



SDK

Format des données 3D

Tableau des mises à jour

Date	Révision	Initiales	Objet

Table 1.

Tableau des mises à jour	2
1. Contexte	4
2. Nuage de points non ordonnés	4
2.1. Variante 1: VN_ExecuteSCAN_XYZ	4
2.2. Variante 2: VN_ExecuteSCAN_XYZI16	4
3. Nuage de points ordonnés	4
3.1. Variante 1 : VN_ExecuteSCAN_Matrix_XYZI8	5
3.2. Variante 2 : VN_ExecuteSCAN_Matrix_XYZRGB	7
3.3. Variante 3 : VN_ExecuteSCAN_CameraCOP_MiddleCam2_XYZI	7
3.4. Variante 4 : Carte de profondeur (à venir)	9

1. Contexte

Ce document a pour vocation de présenter en détail les différents formats de données 3D fourni par le Cirrus3D.

2. Nuage de points non ordonnés

2.1. Variante 1: VN_ExecuteSCAN_XYZ

Ce format de nuage de points est assez simple. Il contient une liste de point 3D XYZ calibré. Les coordonnées X, Y et Z sont des réels 32 bits au format binaire à virgule flottante IEEE-754 simple précision.

Aucun lissage ou interpolation n'est appliqué sur les points 3D.

2.2. Variante 2: VN_ExecuteSCAN_XYZI16

Ce format de nuage de points est assez simple. Il contient une liste de point 3D XYZ et pour chaque point 3D une information de **luminance sur 16 bits**. Les coordonnées X, Y et Z sont des réels 32 bits au format binaire à virgule flottante IEEE-754 simple précision.

Aucun lissage ou interpolation n'est appliqué sur les points 3D.

3. Nuage de points ordonnés

Le Cirrus3D propose deux types de matrices :

- Une matrice (type 1) avec un **pas fixe en mm** entre deux pixels, correspondant à une vue de dessus sans perspective. Le pas lui-même dépend du capteur et de la densité choisie pour le scan, et est déterminé automatiquement pour ne perdre que peu de points 3D. Ainsi, la densité de points restera identique quelle que soit la hauteur.
- Une matrice (type 2) avec un **pas angulaire** qui préserve l'effet de perspective et minimise le nombre de points 3D perdus. Le pas angulaire est également ajusté automatiquement selon la densité choisie pour le scan. Ainsi, la densité de points varie selon la hauteur.

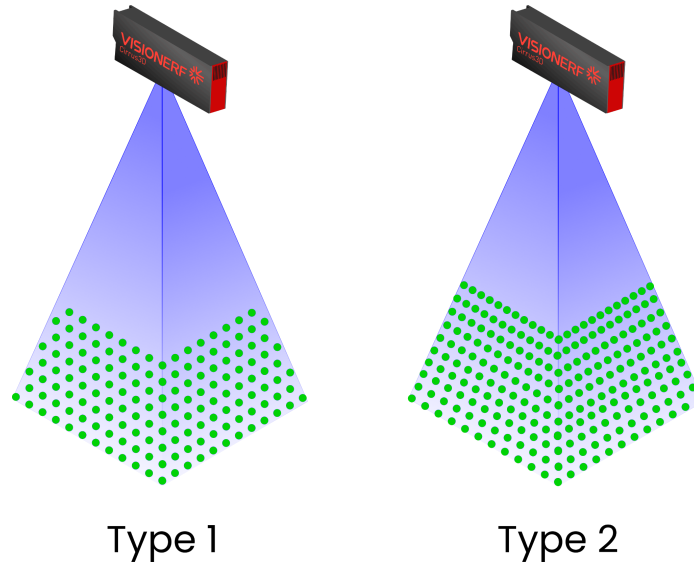


Figure 1 : À gauche, la densité du nuage de points est constante, à droite, elle varie selon la hauteur.

