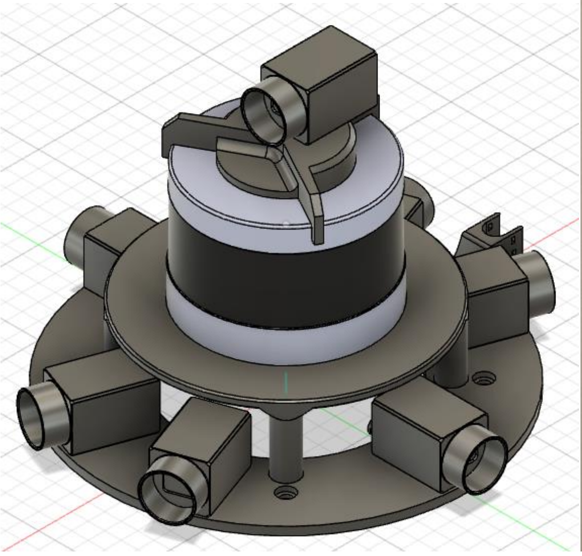
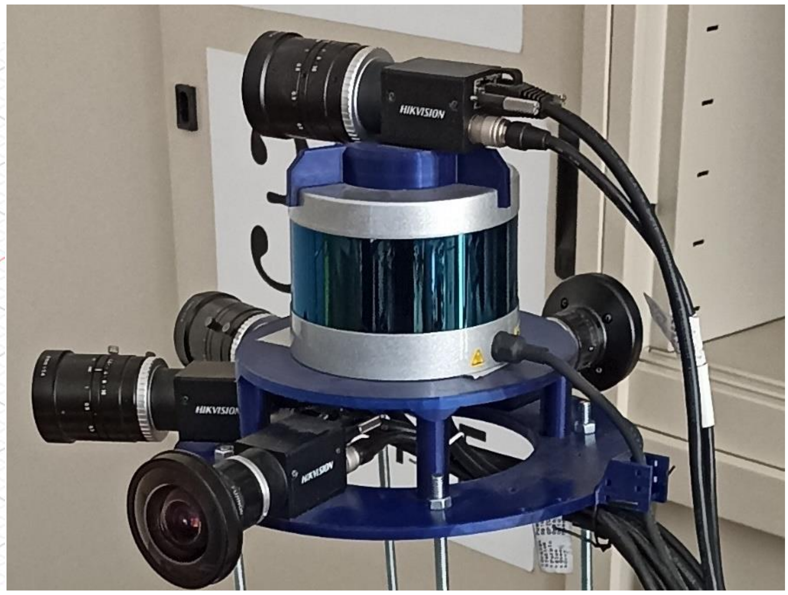
# Introduction to the ELTE 3D sensor pack

# (Version 1.3*)

The currently used version of the sensor pack consists of several cameras, a LiDAR device and some miscellaneous sensors. The cameras and the LiDAR are mounted on a 3D printed fixture, so their positions are constrained to each other.

The fixture allows the sensor pack to be mounted on multiple vehicles, mostly used on the roof rack of the car, but it can be mounted to a go-kart for indoor use, or a shopping cart, for manual movement. The fixture in its current form allows the use of seven cameras as shown below, but only 5 of the slots are used, and only 3 of them are used. **For the Bosch-ELTE Student Competition for Parking Car Detection, only the 3 forward facing cameras are in use**, the rest of them are there only to show the capabilities of the sensor pack.

