

浙江大学学士学位论文

主 题：虚拟现实系统中彩塑的阴影
 克隆

作 者：陈 兆 基

专业班级：竺可桢学院混合班 984 班

指导老师：鲁 东 明 教授

二零零二年六月十四日



目 录

目 录.....	1
摘 要	2
Abstract.....	3
第一章 前 言.....	4
第二章 虚拟现实背景.....	5
2.1 虚拟现实的概念.....	5
2.2 虚拟现实的特征.....	5
2.3 虚拟现实技术的组成.....	6
2.4 虚拟现实与仿真.....	8
2.5 虚拟现实系统的类型.....	8
第三章 敦煌石窟漫游系统简介.....	10
3.1 系统组成简介.....	10
3.2 石窟建筑结构和壁画模型.....	11
3.3 石窟彩塑模型.....	11
3.4 敦煌彩塑建模与绘制.....	11
第四章 彩塑阴影生成问题和克隆思想.....	13
4.1 纹理映射（Texture Mapping）.....	13
4.2 彩塑阴影问题.....	16
4.4 阴影克隆.....	16
第五章 阴影克隆的具体实现.....	18
5.1 阴影克隆中源数据的获取。.....	18
5.2 阴影克隆中使用到的 2D 图像变形算法.....	19
5.2.1 多项式卷绕（Polynomial Warping）.....	19
5.2.2 网孔弯曲（Mesh Warping）.....	19
5.2.3 基于特征线段的图像变形.....	21
5.3 阴影克隆中对颜色的处理.....	25
5.3.1 彩色规范.....	25
5.3.2 彩色坐标变换.....	28
5.4 阴影克隆中对灯光强度的处理.....	30
第六章 总 结 与 展 望.....	32
6.1 总结.....	32
6.2 问题和扩展.....	32
第七章 致 谢.....	34
参 考 文 献.....	35



摘 要

目前,对于虚拟现实的技术研究倍受注目,其应用领域涉及从娱乐、军事、医疗到建筑、工业、远程教育,艺术等方方面面。我参与的“石窟漫游系统”作为“多媒体与智能技术集成及艺术复原”的一个子系统,在其他子系统的基础上,将提供基于桌面的访问、浏览敦煌石窟的多媒体界面。

阴影是真实感绘制的主要组成。阴影实现包括物体在周围环境中投下的阴影和光照下物体在本身上形成的阴影两部分。彩塑因为形状不够规整,它的阴影需要根据表达方式,重新设计和实现。真实感是虚拟现实系统很重要的特性。而对于漫游系统,图像的绘制速度也是重要的指标。本文的主要内容将针对如何尽量快速的生成彩塑表面阴影的问题展开。

因为彩塑的形状大体一致,如果能事先获取某一模型在不同灯光下的表面阴影情况作为原始数据。对于其他模型,参照源模型表面的变化情况,对模型表面产生类似的变化,则可以大大加快阴影的生成速度。为了实现这一目标,本人主要从事了以下几个方面的工作:

● 图像变形

我们得到的资源数据都是源模型的二维图像。目标模型与源模型只是近似而非完全一致,为了保证阴影生成的准确性,需要对两个模型进行变形匹配。

由于彩塑模型造型类似,而且往往变化体现在姿势不同上,有明显的边界和面部五官,比较容易提取特征线段,因此本系统采用基于特征线段的变形。

● 颜色处理

通常图像都是以 RGB 格式存储,但因为灯光主要引起彩塑表面的亮度变换,如果使用 RGB 格式,则每个像素都需要对 R、G、B 三个分量分别进行计算,计算量大。为了简化运算量,选择在阴影克隆前将读入的目标图像和源图像都转化为 HSI 格式,当克隆完成后在恢复为 RGB 格式。这样对于每个像素点只需要计算亮度值(I 值)。

● 强度处理

因为灯光强度对彩塑表面的影响接近于线性,采取在最后处理时添加强度因子 λ 来进行控制。通过源模型来计算 λ 。获取源模型在仅仅改变灯光强度的两张图片,计算其与源模型图像 I 值的变化,两个变化量的比值除以灯光强度的比值就是 λ 。

系统对于设定的灯光情况在资源库中选择相应的灯光效果图像,对于目标图像的每一个像素点,用变形算法得到与之相对应的源图像点坐标,从灯光效果图像获取源模型在该点的亮度变化,然后根据相应的比例系数调节目标图像点的亮度值,从而实现阴影生成。从输出的情况看,基本能够实现类似的阴影效果,但在细节方面还不够理想,主要需要对图像对应关系的计算做进一步的改进。

关键词:

阴影克隆, 阴影生成, 虚拟现实, 真实感绘制, 纹理映射



Abstract

Now days, the research about Virtual Reality Technology earns more and more observations. The “Carven Roam System” is a sub system of the “Integration of Multimedia and Intelligent techniques and Art Restoration”. On the basis of the other sub systems, it provides a multimedia interface to visit and browse the Dunhuang Carven on desktop.

Shade is a main part of the Realistic Rendering. Shade creation includes shade projected by the object to the surrounding environment and shade on the surface of the object created by the light. Since its shape is not very regular, the shade of those painted sculptures needs re-design. Realistic Rendering is a very import property of the virtual reality system. And for the roam system, the rendering speed is also a very important target. This paper is focus on how to create the surface shade for painted sculpture as soon as possible. In order to achieve this goal, I focus on the following things:

- **Image morphing**

The source data we get is the 2D image of the source modal. The target modal is similar but not identical, so we need to find the correspondence between the two modals.

Since painted sculptures are similar, usually differ in gestures, there are clear boundaries and facial features could be used as control line. I use Line Based Morphing in this system.

- **Color management**

The image is usually stored in RGB format, but since lights mainly the brightness of the sculpture surface, if we use RGB, every pixel needs three calculations according to R, G, B. It could be a time-consuming calculation. In order to reduce calculations, we choose to change the images to HSI format before the Shade Cloning, and change it back to RGB after the Shade Cloning is completed. And now, every pixel needs only one calculation, just for the brightness (I).

- **Intensity Control**

Since the light intensity approximately causes a liner effect on the sculpture surface, I choose to add an intensity factor to control the light effect. I get two images of the source modal only differ in the light intensity. Calculate the respective change in brightness. The ratio of the two changes to the light intensity is λ .

We first choose an appropriate light effect image according to the light setting. Then, for every pixel in the target image, use morphing algorithm to find the corresponding pixel in the source modal. We get the surface change of the source modal from the light effect image, and change the lightness in the target image. When every pixel finishes, the shade creation is done. From the output image we could see that a similar shape effect has been created for the target image. But on the boundary, the effect is not so good, we need to make some adjustment to the mapping algorithm.

