

计算机视觉 实验课程

实验1 数字图像处理基础与相机标定



数字图像处理基础

1、数字图像处理简介

➤ 20世纪20年代初期：报纸业Bartlane 电缆图片传输系统

通过海底电缆将图像从伦敦传往纽约；

为了使用电缆传输，图像需要首先编码，并在接收端通过电报打印机进行重构。

最早的数字图像
5级色调



1、数字图像处理简介

➤ 20世纪20年代中期到末期：改进海底电缆传输系统

打印过程采用了新的光学还原技术

增加了图像的灰度等级

改进海底电缆传递照片

15级色调

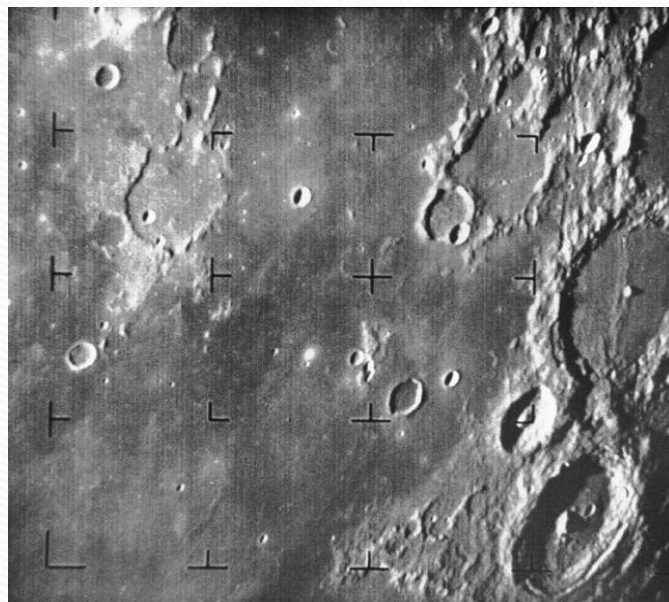


1、数字图像处理简介

➤ 20世纪60年代中期：太空计划的推动

1964年美国喷气推进实验室（JPL）使用数字计算机对“徘徊者7号”太空船传递的四千多张月球照片进行处理。

美国航天器传送
月球照片



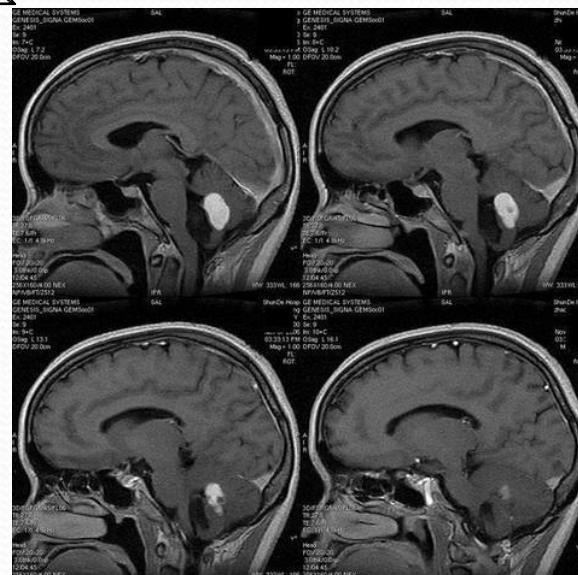
1、数字图像处理简介

➤ 20世纪70年代：应用到医学领域

1979年 美国科学家科马克、英国科学家豪斯费尔

德 因发明CT扫描而共同获得诺贝尔生理学或医学奖。

CT扫描



1、数字图像处理简介

➤ 20世纪80年代至今：迅猛发展

广泛应用于太空探索，遥感应用，生物医学工程，工业应用，军事应用等方面。



1、数字图像处理简介

◆数字图像处理定义

又称为计算机图像处理,它是指将图像信号转换成数字信号并利用计算机对其进行 处理的过程。

◆数字图像处理目标

为了便于人们分析而对图像信息进行改进;

为了使机器具有类似于人类的分析和理解图像的能力而进行的各种分析和研究;

1、数字图像处理简介

◆ 图像处理的研究分类

➤ 低级处理

灰度化、色彩增强、去噪声、调亮度、局部修复、模糊化；例如，常见的*Photoshop*、美图工具 等。（输入图像，输出图像）

➤ 中级处理

图像分割，轮廓提取等；例如，在遥感地理信息系统，经常对航拍图片进行分析，来判断某个区域的环境污染情况。（输入图像，输出属性）

➤ 高级处理

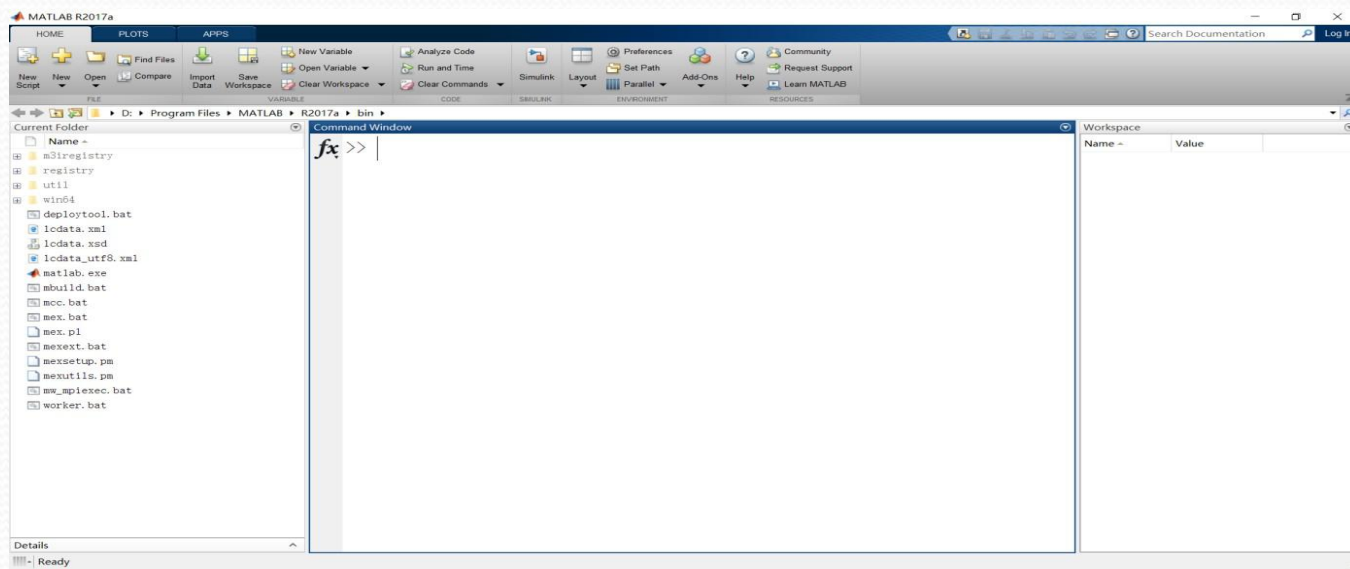
基于图像的内容分析、理解和识别；例如，车牌识别、虹膜识别、人脸识别等。（输入图像，输出分析结果）

2、开发工具简介

◆ Matlab简介

Matlab (Matrix Laboratory) 是当今很流行的科学计算软件。信息技术、计算机技术发展到现在，科学计算在各个领域得到了广泛的应用，在诸如控制论、时间序列分析、系统仿真、图像信号处理等方面产生了大量的矩阵及其他计算问题。Matlab软件适时推出，为人们提供了一个方便的数值计算和动态仿真平台。

文件管理器
变量管理器
M文件编辑器
命令行执行区
菜单区



2、开发工具简介

◆ *Matlab*图像处理工具箱简介

Matlab对图像的处理功能主要集中在它的图像处理工具箱（**Image Processing Toolbox**）中。

图像处理工具箱是由一系列支持图像处理操作的函数组成，可以进行如下操作：

- 线性滤波和滤波器设计
- 图像变换
- 图像增强
- 数学形态学处理
-

3、开发技能基础

◆ 什么是图像？

图像是当光辐射能量照在物体上，经过它的反射或透射，或由发光物体本身发出 的光能量，在人的视觉器官中所重现出的物体的视觉信息。

◆ 模拟图像和数字图像

- 按照图像信息的坐标取值是否连续，分为模拟图像和数字图像
- 模拟图像：坐标 (x, y) 为连续值， $I = f(x, y)$ 表示该点的某个性质的度量值。比如说照片，绘画等。
- 数字图像：坐标 (x, y) 为离散值，采样形成像素点(pixel)，从而形成了 离散的二维矩阵。数码相机，计算机处理的图像都属于数字图像。

