# Introduction to the ELTE 3D sensor pack

# (Version 1.4\*)

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 4 of the slots are used. **For the Bosch-ELTE Student Competition, the sensor pack is mounted on a car; the camera on the top of the LiDAR, and cameras facing backwards are not available**, they 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, two cameras are facing forward, angled ±20° relative to the forward axis. The forward facing cameras use a “normal” optic (~70°FOV). 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). **For the Competition, 4 cameras are available, ±20° with normal and ±90° fisheye lenses.** The cameras currently in use are HikVision/HikRobot MV-CA020-20GC, the exact documentation can be found [here](https://ikelte-my.sharepoint.com/%3Ab%3A/g/personal/kovbando_inf_elte_hu/EYr4wjqa7j5FmFMiHwxpYL0BEVrSCTh_ilqlXVe_SZhfDQ?e=AqoXbZ). Calibration parameters and information about the optics can be found later in this document.

*\*: V1.4 is the latest iteration as of 2022.10.10, 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](https://ikelte-my.sharepoint.com/%3Ab%3A/g/personal/kovbando_inf_elte_hu/EWusRDlKp85EuWJjUZdqO1gBUpOuwRKhgtrvqkEU-K1omQ?e=9mjomN).



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 LiDAR pointcloud is available, along with the camera pictures, so no GPS or IMU information.**

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.**



**dev3**

**dev1**

**dev2**

**dev0**

**dev3**

**dev2**

**dev1**

**dev0**

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  | 0 | -107,2 | -66,2 | 125,993 |
| dev1  | 100,735 | 36,665 | -66,2 | 125,993 |
| dev2 | 100,735 | -36,665 | -66,2 | 125,993 |
| dev3  | 0 | 107,2 | -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](https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html). 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 framerate, because our PC 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, along with the latest GPS position, and all incoming IMU readings since the last camera frame. **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 full, which is 1920×1200px.** The camera has a resolution of 1920×1200px, and a pixel size of 4.8×4.8µm.

## 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

$$K=\left[\begin{matrix}kf&0&u\_{0}\\0&kf&v\_{0}\\0&0&1\end{matrix}\right],$$

where $f$ is the focal length, $k$ is the pixel size and $\left[u\_{0},v\_{0}\right]^{T}$ is the principal point.

If there is a spatial point $P\_{C}=\left[X\_{C},Y\_{C},Z\_{C}\right]$ given in the camera coordinate system, the projection is given by

$$\left[\begin{matrix}u\\v\\1\end{matrix}\right]\~\left[\begin{matrix}kf&0&u\_{0}\\0&kf&v\_{0}\\0&0&1\end{matrix}\right] \left[\begin{matrix}X\_{C}\\Y\_{c}\\Z\_{c}\end{matrix}\right]$$

where $\~$ 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. Therefore, $k=\frac{1}{9.6∙10^{-6}}pixel/m$. Then

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

Moreover, as we use high-quality Fujinon SV-0614H optics, the principal point is at the midpoint of the image: $\left[u\_{0},v\_{0}\right]=\left[480,300\right]$.

Thus, the intrinsic parameter matrix is as follows:

$$K=\left[\begin{matrix}625&0&960\\0&625&600\\0&0&1\end{matrix}\right]$$

### LiDAR-Camera and Camera-Camera poses

In our system, the world coordinates are fixed to the LiDAR device. The rigid transformation between the LiDAR and the cameras or a camera and the other cameras are represented by an orthonormal matrix $R$ and a translation vector $t$. If a 3D location is given in the world as $P\_{L}=\left[X\_{L},Y\_{L},Z\_{L}\right]$, the corresponding location in the camera system can be given as:

$$\left[\begin{array}{c}X\_{c}\\Y\_{c}\\Z\_{c}\end{array}\right]=RP\_{L}+t=\left[\begin{matrix}r\_{11}&r\_{12}&r\_{13}\\r\_{21}&r\_{22}&r\_{23}\\r\_{31}&r\_{32}&r\_{33}\end{matrix}\right]\left[\begin{array}{c}X\_{L}\\Y\_{L}\\Z\_{L}\end{array}\right]+\left[\begin{array}{c}t\_{x}\\t\_{y}\\t\_{z}\end{array}\right].$$

**For the competition, the** $\left[t\right]$ **matrices are given in the calibration (text) file:**

$$[R|t]=\left[\begin{matrix}r\_{11}&r\_{12}&r\_{13}&t\_{x}\\r\_{21}&r\_{22}&r\_{23}&t\_{y}\\r\_{31}&r\_{32}&r\_{33}&t\_{z}\end{matrix}\right].$$

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

The fisheye distortion parameters are also published. All the sensors are calibrated by MATLAB toolboxes, if you would like to interpret the values, read the related documentation(s).