Keywords:

Shade Cloning, Shade Creation, Virtual Reality, Realistic Rendering, Texture Mapping



第一章 前言

目前，对于虚拟现实的技术研究倍受注目，其应用领域涉及从娱乐、军事、医疗到建筑、工业、远程教育，艺术等方方面面。一些相关的关键技术和硬件平台正日趋成熟起来，并逐渐由专业型应用转向大众型应用。而且，这种三维的绘制技术很可能成为未来的用户界面和人机接口。

我参与的“石窟漫游系统”作为“多媒体与智能技术集成及艺术复原”的一个子系统，在其他子系统的基础上，将提供基于桌面的访问、浏览敦煌石窟的多媒体界面。它集成文字、图像、声音、视频以及三维模型等多种数据，为用户提供了一个全方位的友好、方便的浏览工具。可以说，这部分系统是整个敦煌项目的三维的用户界面。

系统采用了纹理映射（**texture mapping**）的方法，即通过建模生成物体的形状，然后将真实物体的照片贴在模型表面。这样的方法容易实现，实际效果也不错。但这种方法也存在缺陷。系统使用的照片都是从正面打光拍摄的。在这样的系统里，如果对一个物体从侧面的其他角度使用灯光，由于用来展示物体表面情况的照片本身没有阴影，系统也无法虚拟侧面灯光造成的阴影。如果为了得到阴影的照片而对真实物体在各种灯光条件下都进行拍照，代价太大。

阴影是真实感绘制的主要组成。阴影实现包括物体在周围环境中投下的阴影和光照下物体在本身形成的阴影两部分。敦煌石窟中，能够形成阴影的主要部分是中心柱，中心佛坛和彩塑。其中，中心柱和中心佛坛形状较为规整，可以比较容易地简化为标准图形处理；彩塑的阴影则需要根据表达方式，来重新和实现。

真实感是虚拟现实系统很重要的特性。而对于漫游系统，图像的绘制速度也是重要的指标。本文的主要内容将针对如何尽量快速的生成彩塑表面阴影的问题展开。在接下来的第二章中将主要介绍虚拟现实的背景。第三章将简单介绍敦煌石窟漫游系统。第四章讨论彩塑阴影克隆方法的思想。第五章将结合阴影生成系统重点讨论在具体实现中的如果获取源数据，处理变形，颜色以及光照的强度。第六章将探讨对这一方法可能进行的改进。



第二章 虚拟现实背景

2.1 虚拟现实的概念

1965 年,美国学者 Ivan Sutherland 在他的博士论文中提到了虚拟现实这样一种思想,这是最早的对虚拟现实的研究。虽然美国科学家 Willian Gibson 于 1984 年最早提出了“虚拟现实”(Virtual Reality-VR)一词,但大多数人仍认为真正赋予“VR”科学含义的应是另一位美国科学家 Jaron Lanier,他于 1989 年提出使用“VR”一词来统一表述新的人机交互技术。关于 VR 这个词的汉译,有的人认为“虚拟现实”一词译法不佳,而主张翻译成“灵境”,另外还有译为实时环境、虚拟空间、人造现实等。但在科学界,大多数人仍主张直译为虚拟现实。

虚拟现实是人们通过与计算机之间相互进行信息交流来再现设计者头脑中的世界的一种交互式方法。

2.2 虚拟现实的特征

理论上的 VR 系统应具有 4 个基本的特征:

(1) 沉浸感 即计算机使用者已感觉不到自身所处的外部环境,而是“融合”进了计算机所给出的虚拟的世界中;

(2) 交互性 即计算机使用者可以通过三维交互设备直接操纵计算机所给出的虚拟世界中的对象,虚拟世界中的对象也能够实时地做出相应的反应;

(3) 多感知 即计算机使用者可以通过 VR 系统感知其所具有的视、听、触、嗅、味等感觉信息;

(4) 自主性 即 VR 系统中的物体可按各自的模型和规则按照计算机使用者的要求自主运动。其中,多感知性在整个系统中是最重要的,如:模拟驾驶系统中两车相撞时,计算机使用者会感觉到振动,车在抖动,能体会到两车相撞时的真实的感受。VR 系统的最大特点在于它与计算机使用者的直接交互性。在 VR



系统中，计算机使用者可以直接控制对象的各种参数。它将使人们的看法由“以计算机为主体”逐渐转变到“人是信息环境的主体”这一思想上来。

目前常见的有 3 种 VR 经历：桌面虚拟现实、投入式（沉浸方式）和增强现实性。桌面虚拟现实技术是使用个人计算机和 workstation 去产生仿真，通过专用软件，参与者可以在仿真过程中设计各种环境；投入式（沉浸方式）主要依赖于各种硬件设备；增强现实性通过使用透明的头戴式显示器，将计算机图形与参与者的即时环境中的物体叠加在一起，使得操作者对环境真实性的感受得到增强。

2.3 虚拟现实技术的组成

VR 技术是在现代科学技术诸多领域（如计算机图形学、图像处理与识别、计算机仿真技术、人机接口技术、实时分布处理技术、数据库技术、多媒体技术、多传感器技术等）的基础上发展起来的一门交叉科学技术。在硬件上，要实现比较理想的 VR 系统需要有：操纵控制设备（如 3 个以上自由度的鼠标、数据手套、数据服装）；定位设备（如机电式头盔）；跟踪设备（如电磁传感器）；显示设备（如头盔式显示器）；触觉器与运动反馈设备；耳机、立体声设备、高性能计算机系统。在软件上，VR 软件主要包括实时处理、计算机仿真、声音处理、虚拟世界的渲染、触觉如力的反馈处理、视觉等其他知觉的处理、多种数据库的管理等。

由于虚拟现实 (Virtual Reality-VR) 是一种最有效地模拟人在自然环境中视、听、动等行为的高级人机交互技术，它依托于计算机科学、数学、力学、声学、光学、机械学、生物学乃至美学和社会科学等多种学科，在计算机图形学、图像处理与模式识别、智能接口技术、人工智能技术、多传感技术、语音处理与音响技术、网络技术、并行处理技术和高性能计算机系统等信息技术的基础上迅速发展起来。它涉及到计算机图形技术、多媒体技术、传感器技术、并行实时计算技术及仿真技术等多种高科技技术领域，是这些技术更高层次的集成和渗透。被专家学者们公认为是 21 世纪可能促使社会发生巨大变化的几大技术之一。虚拟现实涉及的关键技术和研究内容主要包括：

(1) 动态环境建模技术：虚拟环境的建立是 VR 技术的核心内容，动态环



境建模技术的目的是获取实际环境的三维数据，并根据应用的需要，利用获取的三维数据建立相应的虚拟环境模型。三维数据的获取可采用 CAD 技术（有规则的环境），而更多的情况则需采用非接触式的视觉建模技术，二者有机结合可有效地提高数据获取的效率。

（2）实时三维图形生成技术：三维图形的生成技术已较成熟，而关键是如何“实时生成”，图形生成是虚拟现实技术的重要瓶颈。为了达到实时的目的，至少要保证图形的刷新频率不低于 15 帧/秒，最好高于 30 帧/秒。在不降低图形的质量和复杂程度的前提下，如何提高刷新频率将是该技术的研究内容。同时，计算机图形技术，仿真技术的提高对其发展都会产生重要影响。这里，图形生成的硬件体系结构以及在虚拟现实的真实感图形生成中用于加速的各种有效技术是关键。

（3）立体显示和传感器技术：VR 依赖于立体显示和传感器技术的发展。现有的 VR 设备还不能满足系统的需要，例如头盔式三维立体显示器有以下缺点：过重（15~20kg）、分辨率低、刷新频率慢、跟踪精度低、视场不够宽、眼睛容易疲劳等；同样，数据手套、数据衣等都有延迟大、分辨率低、使用不便等缺点；因此，有必要开发新的三维显示技术。而各种传感器则是有效进行视、听、动等高级人机交互技术的关键。比如人抓取物体时，机器应感应其动作，并对虚拟环境作相应的变化。

（4）应用系统开发工具：VR 应用的关键是寻找合适的场合和对象，即如何发挥想象力和创造性。选择适当的应用对象可大幅度地提高效率，减轻劳动强度，提高产品质量。为了达到这一目的，必须研究 VR 的开发工具。例如，VR 系统开发平台、分布式 VR 技术等，这些都直接与计算机技术，多媒体技术的快速发展密切相关。

（5）多种系统集成技术：由于 VR 系统中包含大量的感知信息和模型，因此系统的集成技术起着至关重要的作用。集成技术包括信息的同步、模型的标定、数据转换、数据管理模型、模式识别与合成等技术。



2.4 虚拟现实与仿真

一个 VR 系统没有仿真是不可能再现相同或类似于真实世界的特征的，也就是说 VR 系统需要仿真来模拟自然世界并确定虚拟世界的状态，可以说仿真是 VR 系统的核心。

