VISIONERF

awareness is power

SDK

3D data format

Update table

Date	Review	Initials	Object

Table 1.

1. Context

This document presents in detail the different 3D data formats provided by the Cirrus3D.

2. Unordered point cloud

2.1. Variant 1: VN_ExecuteSCAN_XYZ

This point cloud format is quite simple. It contains a list of calibrated 3D XYZ points. X, Y and Z coordinates are 32-bit real numbers in IEEE-754 single-precision floating-point binary format.

No smoothing or interpolation is applied to 3D points.

2.2. Variant 2: VN_ExecuteSCAN_XYZI16

This point cloud format is quite simple. It contains a list of 3D XYZ points and, for each 3D point, **16-bit luminance information**. X, Y and Z coordinates are 32-bit real numbers in IEEE-754 single-precision floating-point binary format.

No smoothing or interpolation is applied to 3D points.

3. Ordered point cloud

The Cirrus3D offers two types of matrices:

- A matrix (type 1) with a fixed pitch in mm between two pixels, corresponding to a top view without perspective. The pitch itself depends on the sensor and the density chosen for the scan, and is determined automatically to ensure that few 3D points are lost. In this way, the point density remains the same whatever the height.
- A matrix (type 2) with an angular pitch that preserves the perspective effect and minimizes the number of lost 3D points. The angular pitch is also automatically adjusted according to the density chosen for the scan. As a result, point density varies with height.

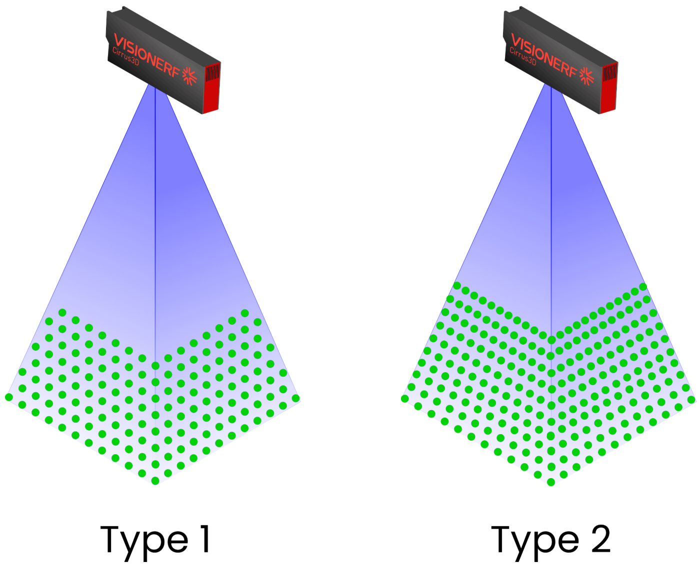

Figure 1 : On the left, the density of the point cloud is constant; on the right, it varies with height.

3.1. Variant 1 : VN_ExecuteSCAN_Matrix_XYZI8

The point cloud is in the form of a **type 1 matrix**. The size of the matrix depends on the type of Cirrus3D, density and ROI used.
Points are ordered according to their x and y values. So, whatever its position in z, an object will always cover the same size in the matrix. The perspective effect is eliminated, so that two identical parts will appear with the same "dimensions" (in number of pixels) even though they are at different distances. 8-bit luminance information is available for each 3D point. X, Y and Z coordinates are 32-bit real numbers in IEEE-754 single-precision floating-point binary format.

