



城市空间建模与仿真

第三讲 结构光与TOF三维重建与数据处理

任课教师：汤圣君

建筑与城市规划学院 城市空间信息工程系

其中部分图片来自互联网和同行专家

目录

CONTENTS

01 3D光学测量方法概述

02 结构光3D成像原理

03 TOF 3D成像原理

04 图像畸变与矫正

05 视觉与深度数据融合

目录

CONTENTS

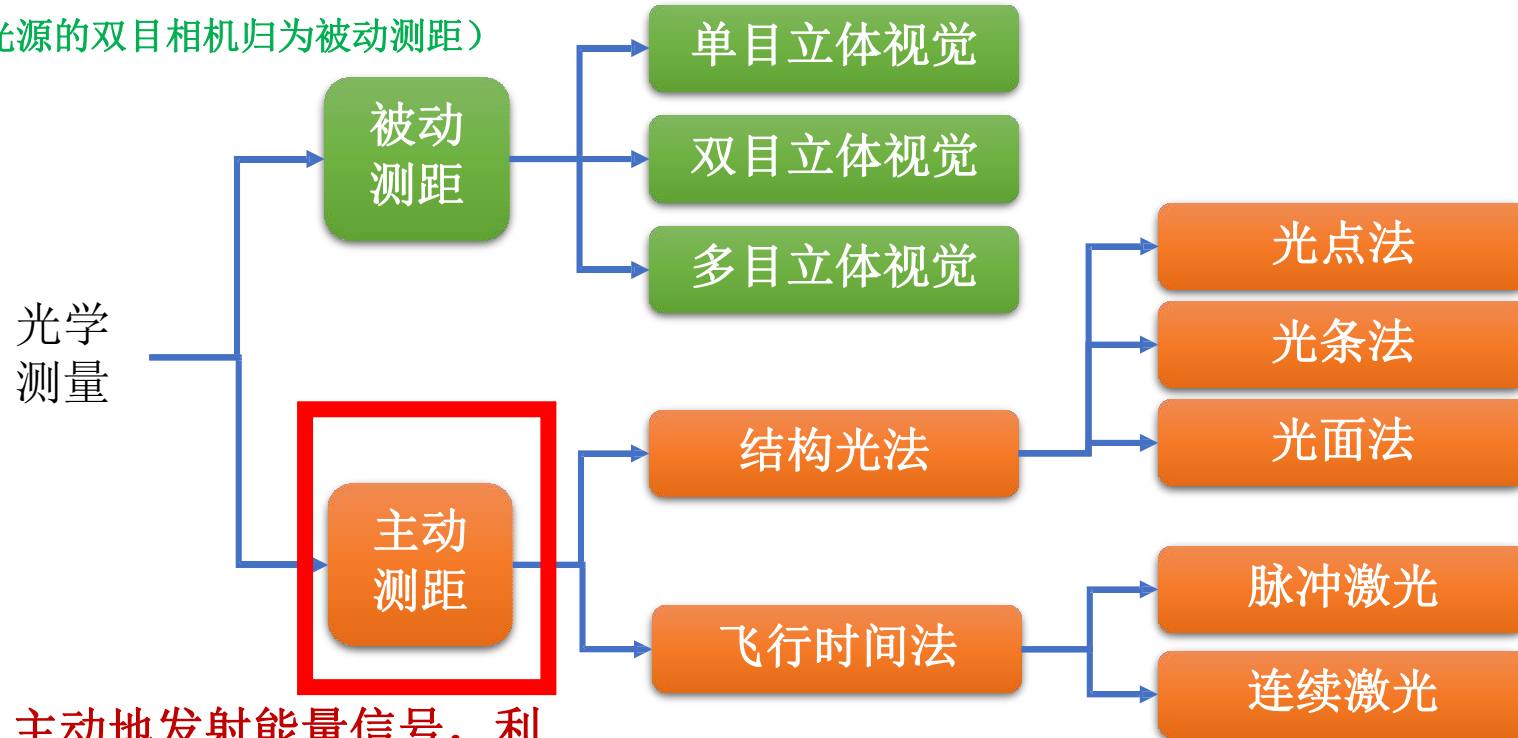
01

3D光学测量方法概述

3D光学测量方法概述

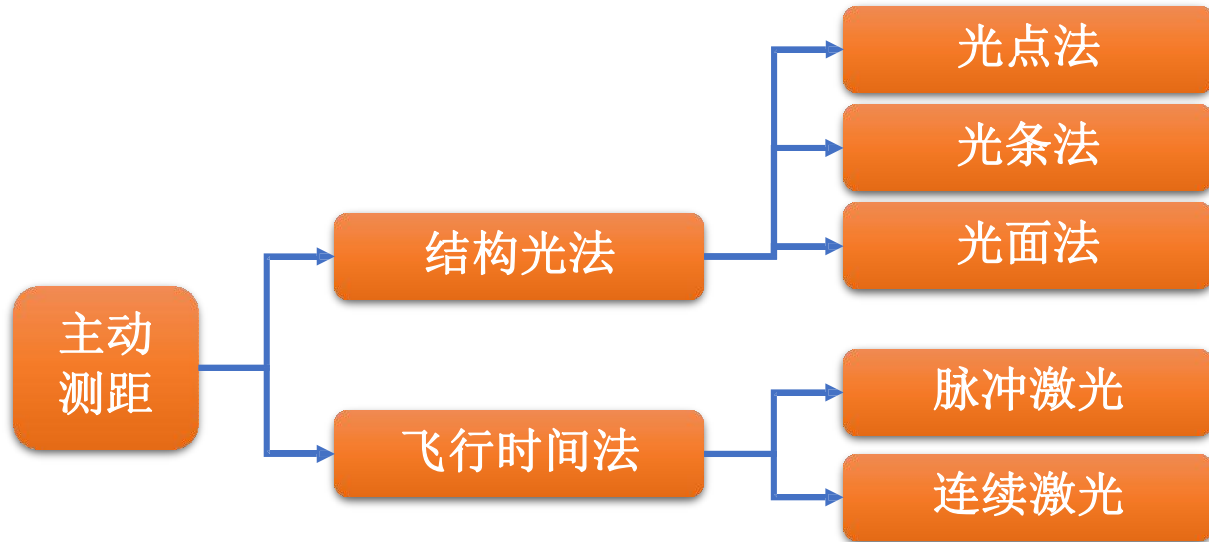
利用自然光在物体表面反射形成的2D图片进行3D信息的重建

(带有补光光源的双目相机归为被动测距)



