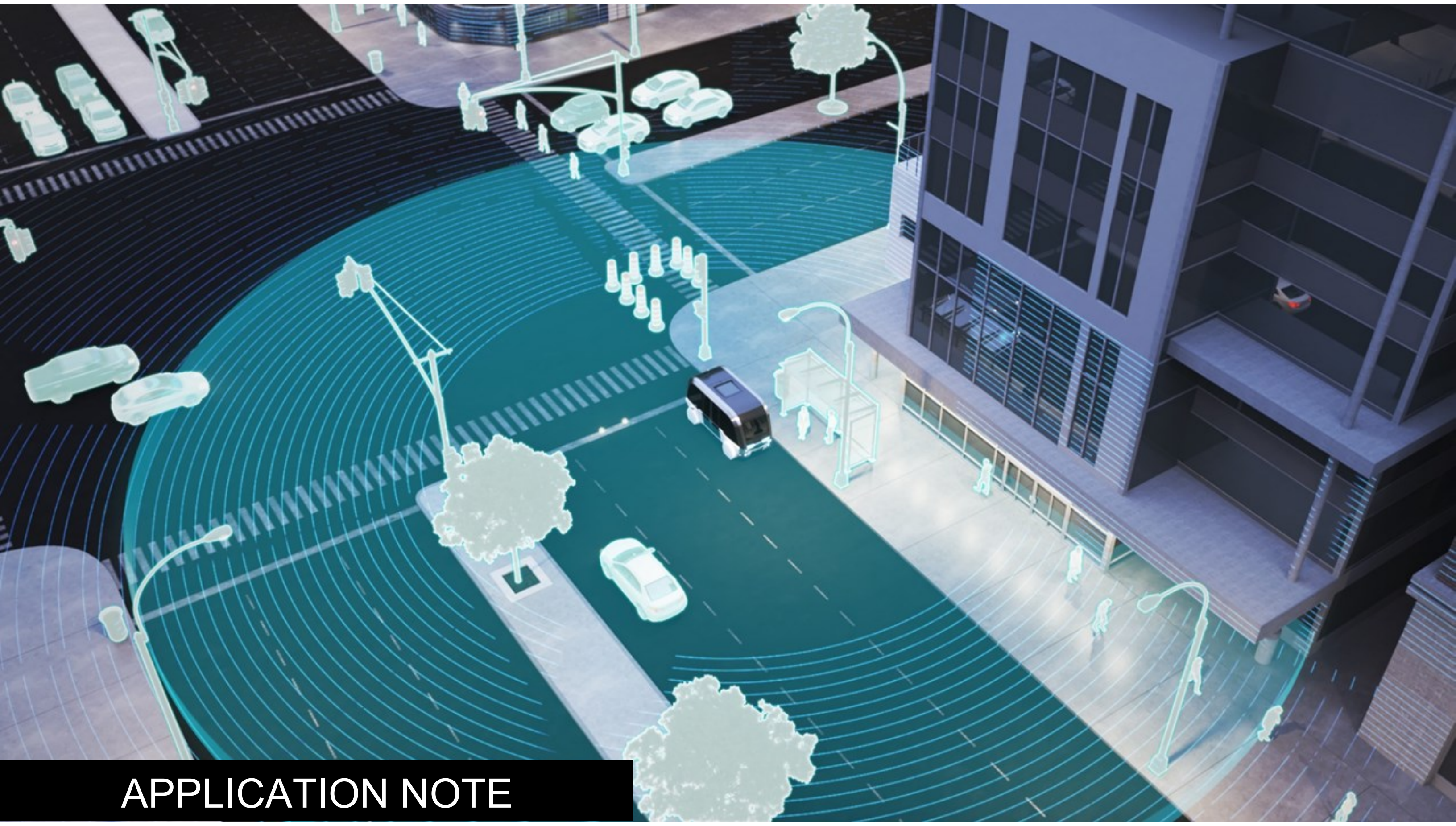
Converting Leddar Pixell Flash LiDAR Sensor's Angular Data and Distance Reporting and Mapping to Real-World Coordinates

By Robert Larose, Principal System Architect and Jean-François Gilbert, Optical Designer



Each Leddar™ Pixell Cocoon LiDAR sensor is thoroughly calibrated on a dedicated production bench both in angular positions and in absolute distance using various traceable targets and carefully selected scenes to reach the required accuracy with precise angular mapping. The sensor comes with accessible internal data providing the angular mapping done in production. This Application Note explains how to transform this basic internal mapping to precise Cartesian coordinates to map the "real world" as seen by the Leddar Pixell.

