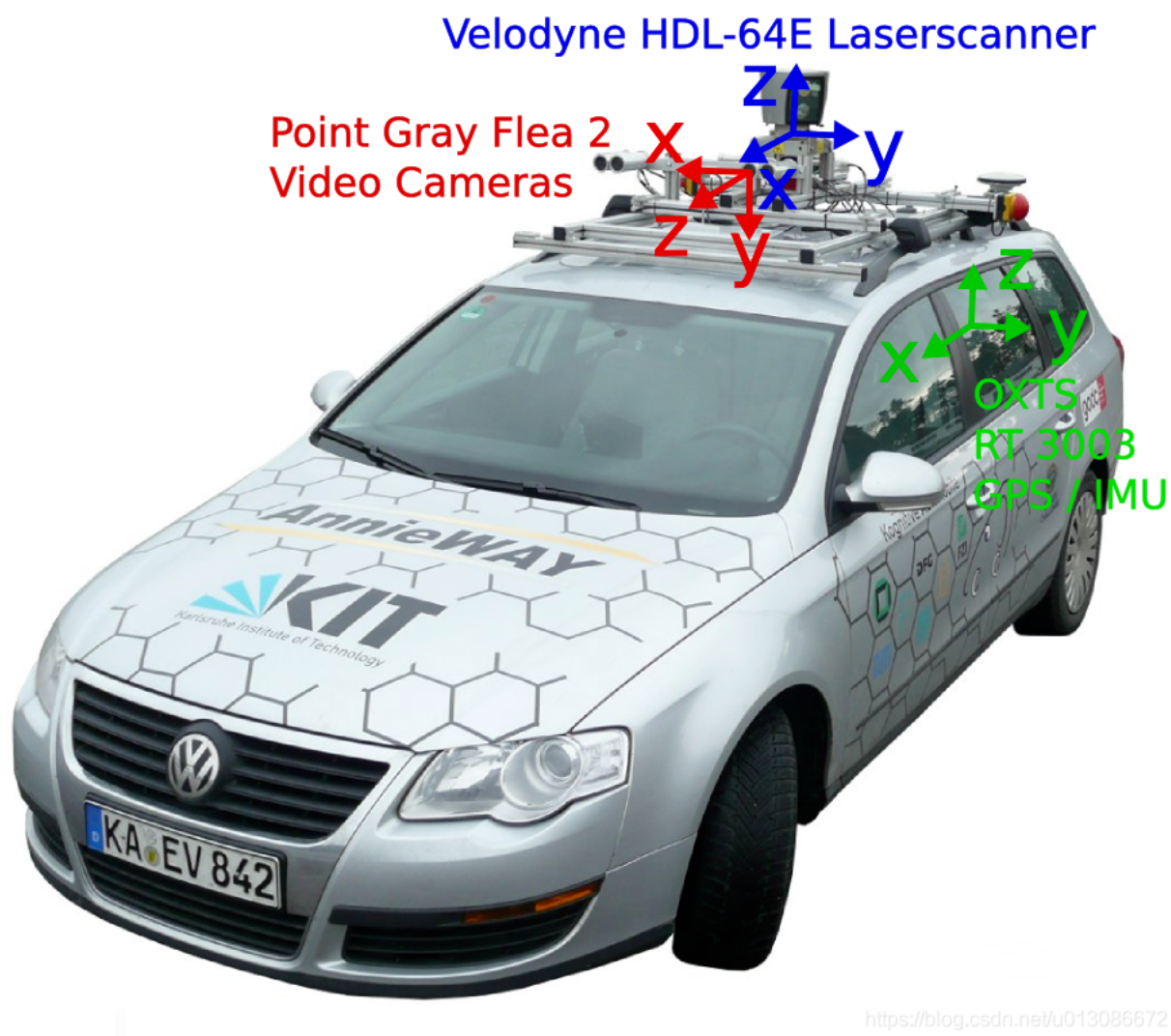


# KITTI数据集下载及解析



## KITTI数据集下载及解析

W.P. Xiao, Vision group, SHUSV

|   | 版本    | 更新时间       | 更新内容   | 作者        |
|---|-------|------------|--------|-----------|
| 1 | V 1.0 | 2020.01.09 | 完成主体内容 | W.P. Xiao |
| 2 |       |            |        |           |