人为地、主动地发射能量信号，利用物体表面反射特性及信号的传播特性实现对物体的3D信息的测量

3D光学测量方法概述



结构光法

投射特定纹理光线到被测物体，通过相机拍摄物体表面的投影纹理，经过计算得到距离。

飞行时间（ToF）法



将调制光信号发射到物体表面，通过测量接收反射信号与发射信号的时间差计算物体表面到相机的距离。基于以上两种原理的相机目前已经成为消费级产品

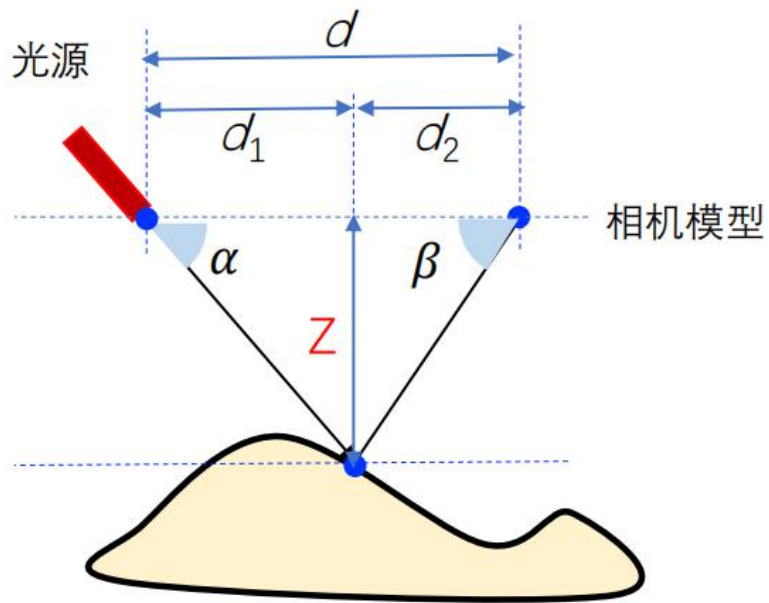
目录

CONTENTS

02

结构光3D成像原理

结构光3D成像原理



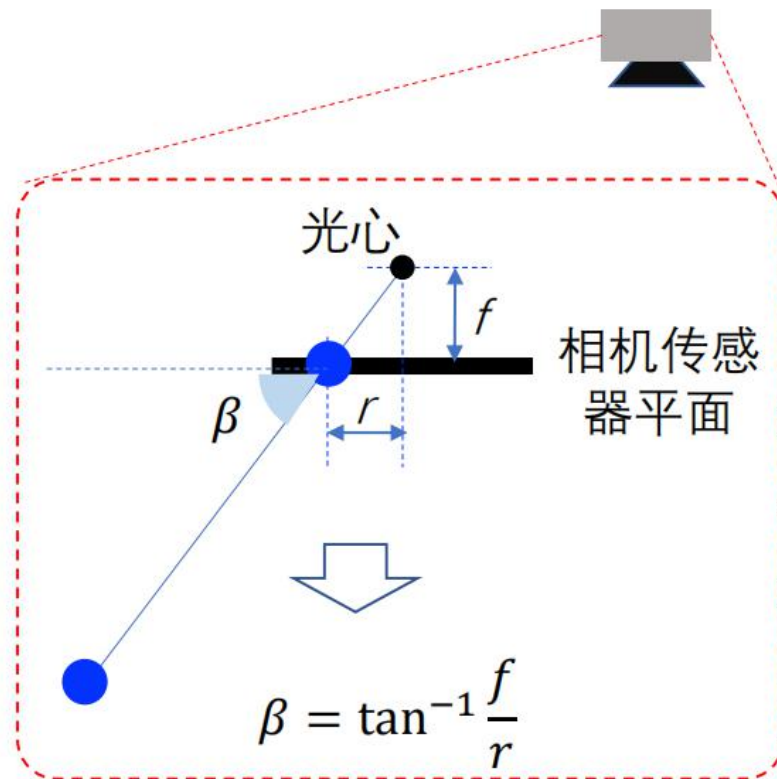
$$\frac{Z}{\tan \alpha} + \frac{Z}{\tan \beta} = d$$