About the Leddar Pixell 3D Flash LiDAR

LeddarTech® specifically developed the Leddar Pixell cocoon LiDAR to meet the requirements for vulnerable road user detection in a vehicle's surroundings. This road-ready solid-state 3D Flash LiDAR sensor has a wide field of view coverage of 180° x 16° that provides highly reliable detection of pedestrians, cyclists and other obstacles. This field of view is illuminated at once and without any moving parts by the sensor's light sources, and the return signal is captured by 96 horizontal and 8 vertical detection segments, providing 768 independent surfaces with simultaneous acquisitions. This wide field of view is theorical and varies in production due to assembly tolerances. Distance and angle calibrations for each segment are done during assembly of the sensor.

How Would You Like to View the World?

The Leddar Pixell sensor reports echoes using a distance and segment number where the detection has occurred for each echo in the list. The distance is a direct line-of-sight distance between the object detected and the sensor. To position this echo in space around the platform on which the sensor is installed, different ways of mapping the space could be used, as shown in Figure 1.

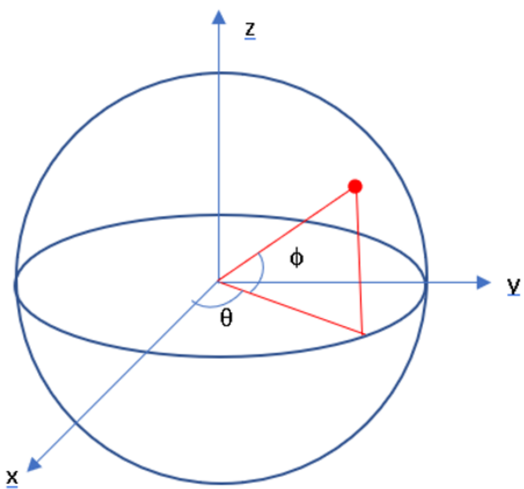


Figure 1 – Coordinates with two well-known reference systems

The red dot could be located in space using either Cartesian coordinates {x, y, z} or spherical coordinates {R, θ, ϕ}. The following equation set could be used to move from one to the other. The reference point of the Leddar Pixell sensor is the origin 0, 0, 0, with the x-axis pointing forward and the y-axis pointing to the left.

$$x = R\cos\theta\,\cos\phi$$

Equation 1: Coordinate transformation for x

$$y = R\sin\theta\,\cos\phi$$

Equation 2: Coordinate transformation for y

$$z = R\sin\phi$$

Equation 3: Coordinate transformation for z

Mechanical Positions of the Reference Point for the Leddar Pixell

The Leddar Pixell sensor is calibrated in production relative to a preferred mechanical reference point rather than the mounting holes in order to reduce variability (Figure 2 and Figure 3).