3.1. Variante 1 : VN_ExecuteSCAN_Matrix_XYZI8

Le nuage de points est sous forme de **matrice de type 1**. La taille de la matrice dépend du type de Cirrus3D, de la densité et de la ROI utilisés.

Les points sont **ordonnés suivant leur valeur en x et y**. Ainsi, quelle que soit sa position en z, un objet couvrira toujours la même taille dans la matrice. L'effet de perspective est supprimé si bien que deux pièces identiques apparaîtront avec les mêmes « dimensions » (en nombre de pixels) bien qu'elles soient à des distances différentes.

Pour chaque point 3D, une information de luminance sur 8 bits est disponible. Les coordonnées X, Y et Z sont des réels 32 bits au format binaire à virgule flottante IEEE-754 simple précision.

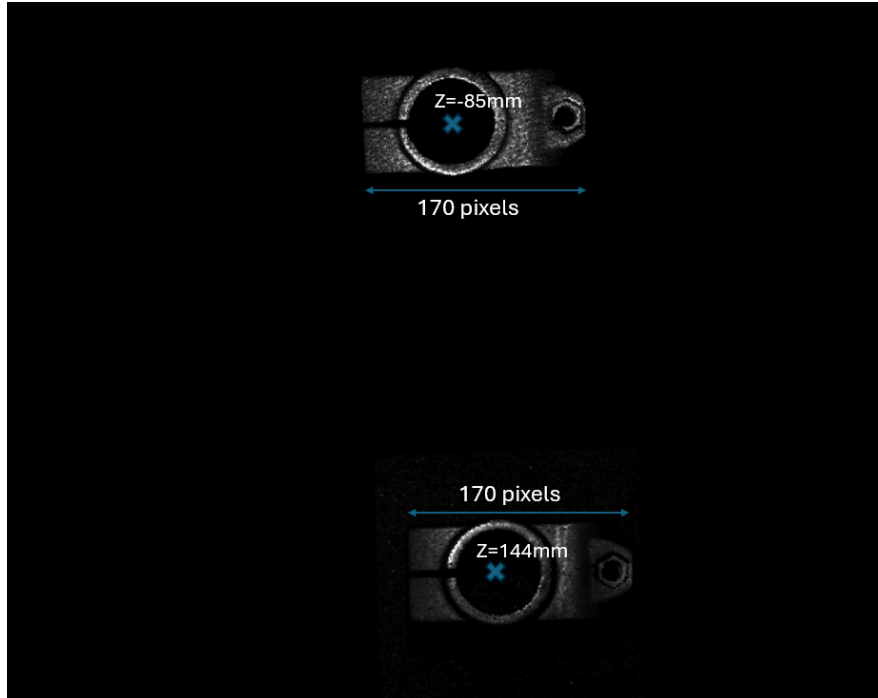


Figure 2 : Représentation 2.5D de la matrice (les coordonnées x, y et z étant ignorés). L'intensité contenue dans la matrice est utilisée pour le niveau de gris de l'image. Les deux pièces apparaissent avec les mêmes « dimensions » (nombre de pixels) bien qu'elles soient à des distances différentes. Il n'y a pas d'effet de perspective.

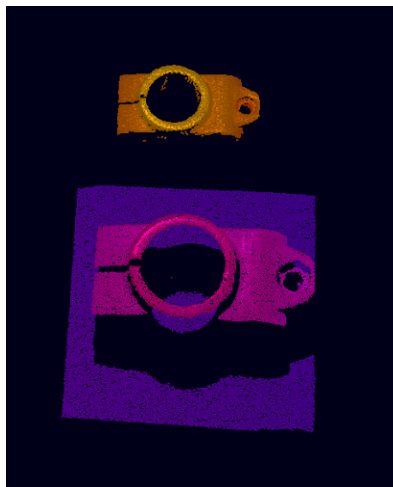


Figure 3 : Image 3D obtenu en utilisant les coordonnées x, y et z. L'image est colorée en fonction de la valeur z du point 3D.

