
北京师范大学学位论文原创性声明

本人郑重声明： 所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

学位论文使用授权书

学位论文作者完全了解北京师范大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京师范大学。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。保密的学位论文在解密后适用于本授权书。

本人签名： _____

日期： _____

导师签名： _____

日期： _____

基于投影矩阵估计的真实感纹理贴图

摘 要

如何更好地在电脑上展示一个物体的细节是计算机诞生以来一直存在的问题。计算机图形学给出的答案是给物体建立数字化三维模型，并通过三维模型展示物体各角度的细节。在对三维模型重建过程中，一个重点环节是构建物体的三维几何表示，另一个重点环节是对三维模型进行纹理贴图，使其更具真实感。本文针对根据物体实际图像生成三维模型纹理贴图的流程做出优化，并提出了一种自动化生成纹理贴图的思路。主要工作包括：

提出一种利用凸包最近匹配点对投影矩阵进行优化的算法。用最小二乘法对初始对应点估计出初始投影矩阵，并对投影点与图像点都求出凸包。将投影点的凸包和图像点的凸包间的最近点作为补充的对应点，运用到下一次的最小二乘法估计投影矩阵并再次优化。如此迭代多次后，可以求得一个适当的投影矩阵。

提出一种多幅图像自动化生成纹理贴图的流程。自动化处理多幅图像贴图的核心在于自动算出每个图像的投影矩阵，要算出一幅图的投影矩阵需要知晓至少六个初始模型-图像对应点。对于未知初始对应点的图像，采用图像特征检测、匹配算法算出它与已知投影矩阵的图像的对应点，加上利用 z-buffer 算法求得的已知投影矩阵图像模型-图像点对应关系，二者结合可在线求出新图像的初始对应点。最后在生成纹理贴图阶段，以三维模型二维展开图为纹理基准图，生成具有真实感的纹理贴图。

本文的研究给自动化生成纹理贴图的研究者提供了新的思路。可以大幅减少模型平均人为标定初始模型-图像对应点的个数：从以往的每幅图像至少六个初始对应点减少为每个模型至少六个初始对应点。

关键词：三维模型，纹理贴图，投影矩阵

Realistic Texture Mapping Based on Projection Matrix Estimation

Abstract

How to display the details of an object better on a computer is a problem that has been existed since the birth of the computer. The answer given by computer graphics is to establish a digital 3D model for an object and to show the details of the object angles through the 3D model. In the process of 3D model reconstruction, one key link is to construct the 3D geometric representation of objects, and the other is to map the texture of 3D models to make them more realistic. This paper optimizes the process of generating texture map of 3D model according to the actual image of objects, and proposes a way of automatically generating texture map. The main work includes:

An algorithm for optimizing projection matrix by using the nearest matching point of convex hull is proposed. The initial projection matrix is estimated by the least square method, and the convex hull is obtained for both the projection point and the image point. The nearest point between convex hull of projection point and convex envelope of image point is used as supplementary corresponding point, and then the projection matrix is estimated and optimized again by the next least square method. After such iteration, an appropriate projection matrix can be obtained.

A new process of automatic texture mapping is proposed. The core of automatic processing of multiple image maps is to calculate the projection matrix of each image automatically. To calculate the projection matrix of a picture, at least six initial models image corresponding points should be known. For the image with unknown initial correspondence point, the corresponding points of the image with known projection matrix are calculated by using image feature detection and matching algorithm. In addition, the model image point correspondence of known projection matrix image obtained by z-Buffer algorithm can be calculated online. Finally, in the phase of texture mapping, the texture map with realistic sense is generated by taking the 3D model 2D expansion map as the texture reference map.

The research of this paper provides a new idea for the researchers who automatically generate texture map. The number of corresponding points of the initial model image can be reduced significantly: from at least six initial corresponding points of each image in the past to at least six initial corresponding points of each model.

KEY WORDS: three-dimensional model, texture mapping, projection matrix

目 录

1. 引言	1
1.1. 三维模型重建的背景与意义	1
1.2. 三维模型纹理贴图的研究现状	2
1.3. 本文工作介绍	3
2. 投影矩阵的估计与优化	4
2.1. 相机模型与投影矩阵	4
2.2. 初始投影矩阵估计	6
2.2.1. 得到模型-图像的初始对应点	7
2.2.2. 使用最小二乘法得到初始投影矩阵	10
2.3. 基于凸包的 ICP 算法优化投影矩阵	12
3. 基于投影矩阵的真实感纹理贴图	15
3.1. 纹理贴图简介	15
3.2. 可见模型点对应的颜色	16
3.2.1. 使用单幅图像确定可见模型点颜色	16
3.2.2. 多幅图像确定可见模型点颜色	17
3.3. 使用可见点颜色生成纹理贴图	17
3.4. 拟合得到最终的纹理贴图	18
4. 总结与展望	19
4.1. 本文总结	19
4.2. 未来展望	19
参考文献	21
附 录	23
致 谢	24

1 引言

1.1 三维模型重建的背景与意义

如何更好地在电脑上展示一个物体的细节是计算机诞生以来一直存在的问题。计算机图形学给出的答案是给物体建立数字化三维模型，并通过三维模型展示物体各角度的细节。并且数字化三维模型可以方便地通过互联网进行传播，使得大众能足不出户观察到物体的全貌。

在教育领域中，借用实体三维模型作为教学用具已经非常普遍。但实体三维模型遇到的问题有老化、磨损严重、并且在完成当前教学任务后就不再使用，造成浪费，并且保管起来也很麻烦。借用三维模型就不会遇到这些问题，一台电脑、一张普通的独立显卡就可以借由三维数字模型满足大多数教学的需要了。

在游戏、影视等领域中，常常需要描述一些现实世界不存在或无法观测全貌的物体。三维模型重建可以给虚拟物体建立数字化三维模型，并用该模型对虚拟物体进行展示。

文物展览过程中，由于一般的实体展览容易遇上文物磕碰、氧化等问题，可以考虑使用文物的数字化三维模型代替文物实体代为展览。一些珍贵的文物还不便于实体借用供其他学者研究，但如果对其进行三维数字化建模后，便可以无损分享了。在我校的文物修复科研项目中，同样遇到了对文物进行三维建模的问题。

在对三维模型重建过程中，一个重点环节是构建物体的三维几何表示，另一个环节是对三维模型进行纹理贴图，使其更具真实感。

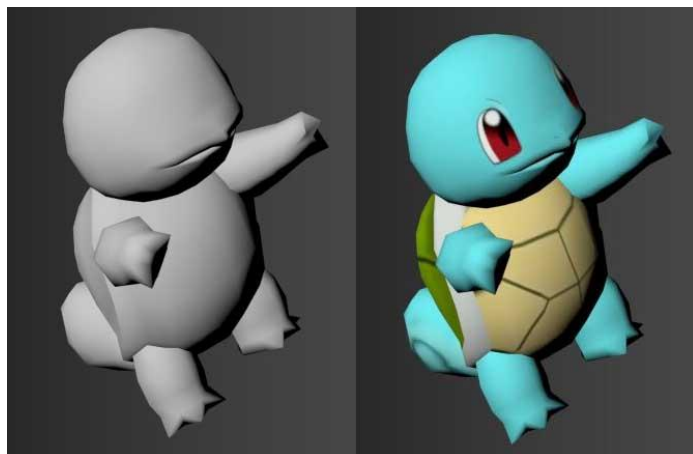


图 1 左：龟的三维模型，右：龟的带纹理三维模型

图 1 是三维模型的一个具体示例，左图为龟的未贴纹理的三维模型，右图为贴上了卡通纹理的龟的三维模型。

1.2 三维模型纹理贴图的研究现状

物体数字化的一个主要方法是，首先使用三维扫描技术，直接获取三维的形状信息，并进行三维模型的重建。获得三维模型之后，进行纹理贴图，以确定模型各点的颜色信息^[1]。纹理贴图中，可以使用专用的图形处理软件进行，3Dmax 就可以完成该步骤^[2]。软件的使用过程中，需要手工将不同角度的图片与模型进行结合，费时费力。硬件也可以实现在采集物体形状信息的同时，采集到物体的颜色信息^[3]。但是，采集到的信息普遍较为粗糙，因为硬件的使用需要严格的环境和光照亮度条件，现实中难以操作^[4]。三维数字影像测量可以解决硬件直接建立 3D 模型不准确等问题，但是其造价昂贵，不适合大量普及使用^[5]。由于以上方法都具有其局限性，后续的研究中，自动化纹理贴图就成为 3D 建模的热点问题。

