
可变焦距 3D 扫描仪及数据处理系统 的研究与设计

学校：广州市执信中学

作者：练孙鸿

指导老师：梁志成

摘要：本文描述了笔者对可变焦 3D 扫描的研究，包括物理设计与数学实现，已完成的部分的物理和编程实现。其主要流程是，在较暗环境中，用激光线（结构光）进行照射与拍照后要对数据进行处理。主要处理流程包括：（1）对图片的采样计算获取 2D 点；（2）从 2D 信息中生成 3D 顶点；（3）标识点的识别分析并计算出当前照片相机的部分刚体变换信息；（4）坐标变换；（5）网格重构（6）曲面细分。并且本文运用了光学规律与定律、传统几何、线性代数、计算机三维图形学理论等来设计了一个从现实世界的物体转化为 3D 数字模型文件的系统，在理论上已经实现了一个手持、可变焦的 3D 扫描仪。物理实现上主要的传感器仅为重力传感器。

关键词：3d 扫描、变焦、坐标变换、网格重构

Abstract

This passage which mainly describes a research to 3D scanning with focal-changeable camera ,includes physics design and mathematical implementations, partially-completed physics and programming implementations. The main procedure of our scanning ,is to illuminate the target object with laser line(structured light) and take photos in a dark environment, then process the data which was obtained recently. The main procedure of data processing is :

- 1.Sampling from pictures and compute 2D points
- 2.Generate 3D points from 2D information
- 3.Analyze the identification points in order to compute some of the rigid transformations information of the camera in each picture. (*)
- 4.Coordinate Transformation
- 5.Mesh reconstruction
- 6.Surface Subdivision

This passage use optical laws ,traditional geometry, linear algebra, theory of computer 3D graphics to design a system that is able to transform the real object into 3D digital model file. Theoretically, the implementation of handheld ,focal-changeable 3D scanner was attained. The only sensor needed is gravity sensor.

Key word: 3d modeling, Variable focal length, Coordinate transformation, Mesh reconstruction

目录

1.研究背景.....	1
2.前期分析.....	1
2.1 - 3D 扫描的概述.....	1
2.2 -本文简介	2
2.3 实验环境.....	3
3. 设计、算法推导、算法分析.....	4
3.1 实物结构设计与 GetDepthFromOffset 函数.....	4
3.2 图片采样.....	9
3.2.1 原创方法的分析与改进.....	9
3.2.3 成像的 Y 坐标区间重叠.....	11
3.2.4 简化计算.....	12
3.2.5 精度.....	13
3.3 生成三维顶点 - 局部坐标	15
3.3.1 XYZ 分量分析.....	错误！未定义书签。
3.3.2 数学模型 1——长方体模型.....	错误！未定义书签。
3.3.3 数学模型 2——圆柱体模型	错误！未定义书签。
3.3.4 数学模型 3——散乱点云.....	错误！未定义书签。
3.4 世界变换.....	17
3.4.1 长方体模型的转换矩阵.....	错误！未定义书签。
3.4.2 圆柱体模型的转换矩阵.....	错误！未定义书签。
3.4.3 散乱点云的转换矩阵.....	错误！未定义书签。
3.5 标识点.....	22
3.5.1 标识点的作用.....	22
3.5.2 标识线段.....	24
3.5.3 去除翻滚角影响时（EulerZ）空间点的成像.....	25
3.5.4 去除仰角影响时（EulerX）标识点的局部 3D 坐标	27
3.5.5 去除了仰角、翻滚角之后计算世界坐标、水平角	35
3.6 网格重构.....	39
3.6.1 - 国内外的研究	39
3.6.2 - 本文的结构点云的网格重构.....	错误！未定义书签。
3.6.3 - 曲率变化散乱点云的重构.....	错误！未定义书签。
3.7 曲面细分.....	错误！未定义书签。
3.7.1 曲面细分的简介.....	错误！未定义书签。
3.7.2 3 细分模式	错误！未定义书签。
3.7.3 Loop 细分模式.....	错误！未定义书签。
3.7.4 本文提出的细分模式.....	错误！未定义书签。
4. 数学模型的验证.....	42
4.1 - 3DMAX 模拟图	42
4.2 - 理论与实际模型的匹配.....	42
4.2.1 - 工作开始前测量视角.....	错误！未定义书签。
4.2.2 - 即时计算视角	错误！未定义书签。

4.3 - 圆柱形模型的原型机截图（变焦、无手持、固定）	44
4.4 - 仍待解决的问题	44
5 - 结论	45
参考文献.....	47

1.研究背景

随着计算机的不断发展,数字媒体的发展经历了一维的声音,二维的图像和视频。近年来开始有了三维几何模型,这是一种新兴的数字媒体。而用于描述几何体的图元一般是三角形,因为三角形可以近似描述绝大部分的物体拓补结构,而且对光滑度等的要求不高。很多地方开始对快速的 3D 建模有需求,如 3D 实时游戏里面的物体与人体建模,虚拟现实,考古物体数字化重建,电子商务,3D 打印等。但是由于复杂性、时间成本等原因,用 3DMAX 等软件进行正向的构建是有很大的难度的。如果可以免除对照着现实物体进行建模的繁琐而且耗时的程序,那么将可以大大节省成本。而在国内 3D 扫描仪大都引进外国设备,价格昂贵,但其实真正的成本并不算太高,物理成本最高的也只是单反相机,当然降低成本可以使用工业摄像头,真正昂贵的或许是算法,我们打算自己研究出一套系统。

外国在理论和实物上已有比较成熟的发展。其中已发展出了民用的扫描仪,美国 3D 打印机厂商 MakerBot CEO 布瑞·佩蒂斯(Bre Pettis)在 SXSW 互动大会上展示了 3D 扫描仪的原型品 Digitizer。此扫描仪的价格在 1800 美元左右,但这个转盘式的扫描仪,虽然没有手持扫描仪的方便,而且表面信息容易获取不全,但是和国内四五位数的价格比起有着较大的优势。而且相对与转盘式等扫描仪,手持扫描仪因为其更为一般化的算法和技术,价格也是异常昂贵。

研究此课题有利于普及 3D 扫描仪的技术和产品,促进低成本家用扫描仪的普及,促进国内相关研究的发展。

2.前期分析

2.1 - 3D 扫描的概述

最早出现的是接触式测量方法,代表是三维坐标测量机,虽然精度达到微米量级(0.5mm),但是由于体积巨大、造价高以及不能测量柔软的物体等缺点,使其应用领域受到限制。

于是出现了非接触式测量方法,主要分两类。

一类是被动方式,就是不需要特定的光源,完全依靠物体所处的自然光条件进行扫描,常采用双目技术,但是精度低,只能扫描出有几何特征的物体,不能满足很多领域的要求。

另一类是主动方式,就是像物体投射特定的光,其中代表技术激光线式的扫描,精度比较高,但是由于每次只能投射一条光线,所以扫描速度慢。另外,由于激光会对生物体以及比较珍贵的物体造成伤害,所以不能应用于某些特定领域。

新兴的技术是结构光式的扫描,结构光也属于主动方式,通过投影或者光栅投射同时多条光线,就可以采取物体的一个表面,只需要几个面的信息就可以完成扫描,最大的特点是

扫描速度快，而且可编程实现。

还有一种是低频脉冲波(低频声波)式原理，主要应用于物位测量方面。



图 1 3D 扫描仪

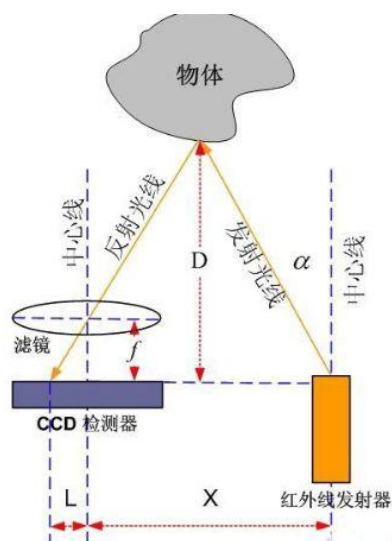


图 2 三角法测距原理

以上是现在 3D 扫描获取点云的普遍方法。

而本文利用了透视成像的特点来获取点云的坐标信息。这个与三角测距法有所不同。下文还会讨论点云拼接，网格重构，曲面细分等技术。

2.2 - 本文简介

本文将会围绕着 3D 扫描系统的构建，紧抓成像和透视投影的规律，对整个数据处理管道(**processing pipeline**)进行叙述。一开始要在保证正确性的同时从 2D 的图片中获取 3D 的顶点，这是难点之一，本文将会从“获取深度”函数开始，有基于不同方式扫描的扫描模型供生成 3D 局部坐标，并计算其对应的转换矩阵(**transformation matrix**)。生成点云(**point cloud**)后，就要开始进行结构或散乱点云进行网格重构，这个也是难点之一，国内研究很少或者不成熟。最后要进行曲面细分(**surface subdivision**)或曲面拟合进行来优化模型的形状和精度。实际上到这里，扫描的模型就可以供实际使用了，只需要把网格数据保存成常用的格式。

2.3 实验环境

本文所提到的东西都要用硬件和软件结合，主要靠编程来实现算法。下面来简单介绍一下实验环境。

IDE: (集成开发环境)

Visual Basic 6.0

Visual Studio 2012 (C++)

Arduino 1.0.5

库(library)引用:

TrueVision3d 6.5 (3D 引擎, 主要用其已封装好的加载与渲染)

MsComm Control 6.0 (用于与 Arduino 微处理器进行通讯, 实现部分的硬件接口)

实验数据模拟工具:

3DMAX 2009. (游戏动画电影专业 3D 建模软件)

3. 设计、算法推导、算法分析

3.1 实物结构设计与 GetDepthFromOffset 函数

先介绍一下三维扫描仪的发展历程

第一代三维扫描仪：点测量

代表系统有：三坐标测量仪；点激光测量仪；关节臂扫描仪（精度不高）通过每一次的测量点反映物体表面特征，优点是精度高，但速度慢，如果要做逆向工程，只能在测量较规则物体上有优势。

定义：适合做物体表面误差检测用。

第二代三维扫描仪：线测量

代表系统有：三维台式激光扫描仪，三维手持式激光扫描仪，关节臂+激光扫描头。通过一段（一般为几公分，激光线过长会发散）有效的激光线照射物体表面，再通过传感器得到物体表面数据信息。

定义：适合扫描中小件物体，扫描景深小（一般只有 5 公分），精度较低，此代系统是发展比较成熟的，受产品原理影响已经是过渡性产品。

第三代三维扫描仪：面扫描代表系统：拍照式三维扫描仪，三维摄影测量系统等。通过一组（一面光）光栅的位移，再同时经过传感器而采集到物体表面的数据信息。

而我们这里先讨论的是线测量，而这些“线”多了（光栅），也就成了面扫描了。

这个设计的基本思想：利用与主光轴平行的红线照在物体上，红线成像会有一些的**偏移(offset)**（或者说是**扭曲(distortion)**），就可以计算出物体的凹凸了。

物体成像之所以有偏移，是因为透视关系。在 3D 渲染中，描述一个摄像机一般都用视载体，如图 3。这样就很容易让人联想到**投影(projection)**。

4.3 视载体

图 4.9 给出了视载体的示意图，视载体用来表示一个空间范围，位于这个空间范围内的三维场景的任何物体都可以被看到。视载体的形状是一个棱锥，其顶点表示摄像机坐标系的原点，之所以具有这样的形状，是因为要用视载体来表示摄像机通过一个像和原（实际上就是计算机所能看到的实际范围）。视载体由六个平面围成，其中的四个平面与场景的边缘相对应，分别被称为左、右、底、顶视界面。另外两个平面称为近视界面和远视界面，它们定义了最近和最远距离，对于场景中的物体，只有位于该距离范围内的物体才能够被摄像机所看到。

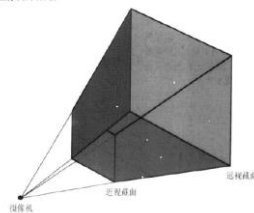


图 3 视载体的介绍

为了配合选用的 3D 引擎使用的坐标系，要规定选用 Y 轴竖直向上，如图 5。

并且经过分析比较，决定选择着红线与摄像头主光轴平行射出的激光投影方式。

$$\vec{X} \times \vec{Y} = \vec{Z}$$

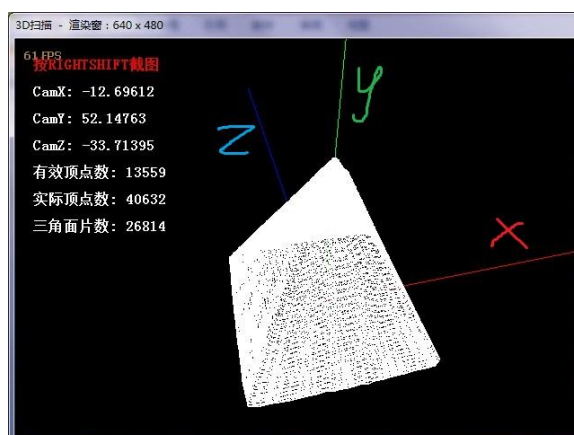


图 4 3D 引擎用的是 Y 轴向上左手系坐标

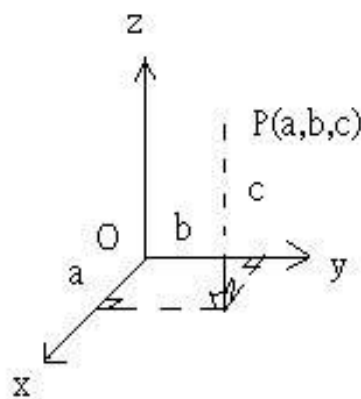


图 5 一般的数学坐标系右手坐标（也是 3DMAX 用的）

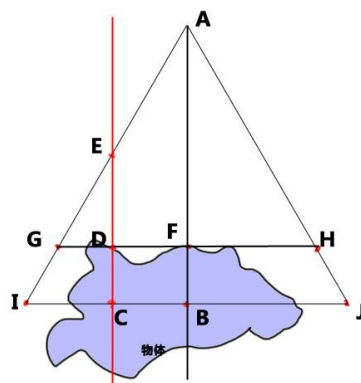


图 6 激光投射方式的平面数学模型（俯视图）

透视投影的过程中存在非线性(non-linear)关系。例如三个共线的成像点 $A'B'C'$, 若线段 $A'B'=B'C'$, 则其原来对应的 3D 空间中的线段 $AB \neq BC$, 即屏幕空间到世界空间的映射是非线性的, 但经过上述简化以后投影的定量关系应该会简化

不少。

选择了红激光(Red laser)到主光轴(the Main Axis)的距离为定值的方法是来主要自于灵感,就是感觉比红线转动或移动合理和方便。于是我们要寻求数学的定量关系了。我们把问题抽象为如下的数学模型,图 6。

基本所有的薄透镜成像的都可以用 $\angle IAJ$ 来表示其视角。而在研究过程中我们一直把 A 当作了透镜的焦点,但这明显是不对的,直到我们用单反来做取景器时,由于单反镜头是有多组透镜,所以我们只能关注这组镜片等效的一个透镜。在理论模型与实现的匹配的研究中,发现这个数学模型的视角起点 A,并不是等效透镜的焦点,而是 光心(The optical center)。然后我们要求解的是“深度”CD,下图 7 是成像示意图及各个图形元素的实际意义。

AB: 主光轴所在直线上的线段,之后延伸为光心到背景板(“Background Board”)的距离,命名为“标准深度”(“Standard Depth”),但是有一点要注意的,所谓的背景板只是选用一种标准,是想象出来的,无论是否有真实的背景板。实际上采用多远的背景板,测量点到光心的距离都是不变的,变的只是测量的深度。

