内,这一步仍然可以采用包围盒检验。先判断当前点是否在包围该四边形的最小矩形之内,只有当前点在该矩形内才进一步判断当前点是否在该四边形内。

第二步:求出该点与已处理相邻点之间的线段(这样的

线段称为潜在隐藏线段)与曲面的交点。

这里的相邻点是指曲面上与当前点在同一最小四边形网格上,且与当前点共边的另一点。假设透视点阵为 $ptPers[m][n]$,则某透视点 $ptPers[i][j]$ 的相邻点都存在的时候有四个,分别是左相邻点 $ptPers[i-1][j]$ (当 $i < 1$ 时,该点不存在),右相邻点 $ptPers[i+1][j]$ (当 $i+1 \geq m$ 时,该点不存在),上相邻点 $ptPers[i][j+1]$ (当 $j+1 \geq n$ 时该点不存在),下相邻点 $ptPers[i][j-1]$ (当 $j < 1$ 时该点不存在)。如图 1 所示。

$P_{i-1,j+1}$	$P_{i,j+1}$	$P_{i+1,j+1}$
$P_{i-1,j}$	$P_{i,j}$	$P_{i+1,j}$
$P_{i-1,j-1}$	$P_{i,j-1}$	$P_{i+1,j-1}$

图 1 相邻点

曲面上任意两相邻点之间的线段很可能被曲面自身所遮挡,所以必须将其与曲面上的多边形求交。将所有已形成的遮挡四边形,依次与该线段求交,并记下每次求交后的隐藏线段(记下线段端点),最后对所有非隐藏线段取并集,即可得到该线段与曲面的交点。在求交时,以四边形形成的相反顺序与线段求交,即后形成的四边形先求交。这是因为先形成的四边形可能遮挡了后形成的四边形,后形成的四边形与线段的交点可能会被先形成的四边形遮挡。线段与四边形求交,实际上是线段与各四边形边求交。

设潜在隐藏线段的端点是 $p_1(x_1, y_1)$ 和 $p_2(x_2, y_2)$,则可将其参数化为(5),同时设透视到屏幕上的遮挡四边形为 $q_1q_2q_3q_4$,则该四边形可表示为(6)。

$$p(t) = p_1 + (p_2 - p_1)t, 0 \leq t \leq 1 \quad (5)$$

$$q(s_i) = q_i + (q_{i+1} - q_i)s_i, 0 \leq s_i \leq 1 \quad (6)$$

由(5)和(6)构造方程(7):

$$p_1 + (p_2 - p_1)t = q_i + (q_{i+1} - q_i)s_i \quad (7)$$

由该方程即可求出线段与多边形的交点^[6]。如果线段与四边形任意一条边都无交,则线段被四边形完全遮挡或者线段与四边形分离。如果存在交点,那么这些交点会将线段的参数区间 $[0, 1]$ 分割为若干个子区间,每个子区间对应于一条子线段。每条子线段上的所有点的隐藏性是相同的。与某四边形求交后,滤去隐藏的线段,将非隐藏的线段继续与其他多边形求交。待线段与所有多边形求交完后,将所有子线段进行合并,并由此得到交点。

为减少不必要的计算,提高计算效率,可以采用包围盒检验技术和深度检验技术^[7]。包围盒技术前面已经提到过,只有线段与包围四边形的最小矩形有交,或者线段完全落在矩形之内时,线段才可能与四边形有交。而深度检验是指判断线段和四边形在视线方向上的前后关系。当整个线段位于四边形之前时,线段不可能被多边形遮挡,不必求交。由于视线方向沿 x 轴负向,可以将线段的两个端点对应的原始点的 x 方向分量与四边形的四个顶点所对应的原始点的 x 方向分量进行比较。当线段的每个端点对应的原始点的 x 方向分量都比四边形的四个顶点所对应的原始点的 x 方向分量小时,整个线段在四边形之前。

第三步:判断该点是否形成了新的遮挡四边形。其方法是判断与该点能形成最小四边形网格的相邻点,如果它们都被处理则形成一个新的遮挡四边形。例如,能与点 $ptPers[i][j]$ 形成遮挡四边形的点组分别是:

$$\{ptPers[i-1][j+1], ptPers[i][j+1], ptPers[i][j],$$

```

ptPers[i-1][j]
{ptPers[i][j+1], ptPers[i+1][j+1], ptPers[i+1][j],
ptPers[i][j]}
{ptPers[i][j], ptPers[i+1][j], ptPers[i+1][j-1],
ptPers[i][j-1]}
{ptPers[i-1][j], ptPers[i][j], ptPers[i][j-1],
ptPers[i-1][j-1]}