纹理贴图已成为计算机图形学界一种流行的工具，因为它是一种无需花费太多精力就可以在计算机生成的图像中实现高度真实感的简便方法^[6]。纹理贴图的技术在各领域都有应用。物体三维重建中，在 3D 建模后，需要进行纹理映射，以将图片与模型点相对应^[7]。早期的纹理贴图针对于简单图形的算法较多，后来随着大规模集成电路的开发，贴图的真实感成为计算机图形学领域研究的热点^[8]。真实感纹理贴图中，主要涉及到的问题有画面绘制、明暗处理、光线跟踪等技术。

在曲面的 3D 显示中，通常是使用多边形或三维曲面^[9]。为计算三维曲面上每一点的颜色值，需要将多边形或三维曲面上的每一点进行着色。Catmull 在 1974 年提出传统算法，应用映射函数将图片上点的颜色值映射到相应的三维曲面上^[10]。基于视图的纹理映射（view dependent texture-mapping, VDTM）可以将形状和场景结合起来，最大程度地利用已有的图片信息^[11]。为使物体的表面更加真实，可以利用纹理函数，改变物体的透明度^[12]，也可以使用扰动函数，改变物体表面的其他参数，可以真实地描绘人体等实物^[13]。在以上两个方法的基础上，对于光线信息，可以使用数字信号理论对光线进行建模，添加到纹理贴图中^[14]。对于图片中的阴影信息，目前的算法已经可以成功地识别光线方向，并且在贴图上去除图片中的这些信息，随着观察者的角度而改变光照，做到真实的呈现模型各角度^[15]。其主要方法是把真实世界中照片的点进行对应以相互覆盖，并计算出物体原本的颜色。之后再根据模型呈现的角度，进行光线和阴影的添加^[16]。

为使物体的外观更加真实，纹理贴图有很多优化的方法。可以提高采样率，即提高图片的分辨率^[17]。也可以采用滤波器，将像素作为区域进行处理^[18]。这个过程中，进行表面参数化可以将多边形组成的曲面更好地平滑，以方便展示^[19]。在纹理贴图的过程中，图像

可能会失真。这时，改变投影矩阵对应的映射函数可以更准确地将图片上的点对应到模型上^[20]，还可以采用插值法描绘更加细节的纹理图像。浮雕纹理贴图可以更好地将算法生成的纹理映射到模型的多边形上，可以更清晰的呈现视差，加强视觉效果^[21]。

由于目前对自动化贴图的需求日益增加，图片的分辨率不断提高，在对贴图算法的优化中，如何更好地利用计算机的算力也是一个重要问题。可以采用优化缓存的方法，更好地利用计算机的系统资源，加快计算速度^[22]。从数学算法上改善算法的速度也是近年来研究的热点。体绘制积分（广义 Radon 变换）和滤波后投影积分（逆 Radon 变换）将复杂的线性变换转化为积分的求解，都可以加速纹理贴图的过程^[23]。对于不同的场景，可使用的最佳渲染算法也不同。点采样和 Mipmap 结合线性滤波算法重构可减少数据移动量，点采样和双线性滤波算法重构可减少计算量，点采样算法可降低计算机的功耗，而双线性滤波算法可提高缓存命中率^[24]。

对于纹理贴图，目前的技术已非常成熟。阿里巴巴公司开发了一个可以方便地生成纹理贴图的程序，仅需要将物体放入，就可以直接生成彩色的模型，减少人工建模和贴图所需的工作量^[25]。该专利使用昂贵，必须放入实际物体到容器中，不便于大批量进行三维模型制作。目前，自动化贴图技术仍然需要人工在后期进行一些处理，对于大批量进行贴图值得后续研究。

1.3 本文工作介绍

对单幅图像的投影矩阵进行估计，提出一种基于最小二乘法、ICP（最近对应点迭代）算法的投影矩阵估计算法。用已知的三维模型与图像的对应点对进行投影矩阵的初始化估计。并基于 ICP 算法，利用凸包匹配点对投影矩阵进行优化。

利用得到的投影矩阵，用 z-Buffer(最近缓冲区)算法得到部分可观察三维点的对应图像点。

用 KAZE（风）特征提取算法求得已知投影矩阵图像与未知投影矩阵图像的图像公共对应点，并利用已知投影矩阵图像可见点与模型点的对应关系，求得未知投影矩阵的图像的部分点与模型点的对应关系。

重复以上过程，直至所有图像的投影矩阵都被估计出来。

根据各个图像点与模型点的对应关系并用 CGAL（计算几何算法库）中的函数向量化三维模型，生成模型对应的纹理贴图。

2 投影矩阵的估计与优化

2.1 相机模型与投影矩阵

相机模型与投影矩阵的具体推导过程可以参考 Moons T U 的研究^[26]，这里不再详述。仅把本文实际用到的部分再介绍一次：

设三维空间坐标中某一模型点的坐标为 $(x_i, y_i, z_i)^T$ ， $x_i, y_i, z_i \in \mathbb{R}$ ，那么对于该幅图像存在一个投影矩阵 P ：

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}, p_{ij} \in \mathbb{R} \quad (2.1)$$

满足：

$$\lambda_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = P \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (2.2)$$

其中， P 是一个 3×4 的矩阵，是投影矩阵，该空间点的齐次坐标 $(x_i, y_i, z_i, 1)^T$ 经过该投影矩阵投影后（矩阵相乘）得到一个未归一化的投影点坐标 $(\lambda_i u_i, \lambda_i v_i, \lambda_i)^T$ 。在该未归一化坐标中， λ_i 表示该三维模型点距离相机的远近情况，数字越大那么距离相机越远：

$$\lambda_i = p_{31} * x_i + p_{32} * y_i + p_{33} * z_i + p_{34} \quad (2.3)$$

将未归一化的投影坐标归一化后，得到的是投影点的实际坐标 $(u_i, v_i, 1)$ ：

$$Normalize(\lambda_i u_i, \lambda_i v_i, \lambda_i) = \frac{(\lambda_i u_i, \lambda_i v_i, \lambda_i)}{\lambda_i} = (u_i, v_i, 1) \quad (2.4)$$

设模型点集合为

$$M = \{\mathbf{M}_i = (x_i, y_i, z_i, 1)^T | i \in [1, n_{model}]\} \quad (2.5)$$

其中 \mathbf{M}_i 指第 i 个模型点， n_{model} 是模型点的数量。

设图像点集合为

$$F = \{\mathbf{F}_i = (u_i, v_i, 1)^T | i \in [1, n_{image}]\} \quad (2.6)$$

其中 \mathbf{F}_i 指第 i 个图像点， n_{figure} 是图像点的数量。

设模型点在投影矩阵作用后的投影点集合为 $Projection(M)$:

$$Projection(M) = \{P_i = Normalize(P * M_i) | M_i \in M\} \quad (2.7)$$

设全部模型点的模型-图像对应点的关系集合为

$$R = \{(i, j) | i \in [1, n_{model}], j \in [1, n_{image}], \text{第} i \text{个模型点与第} j \text{个图像点对应}\} \quad (2.8)$$

