INSTITUT NATIONAL
DE L'INFORMATION
GÉOGRAPHIQUE
ET FORESTIÈRE

FORMATION DIFFUSION DE DONNÉES 3D

Clément Drouadaine
Benoît More

20/05/2019

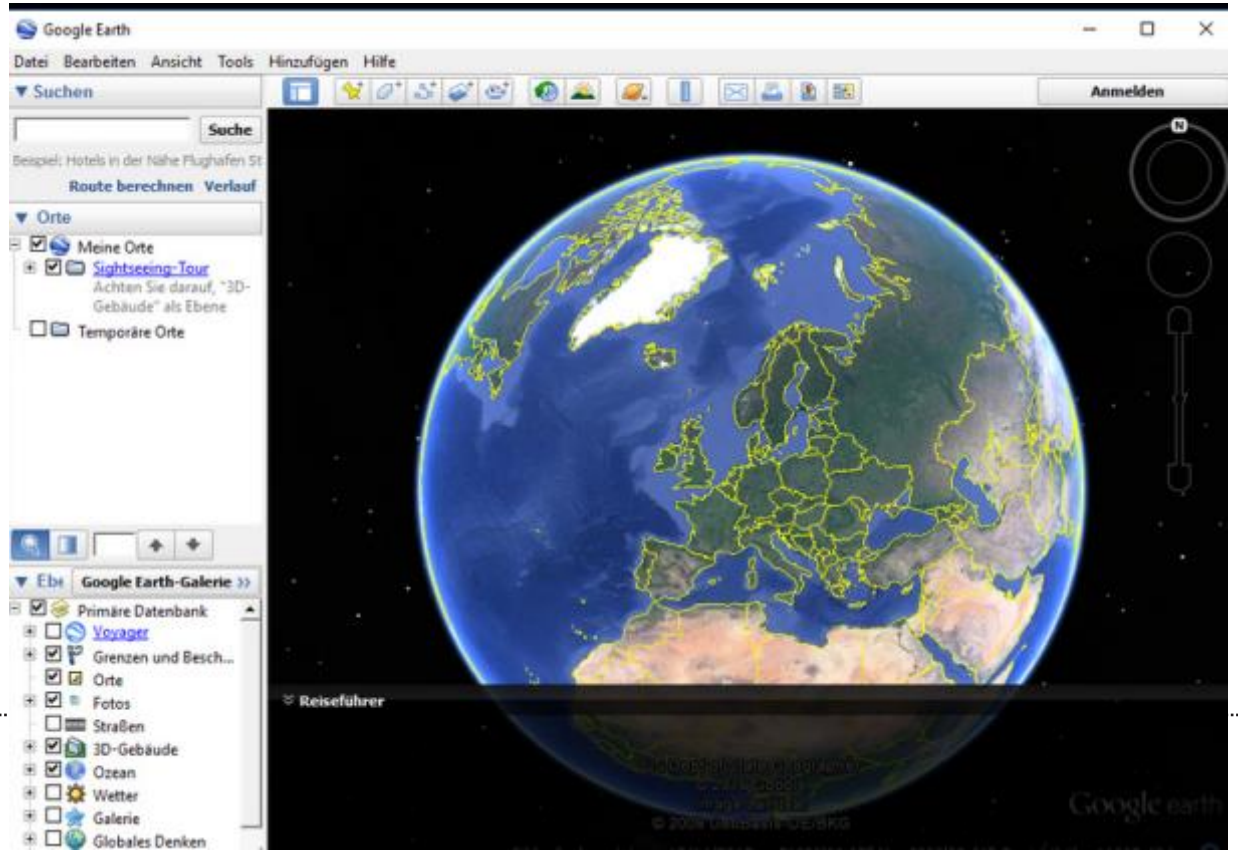
Modèle TN-02.018-1.7

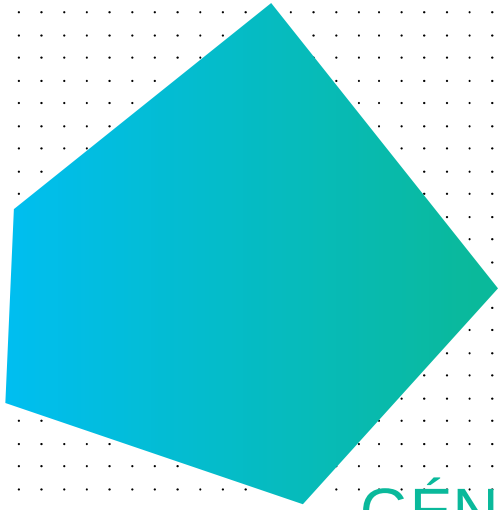
AMOAIGN-
SPP/19.0206

GÉNÉRALITÉS SUR LES DONNÉES 3D

QU'EST LA DIFFUSION 3D ?

- Diffusion d'images 3D
  - Anaglyphes
  - Projection polarisée
  - Projection alternée
- Diffusion en 2D de données 3D



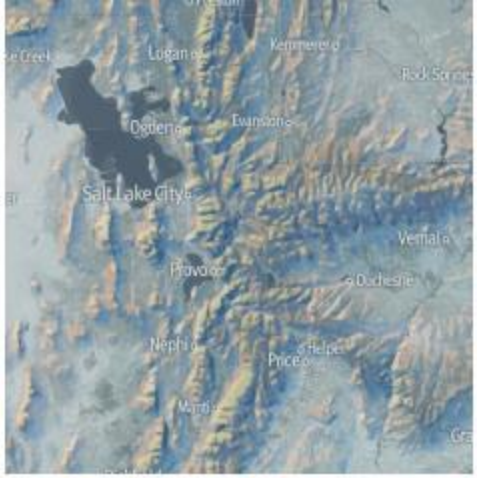


GÉNÉRALITÉS SUR LES DONNÉES 3D

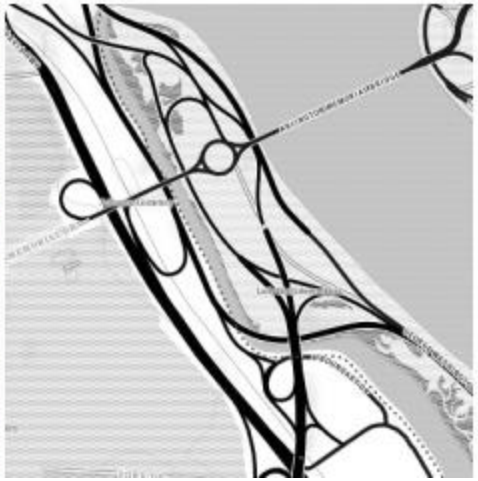
Tuilage 3D


NEW MARKET
Marine
Kattenburg
Wittenburg
Oostenburg
Plantage
Artis
UvA Roetterseiland
Oosterpark


Lake
Rotomahana


Salt Lake City
Ogden
Evanston
Provo
Orchestra
Nephi
Price
Helper

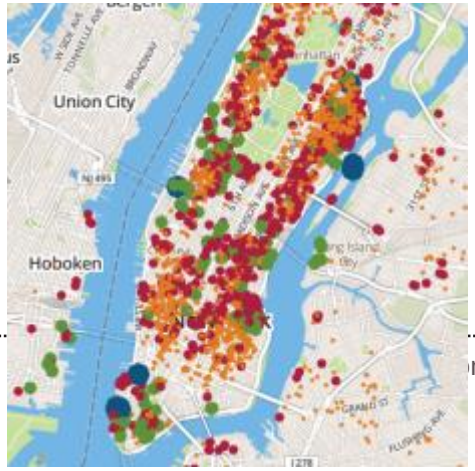

mount
auburn
coolidge
hill
mount auburn
cemetery
cambridge cemetery
north brighton
allston



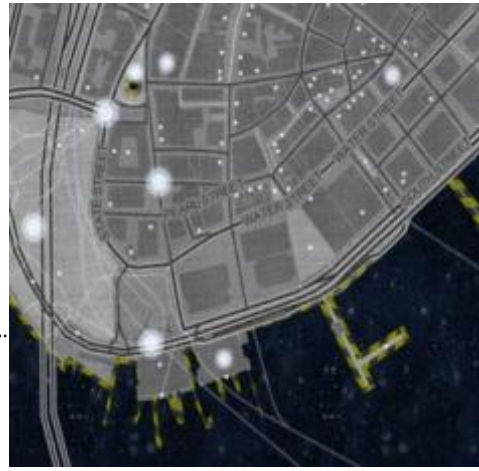

Schmallauigädmer


Berkeley
Oakland
San Francisco
San Leandro
Daly City
San Mateo


SAN
FRANCISC
280


Union City
Hoboken

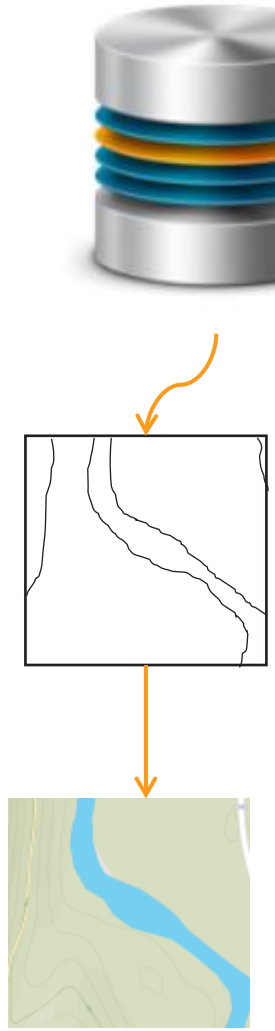

16th St
Mission
District
18th St
20th St
21st St
22nd St
23rd St
Noe
Valley
24th St
25th St
28th St
Dolores St
Guerrero St
Capp St
Ness Ave
York St
Bryant St
Alabama St
Florida St
Treat Ave



DEUX TYPES DE TUILAGE :

Tuilage axé rendu

- centré sur la visualisation fluide des objets vectoriels.



