SLAM14讲—05相机与图像

5.1相机模型

针孔模型最常见，它描述了一束光线通过针孔之后，在针孔背面投影成像的关系。同时，由于相机镜头上的透镜的存在，会使得光线投影到成像平面的过程中产生畸变。这两个模型能够把外部的三维点投影到相机内部成像平面，构成了相机的内参数。

5.1.1针孔相机模型

1.成像过程

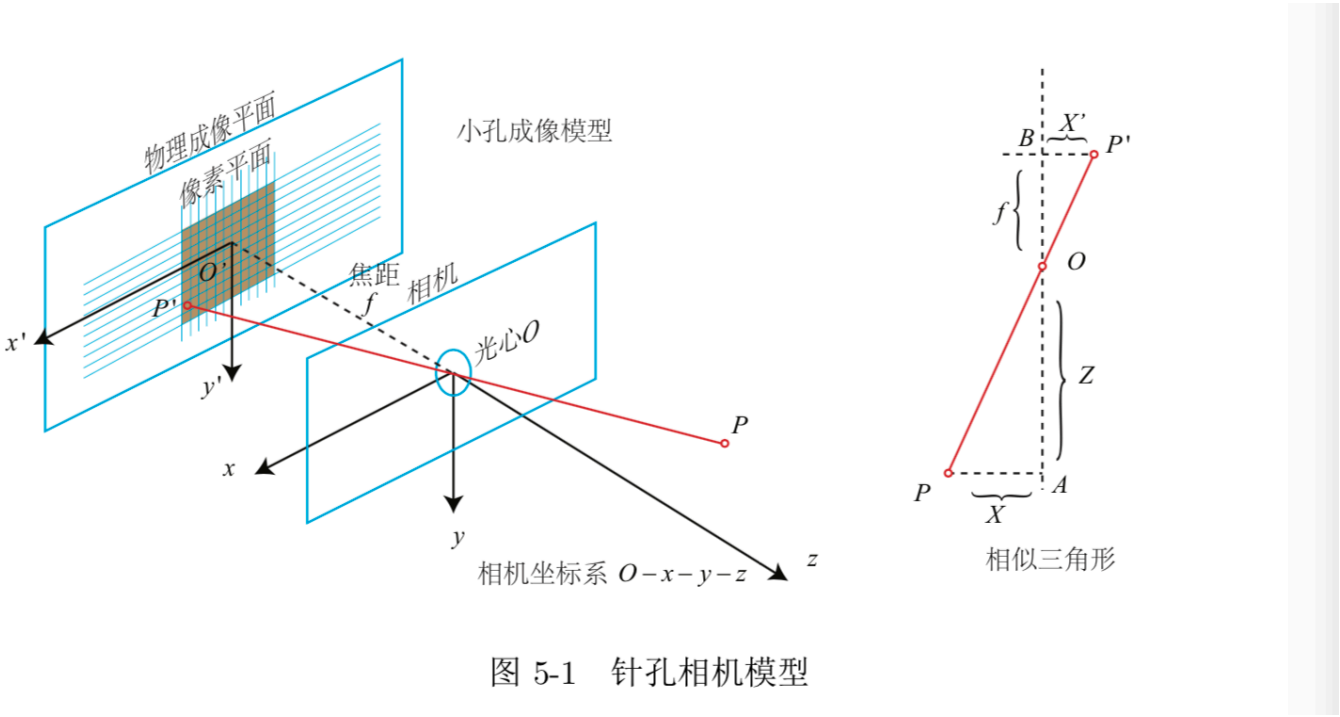


图 5-1　针孔相机模型

设O-x-y-z为相机坐标系，习惯上我们让z轴指向相机前方，x向右，y向下。O为摄像机的光心，也是针孔模型中的针孔。现实世界的空间点P，经过小孔O投影之后，落在物理成像平面O'-x'-y'上，成像点为P'。设P的坐标为$[X,Y,Z]^T$，P'为$[X',Y,Z']^T$，并且设物理成像平面到小孔的距离为f（焦距）。那么，根据三角形相似关系，有：

$$\frac{Z}{f} = -\frac{X}{X'} = -\frac{Y}{Y'}. \tag{5.1}$$

其中，负号表示成的像是倒立的。为了简化模型，可以把成像平面对称到相机前方，和三维空间点一起放在摄像机坐标系的同一侧，如下图中间所示，可以把公式中的负号去掉，使式子更加简洁：

