

基于灰度图像的三维曲面重建系统设计

胡志勇¹, 张秀芬¹, 蔚刚², 梁发周¹

(1. 内蒙古工业大学机械学院, 内蒙古呼和浩特 010051;

2. 内蒙古机电职业技术学院, 内蒙古呼和浩特 010051)

摘要:由二维图像重建三维模型是计算机科学中的一个研究热点。对该问题进行了深入研究, 提出了一个基于 Shape From Shading (SFS, 基于阴影恢复形状) 方法进行曲面重建的方法。该方法从图像入手, 采用图像处理技术对图像进行必要的处理, 由图像的亮度进行曲面三维形状的重建, 并利用 OpenGL 和 Visual C++ 开发了一个三维曲面重建系统, 可以由曲面的二维灰度图像方便地重建出其三维形状。该系统的设计对实现产品的快速开发具有重要的意义。

关键词:阴影恢复形状; 曲面重建; OpenGL

中图分类号: TP391.41

文献标识码: A

文章编号: 1673-629X(2007)10-0061-04

Design of 3D Reconstruction Based on Gray Image

HU Zhi-yong¹, ZHANG Xiu-fen¹, YU Gang², LIANG Fa-zhou¹

(1. Mechanical Engineering College, Inner Mongolia University of Technology, Huhehaote 010051, China;

2. Inner Mongolia Machine Electricity Occupation Technical College, Huhehaote 010051, China)

Abstract: Recovery of a three-dimensional model from a two-dimensional image is a well-known problem of computer science. After studying this problem deeply, presents a method of 3D reconstruction based on shape from shading (SFS). This method from image obtaining, uses the imagery processing technology to carry on essential processing to the image, carries on the curved surface three dimensional shape by image brightness reconstruction. In addition, a system is developed with VC++ 6.0 and OpenGL. This system may reconstruct its three dimensional shape conveniently by the curved surface two-dimensional gradation image. This system design to realize the product fast development to have the vital significance.

Key words: shape from shading; recovery of surface; OpenGL

0 引言

由图像重建物体表面是计算机视觉领域中的一个重要任务。在各种重建技术中, SFS (Shape from shading) 算法以其算法简单、使用方便等卓越性能而引起了许多学者的重视^[1,2]。图像上各点灰度值反映了三维物体表面上相应点的反射光强度, 由物理学和几何光学的分析表明, 反射光的强度和物体表面性质与物体表面的几何形状有关系。SFS 算法正是基于以上原理提出的一种三维重建方法, 其实质是利用二维灰度图像中各点的明暗变化来恢复其表面三维形状。自 Horn^[1]于 1970 年提出该算法以后, 一直是计算机视觉领域里的一个研究热点, 其基本思路是首先构造一个能量函数, 然后加入合适的约束条件, 寻找能量函数的

最小化 Euler 方程, 再将该方程离散化处理, 最后通过设计计算离散 Euler 方程的迭代过程来获得使能量函数最小化的解。

1 迭代算法

在理想的成像条件下, 图像的灰度满足反射图函数:

$$E(x, y) = I(x, y) \frac{n_0 \cdot n}{||n_0 \cdot n||} = I(x, y) \frac{n_{01}n_1 + n_{02}n_2 + n_{03}n_3}{\sqrt{n_{01}^2 + n_{02}^2 + n_{03}^2} \sqrt{n_1^2 + n_2^2 + n_3^2}} = R(p(x, y), q(x, y)) \quad (1)$$

其中物体表面法方向 $n = (n_1, n_2, n_3)$, 光源方向 $n_0 = (n_{01}, n_{02}, n_{03})$ 或 $(-p_0, -q_0, 1)$, $E(x, y)$ 为图像在像素点 (x, y) 处的亮度, $p = \frac{\partial Z}{\partial x}$, $q = \frac{\partial Z}{\partial y}$ 或 $p = -\frac{n_1}{n_3}$, $q = -\frac{n_2}{n_3}$ 为表面梯度。要想求出物体表面高度 Z 只需

收稿日期: 2006-12-08

基金项目: 内蒙古自然科学基金资助项目 (200408020814)