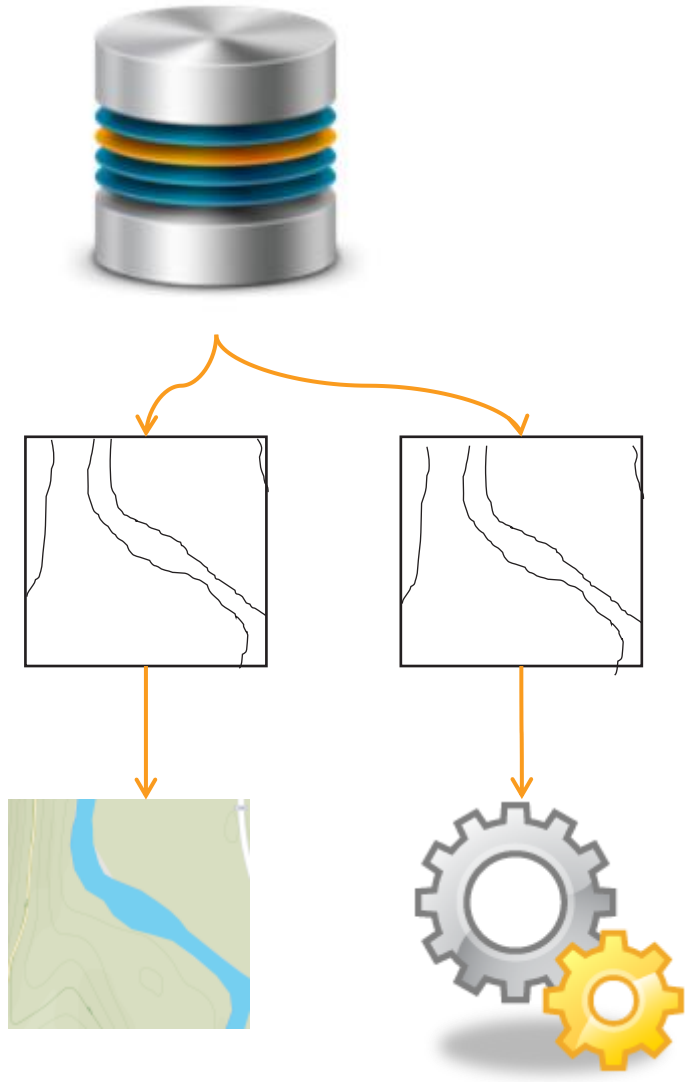
DEUX TYPES DE TUILAGE :

Tuilage axé rendu

- centré sur la visualisation fluide des objets vectoriels

Tuilage axé objets

- axé sur le maintien de l'intégrité des objets vectoriels pour le stockage et l'archivage analytique

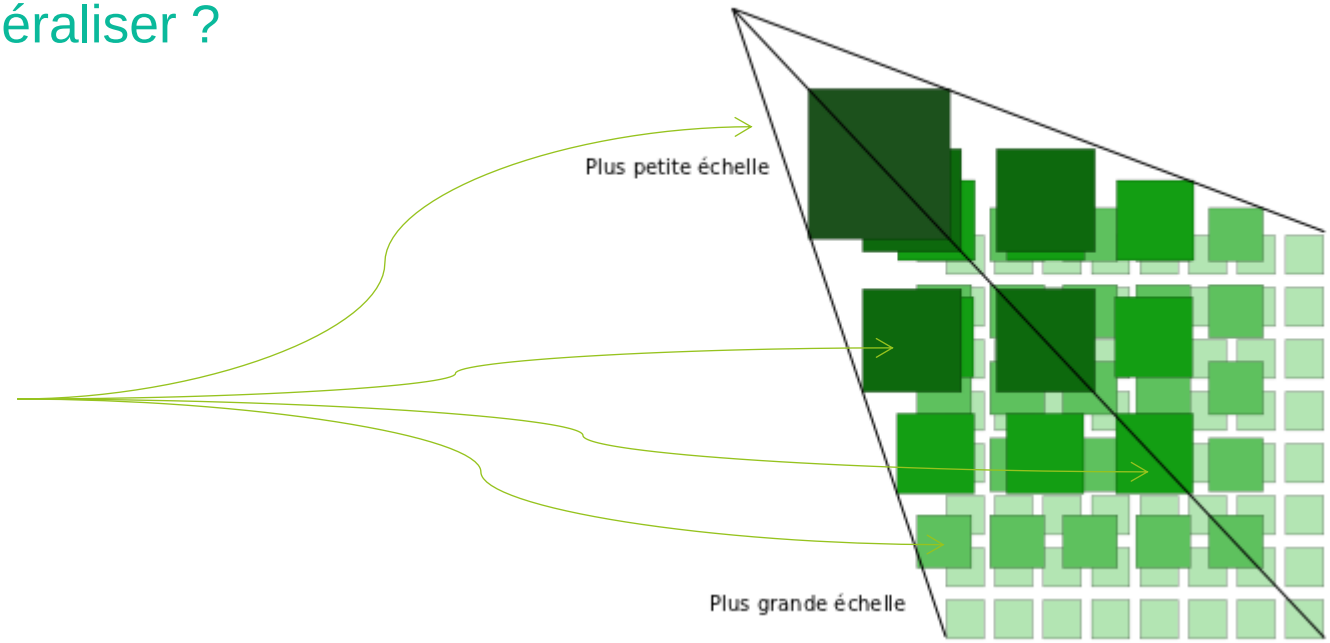


IMPACT DE CE CHOIX

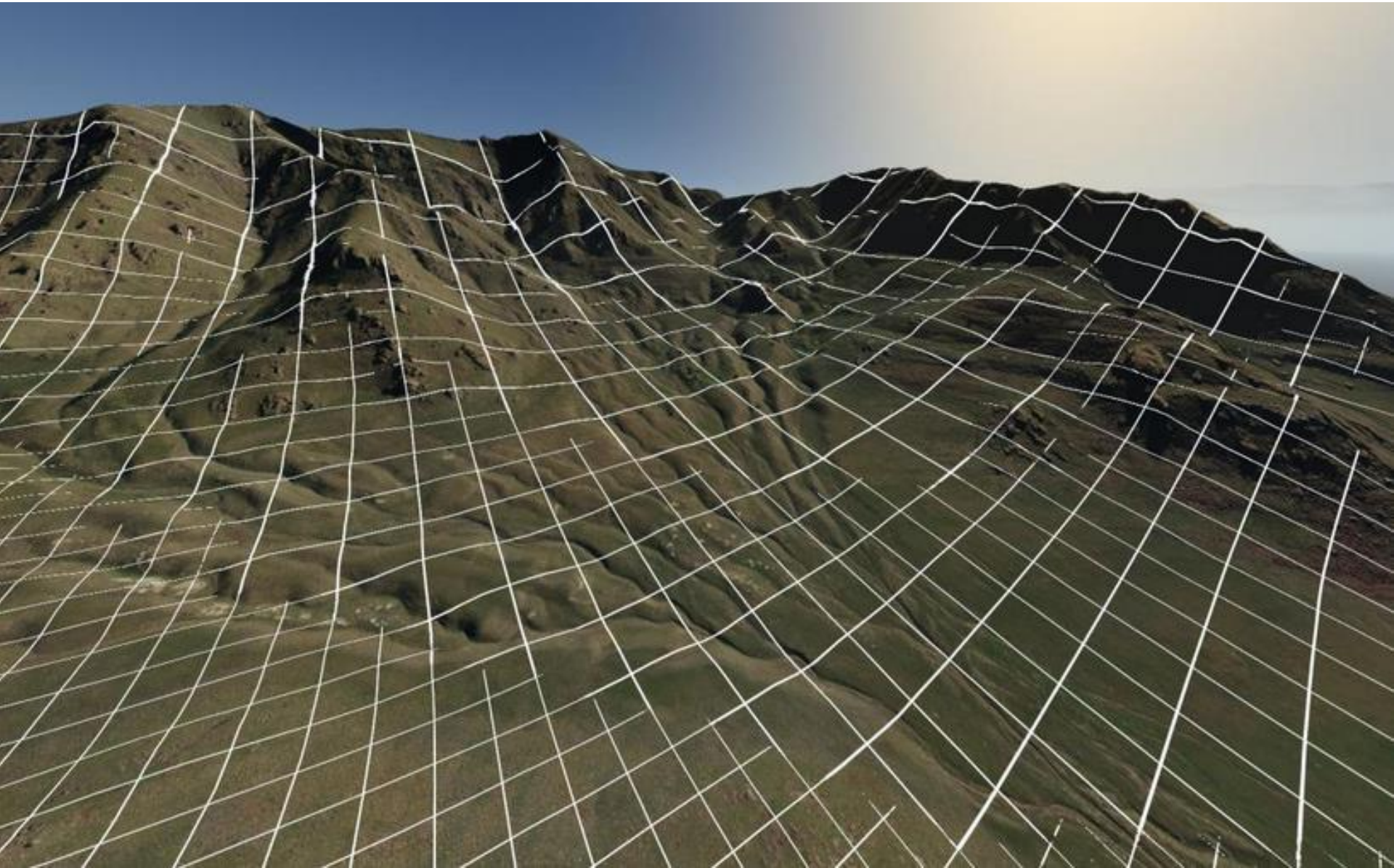
- Format
  - Binaire plus rapide à parser
  - Autres formats plus faciles à traiter suivant telle ou telle propriété
- Modélisation
- Pyramidage
  - Nombre de niveaux important pour la visualisation
  - Peu de niveaux pour le traitement

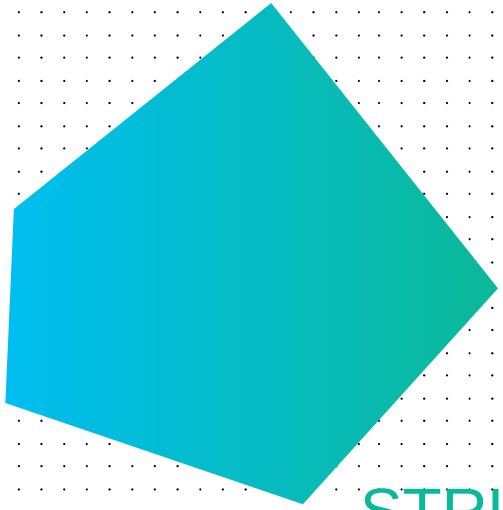
LES STRATÉGIES DE TUILAGE

- Quelle pyramide ?
- Découpage des objets ?
- Comment généraliser ?



DIFFICULTÉ SUPPLÉMENTAIRE POUR LA 3D





STRUCTURE DES DONNÉES

1

UTILITÉ DES STRUCTURES DE DONNÉES

STRUCTURER SES DONNÉES, À QUOI ÇA SERT ?

- Besoin de rechercher des données pour l'affichage : par zone, par voisinage, par direction…
- Nécessité d'ordonner les données pour une recherche rapide.
- Quantité de données énormes en géo.
- Performances pouvant aller du simple au centuple sur des jeux de données volumineux.

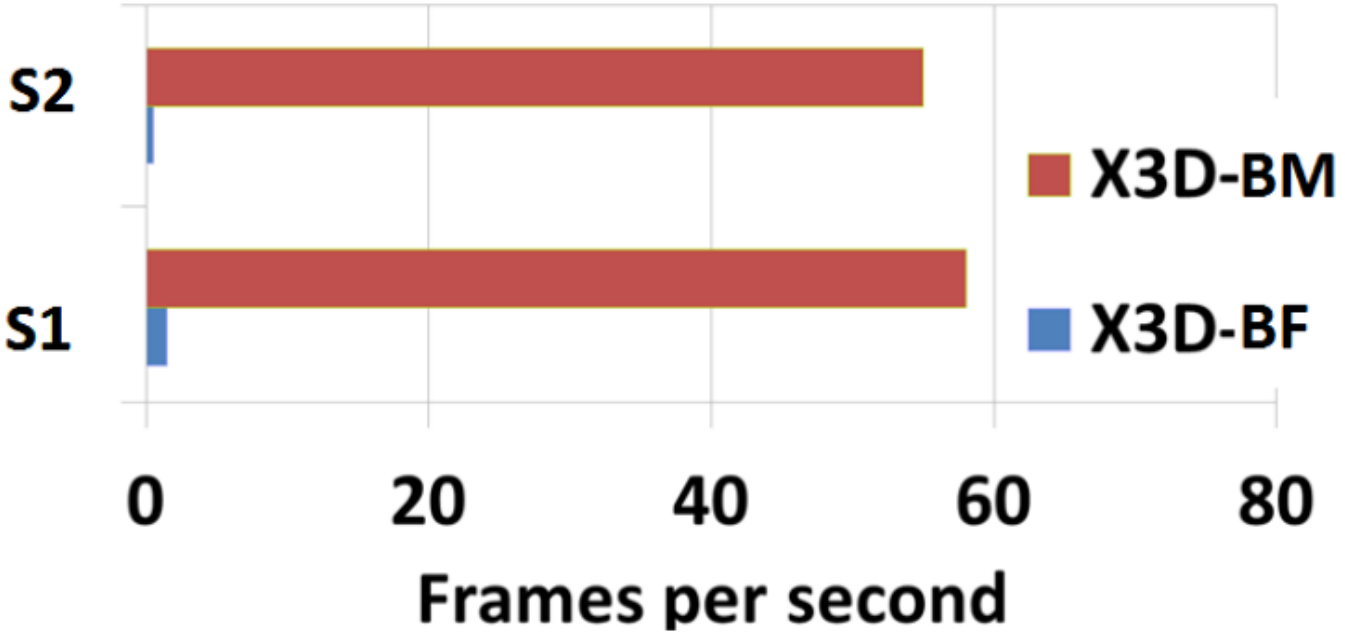
STRUCTURER SES DONNÉES, À QUOI ÇA SERT ?

