

2015 湖南省研究生数学建模竞赛参赛承诺书

我们仔细阅读了湖南省研究生数学建模竞赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们授权湖南省研究生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号是（从组委会提供的试题中选择一项填写）：A

我们的参赛报名号为（如果组委会设置报名号的话）：201518001007

所属学校（请填写完整的全名）：国防科技大学

参赛队员（打印并签名）：1. 侯振宇

2. 戴鑫志

3. 刘李旭

指导教师或指导教师组负责人(打印并签名)：

日期：2015 年 9 月 15 日

评阅编号（由组委会评阅前进行编号）：

2015 湖南省研究生数学建模竞赛

编号专用页

评阅编号（由组委会评阅前进行编号）：

评阅记录（可供评阅时使用）：

[illegible]

湖南省首届研究生数学建模竞赛

题 目 基于光栅投影法的 3D 扫描技术研究

摘 要：

3D 扫描仪能够获得目标物体的三维信息，在生产自动化、机器人视觉、文化遗产保护、虚拟现实和计算机辅助医学等众多领域中有着广泛的用途。三维测量技术是 3D 扫描仪的核心。目前国内外实用的三维测量技术还比较少，而光栅投影技术是三维测量技术中最有发展前途的技术之一。

本文利用普通的投影仪、照相机和计算机构建了一个简单的 3D 扫描系统。投影仪投射经过格雷码编码的光栅，利用照相机采集图像，输入到计算机中通过 matlab 编程进行分析与处理，重建目标物体的三维轮廓。该系统工作原理简单，易于实现。实验结果表明，本文构建的 3D 扫描系统能够快速有效地重建目标物体的三维轮廓，在扫描速度、分辨率以及重建精度与速度等技术指标上优势明显，有利于推广与应用。

1 问题的重述与分析

1.1 问题重述

3D 扫描仪用于采集现实物体的形状、外观等数据，在 3D 打印、工业设计、动画制作等方面有广泛的用途。3D 扫描仪有多种类型，其扫描方式、测距原理、重构算法各不相同，在分辨率、速度、造价、便携性等方面差异较大。最常见的是非接触主动式扫描仪，其基本原理是先将某种光投射到物体表面，再根据采集到的反射信息重构物体表面三维信息。

商用专业 3D 扫描仪通常需要专用硬件设备，价格较高，因此可以考虑利用普通投影仪和家用数码相机、摄像机等常见设备构建一个简单的 3D 扫描系统，具体要求如下：

- ① 详细描述该 3D 扫描系统的设备构成，工作方式和重构算法。
- ② 讨论系统的分辨率、速度、扫描区域的尺寸限制等技术指标。
- ③ 构造合适的实例来说明该 3D 扫描系统的应用效果。

1.2 问题分析

3D 扫描是集光、机、电和计算机技术于一体的技术，主要用于对物体空间外形和结构色彩进行扫描，以获得物体表面的空间坐标。3D 扫描仪通过创建物体几何表面的点云，插补成物体的表面形状，在虚拟世界中创建实际物体的数字模型，即实现三维重建。

不同扫描仪的设备构成、工作方式和重构算法都不尽相同，大体分为接触式和非接触式两类。接触式 3D 扫描仪通过实际触碰物体表面的方式计算深度，精确度高，但是扫描时间较长，且可能导致对物体表面和触头本身的磨损，存在较大局限性。非接触式 3D 扫描仪根据三维视觉传感器所获得的深度图像来恢复物体的三维形状，具有效率高、自动化程度高等优点。非接触式 3D 扫描仪按照光源的照明方式又可分为被动式和主动式两类。被动式是指在自然光照明的情况下使用视觉技术获得所需的三维数据，特别适合于需要保密的军事应用场合。主动式是指使用一个专门的光源装置来提供目标物体周围的照明，通过扫描、编码或调制技术，结合立体视觉技术来获得被测物体的三维信息^[1]。

在结构光三维重建中有点结构光、线结构光、多线结构光、编码结构光等。其中，编码结构光既具有多线结构光的优点，也能解决线条匹配的问题，而且使用投影仪很容易产生编码结构光。因此，可以考虑使用编码结构光。

使用编码结构光需要考虑结构光的编码方式，为了得到较高的精度与较低的差错率，考虑选择格雷码的编码方式。

2 模型假设与符号说明

2.1 模型假设

- (1) 忽略环境光的影响。
- (2) 投影墙面和光栅均垂直水平面向上。
- (3) 照相机成像符合标准的针孔模型，忽略镜片变形等误差影响。

2.2 符号说明

- L : 原始条带的长度
 l : 条带形变后的长度
 d : 投影仪光心到投影墙面的距离
 q : 相机光心到投影墙面的距离
 m : 投影仪光心与相机光心之间的距离
 s : 放上目标物体后光栅移动的距离
 $\frac{\partial z}{\partial s}$: 系统分辨率

3 模型的建立

物体形状三维信息的快速检测技术简称三维测量技术,是利用快速的、非接触的检测手段,对于实际空间中的物体进行数字化建模,并在虚拟环境中再现的一种可视化的现代检测技术,能完成复杂形体的点、面、形的三维测量,能够实现非接触测量,由于其效率高、精度高等优点,被广泛应用于生产自动化、机器人视觉、文化遗产保护、虚拟现实和计算机辅助医学等领域。

在光学三维测量中,根据其工作原理的不同,测量方法被分成很多类,包括摄影测量法、飞行时间法、三角法、光栅投影法、成像面定位法、干涉测量法、隧道显微镜法等等。在以宏观物体为测量对象的方法中,光栅投影法是最常用的、最有发展前途的技术,这即是本文选择该方法的基本依据。

本文主要建立基于光栅投影法的 3D 扫描模型。其基本工作原理是将光栅图样投影到目标物体表面,光栅投影场由于受物体三维形状的调制而发生变形,通过对变形的光栅场进行处理,解调出代表物体深度的相位信息,再经过系统定标就可以获得物体的三维几何信息。下面有序地进行模型的建立,并作出相应的解释说明。