3、开发技能基础

数字图像主要具有如下优点：

- ◆精度高：目前的计算机可以将一幅模拟图像数字化为任意的二维数组，即数字图像可以由无限多个像素组成。
- ◆易用性：由于数字图像本质上是一组数据，所以可以利用计算机对它进行任意方式的修改。
- ◆持久性：模拟图像（例如照片）即便是使用非常好的底片和相纸，也会随着时间的流逝而褪色、发黄，而数字图像可以存储在光盘中，上百年再用计算机重现也不会有丝毫的改变。

3、开发技能基础

一幅 $m \times n$ 的数字图像可用矩阵表示为

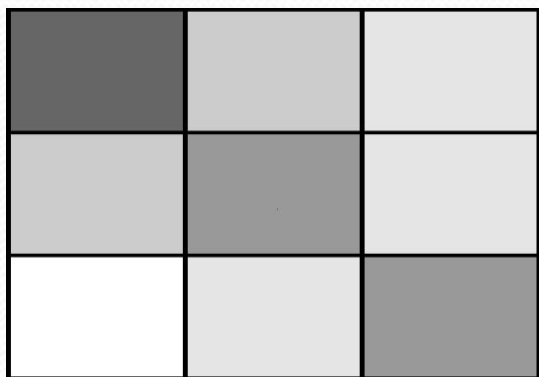
$$F = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,n-1) \\ f(1,0) & f(1,1) & \dots & f(1,n-1) \\ \dots & \dots & \dots & \dots \\ f(m-1,0) & f(m-1,1) & \dots & f(m-1,n-1) \end{bmatrix}$$

数字图像中的每个像素(*pixel*)都对应于矩阵中相应的元素。
把数字图像表示成矩阵 的优点在于，能应用矩阵理论对图像进行分析处理。

3、开发技能基础

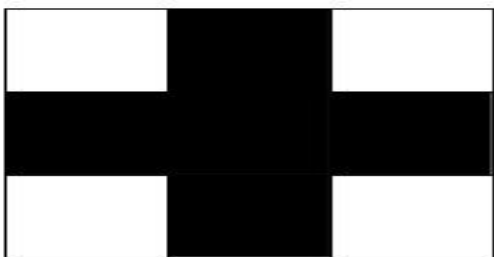
图像表示对比

◆灰度图像是指每个像素由一个量化灰度来描述的图像，没有彩色信息；



$$\begin{pmatrix} 0 & 100 & 150 \\ 90 & 50 & 130 \\ 250 & 220 & 180 \end{pmatrix}$$

◆二值图像是指像素灰度只有两级（通常取0(黑色)或1(白色)）；

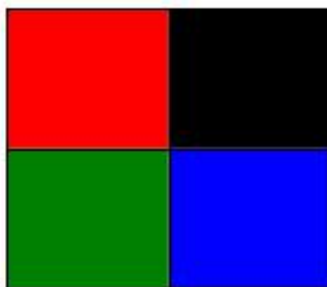


$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

3、开发技能基础

图像表示对比

◆彩色图像是指每个像素由红、绿、蓝（分别用 R 、 G 、 B 表示）3原色构成的图像，其中 R 、 G 、 B 是由不同的灰度级描述的



$$R = \begin{pmatrix} 255 & 0 \\ 0 & 0 \end{pmatrix}$$

$$G = \begin{pmatrix} 0 & 0 \\ 255 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 0 \\ 0 & 255 \end{pmatrix}$$

3、开发技能基础

*matlab*图像处理支持如下的图像类型
:

- ◆ 真彩色图像 (*RGB images*)
- ◆ 索引色图像 (*index images*)
- ◆ 灰度图像 (*intensity images*)
- ◆ 二值图像 (*binary images*)
- ◆ 图像序列 (*sequence image*)

3、开发技能基础

- 真彩色图像

- 真彩色图像用 R 、 G 、 B 3个分量表示1个像素的颜色，所以对1个尺寸为 $m \times n$ 的真彩色图像来说，其数据结构就是一个 $m \times n \times 3$ 的多维数组。
- 如果要读取图像中 (r, c) 处的像素值，可以查看三元组 $(r, c, :)$ 。
- 真彩色图像常用的存储方法是用无符号整型存储`UInt8`，其亮度值的范围为 $[0, 255]$ 。
- 此外，真彩色图像可用双精度存储，此时亮度值的范围是 $[0, 1]$ 。

3、开发技能基础

- 真彩色图像

双精度类: Double (每个像素占8个字节)	整数类: Uint8 (每个像素占1个字节)
数组大小: $m \times n \times 3$ (:,:,1) - 红色分量 (:,:,2) - 绿色分量 (:,:,3) - 蓝色分量 像素取值: [0,1]	数组大小: $m \times n \times 3$ (:,:,1) - 红色分量 (:,:,2) - 绿色分量 (:,:,3) - 蓝色分量 像素取值: [0,255]

3、开发技能基础

- 索引色图像

- 索引图像是把像素值直接作为RGB调色板下标的图像。
- *Matlab*中的索引色图像包含2个结构，一个是调色板`map`；另外一个是指图像数据矩阵`X`。调色板是一个有3列和若干行的色彩映像矩阵，矩阵的每行都代表一种色彩，通过3个分别代表红、绿、蓝颜色强度的双精度数，形成一种特定的颜色。图像数据可以是`uint8`或是双精度的。
- 此外，*Matlab*中的调色板的色彩强度是`[0,1]`中的浮点数，0代表最暗，1代表最亮。

3、开发技能基础

- 索引色图像

双精度类:Double (每个元素占8个字节)	整数类: Uint8 (每个元素占1个字节)
图像数组大小: $m \times n$ 图像元素取值:[1,p] 调色板矩阵: $p \times 3$	图像数组大小: $m \times n$ 图像元素取值:[0,p-1] 调色板矩阵: $p \times 3$

3、开发技能基础

- 灰度图像

- 存储灰度图像只需要一个数据矩阵
- 数据类型可以是`double`也可以是`uint8`
- 存储时会使用一个默认的调色板来显示图像

- 二值图像

- 二值图像只需一个数据矩阵
- 每个像素只有2个灰度值
- 可以采用`uint8`或`double`类型存储

3、开发技能基础

- 图像序列

matlab图像处理支持将多帧图像连接成图像序列。一般可以视为一个四维的数组，图像帧的序号在图像的长、宽、颜色深度之后构成第四维。

例如

- 一个包含了5幅400×300真彩色图像的序列A，其大小为400×300×3×5
- 一个包含了5幅400×300灰度或是索引图像的序列A，其大小为400×300×1×5
- 提取其中一帧（如第2帧），即A(:, :, :, 2)

3、开发技能基础

- 图像基本读写

imread: 读取图形文件格式的图像

imwrite: 写入图形文件格式的图像

imfinfo: 获取图像的信息

load+save: 以*Mat*文件加载或保存矩阵数据

imshow: 显示加载到*Matlab*中的图像

3、开发技能基础

- *imread*

利用函数*imread*可以完成图形图像文件的读取操作，其语法如下

➤ 默认图像: *A=imread(filename, fmt)*

读取的大多数图像均为8bit。当这些图像加载到内存中时，*Matlab*就将其存储 在类*uint8*中。

➤ 索引图像: *[X, map] =imread(filename, fmt)*

即使图像阵列的本身为*uint8*或*uint16*，*imread*函数仍然将颜色映像表读取并 存储到一个双精度的浮点类型的阵列中。

3、开发技能基础

- *imwrite*

利用*imwrite*函数可以完成图形图像文件的写入操作，其语法为：

- *imwrite(A, filename, fmt)*
- *imwrite(X, map, filename, fmt)*

Matlab 缺省的保存方式：*uint8*数据格式。

在*Matlab*中使用的许多图像都是*8bit*，并且大多数的图像文件并不需要双精度 的浮点数据。

3、开发技能基础

- *imfinfo*

*Matlab*提供了*imfinfo*函数用于从图像文件中查询其信息。所获取的信息依文件类型的不同而不同，但至少包含下面的内容。

文件名

文件格式

文件格式的版本号

文件修改时间

文件的字节大小

图像的宽度（像素）

图像的长度（像素）

每个像素的位数

图像类型（即该图像是*RGB*(真彩)图像、灰度图像还是索引图像）

相机标定

引言

几个问题

什么是摄像机标定？

为什么要对摄像机进行标定？

如何进行摄像机标定？


引言

三维重建是人类视觉的主要目的，也是计算机视觉的最主要的研究方向. (Marr 1982)