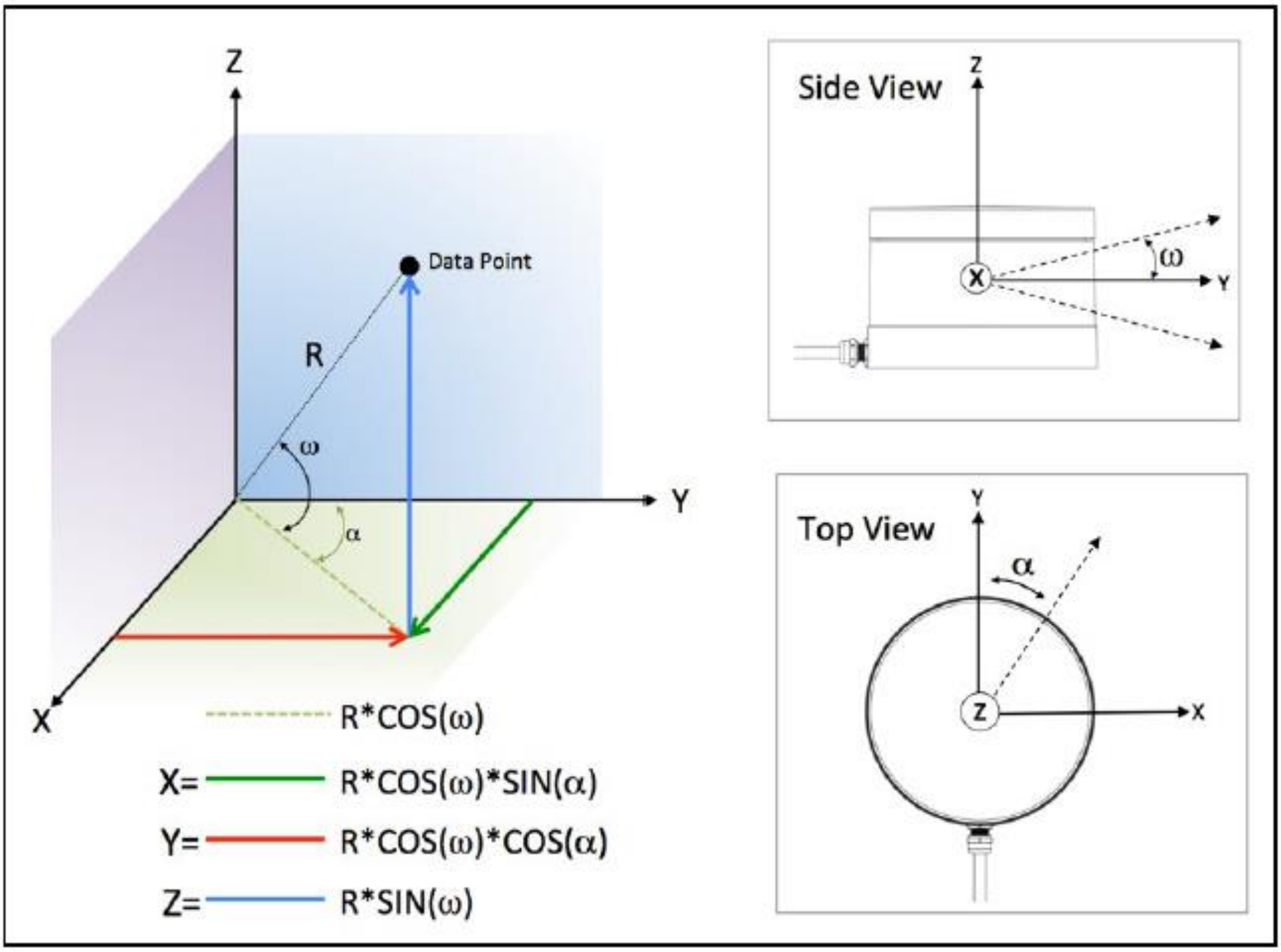





The optics for the cameras are not shown in the Computer Aided Design (CAD). In the current configuration, three cameras are facing forward, two of them are angled ±20° relative to the forward axis, one of them is facing directly forward, on top of the LiDAR. All three forward facing cameras use a "normal" optic. Two cameras are mounted on the plate at 90° angle relative to the forward axis, they use a "fisheye" optic, with 180°+ FOV (only 170° usable). They are currently not in use, but can be turned on anytime. **For the Competition, the available cameras are the ±20° cameras and the fully forward facing one, on the top of the LiDAR.** The cameras currently in use are HikVision/HikRobot MV-CA020-20GC, the exact documentation can be found here. Calibration parameters and information about the optics can be found later in this document.

**: V1.3 is the latest iteration as of 2022.03.01, but there is constant development for the software and mounting mechanism*

The LiDAR is mounted on top of the main 3D printed carrier. The LiDAR used here is a Velodyne VLP-16, (later renamed as PUCK). It uses 16 channels to scan in the vertical direction, with a vertical FOV of 30° (and horizontal FOV of 360°). For reading the exact parameters and documentation of the LiDAR click here.

