CE30-C Solid State Array LiDAR
Operation Manual

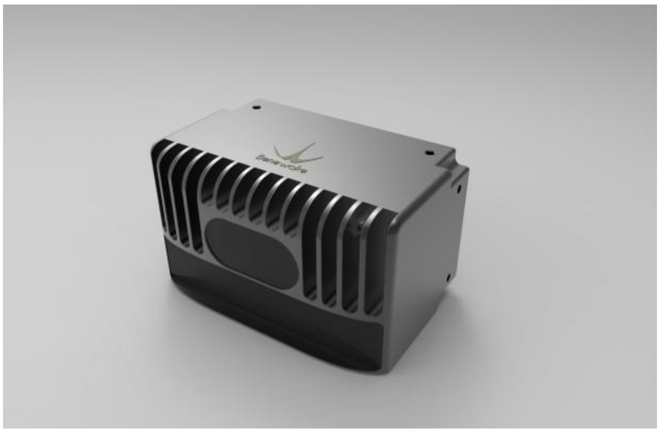
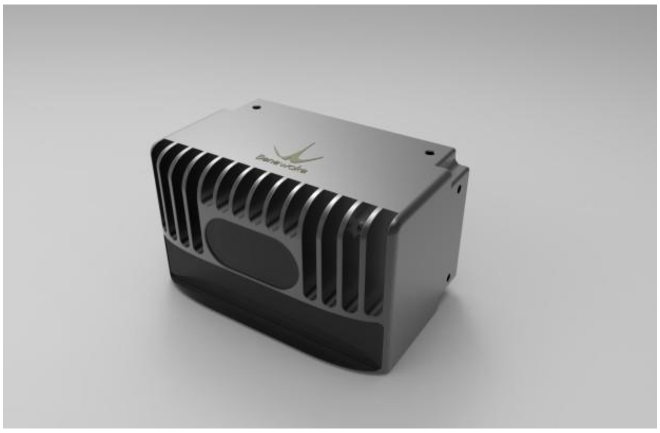
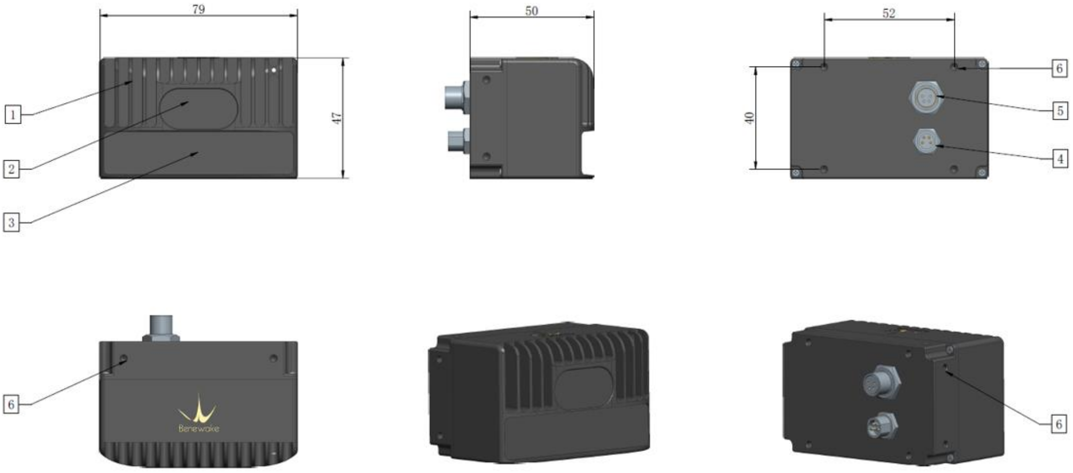


Table of Contents

1. Introduction



CE30-C is an infrared LiDAR developed on the basis of ToF principle. Compared with scanner LiDAR, CE30-C does not have any rotating components in the interior, which could ensure the reliability of long-time work and wider vertical angle.