所谓三维重建就是指从二幅和二幅以上图象恢复空间点三维坐标的过程。

三维重建的三个关键步骤

- 图象对应点的确定
- 摄像机标定
- 二图象间摄像机运动参数的确定



图象的形成过程 和 摄像机针孔模型

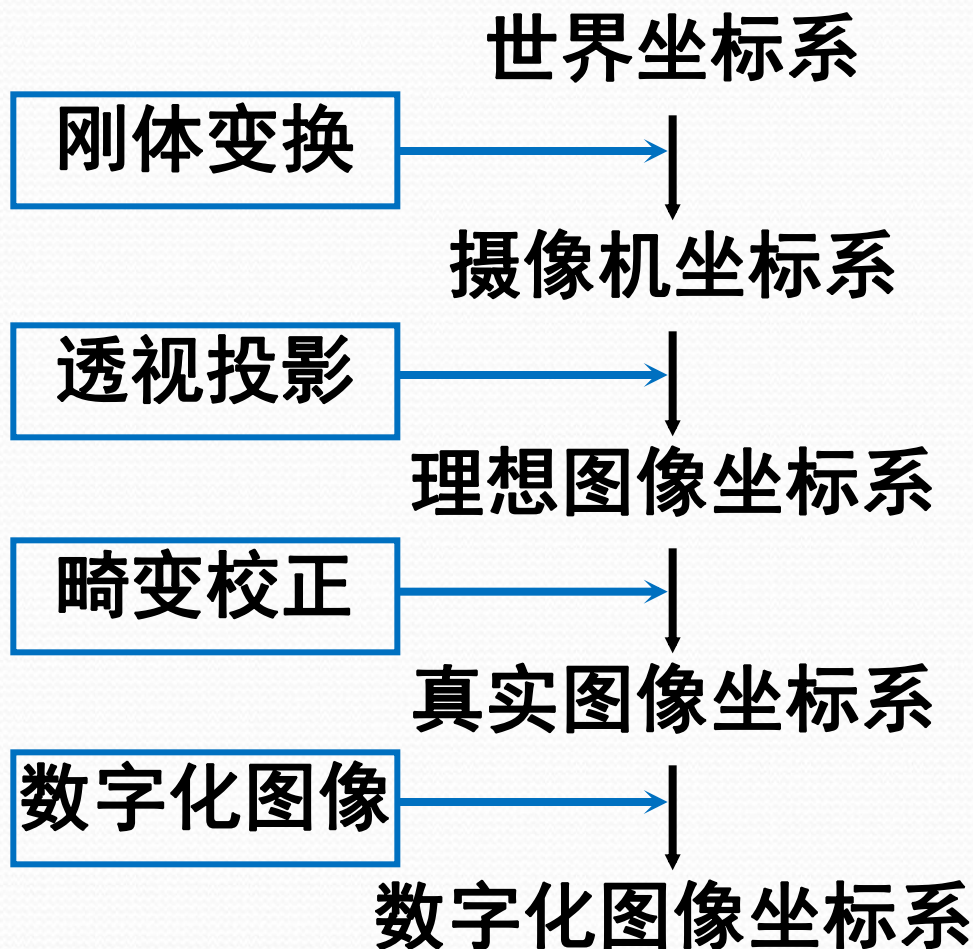
摄像机光学成像过程的四个步骤

1、刚体变换公式

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + t$$

齐次坐标形式

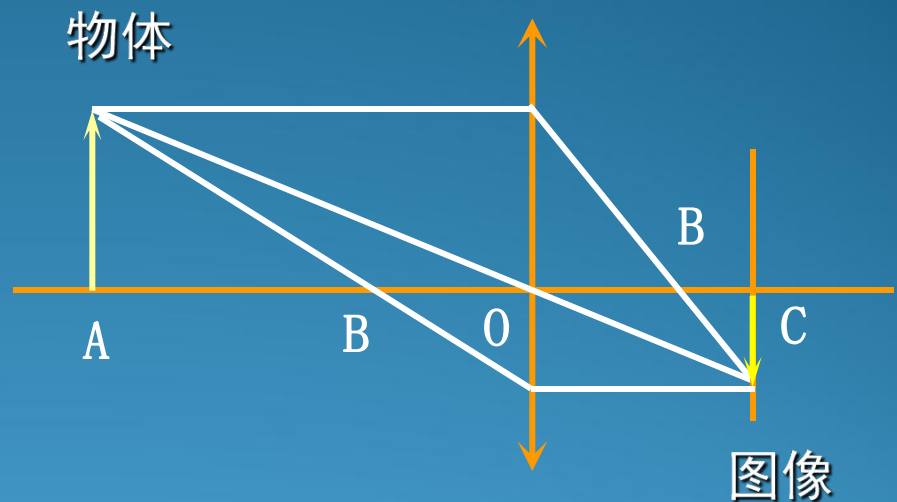
$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$