DF=BC: 红激光线到主光轴的距离

CD: 物体到背景板的距离,命名为深度

GH: 视截面的长度,D 在此视截面上。

IJ: 视截面的长度,C 在此视截面上。

$\angle JIA$: 记作 $\angle \theta$, 可视角范围

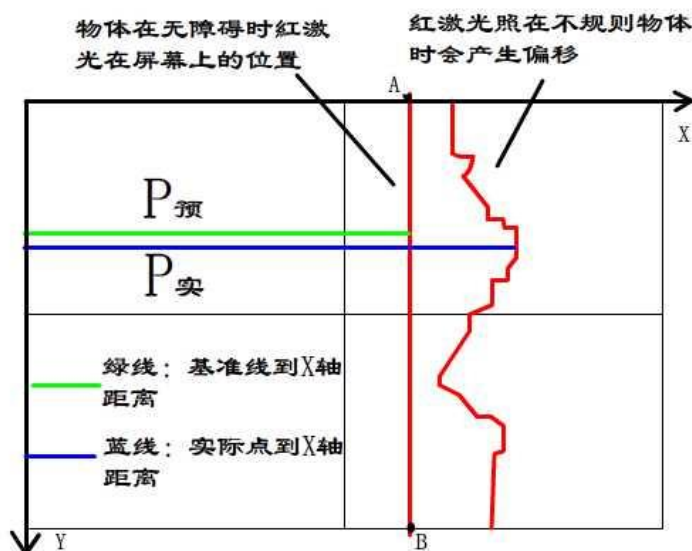


图 7 激光线在主光轴右端时的成像示意图

然后我们把目光从世界空间转到屏幕空间。要结合成像才能推出公式。

P 预: “基准线”(“Standard Line”)在屏幕上的 X 坐标,这个基准线与图片横边垂直平分线的距离,与上文提到的“标准深度”有关,原因显而易见。

P 实：实际成像点在屏幕上的 X 坐标

PictureHeight：照片纵向像素数

PictureWidth：照片横向像素数

注*：为了配合计算机图形处理的规则我们要把屏幕空间中的坐标系的 Y 轴正方向指向下方。

算法基本思想（由偏移得高度）：照片上偏移的比例等于实际偏移的比例

如图 6 所示，按照这个思路，我们可以列出方程：

设 CD 为 $h_{物}$ ，

标准深度 AB 为 H，

DF=BC 为 d，

可视角 $\angle JAI$ 为 $\angle \theta$

$$\frac{DF}{GH} - \frac{BC}{IJ} = \frac{P_{实} - P_{预}}{PictureWidth}$$

即：

$$\frac{d}{GH} - \frac{d}{IJ} = \frac{P_{实} - P_{预}}{PictureWidth}$$

视截面 GH 长为：

$$2(H - h_{物}) \cdot \tan \frac{\theta}{2}$$

视截面 IJ 长为：

$$2H \cdot \tan \frac{\theta}{2}$$

设 PictureWidth 为 W，经过合并与移项得：

$$h_{物} = \frac{2H^2 \cdot \tan(\frac{\theta}{2}) \cdot (P_{实} - P_{预})}{Wd + 2H \cdot \tan(\frac{\theta}{2}) \cdot (P_{实} - P_{预})}$$

上面那条式子适用于红激光在主光轴右侧，若是在左侧，解应该为：

$$h_{\text{物}} = \frac{-2H^2 \cdot \tan(\frac{\theta}{2}) \cdot (P_{\text{实}} - P_{\text{预}})}{Wd - 2H \cdot \tan(\frac{\theta}{2}) \cdot (P_{\text{实}} - P_{\text{预}})}$$

其中，

$$P_{\text{预}} = \frac{\text{PictureWidth} \cdot \left(1 - \frac{d}{H \cdot \tan(\frac{\theta}{2})}\right)}{2}, \quad (\text{激光在主光轴左边})$$

$$P_{\text{预}} = \frac{\text{PictureWidth} \cdot \left(1 + \frac{d}{H \cdot \tan(\frac{\theta}{2})}\right)}{2}, \quad (\text{激光在主光轴右边})$$

以上算法推导，需要实验进行验证。实验方法为：放了一个正方体，扫描面到背景板的距离是 300mm，然后编程，测量、输入数据，对正方体部分的偏移像素采样，计算(编程在后面会详细讲述)，得到结果约 319mm，如图 8 所示。而且在测量的时候很多估测，由此我们得出算法较合理的结论，并给函数命名为 **GetDepthFromOffset**（从偏移得到深度）

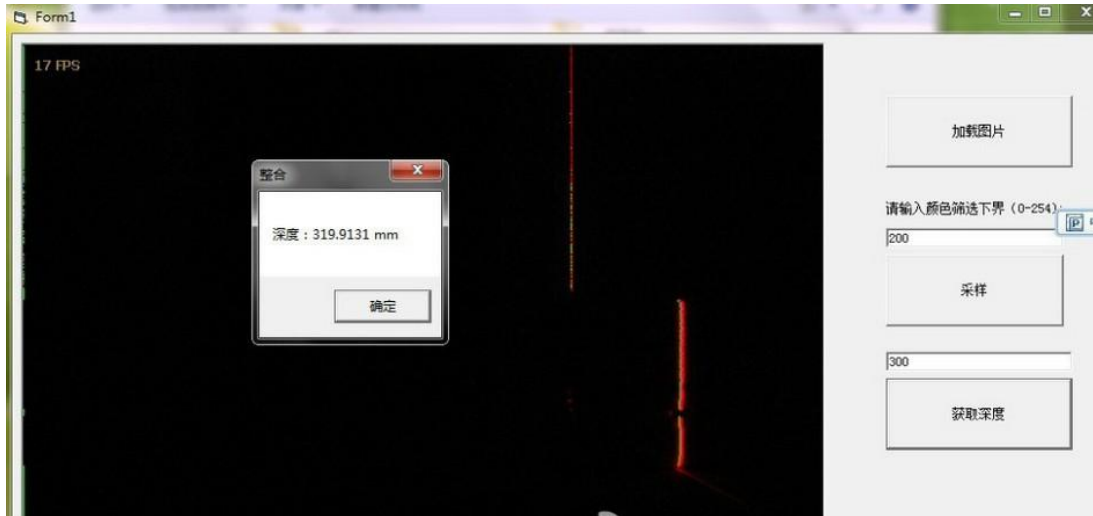


图 8 函数的编程验证

值得一提的是，根据透镜成像规律，如果激光放在主光轴右边，则深度越深，偏移点会往右偏移；同样，激光放左边，偏移成像点只会往左偏移。所以在图片分析和采样的时候至多只用分析左或右半张图片，而无需处理整幅图片。

3.2 图片采样 (Sampling)

3.2.1 原创方法的分析与改进

因为射出来的激光会有一定宽度，所以在图片上一般会占大于 1 个像素，但计算的时候使用的是矢量化(vectorisation)信息。所以就要进行图片采样。图片采样是对像素进行遍历，分析出矢量信息。

由于我们采用红色激光，所有我们先讨论用红色激光时的情况

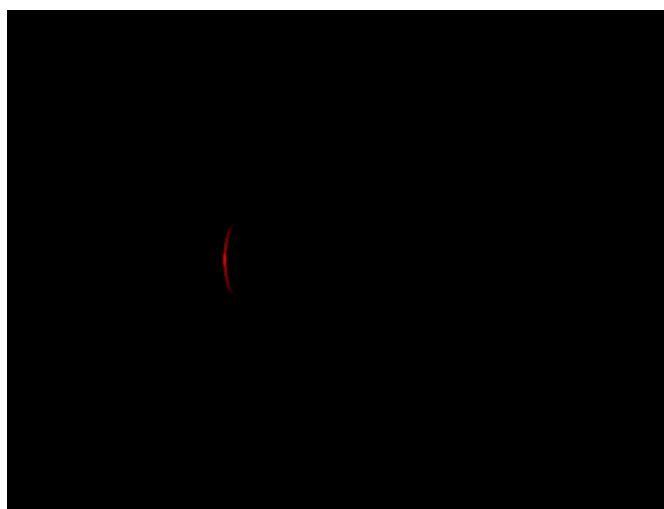


图 9 扫描照相的效果图

方法 1:

- (1) 设定颜色筛选阈值(“color filter threshold”), 即 R 分量不达到此值就不做任何处理(相当于当作黑色)
- (2) 记录下第一个入选的红点: x_1
- (3) 记录下最后一个入选的红点: x_2
- (4) 采样点的 x 为 x_1 与 x_2 的中点

但这样的采样方法过于草率，不一定适用于不同的情况。

方法 2:

这个方法是先进行 RGB 分量(component)的阈值筛选，选出一行中最“红”的那几个点，再求带权平均值。

我们根据实验照片的取色分析，画出理想状态时 R 与 X 像素坐标之间的图像(离散点组)，如图 10。

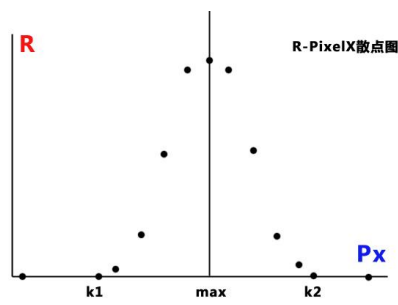


图 10 R 与 X 像素坐标之间的散点图像

理想状态是会出现 $[k1,k2]$ 区间的一个突起区域，并且会大概在 $[k1,k2]$ 中点处取得最大值 $R(max)$ ，即 $k1$ 与 $k2$ 在 Px 轴上关于 max 对称，其中最大值点可以是几个临近的点。

但是由于激光具有一定的粗细，可以粗达几毫米，则在 Max 附近的衰减的光很容易就会被黑色区域等“吸光区”吸掉，则图像就会变得**不对称**， R 分量最大值也不会在 $[k1,k2]$ 中点处取得，这样就会造成很大的采样误差。

而且这样做一个有一个问题，如果取整像素点，意味着扫描出来的最小单位(精度)会是

$$\frac{\text{横向实际可见宽度(mm)}}{\text{横向像素点数(个)}}$$

这样不平滑现象会非常严重，会出现较明显的转角。可以通过提高分辨率来细化。

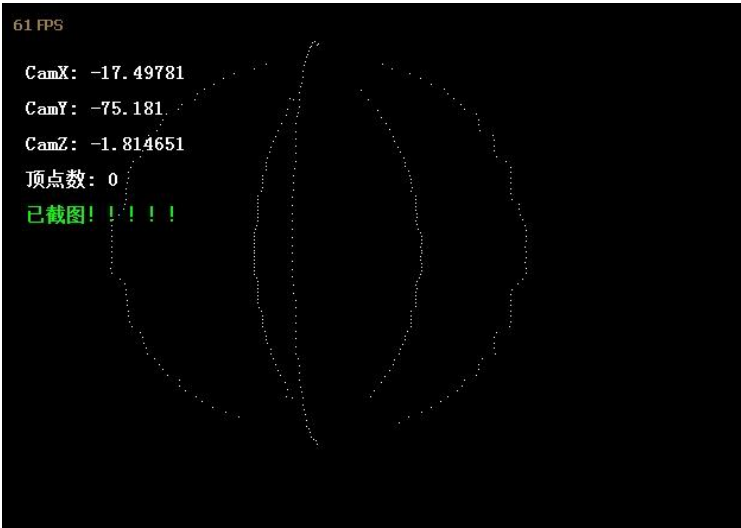


图 11 取整像素点的后果，扫描体为球(用 3DMAX 模拟图片来处理)

表 1 采样方法比较

采样方法	评价
------	----

像素 X 坐标跨度的平均值	草率
给定容差对 RGB 分量均做判断	抗噪能力最强，但要注意颜色的匹配

3.2.2 成像的 Y 坐标区间重叠

在后续实验当中发现，一行像素不一定只有一个入选点，可以有多个。下面是物体的 3D 构造和正视图成像。

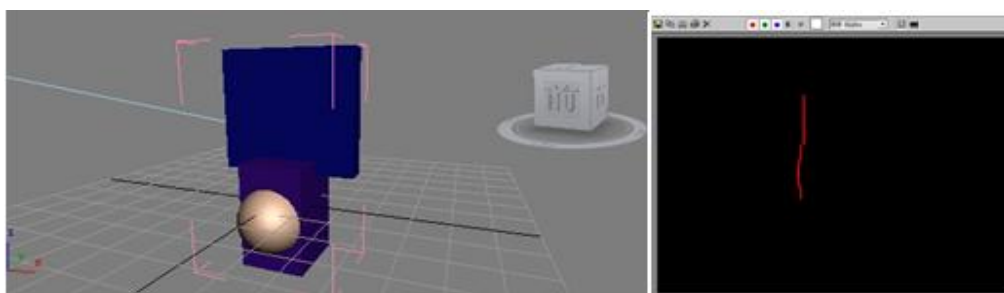


图 12 (a) 原来的几何体（上半部分有个可移动的长方体）；(b) 激光扫描线

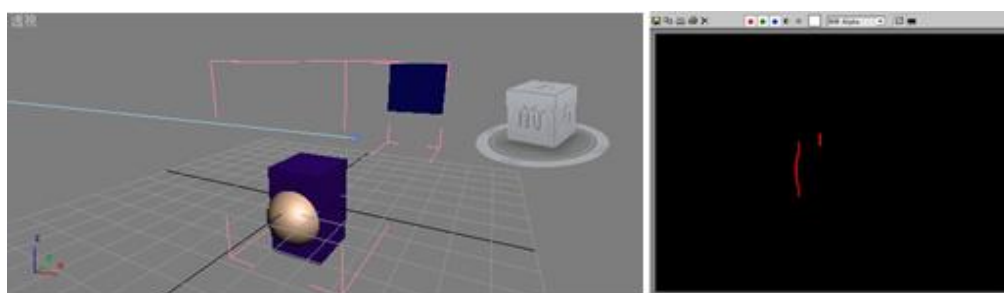


图 13 (a) 把上半部分的长方体移到远处的几何体；(b) 激光扫描线

图：原来几何体的成像

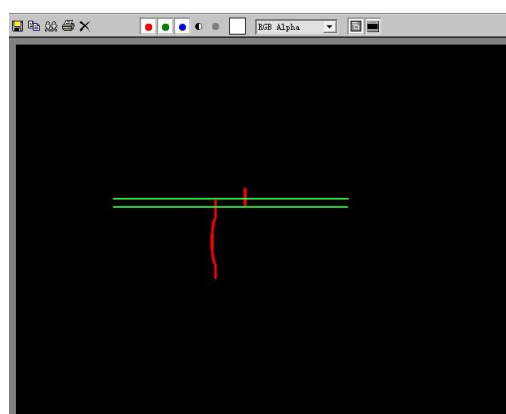


图 14 扫描线错位示意图

从图 14 扫描线错位示意图可以看到，在绿线之间的部分，每一行像素都会有两条激光光线这个就意味着把一行看作一条激光的数据采样单位是不合理的。需要有替代的解决办法。

如图 15 所示,在遍历(iterate)每一行像素时可以判断像素 P_i , P_{i+1} 是否符合入选条件。设入选的像素 $P_i = 1$, 没入选的 $P_i = 0$, 则:

$$\begin{cases} P_{i+1} - P_i = 1 (i \leq n-1), \text{ 则设 } P_{i+1} \text{ 为采样区间左端点} \\ P_{i+1} - P_i = -1 (i \leq n-1), \text{ 则设 } P_i \text{ 为采样区间右端点} \end{cases}$$

这样就能把每一行像素分成多个采样区间分别进行采样,这样结果不会被多条激光线共同影响,也不会漏点。

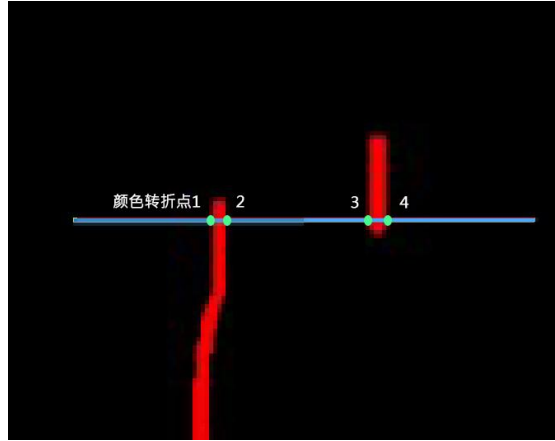


图 15 激光线分区采样示意图

3.2.3 简化计算

在实验时发现红激光线成像的横向偏移只会往其中一个方向,这取决于激光是在摄像机的左边还是右边:若激光在相机左边,则激光会往左偏;反之则是往右偏。这个原因很明显,激光的成像是投影得来的,所以在主光轴左边的光只会投影到左半边的感光底片,至于测得深度越深时,即某个空间点越靠近观察者位置时,其到主光轴的距离与此深度视截面的比会变大。这样一来,在确定激光的摆放位置后,分析图片时只需要遍历一半的像素了。