The parameters, also available in a separate text file, are as follows:

### Setup: #####################################

\* \_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \*

\* /\ \ / \ / /\ \*

\* / \ \\_\_\_/ \\_\_\_/ / \ \*

\* / / CAM2 CAM1 \ \ \*

\* /\_\_\_/ \\_\_\_\ \*

\* CAM0 \_\_\_\_\_\_ CAM3 \*

\* / \ \*

\* / \ \*

\* \ / \*

\* \\_\_\_\_\_\_/ \*

\* LiDAR \*

\* \*

################################################

CAM0 Intrinsics:

 - fx = fy = 120

 - [u0, v0] = [960.5, 600.5]

 | 120.0 0.0 960.5 |

 - K = | 0.0 120.0 600.5 |

 | 0.0 0.0 1.0 |

CAM0 FisheyeParams:

 - MappingCoefficients: [562.1736 -7.1999e-04 4.3601e-07 -4.9236e-10]

 - ImageSize: [1200 1920]

 - DistortionCenter: [953.4174 593.1634]

 - StretchMatrix: |1 0|

 |0 1|

CAM1 Intrinsics:

 - fx = 1280.7

 - fy = 1281.2

 - [u0, v0] = [969.4257, 639.7227]

 - RadialDistortion: [-0.0020 -0.0015]

 | 1280.7 0.0 969.4257 |

 - K = | 0.0 1281.2 639.7227 |

 | 0.0 0.0 1.0 |

CAM2 Intrinsics:

 - fx = 1276.1

 - fy = 1275.4

 - [u0, v0] = [965.9650, 618.2222]

 - RadialDistortion: [0.0157 -0.0104]

 | 1276.1 0.0 965.9650 |

 - K = | 0.0 1275.4 618.2222 |

 | 0.0 0.0 1.0 |

CAM3 Intrinsics:

 - fx = fy = 120

 - [u0, v0] = [960.5, 600.5]

 | 120.0 0.0 960.5 |

 - K = | 0.0 120.0 600.5 |

 | 0.0 0.0 1.0 |

CAM3 FisheyeParams:

 - MappingCoefficients: [576.4125 -6.7844e-04 3.5519e-07 -4.2788e-10]

 - ImageSize: [1200 1920]

 - DistortionCenter: [972.8745 663.9413]

 - StretchMatrix: |1 0|

 |0 1|

### LiDAR to Camera Transform explanation #######################

\* - R: rotation, 3x3 \*

\* - t: translation, 1x3 \*

\* - T: transformation matrix \*

\* | | 0 | \*

\* | R | 0 | \*

\* T = |\_\_\_| 0 | \*

\* | t | 1 | \*

\* - Forward rigid transformation: [x y z 1] = [u v w 1] \* T \*

#################################################################

LiDAR --> CAM0:

 | -0.0153657793937061 -0.00365119568008550 -0.999875273018455 |

 - R = | 0.999773464238133 0.0146735043624255 -0.0154177973462293 |

 | 0.0147279675755847 -0.999885671984468 0.00342489872272477 |

 - t = [ 0.00365920154169145 -0.101481126401536 -0.183123241266722 ]

 | -0.0153657793937061 -0.00365119568008550 -0.999875273018455 0 |

 - T = | 0.999773464238133 0.0146735043624255 -0.0154177973462293 0 |

 | 0.0147279675755847 -0.999885671984468 0.00342489872272477 0 |

 | 0.00365920154169145 -0.101481126401536 -0.183123241266722 1 |

LiDAR --> CAM1:

 | 0.938087304116687 -0.000283237125790693 0.346398801429533 |

 - R = | -0.346205150200322 -0.0342096517324521 0.937534902658625 |

 | 0.0115846376662857 -0.999414638428453 -0.0321896981201434 |

 - t = [ 0.0737416004843908 -0.0317341412646937 -0.178044453316751 ]

 | 0.938087304116687 -0.000283237125790693 0.346398801429533 0 |

 - T = | -0.346205150200322 -0.0342096517324521 0.937534902658625 0 |

 | 0.0115846376662857 -0.999414638428453 -0.0321896981201434 0 |

 | 0.0737416004843908 -0.0317341412646937 -0.178044453316751 1 |

LiDAR --> CAM2:

 | 0.936872686224469 0.00407855516588752 -0.349646872135739 |

 - R = | 0.349521572195037 -0.0401194110186825 0.936068962966844 |

 | -0.0102098176700750 -0.999186568288061 -0.0390123233846818 |

 - t = [ 0.0220629410603821 -0.0314601062659720 -0.196604790031003 ]