2、透视投影——透镜成像原理图

$$\frac{1}{f} = \frac{1}{m} + \frac{1}{n}$$

一般地由于 $n \gg f$
于是 $m \approx f$ 这时可
以将透镜成像模型近
似地用小孔模型代替



$f=OB$ 为透镜的焦距
 $m=OC$ 为像距
 $n=AO$ 为物距

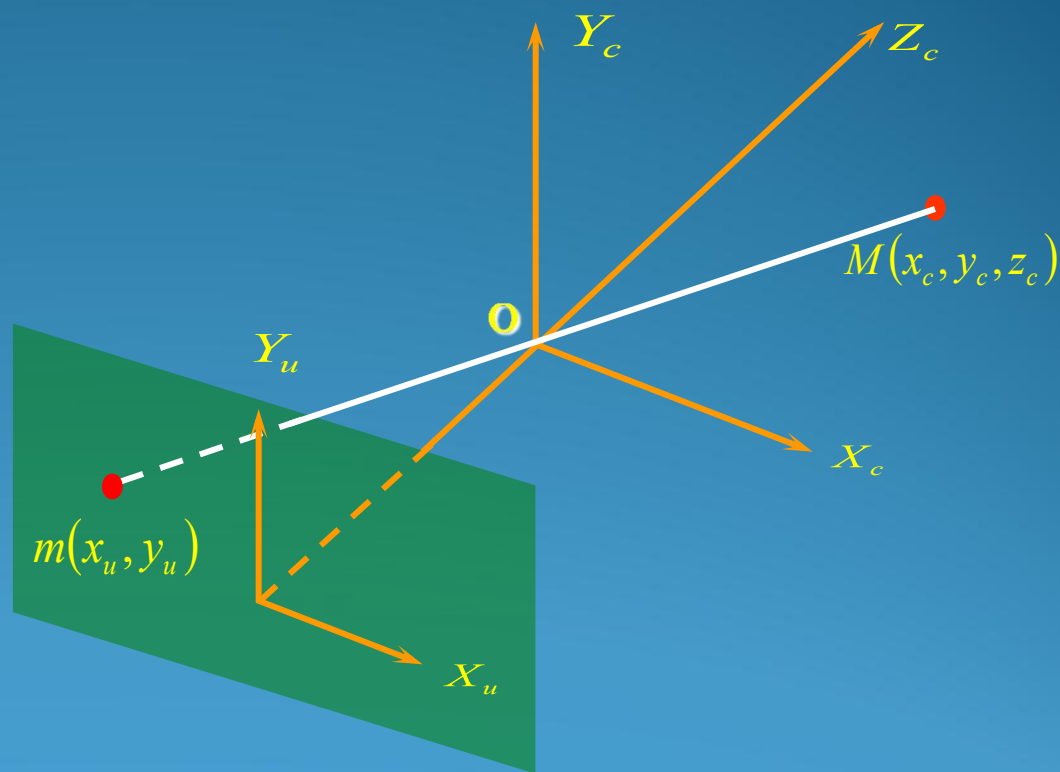
2、透视投影——小孔成像模型

$$x_u = -f \frac{x_c}{z_c}$$

$$y_u = -f \frac{y_c}{z_c}$$

写成齐次坐标形式为

$$z_c \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$



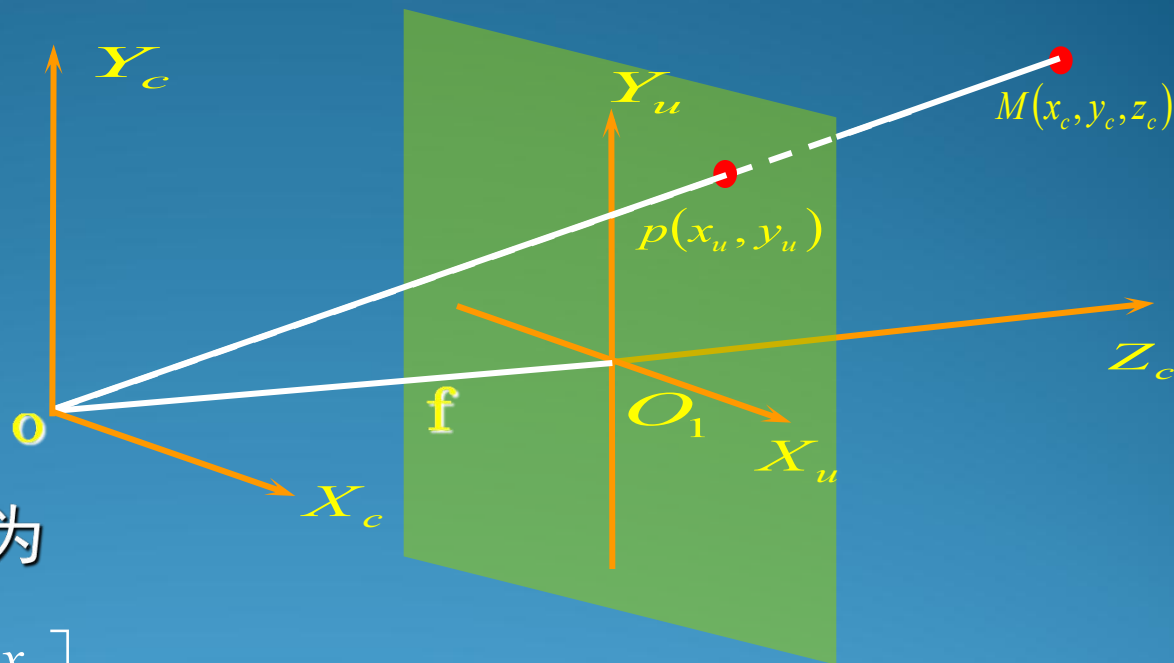
2、中心透视投影模型

$$x_u = f \frac{x_c}{z_c}$$

$$y_u = f \frac{y_c}{z_c}$$

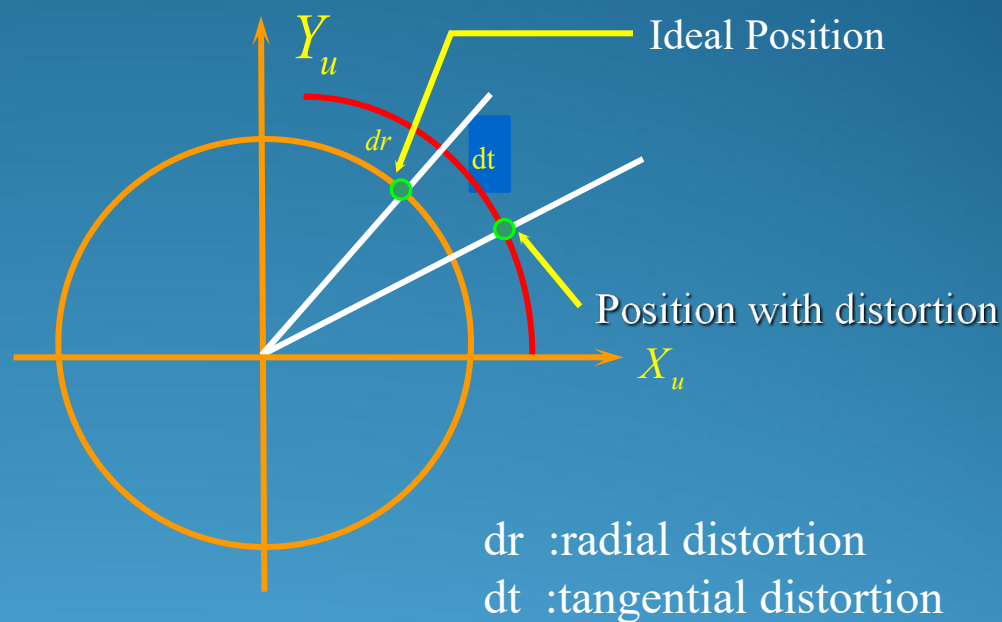
写成齐次坐标形式为

$$z_c \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

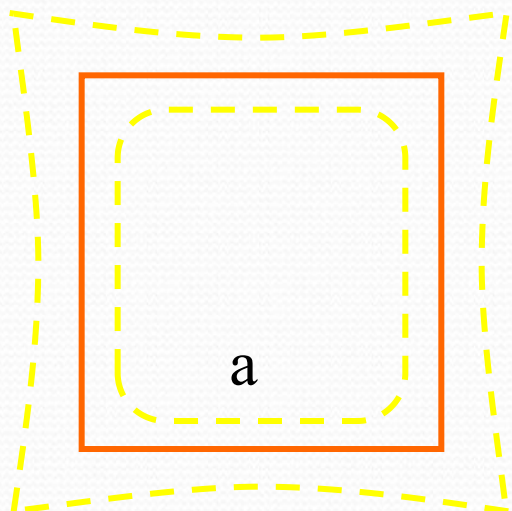


3、畸变校正——径向和切向畸变

$$x_d = x_u + \delta_{x_u}(x_u, y_u)$$
$$y_d = y_u + \delta_{y_u}(x_u, y_u)$$



3、畸变校正——其它畸变类型



a :barrel distortion
b :pincushion distortion

桶形畸变a和枕形畸变b



桶形畸变



枕形畸变

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_d \\ y_d \end{pmatrix} (1 + k_1 r^2 + k_2 r^4)$$

现在常用模型

4、图像数字化

O_1 在 u, v 中的坐标为 (u_0, v_0)

像素在轴上的物理尺寸为 dx, dy

Affine Transformation :

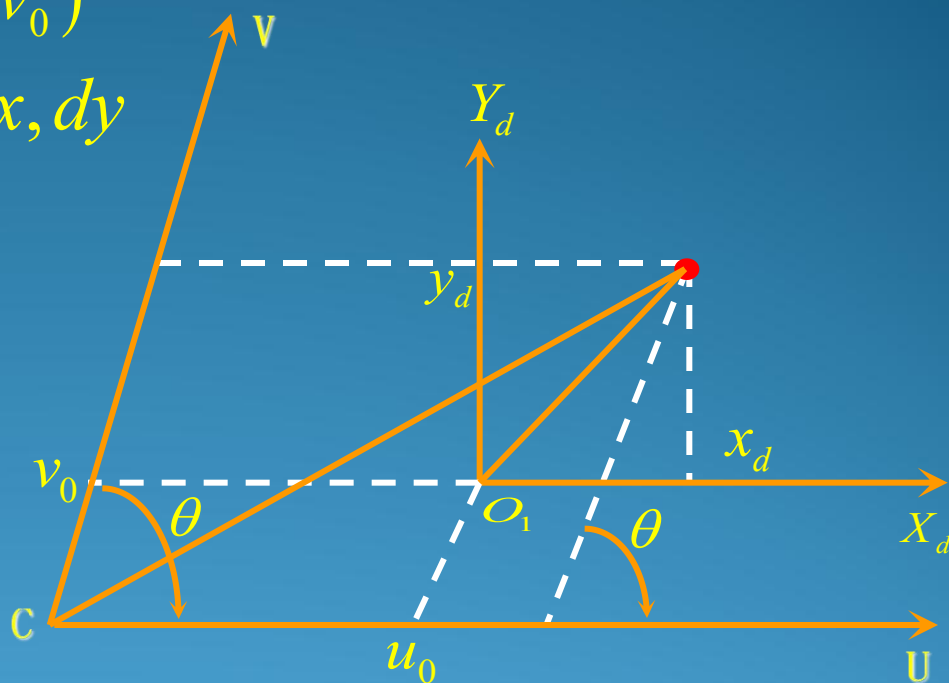
$$u = u_0 + \frac{x_d}{dx} - \frac{y_d \cot \theta}{dx}$$

$$v = v_0 + \frac{y_d}{dy \sin \theta}$$

齐次坐标形式:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & -f_u \cot \theta & u_0 \\ 0 & f_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix}$$

其中 $f_u = \frac{1}{dx}, f_v = \frac{1}{dy}$



摄像机的内参数矩阵 \mathbf{K}

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_u & -f_u \cot \theta & u_0 \\ 0 & f_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix}$$

线性摄像机成像模型

图像像素坐标系 \longleftrightarrow 图像物理坐标系 \longleftrightarrow 摄像机坐标系 \longleftrightarrow 世界坐标系

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & -f_u \cot \theta & u_0 \\ 0 & f_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{z_c} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

最终得到： 图像像素坐标系 \longleftrightarrow 世界坐标系

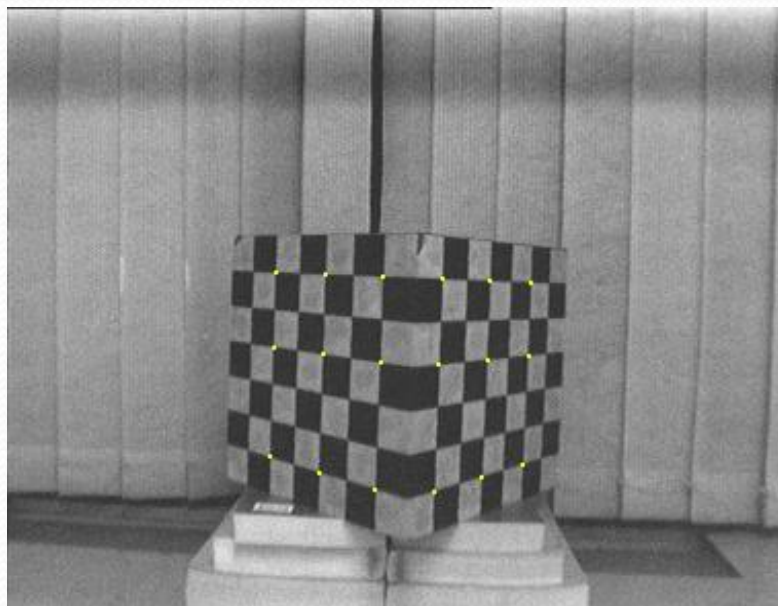
$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_u & -f_u \cot \theta & u_0 \\ 0 & f_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_K \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