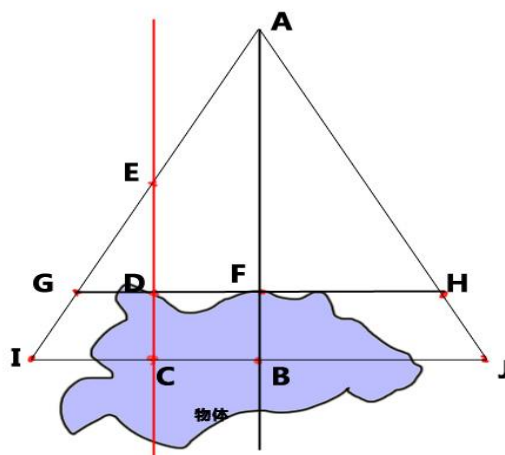


图 16 $DF/GH > BC/IJ$, 而 D 的深度比 C 深, 则 D 在成像中会更靠右边

3.2.4 精度 (precision)

现在最影响精度的就是图片光栅化造成的量化误差。

经过分析, 可以知道要增加生成点的精度, 需要:

1、用更大分辨率的图片。

2、适当增加 CamToLight, 因为增加 CamToLight 之后激光线对深度变化的敏感度(sensitivity)增高, 但要注意万一激光照射物体的点太靠近镜头, 可能会出了视线范围。



图 17 CamToLight 影响 x 坐标对深度变化的敏感度

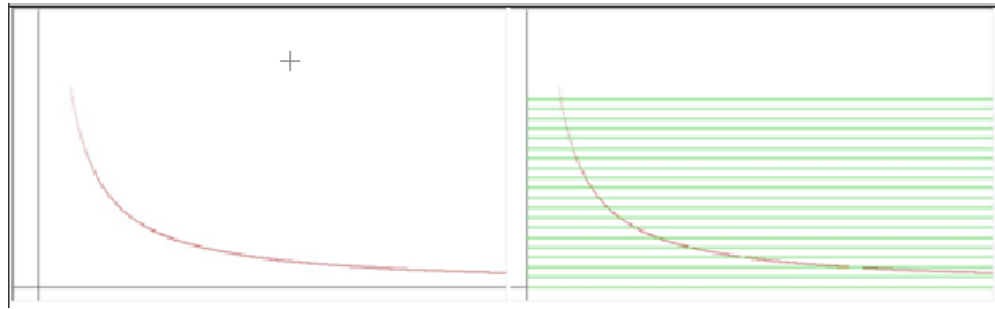
左图: 两个深度的 x 坐标差为 $\left(\frac{1}{3} - \frac{1}{5}\right) \cdot \text{PictureWidth}$

右图: 两个深度的 x 坐标差为 $2 \cdot \left(\frac{1}{3} - \frac{1}{5}\right) \cdot \text{PictureWidth}$

3、改进采样算法。

4、用更加细的激光。

5、镜头在还能看到激光照射点的前提下尽量靠近物体, 越靠近物体成像 x 坐标对深度的敏感度越高。


 图 18 函数 $u = \text{CamToLight} / (z \cdot \tan(\theta/2))$ 的图像

如图 18 函数 $u = \frac{\text{CamToLight}}{z \cdot \tan(\frac{\theta}{2})}$ 的大致图像， z 为局部 z 坐标，而随着 z 的

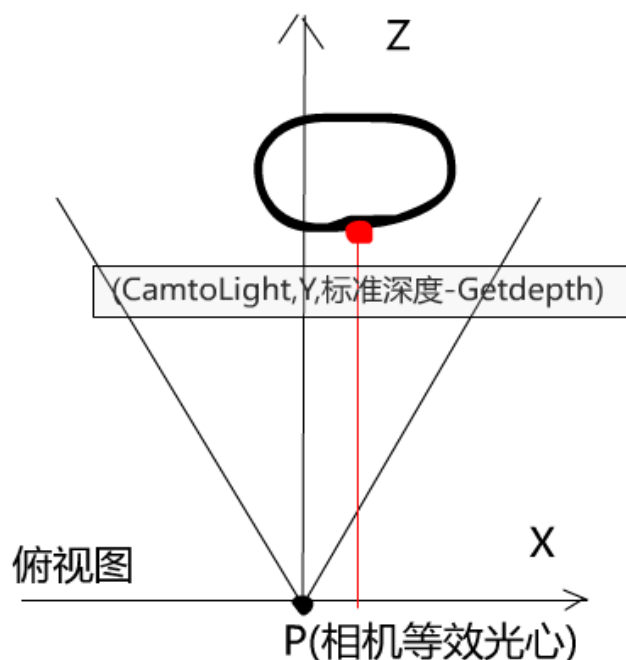
线性增长，成像线在屏幕的 x 坐标会以反比例函数的形式向屏幕垂直中线逼近，(as Z increases linearly, the x coordinate will approach to the middle of screen in inverse proportion) 函数 u 就反映了这一种趋势。而这个比例 u ，被 CCD 等光学成像传感器栅格化成了像素，则实际上每个像素的横坐标对应的深度应该是像下图那样，把这个反比例函数 n 等分， n 为一半的横向屏幕的像素数，然后求这些平行于横轴的直线和反比例函数交点的 x 坐标，即每个像素的横坐标对应的深度。

则非常明显，随着 $z \cdot \tan(\frac{\theta}{2})$ 的增长，两个交点的 x 坐标差越来越大，则可以知道相机（激光）越靠近物体，精度就越高。

3.3 生成三维顶点 - 局部坐标(Local Coordinate)

在完成了对单点测深度的算法与编程实现以后，开始进入生成三维空间点的步骤。由于拍摄成像的投影使三维信息掉落至二维，丢失了信息，不可能凭空恢复。不止在欧氏几何空间，所有的投影在丢失信息后再升维都不再可能获得唯一解，除非加限制条件。高维向量的信息量比低维向量的信息量多，这是一个事实，所以从 2D 图片生成 3D 顶点必须要有足够的信息量才能实现升维。而这些多出来的信息一般都是限制条件。

为了简化工作，我们确立了一个世界坐标系(World Coordinate System)。并且求得局部坐标(Local Coordinate)。以摄像机位置（等效光心）为原点，摄像机向量为局部坐标系 z 轴，局部 x 轴指向观察者正右方，局部 y 轴指向观察者正上方，建立局部坐标系（局部参考系）。



LocalX: 绝对值(absolute value)为 CamToLight，符号由激光在主光轴(main axis?)的左右边决定。摄像机正交基 x 轴正方向朝着视角的右边， z 轴是视向量。

LocalY: 这个轴是垂直于水平面的轴，求 Y 坐标即求高度。但是有一点是我一开始没有考虑到，就是成像点对应的空间点的实际分布是非线性(non-linear)的，即不是均分的。经过探究，竖直方向的坐标应满足如图 19 规律。

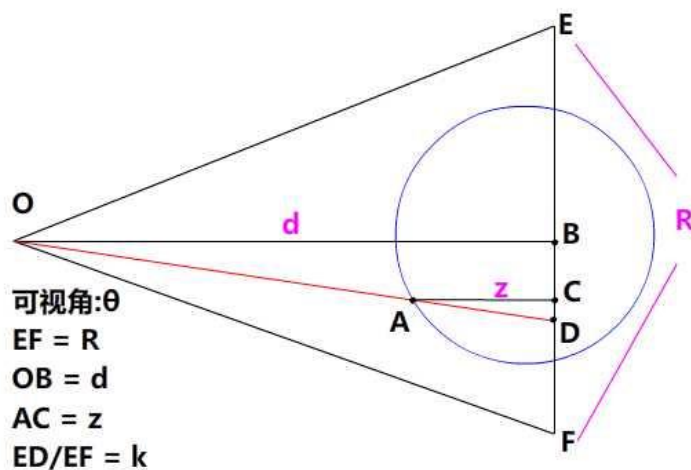


图 19 y 与深度关系示意图

如图所示, (图为侧视图)

EF : 长度为 R, 某一视截面的实际高度, 方便起见我们还假设 EF 就是背景板

A: 物体上的一个可视点

OB : 标准深度, 焦点到背景板的距离

AC: 测得深度, 值为 z

ED/EF: 这是一个比值, 值为 k, 可以根据成像获得, 即 (屏幕上端到 A 点成像的距离) / (屏幕高度), 这个可以用像素坐标去计算。

d: 标准深度

z: 到背景板的距离(不是 z 坐标!)

现已知 d, z, k, 要求出 EC 的表达式

$$|ED| = Rk$$

$$|EB| = \frac{R}{2}$$

$$\therefore |EB| = Rk - \frac{R}{2}$$

$$\therefore |CD| = \frac{z}{d} \left(Rk - \frac{R}{2} \right)$$

$$\therefore |EC| = |ED| - |CD| = R \left(k - \frac{kz}{d} + \frac{z}{2d} \right)$$

其中，

$$R = 2d \tan\left(\frac{\theta}{2}\right), \theta \text{ 为纵向视角}$$

同理可得 A 若在 OB 上方时 EC 表达式(expression)不变。

所以最后有，

$$LocalY = \frac{R}{2} - R \left(k - \frac{kz}{d} + \frac{z}{2d} \right)$$

显然，y 坐标与深度（局部 z 坐标）有关联，则在编程实现的时候应该先求 z 坐标。而且由这个性质，可以有一个推论：同样的 3D 坐标-Y 分量可以由 2D 坐标-Y 分量一样的像素点生成（两个像素点的 2D-X 分量不一样）。这就意味着遍历同一行像素可能会遇到多个像素点。

LocalZ: “背景板”只是一个虚拟的概念，实际上背景板只是一个深度为 standardDepth 的一个视截面。所以局部 z 坐标值为 StandardDepth - GetDepth。实验证明 GetDepth 函数可以为负值，照射点的 z 坐标可以大于 StandardDepth。

散乱点云需要在转换矩阵上讨论更多，请看下一节。

在编程实现(programming implementation)的处理流程中，要先生成 z 坐标才能生成 y 坐标。

3.4 世界变换(World Transformation)

生成顶点只是完成对局部凹凸的刻画，此时的坐标仍是局部坐标。要对每一组的局部点云，进行世界变换(world transformation)，合成一个处在同一坐标系的点云(point cloud)。

就如手持式扫描仪一样，物体可以不动，只是扫描仪器在改变位置和方向来扫描，这时每张图片（每一列激光线成像）就对应着一个变换矩阵(matrix)。仿射变换(affine transformation)一般需要一个偏移的向量和旋转的 3X3 矩阵才足以实现变换，所以一般情况下都采用 4X4 矩阵，4 分量的坐标(x,y,z,w)，其中 w 默

认为 1, w 分量是齐次坐标(Homogeneous coordinates)), 这样可以实现任意刚体变换(arbitrary rigid transformation)。

为了获得这个矩阵, 有些物理上的设计和数学限制条件是必要的。

我们需要定义一个真实世界中的世界坐标系, 对应着虚拟世界的世界坐标系。这里就要假设那个真实世界的世界坐标系原点是一个不会移动的物体。但是由于摄像机必须移动以达到自由移动的目的, 所以要加一个传感器主机(不动)并且用角度传感器(电线或机械臂, 用于测量相对于主机的位置)连接着摄像机。

在理想化模型中, 连接传感器主机和摄像机的机械臂分段都可以用向量来描述, 如果设主机的质心为世界坐标系原点, 则在确定世界坐标系的同时就可以知道:

$$\overrightarrow{CamPos} = \sum_{i=1}^n \overrightarrow{Offset(i)}$$

\overrightarrow{CamPos} 为 Camera 坐标, $\overrightarrow{Offset(i)}$ 为每个机械臂分段的向量。这里还假设知道了摄像机视向量 $\overrightarrow{CamVector}$ 。

这里使用的是列向量 (Column vector), 则此 4x4 仿射变换矩阵为:

$$M = \begin{bmatrix} x_1 & x_2 & x_3 & CamX \\ y_1 & y_2 & y_3 & CamY \\ z_1 & z_2 & z_3 & CamZ \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

可以想到用 M 左乘(premultiplication)某个向量的结果向量, 就是计算原来坐标分量分别在新基上的投影和, 得到在世界坐标系中的新分量。

下标 1,2,3 分别指摄像机局部参考系的正交基(Orthogonal basis)的 x,y,z 轴, 其中摄像机向量设为摄像机正交基的 z 轴,

$$\overrightarrow{CamVectorZ} = (x3, y3, z3)$$

(跟局部坐标深度的定义相似), 模(modulus)为 1。其余两个轴需要进行正交化(orthogonalization)。由于选择的世界坐标系是 y 往上的左手系, 所以摄像机局部坐标系要保持手向性(英文?), 故:

$$\overrightarrow{CamVectorX} = \text{Normalize}(z3, 0, -x3)$$

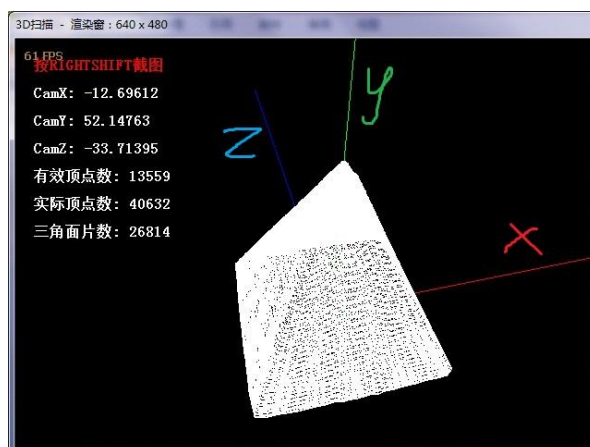


图 20 camera 正交基要 and 世界坐标系手向性一致

Normalize()为归一化函数。上述式子设了 Y 坐标为 0，这个就需要规定摄像机不会绕主光轴（视向量）翻转，如果要处理倾侧（翻滚）的问题，就需要再合并一个处理翻滚角(rolling angle)的旋转矩阵(rotation matrix)。

根据坐标系定义，

$$\vec{x} \times \vec{y} = \vec{z}$$

根据拉格朗日公式，

$$\vec{a} \times (\vec{b} \times \vec{c}) = (\vec{a} \cdot \vec{c}) \vec{b} - (\vec{a} \cdot \vec{b}) \vec{c}$$

可知，

$$\vec{x} \times \vec{z} = \vec{x} \times (\vec{x} \times \vec{y}) = (\vec{x} \cdot \vec{y}) \vec{x} - (\vec{x} \cdot \vec{x}) \vec{y} = -\vec{y}$$

所以，

$$\overrightarrow{CamVectorY} = \text{Normalize} (-\overrightarrow{CamVectorX} \times \overrightarrow{CamVectorZ})$$

假设正交基已化为标准正交基，则上文中的转换矩阵 **M** 可以分解为一个平移阵和方向余弦阵（的乘积）：

$$\text{记 } M = T_{cam} M_{DirCos} = \begin{bmatrix} 1 & 0 & 0 & CamX \\ 0 & 1 & 0 & CamY \\ 0 & 0 & 1 & CamZ \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 & 0 \\ y_1 & y_2 & y_3 & 0 \\ z_1 & z_2 & z_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

在真正的物理实现中，一般不会直接有向量表达，而是用三维水平仪获取三个方向的角度。我们可以用多个 tait-bryan 角来描述，可以描述摄像机的朝向。But tait-bryan angle is often confused with Euler angle. In order to clarify the problem, We name the angles in terms of rotation axis. 设摄像机绕世界 Y 轴（竖直）旋转的角为 Y，本文中称为 EulerY，也称偏航角(yaw angle)，与水平面的线面角

的角度为 X ，本文中称为 EulerX，也叫**俯仰角(pitch angle)**，绕摄像机向量的旋转角为 Z ，本文中称为 EulerZ，叫**翻滚角(rolling angle)**。