3.1 3D 扫描系统架构

基于 3D 扫描技术的基本要求,光栅投影法的实现过程基本都是相近的,包括设计投影模式、投影信息编码、定标还原、投影表面重构等技术。系统基本工作模式如下图所示:

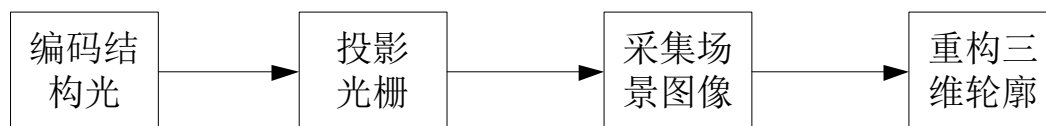


图 3-1 系统基本工作模式

采用投影仪投影方式,通过放映制作的 ppt 将等距移动的光栅条纹图案依次投影到目标物体表面,从而实现对物体表面的编码调制。

本文利用一些常见设备设计了一个简单的 3D 扫描系统，其硬件构成主要包括：普通投影仪，家用数码相机，个人计算机以及目标物体。系统硬件构成示意图如图 3-2 所示。首先，连接计算机的投影仪通过放映制作的 ppt 来产生随时间发生位移变化的平行光栅，投射到固定位置 P 上，由照相机拍摄到不同时刻的光栅图样；然后，在固定位置 P 上放上目标物体，让投影仪重新放映该 ppt，由照相机在相应的放映时刻拍摄到另一组光栅图样；最后，将照相机采集到的两组光栅图样数据导入到计算机相应的分析软件中进行处理，获得最终实验结果。需要强调的是，在整个数据采集过程中，投影仪和照相机的位置始终保持不变。

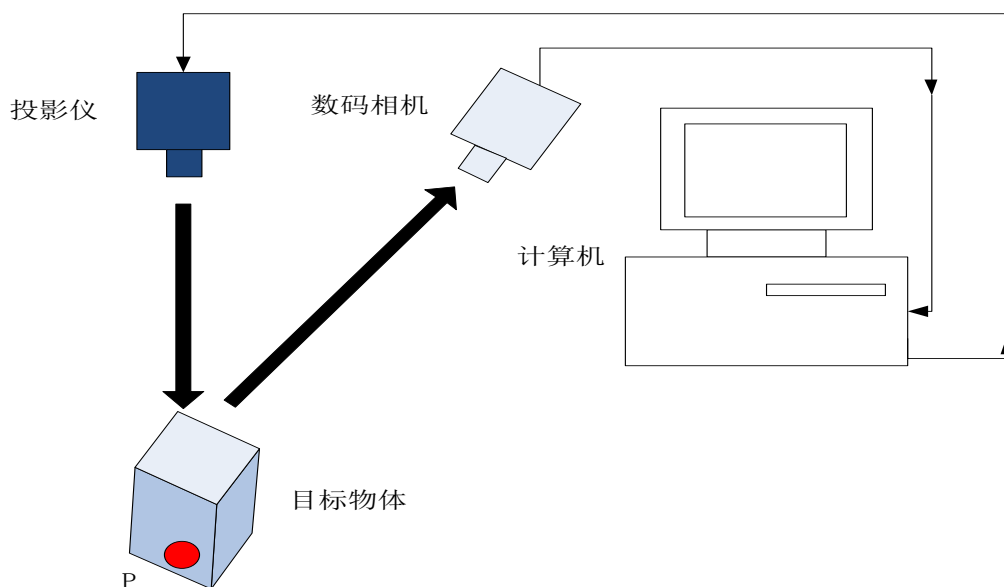


图 3-2 3D 扫描系统的硬件构成

3.2 光栅投影成像的基本原理

光栅的定义为：制有按一定要求或规律排列的刻槽或线条的透光或不透光（反射）的光学元件。

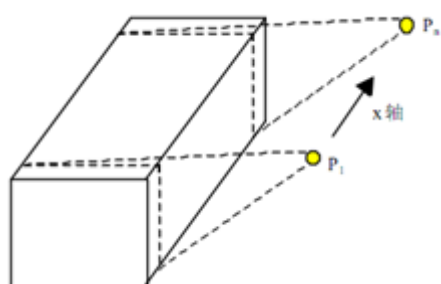


图 3-3 单缝光栅投影

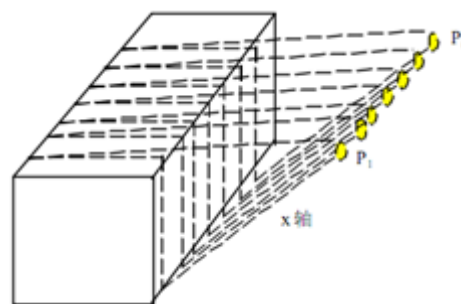


图 3-4 多缝光栅投影

把由光栅 P 产生的一束很细的光投影到空间的三维物体上，如上图所示，在物体表面上形成一道光线。这道光线好比是一个光平面切割三维物体后形成的三维物体切面的边沿，这个边沿相当于三维物体在光切面处的表面点的集合。如果光栅是一个单缝光栅，如图 3-3 所示，在与光栅投影光平面垂直的方向（x

轴) 匀速平移的光栅 (P1-Pn) 或物体, 就可在物体表面形成移动的光切口。这些移动的光切口的集合, 构成了三维物体的表面图像。当光栅是一个多缝光栅 (P1-Pn) 时, 如图 3-4 所示, 多缝光栅的一组平行光线投射到三维物体上, 在三维物体表面上形成一组平行光切口, 这组平行光切口相当于三维物体的一个表面带。当这组投影光覆盖整个物体时, 这些光切口就相当于这个物体的整个表面, 于是这些光切口的集合就构成了三维物体的表面图像^[2]。

3.3 投影图像的编码

由于光栅投影一次投射多个光条, 这些光条组成一个面投射到目标物体上, 照相机采集到的图像是多个随目标物体的表面起伏变化的光条。因此该技术需要解决的一个关键问题就是如何区分投影仪投射的多个光条, 即在照相机采集到这些形变的光条后, 投射的光条和所拍摄图像上形变的光条如何一一对应。由于在投影范围内投射光条的数量是一定的, 因此其采集密度也是一定的。