作者简介: 胡志勇 (1961-), 男, 教授, 研究方向为计算机辅助设计、计算机图形学。

求解式(1)得出 (p, q) , 然后积分。但该模型所确定的 SFS 问题是病态的, 因此为了求解, 需要另加约束条件。

Horn^[3] 在连续可微域上讨论 SFS 问题, 认为已知图像数据与由反射模型所确定的物体表面图像亮度之间存在误差, 将亮度方程(1)转换为如下误差函数形式:

$$e = \iint_{\Omega} (E(x, y) - R(p(x, y), q(x, y)))^2 dx dy \quad (2)$$

为了弥补该方法不稳定的缺点, 又加入了光滑性约束和可积性约束, 得到如下函数:

$$\iint_{\Omega} [(E(x, y) - R(p(x, y), q(x, y)))^2 + \lambda(x, y)(p_x^2 + p_y^2 + q_x^2 + q_y^2) + \mu((z_x - p)^2 + (z_y - q)^2)] dx dy$$

其中, λ 和 μ 是拉格朗日常数。

SFS 模型是个非线性反射函数, 考虑到非线性项对重建结果的影响不大, 所以可以将反射函数进行泰勒展开。Pentland^[4] 以表面梯度 (p, q) 为变量对反射函数进行线性化, 然后对亮度方程的两边都进行傅里叶变换, 整理之后, 进行逆傅里叶变换, 得到物体表面的高度值。而 Tsai 和 Shah 首次利用后向有限差分方法将反射函数离散化:

$$p = \partial Z / \partial x = z_{ij} - z_{i,j-1}, q = \partial Z / \partial y = z_{ij} - z_{i-1,j}$$

图像辐射方程变成:

$$f(z_{i,j}) = E_{i,j} - R(z_{ij} - z_{i,j-1}, z_{ij} - z_{i-1,j}) = 0$$

然后对上式进行泰勒展开:

$$0 = f(z_{ij}) \approx f(z_{i,j}^{n-1}) + (z_{i,j} - z_{i,j}^{n-1}) \frac{\partial f}{\partial z_{i,j}}(z_{i,j}^{n-1})$$

令第 n 次迭代结果 $z_{i,j}^n = z_{i,j}$, 则得到

$$z_{i,j}^n = z_{i,j}^{n-1} - \frac{f(z_{i,j}^{n-1})}{\frac{\partial f}{\partial z_{i,j}}(z_{i,j}^{n-1})}$$

$$\text{其中 } \frac{\partial f}{\partial z_{i,j}}(z_{i,j}^{n-1}) = \frac{(p+q)(pp_0+qq_0+1)}{\sqrt{1+p^2+q^2}^3 \sqrt{1+p_0^2+q_0^2}} - \frac{p_0+q_0}{\sqrt{1+p^2+q^2} \sqrt{1+p_0^2+q_0^2}}$$

2 曲面重建系统的编程实现

2.1 系统物理模型

根据系统的功能需求, 建立系统的总体模型如图 1 所示。实际应用时, 所获得的图像都是彩色图像, 不能直接进行使用, 需要经过图像处理, 然后将图像导入到系统中, 用合适的算法进行数据的三维重建。但一般重建出的模型比较粗糙, 系统通过数据接口将其输出到其它三维造型软件中(如 3DMax)进行进一步修

改, 以备使用。

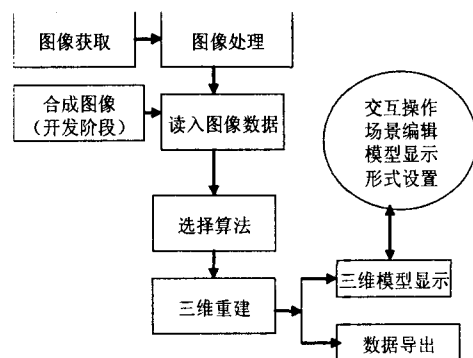


图 1 系统物理模型

2.2 图像的预处理

利用数码相机获取的物体的图像难以满足上述假设, 为了使重建结果尽量接近真实形状, 必须对真实图像进行处理。主要是对图像进行平滑处理, 去除噪声, 然后进行图像分割, 采用基于边缘的图像分割技术将图像与背景分离。处理后的图片如图 2 所示。