为了方便，我们不妨来定义上述旋转方向为旋转正方向。经几何上的分析可以得，LocalZ 轴（camera 正交基 z 轴）

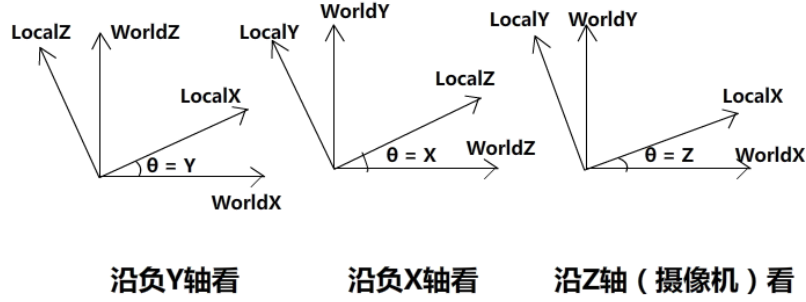


图 21 不同角度观察坐标系

经过几何分析可以得，相机局部 Z 轴（camera 正交基 z 轴，下标 3 代表 z 轴）

$$x_3 = -\cos(X)\sin(Y)$$

$$y_3 = \sin(X)$$

$$z_3 = \cos(X)\cos(Y)$$

而且 z 轴模为 1， x 轴模为 $d = \sqrt{1 - y_3^2}$ ， y 轴模为 $1 \cdot \sqrt{1 - y_3^2} \cdot \sin\left(\frac{\pi}{2}\right) = \sqrt{1 - y_3^2}$

若只取矩阵左上角 3×3 子矩阵来讨论，则可得

$$M_{\text{DirCos}} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{z_3}{\sqrt{1 - y_3^2}} & \frac{-x_3 y_3}{\sqrt{1 - y_3^2}} & x_3 \\ 0 & \frac{x_3^2 + z_3^2}{\sqrt{1 - y_3^2}} & y_3 \\ \frac{-x_3}{\sqrt{1 - y_3^2}} & \frac{-z_3 y_3}{\sqrt{1 - y_3^2}} & z_3 \end{bmatrix}$$

$$\begin{aligned}
 &= \begin{bmatrix} \frac{\cos(X) \cos(Y)}{\cos(X)} & \frac{\sin(X) \cos(X) \sin(Y)}{\cos(X)} & -\cos(X) \sin(Y) \\ 0 & \frac{\cos^2(X) (\cos^2(Y) + \sin^2(Y))}{\cos(X)} & \sin(X) \\ \frac{\cos(X) \sin(Y)}{\cos(X)} & \frac{-\cos(X) \cos(Y) \sin(X)}{\cos(X)} & \cos(X) \cos(Y) \end{bmatrix} \\
 &= \begin{bmatrix} \cos(Y) & \sin(X) \sin(Y) & -\cos(X) \sin(Y) \\ 0 & \cos(X) & \sin(X) \\ \sin(Y) & -\cos(Y) \sin(X) & \cos(X) \cos(Y) \end{bmatrix}
 \end{aligned}$$

但其实这个方向余弦阵也可以用两个绕轴旋转矩阵的结合：

$$\begin{aligned}
 R_{\text{EulerY}} R_{\text{EulerX}} &= \begin{bmatrix} \cos(Y) & 0 & -\sin(Y) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(Y) & 0 & \cos(Y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(X) & \sin(X) & 0 \\ 0 & -\sin(X) & \cos(X) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos(Y) & \sin(X) \sin(Y) & -\sin(Y) \cos(X) & 0 \\ 0 & \cos(X) & \sin(X) & 0 \\ \sin(Y) & -\cos(Y) \sin(X) & \cos(X) \cos(Y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= M_{\text{DirCos}}
 \end{aligned}$$

而用角度表示的矩阵有利于结合传感器进行编程

用正交基或欧拉角旋转矩阵在模拟实验和实际使用时各有优点。然后上文显示了，用正交基表达的矩阵只能表示两个欧拉角（事实上向量绕着自身旋转还是原来的样子），所以要完成转换还是需要加个倾侧的矩阵 R_{EulerZ} ，即绕视向量旋转的矩阵

$$R_{\text{EulerZ}} = \begin{bmatrix} \cos(Z) & -\sin(Z) & 0 \\ \sin(Z) & \cos(Z) & 0 \\ 0 & 0 & E \end{bmatrix}$$

在计算机 3D 图形学中，渲染管道的变换阶段，从世界空间转到摄像机空间，摄像机向量的变换会乘一个 4X4 的矩阵实现，左上 3X3 为向量对齐，右边 1X3 为平移，下边 3X1 含有 roll 的信息，右下角的元为齐次坐标 w 。但是这个会等价于先把摄像机用 T_{cam}^{-1} 把摄像机位置对齐到原点，用 M_{vector}^{-1} 把摄像机向量对齐到 z 轴上（ M_{vector} 是用世界坐标表达的，所以是从局部转到世界的矩阵），然后再实施 roll 旋转，再投影。

而我們是在从局部坐标变换回世界坐标，所以应该对投影前的摄像机空间的

点进行 roll 旋转（与上面的 roll 旋转方向相反），再用旋转变换转换到世界坐标，再平移。

所以，3D 点从摄像机空间（局部坐标）到世界空间的完整刚体变换矩阵为：

$$M_{rigid} = T_{cam} R_{EulerY} R_{EulerX} R_{EulerZ} = T_{cam} M_{DirCos} R_{EulerZ}$$

如果空间点要先和方向余弦阵相乘，则需要紧接下来乘上一个绕轴旋转矩阵，根据 Real Time Rendering 的说法绕轴旋转矩阵是：

$$M_{axis} = \begin{bmatrix} (1 - \cos\varphi)r_x^2 + \cos\varphi & (1 - \cos\varphi)r_x r_y - r_z \sin\varphi & (1 - \cos\varphi)r_x r_z + r_y \sin\varphi \\ (1 - \cos\varphi)r_x r_y + r_z \sin\varphi & (1 - \cos\varphi)r_y^2 + \cos\varphi & (1 - \cos\varphi)r_y r_z - r_x \sin\varphi \\ (1 - \cos\varphi)r_x r_z - r_y \sin\varphi & (1 - \cos\varphi)r_y r_z + r_x \sin\varphi & (1 - \cos\varphi)r_z^2 + \cos\varphi \end{bmatrix}$$

这个绕轴旋转变换的旋转正方向也是取决于手向性的。

顺带一提，取景部分的扫描结构（相机和激光）要一直保持原有的相对位置和朝向上述变换才可以得到正确的结果。也就是激光的位置必须在相机局部坐标系的 x 轴上距离为 CamToLight 的位置，这个在摄像机翻滚的时候必须要注意。（实际操作时没什么问题，但是在模拟实验时要注意整体翻滚才能得到正确的结果。）

3.5 标识点

3.5.1 标识点的作用

考虑到在实际操作中，机械臂（或角度传感器）在工程学上实现较为复杂，而如果用超声波定位，则会涉及滤波的问题而且效果不一定好，超声波发射器还有发射角度的限制。这里就要用一种方法去求解摄像机的世界坐标了。而且，如果用三维水平仪来获取摄像机的倾角，也还会有一个角（确切的说是 tait-bryan 角，虽然有时也混称欧拉角）获取不了，就是绕竖直轴旋转的欧拉角，即水平角，因为三维水平仪是用重力来测定角度的。

所以问题就是还有部分刚体信息难以直接从传感器中获取。包括世界坐标和偏航角。**(So the problem is that some of the rigid information cannot be obtained directly by sensor. These information includes world position and yaw angle.)**

在计算机 3D 图形学中，当所有顶点从模型空间合并到世界空间后，3D 点到成像的像素坐标要有如下变换：

$$\overline{P_{2D}} = M_{viewport} M_{proj} R_Z^{-1} R_X^{-1} R_Y^{-1} T_{cam}^{-1} \overline{P_{3D}}$$

如果忽略深度坐标的保留的话，

$$M_{proj} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

作用是把摄像机空间的顶点投影到 $z=1$ 平面上(齐次坐标 w 最后要化为 1)。但是 3D 渲染管道中一般使用另外一种用视截面参数定义的矩阵，把视截体转换成单位正方体再投影，方便三角形投影的裁剪。但是无论如何投影必定丢失信息。

可以看到， M_{proj} 最右一列为 0 向量，所以 $|M_{proj}| = 0$ ，所以下文就不考虑“解投影矩阵”了。

要提一下视口变换矩阵，

$$M_{viewport} = \begin{bmatrix} \frac{ScrPixelWidth}{2} & 0 & 0 & \frac{ScrPixelWidth}{2} \\ 0 & \frac{-ScrPixelHeight}{2} & 0 & \frac{ScrPixelHeight}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

这个视口变换矩阵是把投影点 (x 和 y 都在 -1 到 1 之间) 转换到以像素为单位的屏幕空间 (也叫视口)，这个矩阵的逆矩阵在下面要用到。

现在作者提出一种光学测量法，不仅不会有传感器的误差积累(error accumulation)，而且实现也非常方便。不少研究人员也在研究标识点提取的问题，然后计算出每幅图片的相对位移等信息。而这里要算出其绝对世界坐标和水平角。

根据一轮推算和设计，最终确定了一个信息量足够的标识结构。这个标识结构由一个正方形组成，其四个顶点均为有颜色 (实现时可以用高亮度的灯) 的可识别点。其中左下角的蓝色点是世界坐标系原点 (的成像)，如图 22。

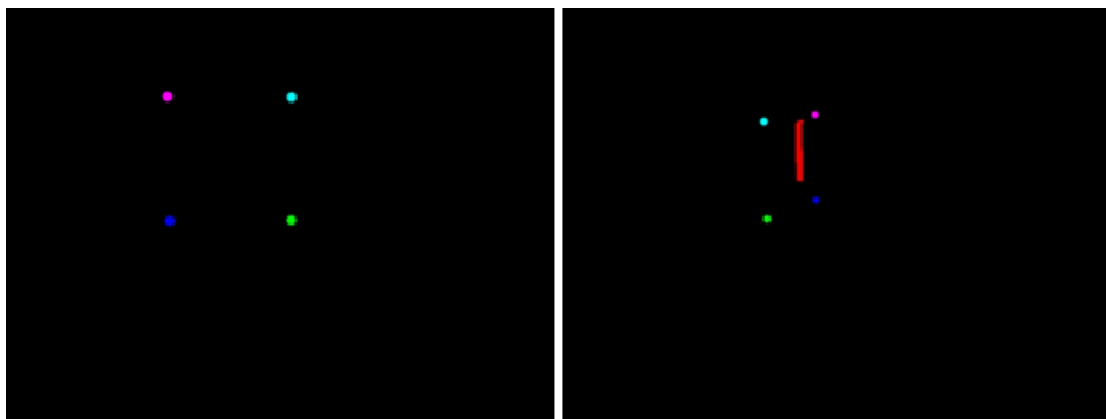


图 22 发光标识点示意图 (3DMAX 模拟渲染图)

图:

此标识结构的作用有几个:

- 1, 提供成像信息用以计算摄像机等效焦点(摄像机位置)世界坐标;
- 2, 三维水平仪不能测得的绕竖直方向渲染的摄像机转角;

这样一来, 不仅省了工程实现上的复杂, 而且光学测量会比传感器好, 因为如果传感器测量的数据还要对时间进行积分 (或者是求和), 误差也会随时间流逝而累积。

至于把一堆像素抽象为一个矢量标识点, 可以用区域生长法 (Region growing method) 把指定颜色的标识点像素群(pixel group)找出来, 再取其包围框(bounding box)的中心。这样可以粗略地提取标识点。

3.5.2 标识线段

这个标识结构之所以要用有颜色的点, 是为了提供线段 (向量) 信息。

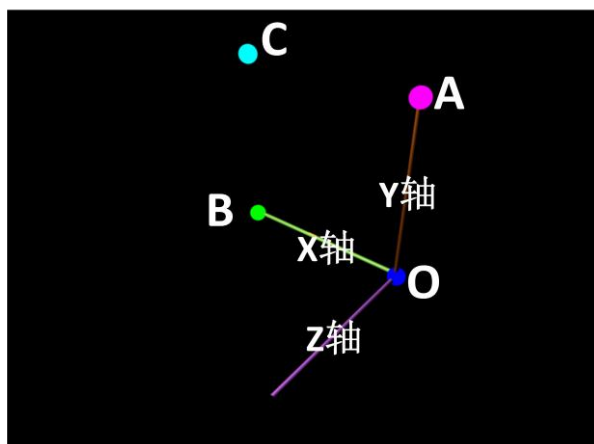


图 23 标识点的颜色可以获取线段信息

标识结构的特殊设计：

1，标识结构由四个顶点组成,且围成一个正方形，在真实世界是固定不动的。正方形的边长是已知的，物理实现和编程实现均用毫米做单位。

2，点 O 是世界坐标系的原点,则可以规定 OB 为世界坐标系 X 轴， OA 为 Y 轴。

标识点的基本思路：

(1)如果知道了标志点 O 和 B （在世界 X 轴上），在相机只有水平角时候的局部坐标，那么就可以算出水平角和摄像机世界坐标了。

(2)但一般相机会同时有偏航角、俯仰角、翻滚角，所以在这之前要先做一些透视校正的工作，预测出没有翻转角(EulerZ) 和仰角(EulerX) 时标识点成像或者局部 3D 点的信息。

3.5.3 去除翻滚角影响时（EulerZ）空间点的成像

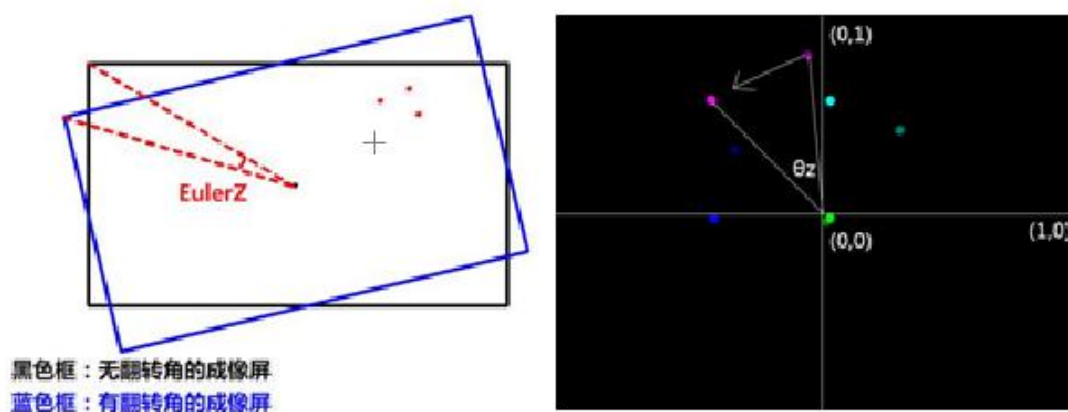


图 24 翻滚角对标志点影响示意图

在用 M_{viewport}^{-1} 把像素点映射到投影空间中，设 2D 点的坐标

$O(X_{o-2d}, Y_{o-2d})$, $A(X_{a-2d}, Y_{a-2d})$, $B(X_{b-2d}, Y_{b-2d})$, $C(X_{c-2d}, Y_{c-2d})$ (x 和 y 都在区间 $[-1,1]$ 上)。但是要处理翻滚角 EulerZ，却不能直接用下面的把用下面的矩阵 $R_{\text{PixelSpace}}$ 左乘来实现旋转：

$$R_{PixelSpace} = \begin{bmatrix} \cos(Z) & -\sin(Z) \\ \sin(Z) & \cos(Z) \end{bmatrix}$$