“模型点与图像点对应”指图像上的点与该图像点实际对应的模型点，这样的一对点称为模型-图像对应点。

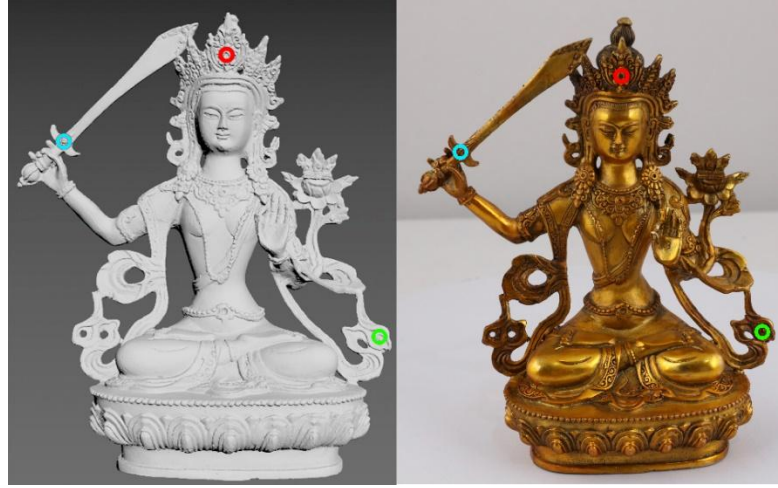


图 2 模型-图像对应点，一共三组

图 3 模型-图像对应点，左边是图像点示意、右边是模型点示意。展示了三组模型-图像对应点，分别是红色的头冠对应点、青色的剑柄对应点和绿色的云带对应点。

要评判一个投影矩阵 P 对某对应关系 $Relation$ 的好坏，一个简单的标准是使每个模型点经过投影后的图像点与它理应对应的图像点的平均距离最小：

$$Distance(P, R) = \frac{\sum_{(i, j) \in Relation} |Normalize(P * M_i) - F_j|}{|R|} \quad (2.9)$$

其中 $|(u, v, 0)^T| = \sqrt{u^2 + v^2}$ 。



图 3 左：模型的正面原始图像，中：一个差的投影矩阵叠放原始图像，右：一个好的投影矩阵叠放原始图像

图 3 中左图是一个文物模型正面的原始图像。

中间图片中的蓝色点，即是该模型点用某一个正面投影矩阵投影，并且归一化后的投影点集合。可以看到的是，由于该投影矩阵不是非常好，所以在投影后的点集中，皇冠、底座、右侧边缘等部位的投影对照情况不好。

右边图片的蓝色点是一个另一个较好的投影矩阵投影后的点。可以看到的是，大部分的点都对应上了。投影点与原图的对应点间距之和大大缩小。

这里对投影矩阵估计的目的就是为一幅图像得到一个它对应的好的投影矩阵。

但是实际应用中，知道所有模型点与图像点的对应关系是非常困难的，所以常用模型点集的某些子集（例如特征点集、凸包子集）代为判断。

即使用

$$Distance(P, R_{sub}), R_{sub} \subseteq R \quad (2.10)$$

去代替(2.9)作为投影矩阵好坏的判断。

另一个需要介绍的点是对投影点，有一个栅格化操作以得到栅格化后的点：

$$Grating((u, v, x)) = ([u], [v], x) \quad (2.11)$$

$[x]$ 表示对 x 向上取整。对于投影得到的投影点，认为栅格化后的图像点是该模型点投影后对应的真正图像坐标，取整处理后再和图像点作比较。

2.2 初始投影矩阵估计

利用已知的模型-图像对应点，使用最小二乘法得到初始矩阵。

2.2.1 得到模型-图像的初始对应点

2.2.1.1 首幅图像的初始对应点

对于首幅图像，采用人工标定的方法得到图像上的点和模型上的点的对应关系。以下介绍一种用 3ds Max 软件和 matlab 图片查看器的标定方法：

1. 3ds Max 软件中打开 obj 文件，并且设置模型的偏移坐标是(0,0,0)，见图 4。

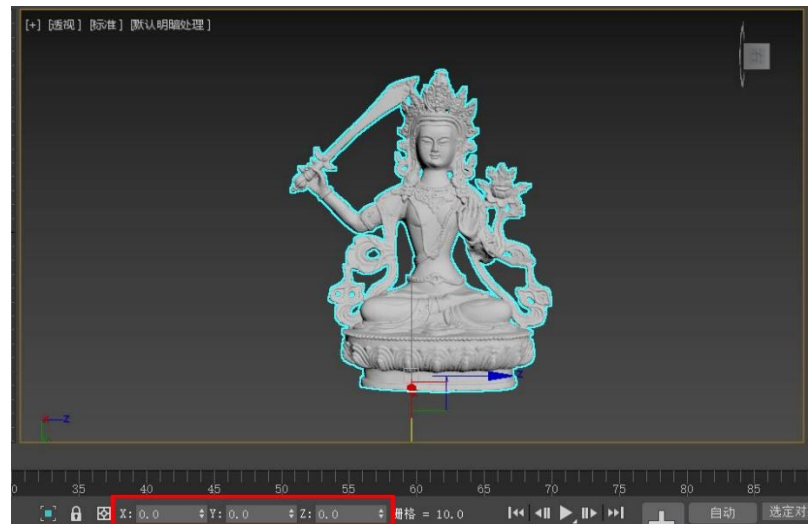


图 4 用 3ds Max 打开 obj 模型文件，并且设置偏移坐标为(0,0,0)

2. 在 3ds Max 中选取模型上的至少 6 个点，并且记录下坐标。以选取模型鼻尖上的一点为例：

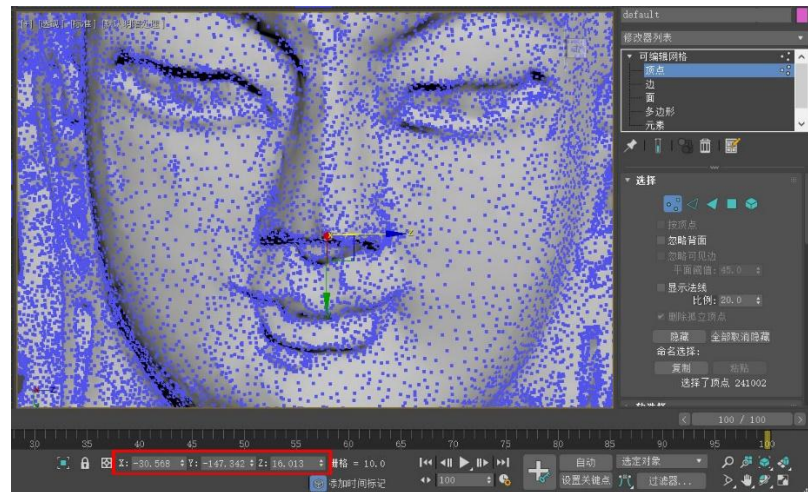


图 5 3ds Max 选取模型点，并记录坐标

图 5，可编辑网络-顶点，点选鼻尖上的点，记录下方该点的坐标。

3. 在 matlab 中用命令 `IMSHOW(IMREAD('图片名称'))` 打开图片，工具-数据提示，点选鼻尖，记录 X,Y 坐标。

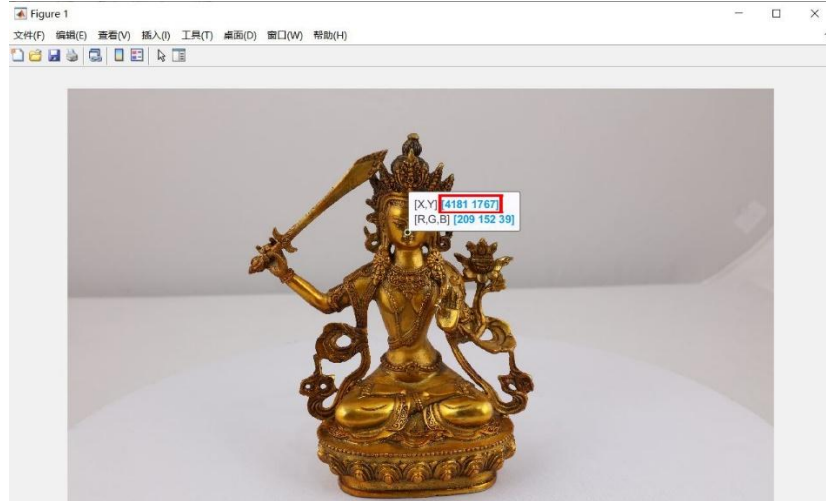


图 6 Matlab 图片查看器选取图像点，记录坐标

4. 整理所得点数据，一一对应并且记录下来，按顺序特定格式记录下来。一个记录的例子：

```
points_description.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
,X,Y,Z,x,y
1,1.266,-117.126,37.809,405,284,食指
2,-3.826,-181.0,16.762,109,390,皇冠中
3,-16.144,-143.94,-58.441,305,723剑柄中
4,-10.465,-191.655,-3.863,77,492,剑尖中
5,-6.019,-71.621,-41.177,629,640,剑端带连接
6,-2.337,-63.643,15.624,652,380,肚脐
```