$$\frac{Z}{f} = \frac{X}{X'} = \frac{Y}{Y'}. \tag{5.2}$$

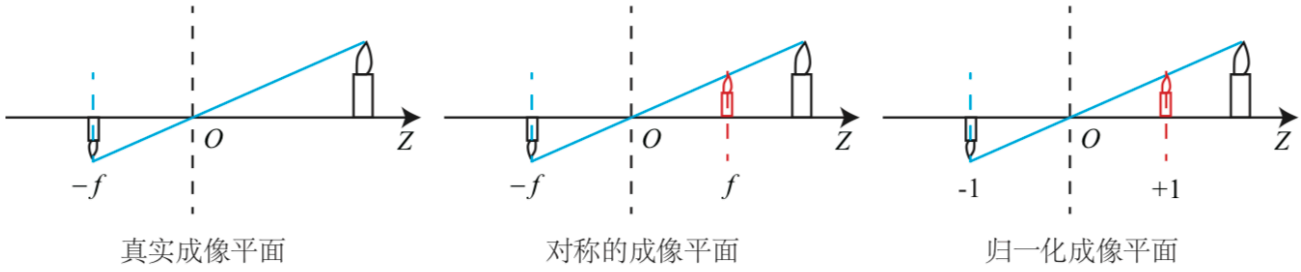


图 5-2　真实成像平面，对称成像平面，归一化成像平面的图示。

整理可得：

$$\begin{aligned} X' &= f\frac{X}{Z} \\ Y' &= f\frac{Y}{Z} \end{aligned} \quad . \tag{5.3}$$

大多数相机的输出并不是倒像，而是经过预处理后正着的像。故在不引起歧义的情况下，我们称此时也为针孔模型。

2.图像像素、内参

在相机中，我们需要再成像平面上对像进行采样和量化。为了描述传感器将感受到的光线转换成图像像素的过程，我们设在物理成像平面上固定着一个像素平面o-u-v。我们在像素平面得到了P'的像素坐标：$[u,v]^T$。
像素坐标系通常的定义方式是：原点o'位于图像的左上角，u轴向右与x轴平行，v轴向下与y轴平行。像素坐标系与成像平面之间，相差了一个缩放和一个原点的平移。我们设像素坐标在u轴上缩放了α倍，在v上缩放了β倍；同时，原点平移了$[c_x,c_y]^T$。那么，此时有：

$$\begin{cases} u = \alpha X' + c_x \\ v = \beta Y' + c_y \end{cases}. \tag{5.4}$$

代入式（5.3）并把 αf 合并成 f_x，把 βf 合并成 f_y，得：

$$\begin{cases} u = f_x \frac{X}{Z} + c_x \\ v = f_y \frac{Y}{Z} + c_y \end{cases}. \tag{5.5}$$

其中，f 的单位为米，α, β 的单位为像素每米，所以 f_x, f_y 的单位为像素。把该式写成矩阵形式，会更加简洁，不过左侧需要用到齐次坐标：

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \triangleq \frac{1}{Z} \boldsymbol{KP}. \tag{5.6}$$

我们按照传统的习惯，把 Z 挪到左侧：

$$Z \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \triangleq \boldsymbol{KP}. \tag{5.7}$$

该式中，我们把中间的量组成的矩阵称为相机的内参数矩阵**K**。内参在相机出厂后是固定的，有的相机生产厂商会告诉你相机的内参，有的则需要你自己确定，即标定。

3.外参

（5.6）中我们使用的是P在相机坐标系下的坐标。由于相机在运动，故P的相机坐标应该是它的世界坐标（记为$\mathbf{P}_{\mathrm{W}}$），根据相机当前位姿变换到相机坐标系下的结果。相机的位姿由它的旋转矩阵**R**和平移向量**t**来描述。那么有：

$$Z\boldsymbol{P}_{uv} = Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \boldsymbol{K}(\boldsymbol{RP}_w + \boldsymbol{t}) = \boldsymbol{KTP}_w. \tag{5.8}$$