目前, 主要是通过编码来解决投射的光条和所拍摄图像上形变的光条如何一一对应的问题。通过对投影图案上的点(或者光条)进行编码, 然后对照相机采集的图像进行解码, 就能保证投射的点(或者光条)和所拍摄图像上的点(或者光条)一一对应^[3]。这种经过编码的投影图案被称为编码图案, 因此, 投影仪产生光栅技术的关键问题就是投影图案的编码设计问题。

鉴于其在减少扫描时间和提高重建精度上有较明显的优势, 本文采用格雷码的编码方法。格雷码, 又叫循环二进制码或反射二进制码, 它是一种无权码, 采用绝对编码方式。格雷码属于可靠性编码, 是一种错误最小化的编码方式, 其任意两个相邻的编码之间只有一位不同。因此, 格雷码具有良好的抗干扰性、可靠性与稳定性, 这也使得格雷码在信号领域中有广泛的应用。

格雷码是一种循环码, 其编码方式并不唯一, 本文采用一种简单的编码方式, 介绍如下。

产生 n 位格雷码的方法为: 先将 $n-1$ 位格雷码的 2^{n-1} 个编码全部列出来, 然后以相反的顺序再列一次, 形成一个 $2 \times 2^{n-1}$ 的序列, 也就是 2^n 个序列, 最后将前半部分 2^{n-1} 个编码的前面补一个 0, 后半部分 2^{n-1} 个编码的前面补一个 1, 此时就完成了 n 位格雷码的编码。最后将每个格雷码的第 x 位按照从高到低的顺序组成一个二进制序列, 并且序列中的值若为 0 则置为黑色, 为 1 则置为白色, 此时便得到了 n 幅 n 位的格雷码编码图案。

3.4 三维信息重构算法

基于光栅投影的三维测量就是将光栅图样投影到被测物体表面, 由照相机光敏面获取光栅像, 并由形变量与物体表面深度的关系来确定轮廓相对参考平面的高度信息。根据形变量与高度的关系的描述方法以及对光栅条纹的处理方法的不同, 可以将光栅投影测量大致分为两类: 结构光投影法(三角法)和相位测量法。本文采用基于结构光投影法的三维信息重构算法, 其原理和流程如下所述。

首先测量光栅在投影墙面的长 x 和宽 y 。由于我们采用的是 8 位的格雷码编码方法, 产生的条带数为 256, 每个条带的宽度为 $x/256$ 。从左往右以第 1 条黑色条带与白色条带的边界作为横坐标 x 为 0 的边, 即 y 轴。如下图所示。

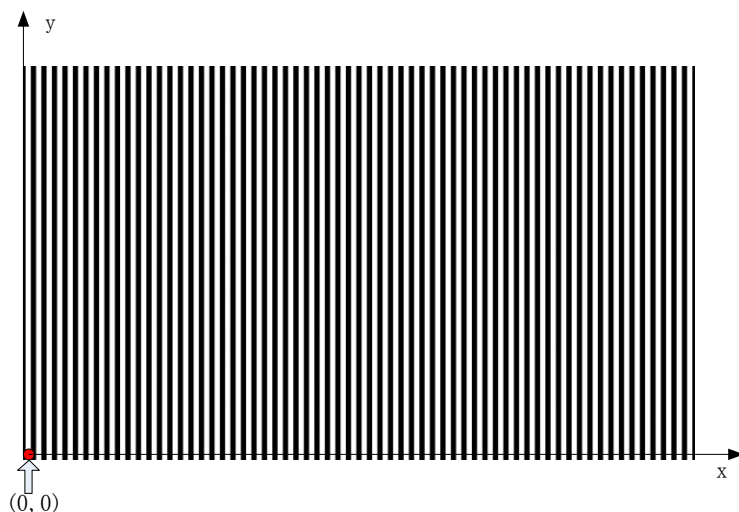


图 3-5 xOy 坐标系定位示意图

这样很容易就能得出每个条带在横轴上占有的区间。假设同一条带内的像素点具有相同的横坐标，这样根据像素点所在的条带编码就能得出其相应的横坐标值 x 。

由于相机在拍摄时相对投影墙面有一定的倾斜，因此在照片上矩形的投影区域将会变形。如下图所示。

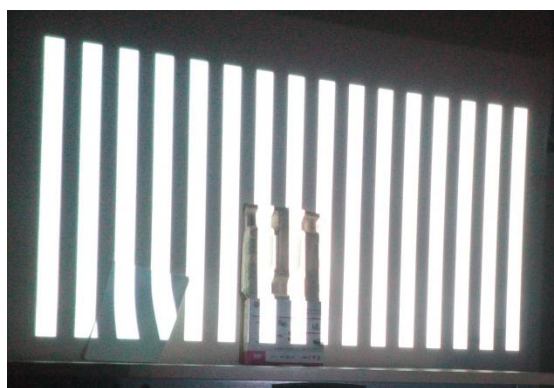


图 3-6 光栅投影图

不考虑在 y 方向上的倾斜，并且认为 y 方向上条带的形变是线性的。首先找出每个条带的最高点和最低点，那么该条带中任意点的纵坐标可以由一个简单的比例关系求出：

$$y = \frac{L}{l} \lambda \quad (3.1)$$

其中， y 表示像素点纵坐标值， L 表示原始条带的长度， l 表示条带形变后的长度， λ 表示形变后的条带中像素点到最下方点的距离。这样就得到了像素点

的纵坐标值 y ，下面再计算像素点的高度值 z 。

光源发射的光照射到参考面上，运用多个相同的光条透视投射。当目标物体轮廓高度发生变化时，像点在光敏面的位置相应地发生变化，即三维面型结构体对结构光束产生的空间调制改变了成像光束的角度，也就改变了成像光点在检测器阵列上的位置。通过系统光路的几何参数标定和对成像光点位置的确定，可以计算出相应的物体高度。基本原理图如下所示。