```

某一点组能形成遮挡四边形的前提条件是点组中的点都存在,并且所有点均已被处理。这样,与当前点相关的遮挡四边形最多是四个,能形成的遮挡四边形最多也是四个。

为了减少遮挡四边形的个数,可以判断一下该遮挡四边形是否完全被其他已形成的四边形遮挡,如果是,则舍弃该遮挡四边形。其方法是判断该四边形的四个顶点是否都落在某遮挡四边形之内,并且四个顶点的深度处于该遮挡四边形之后。

至此,所有隐藏点和自隐藏线就被求出来了,消隐算法就完成了。

图2是消隐算法的流程。

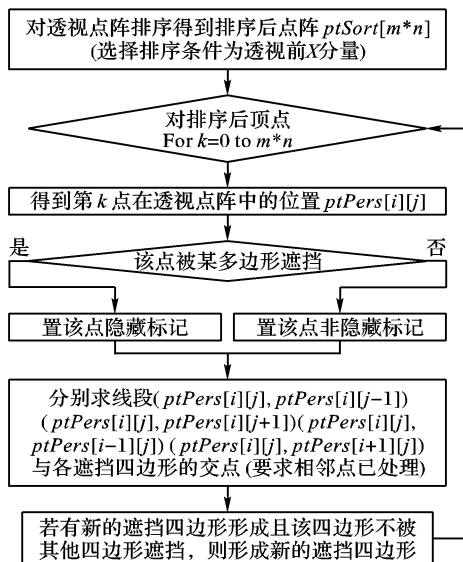


图2 方程曲面消隐算法流程

3 根据方程绘制消隐的曲面

3.1 方程曲面消隐算法的步骤

根据方程绘制消隐的曲面是在绘制未消隐曲面的基础上加上消隐处理实现的。其具体步骤如下:

(1) 输入方程和自变量(或者是参数)的取值范围。如果是显式方程,则输入显式方程和自变量的取值范围,如方程:

$$y = \sin(x \cdot y), -3 \leq x \leq 3, 0 \leq y \leq 4 \quad (8)$$

自变量 x 和自变量 y 的取值范围为分别为 $[-3, 3]$ 和 $[0, 4]$, 在输入自变量取值范围时只需要输入端点值即可。如果是参数方程,则输入参数方程和参数取值范围。

(2) 按照第1节的方法将两个自变量(或者两个参数)的取值区间分别分成 m 和 n 等分,并由这些离散的值对根据方程计算曲面上原始的离散点阵 $ptOrig[m][n]$ 。

(3) 选择适当的视点和透视平面对原始点阵进行透视,得到透视点阵 $ptPers[m][n]$ 。

(4) 按照第2节的消隐算法根据原始点阵及透视后的点阵进行消隐处理,计算出隐藏点及曲面上的自隐藏线段。

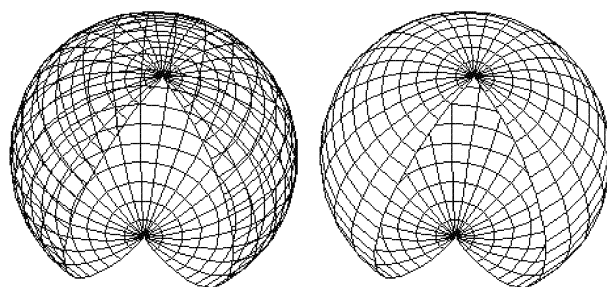
(5) 按照第1节连接曲面上离散点的方法将曲面上的非

隐藏点和交点分别按照横向和纵向的顺序连接起来,形成曲面网格的横线和竖线。在连接的过程中,首先判断该线段是完全隐藏、部分隐藏还是完全可见,线段的隐藏性在消隐过程中已经求出。如果是完全隐藏则不需要连接线段;如果是部分隐藏,则要将非隐藏部分画出来,只需要将非隐藏线段的端点相连即可,这些端点在消隐处理过程中已经求出;如果是完全可见,则连接该线段的两个端点。

3.2 算法应用实例

图3是由参数方程(9)根据上述方法绘制的未消隐曲面和消隐曲面。

$$\begin{cases} x = \cos(u) \times \sin(v) \\ y = \sin(u) \times \sin(v) \\ z = \cos(v) \end{cases} \quad 0.4\pi \leq u \leq 2\pi, 0 \leq v \leq \pi \quad (9)$$