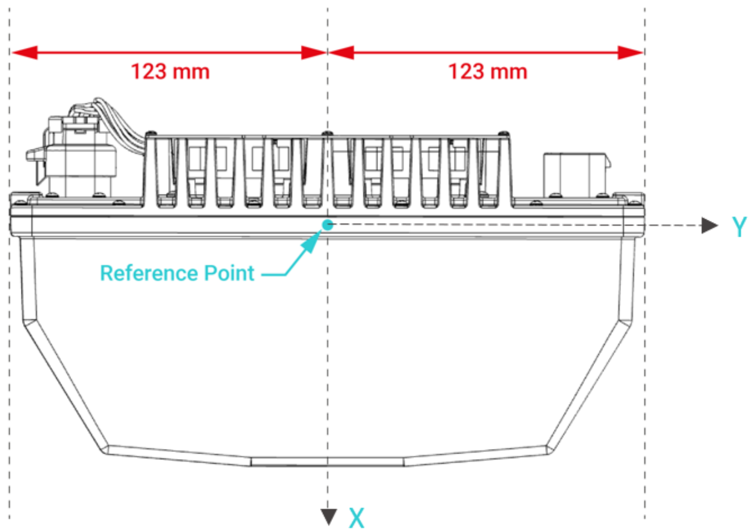


Figure 2 – Top view of the sensor showing the reference point

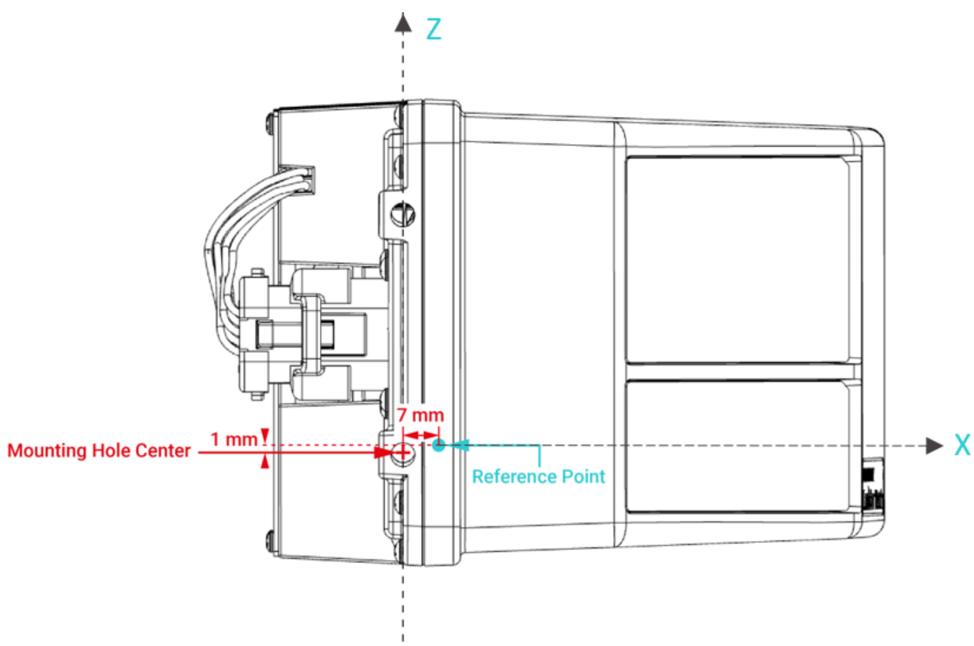


Figure 3 – Side view of the sensor showing the reference point

Information Returned From the Leddar Pixell

The Leddar Pixell sensor returns echoes tagged with a global segment number (0-767) and a distance (see section "Emission Concept" in the Pixell User Guide). For each segment number, there is a general angular mapping stored in the sensor based on production data for a target at maximum range. A transformation should be applied to find the position (x, y, z) in space according to the distance of the echoes and the segment number.

Azimut u and elevation v angles for each segment car be read once at start-up and used in the conversion equations (see section "Constant, Structure and Data Type Definitions" in the Pixell User Guide).