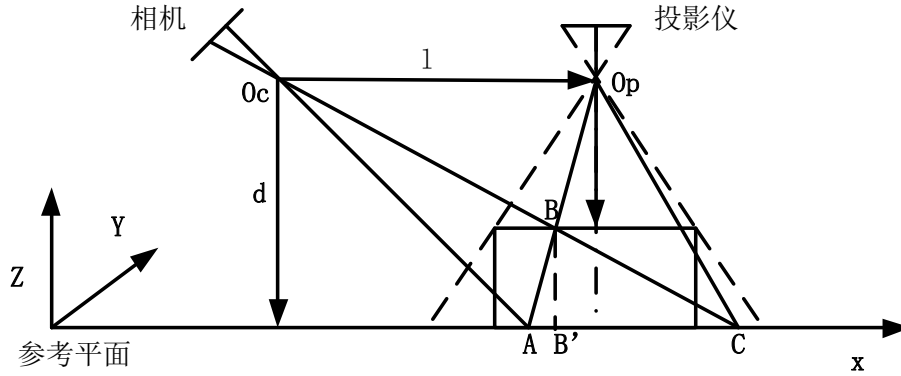


图 3-7 三角法测量原理图

如图 3-7 所示， O_p 和 O_c 分别表示投影仪和照相机的光心，两者的距离为 l ，并且两者连线与参考平面平行，参考平面即为参考坐标系中的 $X-Y$ 平面。照相机光心到参考平面的距离为 d 。当目标物体放置在参考平面时，投射到 A 点的光线被物体上的 B 点遮挡， B' 点为 B 点在参考平面上的投影点。经过 BO_c 的光线，相当于没有防止目标物体时，从 C 点反射的光线。根据相似三角形的知识，三角形 ABC 和三角形 O_pBO_c 相似，因此存在以下比例关系：

$$\frac{h}{d-h} = \frac{s}{l} \quad (3.2)$$

其中 h 表示物体的高度， s 表示 A 点和 C 点的距离。由上式能够得出物体的高度的表达式：

$$h = \frac{d \cdot s}{l + s} \quad (3.3)$$

在上式中，两光心的距离 l 和照相机光心到投影平面的距离 d 可以直接测量，而 s 需要间接测量。具体的方法是为每一道光栅编码，这样在相同光带内的像素点就有相同的编码值。对于图片上同一坐标处的像素点，如果放置目标物体前后其编码值发生变化，说明该点处的光栅发生了移动，那么利用前后两次的编码就很容易找到该点处光栅移动的距离 s 。通过上面的计算，最终得到物体的高度 h ，即在参考坐标系下的 z 坐标。

至此就得到了物体上任意投影点的三维坐标值。

3.5 技术指标分析

3.5.1 分辨率

假设投影仪光心到投影墙面之间的距离为 d ，投影仪光心到相机光心的距离为 m ，放上物体后光栅移动的距离为 s ，如图 3-7 所示，则物体上像素点的深度即 z 坐标为：

$$z = \frac{d \cdot s}{m + s} \quad (3.4)$$

由于实际得到的 s 是一个离散量，因此对应的 z 值会产生跳变，这就反映了 z 的测量精度，即分辨率。由式(5.3)，对 s 求偏导，即得到分辨率：

$$\frac{\partial z}{\partial s} = \frac{d \cdot m}{(m + s)^2} \quad (3.5)$$

从上式可以看出分辨率 $\frac{\partial z}{\partial s}$ 是关于 d 的一次函数，因此 d 越大则分辨率越小，性能越好。

由式(3.4)，两边同时对 m 求二阶偏导可得：

$$\frac{\partial^2 z}{\partial s \partial m} = \frac{d \cdot (s - m)}{(m + s)^3} \quad (3.6)$$

由上式可知，当 $m = s$ 时，取得唯一极大值，故系统的分辨率随着距离 m 的增大而增大。

3.5.2 扫描区域的尺寸限制

由于系统采用结构光重建的方法，因而能够一次性重建物体在照相机视野中的一个面（除去阴影区域）。如图 3-8 所示（俯视图）。

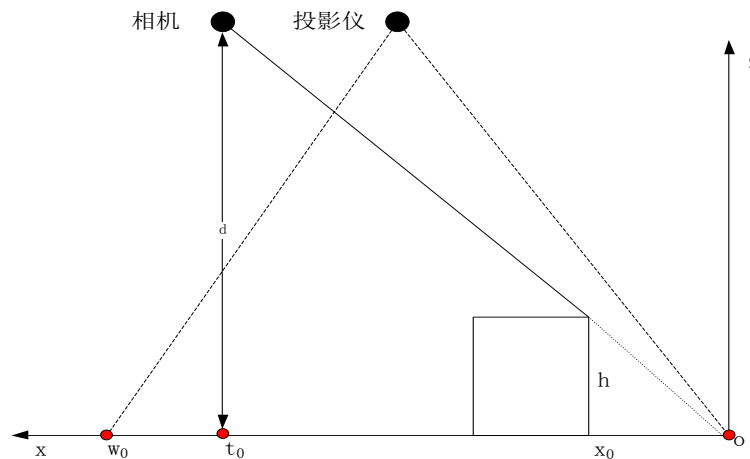


图 3-8 扫描区域示意图

假设投影仪的投影区域为原点到 w_0 ，相机到投影墙面的距离为 q ，相机在墙

面的投影点到光栅原点的距离为 t_0 ，目标物体上一点的横坐标为 x_0 ，且该点在照片上的像点恰好位于光栅边界，那么可以得到物体高度 h 和位置 x_0 的一个关系式：

$$h = \frac{x_0}{t_0} q \quad (3.7)$$

由上式可知，对于在相机右侧横坐标为 x_0 的某点，如果它的高度超过 h ，则它的右侧部分将成为盲区，无法进行三维重构。同样地，也可以计算出目标物体位于左侧时高度 h 的限制，这与相机在墙面上的投影位置与光栅的左边界有关，其原理与上文所述原理相似，此处不再赘述。