1. Shell

2. Reception panel (working area, cannot be covered)

3. Emitting panel (working area, cannot be covered)

4. Power/CANBUS port (M8 aerial socket)

5. Ethernet port (M8 aerial socket)

6. Equipment installation hole (M3)

Fig. 1 Outline Dimension Drawing of DE-LIADAR CE30

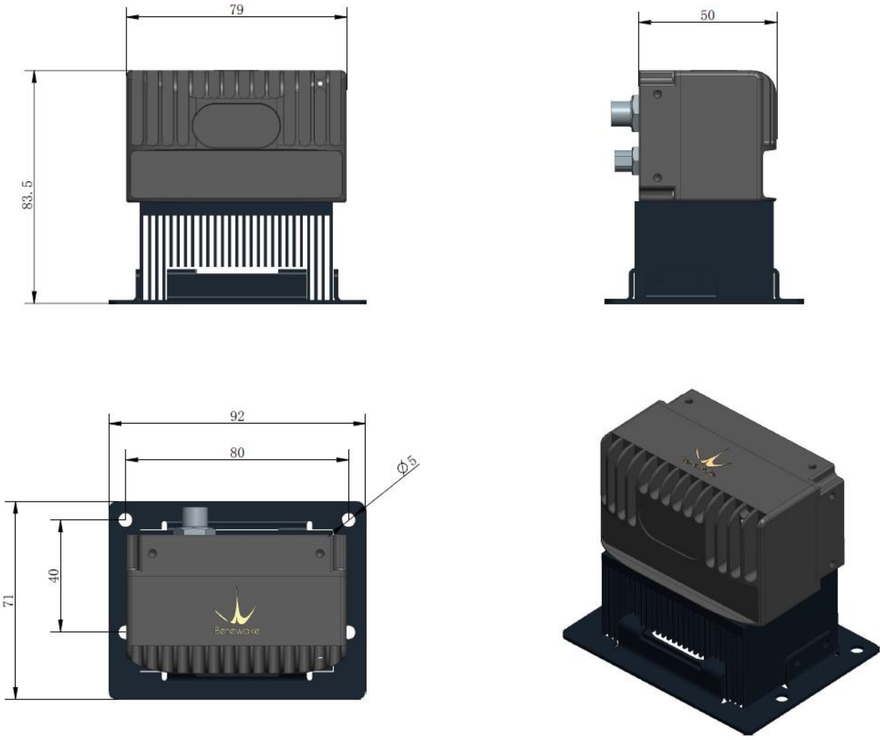
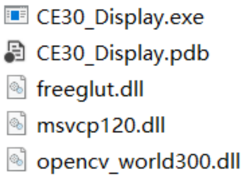


Fig. 2 Dimension and Installation Drawing of CE30-A LiDAR Components - Heat Sink

2. Operation of Windows Display Program

This display program is used for processing and displaying of output data from CE30-C LiDAR in Windows operating system. Please check integrity of files necessary for running before using:



CE30_Display.exe
CE30_Display.pdb
freeglut.dll
msvcp120.dll
opencv_world300.dll

Fig. 3 Windows Display Program and Files Necessary for Running

2.1. Connection

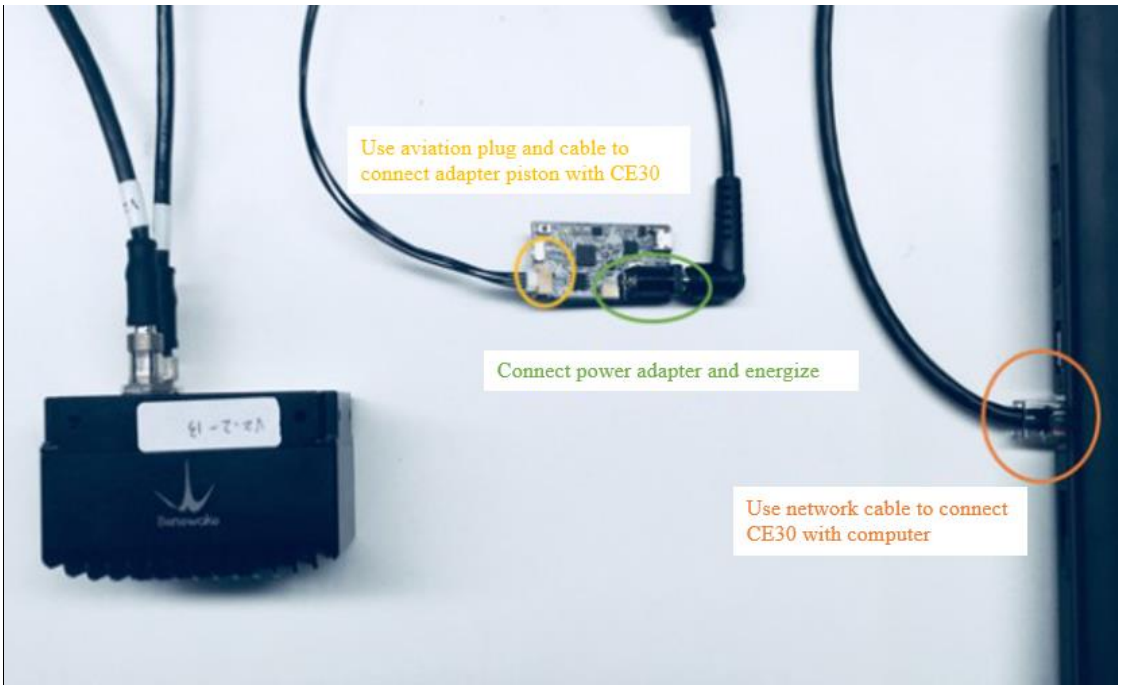
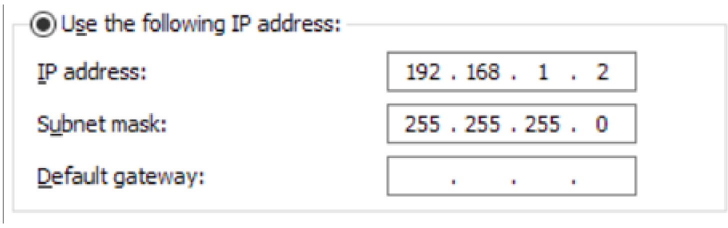

Use aviation plug and cable to connect adapter piston with CE30
Connect power adapter and energize
Use network cable to connect CE30 with computer

Fig. 4 Illustration for Connection of Components

Connect power supply line and Ethernet line and set IP address and subnet mask under the directory *Control Panel\Network and Internet\Network and Internet Connections* as shown in Fig. 5 below.



● Use the following IP address:

IP address: 192 . 168 . 1 . 2

Subnet mask: 255 . 255 . 255 . 0

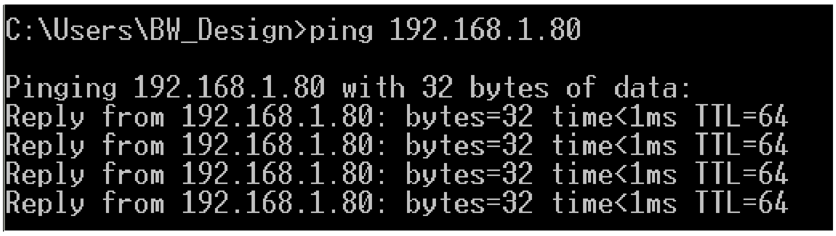
Default gateway: . . .

Fig. 5 Network Setting

To check whether connect is set correctly, start a Command Prompt via Run dialog window and execute command:

ping 192.168.1.80

If the information as shown in Fig. 6 appears, this indicates LiDAR is connected correctly and working.