另一方面，VR 系统与仿真还是有很明显的区别的。仿真的核心是一个计算，仿真并不一定需要将仿真过程表现出来，如数值仿真只需通过对模型的计算最后得到一系列的数据即可，而为了使仿真过程更直观，结果更加容易理解，并能及时了解仿真过程是否正确，在数值仿真的基础上增加文本、图形、图像、动画表现等，这种仿真称为可视化仿真。目前，有些称为多媒体仿真的也只是在可视化仿真的基础上加入声音。这些仿真系统通常都是“静态的”，即不具备交互功能，不能自由交换视角，无触觉如力的反馈信息等知觉。而有些现实中的问题并不一定都需要 VR 系统来解决，它们的二维模型就能很直观、快捷地反映问题，这时最好的方法是采用仿真来解决，而不是 VR 系统。

因此，就目前的 VR 系统与仿真的概念角度来看，虚拟现实包涵了仿真。而就目前的 VR 系统与仿真的应用角度来看，它们之间存在着交叉。

2.5 虚拟现实系统的类型

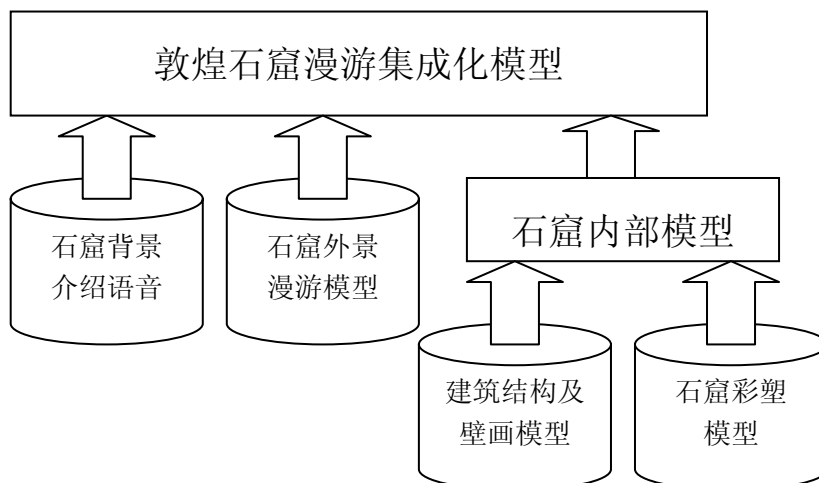
VR 系统按虚拟场景的构造方法分为基于图形的虚拟计算机图像空间、基于图像的虚拟真实图像空间和基于图形、图像的虚拟图像空间 3 大类。

基于图形的虚拟计算机图像空间是对真实世界进行抽象，然后利用计算机建立起它的三维几何模型，并在给定观察点和观察方向后，对其进行着色、消隐、光照以及投影等一系列制作，产生虚拟场景。这种系统由于需要由计算机来构造虚拟场景，而且对一些复杂场景来说往往需要构造上亿个标准三维多边形，相当费时费力，因此，这种系统对计算机的性能要求很高。此外，有时为了满足场景的实时性要求，往往采用降低场景的复杂度的方法，从而降低了虚拟场景的视觉效果。而基于图像的虚拟真实图像空间是通过数码照相机等设备，将采集到的真实世界的离散的图像或连续的视频输入计算机，经过计算机的图像快速算法处



理后，得到投影图像，然后进行无缝拼接后得到没有图像畸变的全景图像（通常为 360 度柱面全景图像），最后，通过合适的空间模型把多幅全景图像组织成为一个虚拟真实图像空间。这种系统由于不需要计算机进行三维建模，因此对计算机性能的要求相对来说不是很高，而且其显示的图像质量高，虚拟环境的处理时间与场景复杂程度没有关系，因而具有快速、简单、逼真等优点。基于图形、图像的虚拟图像空间是将现实中的虚实相结合，即虚物实化和实物虚化来构造虚拟场景，而形成的一种 VR 系统。用这种方法在微机上解决实际工程问题可取得很好的效果。

第三章 敦煌石窟漫游系统简介



3.1 系统组成简介

敦煌石窟虚拟漫游系统，需要向用户展示的内容非常广泛，包括石窟的外部环境、建筑结构、壁画、彩塑、佛龛，以及介绍石窟背景知识的语音。

- 石窟背景介绍语音

石窟背景介绍语音模型主要接受其他模块的调用，为其他模块服务，根据它们为一些指定的内容播放相应的语音。在漫游的同时，为用户提供相关的背景介绍。

- 石窟外景漫游模型

石窟外景漫游模型采用一系列的二维照片而建立一个场景漫游模型。它利用在一条或者多条漫游路线上顺序拍摄的一组照片，支持对敦煌石窟外部场景的环视或贯穿式漫游，用户可以在多条指定的路线中选择自己的漫游路线，在某些指定的关键点上可以选择前进的方向，从而实现半交互的石窟外景虚拟漫游。

- 石窟内部漫游模型

石窟内部漫游模型由石窟建筑结构及壁画模型和石窟彩塑模型两部分组成。它们分别用于表达石窟内部场景的不同部分。



3.2 石窟建筑结构和壁画模型

石窟建筑结构及壁画模型用来表达石窟的建筑结构和壁画信息，如图 1 所示。它采用传统的图形学三维模型。这种表达方式，可以很方便地对模型进行结构上的修改；可以利用绘制软件，实时或者非实时地对模型进行随意地漫游；可以通过绘制技术的改进，轻松地得到一些特殊效果，例如阴影效果、各种光照效果等。从虚拟漫游的角度说，这种表达方式具有更好的交互性。

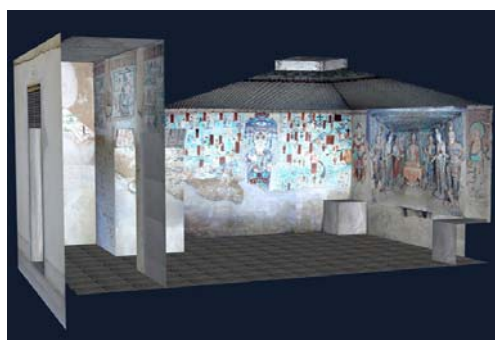


图 1 石窟建筑结构及壁画模型示意图

3.3 石窟彩塑模型

石窟彩塑模型是利用 IBR 建模思想(主要是 Texture Mapping)而建立起来的模型。由于彩塑结构复杂，传统的三维建模非常难以实现，也很难达到比较好的效果。我们采用 Texture Mapping 技术，基于一些预先生成的图像（或环境映照）来生成不同视点的彩塑画面，这样，图形绘制独立于场景复杂度，仅与所要生成的画面的分辨率有关，对计算资源的要求并不高，因而可以在普通工作站和个人计算机上实现复杂场景的实时显示。

3.4 敦煌彩塑建模与绘制

敦煌石窟中除了壁画外，另一类重要的艺术资源就是形态各异的佛像彩塑，几乎每个石窟中都有这样的塑像。它们跨越了各个年代，惟妙惟肖，所以也导致三维的实体建模十分困难。根据上面介绍分析 IBR 的技术，在敦煌石窟漫游系统中，我们采用了基于图像的绘制（IBR）技术来表达石窟彩塑。先通过 3D 建模，

或者 3D 扫描仪得到彩塑的 3D 模型，然后利用多角度拍摄得到的实物图片拼接出全景图像，将其作为纹理映射到模型表面就可以得到一个具有较强真实感的彩塑模型了。过程如图 2，图 3 所示：