4 模型的求解

4.1 实验场景布置

按照 3.1 节的设计，布置了实验场景，如下图所示：

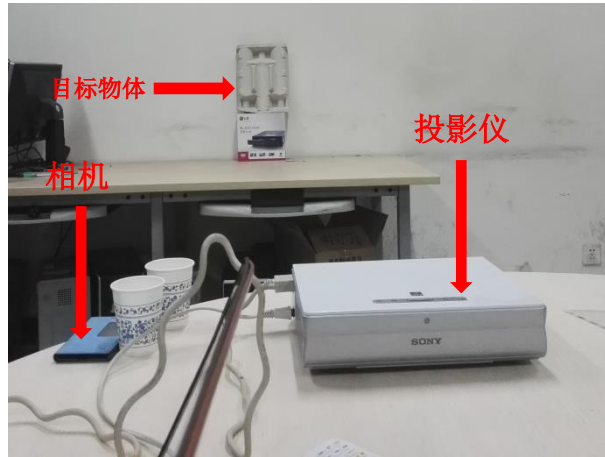


图 4-1 实验场景图



图 4-2 目标物体



图 4-3 目标物体光栅投影

模型求解的系统框图如图 4-4 所示：

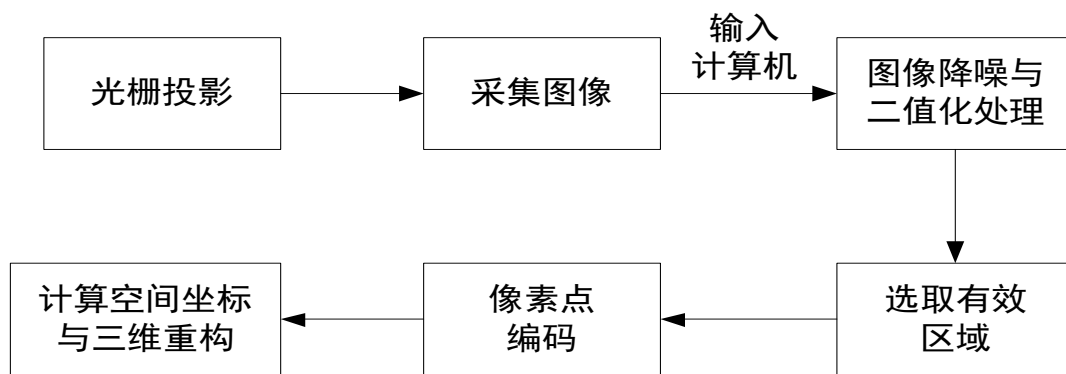


图 4-4 系统框图

4.2 投影图像采集

为了方便，使用手机代替照相机进行投影图像的采集，分别采集了无目标物体时和有目标物体时的光栅投影图像各 8 幅，下面是其中的 4 幅：

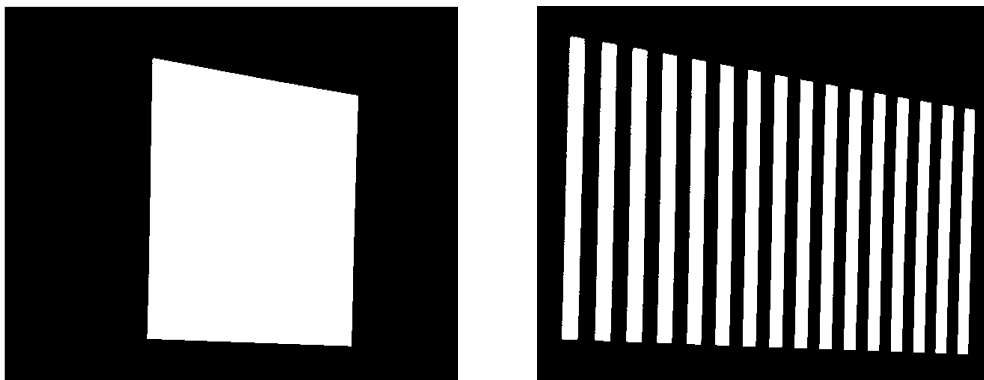


图 4-5 无目标物体时的光栅投影图像

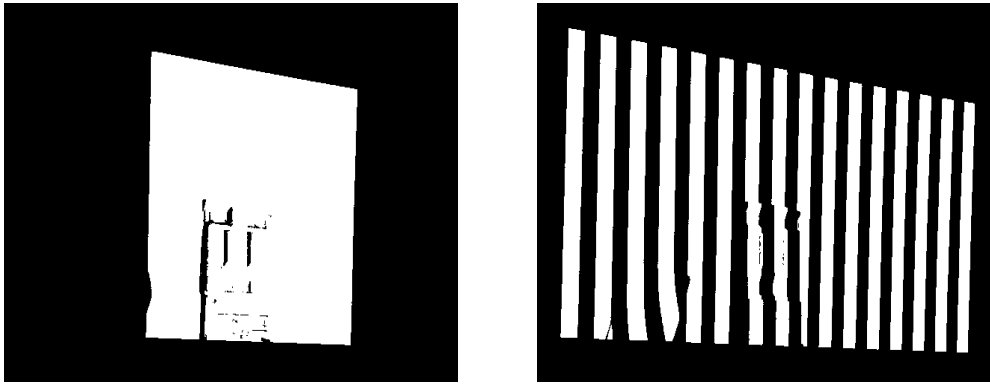
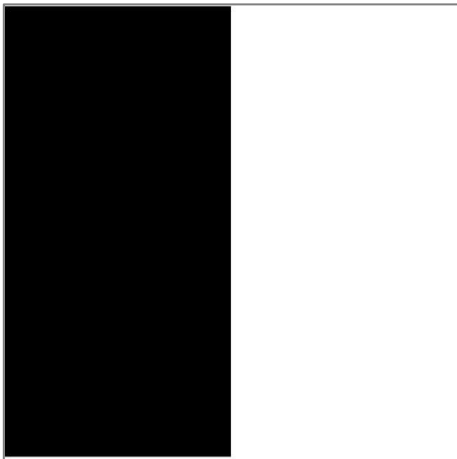