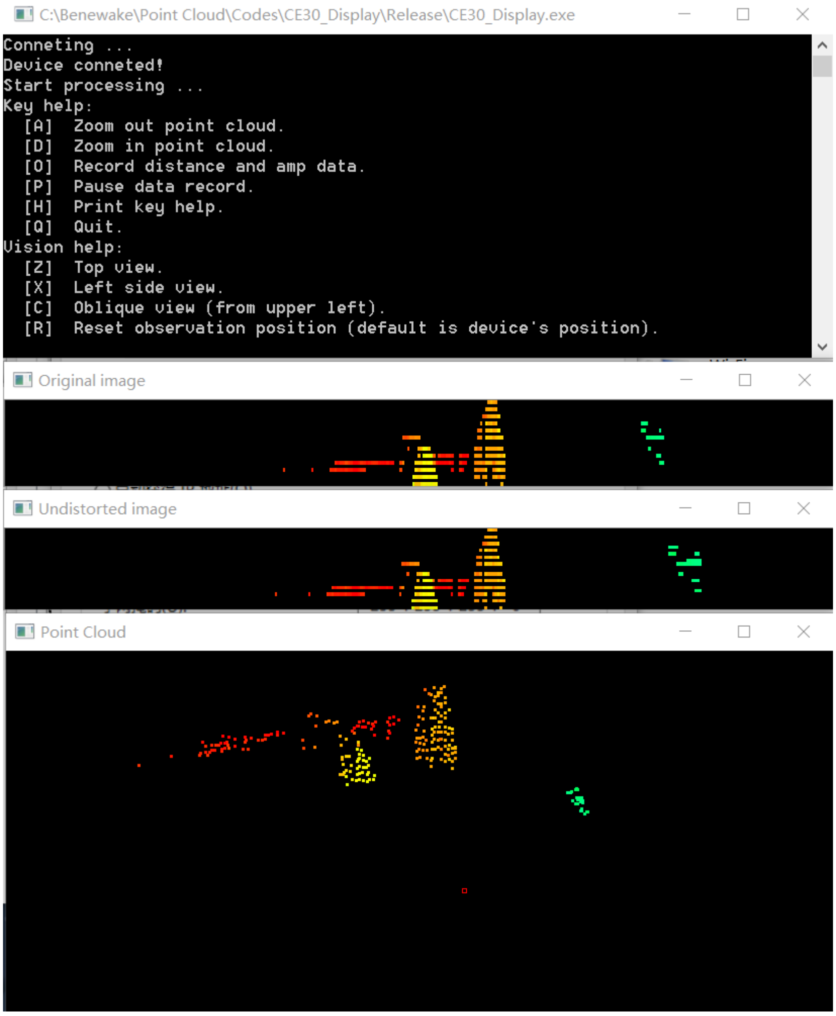
```
C:\Users\BW_Design>ping 192.168.1.80

Pinging 192.168.1.80 with 32 bytes of data:
Reply from 192.168.1.80: bytes=32 time<1ms TTL=64
Reply from 192.168.1.80: bytes=32 time<1ms TTL=64
Reply from 192.168.1.80: bytes=32 time<1ms TTL=64
Reply from 192.168.1.80: bytes=32 time<1ms TTL=64
```

Fig. 6 Check of Data Connection

2.2. Display Program Windows

Double click *CE30_Display.exe* to run display program. The four windows as shown in Fig.7 will pop up: Message

Prompt Window, Original Image Window, Undistorted Image Window and Point Cloud Window.


C:\Benewake\Point Cloud\Codes\CE30_Display\Release\CE30_Display.exe
Original image
Undistorted image
Point Cloud

```
Conneting ...
Device conneted!
Start processing ...
Key help:
  [A]  Zoom out point cloud.
  [D]  Zoom in point cloud.
  [O]  Record distance and amp data.
  [P]  Pause data record.
  [H]  Print key help.
  [Q]  Quit.
Vision help:
  [Z]  Top view.
  [X]  Left side view.
  [C]  Oblique view (from upper left).
  [R]  Reset observation position (default is device's position).
```

Fig. 7 Windows of Upper Computer, From Top to Bottom: Message Prompt Window, Original Image Window, Undistorted Image Window and Point Cloud Window

2.2.1. Message Prompt Window

This window is used to display LiDAR connection information, Key Help for shortcut keys and other running information.

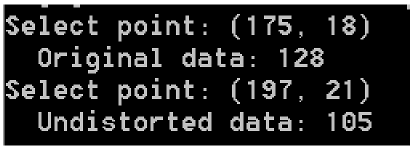
2.2.2. Image Window

Display program outputs two depth images simultaneously:

1. Original depth image from LiDAR (Original image): the depth image constructs with direct output data from LiDAR with field of view 132×9°;

2. Undistorted depth image (Undistorted image): the undistorted depth image corrected by the program with field of view 102×9°.