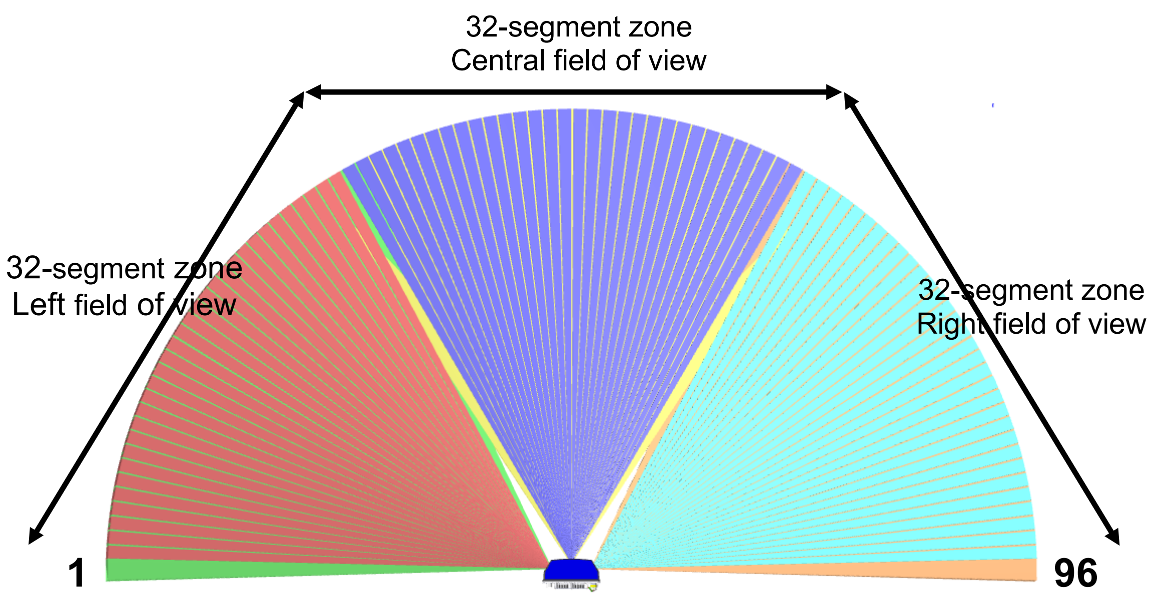


Figure 4 – Diagram showing the three sub-fields of view

From the segment number (0-767) of an echo reported by the sensor, its field of view of origin can be found using the mathematical function modulo with 96 as argument (remainder of the division by 96). A result from this function between 0-31 points to the left field, between 32-63 points to the central field and between 64-95 to the right field of view. This sorting is needed to select the right constants in Table 1 below. For example, a global segment number of 505 represents local segment 25 in the left field of view (remainder of 505 divided by 96).

Mapping Using Opto-Mechanical Parameters of the Leddar Pixell

The following table lists translation values to transform the relative position of the internal sensors toward the external mechanical reference point.

Table 1 – Opto-mechanical constants for each field of view

	Remainder 0-31	***Remainder 32-63***	***Remainder 64-95***
Head	Left field of view	Central field of view	Right field of view
B_x	+0.056 m	+0.0000 m	-0.056 m
B_y	+0.034 m	+0.0396 m	+0.034 m
D	-0.01562 m	-0.01562 m	-0.01562 m

Next, the field of view specific distance R_u can be evaluated using the reported distance R' from the echo, the B_x, B_y and u, v derived from the segment number of the echo of interest.

$$R_u = R' - B_x \sin u \cos v - B_y \cos u \cos v + D \sin v$$

Equation 4: R_u calculation

Transformation to Cartesian Coordinates

From the corrected distance R_u based on actual distance of echo, fixed opto-mechanical parameters B_x, B_y and D and internal angles u, v stored in the sensor specific for each segment, the Cartesian coordinates can be calculated.

$$x = B_y + R_u \cos u \cos v$$

Equation 5: Transformations to Cartesian x

$$y = B_x + R_u \sin u \cos v$$

Equation 6: Transformations to Cartesian y

$$z = -R_u \sin v$$

Equation 7: Transformations to Cartesian z

Those transformations should be applied to all echoes reported by the sensor.

Note: *The way the angle v is reported by the sensor requires a minus sign in the z-axis transformation equation to conform to the universal x,y,z axis definition.*