图 4-6 有目标物体时的光栅投影图像

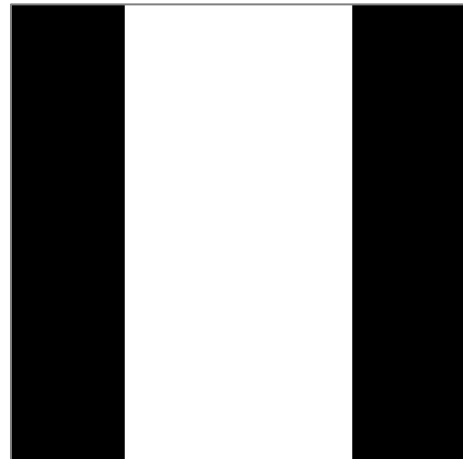
由图对比可知，当多条条纹照射到目标物体上时，光栅投影图案发生了形变，这些形变即包含了物体表面的轮廓信息。

4.3 像素点编码

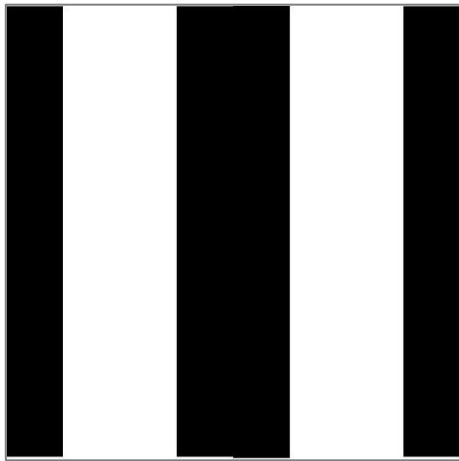
选取 8 位格雷码对像素点进行编码，得到 8 幅编码图案，下面是其中的前 4 幅：



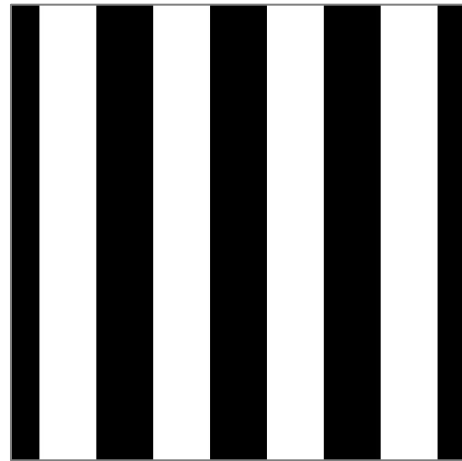
第 1 幅格雷码编码图案



第 2 幅格雷码编码图案



第3幅格雷码编码图案



第4幅格雷码编码图案

图 4-7 8 位格雷码编码的前 4 幅编码图案

4.4 三维信息重构

在 MATLAB 2014 的平台上，依据 3.4 节中设计的重构算法，通过编程实现目标物体三维信息的重构，实验结果如下：

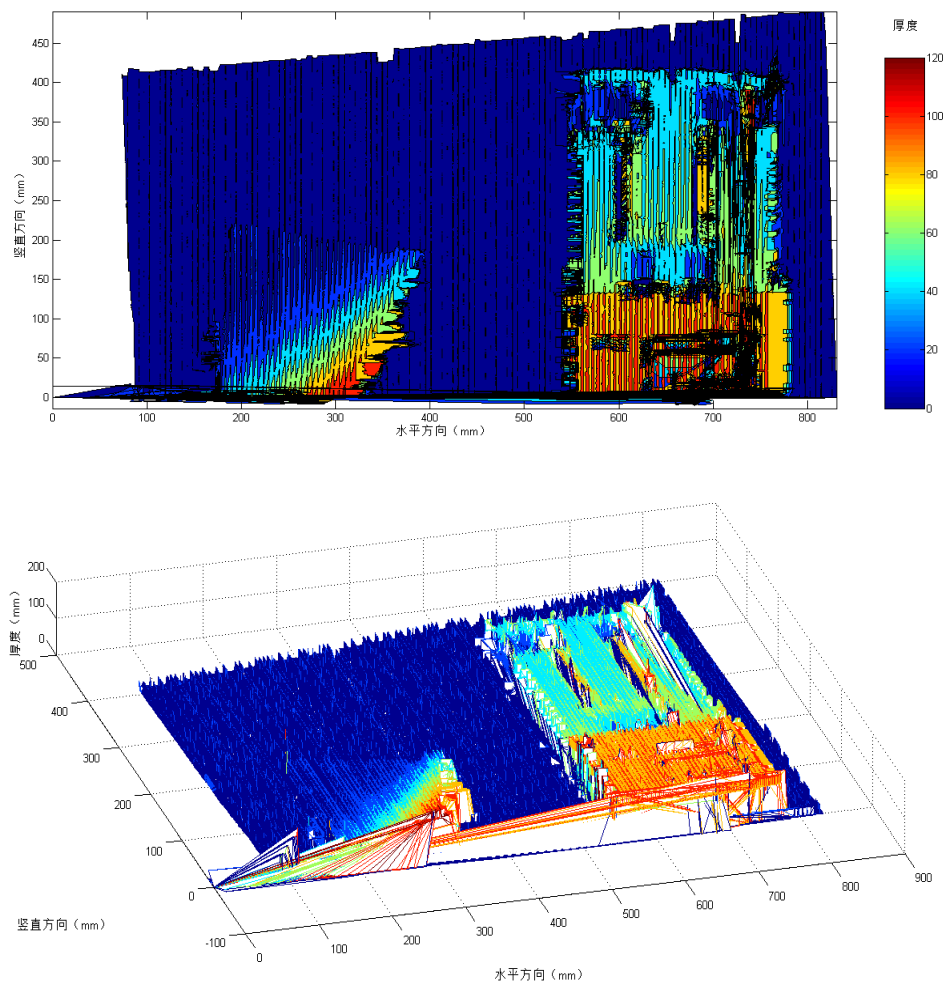


图 4-8 三维信息重构结果图

由结果图可知，目标物体的三维重构效果良好，能够较好地反映出物体的三维视觉信息。

4.5 技术指标计算

通过测量可知，本系统中，投影仪光心到投影墙面之间的距离 $d = 2240mm$ ，投影仪光心与相机光心之间的距离 $m = 521mm$ ，放上物体之后光栅移动的距离 s 的最小值为一个光栅的宽度，即 $s = 5.7mm$ ，代入式(3.5)可求得系统的分辨率 $\frac{\partial z}{\partial s} \approx 4mm$ 。