问题就出在这里的标识点坐标在投影空间中，也就是刚才已经把矩形的图片（下图中所示的像素空间）映射到了正方形的投影空间中。

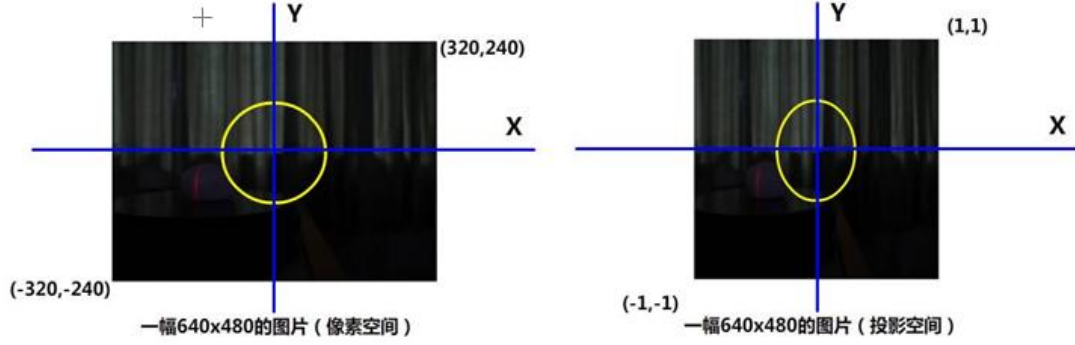


图 25 翻滚角导致像素映射失真示意图

如图 24，若相机底片宽高比不为 1（一般都不是 1），即底片不是正方形，则在像素空间中沿圆走的路径将会在投影空间中被扭曲成一个椭圆，所以要做一定的校正。

对于一个在像素空间中的，圆心在原点的圆

$$x^2 + y^2 = r^2$$

即，

$$x^2 + y^2 = r^2(\cos^2\theta + \sin^2\theta)$$

则圆可以用参数方程来表达，

$$\begin{cases} x = r \cos\theta \\ y = r \sin\theta \end{cases}$$

若设图片像素宽为 d ，像素高为 h ，且 $d > h$ ，则在投影空间中的椭圆：

$$\text{则长轴长 } 2a = \frac{2r}{d}, \text{ 短轴长 } 2b = \frac{2r}{h}$$

椭圆方程：

$$\left(\frac{x}{\left(\frac{r}{d}\right)}\right)^2 + \left(\frac{y}{\left(\frac{r}{h}\right)}\right)^2 = 1$$

即，

$$(dx)^2 + (hy)^2 = r^2(\cos^2\theta + \sin^2\theta)$$

则上述在投影空间中的椭圆的参数方程为，

$$\begin{cases} x = \frac{r}{d} \cos\theta \\ y = \frac{r}{h} \sin\theta \end{cases}$$

$$\text{因为 } a = \frac{r}{d}, \quad b = \frac{r}{h},$$

则在已知 θ_1 和 β 的时候，寻求一个从向量 $\vec{m} = (x_1, y_1)$ 到 $\vec{n} = (x_2, y_2)$ 的线性变换，即

$$\begin{cases} x_1 = a \cos\theta_1 \\ y_1 = b \sin\theta_1 \end{cases} \rightarrow \begin{cases} x_2 = a \cos(\theta_1 + \beta) \\ y_2 = b \sin(\theta_1 + \beta) \end{cases}$$

可以将 \vec{n} 的参数形式的三角函数展开，得：

$$\begin{cases} x_2 = a \cdot \cos\theta_1 \cdot \cos\beta - a \cdot \sin\theta_1 \cdot \sin\beta \\ \quad = x_1 \cdot \cos\beta - \frac{a}{b} \cdot y_1 \sin\beta \\ y_2 = b \cdot \sin\theta_1 \cdot \cos\beta + b \cdot \sin\theta_1 \cdot \cos\beta \\ \quad = \frac{b}{a} \cdot x_1 \cdot \sin\beta - y_1 \cos\beta \end{cases}$$

将这个 某椭圆上的点沿椭圆路径，“旋转”圆心角 β 的线性变换写成矩阵形式：

$$R_{\text{projSpace}} = \begin{bmatrix} \cos\beta & -\frac{a}{b}\sin\beta \\ \frac{b}{a}\sin\beta & \cos\beta \end{bmatrix} = \begin{bmatrix} \cos\beta & -\frac{h}{d}\sin\beta \\ \frac{d}{h}\sin\beta & \cos\beta \end{bmatrix}$$

则 $R_{\text{projSpace}}$ 才是可以直接左乘标识点成像的矩阵，然后就摆脱了翻滚角(EulerZ)的影响了。

其实等价的方法是在像素空间中用 $R_{\text{PixelSpace}}$ 左乘标识点 2D 成像坐标实现旋转，再映射到投影空间中

3.5.4 去除仰角影响时（EulerX）标识点的局部 3D 坐标

到这里线段的成像就是没有了翻滚角的影响时候的成像。然后处理 EulerX 角（仰角）。我们不妨先关注其中一条真实世界的竖直边 OA（或 BC）

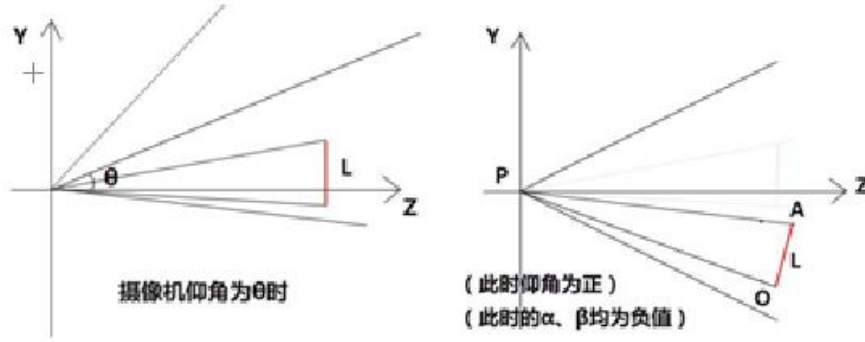


图 26 俯仰角导致成像旋转示意图

如图 26 所示说明若仰角大于 0，即“抬起头”，其成像会相对往下偏。而且很显然，向量 \overrightarrow{OA} 和 $(0,1,0)$ 夹角为仰角的大小。然后由上面的图可以知道，相机在世界空间中改变俯仰角，相当于在相机局部空间中 OA 绕摄像机 X 轴旋转一个 EulerX 角，即：

$$\langle \overrightarrow{OA}, \overrightarrow{O'A'} \rangle = \langle \overrightarrow{OA}, \overrightarrow{Z_{world}} \rangle = |\text{EulerX}|$$

但是一点要注意，由于透视成像特点，在同一个摄像机位置改变仰角会改变空间点在摄像机空间的深度，导致两个点的 X 坐标会有所偏移，其组成的向量不一定平行于屏幕的纵边。

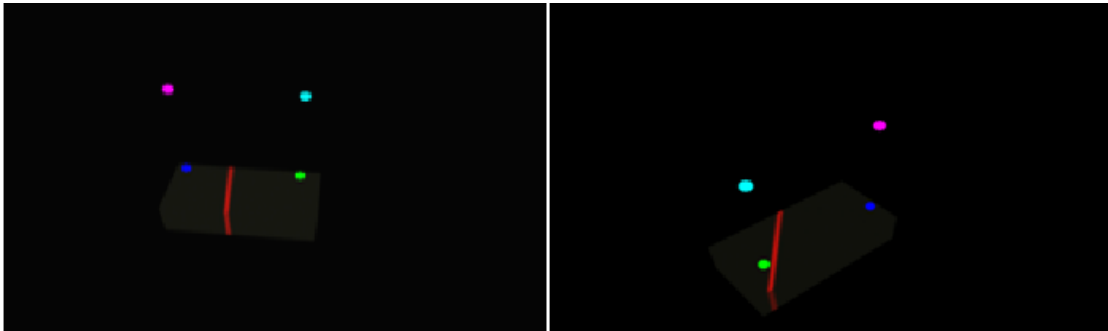
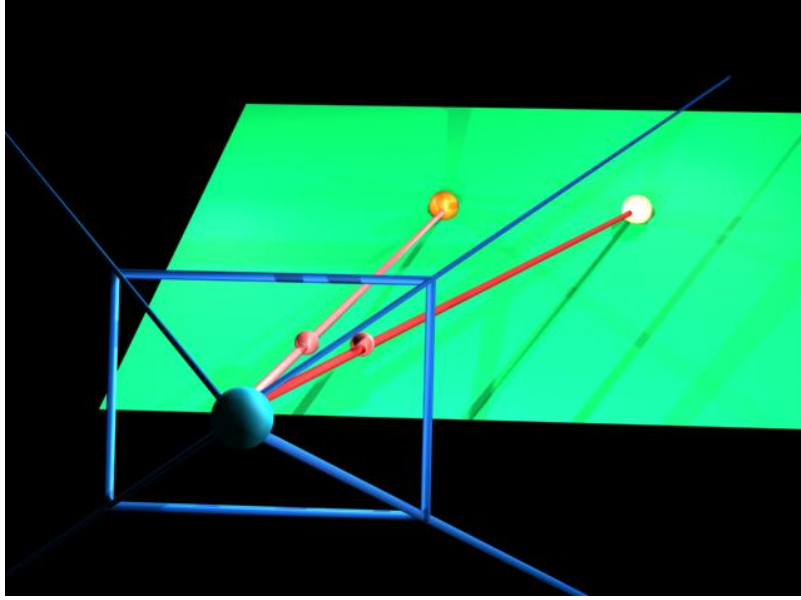


图 27 俯仰角导致激光扫描线及识别点变化示意图

根据图 26 右图的几何关系，设摄像机位置（等效透镜光心）为点 P，先关注两个标识点 O 和 A。设摄像机到 O，A 的三维距离分别为 R_0 ， R_1 ，O 点和 A 点成像像素点对应的屏幕射线(“Screen Ray”)为 $\overrightarrow{PO_{scr}}$ 和 $\overrightarrow{PA_{scr}}$ ，摄像机横向视角为 $\varphi_{横}$ ，纵向视角为 $\varphi_{竖}$ ，标识正方形的边长为 \mathcal{L} ，则可以知道：

$$\overrightarrow{PO_{scr}} = (x_{O-2D} \cdot \tan\left(\frac{\varphi_{横}}{2}\right), y_{O-2D} \cdot \tan\left(\frac{\varphi_{竖}}{2}\right), 1)$$

$$\overrightarrow{PA_{scr}} = (x_{A-2D} \cdot \tan\left(\frac{\varphi_{横}}{2}\right), y_{A-2D} \cdot \tan\left(\frac{\varphi_{竖}}{2}\right), 1)$$



图：“Screen Ray”

其中这些带有“2D”下标都为上一节讨论翻滚角校正里面提到的投影空间的坐标，取值范围是(-1,1)。上面给出了 O 和 A 到光心光线入射的反方向射线，但是还没有归一化。

则在没有校正俯仰角的 3D 空间中，射线 \overrightarrow{PO} 和射线 \overrightarrow{PA} 分别为

$$\overrightarrow{PO} = \frac{R_0}{|\overrightarrow{PO_{scr}}|} (x_{O-2D} \cdot \tan\left(\frac{\varphi_{横}}{2}\right), y_{O-2D} \cdot \tan\left(\frac{\varphi_{竖}}{2}\right), 1)$$

$$\overrightarrow{PA} = \frac{R_1}{|\overrightarrow{PA_{scr}}|} (x_{A-2D} \cdot \tan\left(\frac{\varphi_{横}}{2}\right), y_{A-2D} \cdot \tan\left(\frac{\varphi_{竖}}{2}\right), 1)$$

未旋转的标识线段 OA 垂直于 XZ 平面，此时 O 和 A 的 x 和 z 坐标均相等。绕 x 轴旋转后的 OA 在 XZ 平面上的投影是和 Z 轴平行的，即局部 3D 的 x 坐标不变，因为俯仰角旋转是绕 x 轴进行的。

这就意味着虽然 $\overrightarrow{PA} - \overrightarrow{PO}$ 的表达式一大串很复杂，但是 \overrightarrow{OA} 的 x 坐标为 0。

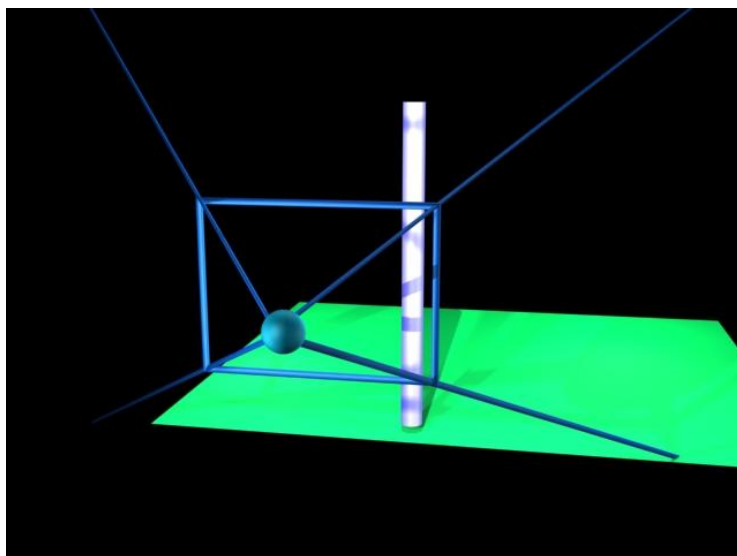


图: no yaw angle, no pitch angle, OA is perpendicular to the horizontal plane

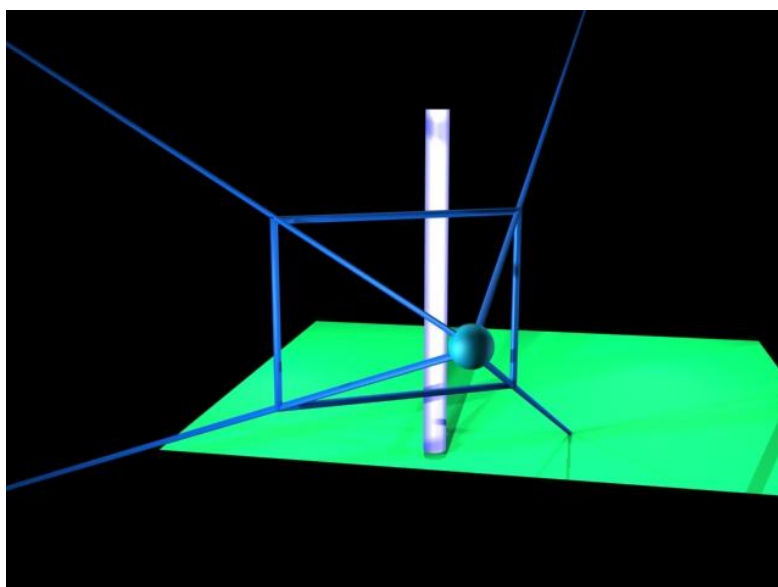


图: yaw angle exists, no pitch angle, OA is still perpendicular to the horizontal plane

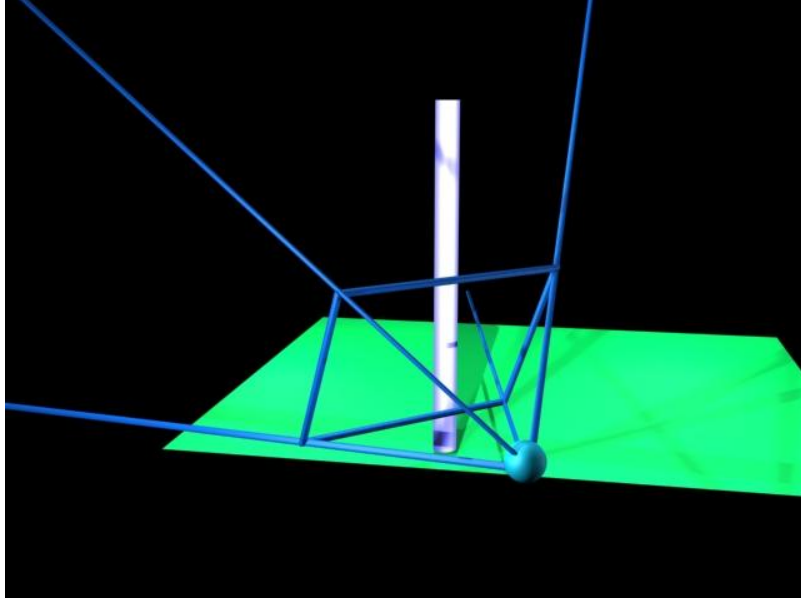


图: yaw angle exists, pitch angle exists

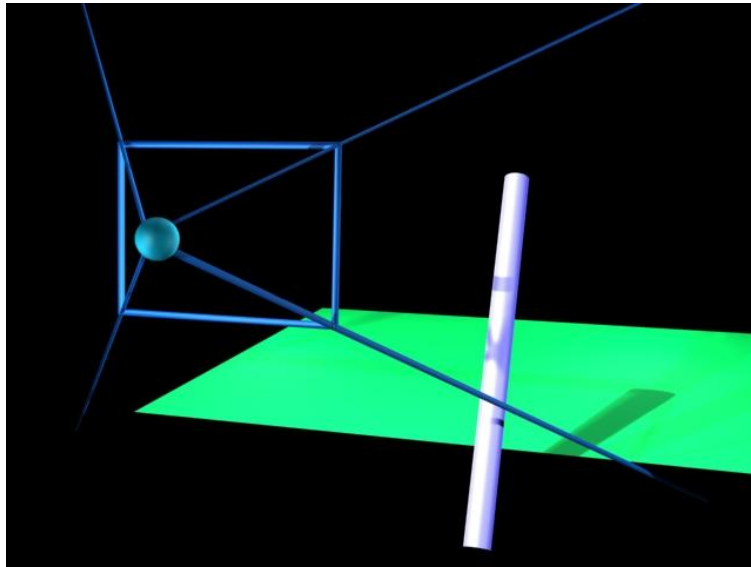


图: yaw angle exists, pitch angle exists, and this is the posture of OA in Local Space, which shows a fact that the X component of OA is equal to 0.