(a) 消隐前 (b) 消隐后
图3 消隐前后的曲面

4 算法的改进

在消隐算法中,为了进一步提高计算效率,可以做如下改进:将其中的遮挡四边形合并成遮挡多边形。由于将遮挡四边形合并成遮挡多边形后,原来四边形中的许多边在合并之后处在多边形之中而被去掉,从而与潜在隐藏线段求交的边数减少,这样就可减少线段的求交次数。同时,在判断离散点的隐藏性时,由于很多四边形被去掉,这些去掉的四边形不需要参与对离散点的遮挡性判断,因此判断离散点的隐藏性的计算量会大大减少。

在将遮挡四边形合并成遮挡多边形的时候,首先判断该四边形是否被某遮挡多边形完全遮挡,如果被完全遮挡,则去掉该四边形,判断方法与前面提到的判断四边形是否被某四边形完全遮挡的方法是一样的。否则,判断该四边形是否与任何遮挡多边形没有重叠的部分,如果是,该四边形为孤立四边形,不能与任何已形成的遮挡多边形合并,则将该四边形作为一个新的遮挡多边形,判断方法是判断该四边形是否与已形成的遮挡多边形没有公共边或者没有任何顶点落在某遮挡多边形内。如果不是上述两种情况,则表明该四边形与其他遮挡多边形有公共边或者有交点,此时应该将该四边形与其他多边形进行合并,去掉公共边以及被其他多边形遮挡的部分边。由四边形合并成多边形的算法较复杂,还需要在这方面做进一步的研究。

5 结语

本文所述的方法在 ScienceWord 软件中得到应用,其效果非常好。在由曲面方程产生原始点阵后,可以对原始点阵进行各种几何变换(如旋转、缩放、平移等),从而得到同一图形的不同表现形式,这种变换方式,要求视点固定不动。另一

(下转第 2889 页)

当所有子类超球面分类模型对待识样本的判别标记都为 -1 时,待识样本为异常样本;否则,为正常样本(即 k 个子类超球面中任意一个或几个对待识样本判别为 $+1$ 时)。

图 2 中虚线圆表示一类支持向量机的单超球面的平面示意图。从图中可以看出,此超球面错误地判别了一些非正常样本,而改进的一类支持向量机子类超球面(实线圆)则可以有效地识别出原来落于单超球面(虚线圆)中的错分样本。

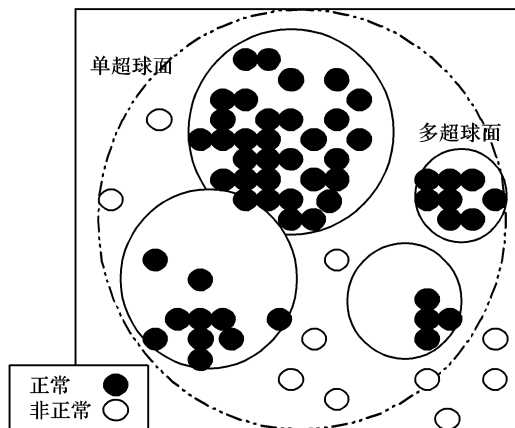


图 2 多超球面一类支持向量机

3.2 算法实现

本文在 Matlab 6.5 环境下,通过改进文献[5]的软件包,实现了多超球面一类支持向量机算法。

Libsvm 结合 SMO^[6]与 SVM^{light}^[7]算法实现了快速解决二次优化问题。它利用 SVM^{light}工作集的思想,在每次循环中,将训练样本分为两个集合 B 和 N , B 为工作集 $B \subset \{1, \dots, n\}$, $N = \{1, \dots, n\} \setminus B$ 。Libsvm 采用工作集的极端情况,每次只优化两个 Lagrange 乘数 α_i, α_j (SMO 算法),从而降低了解二次优化问题的难度。其主要优势在于可以从中得到一个解析解,而不需要优化软件求解。

把 Libsvm 的思想应用到一类超球面支持向量机上,可快速建立一类超球面支持向量机分类器。

4 实验结果及分析

表 1 多超球面一类支持向量机检测精度

软件	训练	测试	$k=1$	$k=2$	$k=3$	$k=4$	$k=6$	$k=8$
F5r11	Data0	Data0 1656	85.45	100.00	100.00	100.00	100.00	93.01
	1000	Data1 2650	20.29	63.21	44.27	39.53	32.98	17.62
H4PGP	Data0	Data1 1656	85.44	100.00	100.00	100.00	100.00	100.00
	1000	Data0 2650	13.87	63.32	64.75	44.27	39.53	11.67
Jsteg41	Data0	Data0 1656	80.13	100.00	100.00	100.00	100.00	100.00
	1000	Data1 2656	24.32	50.64	52.00	60.74	45.23	42.21

