

Types d'items disponibles dans Computree

en_US.png [...english version of this page](#)

Computree comporte des items de différentes natures, afin de gérer les différents types de données nécessaires aux algorithmes développés. Un certain nombre de types d'items standards sont proposés dans le **coeur**, ce qui permet de rendre compatibles les plugins entre eux.

Cependant si nécessaire, un plugin peut définir des types d'items ad hoc, utilisés uniquement par ce plugin.

Gestion des points et types d'items correspondants

Computree est conçu pour gérer des nuages de points 3D. A ce titre les points ont une place spécifique dans la plateforme. Ainsi les points (classe d'item : *CT_Point*) ne sont pas des items, afin d'optimiser la gestion de la mémoire.

Les classes dans Computree sont préfixées par un code permettant de déterminer leur plugin d'appartenance. Le préfixe *CT* signifie que la classe est implémentée dans le coeur de CompuTree.

Un point est simplement un tableau de trois nombres décimaux (de type float, à 7 chiffres significatifs) contenant ses coordonnées (x,y,z). Tous les points créés dans une session sont regroupés dans un **dépôt de points**, et indexés de façon continue.

A ce jour, l'utilisation de nombres décimaux de type float (pour des raisons d'optimisation mémoire), pose des problèmes quand les nuages de points sont projetés en coordonnées géographiques comme par exemple en Lambert 93, du fait du nombre important de chiffres significatifs nécessaires. Dans ces cas il y aura perte de précision par troncature lors de l'importation des points dans **Computree**. Pour le moment il est donc préférable de travailler en géométrie capteur (centre de scan à (0,0,0)).

Il est prévu à court terme d'intégrer un système de changement de repère à l'importation pour éviter ce problème. Cela sera en particulier nécessaire pour l'utilisation de données LiDAR aériennes.

Afin de pouvoir gérer / afficher les points, ceux-ci sont gérés à l'aide d'items "contenant des points", et donc héritant de la classe d'item : *CT_AbstractItemDrawableWithPointCloud*.

Il existe deux types d'items de ce genre à ce jour :

- Les **scènes** (classe d'item : *CT_Scene*), conçue pour gérer des nuages de points importants, en optimisant l'occupation mémoire des index. Dans le cas d'une scène tous les points sont ajoutés en une fois à la création.
- Les **clusters** (classe d'item : *CT_PointCluster*), plutôt adaptés à la gestion de "petits" ensembles de points (même s'il n'y a pas de limite définie au nombre de points d'un cluster). Dans ce cas les points peuvent être ajoutés successivement au cours de l'étapes. De plus les clusters mettent à jour en temps réel leur barycentre (ce que ne font pas les scènes), qui sont donc affichables.

Il est possible d'avoir des algorithmes génériques pouvant utiliser indifféremment tout type d'item dérivé de *CT_AbstractItemDrawableWithPointCloud*.

Maillages

Sur le même modèle que les points, **Computree** peut gérer des maillages. Ainsi un maillage est constitué de **points**, de **faces** et de **demi-arêtes**. Chacun de ces éléments est géré similairement au points #00 00 00 0000 000 000 000000 000 **dépôt de faces** et un **dépôt de demi-arêtes**.

Comme pour les points, il existe un type d'item gérant l'agencement des **points, faces et demi-arêtes** : les **modèle de mesh** (classe d'item : *CT_MeshModel* #00 0 00000 000 000 00 00000 00 0000000000000000 00 000000000000 00 000000000000 00 maillage.

Attributs

Nous avons vu les items permettant de stocker les informations 3D pour les points, les faces et les demi-arêtes. Cependant, on utilise fréquemment des données associées à ces éléments, tels qu'une intensité, une couleur ou une normale.

Il existe donc des types d'items chargés de stocker ces informations liés aux éléments unitaires cités : les **attributs**.

Il existe un type d'item pour chaque croisement **attribut (nature de donnée) / type d'élément**.

A ce jour, **Computree** gère les **attributs** suivants :

- **Attributs scalaires** : une valeur est attachée à chaque élément. Le type de valeur (entier, décimal, booléen) est spécifié dans un *template*. Les items d'attributs scalaires appartiennent aux classes d'items *CT_PointsAttributesScalarTemplated*, *CT_FaceAttributesScalarTemplated*, *CT_EdgeAttributesScalarTemplated* respectivement pour les points, les faces et les demi-arêtes.
- **Couleurs** : pour chaque élément, on définit une couleur par 4 valeurs entières comprises entre 0 et 255 (rouge, vert, bleu, alpha = transparence). Les items d'attributs "couleur" appartiennent aux classes d'items *CT_PointsAttributesColor*, *CT_FaceAttributesColor*, *CT_EdgeAttributesColor* respectivement pour les points, les faces et les demi-arêtes.
- **Normales / Courbures** : pour chaque élément, on définit une normale par 4 nombres décimaux (float) : un vecteur direction (dx, dy, dz) et une courbure. Les items d'attributs "Normales / Courbures" appartiennent aux classes d'items *CT_PointsAttributesNormal*, *CT_FaceAttributesNormal*, *CT_EdgeAttributesNormal* respectivement pour les points, les faces et les demi-arêtes.

Formes géométriques

Computree propose une série de types d'items pour gérer les formes géométriques 3D, sur une logique commune.

Ainsi il existe un type d'item pour chaque forme géométrique, héritant tous de la classe d'item *CT_AbstractShape* :

- Des **cercles** (classe d'item : *CT_Circle*) : un centre (x,y,x), une direction normée (x,y,z), un rayon
- Des **ellipses** (classe d'item : *CT_Ellipse*) : un centre (x,y,x), une direction normée (x,y,z), un petit rayon et un grand rayon
- Des **cylindres** (classe d'item : *CT_Cylinder*) : un centre (x,y,x), une direction normée (x,y,z), un rayon et une hauteur
- Des **lignes** (classe d'item : *CT_Line*), en fait des segments constitués de deux extrémités (x,y,z)
- Des **polygones 2D** (classe d'item : *CT_Polygon2D*) : une liste de sommets dans un plan horizontal

Il est prévu à moyen terme d'ajouter d'autres formes géométriques : polygones 3D, rectangle, parallélepède, cône, courbes de bézier,...

Grilles de données

Computree propose deux types d'item pour gérer les données sous forme de "grille" :

- Les **grilles 2D** (classe d'item : *CT_Grid2D*) permettent de gérer des grilles en deux dimensions ou **raster**. Ces items contiennent un tableau en deux dimensions, chaque case contenant une valeur.
- Les **grilles 3D** (classe d'item : *CT_Grid3D*) permettent de gérer des grilles en trois dimensions ou **voxels**. Ces items contiennent un tableau en trois dimensions, chaque case contenant une valeur.

Ces classes d'items sont des **templates**, c'est à dire qu'on peut les décliner pour différents types de données dans les cases.

Par exemple des voxels contenant des entiers seront de classe *CT_Grid3D<int>*.

Ces structures fonctionnent de façon très semblables. Elle gèrent les valeurs manquantes (sauf dans le cas de grilles booléennes).

Autres types d'items

Il existe d'autres types d'items plus spécifiques à certains usages :

- Les **positions de scanner** (classe d'item : *CT_Scanner*) stockent la position et les paramètres d'un scan T-Lidar
- Les **rayons** (classe d'item : *CT_Beam*) permettent de gérer un rayon laser, émis depuis un scanner

[Page précédente \(Présentation générale\)](#)

[Retour au sommaire](#)

[Page suivante \(Interface graphique standard\)](#)