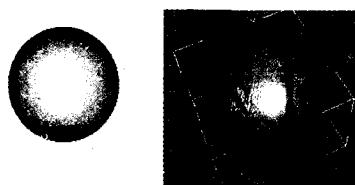


图 2 合成图像球面的原始图像

2.3 离散点云的三角网格化

由物体的二维图像利用三维重构算法计算出来曲面各点的高度是离散点云数据。而运用 OpenGL 进行表面的三维重建需要将这些点进行有序的三角网格化。文中用一个二维数组 $Z[]$ 来存储表面数据点云数据。二维数组 $Z[i, j]$ 中的 (i, j) 和二维图像的像素 (x, y) 的相对应位置是一一对应的, 所以可以根据右手定则使用图 3 循环构筑三角网格。

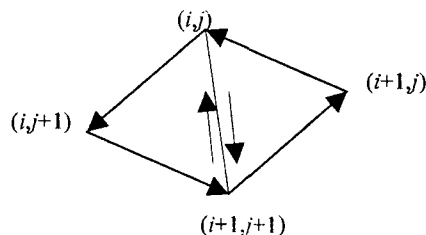


图 3 构筑三角网格

如图 3 所示首先利用右手定则用 (i, j) , $(i+1, j+1)$, $(i+1, j)$ 三个点构筑一个三角形, 然后再根据右手定则用 (i, j) , $(i, j+1)$, $(i+1, j+1)$ 构筑另一个三角形, 这样每一个离散点可以构造两个三角形。利用如上规则, 可以方便地完成三角网格化。

2.4 法向量计算

法向量是面上某一点处垂直于该面的向量。对于

一个平面,其上各点的法向量都一样,所以称为平面法向。而对于一个曲面,虽然在计算机图形学中常用许多小曲面片去逼近,但事实上每个顶点的法向都不一样。在三维物体显示技术中,法向量不但是得到理想的光照效果重要条件,也是产生逼真图形的关键因素。它通过确定物体相对于光源的方位,可以计算出此点的受光情况。

法向量的计算有解析曲面和一般多边形两种方法^[5]。解析曲面是数学方程或方程组描述的平滑的可导曲面,通过求导得出某点处的切向量,然后进行叉积即可求出此点处的法向量。对于一般的多边形(如三角形),其内部点的受光情况由三个顶点按一定的算法计算得到。OpenGL 本身并未提供计算法向量的函数,但它提供了指定法向量的函数。计算法向量的工作由程序员自己去完成。

在系统中,通过定义 $X_1, X_2, X_3, Y_1, Y_2, Y_3$ 六个变量作为一个三角网格三点 $A(x_1, y_1, z_1), B(x_2, y_2, z_2), C(x_3, y_3, z_3)$ 之中的任两个点间的向量(X_1, X_2, X_3), (Y_1, Y_2, Y_3)。从而计算出三角平面的法向量(n_1, n_2, n_3)。 $X_1 = x_1 - x_2; X_2 = y_1 - y_2; X_3 = z_1 - z_2; Y_1 = x_1 - x_3; Y_2 = y_1 - y_3; Y_3 = z_1 - z_3; n_1 = X_2 * Y_3 - X_3 * Y_2; n_2 = X_3 * Y_1 - X_1 * Y_3; n_3 = X_1 * Y_2 - X_2 * Y_1$ 。在编程中,在每次绘制三角形的前面对要绘制三角形的三个点求法向量,这样可以在绘制三角形表面的同时也可以将法向量信息计算完毕,再通过 `glNormal3f(n_1, n_2, n_3)` 指令完成法向量的显示。

2.5 系统的数据接口

由灰度图像重建出的三维模型精度一般达不到应用要求,需要进一步编辑修改,因此必须建立本系统与其他造型软件的数据交换接口。由于 3ds 文件应用比较广泛,而且大部分软件都提供了 3ds 文件的导入导出,所以,该系统采用 3ds 文件格式为数据交换的主要格式。通过编程实现了由自定义的三维模型数据到 3ds 数据文件的转换功能。