标定好投影仪和相机的位置后，系统扫描的时间主要耗费在拍摄不同编码光栅的图像中，我们采用了 8 位格雷码，因此一共需要拍摄 16 张照片，大约用时 1min。计算机成像的时间则很短。

由此可见，该系统在分辨率和扫描时间上性能优越。

6 模型的优缺点与改进方向

6.1 优点

- 系统设备构成简单，成本低，易于实现。
- 投影仪和相机的标定方法相对简便。本系统标定时只需要相机光心和投影仪光心的连线平行于投影面，以及相片的纵轴与投影面平行。以上两点容易实现。
- 三维信息重构速度快。系统采用面结构光，能够一次生成一个目标物体的一个侧面，优于点光源和线结构光。
- 系统分辨率性能较好。理论上系统的分辨率可低至 4mm。

6.2 缺点

- 图像采集时间较长。采用 n 位格雷码编码时，需要采集 $2n$ 张图片，而为了获取较高的精度，就需要 n 的取值较大。
- 三维信息重构的区域受限。由于只使用了一台相机，因此成像区域仅限于目标物体的一个侧面。
- 只能重建静止的物体。由于采用的是格雷码编码的方式，因此重建三维轮廓时目标物体不能运动。

6.3 改进方向

- 🌈 构建多目扫描系统。使用多个相机从不同的方向对物体进行拍照，然后通过图像拼接生成更加完整的三维图像。
- 🌈 采用新的编码方式。目前除了格雷码之外还有一些更优的编码方式，可以进一步减小图像采集时间，同时提高编码精度。
- 🌈 图像去噪。系统的去噪模块不够完善，实验中噪声点对于目标物体三维重构的影响较大，通过完善去噪模块来改善三维重构图像的质量。

参考文献

[1] 侯风芝, 格雷码结合相移的编码结构光三维重建技术研究, 西安: 西安电子科技大学出版社, 2011。

[2] 李东军, 光栅投影在三维测量技术中的应用, 郑州: 河南大学出版社, 2013。

[3] 于晓洋, 结构光时间编码技术进展, 哈尔滨理工大学学报, 2010。

[4] Hesam, DIY 3D scanner based on structured light and stereo vision in Python language, <http://www.instructables.com/id/DIY-3D-scanner-based-on-structured-light>, 2015.9.12。

[5] Frank, 自制低成本 3D 激光扫描测距仪, <http://www.sigvc.org/bbs/thread-2182-1-1.html>, 2015.9.12。

附录

所有程序均使用 MATLAB 2014 软件编写及运行。

1. 图片读取程序

```
clc;
```

```
close all;
```

```
clear;
```

```
[B,lev]=tupian_erzhihua('D:\工作文件\硕士研究生\数学建模\论文\实验图片\使用\左\tl9.jpg');
```

```
C=tiqu_erzhihua(B);
```

```
save tl9 C lev
```

```
[B,lev]=tupian_erzhihua('D:\工作文件\硕士研究生\数学建模\论文\实验图片\使用\左\tl8.jpg');
```

```
C=tiqu_erzhihua(B);
```

```
save tl8 C lev
```

```
[B,lev]=tupian_erzhihua('D:\工作文件\硕士研究生\数学建模\论文\实验图片\使用\左\tl7.jpg');
```

```
C=tiqu_erzhihua(B);
```

```
save tl7 C lev
```

```
[B,lev]=tupian_erzhihua('D:\工作文件\硕士研究生\数学建模\论文\实验图片\使用\左\tl6.jpg');
```

```
C=tiqu_erzhihua(B);
```

```
save tl6 C lev
```

```
[B,lev]=tupian_erzhihua('D:\工作文件\硕士研究生\数学建模\论文\实验图片\使用\左\tl5.jpg');
```

```
C=tiqu_erzhihua(B);
```

```
save tl5 C lev
```

```
[B,lev]=tupian_erzhihua('D:\工作文件\硕士研究生\数学建模\论文\实验图片\使用\左\tl4.jpg');
```

```
C=tiqu_erzhihua(B);
```

```
save tl4 C lev
```

```
[B,lev]=tupian_erzhihua('D:\工作文件\硕士研究生\数学建模\论文\实验图片\使用\左\tl3.jpg');
```

```
C=tiqu_erzhihua(B);
```

```
save tl3 C lev
```

```
[B,lev]=tupian_erzhihua('D:\工作文件\硕士研究生\数学建模\论文\实验图片\使用\
左\tl2.jpg');
C=tiqu_erzhihua(B);
save tl2 C lev
```

```
[B,lev]=tupian_erzhihua('D:\工作文件\硕士研究生\数学建模\论文\实验图片\使用\
左\tl1.jpg');
C=tiqu_erzhihua(B);
save tl1 C lev
```

2. 图片二值化程序

```
function [B,lev]=tupian_erzhihua(filename)
imdata = imread(filename);
%lev = graythresh(imdata);
lev=0.95;
bwimg = im2bw(imdata,lev);
[m,n]=size(bwimg);
```

```
%% %%%%%%%%%%%%%%%降噪
erzhihua_jiangzao(bwimg)
%% %%%%%%%%%%%%%%%降噪
figure;
imshow(bwimg);
title('二值化');
```

```
%% 将二值化照片由结构体保存成数值
B=double(bwimg);
end
```

3. 二值化提取程序

```
function [C]=tiqu_erzhihua(B);
%% %%%%%%%%%%%%%%%提取需要的部分
C=B(50:1400,300:2130);
figure;
imshow(C);
title('二值化_提取后');
end
```

4. 图像降噪处理程序

```
function erzihua_jiangzao(bwimg)
[m,n]=size(bwimg);
for i=1:m
    for j=2:n-2
```

```

        if
        (bwimg(i,j)==1&&bwimg(i,j+1)==0&&bwimg(i,j-1)==0)||(bwimg(i,j)==1&&bwimg(
i,j+1)==1&&bwimg(i,j-1)==0&&bwimg(i,j+2)==0)
            bwimg(i,j)=0;
            bwimg(i,j+1)=0;
        end
    end
end