Avantage :

- Dans la matrice, chaque point est entouré de ses voisins x, y.

Limitation :

- Lors de la construction de cette matrice, 2 points peuvent se retrouver dans la même case. Dans ce cas, un des deux points sera perdu. Ceci peut créer une variation de densité de points lorsqu'on affiche les points 3D bruts. Si cela vous gêne, nous vous conseillons d'utiliser la variante 3 : VN_ExecuteSCAN_CameraCOP_MiddleCam2_XYZI

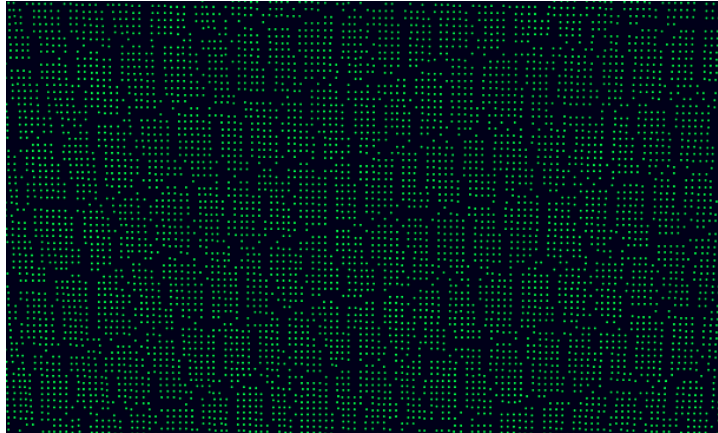


Figure 4 : L'image montrant le nuage de points après arrangement dans la matrice. Un effet de pattern apparait dans le nuage de points.

- La luminance est disponible uniquement lorsqu'un point 3D a été trouvé. L'image de luminance résultante de la matrice contiendra des cases sans données.

3.2. Variante 2 : VN_ExecuteSCAN_Matrix_XYZRGB

Construction de la matrice identique à la variante 1. Seul le format des données change. Les coordonnées X, Y, Z et l'intensité sont définies dans une structure pcl::PointXYZRGB. Le Cirrus3D étant en niveau de gris, les composantes R, G et B sont identiques.

3.3. Variante 3 : VN_ExecuteSCAN_CameraCOP_MiddleCam2_XYZI

Le nuage de points est sous forme de matrice de type 2. La taille de la matrice dépend du Cirrus3D, de la densité et de la ROI utilisés.

Le nuage de points obtenu est calibré et structuré en grille avec un point de vue en perspective.

Pour chaque point 3D, une information de luminance sur 8 bits est disponible. Les coordonnées X, Y et Z sont des réels 32 bits au format binaire à virgule flottante IEEE-754 simple précision.

Avantage :

- Le résultat du scan est transféré durant le scan, ce qui minimise le temps du cycle d'acquisition (prise image+transfert).

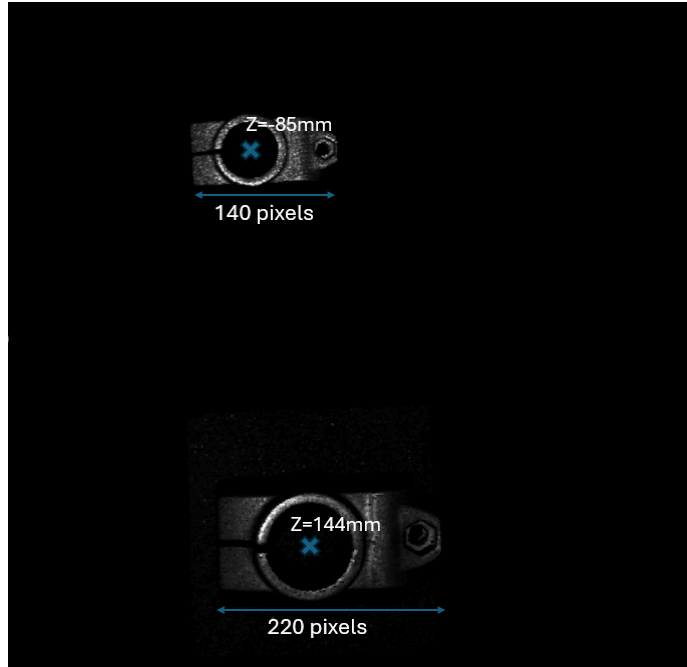


Figure 5 : Représentation 2.5D de la matrice (les coordonnées x et y étant ignorés)