这是忽略畸变的线性成像模型

传统的摄像机标定方法

特点

要求摄像机标定块，算法复杂，精度高

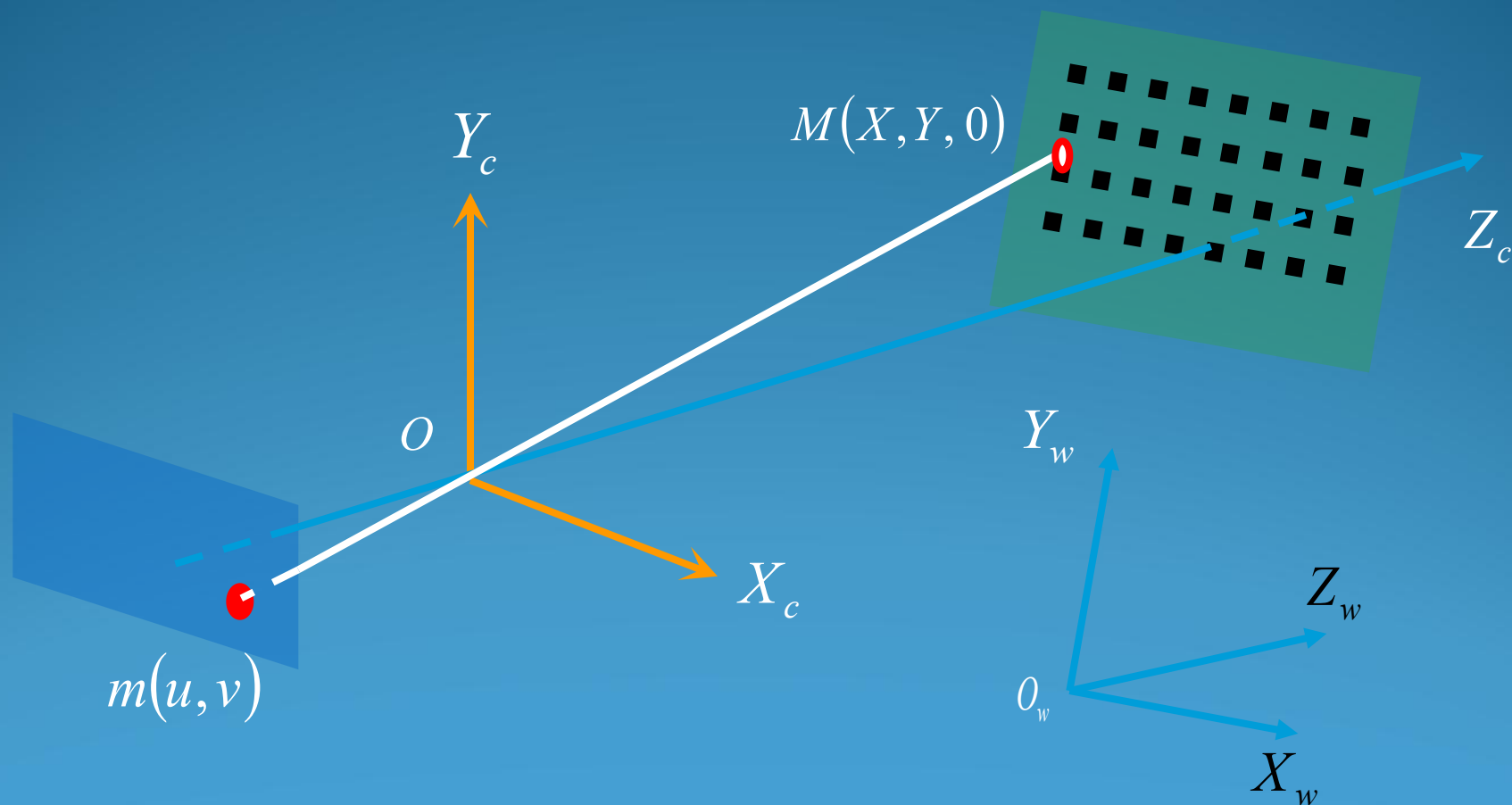


直接线性变换法

- 优点：
 - 标定精度高
 - 计算简单
- 缺点：
 - 需要高精度的标定设备
 - 使用不方便
- => 单独平面可否标定？

张正友的平面标定方法

<http://research.microsoft.com/en-us/um/people/zhang/Calib/>



张正友的平面标定方法

基本原理:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

- 在这里假定模板平面在世界坐标系 $Z = 0$ 的平面上
- 其中, K 为摄像机的内参数矩阵, $\tilde{M} = [X \ Y \ 1]^T$ 为模板平面上点的齐次坐标, $\tilde{m} = [u \ v \ 1]^T$ 为模板平面上点投影到图象平面上对应点的齐次坐标, $[r_1 \ r_2 \ r_3]$ 和 t 分别是摄像机坐标系相对于世界坐标系的旋转矩阵和平移向量

张正友的平面标定方法

$$s\tilde{m} = H\tilde{M} \quad \text{其中} \quad H = [h_1 \ h_2 \ h_3] = \lambda K [r_1 \ r_2 \ t]$$

根据旋转矩阵的性质，即 $r_1^T r_2 = 0$ 和 $\|r_1\| = \|r_2\| = 1$ ，每幅图象可以获得以下两个对内参数矩阵的基本约束

$$h_1^T K^{-T} K^{-1} h_2 = 0$$

$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2$$

由于摄像机有5个未知内参数，所以当所摄取得的图象数目大于等于3时，就可以线性唯一求解出 K

张正友平面标定方法

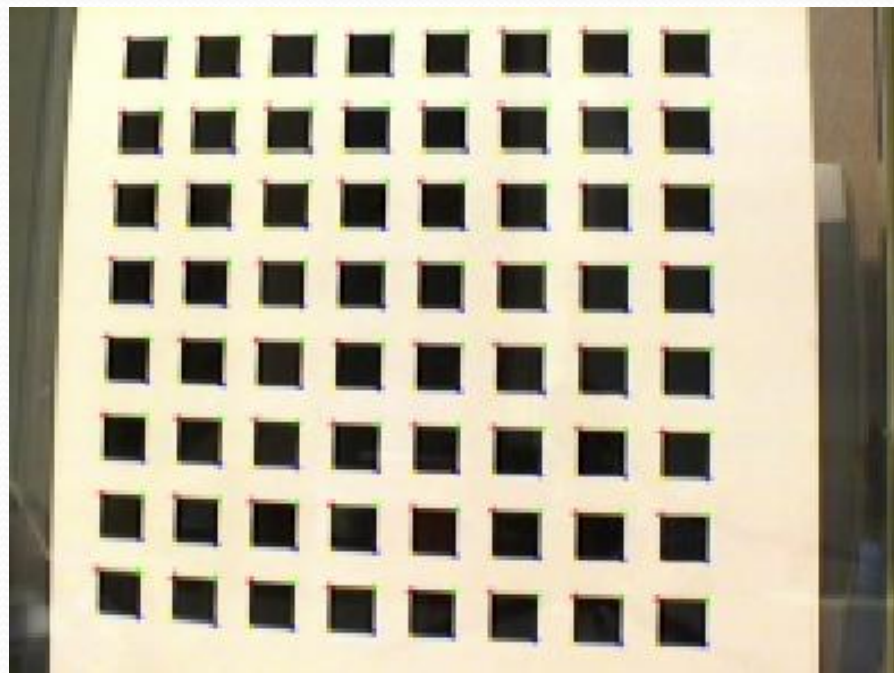
$$H = [h_1 \ h_2 \ h_3] = \lambda' K [r_1 \ r_2 \ t]$$

