

虚拟漫游系统中纹理的纠正与映射

Rectifying and Mapping the Texture of Virtual Tour System

邱建雄 (湖南大学软件学院 410082)
 (长沙大学计算机科学与技术系 410073)
 刘征海 (长沙 湖南大学软件学院 410082)
 肖基毅 (湖南衡阳 南华大学计算机学院 421003)

摘要:利用 3D 视觉分层重建技术,把模型的表面纹理图像从相片中提取出来,并加以纠正。然后根据 OpenGL 的要求把纠正的纹理变换到需要的尺寸,利用 OpenGL 提供的映射函数把它们映射到重建的三维模型的表面上去。使之具有相片般的真实感,并能够进行实时漫游。

关键词:虚拟现实 射影几何 灭线 纹理映射

1 引言

虚拟现实技术作为计算机应用研究的一项前沿技术,比如说在数字化城市规划、飞行仿真、VR 游戏、远程医疗、虚拟实景漫游等方面,有着非常广泛的现实应用。传统的几何造型技术虽然日益改进,但构建稍显复杂的三维模型仍然是一件耗时费力的工作,且模型的绘制结果与真实景物仍存在着较大的差异;目前广泛采用的三维扫描技术可以精确地获取大多数真实景物的几何模型,但它无法处理具有高光、透明以及绒毛表面的物体,一般也无法同时获取模型的颜色信息;近年来,基于图像建模技术受到了广泛的关注[1,2,3,4],其最大的优点就是可以直接从图像中为重建几何模型提取表面纹理和光照信息,从而简单、快速地构建出具有“照片级”真实感的三维模型。

在基于图像的建模方法中,首先要重建出场景的几何模型,然后用基于图像的方法,直接从图像中为重建的几何模型提取纹理并对其表面进行纹理映射,从而简单、快速地生成具有高度真实感的表面细节。这样就避免了传统的计算机图形学绘制中复杂的光照计算过程,解决了长期以来困扰人们的绘制速度与真实感之间的矛盾。其绘制流程可以用图 1 来描述。

2 相关工作

对于基于图像的建模与绘制技术,科学家们作了

大量的研究工作。E. Guillou 等人在给出一些假设的情况下,利用两个垂直方向的灭点和主点,对相机进行标定,和相机相对于世界坐标的旋转矩阵,然后求出相

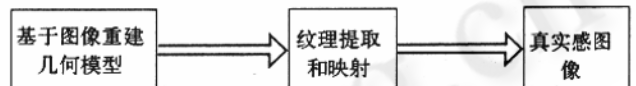


图 1 基于图像的绘制流程图

机的平移矩阵^[5]。Collins 和 Beveridge 提出一种纠正部分平面的技术^[6],如果平面的灭线可以确定,则从世界坐标到图像平面的变换可以简化为一个仿射变换。这一技术被用来对卫星图像进行对正。可使得 8 自由度的全参数射影单应性矩阵简化为一个 6 自由度的仿射变换。经过进一步的研究,D. Liebowitz 把这一过程简化为一个相似变换^[7],经过这种变换后可以对该平面进行度量测量。该平面的纠正单应性矩阵可以由场景的几何信息计算得到。特别是平行线,线段之间的角度以及不同方向线段的比例关系。以上这些方法都是针对平面的纠正方法。Wong 等通过对旋转体的研究,提出利用旋转体的结构和轮廓线对单幅图像进行标定的方法^[8]。Colombo 等从单幅未标定的图像中对旋转体表面进行度量重建和纹理提取^[9]。他们利用表面旋转体固有的三个独立的线性约束,对相机进行自标定,即估计出相机的焦距以及主点的坐标。利用表

面旋转体的比例函数把仿射变换扩展为透视射影。纹理的提取也只是利用以前步骤标定出来的相机的内部参数。

根据人造物场景中存在着大量的平面性质,本文在 D. Liebowitz 算法的基础上,利用 OpenGL 在纹理映射方面的特性,提出一种纠正并提取纹理的新方法,然后利用 OpenGL 实现纹理到场景模型的表面映射。达到对虚拟实景进行实时漫游的目的。通过我们的实验得到了比较满意的结果。