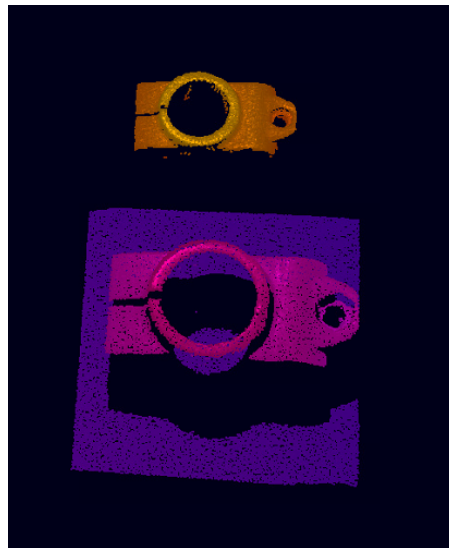


Figure 6 : Image 3D obtenu en utilisant les coordonnées x, y et z. L'image est colorée en fonction de la valeur z du point 3D.

Limitation :

- La luminance est disponible uniquement lorsqu'un point 3D a été trouvé. L'image de luminance résultante de la matrice contiendra des cases sans données. Une case « vide » contient la valeur NaN.

3.4. Variante 4 : Carte de profondeur (à venir)

Les données sont sous forme d'image rectifiée. Les coordonnées x et y sont implicites et peuvent être obtenu grâce au `CoordinateScale(A)` et `CoordinateOffset(B)`. Ce format est une carte de profondeur : une image construite en pixels de 16 bits, où les pixels les plus sombres correspondent à des points plus proches du Cirrus. Les coordonnées 3d x,y,z des points 3d peuvent ensuite être extraites des coordonnées u,v des pixels et de leur `pixVal` correspondant :

$$x = x_scale * u + x_offset$$

$$y = y_scale * v + y_offset$$

$$z = z_scale * pixVal + z_offset$$

Avec, dans ce cas, $x_scale = y_scale$.

Les pixels non valides (les éléments de l'image rectifiée qui ne contiennent pas de données valides) sont définis par une valeur sélectionnée par l'utilisateur (généralement 0 ou 65535). Ce format a été conçu pour être compatible avec une grande variété d'applications, y compris celles qui sont conçues pour être utilisées avec des images 2D.

Cependant, ce format présente quelques inconvénients :

- Les coordonnées X & Y étant implicites, il y a une perte de précision (les points 3D bruts sont rarement générés au centre exact du pixel)
- le Cirrus génère nativement une couverture plus dense de points 3d sur les objets les plus proches que sur les objets plus éloignés. Afin de minimiser le nombre de « trous » dans l'image rectifiée, l'image est sous-échantillonnée, perdant environ 20% de résolution par rapport au format `Matrix_XYZ`.
- le Cirrus génère nativement des points 3D à partir d'une vue en perspective, ce qui signifie que si le Cirrus est accroché au-dessus d'un bac, il est capable (au moins partiellement) de scanner les côtés du bac, et génère des points 3D partageant (à peu près) les mêmes coordonnées x,y, mais avec un z différent. Avec ce format rectifié, ces points sont perdus : dans le cas de plusieurs points 3d projetés sur le même pixel, on ne garde que celui le plus proche du Cirrus.

La résolution de l'image (nombre de lignes et de colonnes) ne varie qu'en fonction de la densité de balayage sélectionnée ; Il s'agit d'un choix arbitraire pour faciliter l'intégration avec des applications où la résolution de l'image est fixe.