已知 $H \ K \Rightarrow r_1 \ r_2 \ t$

$$r_3 = r_1 \times r_2$$

最后计算出 $K \ R \ t$

张正友的平面标定方法



张正友方法所用的平面模板

张正友的平面标定方法

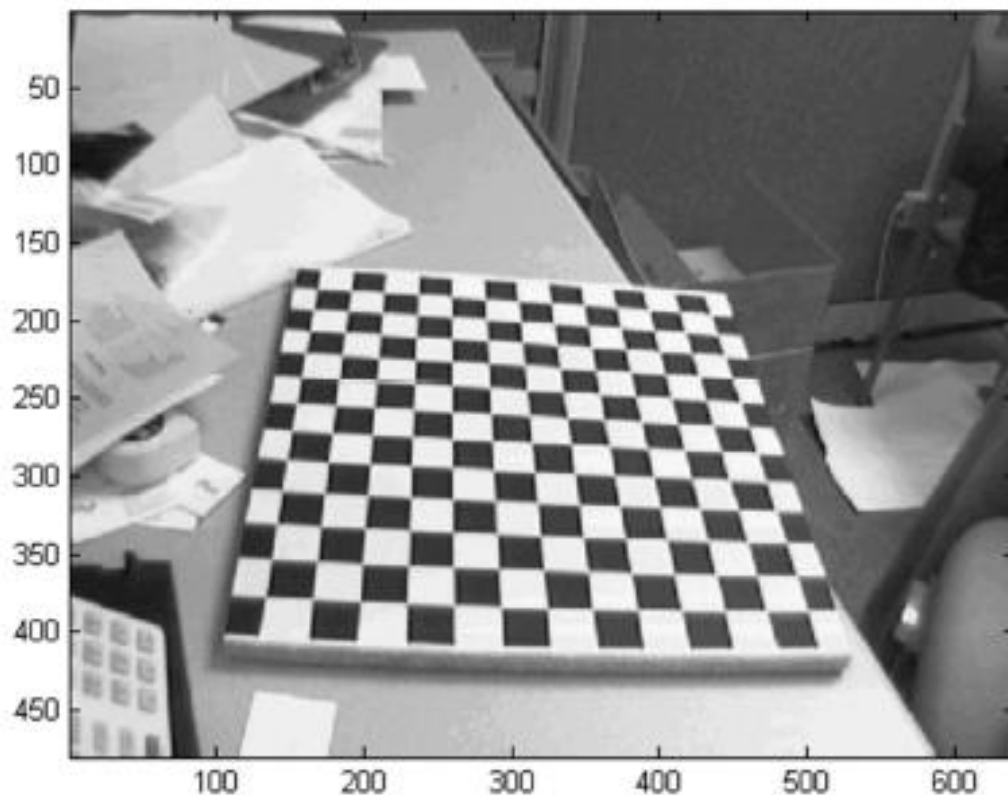
算法简单描述

1. 打印一张模板并贴在一个平面上
2. 从不同角度拍摄若干张模板图象
3. 检测出图象中的特征点
4. 求出摄像机的内参数和外参数

标定过程

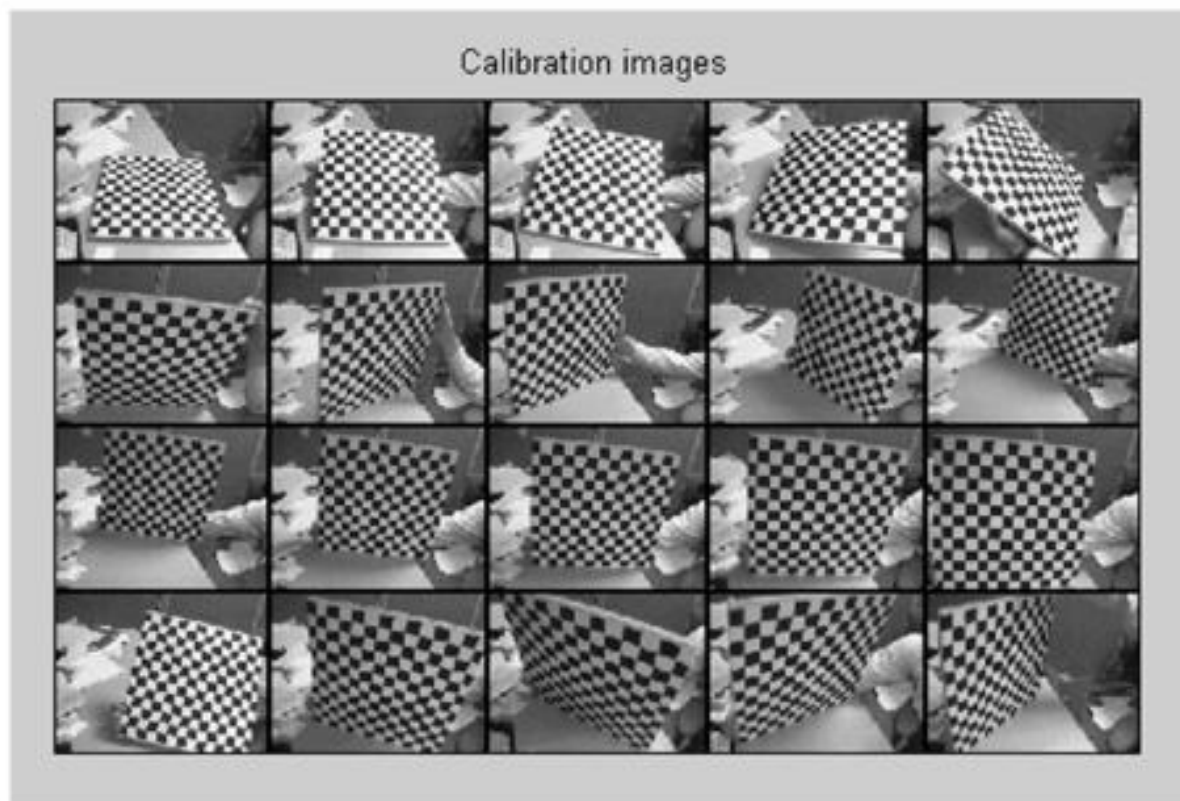
- 打印一张模板并贴在一个平面上

Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1



标定过程

- 从不同角度拍摄若干张模板图像

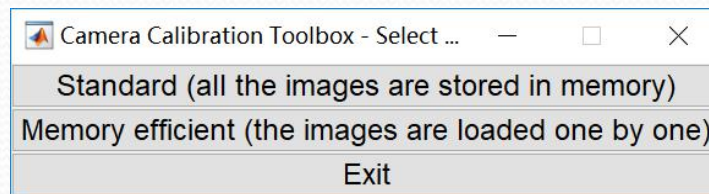


标定过程

Camera Calibration Toolbox for Matlab

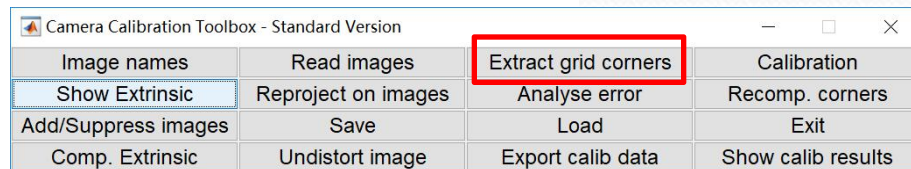
Jean-Yves Bouguet

http://www.vision.caltech.edu/bouguetj/calib_doc/download/index.html 下载好后解压到toolbox 设置
matlab软件Set Path路径 command window输入calib

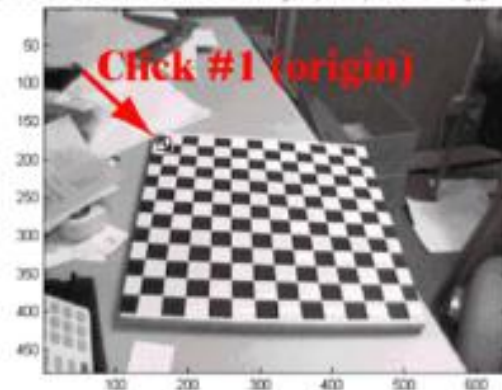


标定过程

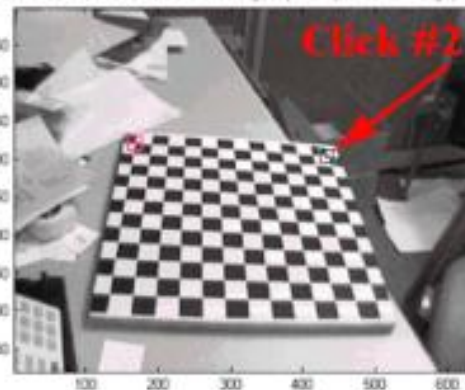
- 检测图像中的特征点



Click on the four extreme corners of the rectangular pattern (first corner = origin) ... Image 1



Click on the four extreme corners of the rectangular pattern (first corner = origin) ... Image 1



Click on the four extreme corners of the rectangular pattern (first corner = origin) ... Image 1

