

文章编号 1004-924X(2002)04-0354-05

自由曲面的透视纹理映射研究

梁伟文, 宾鸿赞

(华中科技大学 机械学院, 湖北 武汉 430074)

摘要:从纹理映射的基本方法出发,分析传统多边形纹理映射存在的不足,提出了自由曲面的透视变换纹理映射。首先介绍透视变换和透视纹理映射原理,并简要分析其在自由曲面纹理映射中的应用。由于多边形纹理映射的精度问题及自由曲面表达的多样性,因此在自由曲面纹理映射中,选择双三次Bezier面片逼近自由曲面。然后详细介绍了透视纹理映射的算法步骤:先得到纹理空间参数与对象空间坐标之间的对应关系,通过计算对象空间坐标与屏幕空间之间关系,确定纹理空间与屏幕空间的对应关系。最后提出了纹理走样的处理方法。该方法不仅可以应用于自由曲面纹理映射,还可有效地应用到多面体、自由形体表面的纹理映射。

关键词:自由曲面;透视变换;纹理映射

中图分类号:TP391.4 **文献标识码:**A

1 引言

纹理映射是计算机可视化中很重要的一种技术,这种技术带给大家丰富的视觉感受。在3D模拟、虚拟现实场景、3D游戏等开发中,纹理映射得到广泛的应用。可以将纹理映射定义为:好比在曲面上贴上一张位图。对于纹理映射可以分为二维纹理映射和三维纹理映射。二维纹理映射又可以分为正向纹理映射和逆向纹理映射。正向映射可以分为独立的两步进行:从纹理空间到景物空间的映射,将原图像映射到3D景物的表面;

从物体空间到屏幕空间的映射,将景物变换成目的图像。三维纹理映射主要为了克服二维纹理映射的邻接表面间连续过渡纹理和走样纹理。二维纹理映射是纹理映射中的主要部分,二维纹理映射实际上是从二维纹理空间到三维景物表面的一个映射。二维纹理是平面空间连续或离散图形,其映射的实质是确定景物表面点与纹理空间的对应位置 (u, v) ,而 (u, v) 处定义纹理值或颜色值,即描述景物表面在 P 点处的某种纹理属性。纹理映射通常可比拟为计算机图形学中的取景变换,如透视投影、仿射变换、平行投影等。国

内外对于纹理映射研究已经很久,大多是对平面纹理映射,而对于自由型曲面的透视纹理映射研究较少,而其在建模、仿真、3D游戏开发等中得到广泛应用。本文主要研究自由曲面透视纹理映射。

2 透视纹理映射原理

根据计算机图形学定义,透视投影的视线(投影线)是从视点(观察点)出发,视线是不平行的。原理如图1。在现实生活中,当用透视方法观察时,随着视点远离对象时,对象的大小随着改变,同时表面的细节也有所变化。即透视投影使得远近的物体根据距离的不同,投影面上的投影大小不同,看上去更加真实。透视投影的本质就是对象的 z 值不同而得到不同投影对象。由于纹理映射存在很多对象(包括曲面、多面体等),所以纹理映射本身是一个相当大的、复杂的研究对象。当使用纹理时,必须全面考虑实际的需要,如自由曲面在工程中应用很广,自由曲面的纹理映射显得很重要。自由曲面的纹理映射可以把纹理映射到一系列网格控制点上,或直接映射到曲面上,也可以映射在标准面上,利用这些方法,可以实现自

由曲面的纹理映射。本文根据透视原理,将纹理映射到自由曲面的网格控制点上,实现自由曲面的纹理映射。透视纹理映射原理如图 2。

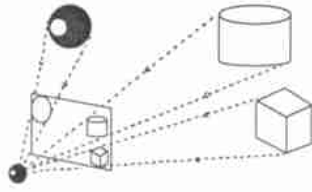


图 1 透视映射原理

Fig. 1 Principle of perspective projection.

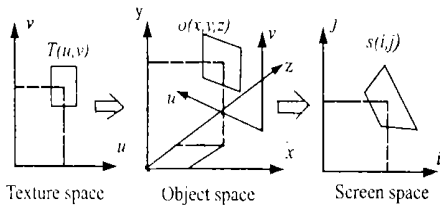


图 2 透视纹理映射原理

Fig. 2 Principle of perspective texture mapping.