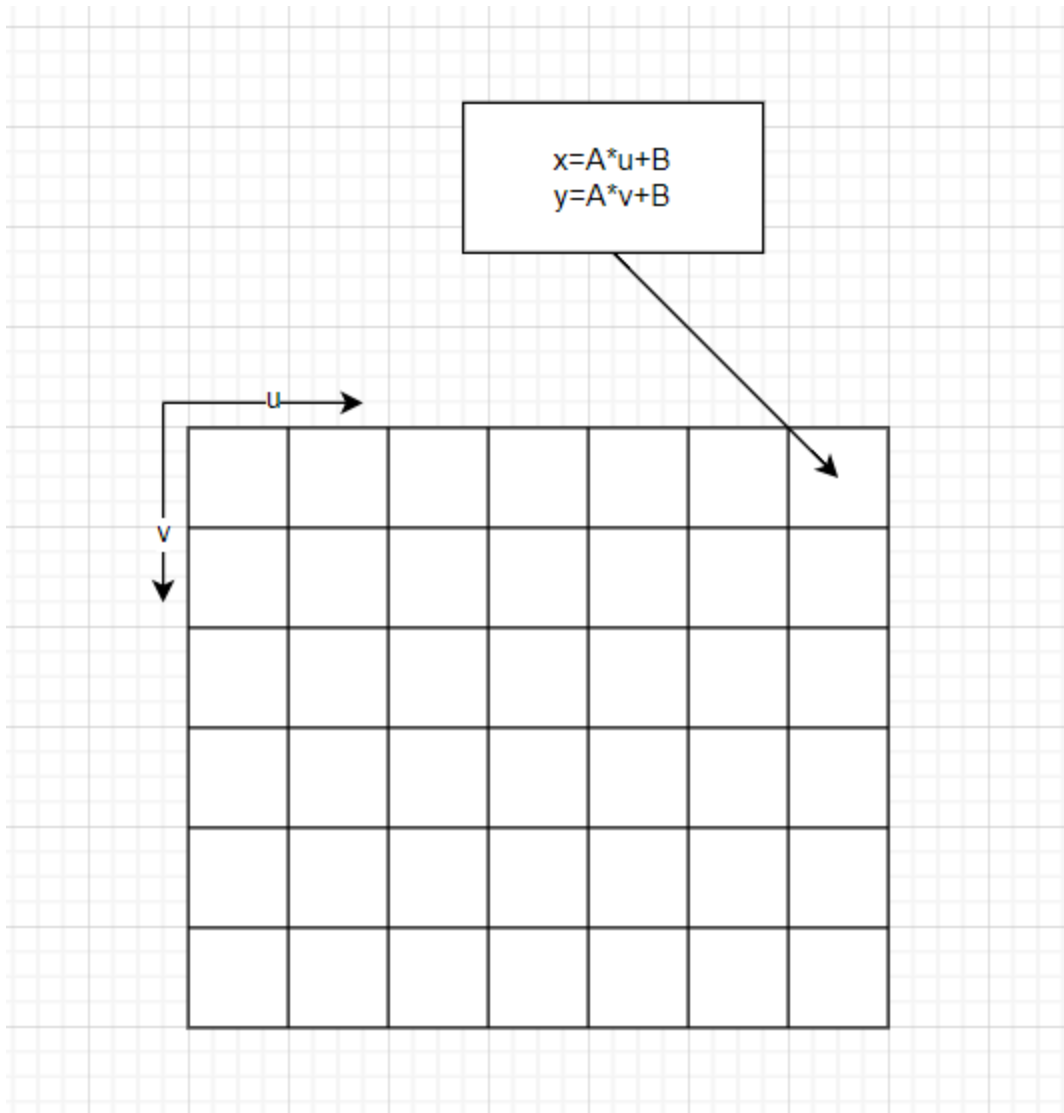


Figure 7 : Représentation d'une carte de profondeur