图 2 (a) 彩塑头部的全景图



(b) 彩塑身体的全景图

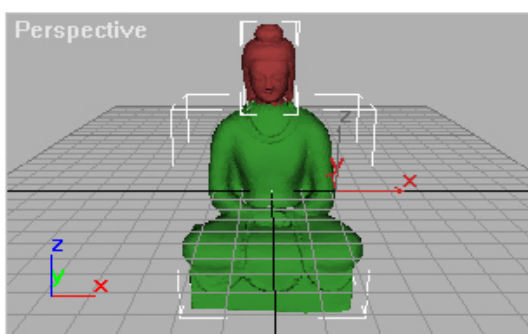


图 3 (a) 没有纹理的彩塑模型



(b) 纹理映射后的彩塑模型



第四章 彩塑阴影生成问题和克隆思想

由上一章可知，漫游系统中彩塑的生成主要是运用了 3D 建模和纹理映射。彩塑表面的情况完全由纹理决定。因此表面阴影的问题主要是利用纹理映射，通过修正被映射的纹理图像来解决。先大致介绍一下纹理映射的过程。

4.1 纹理映射 (Texture Mapping)

由于实体建模并不能完全在显出真实世界里物体的所有细节，纹理映射就成了增强一个场景可视复杂度的重要手段。可以用纹理映射的方法给计算机生成的物体图像加上纹理。纹理映射可以用来表示任何物体表面的任何材质，因此成为为计算机生成场景增强视觉真实感的一种重要方法。它基本的形式是将一幅图像（纹理）覆盖在场景中物体的表面。

因为纹理映射非常有用，图形软件界面和计算机图形硬件都将它作为一种标准的渲染方法提供给用户。因此，在一个场景中使用纹理映射只是稍微增加了一点生成这一场景程序的复杂性，有时候对于场景生成时间几乎没有影响。纹理映射的广泛的应用范围和良好的表现，使它成为一种令人期待的，可以获取大量效果的渲染技术，而这些效果以前通常都需要借助一些特殊设计的绘制硬件。

颜色纹理生成的一般方法是在一平面区域（即纹理空间）上预先定义纹理图案；然后建立物理表面的点与纹理空间的点之间的对应（即映射）。当物体表面的可见点确定之后，以纹理空间的对应点的值乘以亮度值，就可以把纹理图案附到物体表面上。可以用类似的方法给生成几何纹理，给物体表面产生凹凸不平的外观。不过这时纹理值作用在法向量上，而不是颜色亮度上。图4描述了纹理映射的一般过程。

纹理定义有连续法和离散法两种。连续法把纹理函数定义为一个二元函数，函数的定义域就是纹理空间。离散法把纹理定义在一个二维数组中，代表纹理空间中行间隔、列间隔固定的一组网络点上的纹理值。网格点之间的其他点的纹理值可通过网格点上值的插值获得。通过纹理空间与物体空间之间的坐标变换，把

纹理映射到物体表面。

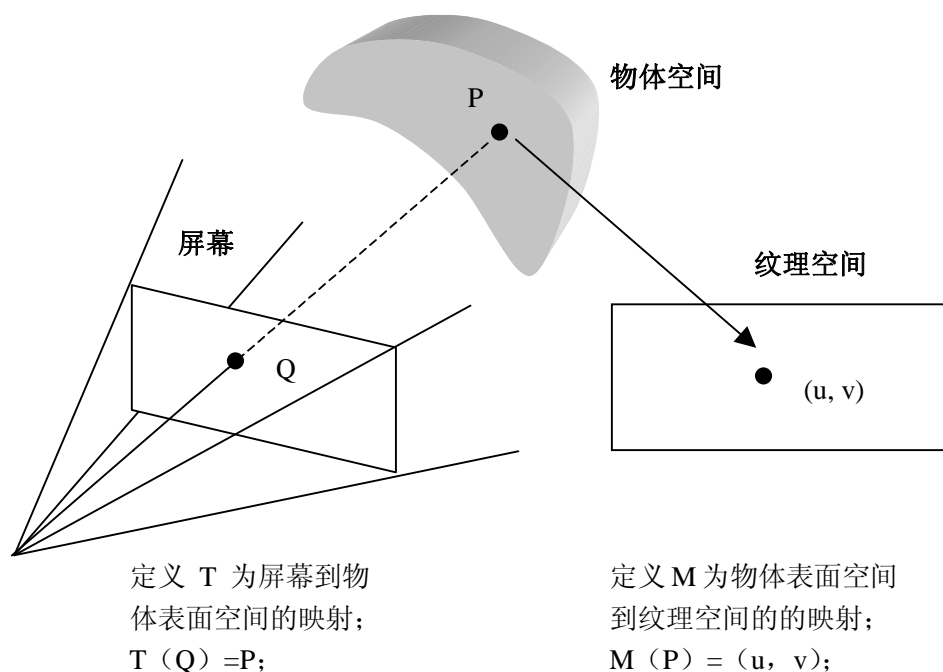


图 4: 纹理映射过程

当将一幅图像映射到一个物体时, 物体每一个像素点的颜色都根据图像中对应点做了修改。一般说来, 从图像中获取这一颜色理论上需要好几步。这一图像通常被作为一个示例数列储存, 所以一个连续图像必须首先通过取样重建。其次, 图像必须弯曲以适应被显示物体的各种变形 (可能是由于观测点变化引起的)。然后这一被弯曲的图像被过滤掉可能在最后一步导致频率混淆的高频成份: 重新取样以获得在纹理映射中要使用的颜色。

传统的纹理映射方法采用单幅纹理图像映射的方法, 被映射的物体外表面和纹理图像间具有全局对应关系。在真实世界中, 物体的表面形状任意、不规则, 如果采用传统的纹理映射方法去模拟它们, 必然会在物体外表面的某些地方 (例如表面衔接处) 产生纹理的变形, 不连续性和明显的周期性, 这在变化剧烈的边界面上更为明显。纹理映射首先要解决的就是纹理变形问题。

随着计算机技术的发展和用户对高精度图片的要求, 传统的纹理映射技术已经得到了很大的发展, 许多新技术被开发出来, 由此产生了许多纹理映射的优化

方法，以期减少纹理的变形。目前应用较广泛的是双线性过滤（Bilinear Interpolation）技术。当纹理被粘贴到屏幕所显示的一个模型时，双线性过滤技术先确定最接近像素的四个图素，然后对它们进行插补计算，其结果决定该点的颜色。然而，除非纹理映射的物体是一个平面（比如一块布），否则这些优化方法并不能完全消除变形。

另一个解决方法就是采用“Pattern_based”纹理映射，就是以一小块矩形纹理图像，作为整个物体表面纹理一个Sample，根据一定的映射模式，反复接连覆盖在物体表面，形成完整的纹理映射。矩形纹理块（Texture Sample）必须满足一定的边界约束条件，如矩形块中的左右边界纹理要互相符合，上下边界的纹理也要互相符合，以保证不会产生纹理映射的不连续性或缺口。矩形纹理块可以由现有的很多交互的二维图像绘制软件获得。采用“Pattern_based”纹理映射的一大好处是可以使用较小的纹理块，获得高精度的纹理映射图像。

Neyret 提出了一种基于三角形纹理块的“Pattern_based”纹理映射方法，极大地解决了纹理映射中的变形和不连续性、周期性的问题。Neyret 从物体表面图像中选取一组满足一定边界约束条件的三角形纹理块，作为“Texture Samples”；同时，在被映射的物体表面生成纹理映射的三角形网格，将三角形纹理块非周期地填充到纹理网格中。在一组纹理块中，选取哪个三角形纹理块，即有随机性，又必须满足边界约束。由于三角形纹理网格能够很好地模拟各种形状的物体模型，该算法产生的纹理映射，几乎不存在纹理变形；由于纹理模式是从一组不同的满足约束的三角形纹理块中随机选取出来的，不连续性和周期性的问题也得到了很好的解决，如图 5 所示。