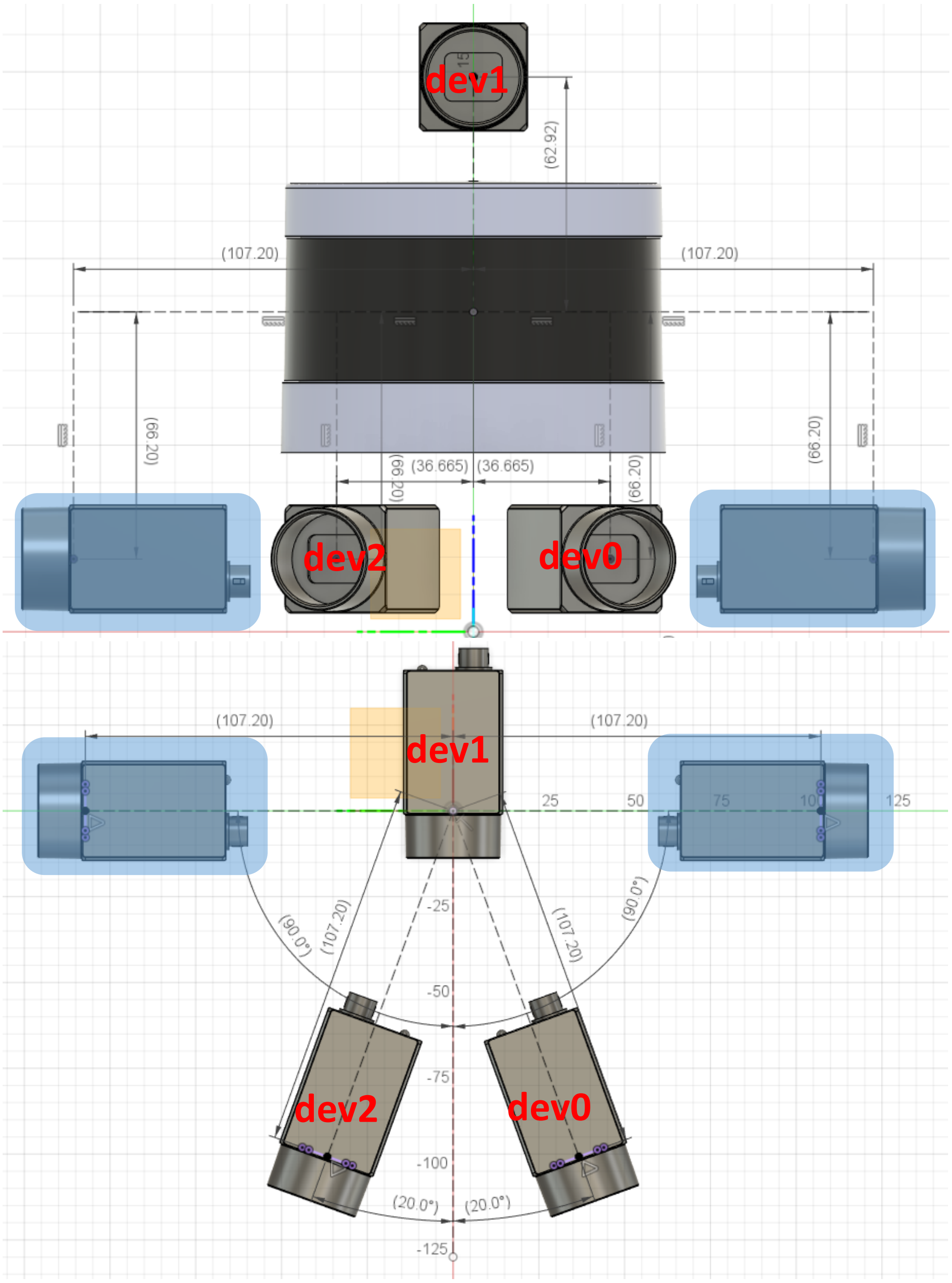



In most of our measurements we use the LiDAR's coordinate system as a reference (world coordinate system). It uses a standard right-handed coordinate system, as shown above. The data coming off the LiDAR uses a polar coordinate system, with a rotation angle for the horizontal and vertical direction, and a distance measurement for the exact position. Our software converts this to XYZIII coordinates, when using the .xyz file format. The I in the XYZIII format stands for intensity, and it is the measured reflectivity of the given point. It is the same number 3 times, to allow some compatibility with XYZRGB point cloud parsers. **In the competition, all the published point clouds are in this .xyz format.**