$$Z = d \frac{\sin \alpha \sin \beta}{\sin(\alpha + \beta)}$$

物体表面光点到相机传感器平面的距离

😞 β 如何得到

😊 从图像中光点位置x计算 β

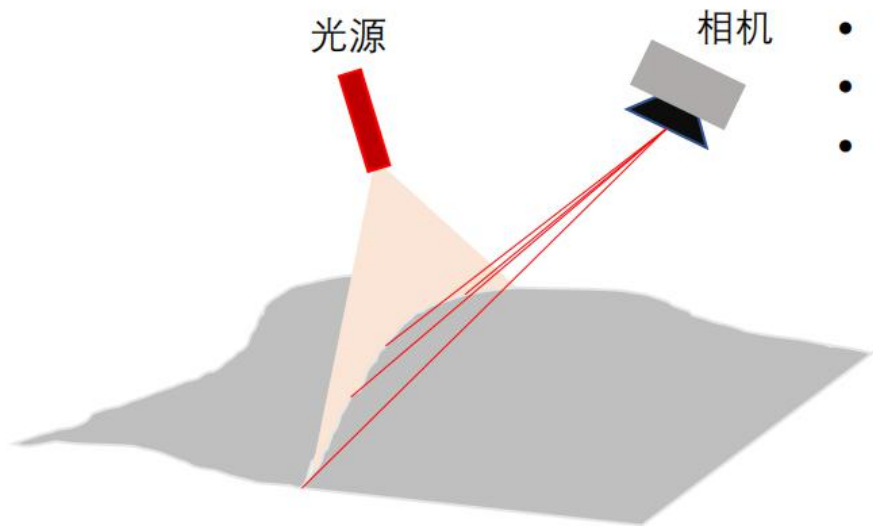


$$\beta = \tan^{-1} \frac{f}{r}$$

😊 通过光源转动进行光点扫描就能够计算出物体表面各个位置的Z值

点光源

结构光3D成像原理

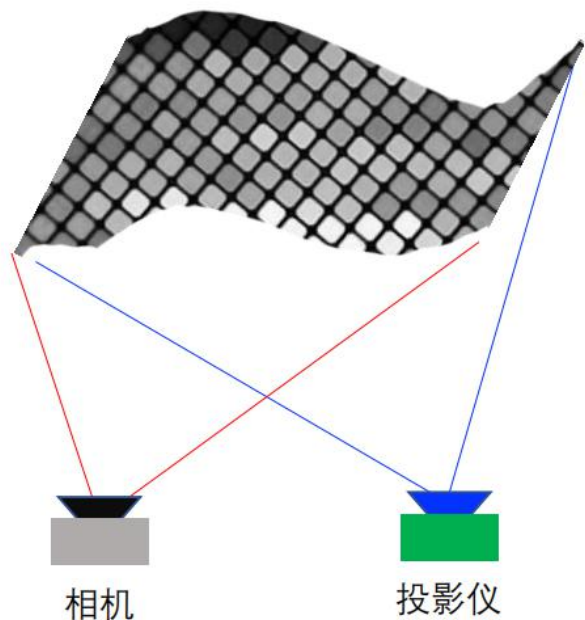


- (x, y) 是相机传感器平面上的点 (已知)
- (X, Y, Z) 是和 (x, y) 对应的空间点的坐标 (未知)
- f 是相机内参

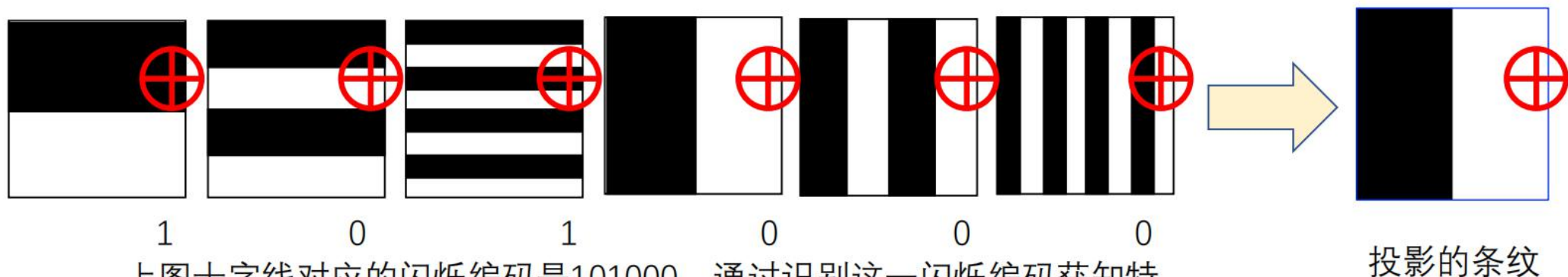
$$\begin{cases}
 AX + BY + CZ + D = 0 & \text{光源在相机坐标系中的方程} \\
 X = \frac{xZ}{f} & \text{针孔相机模型} \\
 Y = \frac{yZ}{f} & \text{针孔相机模型}
 \end{cases}
 \xrightarrow{\text{针孔相机模型带入平面方程}}
 \begin{cases}
 Z = -\frac{Df}{Ax + By + Cf} \\
 X = -\frac{Dx}{Ax + By + Cf} \\
 Y = -\frac{Dy}{Ax + By + Cf}
 \end{cases}$$

线光源

结构光3D成像原理



- ☺ 投影一个面到物体，接收相机通过图案编码识别出是投影仪哪个角度的“射线”，并计算距离。
- ☹ 如何设计投影的图案编码，以及如何识别编码？
- ☺ 时间和空间编码方式