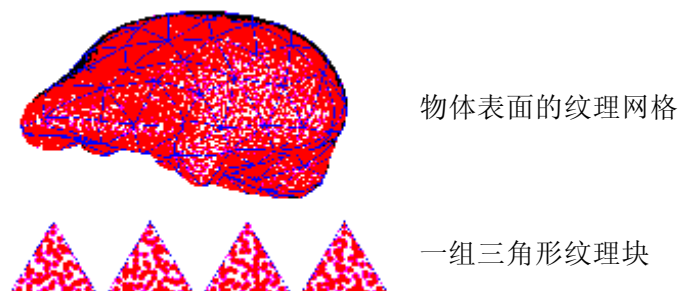


图 5：基于三角形纹理块的“Pattern_based”纹理映射

4.2 彩塑阴影问题

彩塑在不同角度灯光照射下，不仅会产生阴影区（这个根据光的直线传播原理很容易计算得到），而且会在彩塑表面产生明暗变化，如图 6 所示。



图 6 (a)灯光从正面照射



(b)灯光从彩塑左手方向照射

从图中我们可以清楚的看到灯光引起的变化。如果在灯光条件改变后，仍然使用以前正面获取的图片进行纹理映射的话，根本无法实现图 6(b)中的效果，大大影响系统的真实感。可以根据光的直线传播，对彩塑表面进行计算，但这样需要的计算量很大。当观察点随漫游进行而改变时，可能来不及进行实时渲染，因而不适合于这一虚拟漫游系统。如果仅仅是做一段动画，也可以根据观察点的运行轨迹获取相应灯光相应位置的真实图片，再次通过纹理映射实现阴影效果。但对于这样一个面向用户的交互界面，显然是无法预知漫游情况的。因此，通过获取真实图片来实现阴影不仅代价大，而且不现实。我们的目标是通过重用已有数据，快速而准确地生成彩塑阴影。

4.4 阴影克隆

光影效果必须经过一定的计算，但如果能充分利用已有的数据，可以大大简化这些计算。我们的想法是，一旦通过某种方法（在 3D Max 中渲染，或者实际拍摄）得到了源模型在各种灯光条件下的阴影图像，阴影克隆可以通过重用源模型表面在灯光条件下的变化数据对一个新的目标模型生成相似的阴影。全新彩塑的阴影可以基于已有的为许多不同模型生成的高质量阴影库。如果源模型的阴影是准确的，目标模型的阴影也会具有相近的品质。阴影克隆的另一个优点是



执行的速度，通过大计算量 3D 仿真得到的源模型阴影可以被很快地复制到目标模型上。

对于相似的彩塑，在同样的灯光情况下，表面产生的变化也是相近的，这是我们进行阴影克隆的基础。然而，目标模型和源模型间形状未必完全一致，而模型的表面形状会对阴影产生一定影响，因此阴影克隆也同样和图像变形研究有关，因为在这里，在两个不同形状物体间建立对应是一个重要的问题。

当对应关系建立以后，对于目标图像的每一点，只需在源图像中找到对应的点，根据资源库中已有的源模型在该灯光条件下的图像，获取源图像该点亮度的变化值，然后对目标图像做相应调整，就可以将阴影克隆到目标模型上了。因为这一过程不需要进行复杂的 3D 模拟，而只需要进行一些文件操作和加减运算，所以是非常迅速的，可以满足漫游系统中实时渲染的要求。阴影克隆的想法是：

- 1、在 3D Max 中对于一个给定的源模型进行各种角度的灯光渲染，截取对应的正视图图像作为资源库。
- 2、读取用户指定要生成阴影的目标图像以及相应的灯光信息。
- 3、根据用户指定的特征线段的对应，将目标图像变形 (warp)，使每一个像素点与源模型图像的相应点建立对应关系。
- 4、读取目标图像，源模型正面打光的图像以及在用户给定灯光条件下的图像，并将颜色模式改为 HIS 模式。
- 5、对应目标图像的每一个像素点，找到对应的源模型的点，分别获取该点两副图像的亮度值 (I 值)，求出差值 ΔI ，变化比例 $k = \frac{\Delta I}{I_{\text{源}}}$ 。
- 6、根据用户指定的光强，选择合适的强度因子 λ 。
- 7、将目标图像中该像素点的原亮度值 I_t 替换为 $(1 + k * \lambda)I_t$ 。

等每一个像素点亮度都改变后，阴影克隆也就完成了。

第五章 阴影克隆的具体实现

本章将结合具体的例子，对阴影克隆技术做详细的阐述。这些例子一部分是实验中的具体实现，还有一部分是在阅读相关文献过程中对一些重要技术的总结。

5.1 阴影克隆中源数据的获取。

首先，通过 3D 建模得到一个彩塑的模型，使用纹理映射后得到系统使用的源模型。在 3D Max 中模拟各种灯光从各种角度照射源模型，截取其正视图的渲染图作为资源库。图 7 显示了一些不同灯光下，源模型的图像。



图 7 通过 3D Max 渲染得到的源模型在不同灯光条件下的图像



5.2 阴影克隆中使用到的 2D 图像变形算法

因为需要根据用户输入的目标图像对阴影生成做适当修改,首先就必须在输入图像和系统内存源图像之间建立对应关系。因为无法预知用户输入的图像,通过用户指定图像中所选的一系列控制点来进行 2D 图像变形是一种方便可行的方法。由于真正被指定的只有一小部分像素,所以非控制点的位置必须由插值来决定。目前流行的变换方法有以下几种。

5.2.1 多项式卷绕 (Polynomial Warping)

一种常见的思路是,根据指定控制点的位移,求得一个函数表达式。多项式通常被作为变换表达式的一般形式,其参数的选择应能使多项式与控制点及其位移量吻合,然后就可以根据这一表达式,对目标图像的每一点进行计算。

如果多项式的项数与控制点数相同,则可以设计出能准确地映射指定的控制点的变换。通过解线性方程组,可求得多项式的系数,用矩阵求逆一般就能得到所需的结果。

然而对于彩塑这一复杂的模型,控制点的数量肯定不能太少(通常至少要 10 个),解这样的一个方程组是很困难的。如果多项式项数少于控制点数,则必须采用拟合来决定多项式参数。因为这时要求变换整体上与指定控制点最佳拟合,但每一个控制点不一定严格吻合,这样对于我们的实际应用可能会出现背景因为插值而产生我们不希望得到的变换,加上这一方法对于拟合大型图像计算量惊人,所以不予采用。

5.2.2 网孔弯曲 (Mesh Warping)

网孔弯曲算法将源图像和目标图像之间不一致的网孔特征相关联,也就是说,图像被分割成一些小区域,为实现变形,这些小区域被从一个映射到另一个,如图 8 所示。

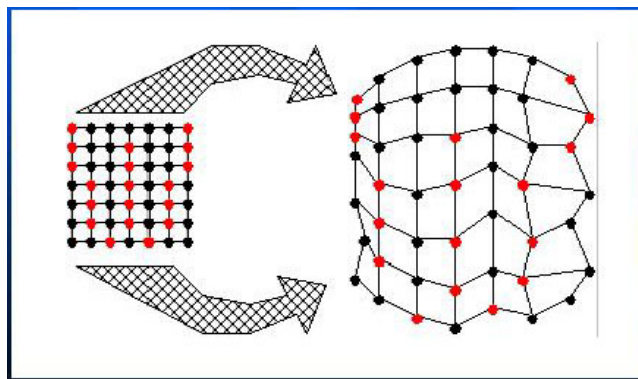


图 8 网孔变形

这个算法获取一个源图像，一个目标图像和两个 2D 坐标数组。第一个数组， S ，确定了在源图像中控制点的坐标。第二个数组， D ，确定了在目标图像中这些点相应的位置。 S 和 D 必须有相同的维度，以确保建立一个一一对应的关系。图 9 和图 10 分别显示了选取控制点前后的源图像和目标图像。