The sensor pack also contains a GPS module connected to the LiDAR, and an IMU. The LiDAR uses the GPS module only for time synchronization, but the full data coming from the GPS is available to log. Our software logs the latest GPS position for every saved point cloud. There is a simple IMU in the sensor pack, in its current configuration it measures linear acceleration and magnetic flux in 3 directions, and the software logs this data to the point clouds. **In the dataset published for the competition, the IMU data is irrelevant; the GPS data is used only to visualize the point clouds on the map.**

To reference each camera's position and coordinate systems, there is a mounting diagram. Each measurement shown is relative to the optical centre of the LiDAR, which is the 0,0,0 coordinate is the LiDAR's coordinate system, all values are given in mm.

**The cameras marked with blue on the diagram below are unavailable for the competition.**

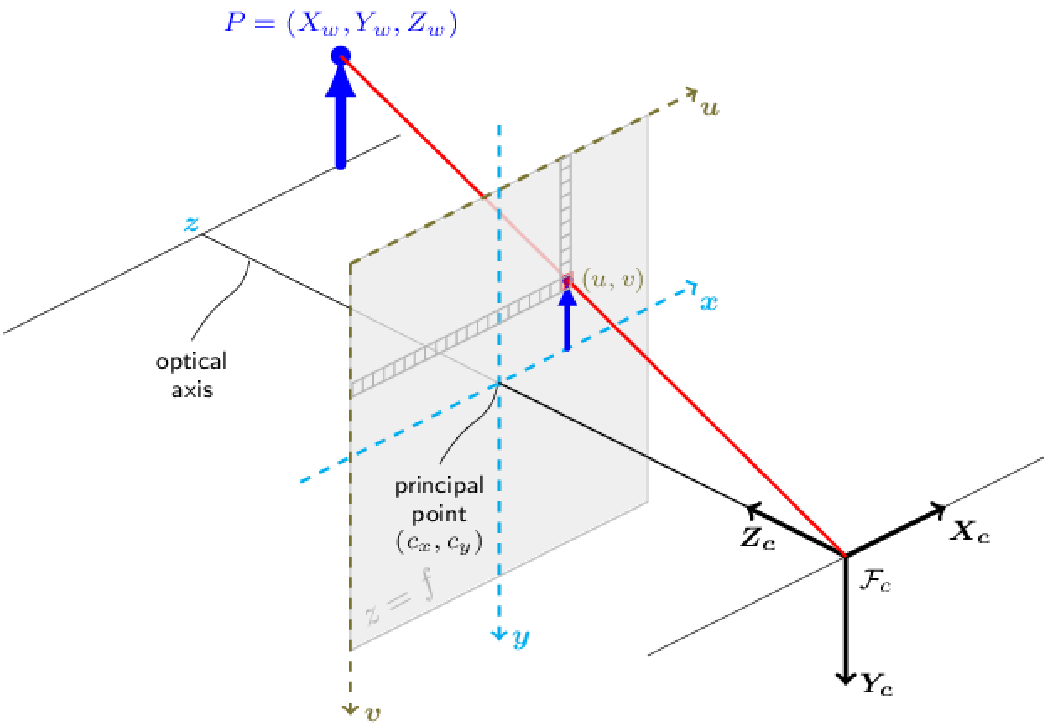



Each of the available camera's coordinates in the LiDAR (world) coordinate system is in the table below.

XYZ is the coordinates relative to the LiDAR system D is the absolute distance, all values are given in mm.

| | X | Y | Z | D |
|---|---|---|---|---|
| dev0 | -36,665 | 100,735 | -66,2 | 125,993 |
| dev1 | 0 | 0 | 62,92 | 62,92 |
| dev2 | 36,665 | 100,735 | -66,2 | 125,993 |

The points measured on the cameras in the above diagram is the optical centre of the cameras, the middle point of the sensor plane, which is NOT the same as the focus point as the focal length is 7mm.