- Exemples de requêtes nécessitant une structure plus rapide qu'un simple parcours successif de tous les objets :

  - Quelles sont les boulangeries proches de moi ?
  - Affiche la ville de Saint-Mandé.
  - Affiche les objets devant moi, jusqu'à 90m. (Navigation 3D)
  - Quelles routes intersectent l'A4 ?
  - Quels bâtiments longent cette route ?

COMPARAISON DE PERFORMANCES

- X3D-BM : toutes les géométries avec le même matériau = 1 nœud
- X3D-BF : 1 bâtiment = 1 nœud



- Plus de détails dans l'ER du testbed 13.

2

DIFFÉRENTS TYPES D’ARBRES

QUADTREE

- Un arbre quaternaire (ou *quadtree*) est une **structure de données arborescente** dans laquelle chaque nœud interne a **exactement quatre enfants**. Les quadrees sont le plus souvent utilisés pour diviser un espace bidimensionnel en le subdivisant récursivement en quatre quadrants ou régions. Les données associées à une cellule feuille varient selon l'application, mais la cellule feuille représente une "unité d'information spatiale intéressante".

QUADTREE

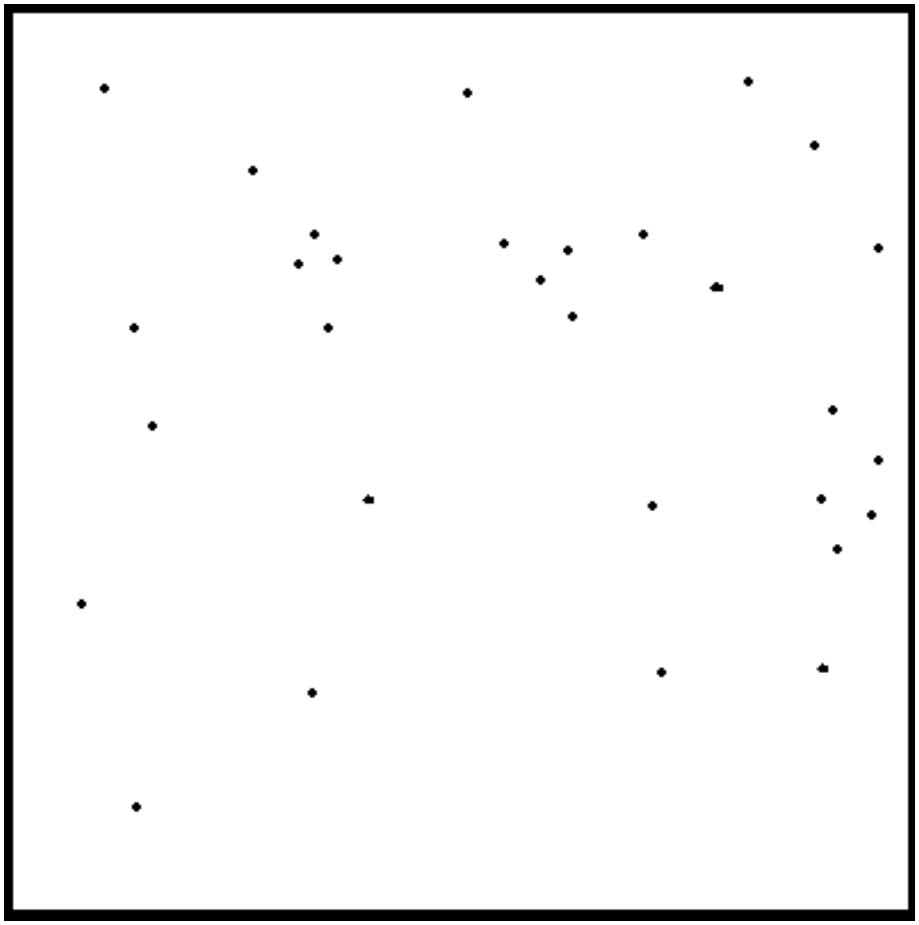
- Un arbre quaternaire (ou *quadtree*) est une **structure de données arborescente** dans laquelle chaque nœud interne a **exactement quatre enfants**. Les quadrees sont le plus souvent utilisés pour diviser un espace bidimensionnel en le subdivisant récursivement en quatre quadrants ou régions. Les données associées à une cellule feuille varient selon l'application, mais la cellule feuille représente une "unité d'information spatiale intéressante".

QUADTREE

- C’est une structure simple à concevoir et à implémenter, et qui peut offrir des performances correctes si les données s’y prêtent.
- Facile à mettre à jour.
- Relativement pratique pour de l’indexation spatiale.

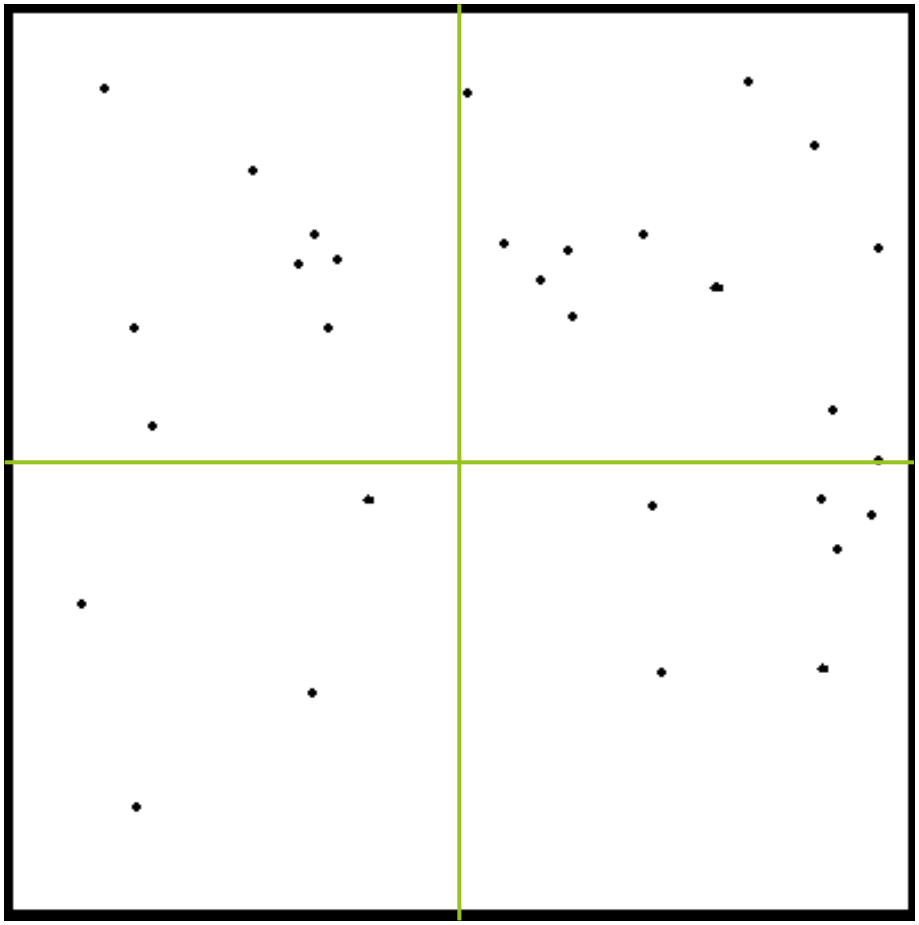
LE QUADTREE, CONCRÈTEMENT

Le quadtree basique (max 2 points par zone) :



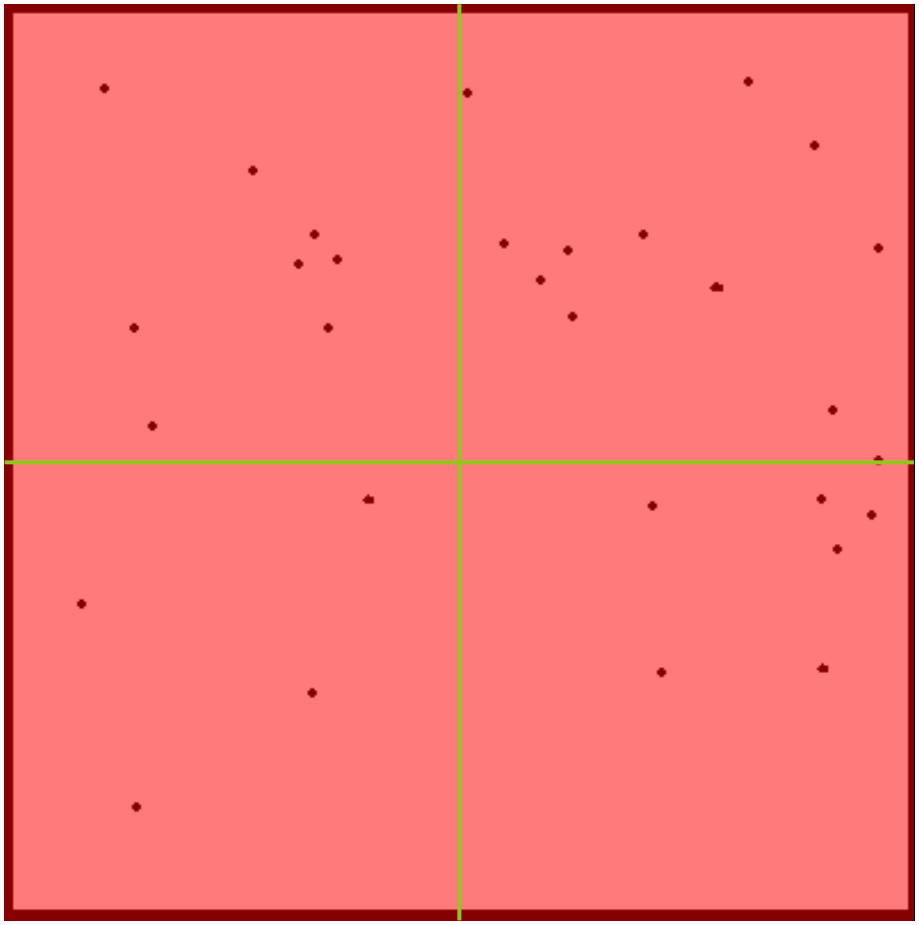
LE QUADTREE, CONCRÈTEMENT

Le quadtree basique (max 2 points par zone) :