Click depth image in both windows, the depth value (in centimeter) of the pixel corresponding to current cursor position will be displayed in the message prompt window, as shown in Fig. 8



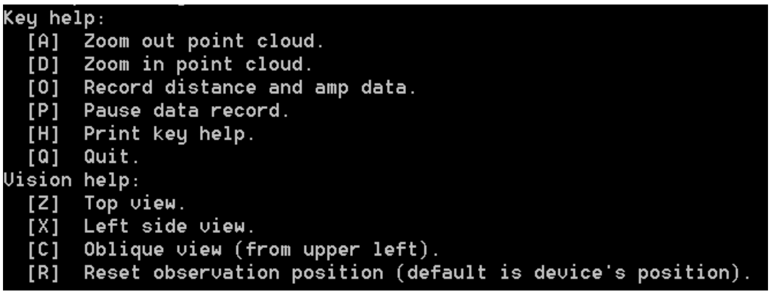
```
Select point: (175, 18)
  Original data: 128
Select point: (197, 21)
  Undistorted data: 105
```

Fig. 8 Display of Depth Value at Pixel Position

2.2.3. Point Cloud Window

Point Cloud Window shows the point cloud image obtained based on the depth image and projection. In this window, hold left mouse button and drag to switch viewing direction of point cloud image; use mouse wheel to zoom point cloud window; use right mouse button to reset observation position of the point cloud window to default angle.

2.2.4. Key Help



```
Key help:
  [A]  Zoom out point cloud.
  [D]  Zoom in point cloud.
  [O]  Record distance and amp data.
  [P]  Pause data record.
  [H]  Print key help.
  [Q]  Quit.
Vision help:
  [Z]  Top view.
  [X]  Left side view.
  [C]  Oblique view (from upper left).
  [R]  Reset observation position (default is device's position).
```

Fig. 9 Key Help

When launching the program, Key Help for shortcut keys will be printed in the Message Prompt Window, as shown in Fig. 9. Controls by shortcut keys consist of two parts:

1. General control: control and enabling/disabling of general functions of the program.

   a) Shortcut A	To zoom out point cloud image

   b) Shortcut D	To zoom in point cloud image

   c) Shortcut O	To record original & undistorted depth image data and amplitude data. Save the data under the directory of the program

   d) Shortcut P	To pause data recording

   e) Shortcut H	To reprint this Key Help messages in message prompt window.

   f) Shortcut Q	To deactivate LiDAR and disconnect the program with LiDAR

2. Vision Help:

   a) Shortcut Z	To adjust point cloud image to top view

   b) Shortcut X	To adjust point cloud image to left side view

   c) Shortcut X	To adjust point cloud image to oblique view from upper left

   d) Shortcut X	To reset observation position to default (default is the position of LiDAR)

Notes: shortcut keys will come into effect only in Original/Undistorted Image Window or Point Cloud Window. When in Message Prompt Window, they will function to input into program.

2.3. Deactivation

Procedures for normal deactivation of LiDAR and display program are:

1. Press "Shortcut Q" in Original/Undistorted Image or Point Cloud Window. The prompt message for

deactivation of LiDAR and disconnection will appear on Message Prompt Window.

2. Press “Enter” key in Message Prompt Window to quit program.

Noted that LiDAR will remain operating and connected with PC terminal if the display program is directly turned off without using “Shortcut Q”. If the display program is rebooted in this case, it will not operate normally as LiDAR will not respond to its request for connection and activation. In that case, the LiDAR needs to be directly power off, followed by reconnection and reboot of the display program.