3 纹理的纠正与提取

为了把射影重建提升至度量重建或欧氏重建之一,可以把 3D 视觉分为四个几何层次,它们分别是射影几何、仿射几何、度量几何和欧氏几何。把 3D 视觉分层有利于执行重建。Faugeras 给出了如何利用把射影几何升级至仿射几何,以及把仿射几何升级至度量几何与欧氏几何来完成重建^[10]。

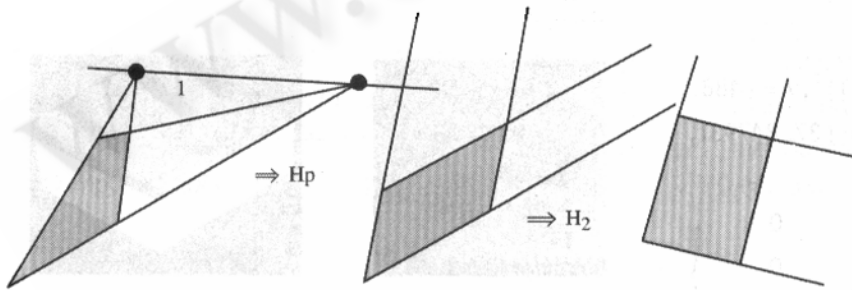


图 2 射影变换和仿射变换示意图

下面我们先介绍射影几何 (Projective Geometry) 方面的知识。 n 维实射影空间 P^n 中的任意一点可以用 $n+1$ 维向量 $[x_1, \dots, x_{n+1}]^T \in R^{n+1}$ 来表示,且满足:

- (1) x_1 到 x_{n+1} 中至少有一个不为零;
- (2) 对于所有非零实数 λ , $[x_1, \dots, x_{n+1}]^T$ 和 $\lambda[x_1, \dots, x_{n+1}]^T$ 表示射影空间 P^n 中的同一个点。此时, $[x_1, \dots, x_{n+1}]^T$ 被称为射影空间点的齐次坐标。

射影空间的变换被称为射影变换,具体地,从射影空间 P^m 到 P^n 的射影变换可以用一个 $(n+1) \times (m+1)$ 阶的矩阵 T^p 来表达: $x' = T^p x$ 。现在回到平面问题上来,我们可以把一个图像平面的点 x 映射到世界平面上的一点 x' 的纠正单应性矩阵用一个 3×3 阶矩阵 H

来表示:

$$x' = Hx \quad (3.1)$$

其中: H 是 8 自由度的。且可以按下式分解:

$$H = MN \quad (3.2)$$

其中: M 是单应性矩阵的度量部分,它是一个相似变换。可以写成以下形式:

$$M = \begin{Bmatrix} sR & t \\ 0^T & 1 \end{Bmatrix} \quad (3.3)$$

其中: R 是旋转矩阵, t 是平移向量, s 是各向同性的比例因子。 M 含有 4 个自由度。剩下的 4 个自由度包含在 H 矩阵的非度量部分 N 中。这里可以把 N 认为是把 l_∞, l 和 J (灭点和两个圆环点) 映射到规范值。即把 l_∞ 映射到 $(0, 0, 1)^T$, 把 l 和 J 映射到 $(1 \pm i, 0)^T$

而 N 又可以分为仿射和射影两部分 (如图 2):

$$N = H_0 H_p \quad (3.4)$$

H_p 把 l_∞ 映射到 $(0, 0, 01)^T$ 。由此,当我们把 H_p 写作 $(P_1, P_2, P_3)^T$, 把 l_∞ 写作 $(l_{\infty 1}, l_{\infty 2}, l_{\infty 3})^T$ 可得到:

$$l_\infty = H_p^T (0, 0, 1)^T = (P_1, P_2, P_3) \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} = P_3$$

$$H_p = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_{\infty 1} & l_{\infty 2} & l_{\infty 3} \end{Bmatrix} \quad (3.5)$$