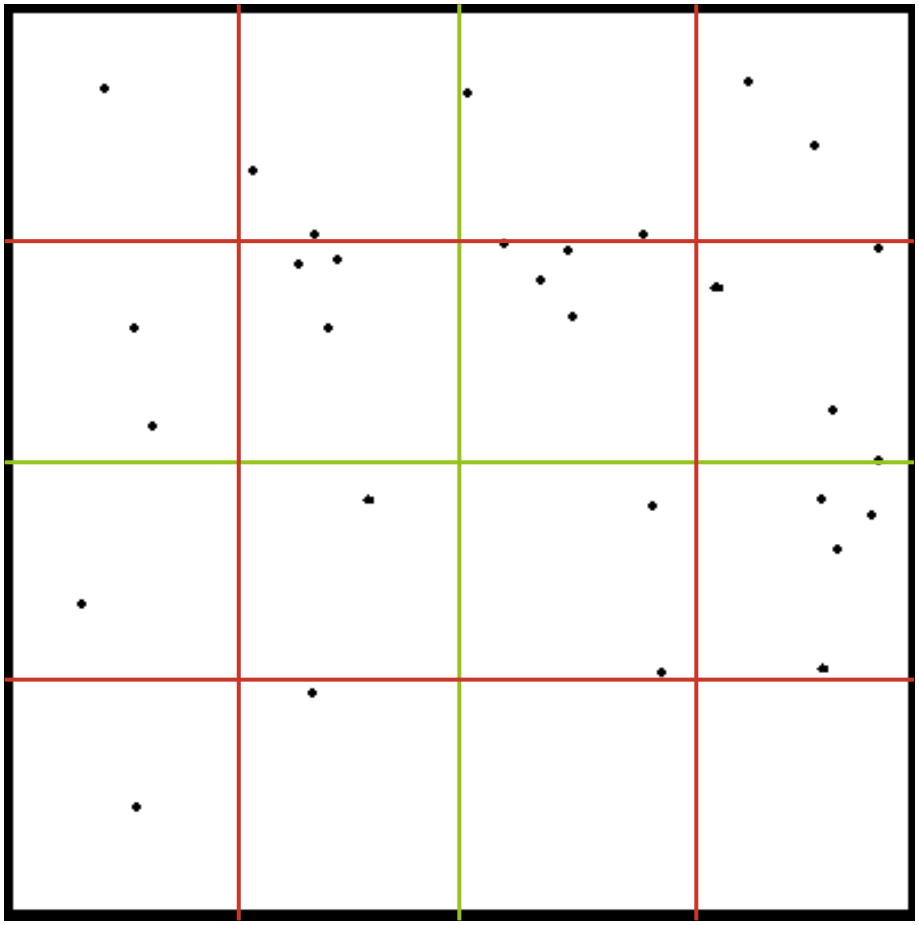
LE QUADTREE, CONCRÈTEMENT

Le quadtree basique (max 2 points par zone) :



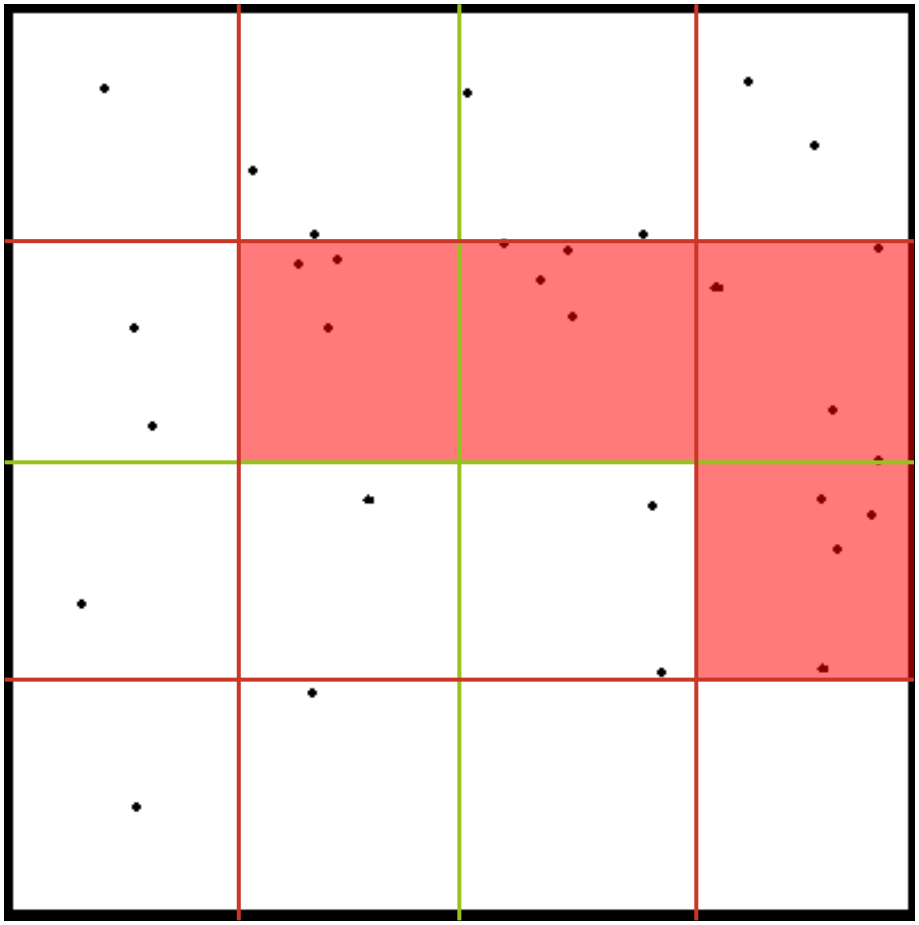
LE QUADTREE, CONCRÈTEMENT

Le quadtree basique (max 2 points par zone) :



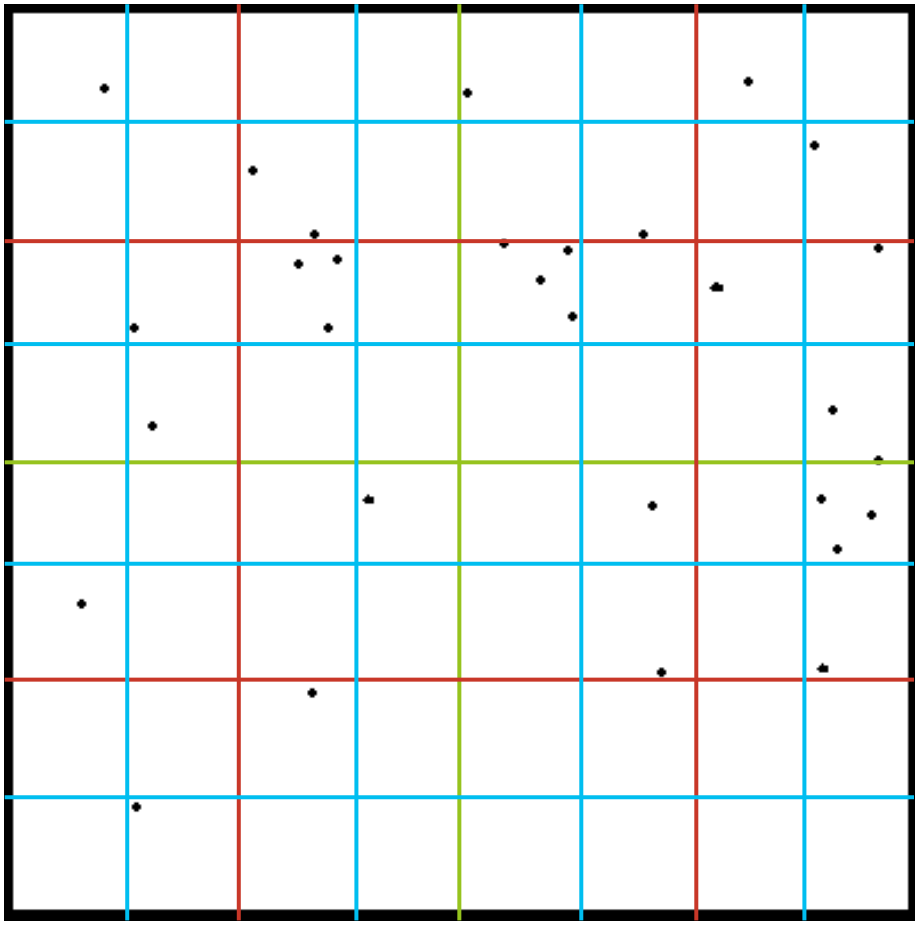
LE QUADTREE, CONCRÈTEMENT

Le quadtree basique (max 2 points par zone) :



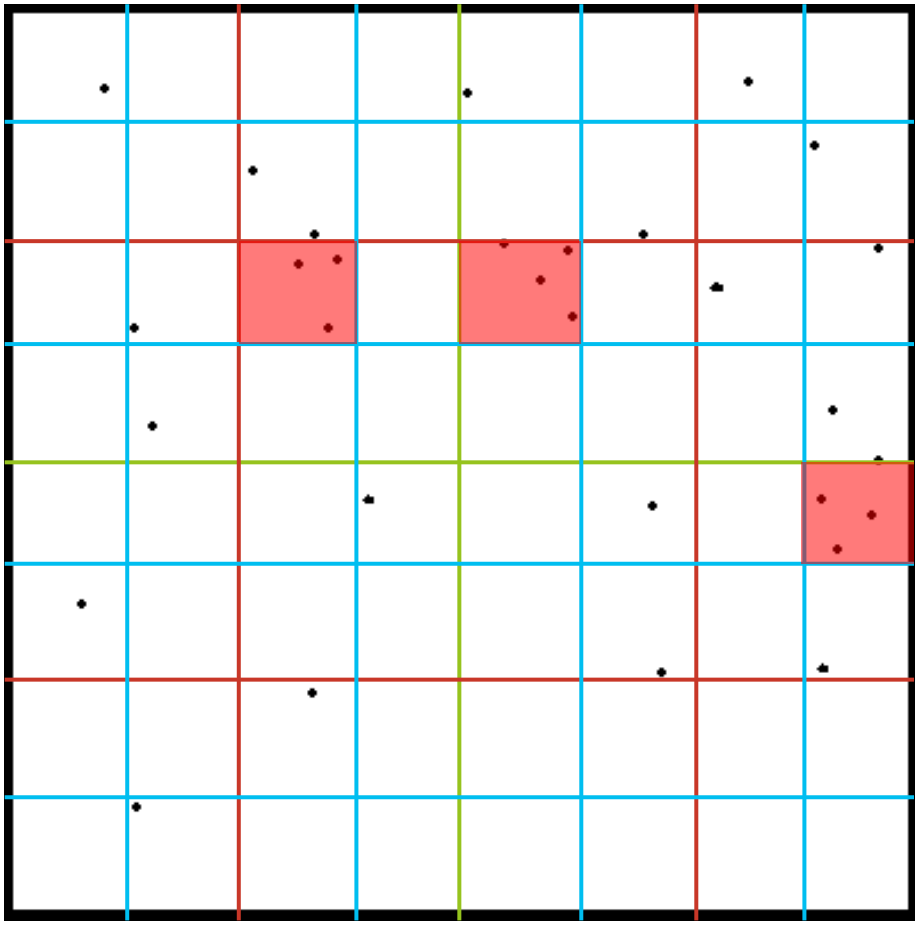
LE QUADTREE, CONCRÈTEMENT

Le quadtree basique (max 2 points par zone) :