### 文章目录

#### KITTI Dataset

##### 1 简介

- 1.1 数据采集平台
- 1.2 坐标系

##### 2 数据解析

- 2.1 image文件
- 2.2 velodyne文件
- 2.3 calib文件
- 2.4 label文件

##### 3 KITTI可视化

# KITTI Dataset

## 1 简介

KITTI数据集由德国卡尔斯鲁厄理工学院和丰田美国技术研究院联合创办，是目前国际上最大的自动驾驶场景下的计算机视觉算法评测数据集。数据集用于评测立体图像(stereo)，光流(optical flow)，视觉测距(visual odometry)，3D物体检测(object detection)和3D跟踪(tracking)等计算机视觉技术在车载环境下的性能。KITTI包含市区、乡村和高速公路等场景采集的真实图像数据，每张图像中最多达15辆车和30个行人，还有各种程度的截断。3D目标检测数据集由**7481**个训练图像和**7518**个测试图像以及相应的点云数据组成，包括总共**80256**个标记对象。

下图红色框标记的为我们需要的数据，分别是彩色图像数据（12GB）、点云数据（29GB）、相机矫正数据（16MB）、标签数据（5MB）。彩色图像数据、点云数据、相机矫正数据均包含training（7481）和testing（7518）两个部分，标签数据只有training数据。

- [Download left color images of object data set \(12 GB\)](#)
- [Download right color images, if you want to use stereo information \(12 GB\)](#)
- [Download the 3 temporally preceding frames \(left color\) \(36 GB\)](#)
- [Download the 3 temporally preceding frames \(right color\) \(36 GB\)](#)
- [Download Velodyne point clouds, if you want to use laser information \(29 GB\)](#)
- [Download camera calibration matrices of object data set \(16 MB\)](#)
- [Download training labels of object data set \(5 MB\)](#)
- [Download object development kit \(1 MB\)](#) (including 3D object detection and [bird's eye view](#) evaluation code)
- [Download pre-trained LSVM baseline models \(5 MB\)](#) used in [Joint 3D Estimation of Objects and Scene Layout \(NIPS 2011\)](#). These models are referred to as LSVM-MDPM-sv (supervised version) and LSVM-MDPM-us (unsupervised version) in the tables below.
- [Download reference detections \(L-SVM\) for training and test set \(800 MB\)](#)
- Qianli Liao (NYU) has put together [code to convert from KITTI to PASCAL VOC file format](#) (documentation included, requires Emacs).
- Karl Rosaen (U.Mich) has released [code to convert between KITTI, KITTI tracking, Pascal VOC, Udacity, CrowdAI and AUTTI formats](#).
- We thank [David Stutz](#) and [Bo Li](#) for developing the 3D object detection benchmark.

<https://blog.csdn.net/u013086672>

- KITTI数据集下载官网（不想爬梯子的朋友可移步我的CSDN博客：KITTI数据集下载（百度云））
- 3D Object Detection经典论文整理【分类/下载/代码/笔记】
- 基于OpenPCDet框架的基线模型下载及性能评估