3. Indicator

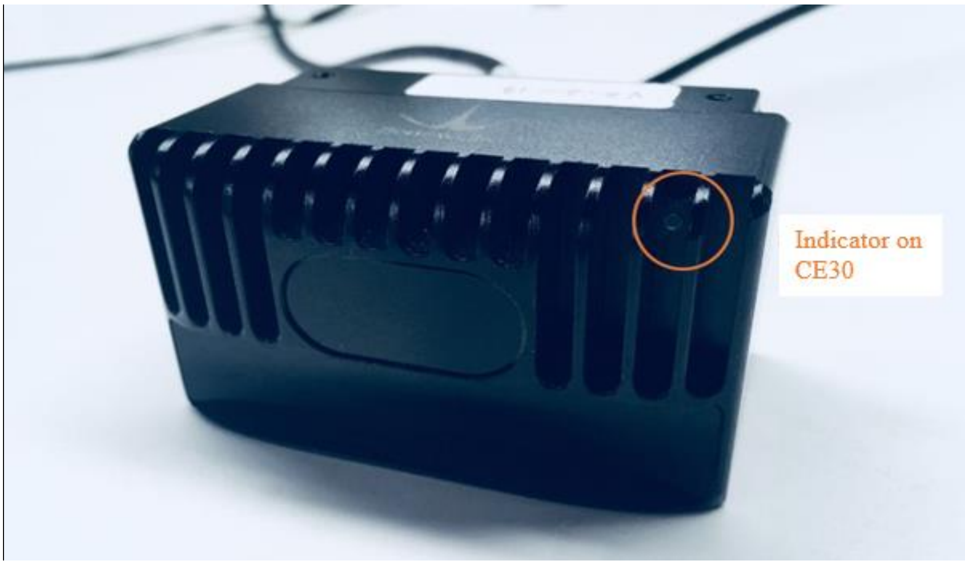


Fig. 10 Position of Indicator

1. Blue light: ready state, ready for connection and running
2. Blue flash: running state
3. Red flash: missing of relevant running files.
4. Red light: fatal error (signal abnormal, interface communication abnormal, I^2C abnormal, etc.)

4. Line Sequence

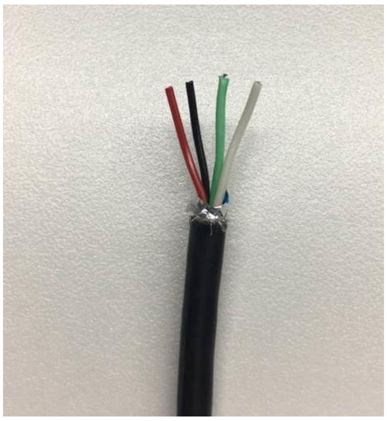


Fig. 11 Line Sequence of Power Supply Port: Red - Positive Pole of Power Supply, Black - Negative Pole of Power Supply

Attention!

1. Current of power adapter shall be above 2A.
2. During energization of LiDAR, there is merely a slim chance of prolonged starting time. If LiDAR is not started after 2 minutes, it is recommended to deenergize and reboot it.
3. Please make sure to disconnect power supply before other connections.

5. Installation Schematics

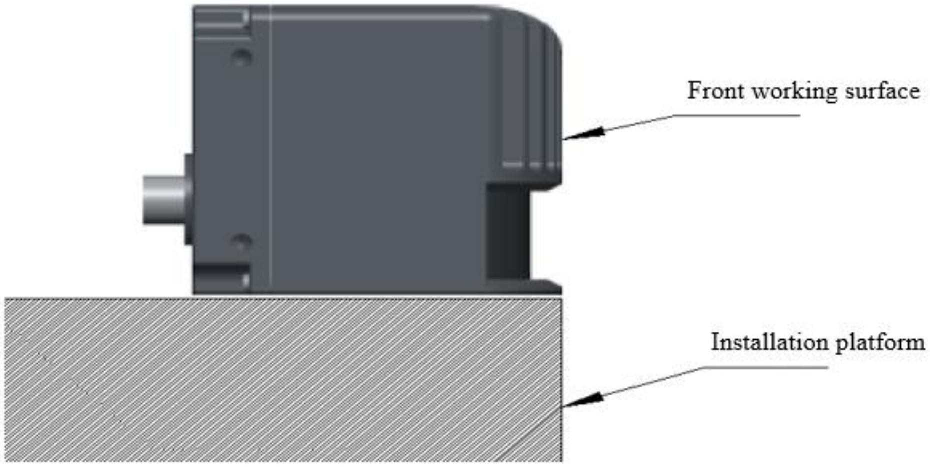


Figure 12 Recommendations on LiDAR Installation Location. The front working surface of LiDAR should exceed or at least be parallel with the installation platform; otherwise, there may be certain interference, further reducing the data accuracy.

6. Introduction of SDK

6.1. Descriptions of SDK

SDK is mainly used to develop applications of this LiDAR on Linux system by using C/C++, of which source codes can be referenced for migration & development of applications on other platforms. SDK consists of following file or folders:

- examples: examples of using SDK, including source codes and makefile;
- include: header files for using SDK;
- src: source codes of SDK;
- makefile: compile command file for compiling SDK into shared object file (.so file) in Linux system;
- README: operation instructions as well as other relevant information and instructions of SDK.