$$\text{记 } \mathcal{E} = |\overrightarrow{\text{PA}_{\text{scr}}}|$$

$$\mathcal{F} = |\overrightarrow{\text{PO}_{\text{scr}}}|$$

$$\mathcal{C} = \frac{y_{O-2D} \cdot \tan\left(\frac{\varphi_{\frac{y_{\text{scr}}}{2}}}{2}\right)}{|\overrightarrow{\text{PO}_{\text{scr}}}|}$$

$$\mathcal{D} = \frac{y_{a-2D} \cdot \tan\left(\frac{\varphi_{\text{竖}}}{2}\right)}{|\overrightarrow{\text{PA}}_{\text{scr}}|}$$

那 $\overrightarrow{\text{OA}}$ 便可以写为

$$\overrightarrow{\text{OA}} = (0, \mathcal{D}R_1 - \mathcal{C}R_0, \frac{R_1}{\mathcal{E}} - \frac{R_0}{\mathcal{F}})$$

根据本文对标识正方形的设计，边长 \mathcal{L} 是已知的。则可以列出方程①：

$$|\overrightarrow{\text{OA}}|^2 = (\mathcal{D}R_1 - \mathcal{C}R_0)^2 + \left(\frac{R_1}{\mathcal{E}} - \frac{R_0}{\mathcal{F}}\right)^2 = \mathcal{L}^2 \quad \text{①}$$

下面开始记 θ_x 为仰角 EulerX。

根据旋转后的 $\overrightarrow{\text{OA}}$ 和竖直轴夹角等于仰角的绝对值的关系，可以知道：

$$\frac{\overrightarrow{\text{OA}}}{|\overrightarrow{\text{OA}}|} \cdot \overrightarrow{(0,1,0)} = \cos \theta_x$$

$$\Rightarrow \frac{\mathcal{D}R_1 - \mathcal{C}R_0}{\mathcal{L}} = \cos \theta_x$$

$$\Rightarrow \mathcal{D}R_1 - \mathcal{C}R_0 = \mathcal{L} \cos \theta_x \quad \text{②}$$

仔细观察，可以发现②与①中一项一样，所以我们可以把②代入①得：

$$(\mathcal{L} \cos \theta_x)^2 + \left(\frac{R_1}{\mathcal{E}} - \frac{R_0}{\mathcal{F}}\right)^2 = \mathcal{L}^2$$

$$\Rightarrow \left(\frac{R_1}{\mathcal{E}} - \frac{R_0}{\mathcal{F}}\right)^2 = \mathcal{L}^2 (1 - \cos^2 \theta_x)$$

$$\Rightarrow \left(\frac{R_1}{\mathcal{E}} - \frac{R_0}{\mathcal{F}}\right)^2 = (\mathcal{L} \sin \theta_x)^2 \quad \text{③}$$

但是这里不能轻易的开方，还是要讨论一下的。当 θ_x 为正，即“抬头”时，看似在摄像机空间中 **A** 会更靠近观察者，但其实仰角和摄像机与标识点的相对方位都会影响 R_0 和 R_1 ，即摄像机位置到标识点 **O** 和 **A** 的三维距离。所以我们应该对上式左右的符号一致性进行讨论。

我们先讨论 $\frac{R_1}{\mathcal{E}} - \frac{R_0}{\mathcal{F}}$ 和 $\mathcal{L} \sin \theta_x$ 同号的情况：

则此时，

$$\frac{R_1}{\mathcal{E}} - \frac{R_0}{\mathcal{F}} = \mathcal{L} \sin \theta_x$$

$$\Rightarrow R_1 = \mathcal{E} \mathcal{L} \sin \theta_x + \frac{\mathcal{E}}{\mathcal{F}} R_0 \quad (4)$$

得到了这一个 R_0 和 R_1 的关系之后,就意味着我们再需要多一个低次方程就可以得到 R_0 和 R_1 的解了。

所以对于空间三角形 POA, 根据余弦定理可以知道:

$$\mathcal{L}^2 = R_0^2 + R_1^2 - 2R_0R_1 \cdot \cos \langle \overrightarrow{PO}, \overrightarrow{PA} \rangle$$

$$\Rightarrow \mathcal{L}^2 = R_0^2 + R_1^2 - 2R_0R_1 \cdot \frac{\overrightarrow{PA_{scr}} \cdot \overrightarrow{PO_{scr}}}{|\overrightarrow{PA_{scr}}| |\overrightarrow{PO_{scr}}|}$$

现在记,

$$\mathcal{K} = \cos \langle \overrightarrow{PO}, \overrightarrow{PA} \rangle = \frac{\overrightarrow{PA_{scr}} \cdot \overrightarrow{PO_{scr}}}{|\overrightarrow{PA_{scr}}| |\overrightarrow{PO_{scr}}|}$$

$$\mathcal{Q} = \mathcal{E} \mathcal{L} \sin \theta_x$$

$$\mathcal{P} = \frac{\mathcal{E}}{\mathcal{F}}$$

则把余弦定理的等式联立③式 R_1 和 R_0 的关系可以得到:

$$R_0^2 + (\mathcal{P}R_0 + \mathcal{Q})^2 - 2\mathcal{K}R_0(\mathcal{P}R_0 + \mathcal{Q}) - \mathcal{L}^2 = 0$$

$$\Rightarrow R_0^2 + \mathcal{P}^2 R_0^2 + 2\mathcal{P}\mathcal{Q}R_0 + \mathcal{Q}^2 - 2\mathcal{K}\mathcal{P}R_0^2 - 2\mathcal{K}\mathcal{Q}R_0 - \mathcal{L}^2 = 0$$

$$\Rightarrow (1 + \mathcal{P}^2 - 2\mathcal{K}\mathcal{P})R_0^2 + (2\mathcal{P}\mathcal{Q} - 2\mathcal{K}\mathcal{Q})R_0 + \mathcal{Q}^2 - \mathcal{L}^2 = 0 \quad (5)$$

然后设方程⑤的二次项、一次项系数和常数项为 \mathcal{A} , \mathcal{B} , \mathcal{C} , 则用求根公式可以得到:

$$\mathcal{A} = 1 + \mathcal{P}^2 - 2\mathcal{K}\mathcal{P}$$

$$\mathcal{B} = 2\mathcal{P}\mathcal{Q} - 2\mathcal{K}\mathcal{Q}$$

$$\mathcal{C} = \mathcal{Q}^2 - \mathcal{L}^2$$

$$\Rightarrow R_0 = \frac{-\mathcal{B} \pm \sqrt{\mathcal{B}^2 - 4\mathcal{A}\mathcal{C}}}{2\mathcal{A}}$$

而在刚才讨论 $\left(\frac{R_1}{\mathcal{E}} - \frac{R_0}{\mathcal{F}}\right)^2 = (\mathcal{L} \sin \theta_x)^2$ ③开方后的符号的时候, 设了两边的底数同号推得了

$$\begin{aligned} R_1 &= \mathcal{E}\mathcal{L} \sin\theta_x + \frac{\mathcal{E}}{\mathcal{F}} R_0 \quad (4) \\ &= \mathcal{P}R_0 + \mathcal{Q} \end{aligned}$$

则两边底数异号时，

$$\begin{aligned} \frac{R_1}{\mathcal{E}} - \frac{R_0}{\mathcal{F}} &= -\mathcal{L} \sin\theta_x \\ \Rightarrow R_1 &= \mathcal{P}R_0 - \mathcal{Q} \quad (6) \end{aligned}$$

可以看到含 \mathcal{Q} 的项只有 \mathcal{B} ，所以用换元法可以得到

$$R_0 = \frac{\mathcal{B} \pm \sqrt{\mathcal{B}^2 - 4\mathcal{A}\mathcal{C}}}{2\mathcal{A}}$$

在讨论完前面由于开方引起的解的讨论之后，现在就要来讨论根的取舍了。

$$(1 + \mathcal{P}^2 - 2\mathcal{K}\mathcal{P})R_0^2 + (2\mathcal{P}\mathcal{Q} - 2\mathcal{K}\mathcal{Q})R_0 + \mathcal{Q}^2 - \mathcal{L}^2 = 0 \quad (5)$$

可以见到，这个方程不仅代换的数多，而且系数的几何意义并不直观和明显，但是实验数据表明，当 $|\sin\theta_x|$ 较大的时候方程会出现没有实根的情况，有根的时候一般都是一正一负的实根。

$$\mathcal{A} = 1 + \mathcal{P}^2 - 2\mathcal{K}\mathcal{P}$$

$$\mathcal{B} = 2\mathcal{P}\mathcal{Q} - 2\mathcal{K}\mathcal{Q}$$

$$\mathcal{C} = \mathcal{Q}^2 - \mathcal{L}^2 = (|\overrightarrow{\text{PA}_{\text{scr}}}| \sin\theta_x - 1) \cdot \mathcal{L}^2$$

由韦达定理可以知道，

$$x_1 x_2 = \frac{\mathcal{C}}{\mathcal{A}}$$

其中 \mathcal{A} 为正数，因为

$$\mathcal{K} = \cos \langle \overrightarrow{\text{PO}}, \overrightarrow{\text{PA}} \rangle$$

$$\Rightarrow \mathcal{A} = 1 + \mathcal{P}^2 - 2\mathcal{K}\mathcal{P} > \mathcal{P}^2 - 2\mathcal{P} + 1 = (\mathcal{P} - 1)^2 \geq 0$$

而 \mathcal{C} 的符号就取决于 $(|\overrightarrow{\text{PA}_{\text{scr}}}| \sin\theta_x - 1)$ 的符号，但 $|\overrightarrow{\text{PA}_{\text{scr}}}| \sin\theta_x$ 并没有直接的联系，因为 $|\overrightarrow{\text{PA}_{\text{scr}}}|$ 不仅受 $\sin\theta_x$ 影响，还受相机位置的影响，所以 \mathcal{C} 可能是正的

$$\Rightarrow x_1 x_2 = \frac{\mathcal{C}}{\mathcal{A}} \text{可能是正的。}$$

所以方程是有两根之积为正的风险，但这这个时候方程可能解出虚根，所以解决办法就是仰角不要太大，然后有实数解时就取正根：

$$R_0 = \frac{-B + \sqrt{B^2 - 4AC}}{2A}$$

同理，设摄像机位置 P 到标识点 B 的距离为 R_2 ，到标识点 C 的距离为 R_3 ，则可以用与上文同样的方式计算出 R_2 。

然后就 B 和 O 的局部 3D 坐标：

$$\overrightarrow{PO'} = (x_{O'}, y_{O'}, z_{O'}) = \frac{R_0}{|\overrightarrow{PO_{scr}}|} (x_{O-2D} \cdot \tan\left(\frac{\varphi_{横}}{2}\right), y_{O-2D} \cdot \tan\left(\frac{\varphi_{竖}}{2}\right), 1)$$

$$\overrightarrow{PB'} = (x_{B'}, y_{B'}, z_{B'}) = \frac{R_2}{|\overrightarrow{PB_{scr}}|} (x_{B-2D} \cdot \tan\left(\frac{\varphi_{横}}{2}\right), y_{B-2D} \cdot \tan\left(\frac{\varphi_{竖}}{2}\right), 1)$$

然后，把 O 点和 B 点绕 X 轴旋转一个仰角，预测出 O, B 在没有俯仰角时候的局部坐标，得到

$$\overrightarrow{PO} = (x_O, y_O, z_O)$$

$$\overrightarrow{PB} = (x_B, y_B, z_B)$$

此时 \overrightarrow{OB} 平行于水平面，到这里就去除了翻滚角和仰角的影响。

在这里标识点 O 和 B 组成 \overrightarrow{OB} 是世界坐标系 X 轴， O 是世界坐标系原点。

3.5.5 去除了仰角、翻滚角之后计算世界坐标、水平角

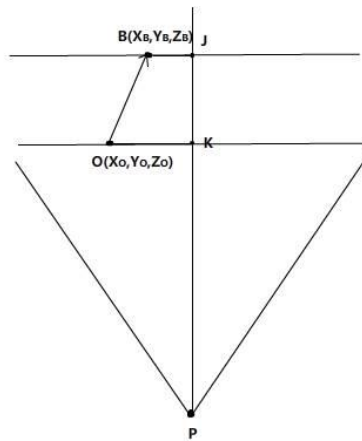


图 28 局部坐标向世界坐标转化俯视图

把上文的处在摄像机位置的坐标原点，对齐到以标识点 O 为原点的坐标系。平移之后，camera 在当前校正空间的坐标可以由下式给出：

$$\overrightarrow{Cam_{adjust}} = -(\overrightarrow{x_0, y_0, z_0})$$

这里的坐标是以标识原点 O 为原点的坐标下的坐标。

至此，相机的在处理水平角之前的位置坐标已求出。现在有足够的信息去求相机的水平角，而根据标识点的定义， \overrightarrow{OB} 为世界坐标系的 x 轴，则只需把 \overrightarrow{OB} 对齐到 世界坐标系 x 轴的话，相机的水平角和世界位置也就随之得到了。

可以知道，

$$\overrightarrow{OB} = (\overrightarrow{x_B - x_0, y_B - y_0, z_B - z_0})$$

如果在实际操作中没有太大的误差，O 和 B 的 Y 坐标应该是大致相等的， \overrightarrow{OB} Y 坐标为 0.