图 7 一个记录对应点关系的样例

这个样例中，找到(X,Y,Z)对应的模型点的下标*i*，(x,y)对应的图像点的下标*j*，作为一组模型-图像点对应关系 $relation$ ：

$$relation = (i, j) \quad (2.12)$$

记所有的 $relation$ 组成的集合为 R ，它就是人工标定的初始图模型-图像对应点关系：

$$R = \{relation\} \quad (2.13)$$

2.2.1.2 其他图像初始对应点

对于其他图片，将利用已知投影矩阵的图片的信息，用特征提取加上特征点匹配的方法找到它的初始化对应点。具体做法是利用已知图像的投影矩阵，使用 **z-Buffer** ^[27] 算法算出其部分可见点的对应模型点；使用特征提取、匹配算法求得两图像的公共点，使新图像匹配上原图像上最近的有对应模型点的图像点，再让其对应上该点的模型点。这样就得到了新图像的初始对应点信息。

1. 对已知投影矩阵的图像使用 z-Buffer 算法求得可见模型点, 并建立可见模型点的对应关系。

设当前图像的已知投影矩阵为 P 。

对任意图像点 $F_j = (u_j, v_j, 1)$, $u_j, v_j \in \mathbb{N}$, 统计投影后对应它的所有模型点下标的集合 PM_j :

$$PM_j = \{i \in [1, N_{model}] | Grating(P_i) = (u_j, v_j, 1)\} \quad (2.14)$$

记对应当前图像点 F_j 的模型点集 PM_j 中距离相机的最近模型点的下标为 $Nearest(PM_j)$:

$$Nearest(PM_j) = i, i \in PM_j, \forall i_2 \in PM_j, \lambda_i \leq \lambda_{i_2} \quad (2.15)$$

计算所有图像像素, 建立模型点集 M 的可见点下标集 $M_{nearest}$:

$$M_{nearest} = \{Nearest(PM_j) | \forall j \in [1, n_{figure}]\} \quad (2.16)$$

设模型可见点的模型-图像点对应关系集合为 $R_{nearest}$, 这里认为图像点的最近投影点总是可见的, 所以最近投影点就对应了该图像点:

$$R_{nearest} = \{(i, j) | \forall j \in [1, n_{figure}], i = Nearest(PM_j)\} \quad (2.17)$$



图 8 左: 对 M 进行投影, 右: 对 $M_{nearest}$ 进行投影

上图显示了原始模型点集合 M 与使用 z-Buffer 算法对模型点集合进行处理后得到的子集 $M_{nearest}$ 的差别。z-Buffer 算法得到的点更稀疏, 在模型点本身是非常稠密的情况下, 可以保证所有最近点是处于模型的正面部分, 都是可见点, 所以这里认为可见点与它投影后

对应的图像点是实际对应的。

2. 使用特征提取、识别算法获得两图像间的公共点

同一对象的两幅图像间有很多相似的点，由此想到用基于特征提取、识别出来的公共点匹配数据来给下一幅图片提供初始模型-图像对应点。对已求得投影矩阵的旧图像和一幅未求得投影矩阵的新图像间求出公共点，用将新图像的公共点和旧图像的公共点对应的模型点作为新图像的模型-图像对应点信息。

对于两幅图像，设图像点集合分别是 F_a, F_b ，其中 F_a 是已求得投影矩阵的图像，并且用 1 步骤中的 z-Buffer 算法已求得其可见点的对应关系 $R_{nearest}$ 。

使用 KAZE 算法^[28]分别对两图像求特征点，再使用特征点匹配算法求得两图像特征点间的对应点，取可信度最高的前十组点，记为集合：

$$Matched = \{(i, j) | F_{ai} \text{ 和 } F_{bj} \text{ 互为图像对应点}\} \quad (2.18)$$



图 9 KAZE 算法找到对应公共点，左：两幅原图，右：对应的 10 个公共点

图 9 展示了使用 KAZE 算法找到的两幅图像的前十个最可信的公共点。

由 $Matched$ 和 $R_{nearest}$ ，求得新图像的初始模型-图像对应点关系如下：

$$R_{new} = \{(m, j) | \forall (i, j) \in Matched, (m, i) \in R_{nearest}\} \quad (2.19)$$

无论是人工标定得到的 R ，还是计算得到的 R_{new} ，以下统称为 R_{init} ，表示一幅图像的初始模型-图像对应点关系。

2.2.2 使用最小二乘法得到初始投影矩阵

估计投影矩阵的算法中最经典的算法就是最小二乘法^[29]。最小二乘法将已知的模型-图像对应点作为目标量，把求解矩阵的问题转化成为一个求最小值问题。最终求解一个能使已知图像-模型点的投影点与对应的图像点距离之和最小的投影矩阵。记作：

$$P_{init} = GetInitProjectionMatrix(R_{init}) \quad (2.20)$$

最小二乘法的好处在于，如果给定的初始模型-图像对应点的对应情况好，那么能得到

一个不错的结果，并且可以使用多组初始对应点。

最小二乘法的坏处在于只考虑了给定的对应点情况，一点也没有考虑其他模型点、图像点。而且对于单值敏感，单个的错误对应点会严重影响估计的投影矩阵效果。

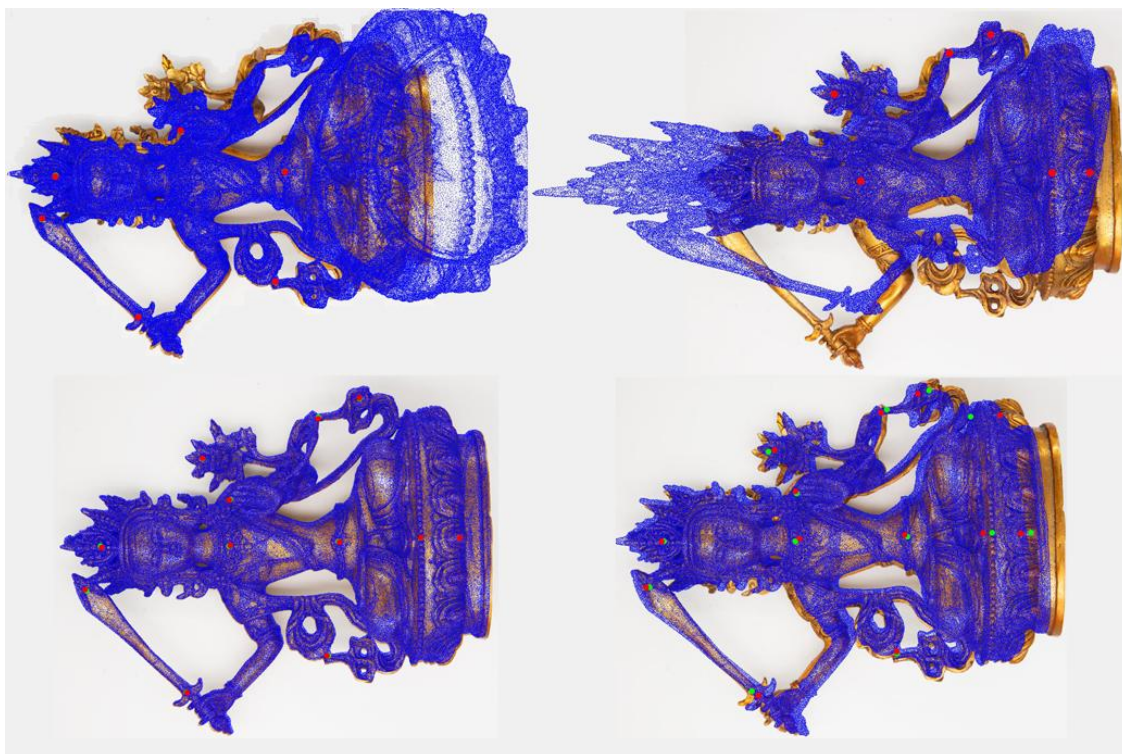


图 10 左上：前 6 组对应点求得的投影矩阵，右上：后 6 组对应点求得的投影矩阵，左下：用全部 12 组对应点求得的投影矩阵，右下：用 12 组对应点外加 1 个错误对应点求得的投影矩阵