源图像



目标图像

图 9 用于变形的原始图像

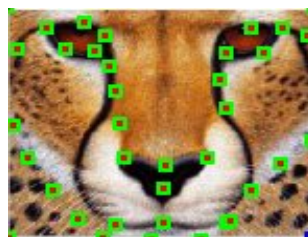


图 10 选取控制点后的图像

接下来两幅图像经过一个两周期的弯曲处理得到两个输出的中间图像 $I1$ 和 $I2$ 。第一个周期负责对每一行独立地重新采样。它将初始图像里所有点的坐标 (u, v) 映射成中间图像里的相应坐标 (x, v) ，从而将每一个输入点放置到它相应的输出列中。接着，第二个周期对中间图像中的每一列重新采样，将 $I1/I2$ 中的每一个点 (x, v) 映射到它们的最终位置 (x, y) 。储存着控制点坐标的二维数组为网孔建立了一个拓扑结构。



更具体的说，在变形过程中的每一帧都用一个中间插值的网孔 M 作为输入网孔点最终位置的集合。 M 是通过在 S 和 D 对应点之间进行线性插值计算出来的。这个“弯曲”程序在这里确实起着非常重要的作用，因为 $I1$ 和 $I2$ 都是将 M 作为目标网孔结构而被弯曲的。因此， $I1$ 是利用网孔 S 和 M 弯曲的。而 $I2$ 是利用网孔 D 和 M 弯曲的。现在，源图像和目标图像中的标记点被匹配起来。虽然任何一种样条函数都足以完成这个任务，但由于 Catmull-Rom 三次样条函数可以提供局部控制，它被用来进行对进行插值。输出如下：



虽然这一方法具有快速而且直观的优点，但它也存在以下一些明显的缺点：

- 用特征点无法保证两副图像边界的对应，这一点由输出图像可以看出。而我们得到的源图像背景是黑的，无任何变化，如果采用这一算法，会导致在目标彩塑以外的地方也产生本不该有的变化。
- 用户必须提前确定使用多少控制点来控制图像，然后获取这些点并把它们移动到正确的位置。
- 那些没有改变的点或者是动画制作者找不到相应关联特性的点仍然会被弯曲算法使用。

因为存在以上这些缺点，我们选择使用另外一种方法——基于特征线段的图像变形。

5.2.3 基于特征线段的图像变形

这一变形算法使用直线来关联源图像和目标图像的特征。它是基于那些被二维控制基元包围的有影响的域。有两种方法可以弯曲一副图像。第一种被称为正向映射。它一个像素一个像素的扫描源图像，并且把它们拷贝到目标图像的合适位置。第二种被称为反向映射。它一个像素一个像素地扫描目标图像，并且对源图像中相应地像素进行取样。反向映射最重要的特性就是目标图像中的每一个像

素都被设置了合适的值。在正向映射的情况下，目标图像中的一些像素点可能不会被描绘，从而必须进行插值。

5.2.3.1 一对直线间的转化

一对直线（一条相对于源图像被定义，另一条相对于目标图像被定义）定义了一个从一副图像到另一副的映射，如图 11 所示。

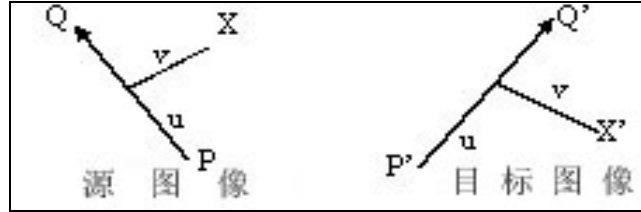


图 11 u 是直线方向， v 是法线方向

$$u = \frac{(X - P) \cdot (Q - P)}{\|Q - P\|^2}$$

$$v = \frac{(X - P) \cdot \text{Perpendicular}(Q - P)}{\|Q - P\|}$$

$$x' = P' + u \cdot (Q' - P') + \frac{v \cdot \text{Perpendicular}(Q' - P')}{\|Q' - P'\|}$$

这个算法通过旋转，平移，和（或）调整比例来对每一个像素进行变换，从而对整个图像进行变形。

5.2.3.2 多对直线间的变形

通常情况下，当采用多对直线间的变形时，图像中会有大量的特征存在。它规定了更为复杂的变形。对于多对直线，一般都是按每对直线对每一点计算位移，然后取其平均值作为最终的结果，如图 12 所示。

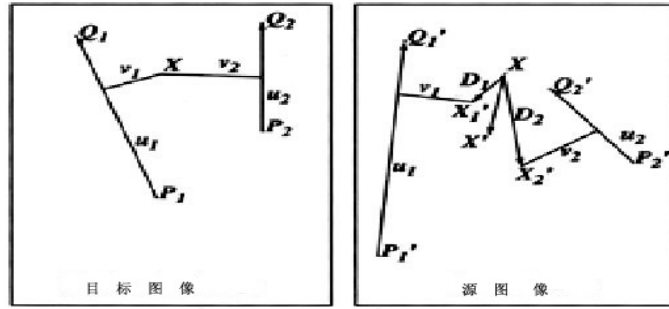


图 12 多对直线的对应

这一方法的优点是：算法用到的那些位置都是用户明确产生的。所有被指定了的点都被准确的移动到用户希望它们到的位置。而其他的点都在这些点的基础上被平滑的弯曲了，但计算量很大，计算时间正比于图像中特征线的数量和像素点的数量的乘积。

为了解决这一问题，我对这一算法做了一些改动。考虑在计算中针对每一条直线计算权值，进行带权的坐标变换，权的大小取决于 X 到直线的距离，以及线段的长度。

$$weight = \left(\frac{length^p}{(a + dist)} \right)^b$$

其中 $length$ 是线段的长度， $dist$ 是像素点到直线的距离， a ， b 和 p 是用来改变直线相对效果的常数。 P 反映了线段长度的重要性，而 b 反应了点 to 直线距离的重要性。当像素点离直线很远时，而特征线段又很短时，权值会很小。这样在计算中，我们可以通过先判断权值，然后抛弃那些距离较远，长度较短的线段，从而减少计算量，提高执行速度。

多重直线算法如下：

```

For each pixel X in the destination {
    DSUM=0
    weightsum = 0
    For each line PiQi {
        dist= shortest distance from X to PiQi
        weight = (lengthP / (a + dist))b
        if ( weight>=CONST) {

            calculate u, v based on PiQi
            calculate X'i based on u, v and P'iQ'i
            calculate displacement Di=X'i-Xi for this line
        }
    }
    DSUM = DSUM + weightsum
    X' = X + DSUM / weightsum
}
    
```



```
DSUM += Di * weight
weightsum += weight

}

}
X' = X + DSUM / weightsum
}
```

根据实际运行速度和输出质量对 a , b , p 的值不断进行修正, 最后发现当 $a=10$, $b=5$, $p=2$, $CONST=0.01$ 时, 可以在运算速度和输出质量上得到一个较好的平衡。虽然阴影克隆本身并不需要得到一个变形后的图像, 而仅仅需要得到对应的坐标, 但为了调整这些参数, 实际中也生成过一些变形图像, 如图 13 所示。



图 13 (a)源模型和作为输入的目标图像



(b)选取了特征线段之后的图像



(c)向源图像变形以后的目标图像

5.3 阴影克隆中对颜色的处理

因为我们周围的世界是丰富多彩的，如何在系统中真实鲜明的表现彩塑的颜色在灯光下的变化也是十分重要的。

三色成像系统是我们最常使用的方法。在这种情况下，可见光谱被分成三个波段——红、绿和蓝——近似于人眼的光谱量化。为了显示，红、绿和蓝色图像叠加在一起。这种叠加近似产生了真实场景在视网膜上的效果，因此人眼看起来是正常的。