最后一个等号的得出是一个齐次坐标到非齐次坐标的转换，可见（3.9）式。其中，相机的位姿R，t又称为相机的外参数。外参数会随着相机运动发生改变，同时也是SLAM种待估计的目标，代表着机器人的轨迹。
上式两侧都是齐次坐标，因为齐次坐标乘上非零常数后表达同样的含义，所以可以简单地把Z去掉：

$$\boldsymbol{P}_{uv} = \boldsymbol{KTP}_w. \tag{5.9}$$

但这样等号意义就变了，成为了齐次坐标下相等的概念，相差了一个非零常数。
右侧的$\mathbf{TP}_{\mathrm{W}}$表示把一个世界坐标系下的齐次坐标，变换到相机坐标系下。为了使它与**K**相乘，需要取它的前三维组成向量——因为$\mathbf{TP}_{\mathrm{W}}$最后一维为1.此时对于这个三维向量，我们还可以按照齐次坐标的方式，把最后一维进行归一化处理，得到了P在相机归一化平上的投影：

$$\tilde{\boldsymbol{P}}_c = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = (\boldsymbol{T}\boldsymbol{P}_w)_{(1:3)}, \quad \boldsymbol{P}_c = \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix}. \tag{5.10}$$

这时$\mathbf{P}_c$可以看成一个二维的齐次坐标，称为**归一化坐标**。它位于相机前方z=1处的平面上，即**归一化平面**。由于$\mathbf{P}_c$经过内参之后就得到了像素坐标，所以我们可以把像素坐标$[u,v]^T$看成对归一化平面上的点进行量化测量的结果。

5.1.2畸变

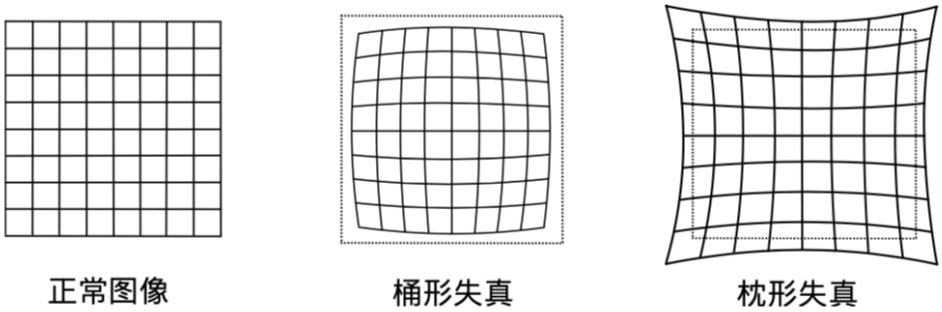


图 5-3　径向畸变的两种类型。

- 径向畸变：由透镜形状引起的畸变
  - 桶形畸变：由于图像放大率随着离光轴的距离增加而减小
  - 枕形畸变：由于图像放大率随着离光轴的距离增加而增加
- 切向畸变：在相机的组装过程中由于不能使得透镜和成像面严格平行时引起的畸变

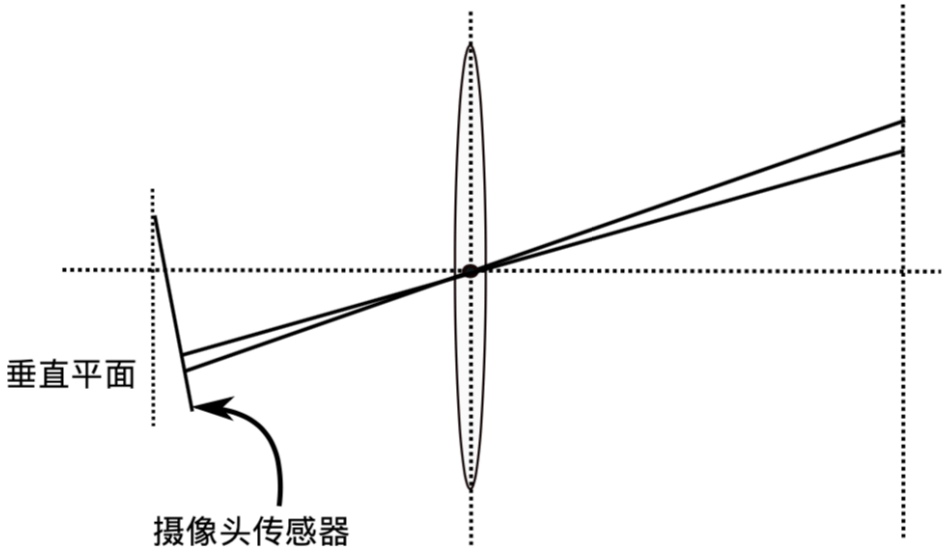


图 5-4　切向畸变来源示意图。

用数学形式描述径向畸变和切向畸变。假设平面上任意一点p坐标为$[x,y]^T$，或$[r,\theta]^T$。**径向畸变可以看成坐标点沿着长度方向发生了变化$\delta\mathbf{r}$；切向畸变可以看成坐标点沿着切线方向发生了变化，即水平家教发生了变化$\delta\boldsymbol{\theta}$**。