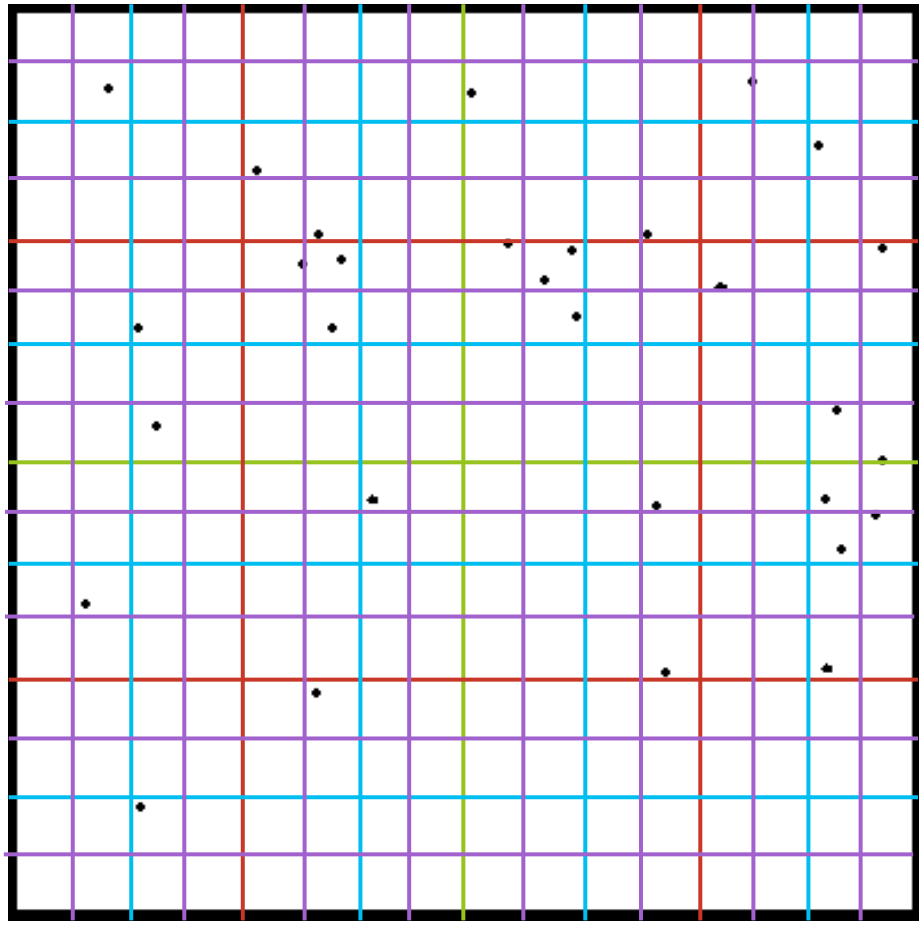
LE QUADTREE, CONCRÈTEMENT

Le quadtree basique (max 2 points par zone) :



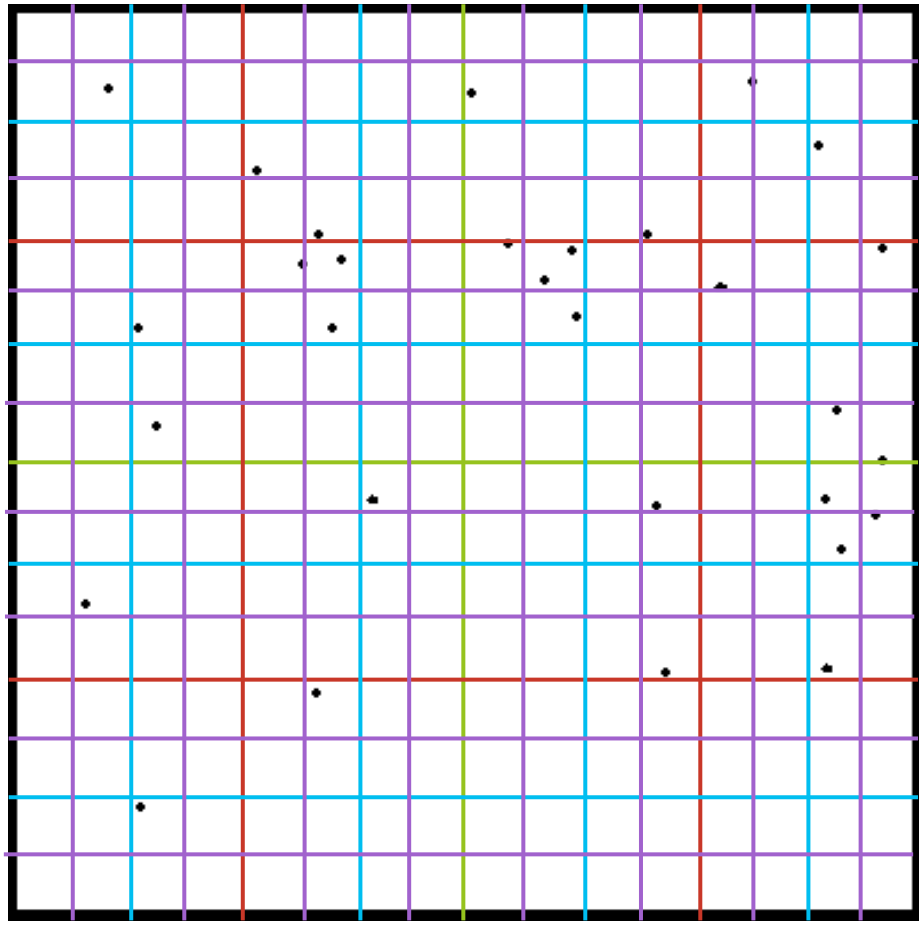
LE QUADTREE, CONCRÈTEMENT

Le quadtree basique (max 2 points par zone) :



LE QUADTREE, CONCRÈTEMENT

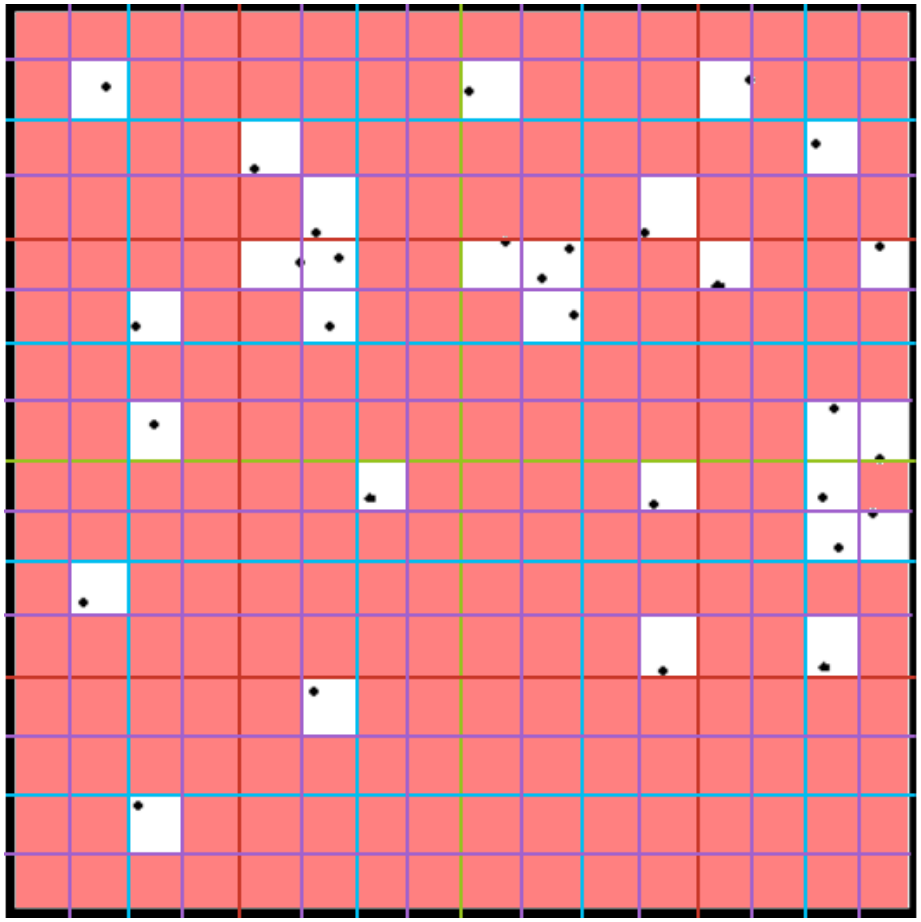
Le quadtree basique (max 2 points par zone) :



LE QUADTREE, CONCRÈTEMENT

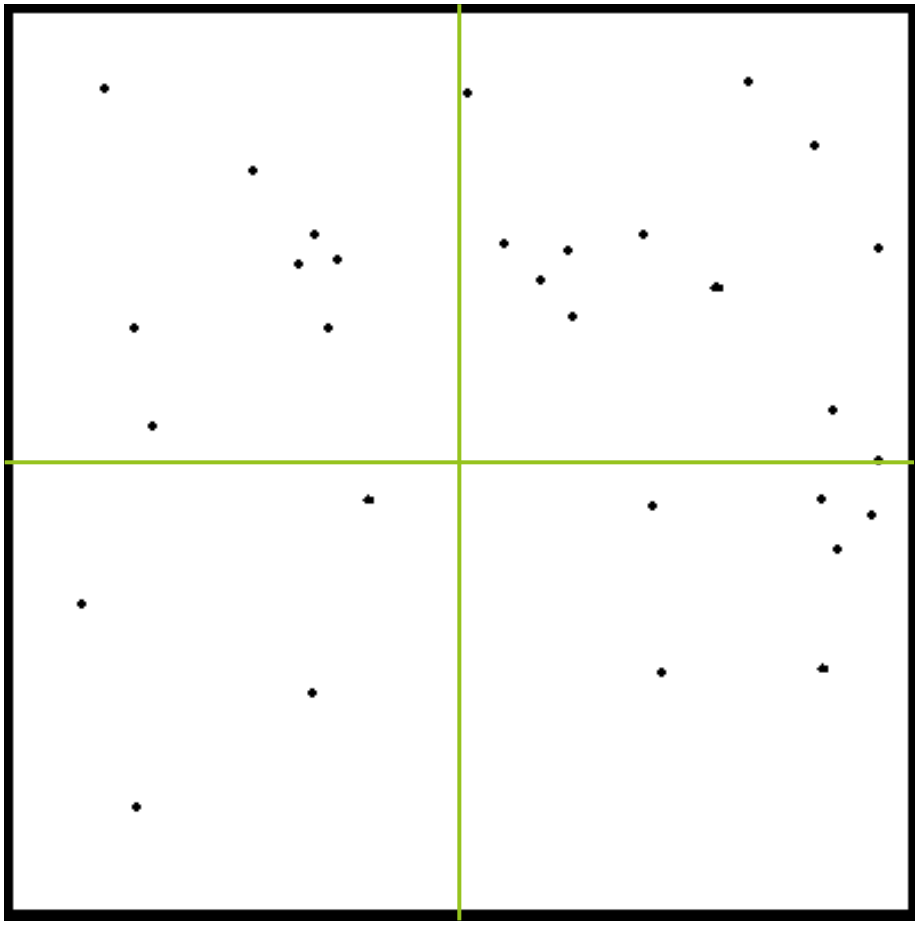
Le quadtree basique : beaucoup de découpages inutiles et de nœuds vides

227 cases vides sur 256 !