```

```

for j=1:n
    for i=1:m-5
        if bwimg(i,j)==0&&bwimg(i+1,j)==1&&bwimg(i+5,j)==0
            bwimg(i:i+5,j)=0;
        end
    end
end
end

```

5. 获得灰度图程序

```

clc
clear
close all

load tl8.mat

graycode=zeros(size(C));
graycode=C*2^7+graycode;
clear C;

load tl7.mat
graycode=C*2^6+graycode;
clear C;

load tl6.mat
graycode=C*2^5+graycode;
clear C;

load tl5.mat
graycode=C*2^4+graycode;
clear C;

load tl4.mat
graycode=C*2^3+graycode;

```

```

clear C;

load tl3.mat
graycode=C*2^2+graycode;
clear C;

load tl2.mat
graycode=C*2^1+graycode;
clear C;

load tl1.mat
graycode=C+graycode;
clear C;

imagesc(graycode)
save target graycode

```

6. 获得 X 坐标程序

```

clc
clear

load X_label.mat

%% 获得 X 坐标
for i=1:m
    for j=1:1300
        if X_label(i,j)>1000
            X_label(i,j)=0;
        end
    end
end

for i=1:m
    for j=1:820
        if X_label(i,j)>700
            X_label(i,j)=0;
        end
    end
end

j=1;
for i=1:m
    j=1;

```

```

while X_label(i,j)==0
    j=j+1;
    if j>n
        j=n;
        break;
    end
end
last=X_label(i,j);index_last=j;
while j<n
    while X_label(i,j+1)==X_label(i,j)
        j=j+1;
        if j==n
            break;
        end
    end
    end

    if j==n
        break;
    end

    if X_label(i,j+1)>last
        temp=1;
        j=j+1;
        next=X_label(i,j);index_next=j;

X_label(i,index_last:index_next)=linspace(last,next,index_next-index_last+1);
        last=next;index_last=index_next;
    end
    j=j+1;
end
end

```

7. 三位重构程序

```
clc
```

```
clear
```

```
close all
```

```
W=1464;
```

```
H=824;
```

```
l=2240;
```

```
d=521;
```

```
code_num=256;
```

```
A=linspace(0,W,code_num);
```

```

%% 标定 X 坐标
X=zeros(1,code_num);

load standard.mat
num=0;
standard_line=graycode;
clear graycode;
[m,n]=size(standard_line);
for i=1:n-1
    if standard_line(800,i+1)~=standard_line(800,i)
        num=num+1;
        X(standard_line(800,i+1)+1)=A(num);
    end
end
X_label=zeros(m,n);

%% 计算 z 坐标
load target.mat
target_line=graycode;
s=zeros(m,n);
for i=1:m
    for j=1:n
        if target_line(i,j)~=0
            X_label(i,j)=X(target_line(i,j)+1);
            s(i,j)=-X(target_line(i,j)+1)+X(standard_line(i,j)+1);
        end
    end
end

figure;
imagesc(X_label);
title('X_label');

Z_label=s*l./(d+s);
%% %%%%%%%%%%%%%%%修改 z
%%%%%%%%%%%%%%
for i=1:m
    for j=1:n
        if Z_label(i,j)<0||Z_label(i,j)>700
            Z_label(i,j)=0;
        end
    end
end

```

```

end
figure
imagesc(Z_label);
title('Z_label');

%% 计算 y 坐标
Y=zeros(code_num,2);

% 找出每个条带最上方点
for k=1:code_num
    i=1;j=1;
    while i<=m
        if standard_line(i,j)==k-1
            Y(k,1)=i;
            break;
        else
            j=j+1;
            if j>n
                j=1;i=i+1;
            end
        end
    end
end
end

% 找出每个条带最下方点
for k=1:code_num
    i=m;j=1;
    while i>=1
        if standard_line(i,j)==k-1
            Y(k,2)=i;
            break;
        else
            j=j+1;
            if j>n
                j=1;i=i-1;
            end
        end
    end
end
end

% 获得 y 坐标
Y_label=zeros(m,n);
for i=1:m

```

```

        for j=1:n
            if(target_line(i,j)~=0)
                line=target_line(i,j)+1;
                Y_label(i,j)=H*(Y(line,2)-i)/(Y(line,2)-Y(line,1));
            end
        end
    end
end
figure;
imagesc(Y_label);
title('Y_label');

%% 获得 X 坐标

%% %%%%%%%%%%%%%%%修改 x
for i=1:m
    for j=1:1300
        if X_label(i,j)>1000
            X_label(i,j)=0;
        end
    end
end

for i=1:m
    for j=1:820
        if X_label(i,j)>700
            X_label(i,j)=0;
        end
    end
end

%% x 插值
j=1;
for i=1:m
    j=1;
    while X_label(i,j)==0
        j=j+1;
        if j>n
            j=n;
            break;
        end
    end
    last=X_label(i,j);index_last=j;
end

```

```

while j<n
    while X_label(i,j+1)==X_label(i,j)
        j=j+1;
        if j==n
            break;
        end
    end

    if j==n
        break;
    end

    if X_label(i,j+1)>last
        temp=1;
        j=j+1;
        next=X_label(i,j);index_next=j;

X_label(i,index_last:index_next)=linspace(last,next,index_next-index_last+1);
        last=next;index_last=index_next;
    end
    j=j+1;
end
end

%% 处理 Z
%% %%%%%%%%%%%%%%%修改 z
for i=1:m
    for j=1:n
        if Z_label(i,j)>150
            Z_label(i,j)=0;
        end
    end
end

end
%% 抽取需要的部分
X_need=X_label(650:1230,200:1120);
Y_need=Y_label(650:1230,200:1120);
Z_need=Z_label(650:1230,200:1120);
%% 画图
figure;
mesh(X_need,Y_need,Z_need);
title('mesh');

figure;
surf(X_need,Y_need,Z_need);

```

```
title('surf');
```

```
figure;  
contourf(X_need,Y_need,Z_need);  
title('contourf');
```

```
figure;  
contour(X_need,Y_need,Z_need);  
title('contour');
```