得出相机水平角，需要从局部空间转到世界空间中。因为这里的 \overrightarrow{OB} 是用局部坐标来表达，所以用这个空间中以 \overrightarrow{OB} 为其中一个分向量的正交基写的矩阵（跟上一节散乱点云的转换矩阵的“正交基”的构建方法一致）是从世界空间转到局部空间。所以需要下面的矩阵 A 左乘当前空间的相机向量 $(0, 0, 1)^T$ 来把摄像机向量转到世界空间：

$$A = \begin{bmatrix} \frac{X_B}{\sqrt{d}} & 0 & \frac{-Z_B}{\sqrt{d}} \\ 0 & 1 & 0 \\ \frac{Z_B}{\sqrt{d}} & 0 & \frac{X_B}{\sqrt{d}} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{X_B}{\sqrt{d}} & 0 & \frac{Z_B}{\sqrt{d}} \\ 0 & 1 & 0 \\ \frac{-Z_B}{\sqrt{d}} & 0 & \frac{X_B}{\sqrt{d}} \end{bmatrix}$$

其中 $d = (X_B)^2 + (Y_B)^2 + (Z_B)^2 = (X_B)^2 + (Z_B)^2$ ，即 \overrightarrow{OB} 的模。

而且正交矩阵的逆等于它的转置。

则 camera 的世界坐标可以由下式给出：

$$\overrightarrow{Cam_{world}} = A \cdot \overrightarrow{Cam_{adjust}}$$

水平角（EulerY）可以这样来获取：

先计算上述空间中摄像机向量在 xz 平面的投影对齐到世界空间后的向量 \vec{v} ，

$$\vec{v} = A \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{Z_B}{\sqrt{d}} \\ 0 \\ \frac{X_B}{\sqrt{d}} \end{bmatrix}$$

其中 A 里面是一个标准正交向量组，所以 $|\vec{v}| = 1$ 。

此时的 \vec{v} 就已经是用世界坐标表达的了。所以就可以得到在 xz 平面上，由于水平角和相机向量和世界 z 轴的夹角，（叉乘符合左手定则）

$$\cos(Y) = \vec{v} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \sin(Y) = \left(\vec{v} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \text{的} y \text{坐标}$$

（上述的 $\left(\vec{v} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right)$ 的 x, z 分量化简后均为 0）

水平角 θ_Y 由下式给出：

$$\begin{cases} \text{当} \sin(Y) > 0 \text{ 时, } \arccos(\cos(\theta_Y)) \\ \text{当} \sin(Y) \leq 0 \text{ 时, } 2\pi - \arccos(\cos(\theta_Y)) \end{cases}$$

至此，通过分析标识点得到了部分的摄像机刚体方位信息，足够进行坐标变换。用这样的光学成像分析，可以免去机械臂、加速度传感器之类的误差累积的物理仪器，从而实现较高准确度和精度的坐标变换。

下图 29、图 30、图 31、图 32、图 33 为不同图形在 320*240 分辨率下三维计算机模拟图片。

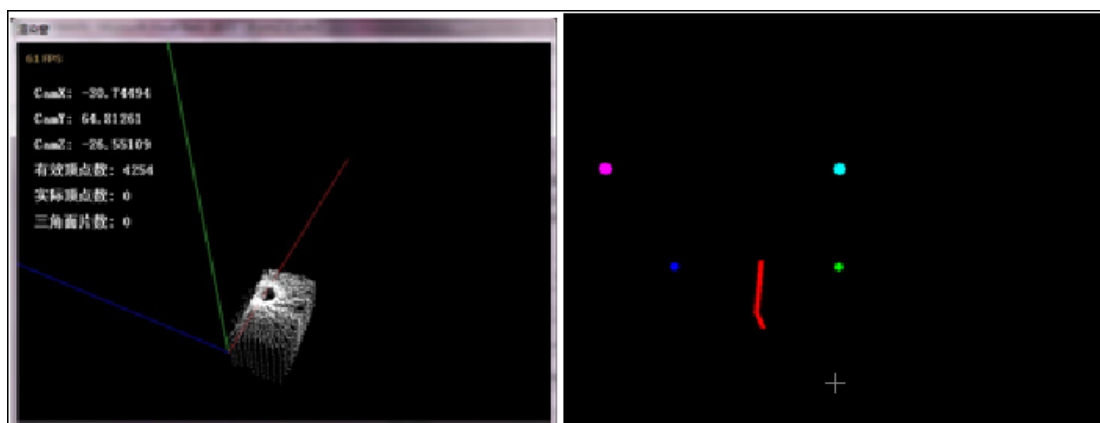


图 29 320x240 分辨率长方体，无翻滚角

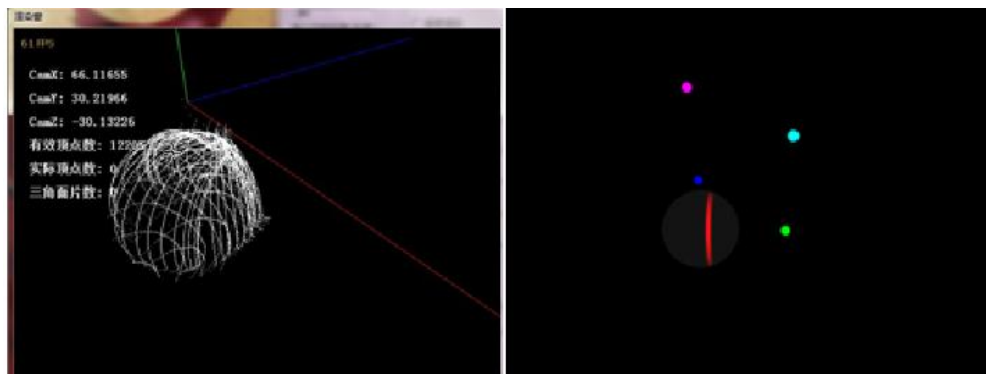


图 30 320x240 分辨率的球，无翻滚角

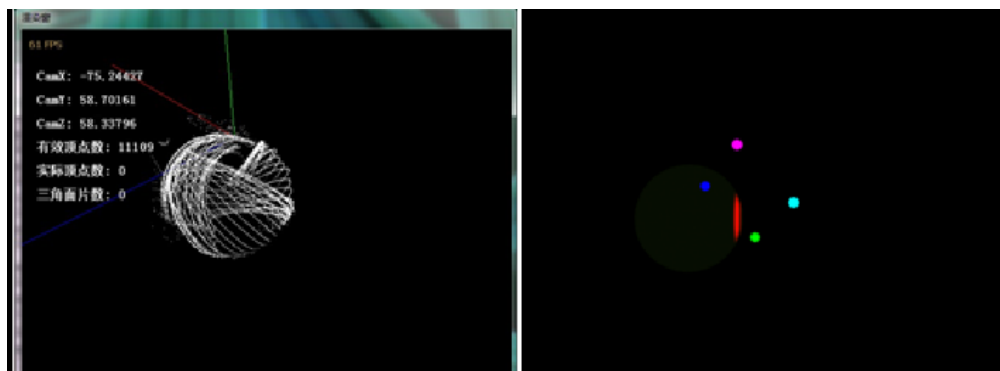


图 31 320x240 分辨率的球，有翻滚角

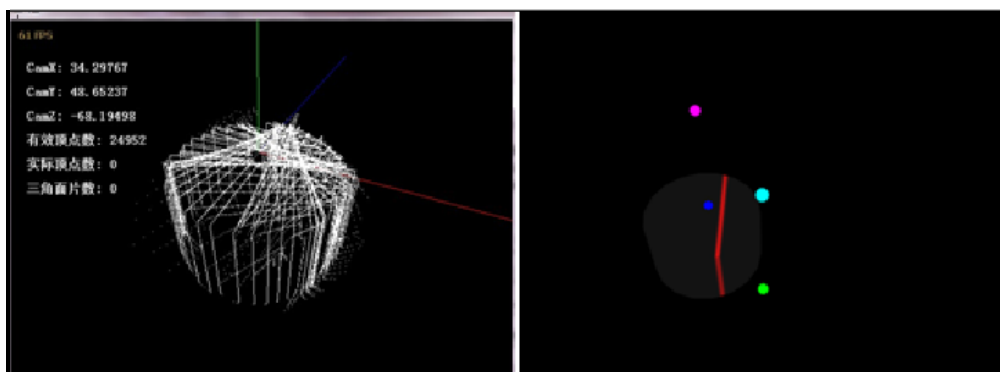


图 32 320x240 分辨率的圆柱体，无翻滚角

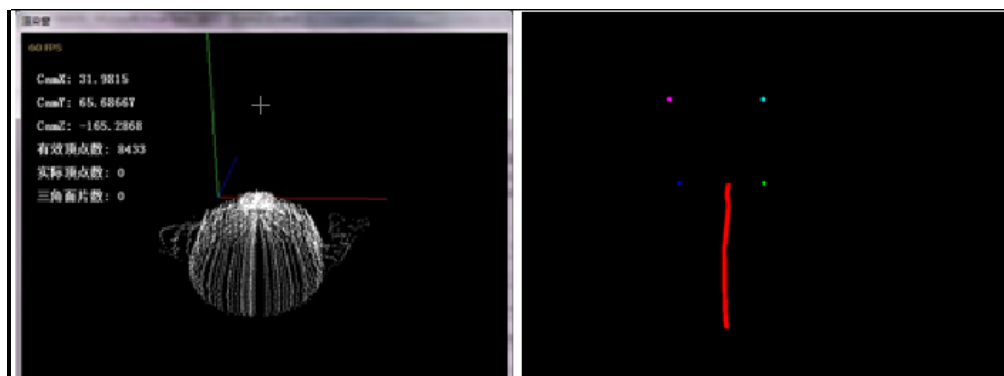


图 33 320x240 分辨率的茶壶，无翻滚角

从效果图中噪点多，可能由以下原因造成：

- 1、图片分辨率(resolution)低（只是因为实验时为了调试更有效率）
- 2、采样算法不够合理
- 3、物体放置不好有时会遮挡标识点

点云稀疏只是因为照片不够，扫描部位不够全面，每个物体只模拟了几十张。

3.6 网格重构

3.6.1 – 国内外的研究

得到了点云以后并不能直接使用，还需要对点云在最大限度保持其拓补结构的情况下进行网格重构，得到以三角形为图元的多面体网格。但是面对不同的点云，要处理起来并不容易，因为要考虑到点云的均匀性，密度，噪点等。

比较流行的方法是用大量三角形逼近曲面对三维集合图形进行一种近似而较为精确的描述，这也是流行 3D 建模软件的一种通用方法，二维已有比较成熟的重构方法。其中有一个法则是 Delaunay 法则，要求 Delaunay 三角形的外接圆内没有第四个点。这样的三角形质量较好，因为这个法则遵循了最小角最大化的原则。在三维散乱点云的重构中，一般使用的方法都分为两个大方向，一个是在凸包(体)上进行“雕刻”(Sculpture-Based)，一种是区域生长法(Region-Growing)。国外已有一些先进、快速、稳定的重构算法，雕刻法有 a-shape（删除外接球半径大于给定阈值的四面体），生长法如 Ball-pivoting，移动最小二乘法（MLS），Marching Cubes 等。

本文在这个主题上不进行深入的研究，因为这是一个很庞大的主题，只提出作者对于网格重构的原创想法与思考。

本文提出一种快速、简洁的算法，针对曲率变化小的点云极其有效。

由于二维平面上的三角形剖分已经有了比较成熟的算法，所以我们要想办法把三维点云降维剖分。只要能找到一种大概保持邻接关系的降维映射就可以了，因为我们只需要找出点与点之间的邻接信息和三角形信息。虽然降维总是面临着信息丢失的风险，但是这样做是简单的，快捷的，近似的。

联想到地球和世界地图，我们可以想到一种降维的办法，把三维点云投影到单位球上，并将球面展平。

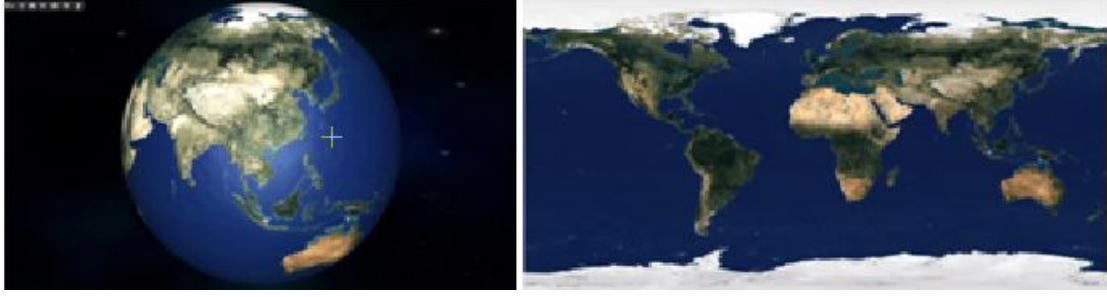


图 34 地球球体表面展开为平面地图

这里的展平不是刚体的展平，刚体球面不可能展平为平面，所以这个过程必定带有扭曲，所以用映射会比较合适。映射的办法可以是这样：直接取投影后三维点的球坐标的两个角度分量，直接映射到 XY 平面直角坐标系上，相当于把**经纬度**看作直角坐标系上的二维点。这样的映射，能在投影到单位球没有重叠顶点的情况下，确定了一张能保持原有几何邻接信息的二维点云，继而重构成二维网格，再映射回三维网格。

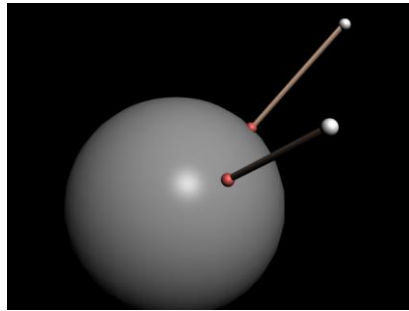


图 35 把三维点（白色），投影到单位球上，交点为图中红色点

若有一个点 $A(x, y, z)$ ，要将其投影到单位球上很容易，只需把 P 向量归一化。即：

$$A_{proj} = \left(\frac{x}{\sqrt{x^2 + y^2 + z^2}}, \frac{y}{\sqrt{x^2 + y^2 + z^2}}, \frac{z}{\sqrt{x^2 + y^2 + z^2}} \right) = (x_p, y_p, z_p)$$

把新向量的球面坐标记为 $A_{proj} = (u, v, w)$ ，则很明显：

$$w = 1$$

然后可以求得

$$u = \text{sgn}(z_p) \cdot \arccos(x_p)$$

$$v = \arcsin(y_p)$$

($\text{sgn}(x)$ 为取符号函数)

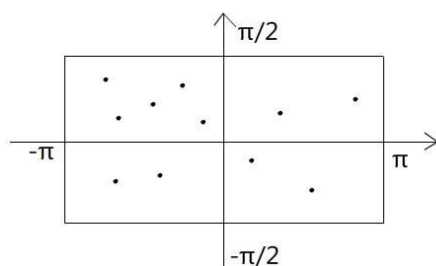


图 36 三维点向平面展开示意图

然后^[4]，可以开始进行平面上的 Delaunay 三角化。

1:以点集中一点 P 为起始点

2:找到与 P 最近的点 N 组成第一条基线 e

3:在邻近的点（距离小于一定阈值）中找到与基线 e 组成 Delaunay 三角形的顶点，即 e 的两个顶点和新顶点组成三角形的外接圆内没有第四个顶点。

4:上述新顶点与基线连成新的边，如果这条边之前是不存在的，那就新增这条边到“待处理基线队列”中。

5:重复 Step3 和 Step4 直至所有基线被处理完。

6:在上面边邻接信息的基础上，利用图的广度优先遍历找出所有三角形。

如果下面的行列式^[2]，大于 0，则可以判断第四点 D 在三角形 ABC 的外接圆内。这个在第三步可以用到。

$$\begin{vmatrix} A_x & A_y & A_x^2 + A_y^2 & 1 \\ B_x & B_y & B_x^2 + B_y^2 & 1 \\ C_x & C_y & C_x^2 + C_y^2 & 1 \\ D_x & D_y & D_x^2 + D_y^2 & 1 \end{vmatrix} = \begin{vmatrix} A_x - D_x & A_y - D_y & (A_x^2 - D_x^2) + (A_y^2 - D_y^2) \\ B_x - D_x & B_y - D_y & (B_x^2 - D_x^2) + (B_y^2 - D_y^2) \\ C_x - D_x & C_y - D_y & (C_x^2 - D_x^2) + (C_y^2 - D_y^2) \end{vmatrix} > 0$$

重构完这张网了还不能直接地映射回三维空间，因为要使这样重构的三维点云闭合，还有考虑这张二维网格的边缘——映射回去的话还没有使三维网格闭合。所以应该把二维网格的边缘上的顶点单独抽出来进行一次三角化。

然后邻接信息都重构好了，就可以把这些邻接边、三角形信息直接用在原来的三维点云上了。

这种算法虽然不能很好的处理复杂结构的点云，但是在平面上重构的话物体贴图非常方便，每个顶点可以直接赋予纹理坐标之后再映射回三维点云，这样就使得贴图变得非常的方便。

4. 数学模型的验证

4.1 - 3DMAX 模拟图

在原型机尚未搭建完成,或者说误差和理论还没有一个清晰的思路时,再考虑到时间人力和研究环境,一开始应该用**理想图片**来进行采样。可以这样做,用电脑的 3D 模拟技术摆好一个理想的模型并渲染,然后再用程序去处理,这样可以大大节省实验阶段的时间,因为一开始的理论并不一定是对的,出现一个错误的结果之后很难判断是因为实验误差过大造成的,还是算法本身有错误。所以如果一开始用理想的采样图片的话,那就可以只关注算法。经过比较,3DMAX 是一款成熟的建模和做 3D 动画的主流软件,模型以三角面片为图元,提供多元化和强大的处理,所以选择 3DMAX 作为生成模拟图片的软件。在渲染一系列图片后并按程序要求的设置好数据,就能开始实验了。下图图 37,是一个球进行模拟扫描时的模型三视图及扫描线。