图 10 左上图中用到前 6 组模型-图像对应点（红色点），分布在佛像的头部、右手侧。用最小二乘法得到的投影矩阵对于给定的对应点的对应情况非常好，但是对于其他部分：底座、左手侧的对应情况就非常不好。

图 10 右上图使用的是后 6 组对应点，与左上图相反，投影矩阵对于佛像右手侧部分的对应情况优于佛像头部、右手侧，因为六组初始模型-图像对应点集中在佛像底部、左手侧。

图 10 左下图使用全部的 12 组对应点，投影矩阵非常好。

图 10 右下图使用 12 组对应点，外加 1 个错误对应点。错误对应点位于佛像的底座左侧处，对应图像点错误左偏移 100 像素。从图中可以看到，投影矩阵受单个错误点的影响非常大。

对于初始图像，它的初始模型-图像对应点是人为标定的，可以认为是准确的，所以一般初始图像求得的初始投影矩阵效果不错。但是对于其他图像，由于它们的初始模型-图像对应点是计算得到、在线生成的，常常出现错误的对应点、不准确的对应点。这些错误的对应点使得其他图像的初始投影矩阵非常不好。

所以以下会使用一种基于凸包的 ICP 算法去优化初始投影矩阵。

2.3 基于凸包的 ICP 算法优化投影矩阵

对于投影后的投影点和原始图像点分别求出各自的凸包^[30]子集，再取两子集中最近点对作为新的凸包模型-图像对应点。再使用初始模型-图像对应点和凸包模型-图像对应点共同使用最小二乘法求得新的投影矩阵。迭代此过程，直至得到较好的投影矩阵。

设模型点集为 $M = \{M_i = (x_i, y_i, z_i, 1)^T\}$ ，图像点集为 $F = \{F_i = (u_i, v_i, 1)^T\}$ ，图像的初始投影矩阵为 P_{init} 。

设所有模型点经过初始投影矩阵生成的投影点集为 PM ：

$$PM = \{P_i = \text{Normalize}(P_{init} * M_i) | M_i \in M\} \quad (2.21)$$

对于投影点集合 PM ，直接应用求凸包的算法即可得到投影点的凸包集合 C_{PM} ：

$$C_{PM} = \text{GetConvex}(PM) \quad (2.22)$$

用 C_{PM} 作为筛查标准，得到投影点的凸包对应的下标集合 CI_{PM} ：

$$CI_{PM} = \{i \in [1, n_{model}] | P_i \in C_{PM}\} \quad (2.23)$$

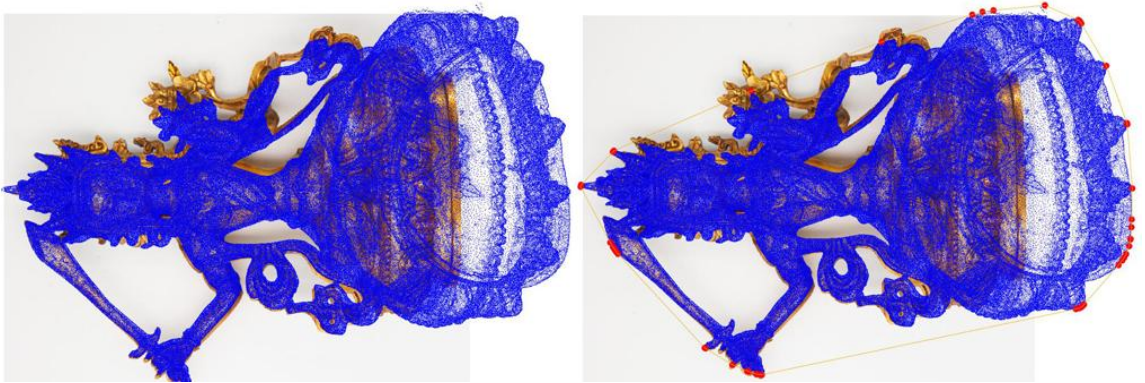


图 11 左：投影点集合，右：投影点的凸包

图 11 右图中的红色点就是蓝色点（投影点）的凸包点集，并用橙线连接。

对于图像点，要提取出图像点的凸包，需要筛选出物体对应的图像点，排除背景对应的图像点。最简单的筛选物体图像点的方法是按照明暗度来区分，背景默认是纯白色。

按图像坐标取出图像的 rgb （红黄蓝系数）函数 $GetRGB$ 为：

$$(r, g, b) = \text{GetRGB}(F_i), \quad r, g, b \in [0, 1], i \in [1, n_{figure}] \quad (2.24)$$

那么假定物体的图像点不接近白色，即 rgb 均小于 0.8，可以得到物体图像点集合 F_{item} ：

$$F_{item} = \{F_i | \text{GetRGB}(F_i) < (0.8, 0.8, 0.8), \forall i \in [1, n_{figure}]\} \quad (2.25)$$

对物体图像点求凸包 C_{item} :

$$C_{item} = GetConvex(F_{item}) \quad (2.26)$$

同样地，用 C_{item} 去筛选出物体点凸包的下标集合 CI_{item} :

$$CI_{item} = \{i \in [1, n_{figure}] | F_i \in C_{item}\} \quad (2.27)$$



图 12 左：原始图像，右：图像的凸包

图 12 右图中的绿色点就是物体图像点的凸包点集，并用橙线连接。

定义一个最近距离函数 $GetNearest$ ：找到投影点凸包下标集合 CI_{item} 中距离投影点 P_i 最近的凸包图像点的下标：

$$GetNearest(i) = j \in CI_{item}, \quad \forall k \in CI_{item}, |F_k - P_i| \geq |F_j - P_i| \quad (2.28)$$