### 1.1 数据采集平台

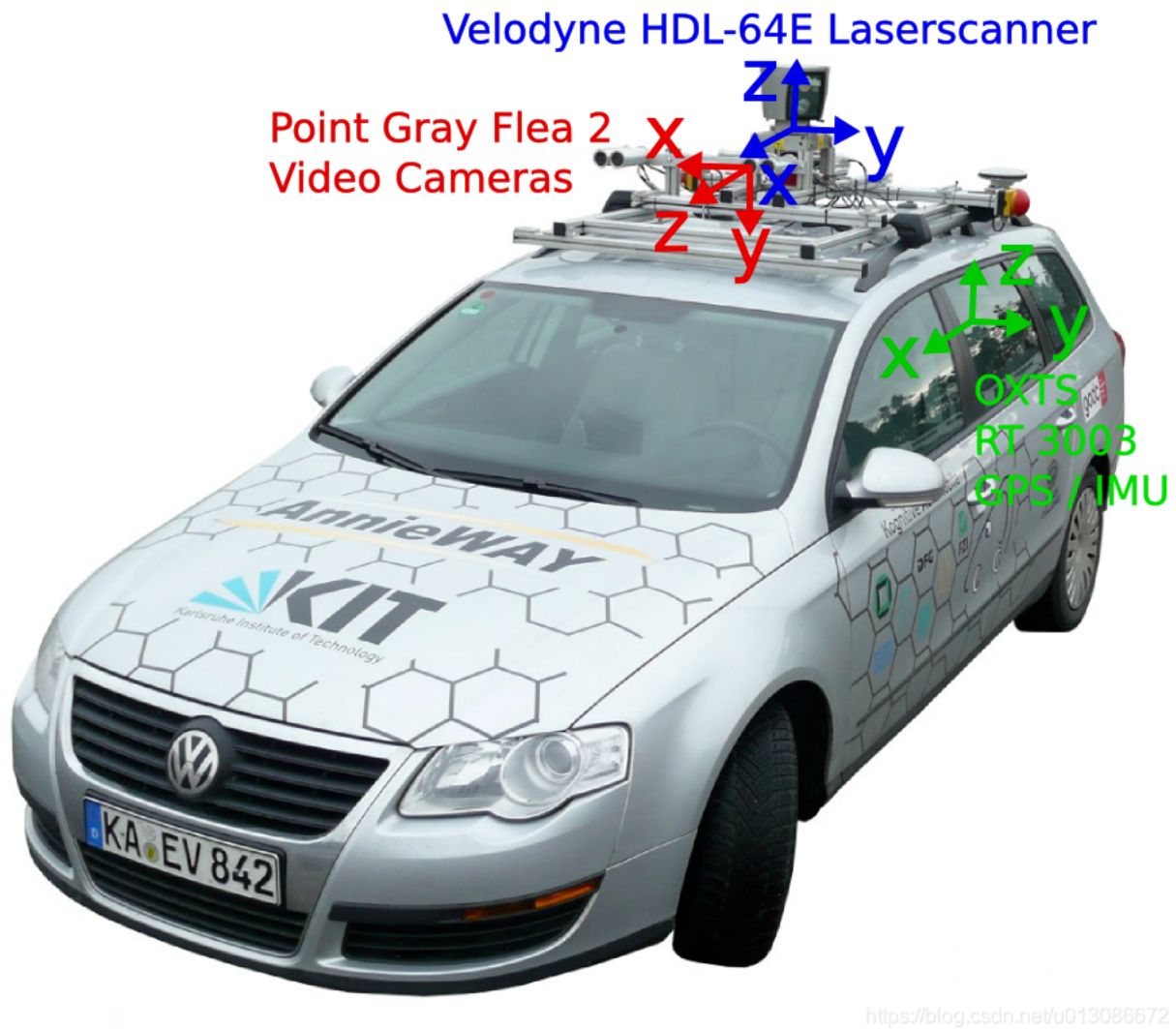


图1.1 3D物体的观察角和方位角

如图1.1所示，KITTI数据集的数据采集平台装配有2个灰度摄像机，2个彩色摄像机，一个Velodyne 64线3D激光雷达，4个光学镜头，以及1个G航系统。具体的传感器参数如下：

- 2 × PointGray Flea2 grayscale cameras (FL2-14S3M-C), 1.4 Megapixels, 1/2" Sony ICX267 CCD, global shutter
- 2 × PointGray Flea2 color cameras (FL2-14S3C-C), 1.4 Megapixels, 1/2" Sony ICX267 CCD, global shutter
- 4 × Edmund Optics lenses, 4mm, opening angle  $\sim 90^\circ$ , vertical opening angle of region of interest (ROI)  $\sim 35^\circ$
- 1 × Velodyne HDL-64E rotating 3D laser scanner, 10 Hz, 64 beams, 0.09 degree angular resolution, 2 cm distance accuracy, collecting  $\sim$  million points/second, field of view:  $360^\circ$  horizontal,  $26.8^\circ$  vertical, range: 120 m
- 1 × OXTS RT3003 inertial and GPS navigation system, 6 axis, 100 Hz, L1/L2 RTK, resolution: 0.02m /  $0.1^\circ$