 | 0.936872686224469 0.00407855516588752 -0.349646872135739 0 |

 - T = | 0.349521572195037 -0.0401194110186825 0.936068962966844 0 |

 | -0.0102098176700750 -0.999186568288061 -0.0390123233846818 0 |

 | 0.0220629410603821 -0.0314601062659720 -0.196604790031003 1 |

LiDAR --> CAM3:

 | 0.0196090975554365 -0.0482366508949077 0.998643434266457 |

 - R = | -0.999590939857959 -0.0217447407207396 0.0185773842311254 |

 | 0.0208191317528957 -0.998599214781101 -0.0486433139457517 |

 - t = [ 0.000343481617694225 -0.0362336504224814 -0.137639495670372 ]

 | 0.0196090975554365 -0.0482366508949077 0.998643434266457 |

 - T = | -0.999590939857959 -0.0217447407207396 0.0185773842311254 |

 | 0.0208191317528957 -0.998599214781101 -0.0486433139457517 |

 | 0.000343481617694225 -0.0362336504224814 -0.137639495670372 |

### Stereo explanation #####################################################################

\* \*

\* - P1: point in image1 (camera 1) in pixel coordinates \*

\* - P2: point in image2 (camera 2) in pixel coordinates \*

\* - F: fundamental matrix \*

\* [P1 1] \* F \* | P2 | = 0 \*

\* | 1 | \*

\* \*

\* - Q1: point in image1 (camera 1) in normalized image coordinates, \*

\* where the origin is at the camera’s optical center. \*

\* The x and y pixel coordinates are normalized by the focal length fx and fy. \*

\* - Q2: point in image1 (camera 1) in normalized image coordinates, \*

\* where the origin is at the camera’s optical center. \*

\* The x and y pixel coordinates are normalized by the focal length fx and fy. \*

\* - E: essential matrix \*

\* [Q1 1] \* E \* | Q2 | = 0 \*

\* | 1 | \*

\* - R, t: relative pose of the second camera \*

\* orientation1 = orientation2 \* R \*

\* location1 = orientation2 \* t + location2 \*

\* \*

############################################################################################

CAM0 (camera 1) --> CAM2 (camera 2):

 | 5.55040115548871e-07 0.000718421864392740 -0.430286874505905 |

 - F = | 0.000717318615105220 1.26807155980575e-05 -0.573607466776144 |

 | -0.415560569370984 -2.00792963754815 1529.30178062434 |

 | 0.0849935365081733 110.012255418965 2.11648099522374 |

 - E = | 109.785651864411 1.94078419090761 156.865887394320 |

 | 3.41254337981043 -156.734372040631 2.03250758446401 |

 | 0.340941122147217 0.0178192514304566 -0.939915754473482 |

 - R = | -0.0294708880197221 0.999531515450624 0.00825932081403361 |

 | 0.939622593379025 0.0248842098415501 0.341306545662746 |

 - t: [ -156.744727960585 3.53979871362266 -109.975692237599 ]

CAM1 (camera 1) --> CAM2 (camera 2):

 | -7.15574776578017e-07 2.36366718461812e-05 -0.0133930185607378 |

 - F = | 1.75203949229116e-05 -3.59607040668389e-07 -0.0909359968898541 |

 | -0.0111398641028333 0.0483997575140576 22.9336066721754 |

 | -1.16941141740456 38.6451610793471 1.31972927081527 |

 - E = | 28.6172668713250 -0.587636736743662 -94.6121299957232 |

 | -1.28012630574243 90.9799750610368 -1.76478042721460 |

 | 0.767878652237968 0.00135995194270285 0.640593885365693 |

 - R = | 0.00272319999182272 0.999981780904770 -0.00538721081162694 |

 | -0.640589540672501 0.00588118944074345 0.767860958762559 |

 - t = [ 90.9843386924595 1.98976081920328 -38.6341627985188 ]

CAM1 (camera 1) --> CAM3 (camera 2):

 | -5.56609014244981e-05 0.000601523107664508 -0.344551998376106 |

 - F = | 0.000898008361551627 3.36822365710805e-05 0.366887304166810 |

 | -0.478699930927659 -0.740228259370356 189.499493662874 |

 | -8.55388695045895 92.4830356060761 -1.64437542805360 |

 - E = | 138.004628180503 5.17857991555810 151.078438986404 |

 | 9.08213930679450 -182.241636356280 -5.24550618299770 |

 | 0.351911003829117 0.0159082021153068 -0.935898271442704 |

 - R = | -0.0397328487714319 0.999208247514709 0.00204421796575303 |

 | 0.935189791492876 0.0364665216883038 0.352264455606107 |

 - t = [ -182.468928313348 -5.20622814489955 -92.7463207676057 ]