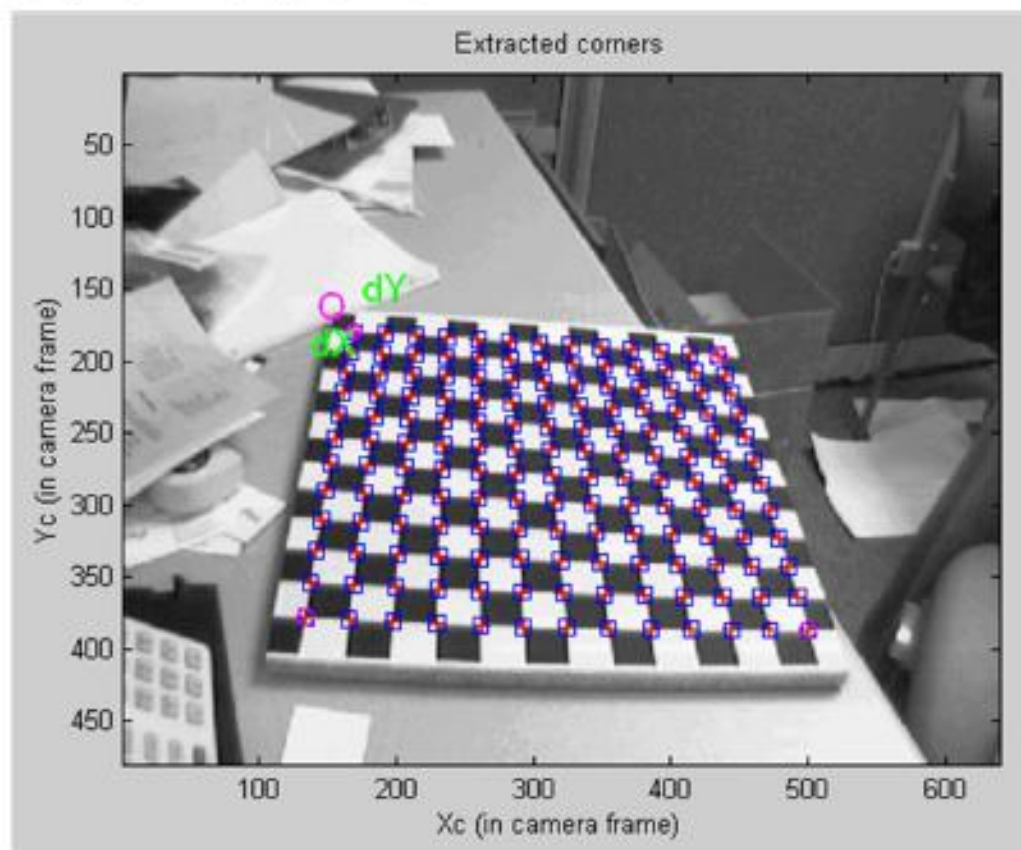


Click on the four extreme corners of the rectangular pattern (first corner = origin) ... Image 1