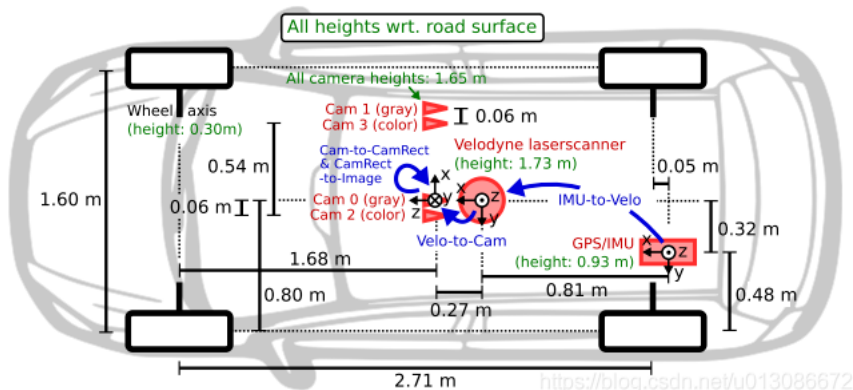
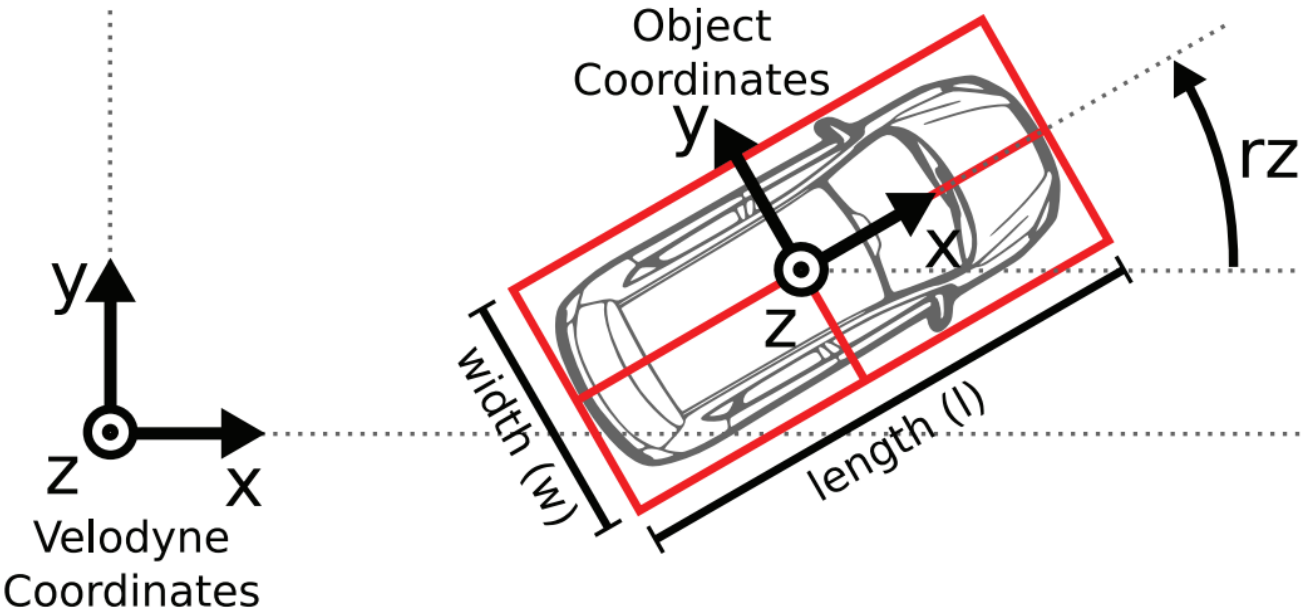


图1.2 传感器布置图

传感器布置平面图如图1.2所示。为了生成双目立体图像，相同类型的摄像头相距54cm安装。由于彩色摄像机的分辨率和对比度不够好，所以还了两个立体灰度摄像机，它和彩色摄像机相距6cm安装。