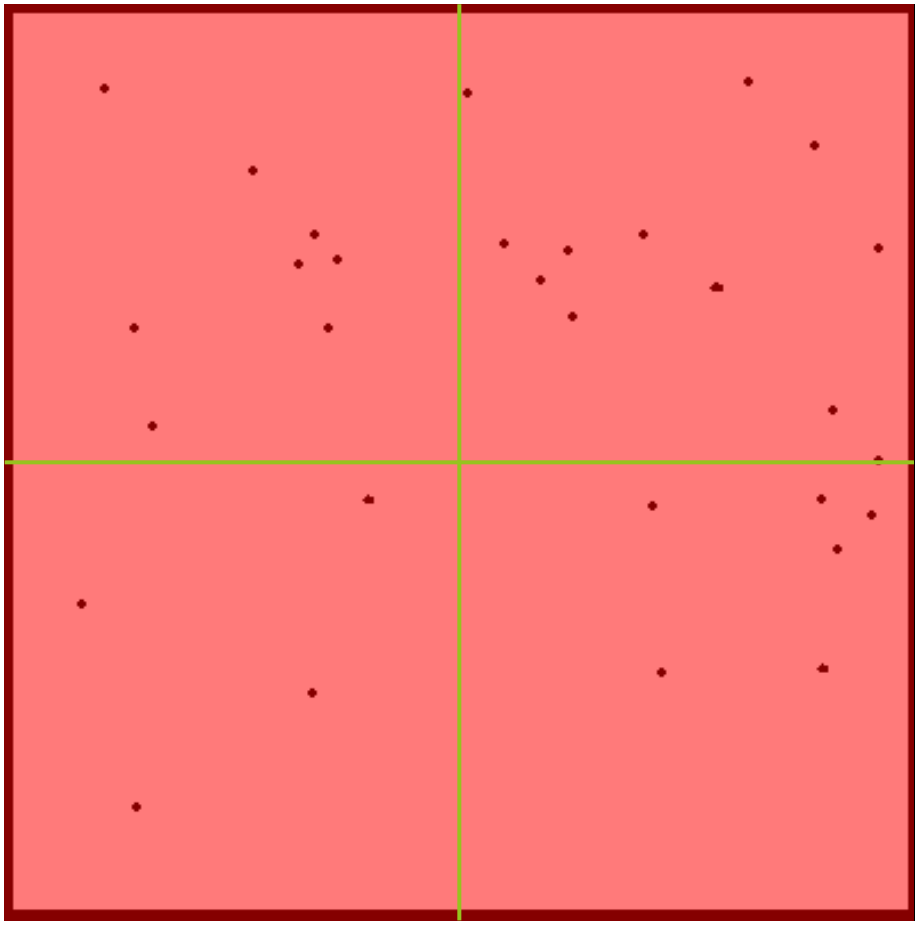
LE QUADTREE, CONCRÈTEMENT

Le quadtree adaptatif (max 2 points par zone) :



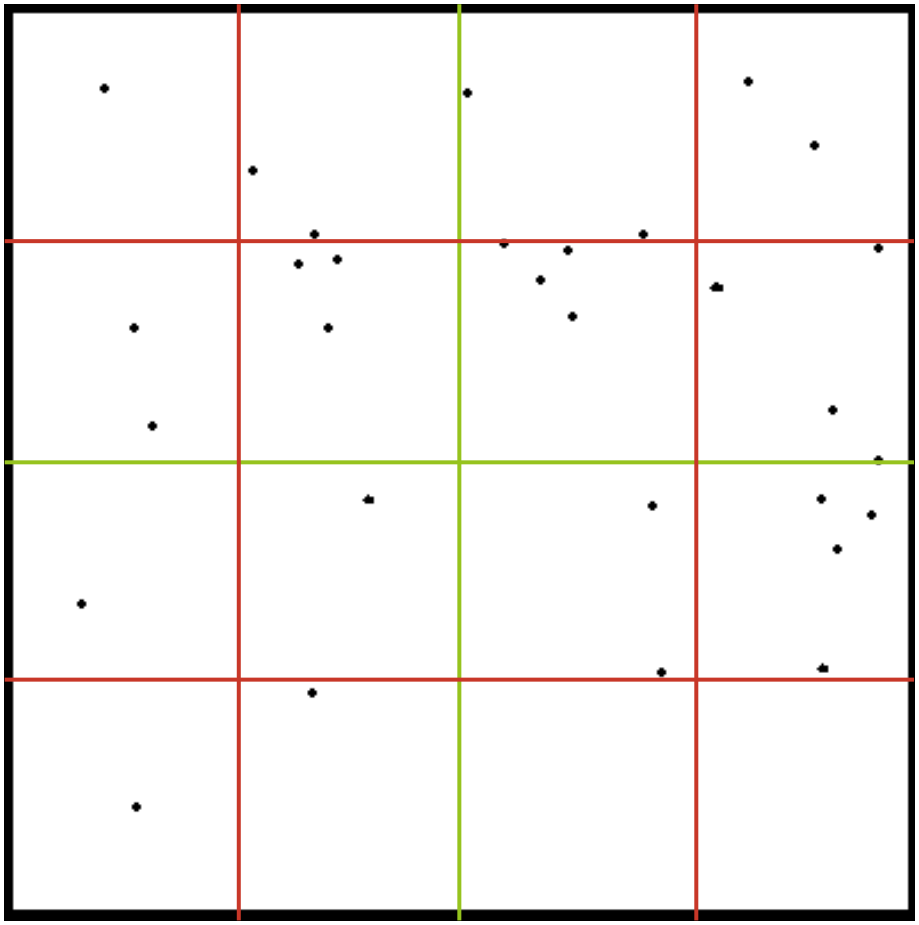
LE QUADTREE, CONCRÈTEMENT

Le quadtree adaptatif (max 2 points par zone) :



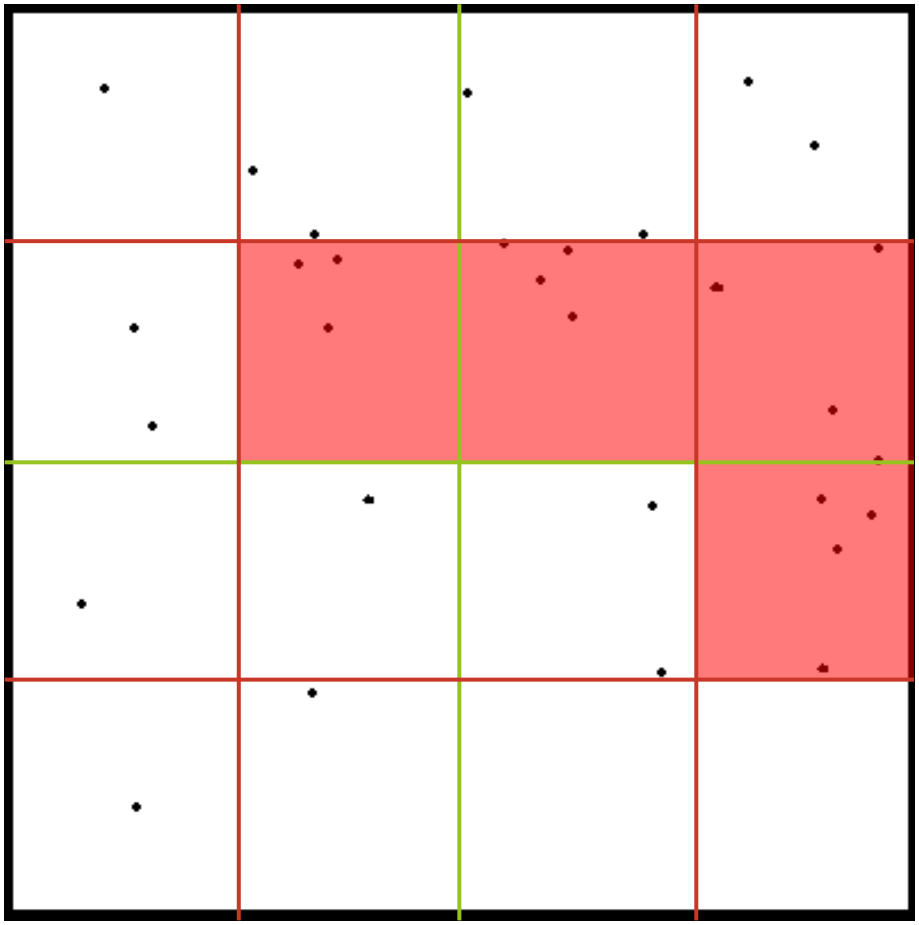
LE QUADTREE, CONCRÈTEMENT

Le quadtree adaptatif (max 2 points par zone) :



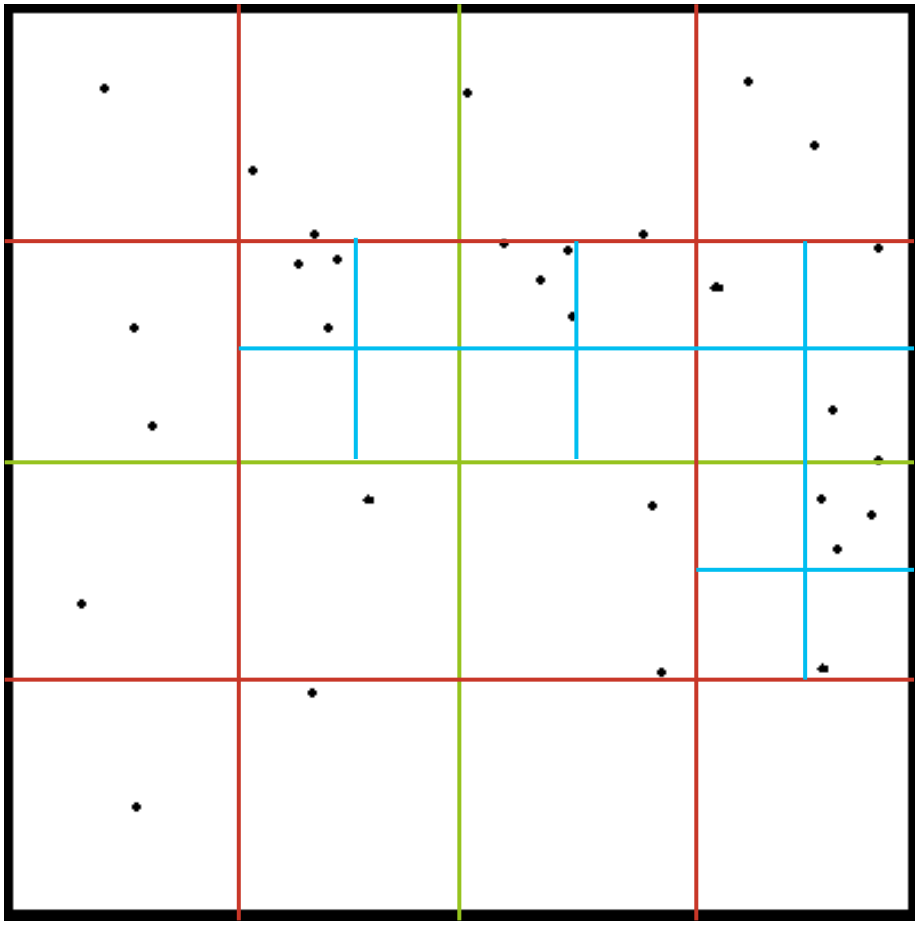
LE QUADTREE, CONCRÈTEMENT

Le quadtree adaptatif (max 2 points par zone) :