3 自由曲面的逼近

在以往应用中,曲面多数用多变形逼近,但这种表示法的精度是一个很大的问题。在实际生活中,有很多物体的表面都是弯曲的,要使用多边形面片来对曲面进行很好描述,就需要多边形的数量足够地多。对大量的多边形来说,除了处理存储和处理过程十分复杂之外,对于建模阶段也不是一个十分适合的方法,同时由多边形构成的多面体的离散点不利于对表面属性进行处理,如表面上一个属性的改变(如表面曲率改变),就要求对许多点的坐标进行修改。所以要提高精度,同时又不增加建模时的难度,还要减少多边形的数量,通常使用较简单的曲面片来对一些复杂的曲面区域进行描述,如图 3 所示,为双三次曲面片。该方法实际上是平面多边形对曲面进行逼近方法的一种扩展。在计算机辅助设计中,设次数低于三的多项式称为二次多项式,其形式比较简单,但由于它只定义了一些灵活点,因此在逼近曲面时有一定的不足。另一方面,次数高于 3 的多项式虽然更加灵活,但表达起来也更加复杂,并且缺少必要的稳定性,只要稍微改变一下参数就可

能产生很大的变化,因此选用三次曲面逼近。平面多边形是曲面的一种特例,它的表达式的最高次数为 1。

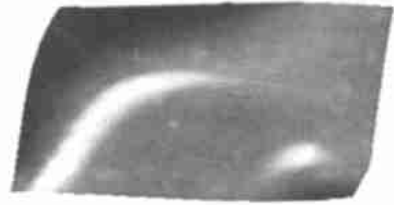


图 3 双三次曲面片

Fig. 3 Bicubic patch.

采用双三次 Bezier 曲面片作为逼近自由曲面的基片。使用双三次曲面的首要动机就是要避免使用平面多边形来对表面进行逼近,但从对象到屏幕的视觉处理过程中的最后阶段,还是使用多边形来对双三次曲面进行渲染。这样至少带来两大好处,双三次曲面在建模时是非常适合的,一是因为它操作的简便性,另一个是因为它的存储量比较小。双三次面片可以用一系列的三次曲线来进行表示,每一个面片由 16 个控制点控制。面片的表达式如下:

$$\begin{pmatrix} x(t, s) \\ y(t, s) \\ z(t, s) \end{pmatrix} = [t^3 \ t^2 \ t] [M] [G] [M]^T \begin{pmatrix} s^3 \\ s^2 \\ s \\ 1 \end{pmatrix}$$

在这个公式中,包含了两个参数: s 和 t 。 $[M]$ 是一个 4×4 的矩阵 ($[M]^T$ 是 $[M]$ 的转置矩阵), $[G]$ 是一个 4×4 的几何矩阵,它包含了面片的控制量。尽管面片的表达式可有多种形式,但在实际程序中用得最多的还是 Bezier 形式。有了面片的解析式后,可以得到面片上的一系列的离散点,根据这些离散点可以构造多变形网格,最后得到空间多边形离散点逼近。

4 透视纹理映射算法

自由曲面的透视纹理映射根据以上三次面片的逼近,可以简化为离散点形成空间多边形的纹理映射,即把纹理空间的坐标系映射到多边形坐标系。透视纹理映射可以根据多边形距离观察者的远近而进行相应的调整从而产生比仿射纹理映射更真实的效果。这里的多边形坐标系定义为:

空间不共线三点形成的向量可以定义一个坐标系,构成一个平面。如果这两个向量不垂直则将使该空间内的图形产生倾斜的效果,而当这两个向量长度不相等时,则会拉伸或压缩该空间内的图形。在一个 3 维空间的多边形上取两个不共线的向量,则这两个向量可以定义一个多边形坐标系,然后就可以写出这个 3 维空间的平面方程(2 维或 n 维空间的平面方程也可同理求得)。如在一个平面上取两点 P_1 和 P_2 可以得到一个向量 $L = P_1 - P_2$,再取一点 P_3 ,可以得到另一个向量 $M = P_1 - P_3$ 。如果将点 P_1 作为该平面空间的局部坐标原点,将该点与视见坐标系的原点相减,则得到一个向量 $N = P_1 - (0,0,0)$ 。则该平面的方程为 $aL + bM + N = 0$,其中 a, b 为任何实数。如果用矩阵表示,形式如下(其中的矩阵称为 T_1):

$$(x, y, z) = (a, b, 1) \times \begin{pmatrix} L_x & L_y & L_z \\ M_x & M_y & M_z \\ N_x & N_y & N_z \end{pmatrix}, \quad (1)$$

用纹理空间的 u, v 来代替 a, b ,则可以得到纹理空间到多边形的映射关系,其中 (L_x, L_y, L_z) 对应纹理空间的 $(1,0)$ 点, (M_x, M_y, M_z) 对应纹理空间的 $(0,1)$ 点, (N_x, N_y, N_z) 对应纹理空间的 $(0,0)$ 点。 $(x, y, z) = (u, v, 1) \times T_1$, T_1 可以根据多变形顶点坐标求出,如此便完成了纹理到多边形的映射,即完成了纹理映射的第一步。