Globel variables in SDK:

- unsigned short gRawDistMatrix[]: original data obtained from LiDAR, arranged in form of one-dimensional array;
- float mapX[]，mapY[]: undistorted projection map, which represents X- and Y-coordinates in original image corresponding to the undistorted image. Arranged in form of one-dimensional array;
- float gCameraMatrix[9]: internal parameters matrix of LiDAR, arranged in form of one-dimensional array and in the sequence {f/dx, skew, u0, 0, f/dy, v0, 0, 0, 1};
- float gDistCoffs[5]: distortion parameters of LiDAR, arranged in the sequence of {k1, k2, p1, p2, k3}.

Methods contained in SDK:

- bool initDevice(int &_sd)

  Parameter:

  _sd socket handle

  In this method, gCameraMatrix [9], gDistCoffs[5], mapX[], mapY[], socket and TCP/IP connection are initialized. LiDAR is literally put into service by invoking such method.

- bool getDistanceData(int _sd, unsignel short *_dst, int _h, int _w)

  Parameter:

  _sd socket handle

  _dst data of undistorted depth image

  _h height of the output depth image. The value should not bigger than 24

  _w width of the output depth image. The value should not bigger than 320

  In this method, original data from LiDAR is accessed and saved in gRawDistMatrix[]. Also the undistorted depth image is output from _dst. A frame of depth image (sized _h * _w, max size 320*24) is output every time this method is called.

- bool closeDevice(int &_sd)

  Parameter:

  _sd socket handle

  In this method, the LiDAR is deactivated and TCP/IP is disconnected.

For details, please refer to sample program and codes in examples

6.2. Output Data

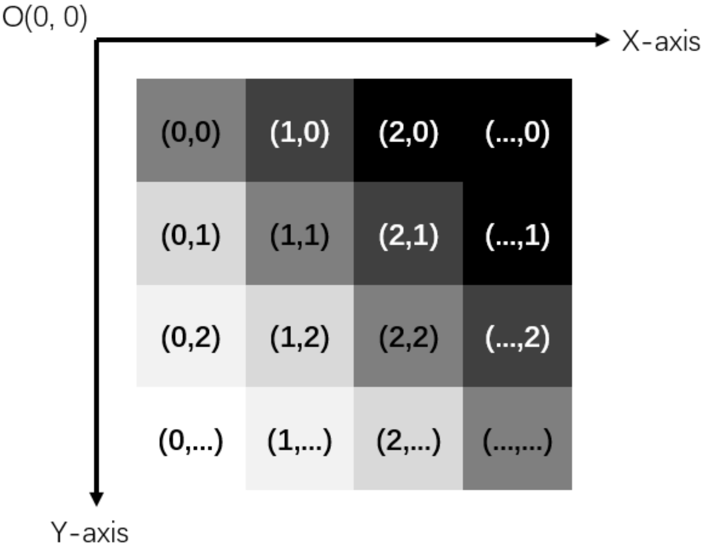


Fig. 13 Illustration of Data Arrangement

The depth image of 320*24 is outputted from the SDK in unit of frame. The data is arranged in form of one-dimensional array. The sequence is from left to right and top to bottom, with the pixel in the upper left corner of the image as the origin. Output along X axis, then Y axis, the following output sequence will be generated as shown in the figure above:

(X,Y) : (0,0) (1,0) (2,0) (…,0) … (0,1) (1,1) (2,1) (…,1) … (0,2) (1,2) (2,2) (…,2) … (0,…) (1,…) (2,…) (…,…) …

For the distance from detected object to the plane of camera, it is expressed in cm by each pixel through the data format of 16-bit unsigned short int.