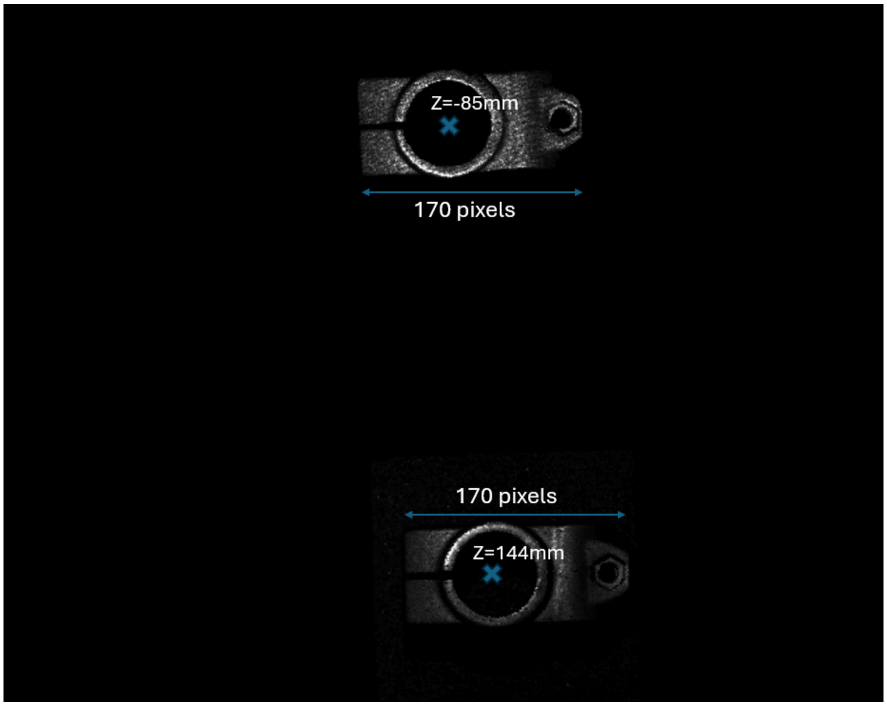

Figure 2 : 2.5D representation of the matrix (x, y and z coordinates ignored). The intensity contained in the matrix is used for the image's gray level. The two parts appear with the same "dimensions" (number of pixels) despite being at different distances. There is no perspective effect.

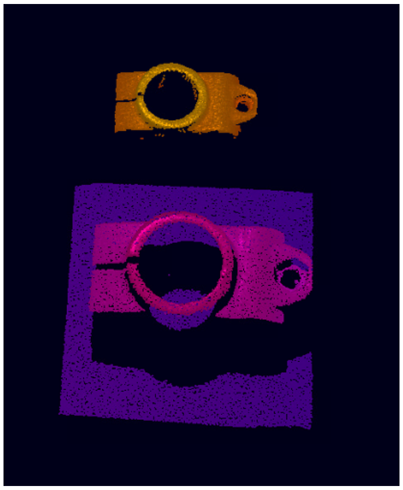

Figure 3 : 3D image obtained using x, y and z coordinates. The image is colored according to the z value of the 3D point.

Advantage:

- In the matrix, each point is surrounded by its x, y neighbors.

Limitation:

- When constructing this matrix, 2 points may end up in the same square. In this case, one of the two points will be lost. This can create a variation in point density when displaying the raw 3D points. If this bothers you, we advise you to use variant 3: VN_ExecuteSCAN_CameraCOP_MiddleCam2_XYZI.

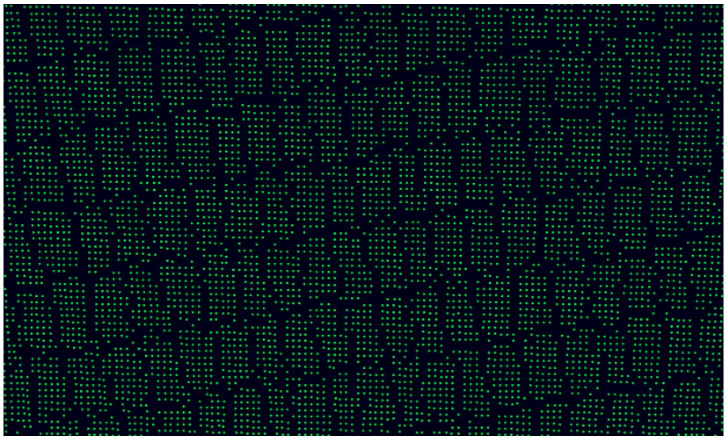

Figure 4 : The image showing the point cloud after arrangement in the matrix. A pattern effect appears in the point cloud.

- Luminance is only available when a 3D point has been found. The resulting luminance image of the matrix will contain boxes with no data.

3.2. Variant 2 : VN_ExecuteSCAN_Matrix_XYZRGB

Matrix construction identical to variant 1. Only the data format changes. X, Y, Z coordinates and intensity are defined in a pcl::PointXYZRGB structure. As Cirrus3D is grayscale, the R, G and B components are identical.