上图十字线对应的闪烁编码是101000，通过识别这一闪烁编码获知特定像素对应的投影图像中的位置，并进一步得到投影光线的角度

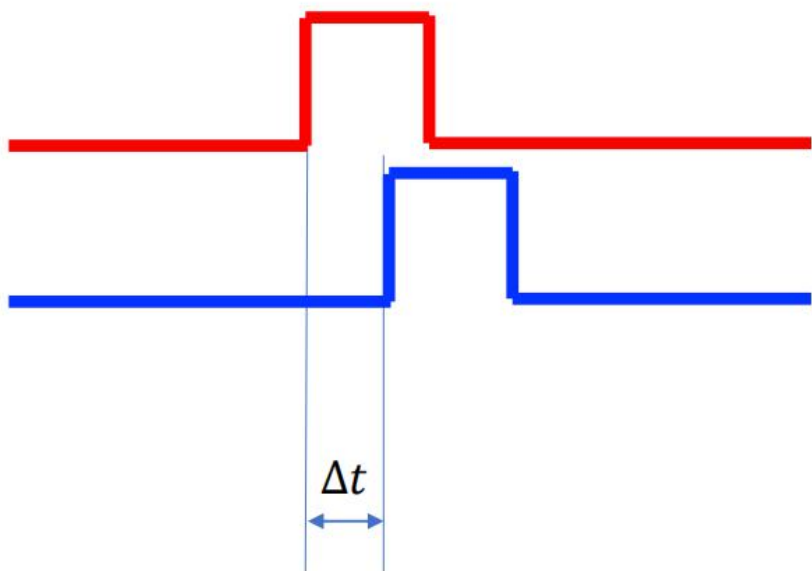
编码方式

目录

CONTENTS

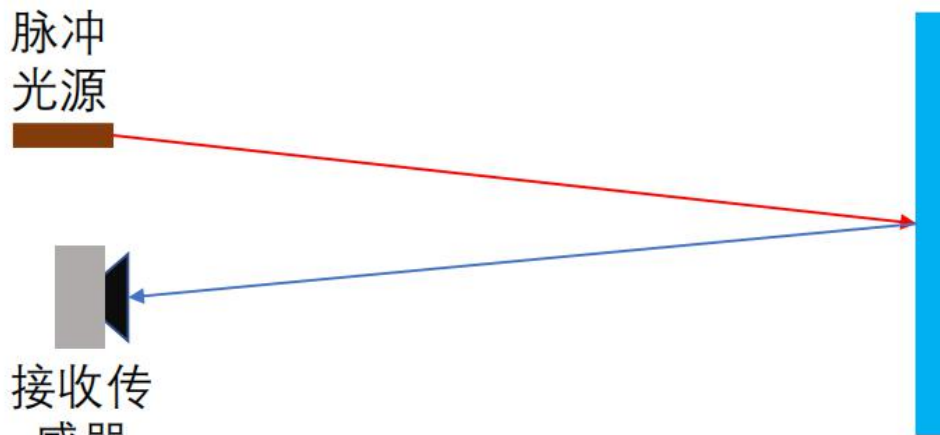
03 TOF3D成像原理

TOF 3D成像原理

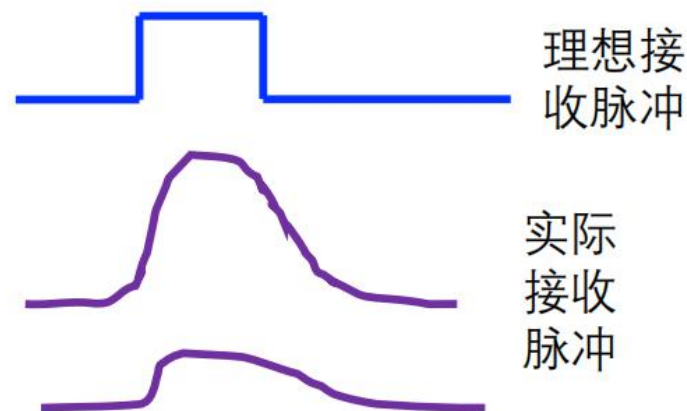


脉冲光源

接收传感器

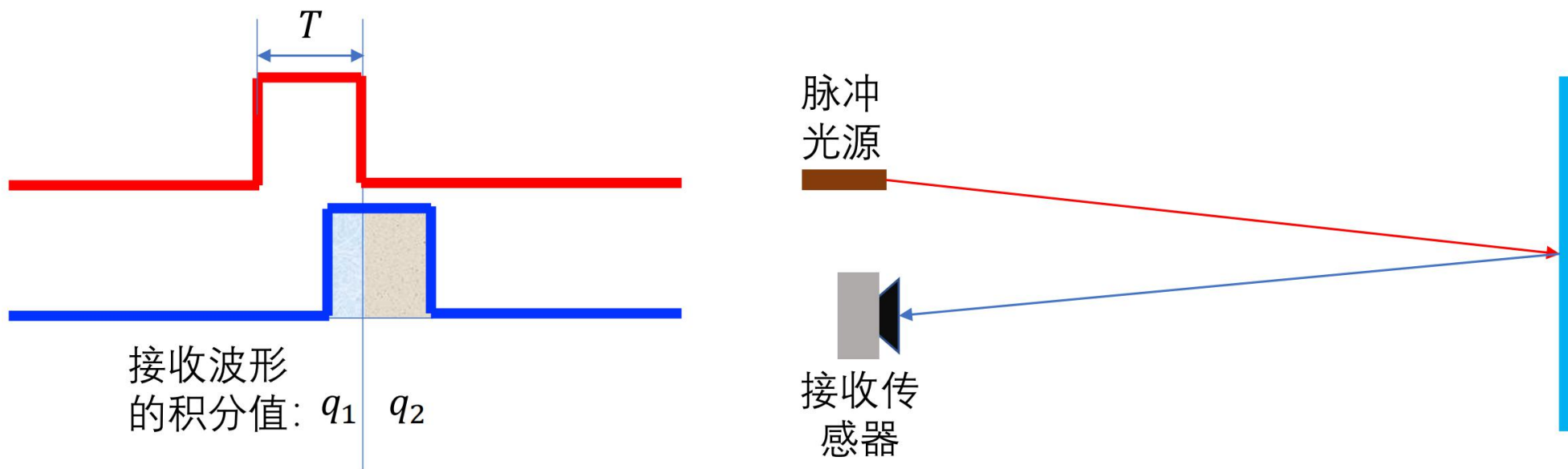


- 通过时间差计算得到总的路径长度: $c\Delta t$
- 当光源位置和相机光心几乎重合时, 到反射面距离为 $d = \frac{c\Delta t}{2}$
- 需要精确测量上升沿时间差收到距离、接收脉冲失真等影响
- 更加稳定的方案是检测整个脉冲波形面积比例