- 对于**径向畸变**，我们用一个多项式函数来描述畸变前后的坐标变化：这类畸变可以用和距中心距离有关的二次及高次多项式函数进行纠正：

$$\begin{aligned} x_{corrected} &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{corrected} &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned}. \tag{5.11}$$

其中$[x,y]^T$是未纠正的点的坐标，$[x_{corrected},y_{corrected}]^T$是纠正后的点的坐标，都是归一化平面上的点。

- 对于畸变较小的图像中心区域，畸变纠正主要是k_1起作用；而对于畸变较大的边缘区域主要是k_2起作用；对于畸变很大的摄像头，如鱼眼镜头，可以加入k_3畸变项进行纠正。

- 对于切向畸变，可以用p_1和p_2进行纠正：

$$\begin{aligned} x_{corrected} &= x + 2p_1xy + p_2(r^2 + 2x^2) \\ y_{corrected} &= y + p_1(r^2 + 2y^2) + 2p_2xy \end{aligned}. \tag{5.12}$$

- 利用五个畸变系数找到这个的在像素平面上的正确位置：

1. 将三维空间点投影到归一化图像平面。设它的归一化坐标为 $[x, y]^T$。
2. 对归一化平面上的点进行径向畸变和切向畸变纠正。

$$\begin{cases} x_{corrected} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_1xy + p_2(r^2 + 2x^2) \\ y_{corrected} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) + p_1(r^2 + 2y^2) + 2p_2xy \end{cases}. \tag{5.13}$$

3. 将纠正后的点通过内参数矩阵投影到像素平面，得到该点在图像上的正确位置。

$$\begin{cases} u = f_x x_{corrected} + c_x \\ v = f_y y_{corrected} + c_y \end{cases}. \tag{5.14}$$

- 去畸变处理方法（Undistort，或称畸变校正）
  - 先对整张图像进行去畸变，得到去畸变后的图像，然后讨论图像上的点的空间位置
  - 也考虑图像中的某个点，然后按照去畸变方程，讨论它去畸变后的空间位置

5.1.3双目相机模型

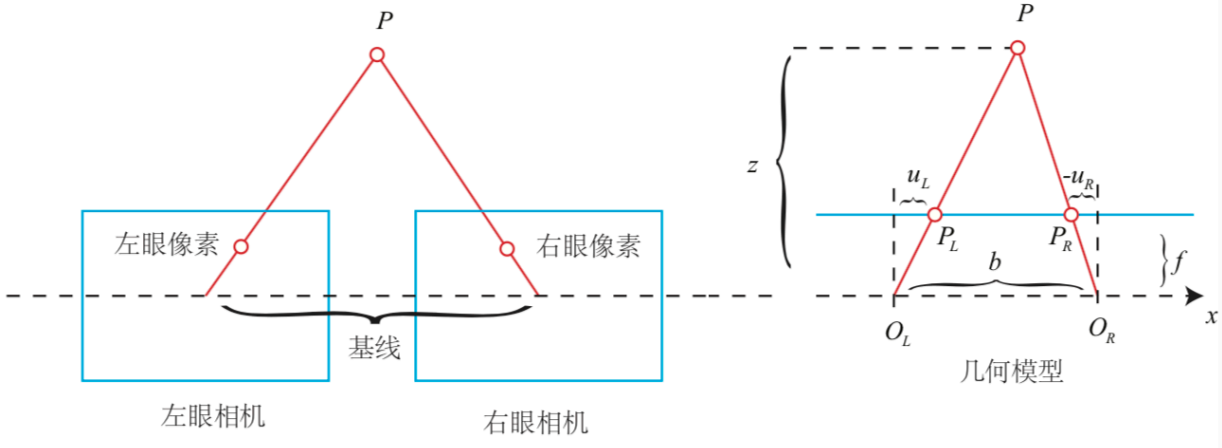


图 5-6　双目相机的成像模型。O_L, O_R 为左右光圈中心，蓝色框为成像平面，f 为焦距。u_L 和 u_R 为成像平面的坐标。请注意按照图中坐标定义，u_R 应该是负数，所以图中标出的距离为 $-u_R$。