虽然一幅三色数字图像可以认为是三个坐标（两个空间和一个光谱）的一个标量函数，但因为我们还需要为每个像素点寻找对应点，将其看作是一幅普通（二维）图像，且在每个像素有三个灰度值（红、绿和蓝）更为方便。

5.3.1 彩色规范

RGB 格式：有几种方法可以定量地表示彩色，如彩色数字图像中像素的颜色。最直接的方法是使用红、绿、蓝的亮度值，大小限定到一定范围，如 0 到 1。我们把这种约定称为 RGB 格式。每个像素——实际上任何可能要量化的颜色——都能用三维空间中第一象限的一个点来表示，如图 14 中的彩色立方体所示。

三色图像的灰度级直方图是 RGB 空间的点分布。

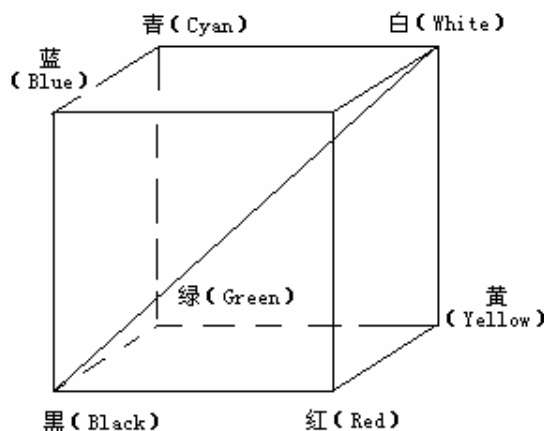


图 14 彩色立方体

在 RGB 彩色空间的原点上，任一基色均没有亮度，即原点为黑色。三基色都达到最大亮度时则表现为白色。亮度较低的等量的三种基色产生灰色的影调。所有这些点均落在彩色立方体的对角线上，该对角线被称为灰色线。彩色立方体中有三个角对应于三基色——红、绿和蓝色。剩下的三个角对应于二次色——黄色，青色（蓝绿色），和品红（紫色）。

HSI 格式：另一种有用的彩色方案称为 HSI 格式，它是 Munseu 提出的彩色系统格式，经常为艺术家所使用。这种设计反映了人观察彩色的方式，同时也有利于图像处理。

在 HSI 格式中，I 表示强度，或亮度。一般，它只是 R、G、B 三个灰度线的平均值，虽然也有人使用对不同分量有不同权值的彩色机制。强度值确定了像素的整体亮度，而不管彩色是什么。人们可以通过平均 RGB 分量将彩色图像转化为单色，这样就丢掉了彩色信息。

包含彩色信息的两个参数是色度（H）和饱和度（S），虽然也有人只用其等价的其它定义。图 15 中的色环描述了这两个参数。色度由角度表示，彩色的色度反映了该彩色最接近什么样的光谱波长。我们可不失一般性，假定 0° 的彩色是红色， 120° 的为绿色， 240° 的为蓝色。色度从 0 变到 240 度覆盖了所有可见光谱的彩色。在 240° 到 300° 之间是人眼可见的非光谱色（紫色）。

饱和度参数是色环的原点（圆心）到彩色点的半径的长度。在环的外围圆周是纯的或称饱和的颜色，其饱和度值为 1。在中心是中性（灰色）影调，即饱和



度为 0。

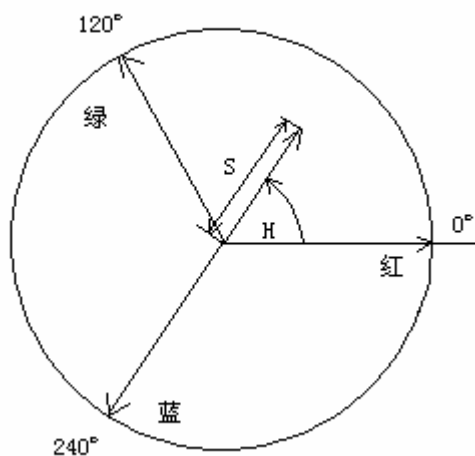


图 15 色环

饱和度的概念可描述如下：假设你有一桶纯红色的颜料，它对应的色度为 0，饱和度为 1。混入白色染料后使红色变得不那么强烈，减少了它的饱和度，但没有使它变暗。粉红色对应于饱和度值约为 0.5 左右。随着更多的白色染料加入到混合物中，红色变得越来越淡，饱和度降低，最后接近于零（白色）。相反地，如果你将黑色染料与纯红色混合，它的亮度将降低（变黑），而它的色度（红色）和饱和度（1.0）将保持不变。

总之，三个彩色坐标定义了一个柱形彩色空间（图16）。灰色影调沿着轴线以底部的黑变到顶部的白。具有最高亮度，最大饱和度的颜色位于圆柱上底面的圆周上。

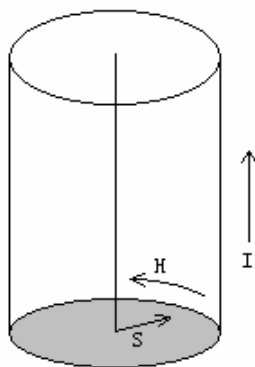


图 16 柱形彩色空间



5.3.2 彩色坐标变换

通常图像都是以 RGB 格式存储,但因为灯光主要引起彩塑表面的亮度变换,如果使用 RGB 格式,则每个像素都需要对 R、G、B 三个分量分别进行计算,计算量大。而使用 HSI 格式,对于每个像素点只需要计算亮度值 (I 值)。因此选择在阴影克隆前将读入的目标图像和源图像都转化为 HSI 格式,当克隆完成后在恢复为 RGB 格式。

5.3.2.1 RGB 到 HSI 变换

由 RGB 到 HSI 格式的转化可用以下方法实现。回忆一下在 RGB 空间中灰度线是彩色立方体的对角线,而在 HSI 空间是垂直中轴。这样,我们可以建立一个 (x, y, z) 坐标系,其中,旋转 RGB 立方体使其体对角线与 z 轴重合而其 R 轴在 xz 平面上。这个旋转由下面的公式给出:

$$\begin{aligned} x &= \frac{1}{\sqrt{6}}[2R - G - B] \\ y &= \frac{1}{\sqrt{2}}[G - B] \\ z &= \frac{1}{\sqrt{3}}[R + G + B] \end{aligned}$$

接下来,我们通过在 xy 平面中定义极坐标系转化为圆柱形坐标系,我们有

$$\rho = \sqrt{x^2 + y^2}$$

$$\phi = \text{ang}(x, y)$$

其中 $\phi = \text{ang}(x, y)$ 是从原点到点 (x, y) 的直线与 x 轴的夹角。 $\text{ang}(x, y)$ 基本是反正切,但应注意此点处在哪个象限中。

我们现在有了圆柱形坐标系,其 (ϕ, ρ, z) 对应于 (H, S, I) 。但这样定义的饱和度有两个问题:

- (1) 饱和度和强度不独立,与我们的要求不符;
- (2) 完全饱和的颜色(即其中没有多于两种的基色存在)落在 xy 平面上的一个六边形上,而不是在圆上。



补救方法是通过除以对应不同 ϕ 角的值 ρ 的最大值使 ρ 归一化。这样推导出饱和度公式为：

$$S = \frac{\rho}{\rho_{\max}} = 1 - \frac{3 \min(R, G, B)}{R + G + B} = 1 - \frac{\sqrt{3}}{I} \min(R, G, B)$$

现在完全饱和的颜色落在 xy 平面内的一个单位圆上了。

在极坐标公式中，色度由 ϕ 来定义，一个等价的方法是计算角度：

$$\theta = \cos^{-1} \left[\frac{\frac{1}{2}[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right]$$