3.3. Variant 3 : VN_ExecuteSCAN_CameraCOP_MiddleCam2_XYZI

The point cloud is in the form of a type 2 matrix. The size of the matrix depends on the Cirrus3D, density and ROI used. The resulting point cloud is calibrated and structured in a grid with a perspective viewpoint.
For each 3D point, 8-bit luminance information is available. X, Y and Z coordinates are 32-bit real numbers in IEEE-754 single-precision floating-point binary format.

Advantage :

- The scan result is transferred during the scan, minimizing the acquisition cycle time (image capture+transfer).

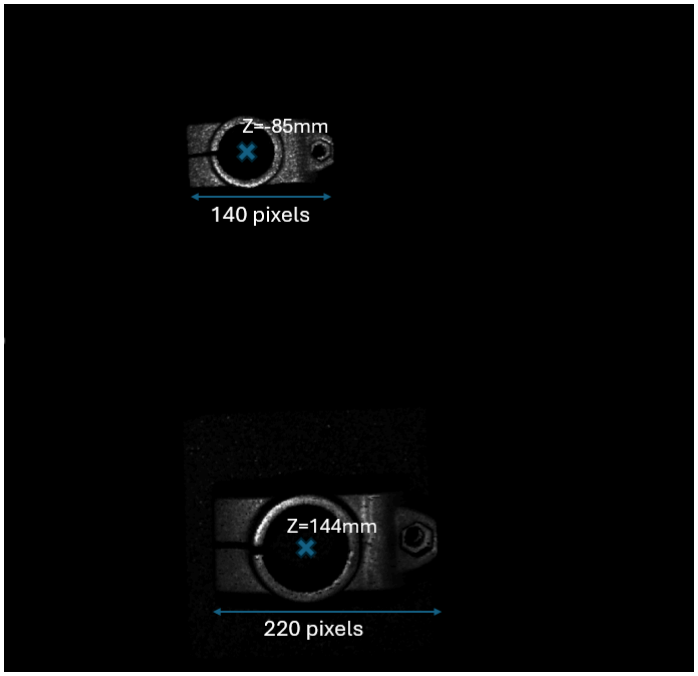

Figure 5 : 2.5D matrix representation (x and y coordinates ignored)

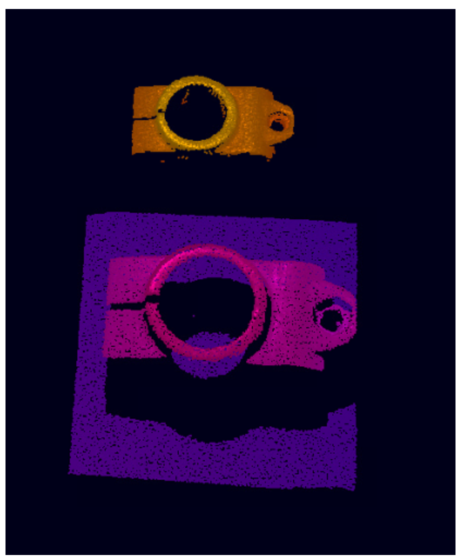

Figure 6 : 3D image obtained using x, y and z coordinates. The image is colored according to the z value of the 3D point.

Limitation :

- Luminance is only available when a 3D point has been found. The resulting luminance image of the matrix will contain boxes with no data. An "empty" box contains the NaN value.

3.4. Variant 4: Depth map

The data is in the form of a rectified image. The x and y coordinates are implicit and can be obtained using CoordinateScale(A) and CoordinateOffset(B). This format is a depth map: an image built of 16-bits pixel, where the darkest pixels correspond to points closer to the Cirrus. The 3d coordinates x,y,z of the 3d points can then be extracted from the u,v pixel coordinates and their corresponding pixVal :

x = x_scale * u + x_offset
y = y_scale * v + y_offset
z = z_scale * pixVal + z_offset

With, in this case, x_scale = y_scale.

Invalid pixels (the elements of the rectified image that do not contain valid data) are set to a user-selected value (generally 0 or 65535).

This format was designed to be compatible with a wide variety of applications, including those who have yet to expand in the "3D processing" territory and are designed for use with 2d images.

However, this format has some drawbacks :

- since the x & y coordinates are implicit, there is a loss of accuracy (the raw 3d points are rarely generated at the exact center of the pixel)
- the Cirrus natively generates a denser coverage of 3d points on closer objects than on farther objects. In order to minimize the number of "holes" in the rectified image, the image is under-sampled, losing about 20% resolution compared to the Matrix_XYZ format.
- the Cirrus natively generates 3d points from a perspective view, meaning that if the Cirrus is suspended above a bin it is able to (at least partially) scan the sides of the bin, and generates 3d points sharing (roughly) the same x,y, coordinates but with a different z. With this rectified format, those points are lost : in case of multiple 3d points projecting to the same pixel, we only keep the one closest to the Cirrus.