其中 $|(u, v)| = \sqrt{u^2 + v^2}$ 。

为投影点集的每个凸包点 P_i , $i \in CI_{PM}$, 在 CI_{item} 中找到离它最近的图像点的下标 j 作为新的模型-图像对应点：

$$R_{convex} = \{(i, j) | i \in CI_{PM}, j = GetNearest(i)\} \quad (2.29)$$

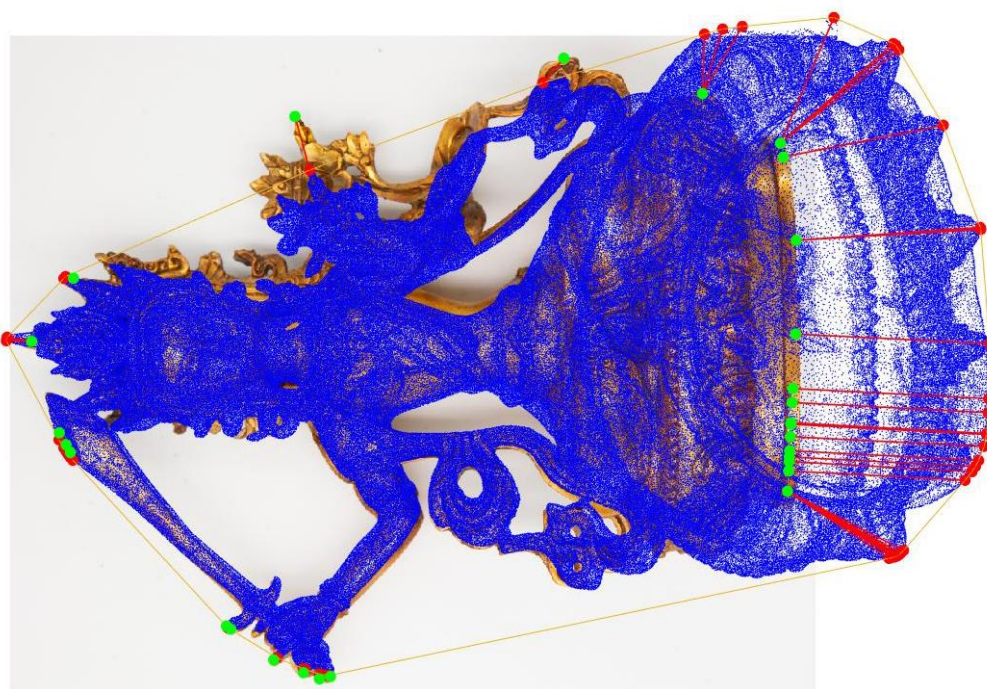


图 13 凸包模型-图像对应点

图 13 展示了凸包点集 R_{convex} 的一个示例，图中红色点是投影的凸包点集、绿色点是图像点的凸包点集。红线连接表示模型-图像点的对应情况。

实际应用中， R_{convex} 有许多冗余信息的对应点，常表现为密集出现、相距过于接近等特点。对这些冗余的对应点进行消除：

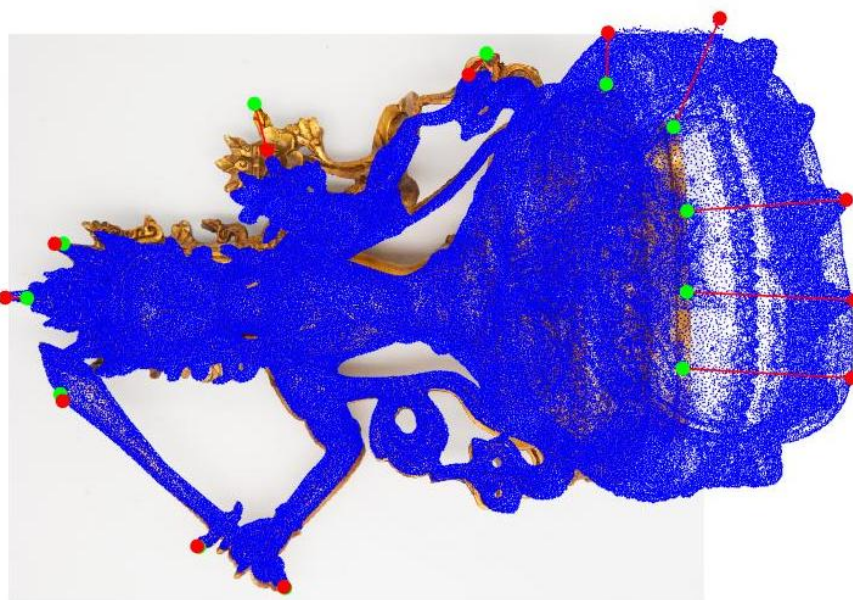


图 14 去除冗余后的凸包模型-图像对应点

图 14 展示了去除冗余模型-图像对应点后的剩余对应关系。实际应用中，应考虑某些凸包点的错误对应关系，所以还会为每组对应点设定一个认定有效的随机概率，使得每组对应点有一定概率被作为实际凸包对应点使用。

至此，通过对图像点、投影点凸包的探索，获得一组新的模型-图像对应点 R_{convex} ，它

与初始的 R_{init} 组成新的图像、投影点对应关系： $R_{new} = R_{convex} \cup R_{init}$ 。

将 R_{new} 作为新的初始模型-图像对应关系，重新进行初始矩阵估计，再次优化。如此迭代 20 次，并逐步减少 R_{init} 的对应点数量，就能得到一个合理的投影矩阵了：

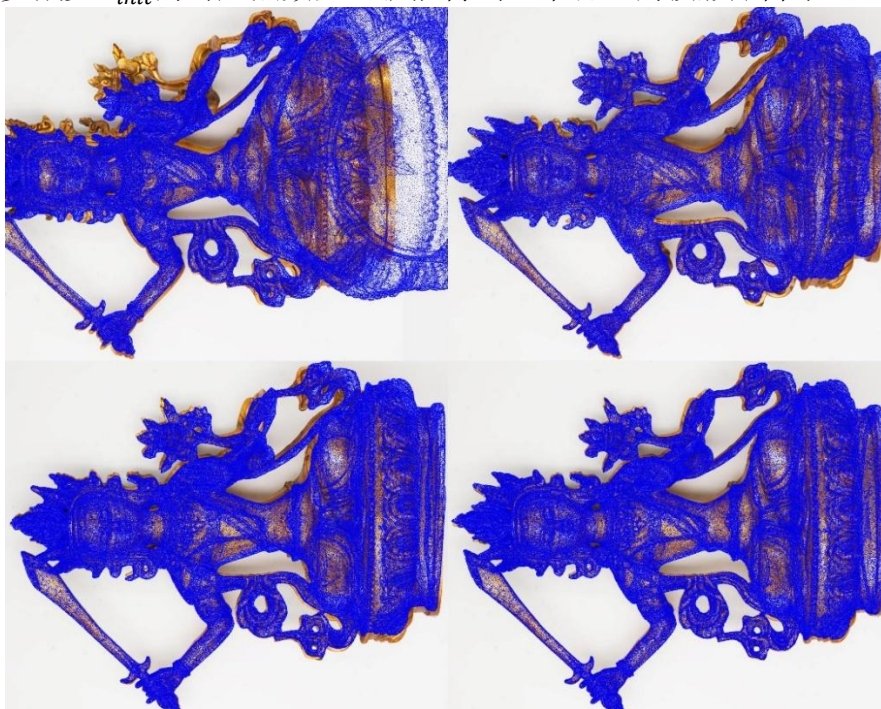


图 15 左上：初始投影矩阵，右上：迭代 1 次，左下：迭代 10 次，右下：迭代 20 次

图 15 展示了不同迭代次数下，投影矩阵从劣变优的过程。

本方法估计投影矩阵的好坏主要与以下因素有关：

图像中物体的边缘亮度，关乎图像点闭包点集的准确性。

初始模型-图像点 R_{init} 的可靠性，如果不可靠的点过多，需要调整迭代过程中减少使用 R_{init} 的速度。

图像中物体本身的棱角是否分明，如果是一个中心对称、圆、边缘近乎直线的物体，可能会使图像点、模型点的凸包匹配不当。

3 基于投影矩阵的真实感纹理贴图

3.1 纹理贴图简介

纹理相当于三维模型的皮肤，三维模型在被展示的时候，会依据纹理来赋予该模型点、模型面片不同的性质。纹理不局限于对颜色（反射光线）的限制，也可以是法线、折射率、吸收率、凹凸程度。本文只考虑最直接的纹理，漫反射纹理。

obj 三维模型文件中会规定使用的 mtl 材质包文件：

mtllib XXX.mtl

并为每个三维模型体规定使用的默认材质：

usemtl mat2

使用的 mtl 材质包文件中会规定每种材质使用的漫反射纹理贴图：

map_Kd tex\zenigame_0_0.tga

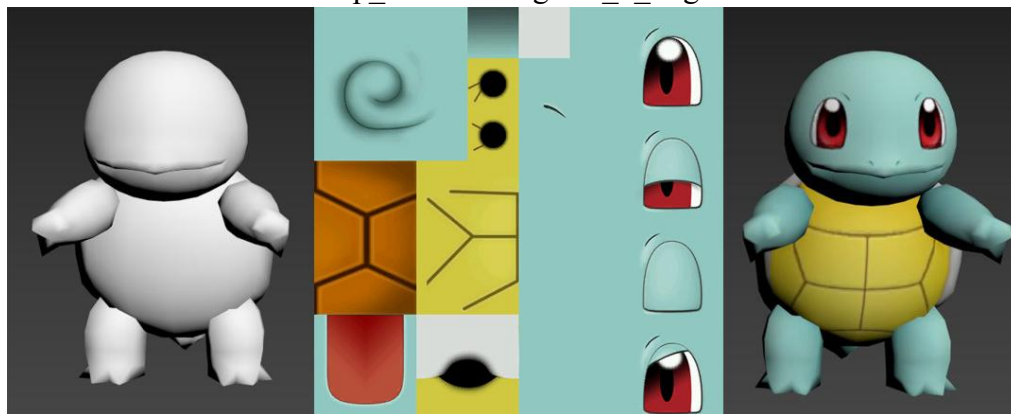


图 16 左：原始模型，中：纹理贴图，右：带纹理的模型

图 16 中间的图片是纹理贴图，它对应贴图后的模型效果是右图。对于中间的纹理贴图，它相当于是把三维模型表面给分块后，二维化，并重用了一些重复的贴图所生成的。模型三角面片找到对应纹理贴图块的方法是通过给定坐标实现。一些 obj 文件中为每个三角面片规定了模型-纹理点对应关系，而另一些 obj 文件则需要使用 CGAL 帮助生成模型-纹理点的对应关系。