接着是从对象空间到屏幕空间的映射。大多数情况下用扫描线法来填充多边形,完成纹理映射,所以需要知道多边形经过透视投影后其在屏幕坐标系上与纹理映射空间坐标系的映射关系,其关系可通过如下几步求得。

根据视见坐标系或世界坐标系的 y 轴方向与屏幕坐标系的 y 轴方向相反,所以矩阵 T_1 应变换为如下矩阵 T 。

$$T = \begin{pmatrix} L_x & -L_y & L_z \\ M_x & -M_y & M_z \\ N_x & -N_y & N_z \end{pmatrix}, \quad (2)$$

然后进行透视变换:

$$\left(\frac{kx}{z}, \frac{ky}{z}, k\right) = \left(\frac{ku}{z}, \frac{kv}{z}, \frac{k}{z}\right) \times T, \quad (3)$$

这里将 x, y, z 乘以 $\frac{k}{z}$ 是做投影变换,将对

象空间点坐标投影在 $z = k$ 平面上。对(3)式进行变换:

$$\left(\frac{kx}{z}, \frac{ky}{z}, k\right) \times T = \left(\frac{ku}{z}, \frac{kv}{z}, \frac{k}{z}\right) \times T \times T, \quad (4)$$

其中 T 为 T 的逆矩阵, T^* 为 T 的伴随矩阵, $|T|$ 为 T 的行列式。利用 $T = T^* / |T|$, 经过变换可得等式:

$$\left(kx/z, ky/z, k\right) \times T^* = (ku \times |T| / z, kv \times |T| / z, k \times |T| / z), \quad (5)$$

以上是纹理空间和对象空间的对应关系。从等式可知求透视纹理映射的关键是 T^* 矩阵。以下是矩阵 T 的伴随矩阵 T^* :

$$T^* = \begin{pmatrix} N_y M_z - M_y N_z & N_y L_z - L_y N_z & L_y M_z - M_y L_z \\ N_z M_x - M_z N_x & L_x N_z - N_x L_z & L_x M_z - M_x L_z \\ N_x M_y - M_x N_y & N_x L_y - L_x N_y & L_y M_x - M_y L_x \end{pmatrix}$$

简化以上伴随矩阵,形式为:

$$T^* = \begin{pmatrix} A_x & A_y & A_z \\ B_x & B_y & B_z \\ C_x & C_y & C_z \end{pmatrix}$$

因此等式(5)可变换成下列形式。

$$kx \cdot A_x / z + ky \cdot B_x / z + k \cdot C_x = ku \cdot |T| / z;$$

$$kx \cdot A_y / z + ky \cdot B_y / z + k \cdot C_y = kv \cdot |T| / z;$$

$$kx \cdot A_z / z + ky \cdot B_z / z + k \cdot C_z = k \cdot |T| / z;$$

为将多边形的屏幕坐标与纹理空间的对应映射关系表达清楚。将纹理空间的参数坐标用对象空间的坐标求解,以下是对应求解表达式:

$$\begin{cases} u = \frac{x \cdot A_x / z + y \cdot B_x / z + C_x}{x \cdot A_z / z + y \cdot B_z / z + C_z} \\ v = \frac{x \cdot A_y / z + y \cdot B_y / z + C_y}{x \cdot A_z / z + y \cdot B_z / z + C_z} \end{cases}, \quad (6)$$

通过(6)式便可得到纹理空间与屏幕空间的对应关系,完成了纹理映射的整个过程。这里有一种特殊情况需要处理,就是有时多边形上的一点对应的纹理空间坐标会大于 1,这时希望能将这一点对应到纹理空间内来。具体做法是去掉整数部分,保留小数部分,例如如果多边形上一点对应的纹理空间坐标是 $(1.3, 2.123)$ 则经过处理后其坐标为 $(0.3, 0.123)$ 。这种处理称为 u 或 v 约束。图 4 和图 5 分别是线性和透视纹理映射示例。

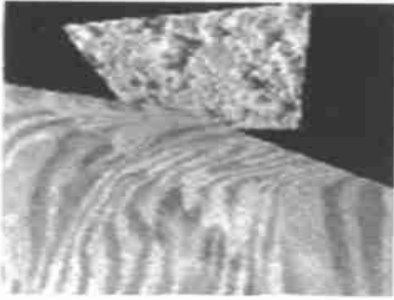


图 4 线性纹理映射

Fig. 4 Linear texture mapping.