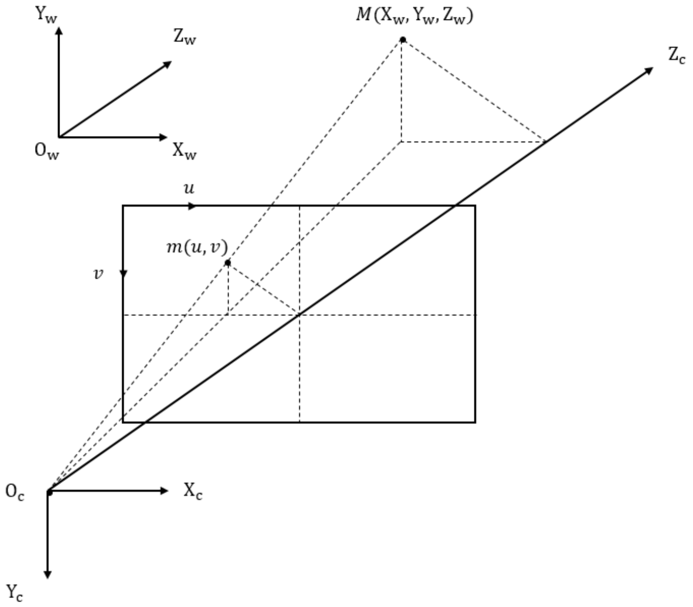


Fig. 14 Projection Relation Diagram

Generally the origin point O_c of camera coordinate system is considered to coincide with origin point O_w of world coordinate system. As such, a point m (u,v) in depth image can be transferred into 3D space point M (X_w,Y_w,Z_w) by following equations:

$$\begin{cases} X_w = Z_c \cdot (u - u_0) \cdot d_x/f \\ Y_w = Z_c \cdot (v - v_0) \cdot d_y/f \\ Z_w = Z_c \end{cases}$$

Where

Z_c output value of each pixel

u_0 x-coordiante of optical center of LiDAR, u_0 = gCameraMatrix[2]

v_0 y-coordiante of optical center of LiDAR, v_0 = gCameraMatrix[5]

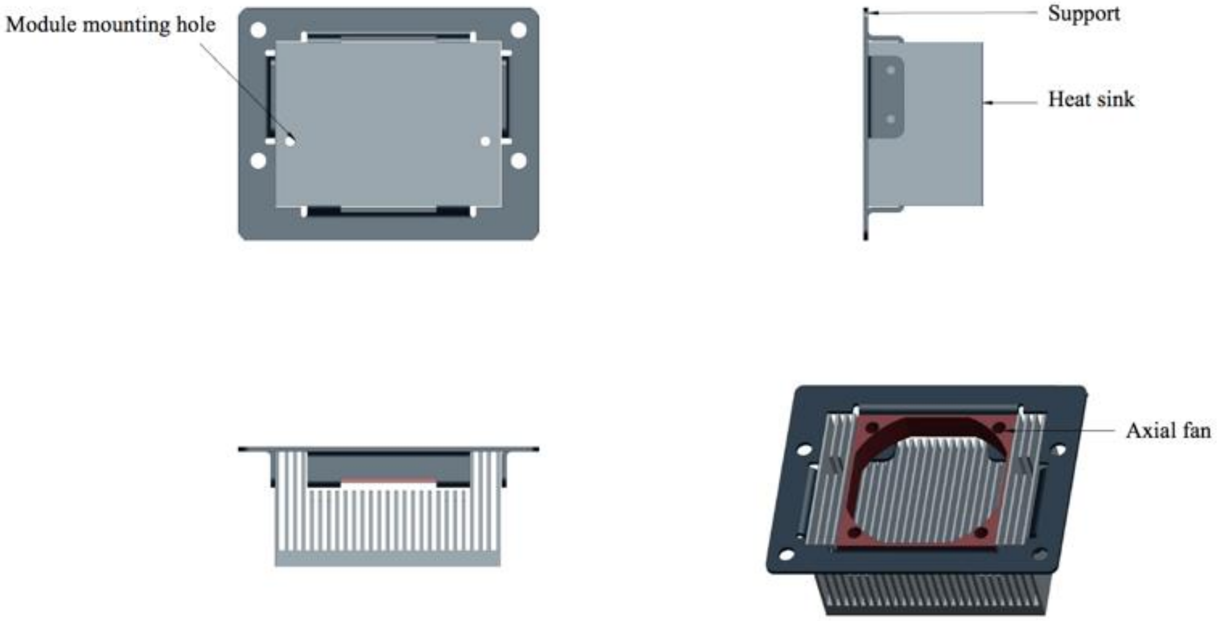
d_x size of pixel along x-axis

d_y size of pixel along y-axis

f focal distance of lens, d_x/f = 1 / gCameraMatrix[0], d_y/f = 1 / gCameraMatrix[4]

7. Cooler and Reference Design

In normal operation, the CE30 LiDAR has an average heat power consumption of around 5W. As shown below is the reference design of heat dissipation component of CE30 LiDAR. Users have the option to design heat dissipation component depending upon actual installation requirements and application scenario.



Fi.g 15 Optional cooler Components