实验对 F5r11、H4PGP、Jsteg41 进行测试,分别对 2560 幅

真彩图像进行随机消息嵌入,嵌入的消息大小从最大消息嵌入量的 10%~90% 之间随机选取。对两类图像进行特征提取,提取方法采用彩色小波包预测模型^[4],每幅图像提取出一个 108 维特征向量。对正常图像进行基于核的多超球面 OC-SVM 训练,训练时的核函数取 Linear 线性核,训练样本使用数据的前 1000 个样本,建立基于单样本的分类器,聚类个数 $k \in \{1, 2, 3, 4, 6, 8\}$ 。使用分类器对正常图像和隐秘图像进行分类,结果如表 1 所示。

表 1 中, $k=1$ 表示单超球面一类支持向量机算法的检测精度, $k=n, n>1$ 表示多超球面一类支持向量机算法的检测精度。结果显示,当 k 取适当值时($k>1$),多超球面一类支持向量机算法要比单超球面算法具有更好的检测精度。在此需要说明, k 的合理选取有赖于数据集本身的特性,可以通过反复实验记录一个经验值。

5 结语

本文提出一种基于核空间 K-均值聚类的多超球面一类支持向量机分类算法。算法将多超球面思想应用到隐秘图像盲检测分类问题当中,解决了一类支持向量机建立超球面模型中样本特征差异问题带来的不便。利用 Kernel 核空间聚类,突出样本特征差异,更好地实现了多超球面模型的建立。实验数据表明,该算法比单超球面一类支持向量机具有更高的精度。

参考文献:

- [1] VAPNIK V. The Nature of Statistical Learning Theory[M]. Springer Verlag, 1995.
- [2] SCHÖLKOPF B, SMOLA A, WILLIAMSON R, et al. New support vector algorithms[J]. Neural Computation, 2000, 12(5): 1207–1245.
- [3] 孔锐,张国宣,施泽生,等. 基于核的 K-均值聚类[J]. 计算机工程, 2004, 30(11): 12–13.
- [4] LYU S, FARID H. Steganalysis Using Color Wavelet Statistics and One-class Support Vector Machines[A]. SPIE Symposium on Electronic Imaging[C]. San Jose, CA, 2004.
- [5] CHIH-CHUNG CHANG, CHIH-JEN LIN. LIBSVM: a library for support vector machines[EB/OL]. <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf>, 2001–06–08.
- [6] PLATT J. Using Sparseness and Analytic QP to Speed Training of Support Vector Machines[A]. Advances in Neural Information Processing Systems 11[C]. Cambridge, MA: MIT Press, 1999.
- [7] JOACHIMS T. Making large-Scale SVM Learning Practical[A]. Advances in Kernel Methods - Support Vector Learning[C]. MIT-Press, 1999.

(上接第 2886 页)

种变换方式是变换图形的透视点,保持图形相对坐标轴不动,从而得到不同视角的透视图。在绘制完成后,还可以对图形进行明暗效果处理,并进行色彩填充,以获得真实的效果。该绘制算法可以进一步推广到其他三维图形的绘制,如凸多面体、凹多面体等。其中的消隐算法对于采用面集表示法的其他三维物体(如立方体等)也是适用的。

参考文献:

- [1] 孙正兴,周良,郑宏源. 计算机图形学基础教程[M]. 北京:清华大学出版社,2004. 173–175.
- [2] JANSSEN LT. A simple efficient hidden line algorithm[J]. Comput-

er and Structures, 1983, 17(4): 563–571.

- [3] SOUKKERS R, LAW HKW. On the hidden line removal problem[J]. Computers and Structures, 1987, 26(4): 709–717.
- [4] 严蔚敏,吴伟民. 数据结构[M]. 北京:清华大学出版社,1997. 273–277.
- [5] PRESS WH, TEUKOLSKY SA, VETTERLING WT, et al. C++ 数值算法[M]. 第 2 版. 胡健伟,赵志勇,等译. 北京:电子工业出版社,2005. 248–251.
- [6] SCHNEIDER PJ, EBERLY DH. 计算机图形学几何工具算法详解[M]. 周长发,译. 北京:电子工业出版社,2005. 171–174.
- [7] 孙家广. 计算机图形学[M]. 第 3 版. 北京:清华大学出版社,1998. 486–488.