因为 l_∞ 是一个齐次向量,所以 H_p 含有两个自由度。

$$H_0 = \begin{Bmatrix} \frac{1}{\beta} - \frac{\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{Bmatrix} \quad (3.6)$$

其中, α, β 是圆环点的实部与虚部。 H_0 中包含了 N 的最后两个自由度。我们也注意到,由于圆环点不确定,在经过这样的变换后,在度量平面上还保留有一个任意比例。而一般情况下,OpenGL 要求映射的纹理图像的长度和宽度是 2^n , 即 128, 256 等。我们利用这一特性可以从纠正好的图像中提取一个要映射的长方形图像,然后把它变换成为一个 $128 * 128$ 的纹理图像加以保存,以便在漫游的时候由 OpenGL 的映射函数使用。这样既可以避免计算出图像的实际长宽比例,又能达到实际应用的要求。

4 实验及结果

我们利用第三节介绍的方法,对包装盒的图像进行处理(如图 3 所示)。

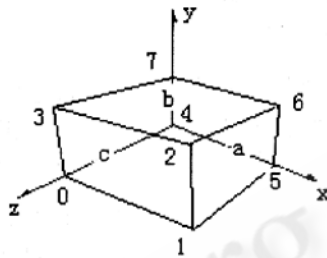
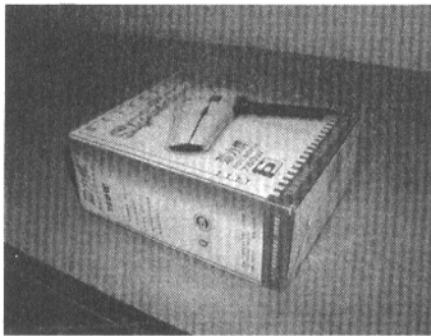


图 3 左图为包装盒的照片图像,右图是为它建立的坐标系示意图

现在我们想提取平面 3267 作为模型上表面的纹理图像。由此可以得到由 l_{32} 和 l_{76} 的交点 u (灭点) l_{73} 和 l_{62} 和的交点 v (灭点) 确定的直线 l_m 即平面 3267 的灭线 l_m , 即可得: $l_m = u \times v$ 。

由于 $u = (-554.11798, 373.52914, 1)^T, v = (488.621765, 411.377045, 1)^T$ 。所以求得 $l_m = (37.847931, -1042.739746, 410465.875)^T$ 。

$$H_p = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 37.847931 & -1042.739746 & 410465.875 \end{Bmatrix}$$

考虑到人造物场景中存在着大量的平行和垂直关系。我们可以利用线段相互垂直这个几何约束来计算 H_o 。 H_p 将两垂直方向的灭点 u 和 v 变换为 $u_A, v_A = (x, x, 0)^T$ 的形式, 它们定义了世界坐标系中的水平和垂直方向。可以把 u_A 写成 $(\cos\Phi, \sin\Phi, 0)^T$ 形式, 其中 Φ 是 u_A 与水平轴的夹角。于是可得到 $u_A = (1, 0, 0)^T$, 其中

$$R_A = \begin{Bmatrix} \cos\Phi & \sin\Phi & 0 \\ -\sin\Phi & \cos\Phi & 0 \\ 0 & 0 & 1 \end{Bmatrix} \quad (4.1)$$

如果 u_A 和 v_A 的夹角为 θ , 则可以通过以下变换使 v_A 与垂直轴重合。

$$A_1 = \begin{Bmatrix} 1 & -\cot\theta & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{Bmatrix} \quad (4.2)$$

$$A_2 = \begin{Bmatrix} u & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{Bmatrix} \quad (4.3)$$