PTOF

TOF 3D成像原理



用积分比值反映延迟信息: $r = \frac{q_2}{q_1 + q_2}$

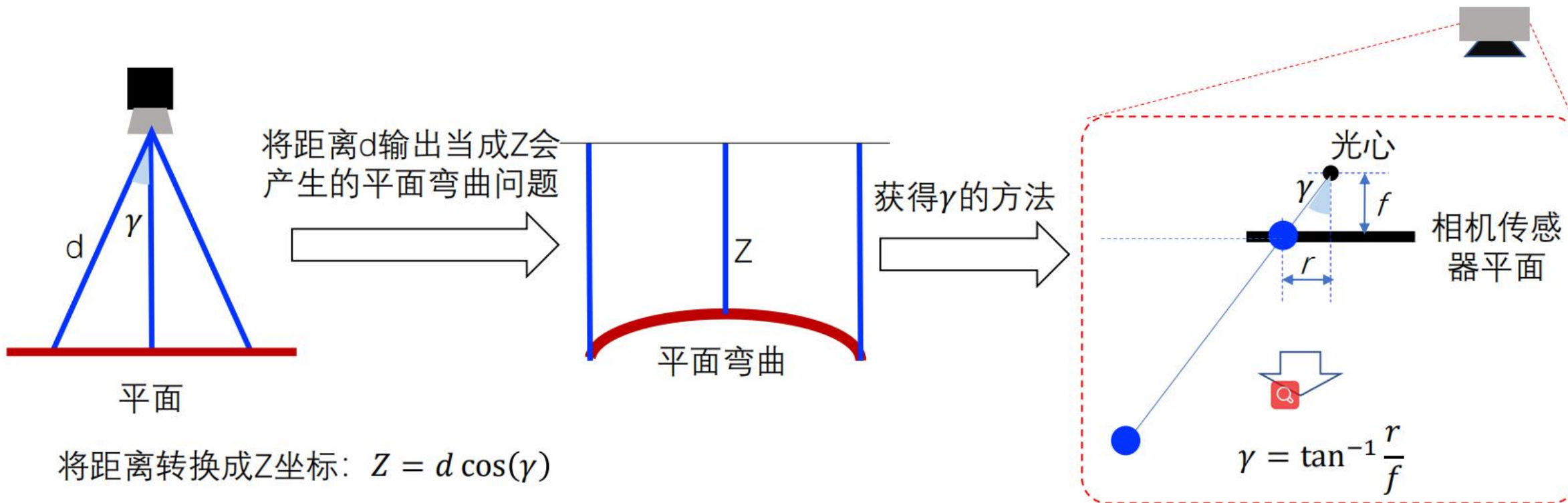
$r = 0$ 对应 $q_2 = 0$, 接收发送信号重合, 没有延迟

$r = 1$ 对应 $q_1 = 0$, 接收发送信号延迟 T

- 延迟量计算: $\Delta t = T \frac{q_2}{q_1 + q_2}$
- 光飞行距离: $cT \frac{q_2}{q_1 + q_2}$, 考虑光的来回, 物体到相机距离是: $d = \frac{cT}{2} \frac{q_2}{q_1 + q_2}$
- 为提高精度, 需要多个脉冲重复测量, 将结果平均

TOF 3D成像原理

- 如果深度相机输出的数据是像素到镜头的距离，那么需要按照视线“射线”进行额外的转换
- 如果将距离d输出当成Z会产生的平面弯曲问题



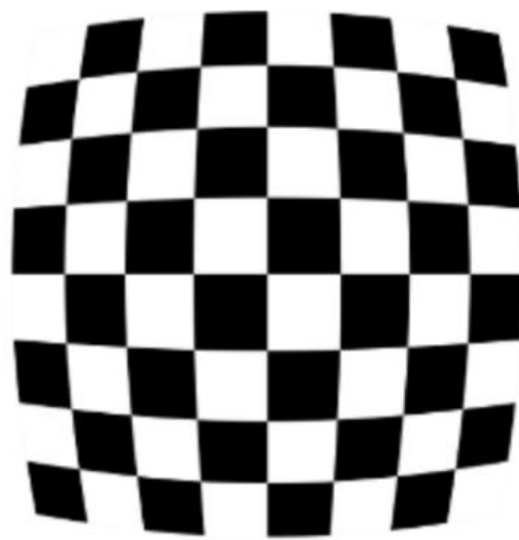
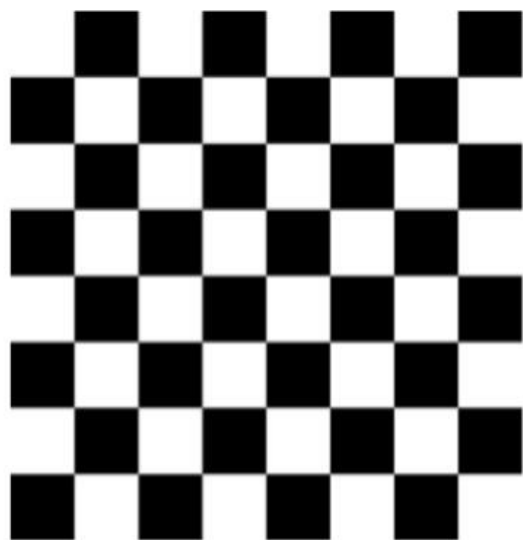
目录