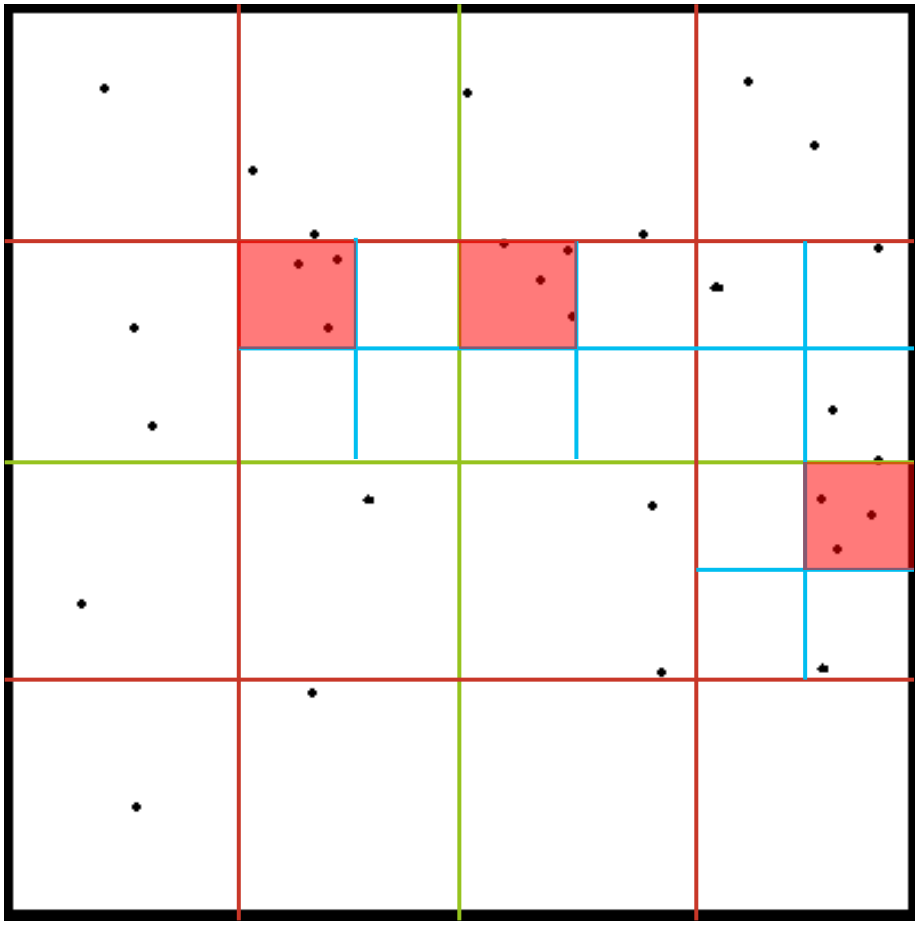
LE QUADTREE, CONCRÈTEMENT

Le quadtree adaptatif (max 2 points par zone) :



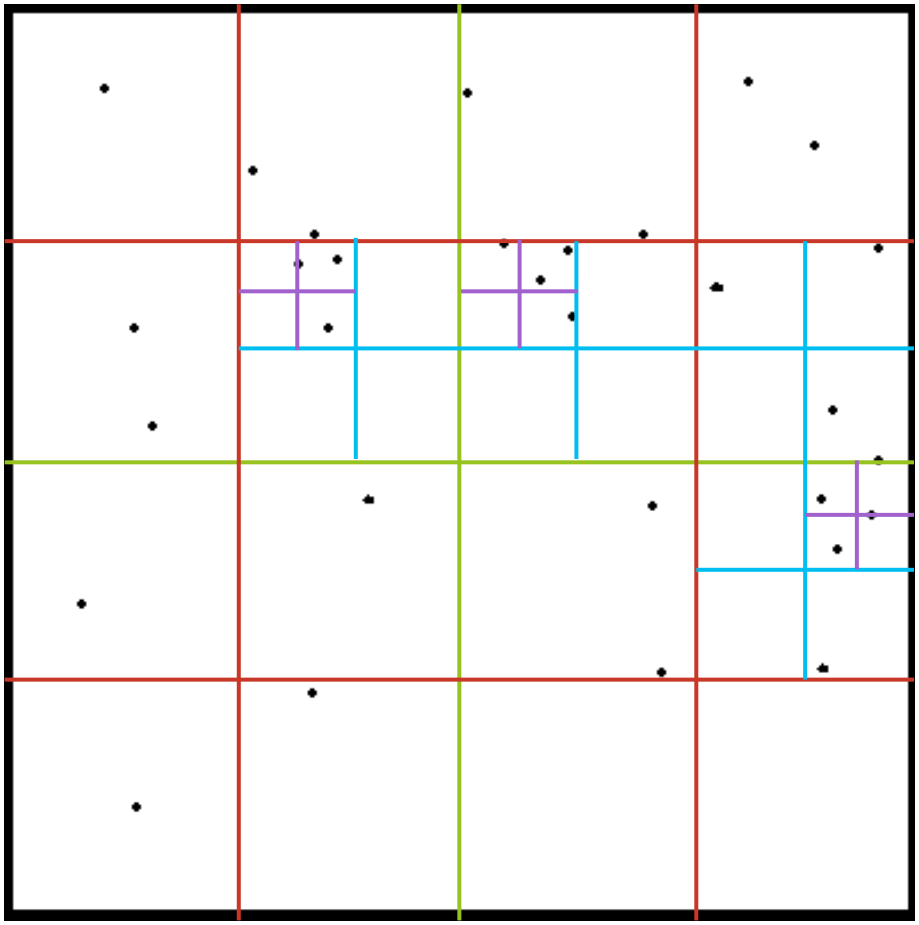
LE QUADTREE, CONCRÈTEMENT

Le quadtree adaptatif (max 2 points par zone) :



LE QUADTREE, CONCRÈTEMENT

Le quadtree adaptatif (max 2 points par zone) :

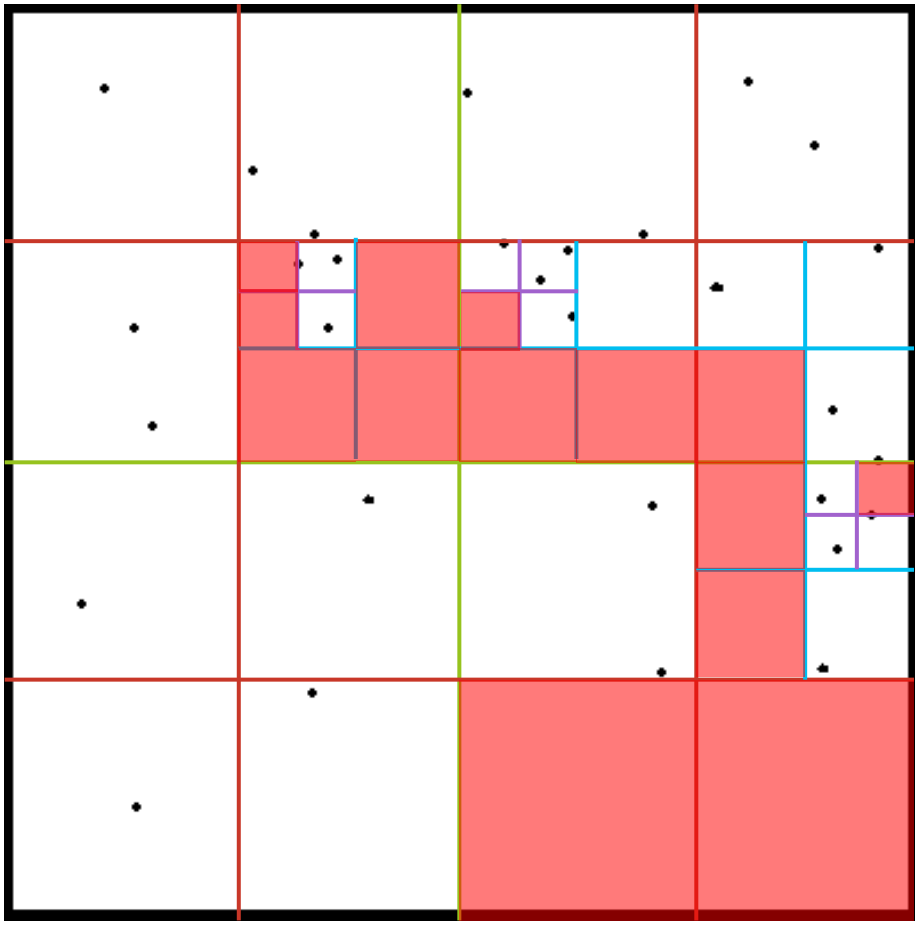


LE QUADTREE, CONCRÈTEMENT

Le quadtree adaptatif : peu de nœuds inutiles

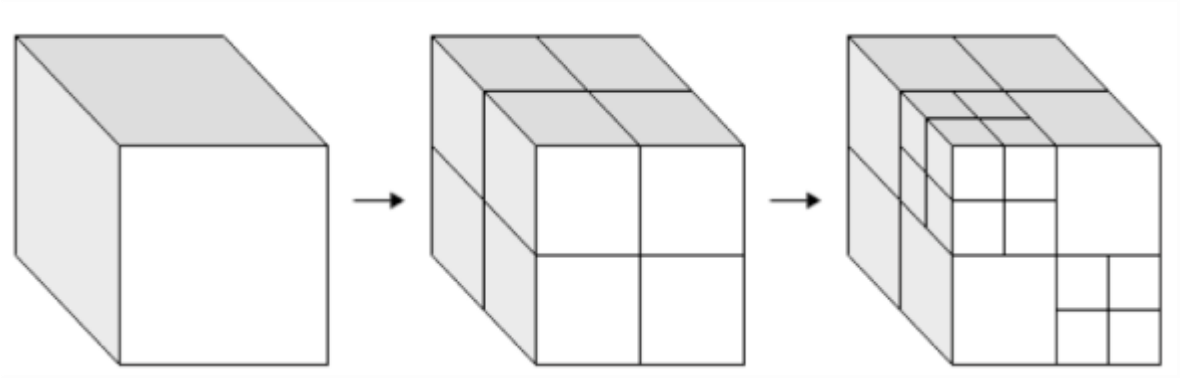
14 cases vides sur 37 !

Un graphe bien moins volumineux.