图 5 透视纹理映射

Fig. 5 Perspective texture mapping.

5 失真处理

由于在计算机上显示的图形是由离散点组成的数字化图像,因而生成的图形必然与真实景物之间存在不同,称为失真现象。这种误差表现为图形上的直线或光滑的曲线呈现锯齿状、彩色花纹失去原有的形态和色彩、细小物体在画面上得不到反映等等。这些都是由于对真实纹理采样频率过低而导致的,一般来说,只要在生成图形时运用采样点技术,都会导致图形走样现象。为了解决这个问题,目前已有许多成熟技术,如:双线性插值法、Mip mapping、三线性插值法、面积采样代替点采样、增加采样点等。在自由曲面的透视纹

理映射中,采用增加采样点方法消除失真,利用参数曲面的双参数,经过不断参数细分,得到更多的自由曲面离散点,然后进行纹理映射,直到达到要求为止。

6 结论

以上介绍的曲面透视纹理映射方法,可以解决自由曲面纹理映射的难题,使纹理映射更加符

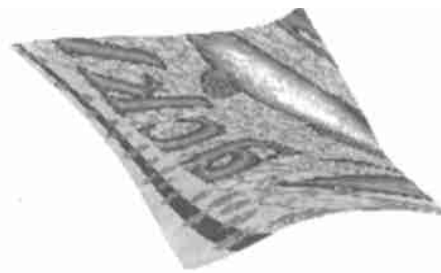
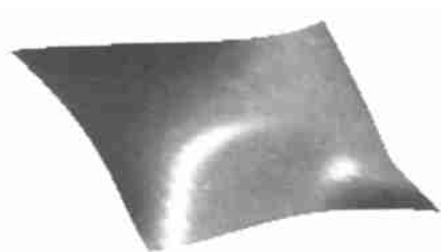


图 6 NURBS 曲面片的透视纹理映射

Fig. 6 Perspective texture mapping of NURBS.

合人的视觉效果。同时该方法还可以应用于多面体、自由形体等的表面纹理渲染,如 3D 仿真、游戏场景模拟、表面纹理细节模拟等。图 6 是自由曲面片透视纹理映射例子。

参考文献:

- [1] 范波. 一种基于透视变换的球面纹理映射[J]. 南京理工大学学报, 1999, (2): 34 - 38.
- [2] Phillip N, Nikos A A. On using planar developments to perform texture mapping on arbitrarily curved surfaces[J]. *Computers and Graphics*, 2000, 24(2): 539 - 554.
- [3] Yizhou Yu. Efficient visibility processing for projective texture mapping[J]. *Computers and Graphics*, 1999, 23(3): 245 - 253.

- [4]刘长松,程连冀. 3D 图形编程指南[M]. 兰州:西北工业大学出版社,2001.
- [5]Sun H Q, Bao H J. Interactive texture mapping for polygonal models[J]. *Computer Geometry*, 2000, 15(1 - 3): 41 - 49.
- [6]Duncan B, Olson A J. Texture mapping parametric molecular surfaces[J]. *Journal of Molecular Graphics*, 1995, 13(4): 258 - 264.
- [7]陈晓钟,孙华燕. 一种自然纹理背景下的图像目标检测方法 [J]. 光学 精密工程,2000,8(5):421 - 424.

Research on the perspective texture mapping of free - form surface

LIANG Wei-wen, BIN Hong-zan

(*School of Mechanical Science and Engineering,
Huazhang University of Science and Technology, Wuhan 430074, China*)

Abstract : Based on texture mapping, a perspective texture mapping is proposed in this paper. First, the principle of perspective transformation and its application in the texture mapping of free-form surface are presented. During the perspective texture mapping, the precision of approximation of bicubic patches is better than that of polyhedrons, so bicubic Bezier patches are used to approximate free - form surface. Then, the arithmetic of the perspective texture mapping is introduced: through getting the corresponding relationship between the texture space and the object space, and the corresponding relationship between the object space and the screen space, we get the corresponding relationship between the texture space and the screen space. Finally the aliasing is discussed. This method can also be effectively used for polyhedron and free - form body.

Key words : free-form surface; perspective transformation; texture mapping

作者简介:梁伟文(1974 -),男,湖北武汉市人,1997年毕业于华中科技大学机械制造及其自动化专业,同年进入华中科技大学CAM室攻读硕士研究生,1999年攻读博士学位,主要研究领域CAD/CAGD/CAM,数字化图形的处理与可视化。
e-mail:wwliang@Public.wh.hb.cn