CONTENTS

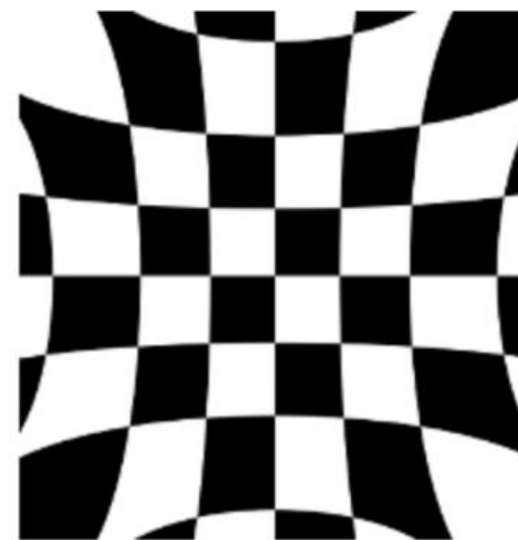
04

图像畸变与矫正

图像畸变与矫正



桶形失真



枕形失真

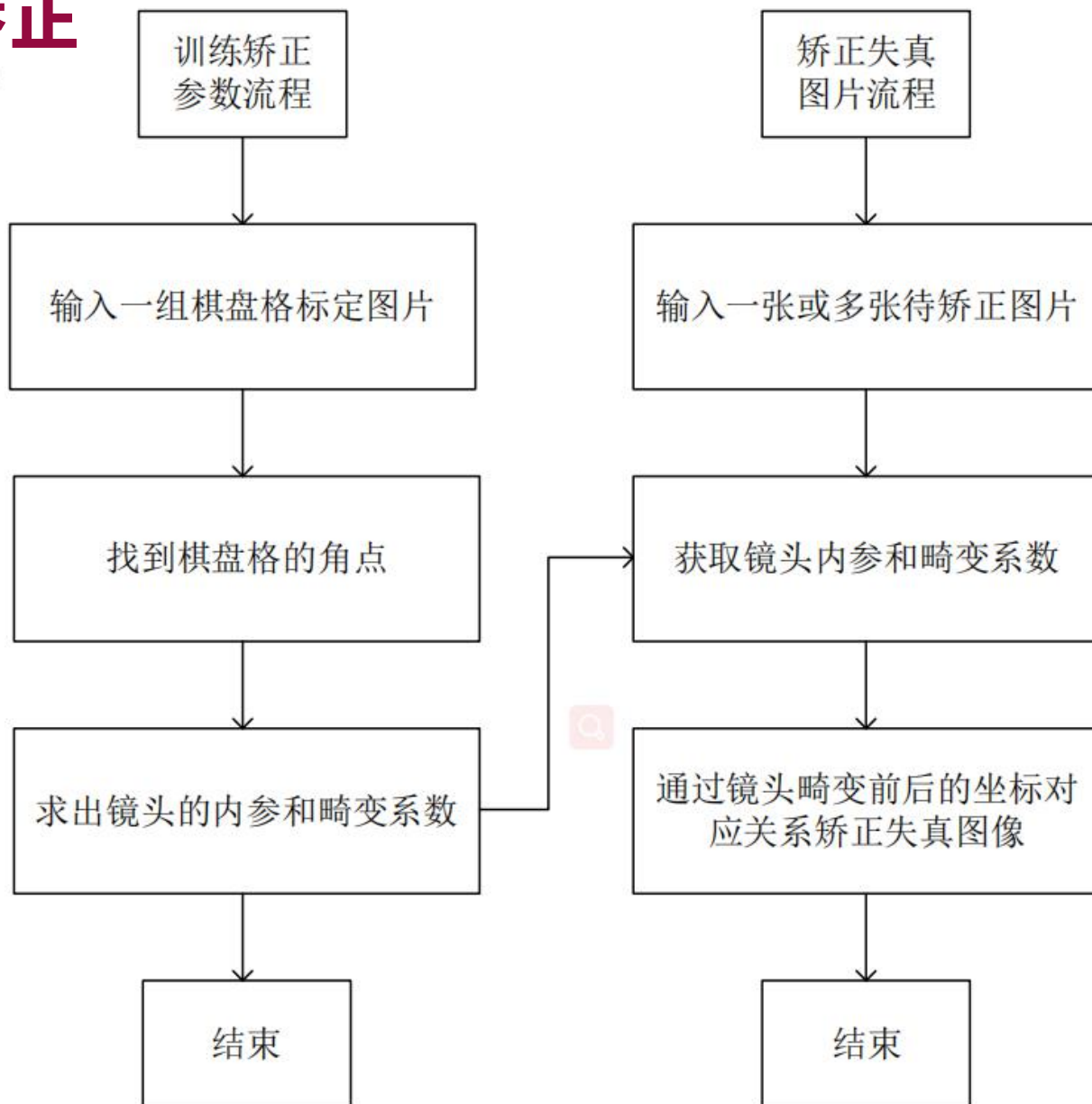
$$\begin{bmatrix} x_{corr} \\ y_{corr} \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2p_1 xy + p_2(r^2 + 2x^2) \\ p_1(r^2 + 2y^2) + 2p_2 xy \end{bmatrix}$$

通过畸变模型
调整像素位置

其中:

- (x, y) 是带有失真的像素坐标
 - $r^2 = x^2 + y^2$ 是像素到图形中心距离
 - (x_{corr}, y_{corr}) 是带有像素坐标修正值
- 失真矫正参数 $\{k_1, k_2, k_3, p_1, p_2\}$

图像畸变与矫正



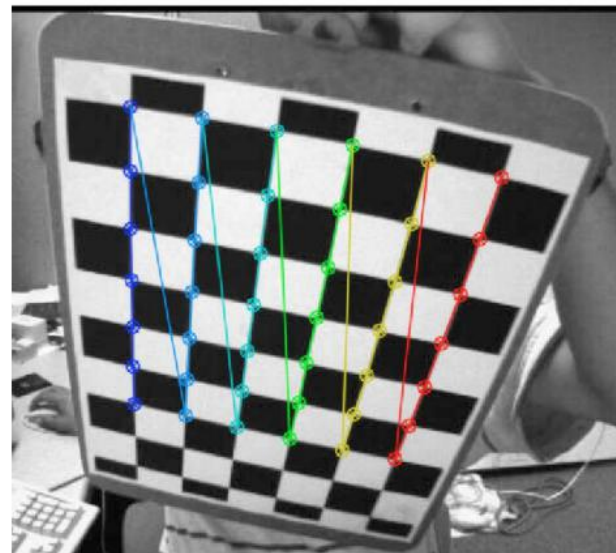
图像畸变与矫正-基于OPENCV的镜头较准教程

```

30 for fname in images:
31     print(fname)
32     img = cv2.imread(fname)
33     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
34
35     # 角点粗检测
36     ret, corners = cv2.findChessboardCorners(gray, (7,6), None)
37     if not ret:
38         continue
39     else:
40         objpoints.append(objp)
41
42     # 角点精检测
43     corners2 = cv2.cornerSubPix(gray, corners, (11,11), (-1,-1), criteria)
44     imgpoints.append(corners2)

```

通过粗检测/精细检测
两级，得到棋盘格角
点的坐标（像素坐标）



```

58 ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoints, gray.shape[:: -1], None, None)
59 print('mtx:', mtx)           # 内参矩阵
60 print('dist:', dist)         # 失真参数

```

计算矫正模型参数

$$\begin{bmatrix} x_{corr} \\ y_{corr} \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2p_1 xy + p_2 (r^2 + 2x^2) \\ p_1 (r^2 + 2y^2) + 2p_2 xy \end{bmatrix}$$

$[k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3]$

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

图像畸变与矫正-基于OPENCV的镜头较准教程

- 图像矫正方法
- 注意入参对应的两个矩阵
- 原始图像中弯曲的边界已经直了
- undistort有多种调用方法, 请参考opencv文档

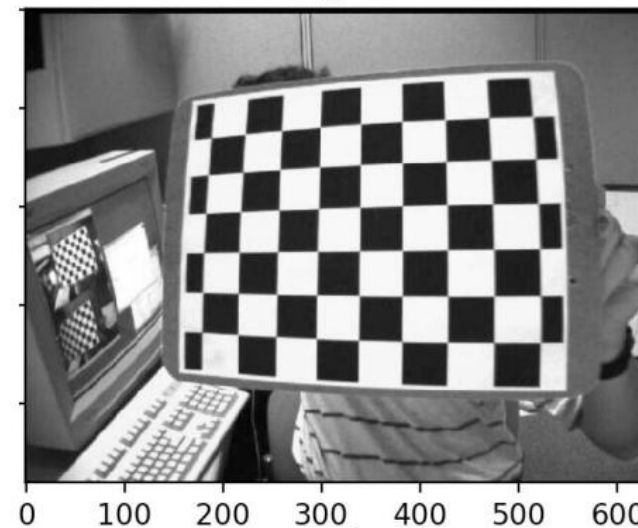
```
104 dst = cv2.undistort(img,mtx,dist)
```