3ds 文件由许多块组成,每个块的信息类别用 ID 标识,并包括了下一个块的位置信息。3ds 文件中有一个基本块,其 ID 是 4D 4D,该块包含在每个 3ds 文件的开头。该基本块内包含两个主块:3D 编辑程序块 3D3D 和关键帧块 B000。系统主要实现模型的几何拓扑信息的导出,具体编程时,首先要对 3D 编辑程序块进行宏定义,然后定义 `export3ds` 类实现数据的导出。当用户触发导出命令时,通过视图类 `CMyView` 调用相应函数实现操作,关键代码如下:

```
int CMyView::export(char * file)
{ ..... 参数初始化
  nv = msh -> get_nv(); // 顶点
  if (! strcmp(c + 1, "3ds")) { export3ds exp3ds(file, 1.0f, matlib, nmatlib);

  .....

  for( int i=0; i<nv; i++ ) if (vertices[i])
  { vert[3 * nvert] = (float)vertices[i][0]; vert[3 * nvert + 1] = (float)vertices[i][1];
    vert[3 * nvert + 2] = (float)vertices[i][2]; flag[i] = nvert + +; }

  for( i=0; i<nf; i++ ) if (tr[i]) {
    faces[nfaces * 3] = tr[i][0]; faces[nfaces * 3 + 1] = tr[i][1];
    faces[nfaces * 3 + 2] = tr[i][2]; nfaces + +; }

  exp3ds.WriteNamedObject(msh -> filename, vert, textcoord, nvert, faces, nfaces, mat, lmat, matlib, 0);

  ..... return 1; } return 0; }
```

2.6 程序实现

利用面向对象的分析方法,将图像、三维模型、重建计算、模型的显示和输出等分别进行集成,设计了 `CShpae` 类、`CMesh` 类、`CImage` 类和 `export3ds` 类。其中 `CShape` 类包括各种重建函数和所需参数如入射光源的方向和迭代次数等, `CMesh` 用于处理重建后的三维模型,记录其几何和拓扑信息, `CImage` 集成了图像读写及图像的相关参数,如图像大小等。另外,利用 `CMyView::drawScene(CDD * cDC)` 成员函数在主窗口中绘制重建后的三维模型。右上角的 `CImageView` 窗口显示原始图像,并进行相应的图像处理操作, `CForm` 区域是一个命令面板,一些常用的命令放在该面板,其它功能采用菜单命令。系统的外部数据接口是通过 `CMyView::OnExport()` 调用 `export3ds` 类成员函数来将模型转换为 *.3ds 格式文件。算法步骤如下:

- Step1: 初始化;
- Step2: 获取光源方向和迭代次数;
- Step3: 计算光源法向量,并开始迭代;
- Step4: 判断误差是否满足收敛条件,如果满足转 Step5, 否则返回 Step3;
- Step5: 结束。

3 实 例

图 4 是该系统的用户界面以及对一曲面进行重建的效果图。使用的重建算法是线性化方法,光源方向为 (0, 0, 1), 迭代次数为 3。该算法简便快捷,但实验证明迭代次数在 2 到 10 次以内效果比较好,当迭代次数增多时,算法变得不稳定。图 5 是利用摄像头获取的狗模型图像及其重建效果。通过比较可以看到线性

化算法对于光滑的物体表面重建效果较理想。

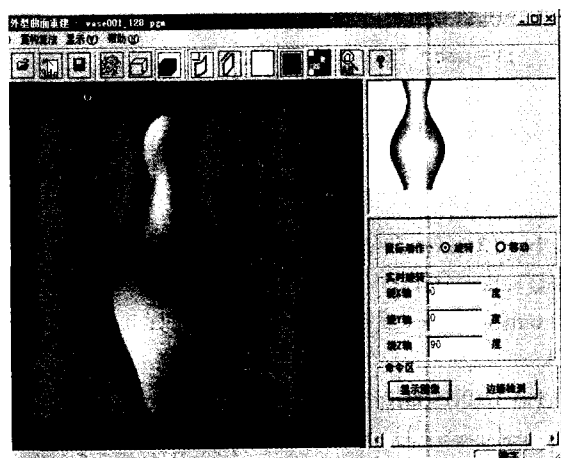


图 4 系统界面及重建效果



图 5 真实图像的重建效果
(左为重建后的模型, 右为图像)