The resolution of the image (number of lines & columns) vary only depending on the selected scan density ; this is an arbitrary choice for easier integration with applications where the image resolution is hard-coded.

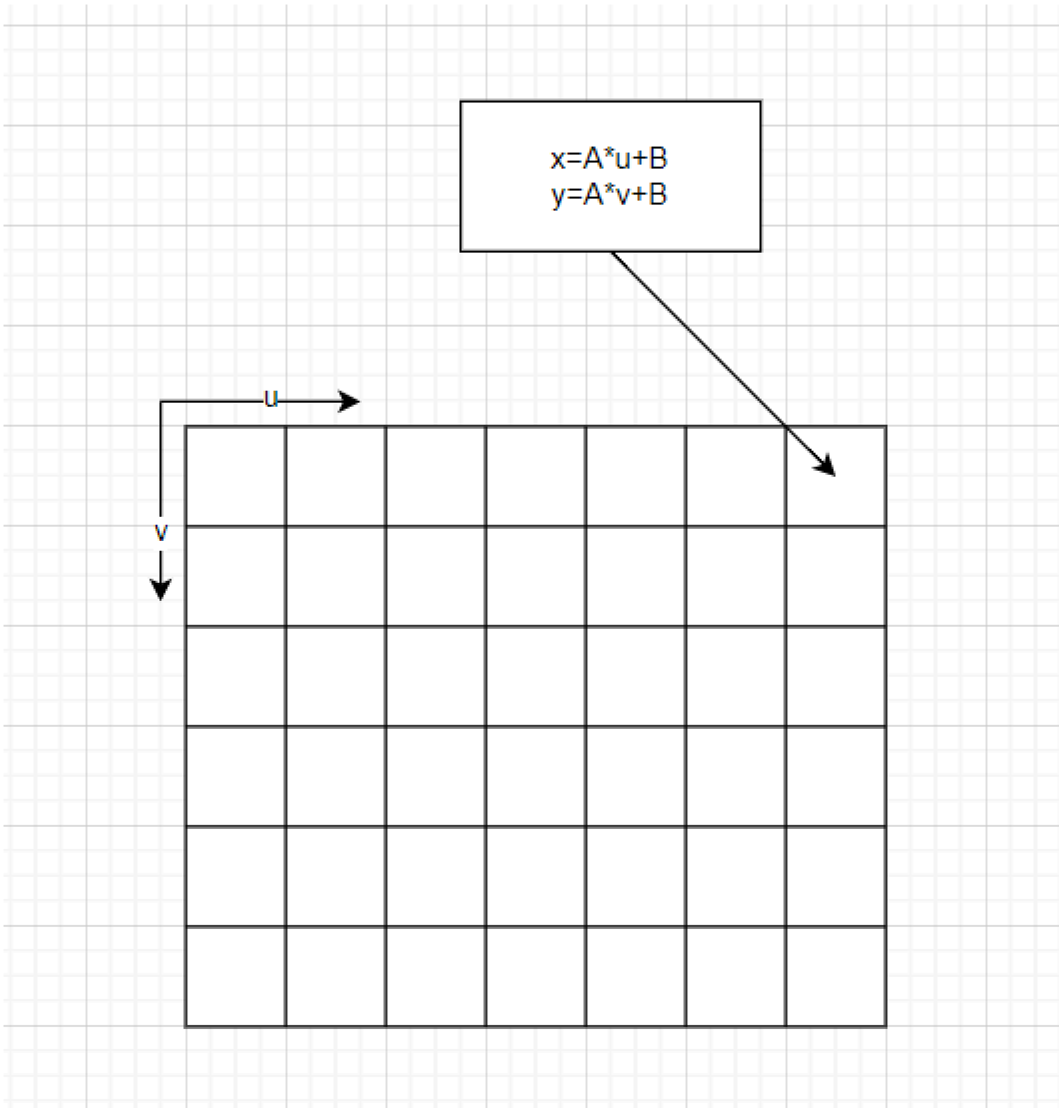

Figure 7 : Depth map