1.2 坐标系



<https://blog.csdn.net/u01301>

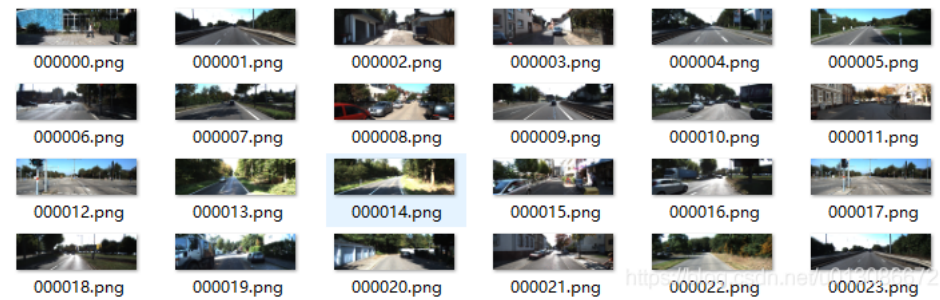
图1.3 物体坐标系

- camera: x = right, y = down, z = forward
- velodyne: x = forward, y = left, z = up
- GPS/IMU: x = forward, y = left, z = up

2 数据解析

2.1 image文件

image文件以8位PNG格式存储，图集如下：



<https://www.cnblogs.com/2>

2.2 velodyne文件

velodyne文件是激光雷达的测量数据（绕其垂直轴（逆时针）连续旋转），以“000001.bin”文件为例，内容如下：

```
1 7b14 4642 1058 b541 9643 0340 0000 0000
2 46b6 4542 1283 b641 3333 0340 0000 0000
3 4e62 4042 9643 b541 b072 0040 cdcc 4c3d
4 8340 3f42 08ac b541 3bdf ff3f 0000 0000
5 e550 4042 022b b841 9cc4 0040 0000 0000
6 10d8 4042 022b ba41 4c37 0140 0000 0000
7 3fb5 3a42 14ae b541 5a64 fb3f 0000 0000
8 7dbf 3942 2731 b641 be9f fa3f 8fc2 f53d
9 cd4c 3842 3f35 b641 4c37 f93f ec51 383e
10 dbf9 3742 a69b b641 c3f5 f83f ec51 383e
11
```

```

11 | 2586 3742 9a99 b741 fed4 f83f 1f85 6b3e
12 |
13 |
14 |

```

点云数据以浮点二进制文件格式存储，每行包含8个数据，每个数据由四位十六进制数表示（浮点数），每个数据通过空格隔开。一个点云数据由8个浮点数数据构成，分别表示点云的**x、y、z、r（强度 or 反射值）**，点云的存储方式如下表所示：

| pointcloud-1 |   |   |   | pointcloud-2 |   |   |   |
|--------------|---|---|---|--------------|---|---|---|
| x            | y | z | r | x            | y | z | r |
| pointcloud-3 |   |   |   | pointcloud-4 |   |   |   |
| x            | y | z | r | x            | y | z | r |
| ...          |   |   |   | pointcloud-n |   |   |   |
| x            | y | z | r | x            | y | z | r |

## 2.3 calib文件

calib文件是相机、雷达、惯导等传感器的矫正数据。以“000001.txt”文件为例，内容如下：

```

P0: 7.215377000000e+02 0.000000000000e+00 6.095593000000e+02 0.000000000000e+00 0.000000000000e+00 7.215377000000e+02
1.728540000000e+02 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00 1.000000000000e+00 0.000000000000e+00

P1: 7.215377000000e+02 0.000000000000e+00 6.095593000000e+02 -3.875744000000e+02 0.000000000000e+00 7.215377000000e+02
1.728540000000e+02 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00 1.000000000000e+00 0.000000000000e+00

P2: 7.215377000000e+02 0.000000000000e+00 6.095593000000e+02 4.485728000000e+01 0.000000000000e+00 7.215377000000e+02
1.728540000000e+02 2.163791000000e-01 0.000000000000e+00 0.000000000000e+00 1.000000000000e+00 2.745884000000e-03

P3: 7.215377000000e+02 0.000000000000e+00 6.095593000000e+02 -3.395242000000e+02 0.000000000000e+00 7.215377000000e+02
1.728540000000e+02 2.199936000000e+00 0.000000000000e+00 0.000000000000e+00 1.000000000000e+00 2.729905000000e-03

R0_rect: 9.999239000000e-01 9.837760000000e-03 -7.445048000000e-03 -9.869795000000e-03 9.999421000000e-01 -4.278459000000e-03
7.402527000000e-03 4.351614000000e-03 9.999631000000e-01

Tr_velo_to_cam: 7.533745000000e-03 -9.999714000000e-01 -6.166020000000e-04 -4.069766000000e-03 1.480249000000e-02
7.280733000000e-04 -9.998902000000e-01 -7.631618000000e-02 9.998621000000e-01 7.523790000000e-03 1.480755000000e-02
-2.717806000000e-01

Tr_imu_to_velo: 9.999976000000e-01 7.553071000000e-04 -2.035826000000e-03 -8.086759000000e-01 -7.854027000000e-04 9.998898000
0.01 -1.482298000000e-02 3.195559000000e-01 2.024406000000e-03 1.482454000000e-02 9.998881000000e-01 -7.997231000000e-01