4 结论与展望

对 SFS 算法进行了研究分析, 并提出一种利用 SFS 进行曲面外形曲面的重建的方法, 在此基础上, 利用 VC 和 OpenGL 开发了一个重建平台, 可以实现一般曲面的重建。该系统基本可以快速地重建出物体的外形, 或者先重建出各个部件, 然后导入 3D Max 等其它成熟的三维造型软件中进行编辑修改, 最后组装成整机。但该系统也存在一些不足, 所以提出以下几点展望: 1) 系统中采用的重建算法有待进一步改进, 以实现任意光照下的灰度图像的三维重建。2) 提高重建精度, 特别是重建误差太大, 需要进一步研究。

参考文献:

- [1] Zhang R, Tsai P S, Cryer J, et al. Shape from Shading: A Survey[J]. IEEE Transactions on PAMI, 1999, 21(8): 690 - 706.
- [2] Ragheb H, Hancock E R. Surface radiance correction for shape from shading[J]. Pattern Recognition, 2005, 38: 1574 - 1595.
- [3] Horn B, Brooks M. The Variational Approach to Shape from Shading[J]. Computer Vision, Graphics and Image Processing, 1986, 33(2): 174 - 208.
- [4] 廖 熠, 赵荣椿. 从明暗恢复形状(SFS)的几类典型算法分析与评价[J]. 中国图象图形学报, 2001, 6A(10): 953 - 960.
- [5] 费广正, 卢丽丹, 陈立新. 可视化 OpenGL 程序设计[M]. 北京: 清华大学出版社, 2001: 36 - 37.

(上接第 60 页)

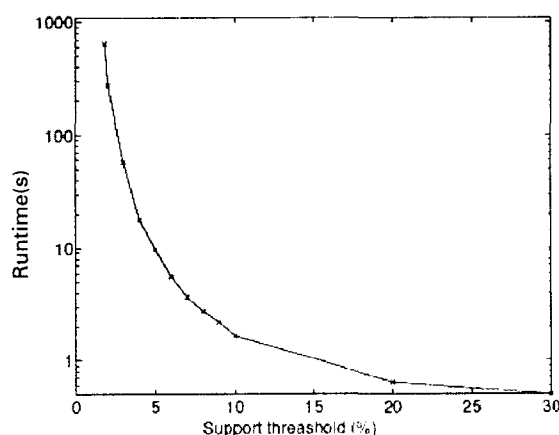


图 4 gSpan 算法性能曲线图

参考文献:

- [1] Yan Xifeng, Han Jiawei. gSpan: Graph - based substructure pattern mining[R]. Madrid: Department of Computer Science, URJC, 2002.

- [2] Agrawal R, Srikant R. Fast algorithms for mining association rules[M]. New York: IBM Almaden Research Center, 1994: 487 - 499.
- [3] Kuramochi M, Karypis G. Frequent subgraph discovery[C]//ICDM2001. California: [s. n.], 2001: 313 - 320.
- [4] Asai T, Kawasoe S, Kawasoe S. Efficient substructure discovery from large semistructured data[C]//SDM2002. Arlington, VA, USA: SLAM, 2002: 225 - 227.
- [5] Pei J, Han J. PrefixSpan: Mining sequential patterns efficiently by prefix - projected pattern growth[C]//ICDE2001. Neuss: [s. n.], 2001: 215 - 224.
- [6] SUN L, ZHANG X. Efficient Frequent Pattern Mining on Web Logs[C]//APWeb2004. Hangzhou: [s. n.], 2004: 533 - 542.
- [7] Zaki M J. Efficiently mining frequent trees in a forest[C]//KDD2002. Edmonton: [s. n.], 2002: 135 - 139.
- [8] Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to Algorithms[M]. 2nd Edition. Massachusetts: MIT Press, 2001: 12 - 14.