两个相机的光圈中心都位于**x**轴上。它们的距离称为双目相机的基线（Baseline，记作b），是双目的重要参数。现在考虑一个空间点P，它在左眼和右眼各成一像，记作P_L，P_R，（假设理想情况下）像在成像平面上左侧的坐标

为u_L，右侧的坐标为u_R。根据相似三角形有：

$$\frac{z-f}{z}=\frac{b-u_L+u_R}{b}. \tag{5.15}$$

稍加整理，得：

$$z=\frac{fb}{d},\quad d=u_L-u_R. \tag{5.16}$$

这里d为左右图的横坐标之差，称为视差（Disparity）。根据视差，我们可以估计一个像素相机的距离。视差与距离成反比：视差越大，距离越近。 视差最小为一个像素，所以双目的深度存在一个理论上的最大值。当基线越长时，双目最大能测到的距离就会变远。
视差的计算比较难。计算每个像素的深度时，计算量与精度都是问题，只有在图像纹理丰富的地方才能计算视差。

5.1.4RGB-D相机模型

可以主动测量每个像素的深度。目前RGB-D相机按照原理可以分为两大类：

- 1.通过红外结构光来测量像素距离的。例如Kinect 1代、Project Tango1代、Intel RealSense等；
- 2.通过飞行时间法原理测量像素距离的。例如Kinect2代和一些现有的ToF传感器等。