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$[k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3]$$

$$\begin{bmatrix} x_{corr} \\ y_{corr} \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2p_1 xy + p_2 (r^2 + 2x^2) \\ p_1 (r^2 + 2y^2) + 2p_2 xy \end{bmatrix}$$

original



corrected 3



目录

CONTENTS

05

视觉与深度数据融合

为从深度相机得到的每个点云“染色”

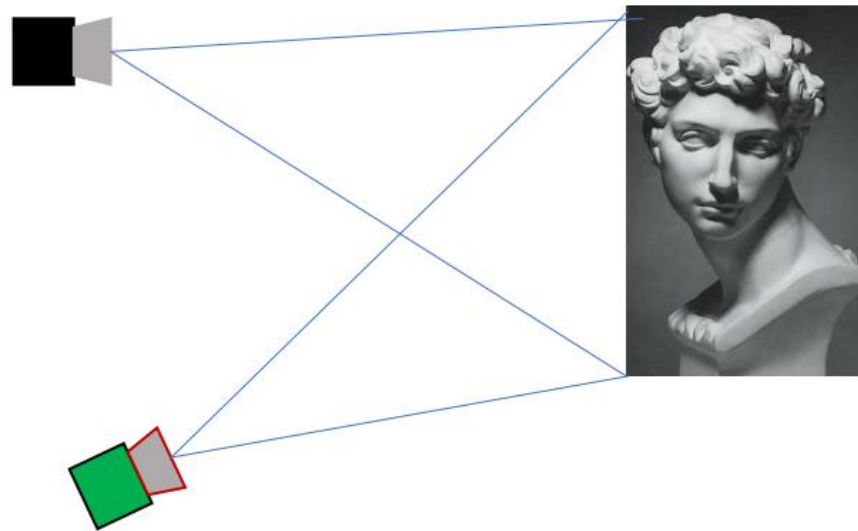
RGB相机中的像素位置
(和TOF相机坐标系
下的点云对应)

TOF相机坐标系
下的点云坐标

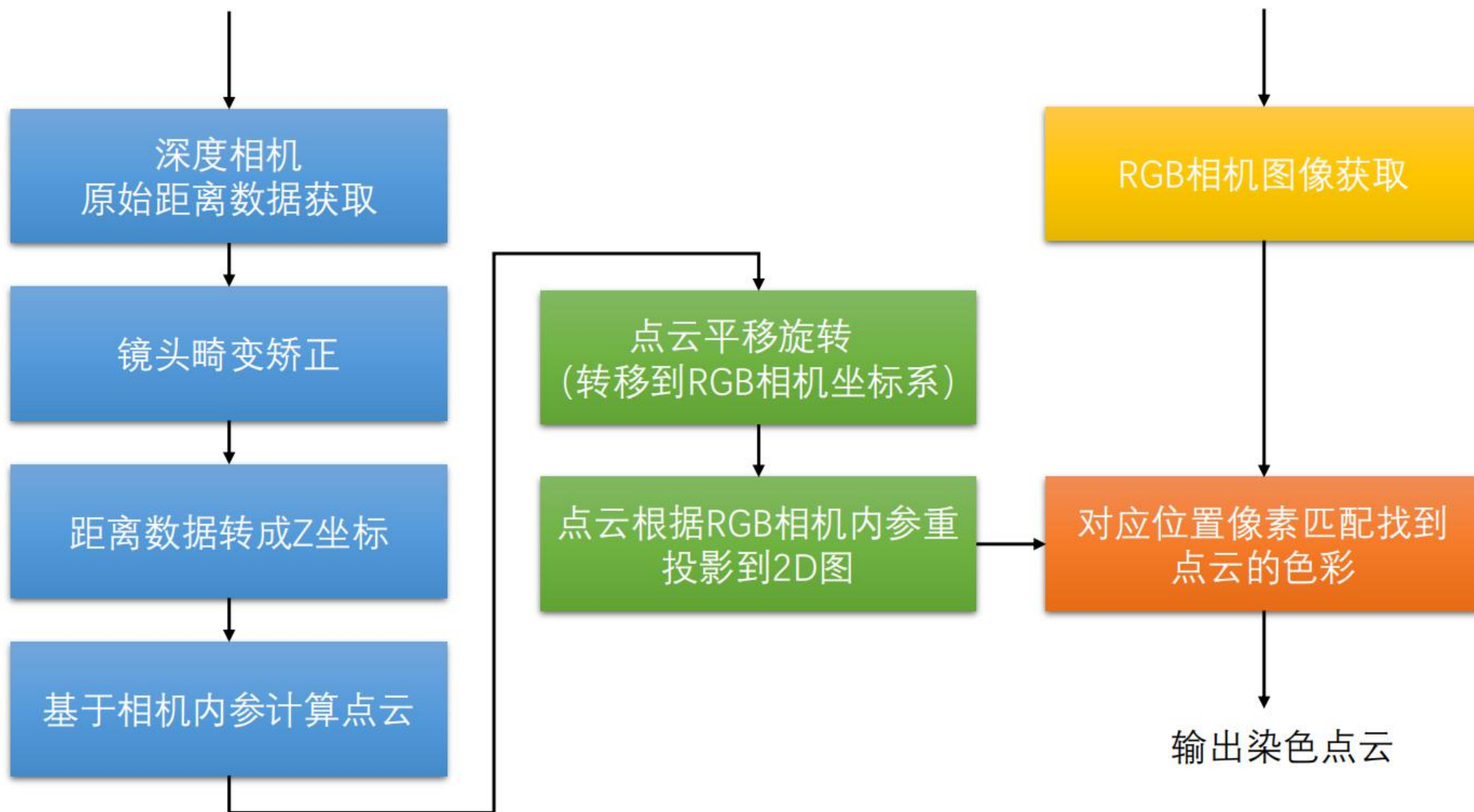
$$Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