于是色度值为

$$H = \begin{cases} \theta & G \geq B \\ 2\pi - \theta & G \leq B \end{cases}$$

5.3.2.2 HSI 到 RGB 转换

由 HSI 到 RGB 的转化公式有些不同，它取决于要转换的点落在色环哪个扇区。当 $0^\circ \leq H < 120^\circ$ 时，

$$R = \frac{1}{\sqrt{3}} \left[1 + \frac{S \cos(H)}{\cos(60^\circ - H)} \right]$$

$$B = \frac{1}{\sqrt{3}} (1 - S)$$

$$G = \sqrt{3}I - R - B$$

当 $120^\circ \leq H < 240^\circ$ 时

$$G = \frac{1}{\sqrt{3}} \left[1 + \frac{S \cos(H - 120^\circ)}{\cos(180^\circ - H)} \right]$$

$$R = \frac{1}{\sqrt{3}} (1 - S)$$

$$B = \sqrt{3}I - R - G$$



而当 $240^\circ \leq H < 360^\circ$ 时

$$B = \frac{1}{\sqrt{3}} \left[1 + \frac{S \cos(H - 240^\circ)}{\cos(300^\circ - H)} \right]$$

$$G = \frac{1}{\sqrt{3}} (1 - S)$$

$$R = \sqrt{3}I - G - B$$

HSI 的转换还有其他几种变形,从图像处理的角度来看,只要色度是一个角度,饱和度与灰度独立,转换是可逆的,则选择哪种形式不会影响处理结果。

5.4 阴影克隆中对灯光强度的处理

因为灯光强度对彩塑表面的影响接近于线性,且强度变换过于随机,无法在资源库中对每一种强度都保存对应的图像。系统的资源库只对灯光角度的变化保存对应的图像。对于灯光强度,采取在最后处理时添加强度因子 λ 来进行控制。

获取源模型两张仅仅是灯光强度不同的图片,灯光强度分别为 $I_{灯}$ 和 $I_{灯}'$ 。在模型表面选取 5 个特征点(鼻尖、眉心、两只眼角以及下颚顶点),分别计算它们与源模型图像中对应点的亮度变化,求其平均值得到 ΔI 和 $\Delta I'$ 。则

$$\lambda = \frac{\Delta I / \Delta I'}{I_{灯} / I_{灯}'}$$

根据我们的源数据,计算出 $\lambda \approx 1.2$ 。

综上所述,对于目标模型的每一点的亮度值 I ,我们进行如下替换:

$$I_t' = (1 + k * \lambda) I_t, \text{ 其中 } k = \frac{\Delta I_{源}}{I_{源}}。$$

对敦煌 057 号彩塑,根据源模型进行阴影克隆的输出结果如图 17 所示。



图 17 用敦煌 057 号彩塑作为输入的输出结果



第六章 总结与展望

6.1 总结

源模型的阴影图像是事先渲染的，为了保证较高的分辨率，耗时较长。但一旦获取了这些原始数据，对于其他的输入图像，可以通过阴影克隆，充分利用已有的原始数据。图像变形的时间，因为有权值对特征线段进行取舍，其时间比原来基于特征线段的变形算法需要的时间（ $O(\text{目标图像像素点数} \times \text{特征线段数})$ ）要少。在对图像进行颜色格式的变换后，只需要进行一次像素点参数值的改变。在对目标图像文件从头到尾进行一次扫描后就可以得到输出，时间比实际 3D 渲染要快很多。效果也基本上达到了我们预期的目的，即和源模型的阴影情况相近。

6.2 问题和扩展

当然，从实际的输出中我们也发现了不少的问题。

首先是边界的效果还不是很好。我想问题主要出在特征线段的选取上，人工操作的精确度不可能太高，在特征线的连接以及摆放位置上都可能出现误差。阴影克隆需要的人为干预应该被最小化，包括对那一小组对应关系中绝大部分的选择。希望在后续的工作中通过对几何形状，和颜色的启发性分析，使对应关系搜索至少在部分程度上是可以自动进行的。如，我们可以利用眼睛和脸部颜色的差异，知道搜索到眼睛的位置加以选取。

此外，在我们的处理中，很大一部分时间是花在了同时打开三个图像文件，并逐点读取源图像两副不同灯光条件下的亮度差上。如果能根据源模型的变化，事先拟合出一个以坐标为输入，亮度变化为输出的函数，则不仅可以省去资源图像库占用的空间，还可以减少同时打开文件的数目，运算速度也可以进一步加快。

目前我们的努力集中在尽可能准确快速地从源模型向目标模型传递同样的阴影。放置一些控制点来对整个彩塑表面和其中一部分增强和减弱一个特定的灯



光效果可能会是有用的。控制点可以直接调整比例因子的大小。同样有可能将一组阴影混合，从而对任一目标模型，得到许多灯光效果的综合。很明显，控制点提供的灵活性可以仅仅用很少的源模型阴影，就产生变化的目标阴影。



第七章 致 谢

这篇论文是本人根据在浙江大学人工智能所参加国家自然科学基金课题“多媒体与智能技术集成及艺术复原”中所积累的技术和经验加以总结而完成。在论文即将完成之际，我深深感谢我的导师鲁东明教授。在我本科的后两年时间里中，自始至终得到了鲁老师的精心指导和热情关怀。鲁老师严谨求实的治学作风，宽厚坦诚的待人之道，诲人不倦的师者风范，对教育事业满腔热情、无私奉献的工作精神，不时地感染着我，教育着我，使我终生受益。本论文的想法是鲁老师提出的。鲁老师活跃的学术思维，扎实全面的专业知识，兢兢业业、一丝不苟的工作精神和关心他人、助人为乐的思想品德，都给我留下了深刻的印象。

感谢同一个课题组的师兄石宜辉在整个过程中的指导。如果没有他的耐心指导以及和他的深入讨论，整个实验的开展就不会如此顺利。他的思想给了我全新的角度，同时他严谨治学的态度令我钦佩。

感谢混合班和计算机系的所有同学，在四年的朝夕相处中，从他们身上学到了很多有用的东西。正是他们无私的帮助，使得我能够在各方面得到长足的进步。

感谢在本科生阶段给予我关心和帮助的所有老师、同学和朋友们。祝他们前程似锦、事业有成！

最后我要把我的论文献给生我养我并寄厚望于我的父母及其他亲人。他们不但在物质上给予无私的帮助，而且在精神上鼓励我不断超越自我，迎接新的挑战。正是他们的厚望和无私奉献使我能够全身心地投入到学习之中。

陈 兆 基

2002 年 6 月于浙大求是园



参 考 文 献

- 【1】 Jun-yong Noh, and Ulrich Neumann: **Expression Cloning**, Computer Graphics and Immesive Technologies Laboratory, University of Southern California.
- 【2】 Matthew Ward: **2D Image Morphing Algorithms**, Computer Science Department, Worcester Polytechnic Institute.
- 【3】 Paul Haeberli, and Mark Segal: **Texture Mapping as a Fundamental Drawing Primitive**, Department of Electrical Engineering and Computer Science, University of California, Berkeley.
- 【4】 Thaddeus Beier, Shawn Neely, **Feature-Based Image Metamorphosis**, Proc. SIGGRAPH'92
- 【5】 David J. Kruglinski, Scot Wingo, and George Shepherd: **Inside Visual C++ 6.0**, Microsoft Press, 1999.
- 【6】 Kenneth.R.Castleman: **Digital Image Processing**, Prentice Hall, 1996.
- 【7】 周长发: **精通 Visual C++图像编程**, 电子工业出版社, 2000
- 【8】 郑正权: **3D Studio Max R3 动画超能量**, 清华大学出版社, 1999