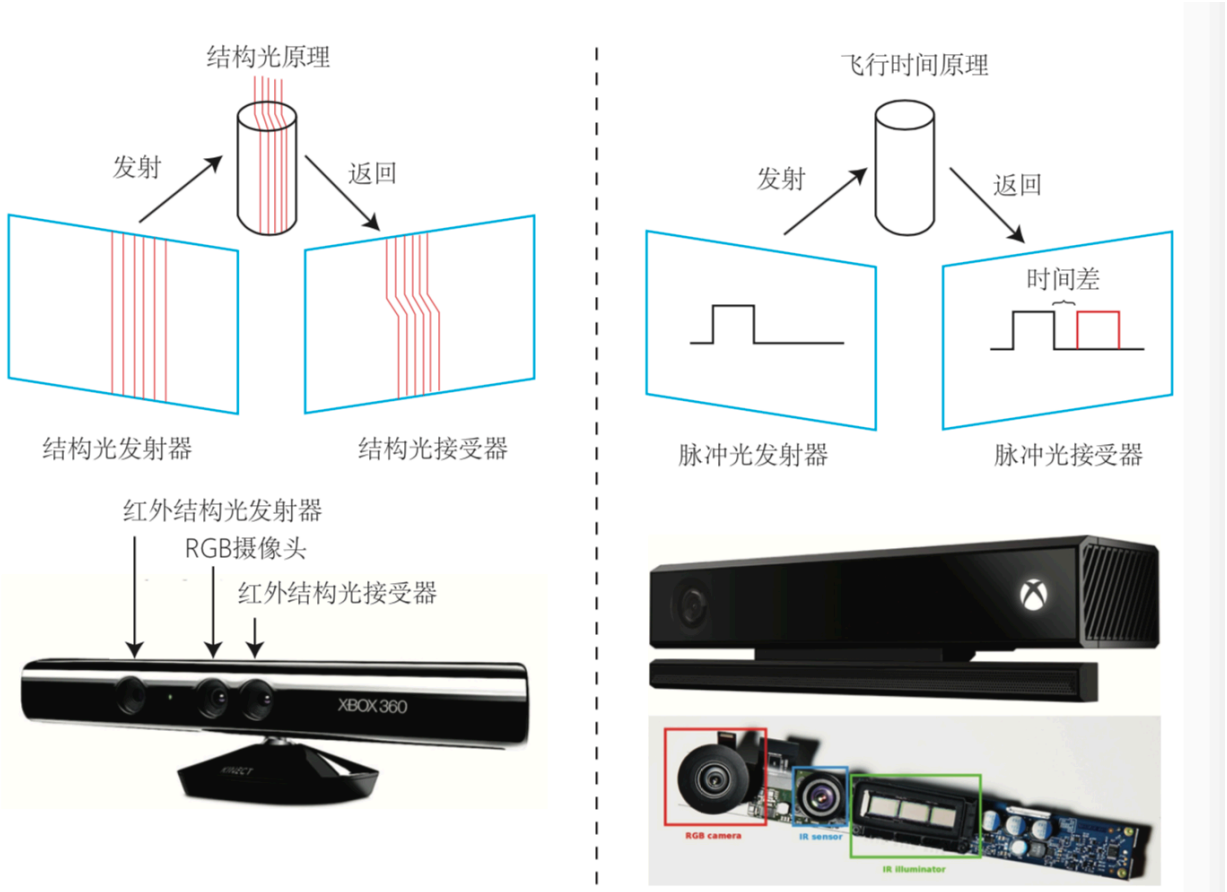


图 5-7　RGB-D 相机原理示意图

- 优势：在测量深度之后，RGB-D相机通常按照生产时的各个相机摆放位置，自己完成深度与彩色图像素之间的配对，输出一一对应的彩色图和深度图。我们可以在同一个图像位置，读取到色彩信息和距离信息，计算像素的3D相机坐标，生成点云。对RGB-D数据，既可以在图像层面进行处理，也可以在点云层面处理。
- 劣势：使用范围有限。容易受到日光或其他传感器发射的红外光干扰，使用多个时也会相互干扰，而且无法测量透射材质的物体的位置。此外成本、功耗也有劣势。

5.2图像

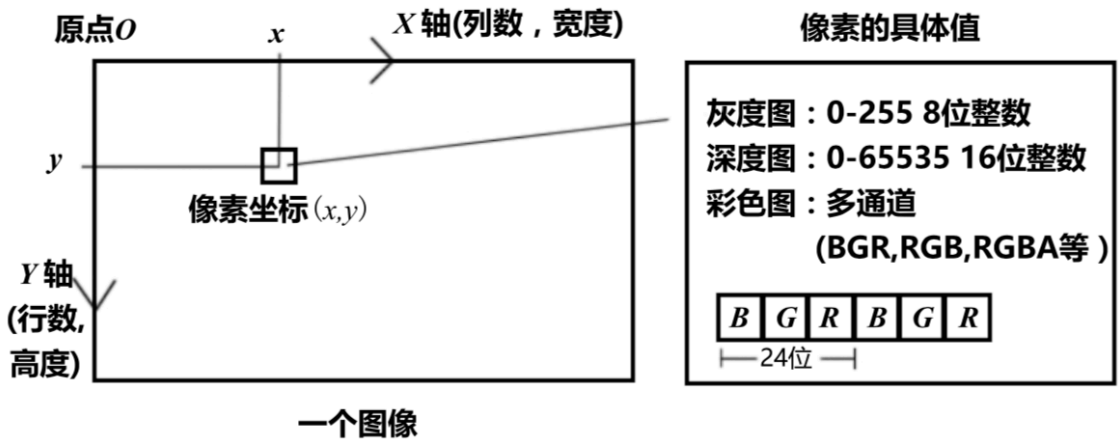
5.2.1计算机中图像的表示

- 灰度图
  一张宽度为640，高度为480分辨率的灰度图表示如下：

```
unsigned char image[480][640];
```

图像以一个二维数组形式存储。第一个下标为数组的行（图像的高度），第二个下标为列（图像的宽度）。

- 图像的坐标



  访问x,y处的像素如下：

```
unsigned char pixel = image[y][x];
```

RGB-D相机的量程通常在十几米范围左右，人们会曹勇十六位整数（unsigned short）来记录一个深度图的信息，即位于0至65536之间的值。

- 彩色图
  计算机中用红绿蓝的组合来表示任何一个色彩。，R,G,B三个数值即三个通道。在OpenCV的彩色图像中，通道的默认顺序是B,G,R。即24位的像素，依次8位表示蓝色、绿色和红色。若需要表示透明度，可以用R,G,B,A四个通道来表示。

5.3实践：图像的存取与访问

5.3.2操作OpenCV图像

读取图像的命令为：

```
image = cv::imread(argv[1]);
```

即程序从argv[1]（命令行的第一个参数中读取图像位置），故编译之后输入以下命令：

```
build/image_basics xxx.jpg
```

5.4实践：拼接点云

pcl安装详见https://blog.csdn.net/dantengc/article/details/78446600
https://blog.csdn.net/qq_43145072/article/details/85953948
注意：depth和color文件夹、pose.txt都放在build下面
终端直接显示pcd文件命令如下：

```
pcl_viewer xxx.pcd
```