单幅图像常用的确定纹理坐标的方法是求出每个模型点对应的图像坐标。但是对于多幅图像，由于每个模型点会对应多幅图像各自的图像点坐标，这里采用一种基于模型点颜色的漫反射纹理贴图生成办法。

3.2 可见模型点对应的颜色

生成三维模型纹理贴图的最重要的一点是确定每个模型点的颜色，通过图像的投影矩阵可以很方便得到单幅图像可见模型点的颜色。

多幅图像间采用取平均值的办法确定可见模型点最终的颜色。

3.2.1 使用单幅图像确定可见模型点颜色

设模型点集为 $M = \{(x, y, z, 1)^T\}$ ，图像点集为 $I = \{(u, v, 1)^T\}$ ，投影矩阵为 P ，使用 z-Buffer 算法得到它的可见模型点下标集 $M_{nearest}$ 。对于每个可见点 $i \in [1, n_{figure}]$ ，算出它的归一化后的对应图像点，使用该图像点的颜色作为该可见模型点的颜色 $ModelRGB(i)$ ：

$$ModelRGB(i) = GetRGB(Grating(Normalize(P * (x_i, y_i, z_i, 1)^T))), \quad i \in M_{nearest} \quad (3.1)$$

在投影矩阵适当的情况下，每个可见模型点都能找到对应的图像点。如果投影矩阵不好，则会出现可见模型点找不到对应图像点的情况，此时应该检查投影矩阵是否适当。

如果找不到对应图像点的可见模型点的数量在可接受范围内(< 1%)，可以认为这些可见模型点没有对应的颜色，不作为已确定颜色的可见模型点。

3.2.2 多幅图像确定可见模型点颜色

对于所有图像 $SF = \{F\}$ ，同一个可见模型点 $i, i \in [1, n_{model}]$ 的颜色可能被确定多次，采用取平均值的办法确定该可见模型点的颜色：

$$ModelRGB(i) = \frac{\sum_{F \in SF} ModelRGB_F(i)}{|SF|}, \quad \forall i \in M_{nearest} \quad (3.2)$$

3.3 使用可见点颜色生成纹理贴图

通过可见点可以生成纹理贴图，生成思路有三种：一是沿用图像作为纹理图片、重构模型点与纹理点的对应关系。二是沿用模型点与纹理点的对应关系，重构纹理图片。三是重新构造模型-纹理点的对应关系与和于对应关系想应的纹理图片。

本文采用第三种思路。

将三维模型的表面二维展开作为纹理基图，三维模型点与二维图像点的对应关系就是模型-纹理点的对应关系。这里使用 CGAL 中将三维模型二维展开的算法，注意这里的二维展开与投影的区别：投影只能兼顾三维物体的一面的二维化，而二维展开会把模型的所有表面体现在结果中。

具体来说：

使用 CGAL 算法生成三维模型的二维展开图作为纹理基准，并记录纹理点 $T_i = (u, v)$ 到模型点 $M_j = (x_j, y_j, z_j, 1)^T$ 的对应函数 $GetModel(T_i)$ 。

纹理点的颜色就是与之对应的模型点的颜色：

$$GetModel(T_i) = ModelRGB(j), \quad j \in M_{nearest} \quad (3.3)$$

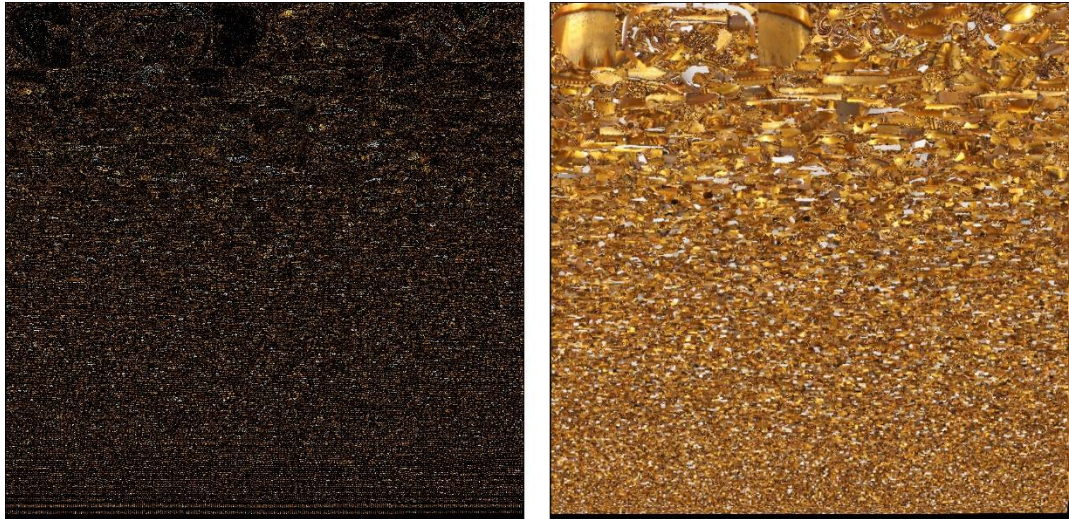


图 17 左：依据纹理点颜色直接生成纹理贴图，右：拟合后的最终纹理贴图

图 17 左图就是直接用已知颜色的纹理点信息绘制的纹理贴图，可以看到有大部分的纹理贴图是黑色的，即没有对应的模型点信息，所以默认是黑色的。

3.4 拟合得到最终的纹理贴图

将对已知纹理点颜色的像素作为已知的量，用自然邻点插值^[31]的方法得到未知的纹理颜色，就能给未知颜色信息的纹理点添加上一个适当的颜色值。

图 17 右侧图片就是自然邻点插值后得到的最终纹理贴图。

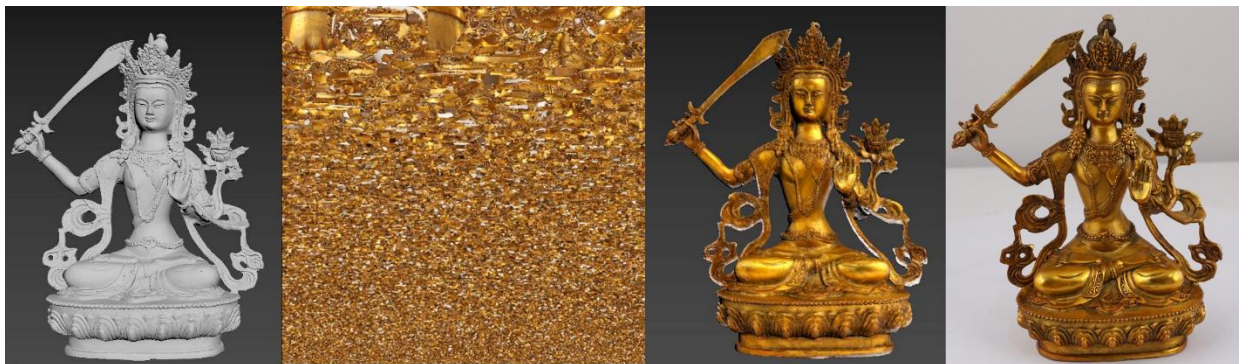


图 18 左 1：原模型，左 2：纹理贴图，左 3：贴图后的模型，左 4：原图像

4 总结与展望

4.1 本文总结

本文在已有的纹理贴图技术的基础上，开发了一个可自动生成纹理贴图的方法。在使用不同角度的几十张图片和已有的三维物理模型的基础上，仅需人工标定一张图片的 6 个模型-图像对应点，即可生成含有完整颜色信息的三维模型贴图。

在估计投影矩阵的过程中，本文利用了最小二乘法等数学方法，可以经过多次迭代，得到准确的投影方式。ICP 算法用于对投影矩阵的计算进行优化。本文采用计算凸包的方法，减少了轮廓线计算中的误差。