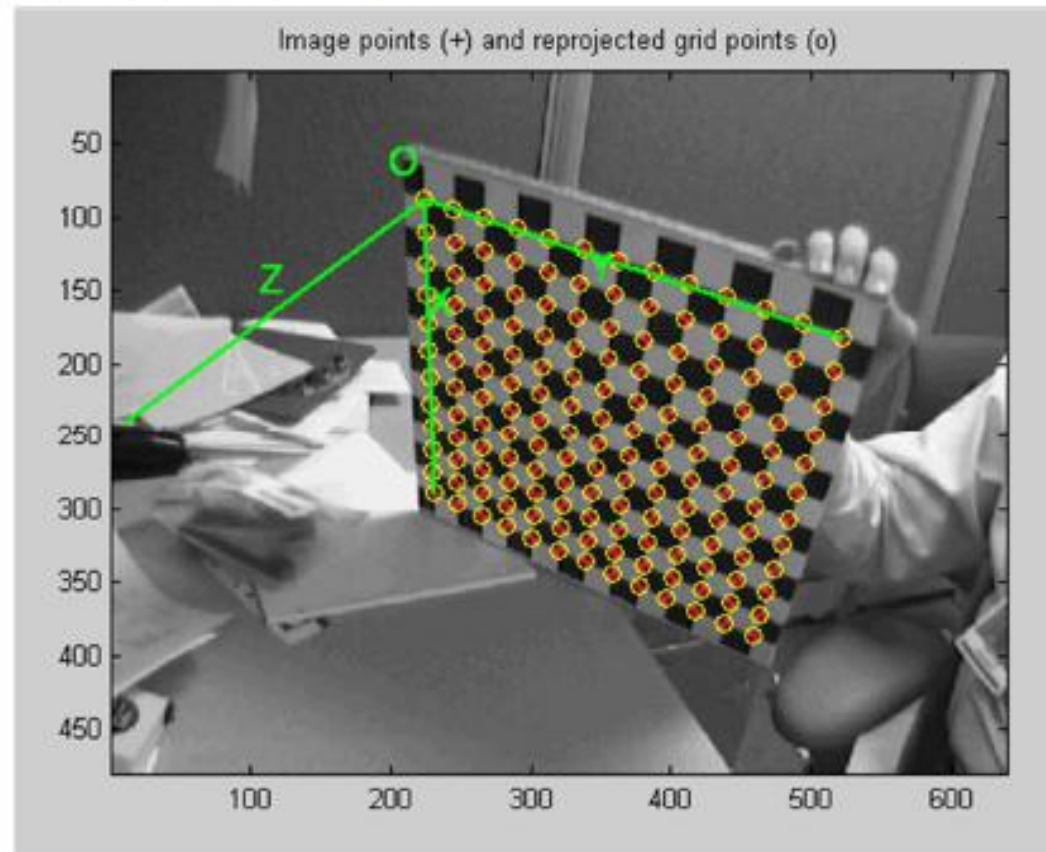
标定过程

- 检测图像中的特征点



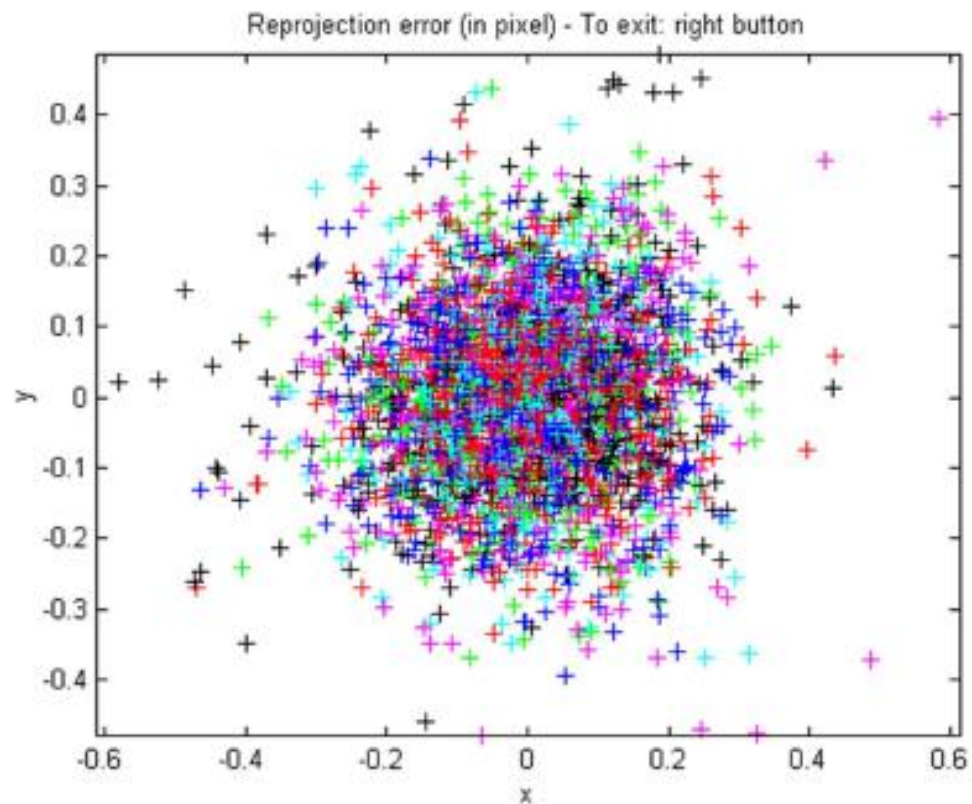
标定过程

- 求解相机内外参数

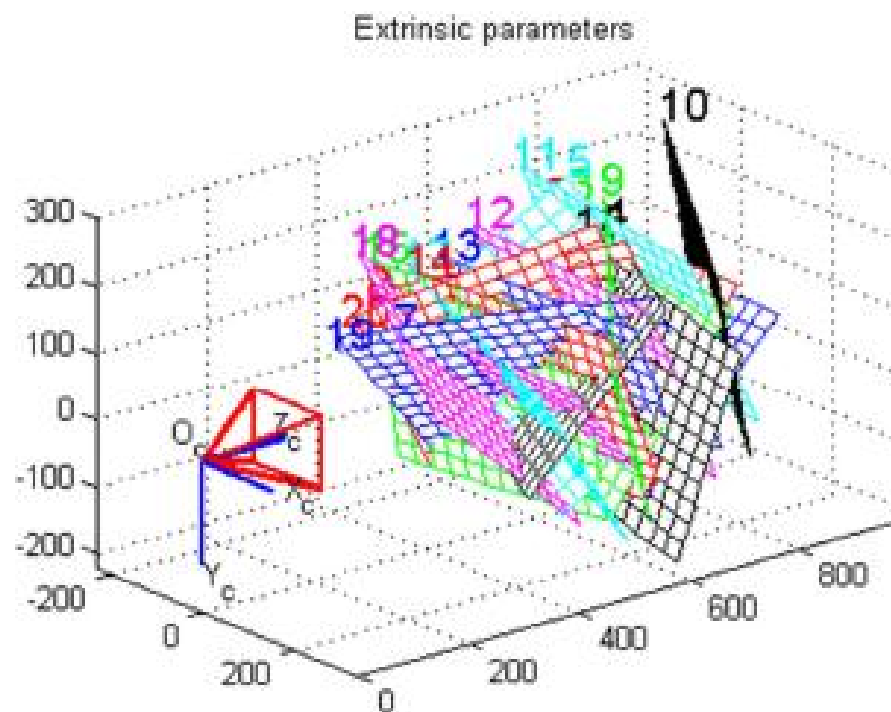


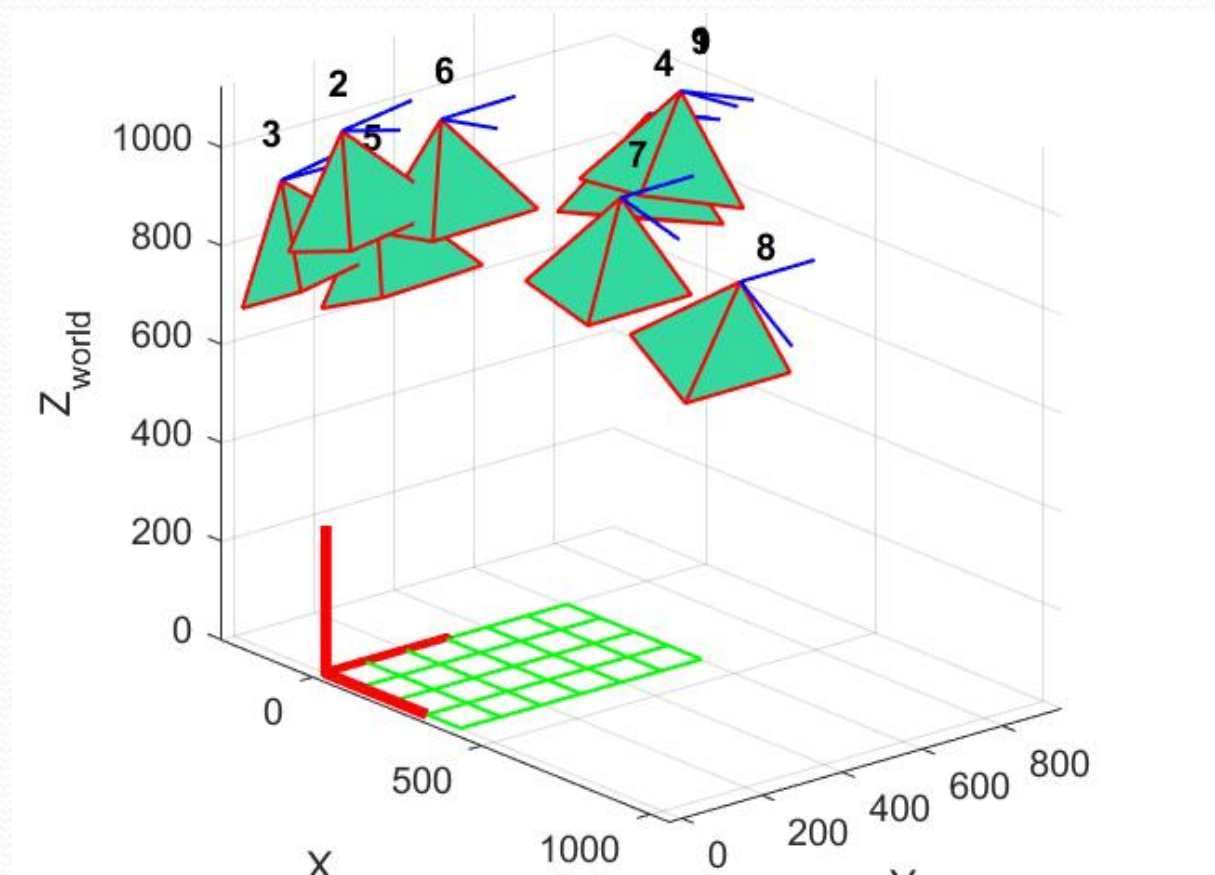
标定过程

- 重投影误差

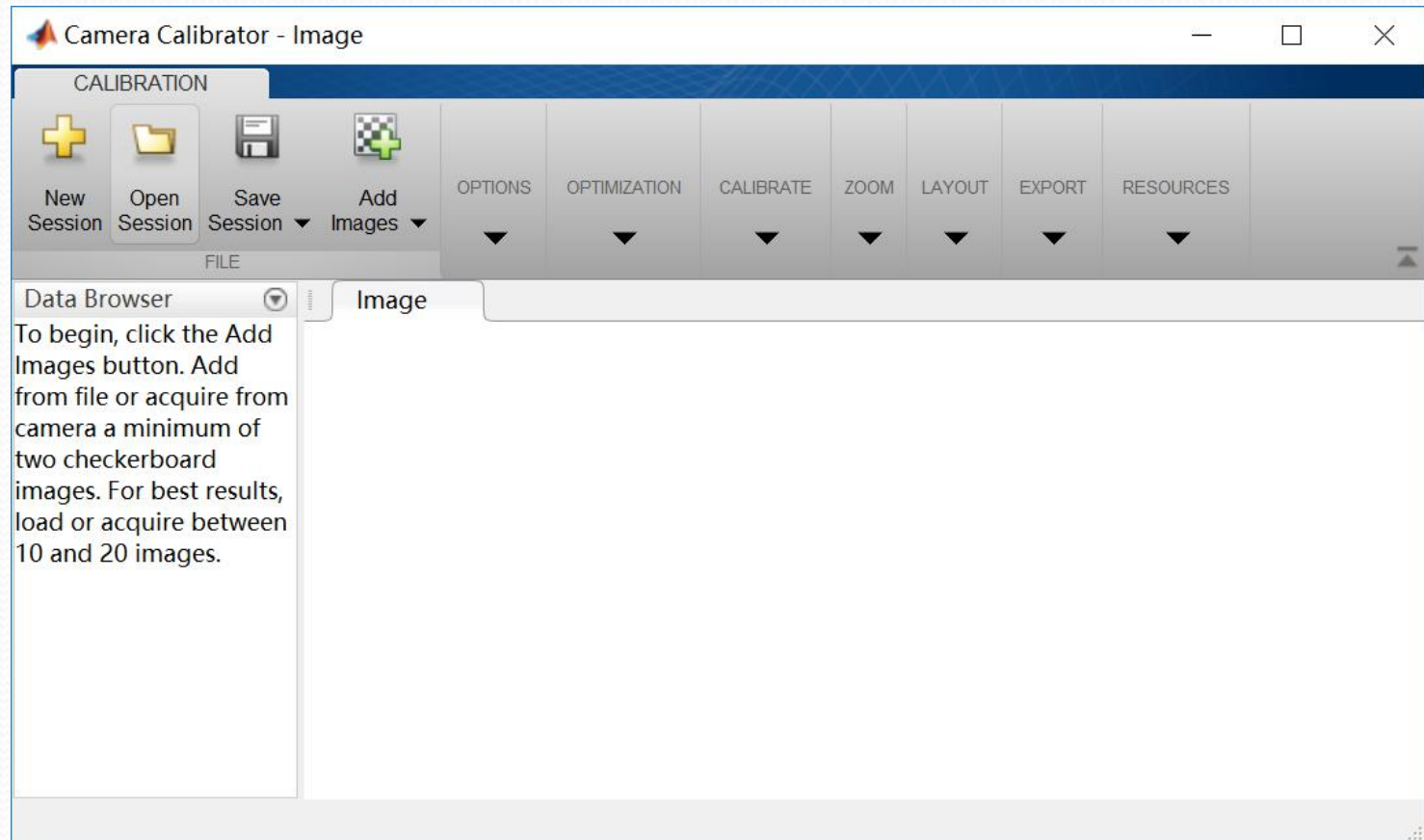


标定过程





使用APPS内的Camera Calibrator进行标定



实验内容

- 1、matlab读取一张图片，在图片左下角用红色字体添加自己的姓名，学号，然后保存
- 2、分别用下载的toolbox和APPS内的工具进行标定，给出相机内部参数和重投影误差截屏，比较分析标定结果

实验报告word或者pdf文件发送至3L3T3T@163.COM

作业命名方式 学号+姓名