```

文件中每一行代表一个参数名，冒号后面是参数矩阵，具体如下：

- **P0~P4** ( $\mathbf{P}_{\text{rect}}^{(i)}$ )：矫正后的相机投影矩阵 ( $\mathbf{R}^{3 \times 4}$ )

0、1、2、3 代表相机的编号，0表示左边灰度相机，1右边灰度相机，2左边彩色相机，3右边彩色相机。

$$\mathbf{T}_{\text{velo}}^{\text{cam}} = \begin{pmatrix} \mathbf{f}_u^{(i)} & 0 & \mathbf{c}_u^{(i)} & -\mathbf{f}_u^{(i)} \mathbf{b}_x^{(i)} \\ 0 & \mathbf{f}_v^{(i)} & \mathbf{c}_v^{(i)} & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

其中 $\mathbf{b}^{(i)}$  x表示相对于参考摄像机0的基准值（以米为单位）。

- **R0\_rect** ( $\mathbf{R}_{\text{rect}}^{(i)}$ )：矫正后的相机旋转矩阵 ( $\mathbf{R}^{3 \times 3}$ )

在实际计算时，需要将该 $3 \times 3$ 的矩阵扩展为 $4 \times 4$ 的矩阵，方法为在第四行和第四列添加全为0的向量，并且将（4， 4）的索引值设为1。

- **Tr\_velo\_to\_cam** ( $\mathbf{T}_{\text{velo}}^{\text{cam}}$ )：从雷达到相机的旋转平移矩阵 ( $\mathbf{R}^{3 \times 4}$ )

在实际计算时，需要将该 $3 \times 4$ 的矩阵扩展为 $4 \times 4$ 的矩阵，方法为增加第四行向量[0,0,0,1]。这个矩阵其实包括了两个部分，其一是 $3 \times 3$ 的旋转矩阵，是 $1 \times 3$ 的平移向量。



$$\mathbf{T}_{\text{velo}}^{\text{cam}} = \begin{pmatrix} \mathbf{R}_{\text{velo}}^{\text{cam}} & \mathbf{t}_{\text{velo}}^{\text{cam}} \\ 0 & 1 \end{pmatrix}$$

$\mathbf{R}_{\text{velo}}^{\text{cam}} \in \mathbb{R}^{3 \times 3}$  ...rotation matrix : velodyne  $\rightarrow$  camera  
 $\mathbf{t}_{\text{velo}}^{\text{cam}} \in \mathbb{R}^{3 \times 1}$  ...translation vector : velodyne  $\rightarrow$  camera

- $\text{Tr}_{\text{imu\_to\_velo}}$  ( $\mathbf{T}_{\text{imu}}^{\text{velo}}$ ) : 从惯导或GPS装置到相机的旋转平移矩阵 ( $\mathbb{R}^{3 \times 4}$ )

与 $\text{Tr}_{\text{velo\_to\_cam}}$ 类似。

\* 如果要将激光雷达坐标系中的点x投影到左侧的彩色图像 (P2) y中, 可使用如下公式:

$$\mathbf{y} = \mathbf{P2} * \mathbf{R0\_rect} * \text{Tr}_{\text{velo\_to\_cam}} * \mathbf{x}$$