RGB相机的
内参

TOF相机坐标系转到
RGB相机坐标系

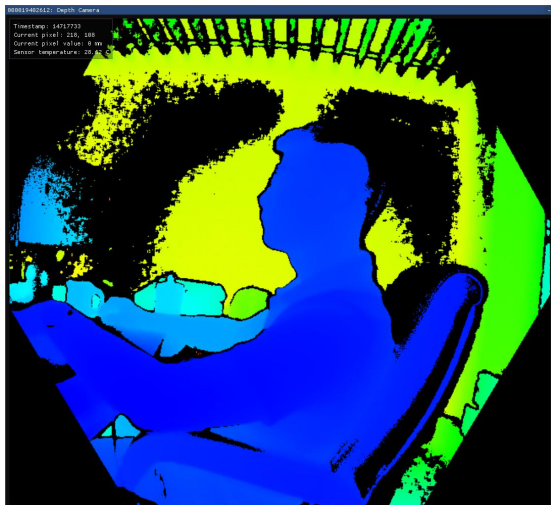


视觉与深度数据融合

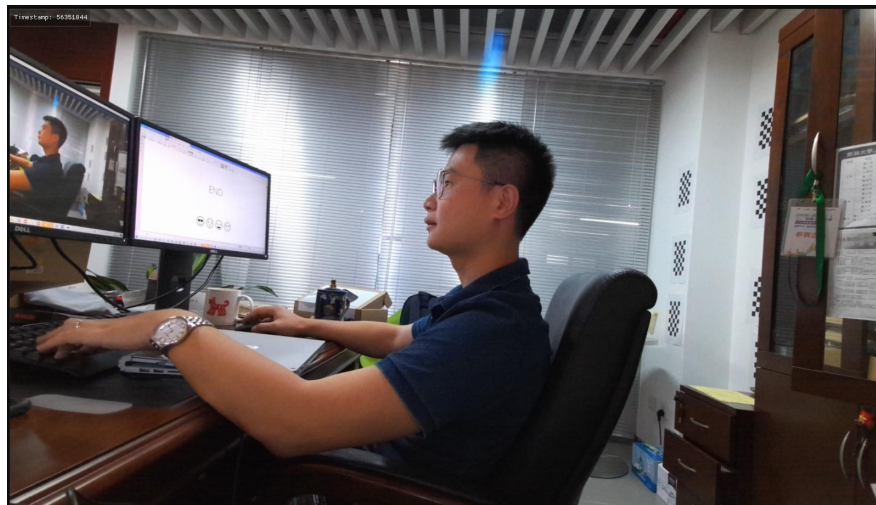


视觉与深度数据融合

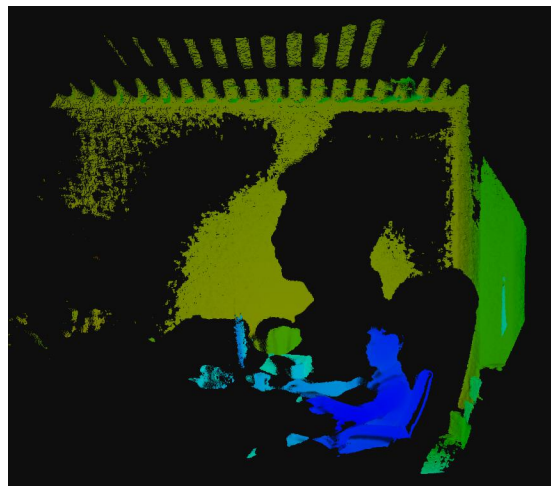
深度图



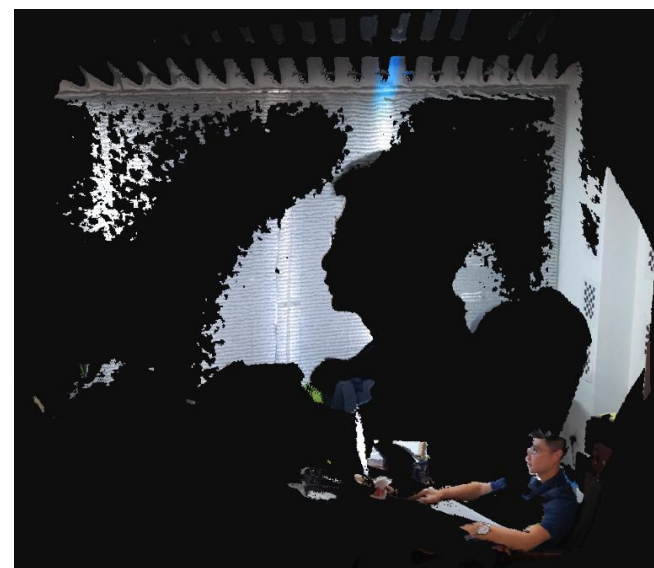
彩色图



三维点云



彩色三维点云



结课任务

三个事情

一、27人分成5组，每组至少5人

二、选题：

- 1、电力线巡检树障分析：涉及到点云滤波，电力线分割，地面分割，碰撞检查等。
- 2、家畜表型测量：涉及到家畜点云自动化数据获取，点云分割，自动化体尺测量（长，宽，高等）。
- 3、人体自动检测与测量：涉及到环境建模，人体检测，人体节点提取，人体测量。
- 4、快递信息快速提取：涉及到快递包裹的数据获取，点云分割，体积计算，尺度量测等方法。
- 5、建筑物违建检测：涉及到建筑物多期数据获取与建模（采用无人机），点云变化检测，建筑物变化信息计算等方法。
- 6、室内空间自动化建模：涉及室内空间三维点云数据采集，点云数据预处理，平面分割，墙面，门窗信息提取等方法。
- 7、博物馆文物三维数字化：涉及文物点云数据获取（不少于十个），文物建模，精细纹理映射，文物可视化展示（手机端或者PC端）等方法。

三：编程环境配置

- 1、编程环境以VS2015或VS2017为主
- 2、PCL环境的配置
- 3、Opencv环境配置（选配）



谢谢