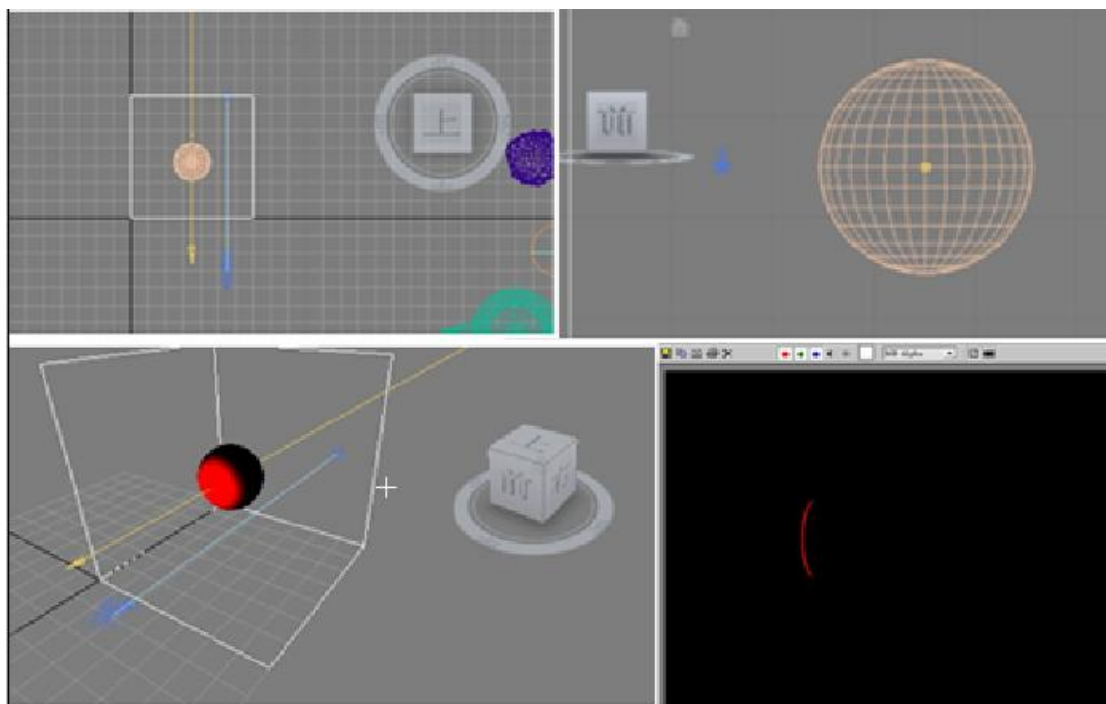


图 37 在 3DMAX 构建虚拟激光扫描过程示意图

4.2 - 计算视角

我们在实现拍照的时候用的是佳能 EOS 1100D 的单反照相机。虽然镜头焦距范围只是 18mm - 55mm,但是单反相对于工业摄像头的优势是光学变焦,这意味着变焦之后分辨率不会降低,不像数码相机的数码变焦,只是做了等效的画面裁剪(这就降低了分辨率)。为了充分发挥单反的优势,我们决定做**可变焦**(相当于实现了变距)的扫描方式。但一定要先解决了镜头在**确定焦距**的条件下理论模型与实际的匹配。

单反的镜头有多组镜片,我们手头上的镜头里面是有十多组透镜的,里面的折射是极

其复杂的，去研究所有细节是没有意义的。我们还关注镜头上标的“18mm-55mm”指的是哪个透镜的焦距，究竟是第一组镜片还是最后一组镜片的焦距，还是纯粹就是这个镜头的等效薄透镜的焦距。因为佳能的单反也是一个商业相机，不可能也没必要公开太多技术参数，所以我们猜测镜头上标的焦距是等效透镜的焦距。

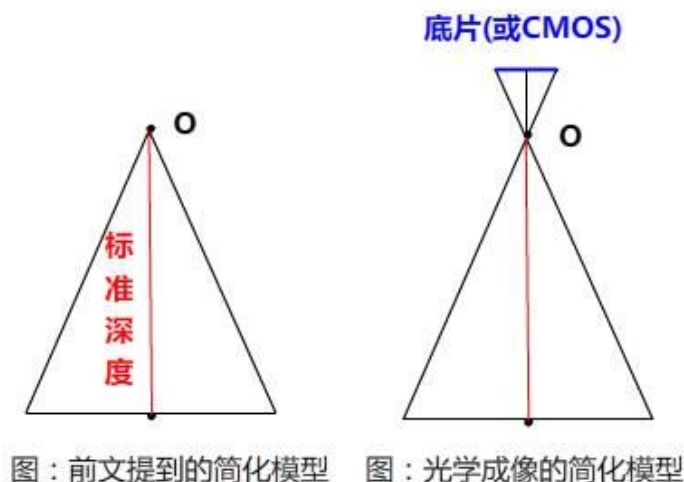


图 38 光学成像简化模型示意图

经过分析，发现 O 点是单反镜头等效薄透镜的光心（根据薄透镜的性质，通过光心的光线方向不改变）。而且如果上一句话真的成立，则“标准深度”就是光心到背景板（可能是虚构的）的距离，也就是物距了。如果能根据这个模型解出所需的参数，要得到标准深度和其他参数就不用靠“目测”与“大致”了（这是我们在实验初期还没有研究得够深时的做法）。

对于一个型号的相机来说 COMS 的尺寸是确定的，而根据等效焦距与 135 胶片画幅等可以算得，1100D 的 CMOS 尺寸是 $\frac{36}{1.6}\text{mm} \times \frac{24}{1.6}\text{mm}$ ，其中 36mm 与 24mm 是 135 胶片的底片大小，称为全画幅。1.6 是这部相机的常量——等效焦距转换常数。焦距转换率是摄影中的一个概念，它在不同底片格式中相同视角对应的不同焦距的建立对应关系。

根据维基百科的说法^[9]，有效焦距与影像格式尺寸足以定义视角。对于直线投射影像，视角 (α) 可以由被选择的大小 (d)，以及有效焦距 (f) 计算出来如下：

$$\alpha = 2 \arctan \frac{d}{2f}$$

d 表示底片 (或感光元件) 的大小再一个方向的计算。譬如，对于 36mm 宽的底片， $d = 36 \text{ mm}$ 可以作为参数取得水平视角

一般我们说：焦距就是透镜中心到焦点的距离。但这仅仅是单片薄透镜的情

况，由于照相机的镜头都是由许多片透镜组合而成的，因此，情况远不是那么简单。

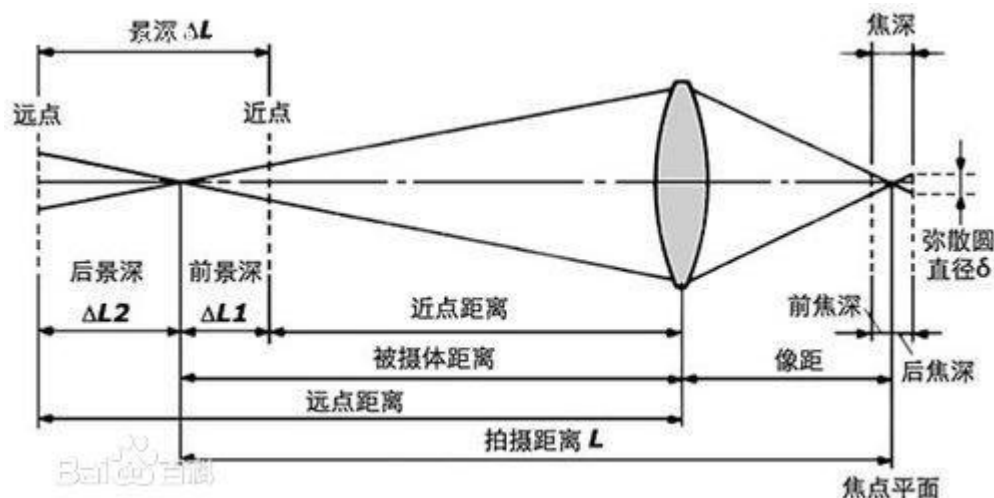


图 39 透镜景深造成视角计算误差光路示意图

所以才造成上述相机的视角计算公式和普通薄透镜的公式有点出入。

4.3 - 前版本原型机的截图（变焦、无手持、固定）



图 40 可变焦距 3d 扫描装置实物图

4.4 - 仍待解决的问题

在理论模型与实际的匹配中，还有一件重要的事要做就是去噪。因为在实际使用中一般都会在开阔的空间中实验，并且图片中背景的未知颜色可能会造成干扰。而且现在由于激光的能量太高，导致的很多反射也对结果有影响，一般是坐标被扭曲、生成了无用的噪波顶点。这些问题要从物理与算法的角度去解决，例如降低激光强度、给背景加限制。而且还有一些是真实的准确的点而用户并不需要的（例如桌子上摆了个瓶子而只需要瓶子的模型），这些

要人工去除。在图片采样上的算法也要改进。

在标识点一章，在解方程时有时会因为仰角太大而解不出实根，这个要在理论上进一步改进。

在网格重构一章，拓补结构复杂、表面曲率变化大的物体不能正确重构。

还有就是物理实现造成的误差。这个在手工实现的时候只能尽可能的去避免，在工业生产中应该可以避免。

在生成 3D 模型之后，可以用 AABB(Axis Aligned Bounding Box)去进行做阈值筛选明显的噪点，再用 2D 逆投影到 3D 的 Picking 射线让用户选择要去除的点。

(*注：picking 射线是 3D 游戏中常用的技术，渲染屏幕上的一个 2D 点可以对应一条 3D 世界空间中的射线，其推导用了更多的 3D 游戏渲染管道的基本变换的元素，在此不予赘述)。

上述问题解决后，从扫描到打印前期的流程就算基本完成了。

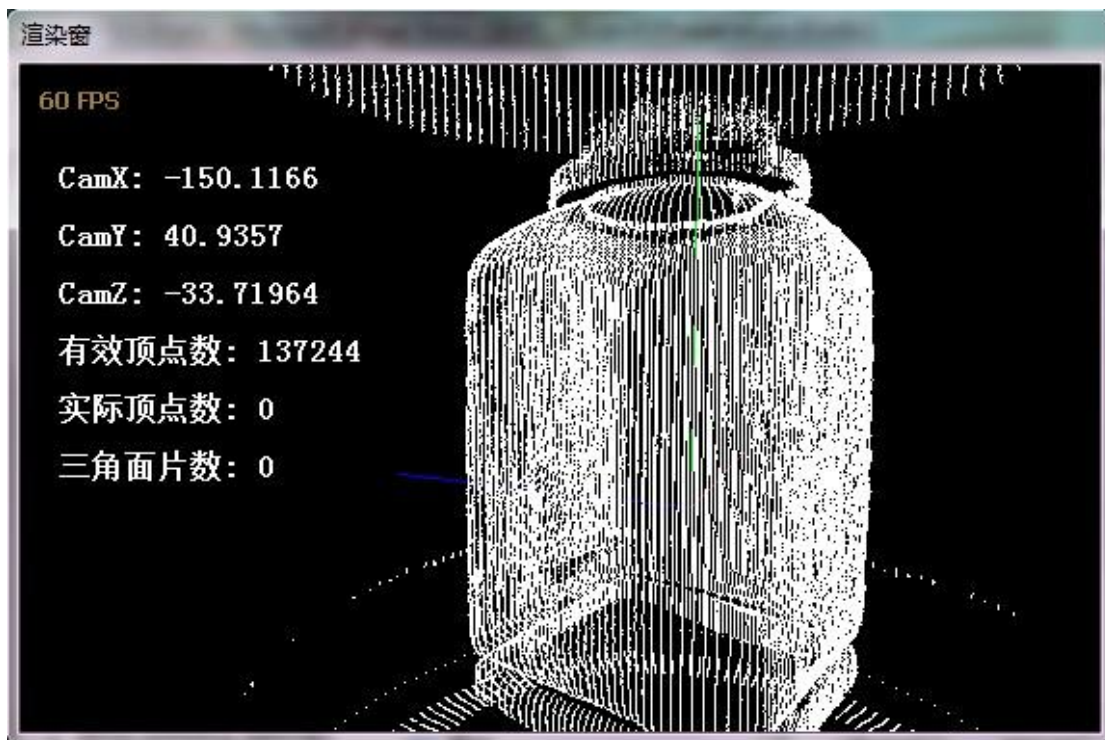


图 41 使用上述装置实际扫描实物的点云显示图

5 - 结论

本文在讨论点坐标升维的时候引出一思考。所有的投影都是一个降维的过程，这就意味着信息的丢失，当然也不可能凭空恢复了。高维向量的信息量比低维向量的信息量多，这是一个事实，所以从 2D 图片生成 3D 顶点必须要有足够的信息量才能实现升维。而这些多出来的信息一般都是限制条件。

本文通过对透视成像的特点进行分析，结合设计上的限制条件，实现了从图片获取 3D 点的局部坐标，从而能继续进行变换合并到世界坐标系。

而可以做到相机在任意位置和角度生成散乱点云的算法已经实现，而且使用了标识正方形（一个固定的框架而不是传统的贴点）来进行光学定位，高精度而且物理实现方便，扫描

仪可以做到无线、手持，并且经过对相机成像特点进行分析，相机可以做到变焦。但仍需解决图片去噪和采样算法优化的问题。

后记

做这个项目加起来都有一年多多了，过程真的是非常艰辛，投入成百上千个小时。为了这个项目在时间和成绩上还是有点牺牲的，最近才恢复了点元气= =。研究的时候在得不到正确的结果时真的是万分纠结，因为根本不知道是算法、还是编程、还是实验数据、还是实验参数出了问题，可能是这里漏了个符号，那里居然没有用临时变量，局部 Y 坐标公式错了 5 次，标识点处理仰角的竖直边居然不平行于屏幕竖直边，维基的欧拉角矩阵看得一头雾水，变换不对时又不知道是变换错了还是标识点识别错了或者是计算错了，实验时设定数据想当然想到一个以为和真实情况等价的结构但是却并不等价.....上面有不少让我单一变量的猜错排错从算法、实验数据、编程、输入参数上去分析，纠结，标识点的问题纠结的最久，居然是实验参数的错误。真是感觉到做探索不容易，平时做证明题都不容易，现在可是在做探索题，简直不是同一个级别的难度。之前我是一直在搞 3D 游戏的，中间停过一段时间想做 3D 引擎，但是知识不够就在那啃线性代数，**Real Time Rendering** 之类的书。结果再返回来做这个项目，不仅对计算机图形学的一些原理更清晰了，项目也做得更顺手了。还是感觉要学的东西太多了，一个领域都能耗尽一个人短暂的一生。这个项目，就当做是锻炼了，没时间搞这些了啦--还要高考啊 T_T。。。感谢指导老师老梁，感谢一直支持我的同学们，谢谢~

参考文献

- [1] 百度百科 - 3D 扫描仪
- [2] 维基百科 - Delaunay Triangulation——
http://en.wikipedia.org/wiki/Delaunay_triangulation
- [3] 百度文库 - 三角测距法
<http://wenku.baidu.com/link?url=7YFCJ1b3SSNrkyXL1-uxw0djNNcGTFTCLrEW0V1ezl15x48sro5J3lzd1k80b09lvm-LycInVCUWNHFRwaYY4SCKfwtMPmqIzfRcvulC457>
- [4] 赵平建 - 基于三维散乱点云的曲面重构技术
<http://www.doc88.com/p-779890339350.html>
- [5] 百度百科 - 向量积
http://baike.baidu.com/link?url=WkujNcgUey9t5WHZY20yHXyzaN2IfTPtSqA5rh386mkoUiTYrZLyAF_gjnJH_3n3hXY2agWZlooh1TKEXbZwKK
- [6] 维基百科 - Delaunay Triangulation -
http://en.wikipedia.org/wiki/Delaunay_triangulation
- [7] Real Time Rendering (Third Edition)
- [8] Introduction To 3D Game Programming with DirectX 11-Frank Luna
- [9] 维基百科 - 视角
<http://zh.wikipedia.org/wiki/%E8%A6%96%E8%A7%92%E5.A4.96.E9.83.A8.E9.80.A3.E7.B5.90>
- [10] 维基百科 - 焦距转换率
<http://zh.wikipedia.org/wiki/%E7%84%A6%E8%B7%9D%E8%BD%AC%E6%8D%A2%E7%8E%87>
- [11] 百度百科 - 焦距
<http://baike.baidu.com/link?url=8962C77mVJowgrcP3eFUouXzGFra0DMPLBrIVhCEterUqcSoAm-w6T3p9XG5LIz3>