The coordinate system used on the cameras are the OpenCV camera coordinate system. It is originated at the focal point of the camera, and uses a left-handed coordinate system, as shown above. The illustration is taken from the OpenCV documentation, for more information, look at the official OpenCV documentation here. Because of the differences in the systems a coordinate conversion is needed.

In the current version of the software and hardware, the cameras are limited in resolution and framerate, because our Ethernet switch cannot handle all the data coming from the cameras, while still maintaining time sync. The cameras are time synchronized via an external trigger source connected to them. The speed of the trigger source determines the framerate. For every full frame that comes from the camera, the latest full rotation of the LiDAR is logged, and the latest GPS and IMU position (this makes the IMU's update frequency slow). **All of the data published for the competition is recorded at 4FPS on the cameras and 1200RPM (20RPS) on the LiDAR, and the resolution of the cameras are set to ¼, which is 960×600px.**

The camera has a resolution of 1920×1200px, and a pixel size of 4.8×4.8µm, but because of the ¼ resolution scaling, the pixel size to use is the double of the actual, 9.6×9.6µm.

## Camera-Lidar calibration

### Camera intrinsic parameters

As it is well-known in computer vision, the intrinsic parameters of digital cameras are usually represented by an upper triangular matrix

$$K = \begin{bmatrix} kf & 0 & u_0 \\ 0 & kf & v_0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $f$ is the focal length, $k$ is the pixel size and $[u_0, v_0]^T$ is the principal point.

If there is a spatial point $P_C = [X_C, Y_C, Z_C]$ given in the camera coordinate system, the projection is given by

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} kf & 0 & u_0 \\ 0 & kf & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_C \\ Y_c \\ Z_c \end{bmatrix}$$

where $\sim$ denotes equality up to scale operator, and the projected coordinates $[u, v]$ can be obtained by a homogeneous division.

For the applied optics, the focal length is 6mm. The pixel size is 4.8×4.8μm. However, we set only ¼ resolution, therefore the virtual pixel size is 9.6×9.6μm. Therefore, $k = \frac{1}{9.6 \cdot 10^{-6}} pixel/m$. Then

$$\text{kf} = \frac{6 \cdot 10^{-3}}{9.6 \cdot 10^{-6}} = 625$$

Moreover, as we use high-quality Fujinon SV-0614H optics, the principal point is at the midpoint of the image: $[u_0, v_0] = [480,300]$.

Thus, the intrinsic parameter matrix is as follows:

$$K = \begin{bmatrix} 625 & 0 & 960 \\ 0 & 625 & 600 \\ 0 & 0 & 1 \end{bmatrix}$$

### LiDAR-Camera pose

In our system, the world coordinates are fixed to the LiDAR device. The rigid transformation between the LiDAR and the cameras are represented by an orthonormal matrix $R$ and a translation vector $t$. If a 3D location is given in the world as $P_L = [X_L, Y_L, Z_L]$, the corresponding location in the camera system can be given as:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = RP_L + t = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}.$$

**For the competition, the $[\boldsymbol{R}|\boldsymbol{t}]$ matrices are given in the calibration (text) file:**

$$[\boldsymbol{R}|\boldsymbol{t}] = \begin{bmatrix} \boldsymbol{r}_{11} & \boldsymbol{r}_{12} & \boldsymbol{r}_{13} & \boldsymbol{t}_x \\ \boldsymbol{r}_{21} & \boldsymbol{r}_{22} & \boldsymbol{r}_{23} & \boldsymbol{t}_y \\ \boldsymbol{r}_{31} & \boldsymbol{r}_{32} & \boldsymbol{r}_{33} & \boldsymbol{t}_z \end{bmatrix}.$$

**Remark that there are three different $[\boldsymbol{R}|\boldsymbol{t}]$ matrices for the three cameras, they are denoted by $\boldsymbol{Rt1}$ , $\boldsymbol{Rt2}$ and $\boldsymbol{Rt3}$ in the calibration file.**