\* 若想将激光雷达坐标系中的点x投射到其他摄像头, 只需替换P2矩阵即可 (例如右边的彩色相机P3)。

## 2.4 label文件

label文件是KITTI中object的标签和评估数据, 以“000001.txt”文件为例, 包含样式如下:

```
1 | Truck 0.00 0 -1.57 599.41 156.40 629.75 189.25 2.85 2.63 12.34 0.47 1.49 69.44 -1.56
2 | Car 0.00 0 1.85 387.63 181.54 423.81 203.12 1.67 1.87 3.69 -16.53 2.39 58.49 1.57
3 | Cyclist 0.00 3 -1.65 676.60 163.95 688.98 193.93 1.86 0.60 2.02 4.59 1.32 45.84 -1.55
4 | DontCare -1 -1 -10 503.89 169.71 590.61 190.13 -1 -1 -1 -1000 -1000 -1000 -10
5 | DontCare -1 -1 -10 511.35 174.96 527.81 187.45 -1 -1 -1 -1000 -1000 -1000 -10
6 | DontCare -1 -1 -10 532.37 176.35 542.68 185.27 -1 -1 -1 -1000 -1000 -1000 -10
7 | DontCare -1 -1 -10 559.62 175.83 575.40 183.15 -1 -1 -1 -1000 -1000 -1000 -10
```

每一行代表一个object, 每一行都有16列分别表示不同的含义, 具体如下:

- **第1列 (字符串)** : 代表物体类别 (type)  
 总共有9类, 分别是: Car、Van、Truck、Pedestrian、Person\_sitting、Cyclist、Tram、Misc、DontCare。其中DontCare标签表示该区域未标注, 比如由于目标物体距离激光雷达太远。为了防止在评估过程中 (主要是计算precision), 将本来是目标物体但是因为某些原因而没有的区域统计为假阳性(false positives), 评估脚本会自动忽略DontCare区域的预测结果。
- **第2列 (浮点数)** : 代表物体是否被截断 (truncated)  
 数值在0 (非截断) 到1 (截断) 之间浮动, 数字表示指离开图像边界对象的程度。
- **第3列 (整数)** : 代表物体是否被遮挡 (occluded)  
 整数0、1、2、3分别表示被遮挡的程度。
- **第4列 (弧度数)** : 物体的观察角度 (alpha)  
 取值范围为:  $-\pi \sim \pi$  (单位: rad), 它表示在相机坐标系下, 以相机原点为中心, 相机原点到物体中心的连线为半径, 将物体绕相机y轴绕相机z轴, 此时物体方向与相机x轴的夹角, 如图1所示。
- **第5~8列 (浮点数)** : 物体的2D边界框大小 (bbox)  
 四个数分别是xmin、ymin、xmax、ymax (单位: pixel), 表示2维边界框的左上角和右下角的坐标。
- **第9~11列 (浮点数)** : 3D物体的尺寸 (dimensions)  
 分别是高、宽、长 (单位: 米)
- **第12~14列 (浮点数)** : 3D物体的位置 (location)  
 分别是x、y、z (单位: 米), 特别注意的是, 这里的xyz是在相机坐标系下3D物体的中心点位置。
- **第15列 (弧度数)** : 3D物体的空间方向 (rotation\_y)  
 取值范围为:  $-\pi \sim \pi$  (单位: rad), 它表示, 在照相机坐标系下, 物体的全局方向角 (物体前进方向与相机坐标系x轴的夹角), 如图1所示。
- **第16列 (浮点数)** : 检测的置信度 (score)