本文使用 z-Buffer 算法，可以得到可观察的三维模型上的点在图片上的对应点。该方法对于文物这种复杂度较高的物体，可以以较小的时间复杂度完成计算，方便地处理大批量图像。

KAZE 特征提取算法用于求出已知投影矩阵的图片与未知投影矩阵的图片上面的公共对应点，在此基础上可以计算两幅图像上的点与模型上的点的对应关系。对于形状较为复杂的物体，该方法可以方便地将多个角度的图像纳入模型中，完整地展示模型各个角度的信息。对于表面凹凸不平花纹较多的物体，该方法的结果更加精确。

在融合全部图片的颜色信息后，最终可生成 obj 文件对应的纹理贴图，以 bitmap 形式保存，可以使用 3Dmax 等软件打开，查看效果。

4.2 未来展望

本文中还可以进行的主要优化工作列举如下：

本文中优化投影矩阵的过程中，按像素明亮程度来区分图像边缘过于绝对化，不能照顾一些较暗、较亮的图像，亦不能照顾一些复杂背景的图像。可以尝试改用主成分分析等其他办法区分图像的边缘。

本文只尝试了生成纹理贴图的第二种思路，可以尝试第一、三种思路，做出来的纹理贴图效果可能会更好。

本文对多幅图像重叠部分采用直接取均值的办法，未照顾到如果多幅图像之间有明暗差异、高亮角度等不同点。可以先尝试对所有图片进行直方图均衡化，这样能缓解由于光线不均带来的影响。

本文综合了已有的物体 3D 建模相关方法和纹理贴图相关方法，可以自动化进行三维模型贴图。本文结合前人已有的成熟方法，将不同算法的优点加以整合，贴图效果好，

计算速度快，鲁棒性高。可以大批量在不同模型中进行重复。

在文物保护领域，本文的成果可以用于 3D 模型制作，加快文物模型的制作速度，将博物馆的大量展品进行 3D 建模，便于在互联网上进行展示。对于文物，仅需不同角度的图片就可以生成对应的模型贴图，纹理效果好，可以减少文物在观赏过程中的自然损耗，能保护文物原貌。对于形状复杂的文物，不需要接触就可以得到其纹理材质的信息，形状细节等都可以原封不动地保持与原物一致，可以在展示过程中给观众真实感。

尽管本文在算法方面和实际应用方面都有创新及改进，但未来在此基础上，还可以有进一步的研究以使得模型制作更加方便。算法方面，由于纹理贴图相关技术并不完善，照片数量较多时，需要较长的时间才能贴图完成。对于投影矩阵的运算方法较为固定，期待未来对投影矩阵估计能有时间复杂度更低的算法。对于贴图部分，图像数量多，算法运行时间长，对计算机的使用时间长，在批量处理文物时较为困难。实际应用中，尽管本方法可以自动化进行，但对文物的取像仍需要较长时间，人工进行拍照，且需要保持光线和背景类似，背景不能有杂物遮挡。对每件文物的贴图过程中，需人工标定至少 6 个点，可以考虑采用固定相机与文物位置的方法，从而提前知晓各图像初始投影矩阵。

参考文献

- [1]张瑞, 骆岩林, 周明全,等. 文物数字化的关键技术[J]. 北京师范大学学报(自然科学版), 2007, 43(002):150-153.
- [2]王静. 3DS MAX 6 标准教程(附光盘)[M]. 清华大学出版社, 2006.
- [3]李丹. 面向三维激光扫描的无变形纹理映射技术研究[D]. 南京理工大学, 2008.
- [4]王淼. 基于多角度图像的三维模型纹理映射方法研究[D]. 西北大学.
- [5]倪宏宇. 三维数字影像测量技术综述[J]. 测绘科学, 2006(02):10+134-136.
- [6]Weinhaus F M , Devarajan V . Texture mapping 3D models of real-world scenes[J]. Acm Computing Surveys, 1997, 29(4):325-365.
- [7]王若苏, 任洪娥, 邱兆文. 文物三维重建技术综述[J]. 科技信息, 2007.
- [8]张英杰, 张铁昌. 纹理映射技术的发展综述[J]. 计算机工程与应用, 1994, 000(003):1-4.
- [9]Catmull, Edwin. Computer display of curved surfaces[J]. Proc. IEEE Conf. Comput. Graphics Pattern Recognition Data Struct, 1998, 11.
- [10]Catmull E , Rom R . A Class of Local Interpolating Splines[J]. Computer Aided Geometric Design, 1974, 74.
- [11]Debevec P , Yu Y , Borshukov G . Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping[C]// Eurographics Workshop on Rendering Techniques. Springer, Vienna, 1998.
- [12]Blinn J F . Simulation of Wrinkled Surfaces[J]. ACM SIGGRAPH Computer Graphics, 1978, 12(3):286-292.
- [13]Kajiya, James T . Anisotropic reflection models[C]// ACM. ACM, 1985:15-21.
- [14]Blinn, Newell J F , Martin E . Texture and Reflection in Computer Generated Images[J]. Communications of the ACM, 1976, 19(10):542-547.
- [15]Williams L R . Casting curved shadows on curved surfaces[M]. ACM PUB 27 New York, NY, USA, 1978.
- [16]Fournier A , Romanzin C . Common Illumination between Real and Computer Generated Scenes[J]. Proceedings Graphics Interface, 1993:254-262.
- [17]Jerzy S , N Sławomir. LARGE TEXTURE STORAGE USING FRACTAL IMAGE COMPRESSION[C]// International Conference on Computer Vision and Graphics (ICCVG 2004). Institute of Computer Science, Warsaw University of Technology, ul Nowowiejska 15/19, 00-665 POLAND, e-mail: J.Stachera@elka.pw.edu.pl; Institute of Control and Computation Engineering, University of Zielona Góra, ul Podgórna 50, 65-246 Zielona Góra, POLA, 2006.
- [18]Larson R D , Shah M S . Method for generating addresses to textured graphics primitives

stored in rip maps: US, US5222205 A[P]. 1993.

[19]Zhang E , Mischaikow K , Turk G . Feature-based surface parameterization and texture mapping[J]. Acm Transactions on Graphics, 2005.

[20]Maillot J , Yahia H , Verroust A . Interactive texture Mapping[C]// ACM. ACM, 1993:27-34.

[21]Oliveira M M , Bishop G , Mcallister D . Relief Texture Mapping[C]// SIGGRAPH conference. 2000.

[22]Hakura Z S , Gupta A . The design and analysis of a cache architecture for texture mapping[C]// Conference International Symposium on Computer Architecture. IEEE Computer Society, 1997:108-120.

[23]Cabral B , Cam N , Foran J . Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. DBLP, 1994:91-98.

[24]樊萌、蒋林、邓军勇、刘洋. 图形处理器中纹理贴图算法的性能分析[J]. 西安邮电大学学报, 2020, v.25;No.143(02):78-83.

[25]李利民, 董民, 慕银锁,等. 图片的纹理贴图方法和装置:, CN108122196A[P]. 2018.

[26]Moons T U , Gool L V , Diest M V , et al. Affine structure from perspective image pairs under relative translations between object and camera[J]. 1993.

[27]Greene N , Kass M , Miller G . Hierarchical Z-Buffer Visibility[J]. ACM SIGGRAPH Computer Graphics, 1993, 27.

[28]Alcantarilla P F , Bartoli A , Davison A J . KAZE Features[J]. Lecture Notes in Computer Science, 2012, 7577(1):214-227.

[29]张捷, 李新德, 戴先中. 基于立体靶标的摄像机标定方法[J]. 东南大学学报(自然科学版), 2011(03):543-548.

[30]崔国华, 洪帆, 余祥宣. 确定平面点集凸包的一类最优算法[J]. 计算机学报, 1997(04):330-334.

[31]Sibson R . A Brief Description of Natural Neighbor Interpolation[C]// Interpreting Multivariate Data. 1981.

附 录

无

致 谢

感谢段福庆老师为我的毕业论文提供研究方向和基本思路，并指导我的研究过程和论文撰写。

感谢课题组的师兄师姐为我的研究提供了图像、模型素材。

袁宇昊

2021 年 4 月