Sample Python Code

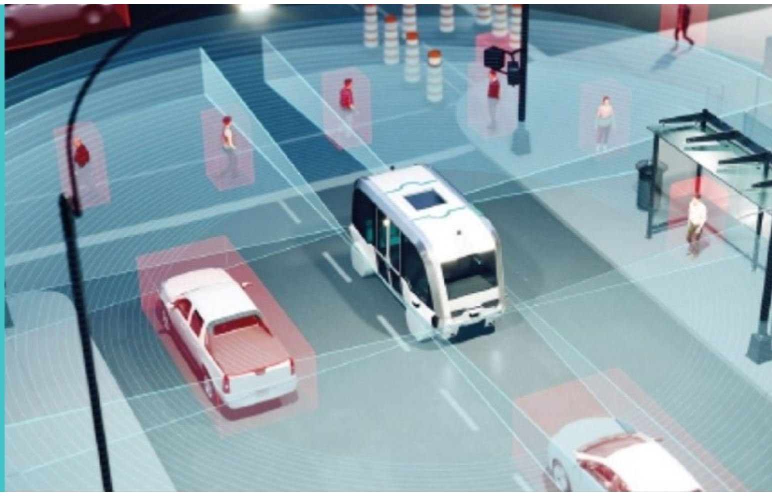
```
#
#  Convert internal Pixell coordinates to real-world X,Y,Z
#  Using echo information segment and distance reported by sensor
#
#  R.Larose
#  2020-06-17
#  LeddarTech
#
import numpy as np

def from_seg_to_bxby(echo_seg):
    '''Convert an echo seg (0-767) from Pixell to seg,line and fov and return coefficient Bx, By and D for this segment
    '''
    seg = int(np.mod(echo_seg,96)) # Segment number in current line 0-95
    line = int(np.floor(echo_seg/96)) # find laser line 0-7 optional
    fov = int(np.floor(seg/32))  # find fov 0,1 or 2
    Bx = [0.056,0.0000,-0.056]   # From sensor's internal design
    By = [0.034,0.0396,+0.034]   # From sensor's internal design
    D = -0.01562                 # From sensor's internal design
    return (
            Bx[fov],
            By[fov],
        D,
            seg,
            line,
            fov
    )
def from_Pixell_to_XYZ(distance,
                u,
                v,
                Bx,
                By,
                D ):
    '''Convert a coordinate (distance, elevation, azimuth) from Pixell points to universal sensor point X,Y,Z.
       Constants: (see: appnote) Note: the way v is stored in sensor requires a minus sign on the Z
    '''
    dx = np.sin(u) * np.cos(v)
    dy = np.cos(u) * np.cos(v)
    Ru = distance - Bx * dx - By * dy + D * np.sin(v)
    return (By + Ru * dy,
            Bx + Ru * dx,
            -Ru * np.sin(v)
    )
```

```
# The following code is for a self-contained example but in operation this is part of the overall system

# Dynamic data received from sensor
# 505 is segment number of an echo, 5.0 is the distance reported for this echo
# Example only, normally all echoes from a frame come in an array from Sensor

echo_segment = 505
echo_distance = 5.0

#Static Data from Pixell
# Should be read once at startup
# Example only, normally should be read in an array from Sensor for all segments
# 42.048 is the azimuth angle u from sensor database for segment 505 (static)
# 3.448 is the elevation angle v from the sensor database for segment 505 (static)
angle_u_seg505 = np.deg2rad(42.048)
angle_v_seg505 = np.deg2rad(3.448)


(Bx,By,D,seg,line,fov) = from_seg_to_bxby(echo_segment)  # Get Bx, By and D from segment number

xyz=from_Pixell_to_XYZ(echo_distance, angle_u_seg505, angle_v_seg505,Bx, By, D)  # Transform to x,y,z

# result (3.6930709093528513, 3.356197069586959, -0.29688932069698526)  (X,Y,Z) in meters
print(xyz)
```


About LeddarTech

LeddarTech is a leader in environmental sensing solutions for autonomous vehicles and advanced driver assistance systems. Founded in 2007, LeddarTech has evolved to become a comprehensive end-to-end environmental sensing company by enabling customers to solve critical sensing and perception challenges across the entire value chain of the automotive and mobility market segments with its LeddarVision™ sensor-fusion and perception platform. LeddarTech delivers a cost-effective, scalable and versatile LiDAR development solution to Tier 1-2 automotive system integrators that enables them to develop automotive-grade solid-state LiDARs based on the foundation of the LeddarEngine™. LeddarTech has 14 generations of solid-state LiDARs based on the LeddarEngine™ platform operating 24/7 in harsh environments. This platform is actively deployed in autonomous shuttles, trucks, buses, delivery vehicles, smart cities/factories and robotaxi applications. The company is responsible for several innovations in cutting-edge automotive and mobility remote-sensing applications, with over 100 patented technologies (granted or pending) enhancing ADAS and autonomous driving capabilities.

For more information: sales@leddartech.com



LeddarTech®

CANADA – USA – AUSTRIA – FRANCE – GERMANY – ITALY – ISRAEL – HONG KONG – CHINA

Head Office
4535, boulevard Wilfrid-Hamel, Suite 240
Québec (Québec) G1P 2J7, Canada
leddartech.com

Phone: + 1-418-653-9000
Toll-free: 1-855-865-9900