其中, u 是水平和垂直方向的相关比例。因为我们可以不用完全恢复度量几何, 可以不计算矩阵 A_2 的值。

$$H_o = A_2 A_1 R_A \quad (4.4)$$

我们设 $H_A = A_1 R_A$

由上述公式和数据可以得到:

$$u_A = (-0.829196, 0.558958, 0)^T,$$

$$v_A = (0.764984, 0.64405, 0)^T$$

$$H_A = \begin{Bmatrix} 0.581257 & 0.862276 & 0 \\ -0.64405 & 0.764984 & 0 \\ 0 & 0 & 1 \end{Bmatrix}$$

我们把图 3 中的左图进行处理后得到纹理图像, 并用 OpenGL 映射函数把它映射到 3D 模型的表面, 得到具有相片般的真实感表面的模型。如下图所示:

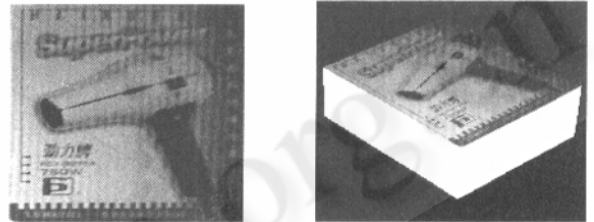


图 4 左图为纠正后的纹理图像, 右图具有一个纹理表面的 3D 模型

5 结论

本文是在 D. Liebowitz 算法的基础上, 提出了一种利用 OpenGL 纹理映射特性, 从场景图像中提取和纠正纹理图像的方法。从而避免了计算出纹理图像是实际尺寸, 由 OpenGL 的映射函数自动完成映射过程, 也能够满足实时漫游的需要。但由于纹理图像是从单幅相片中提取的, 当视点移动比较大时, 纹理的光照等特性不能正确的反映当前视点的情况, 从而造成一定的失真。下一步的工作是尽可能的多利用几幅图像来提取和合成与视点有关的纹理用于映射, 这样就能达到更好的效果。

参考文献

- 1 P. Debevec et al. Modeling and Rendering Architecture from Photographs: A Hybrid Geometry – and Image – based Approach [C]. Proc. SIGGRAPH 96, Aug. 1996, p,11 – 20.
- 2 P. Debevec, YiZhou Yu, et al. Efficient View – dependent Image – based Rendering with Perspective Texture – Mapping [C]. 9th Eurographics Rendering Workshop, 1998, p,105 – 116.
- 3 G. Slabaugh, W. B. Culbertson, T. Malzbender, and R. Schafer, A Survey of Volumetric Scene Reconstruction Methods from Photographs [C], In Proc. Of Joint IEEE TCVG and Eurographics Workshop, 2001, p,81 – 100.
- 4 吴福朝,王光辉,胡占义,由矩形确定摄像机内参数与位置的线性方法[J],软件学报,2003,14(3):703 – 712.
- 5 E. Guillou, D. Meneveaux, E. Maisel, et al, Using vanishing point for camera calibration and coarse 3D reconstruction from a single image [J], The Visual Computer, 2000, 16:396 – 410.
- 6 R. T. Collins and J. R. Beveridge, Matching Perspective Views of Coplanar Structures Using Projective Unwarping and Similarity Matching [C], In Proc. IEEE Conference on Computer Vision and Pattern Recognition, June 1993, p. 240 – 245.
- 7 D. Liebowitz, Camera Calibration and Reconstruction of Geometry from Images [M], Ph. D. thesis, University of Oxford, UK, 2001.
- 8 K. – Y. K. Wong, P. R. S. Mendonca, and R. Cipolla, Camera Calibration from Surfaces of Revolution, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(2): 147 – 161.
- 9 C. Colombo, A. DelBimbo, and F. Pernici, Uncalibrated 3D Metric Reconstruction and Flattened Texture Acquisition from a Single View of a Surface of Revolution [C], 3DPVT02, 2002, p,277 – 284.
- 10 O. Faugeras, Stratification of 3D Vision: Projective, Affine, and Metric Representations [J], Journal of the Optical Society of America, March 1995, 12 (3):465 – 484.