要特别注意的是，这个数据只在测试集的数据中有（待确认）。

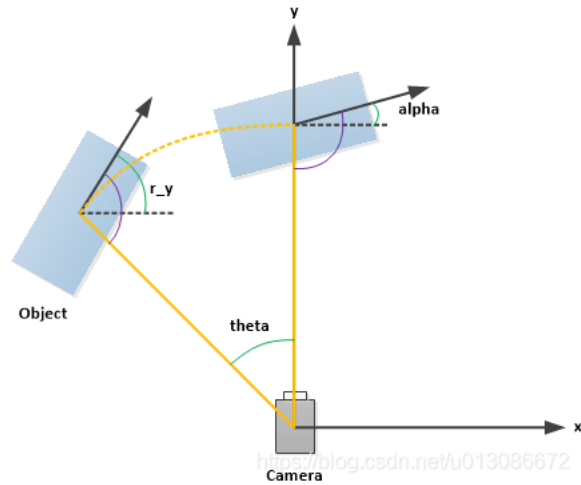
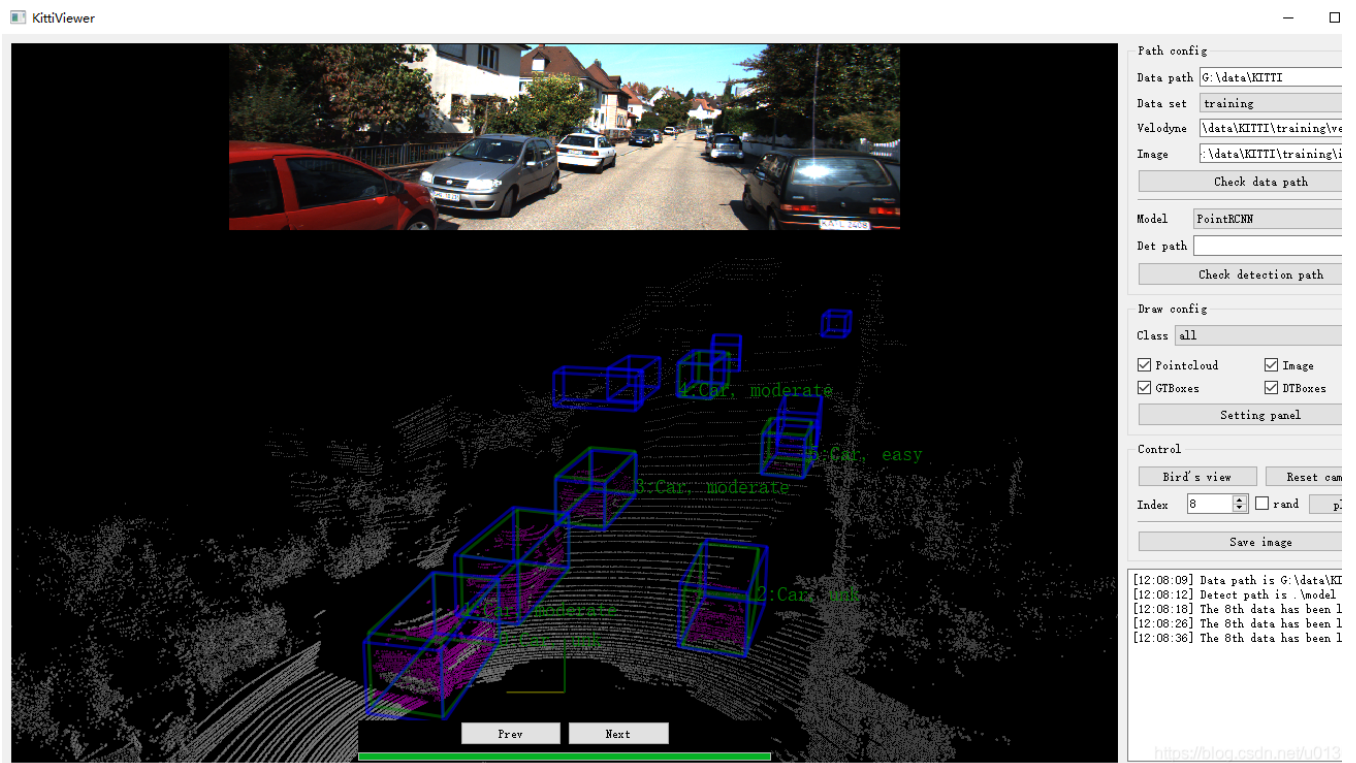


图1 3D物体的观察角和方位角

### 3 KITTI可视化

目前已经完成了pointcloud、gt boxes、label、dt boxes（PointRCNN）等可视化，后续会把体素化加进去，先贴个可视化效果图：



--- end ---