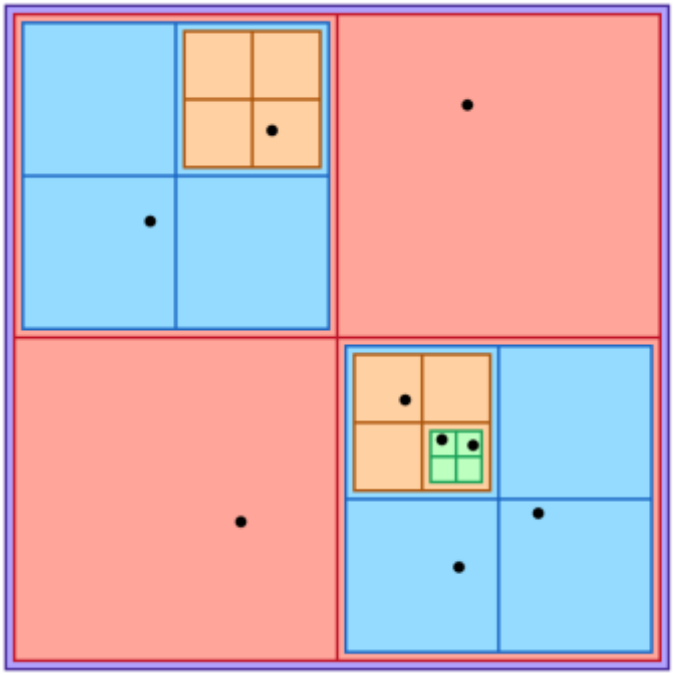
VARIANTE DU QUADTREE : L'OCTREE

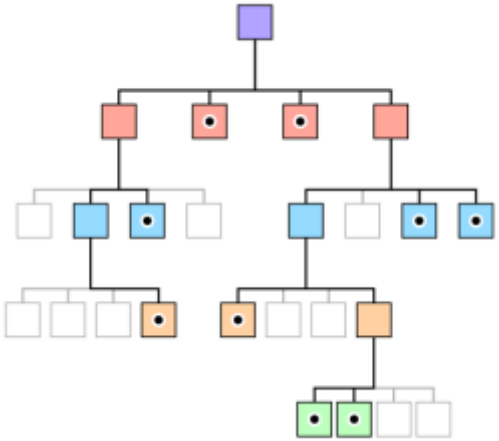
- La même chose, mais en 3D.
- Utile pour des données « très 3D » : en données géo, les données sont souvent étalées en XY mais peu en Z.



POURQUOI PARLER D' «ARBRE » ?

- « Arbre » référence la façon de représenter les données en machine.
- Le nœud principal, en haut, est appelé « racine », les nœuds terminaux des « feuilles ».



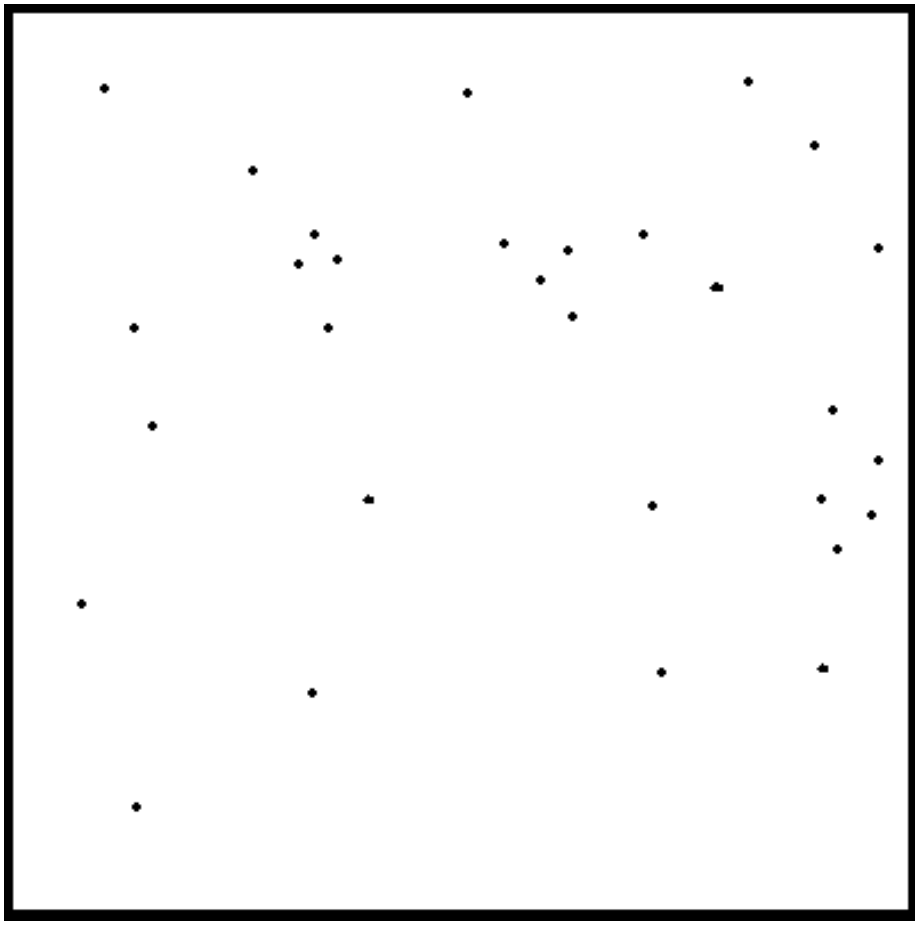


ARBRE K-D

- Les arbres k-d sont des arbres binaires, dans lesquels **chaque nœud contient un point** en dimension k. Chaque nœud non terminal divise l'espace en **deux demi-espaces**. Les points situés dans chacun des deux demi-espaces sont stockés dans les branches gauche et droite du nœud courant. Par exemple, si un nœud donné divise l'espace selon un plan normal à la direction (Ox), tous les points de coordonnée x inférieure à la coordonnée du point associé au nœud seront stockés dans la branche gauche du nœud.
- Pratique pour des recherches de voisinage.
- Arbre binaire.

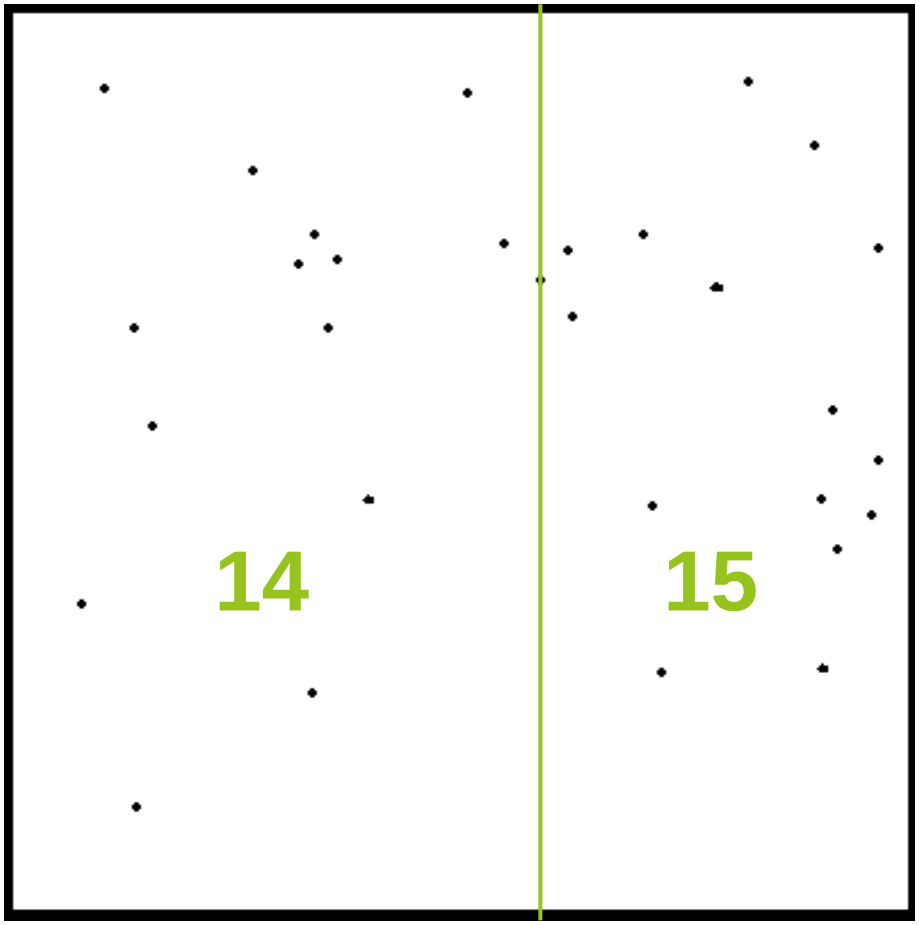
L’ARBRE K-D, CONCRÈTEMENT

Arbre k-d (max 2 points par zone) :



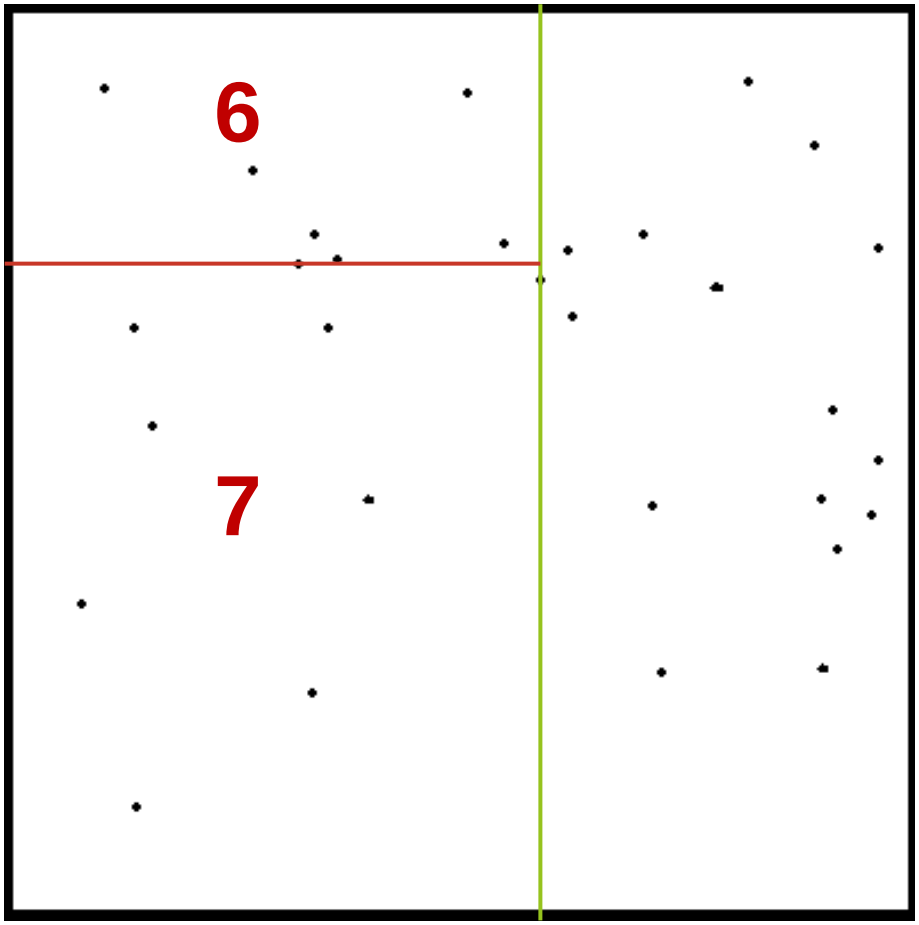
L’ARBRE K-D, CONCRÈTEMENT

Arbre k-d (max 2 points par zone) :



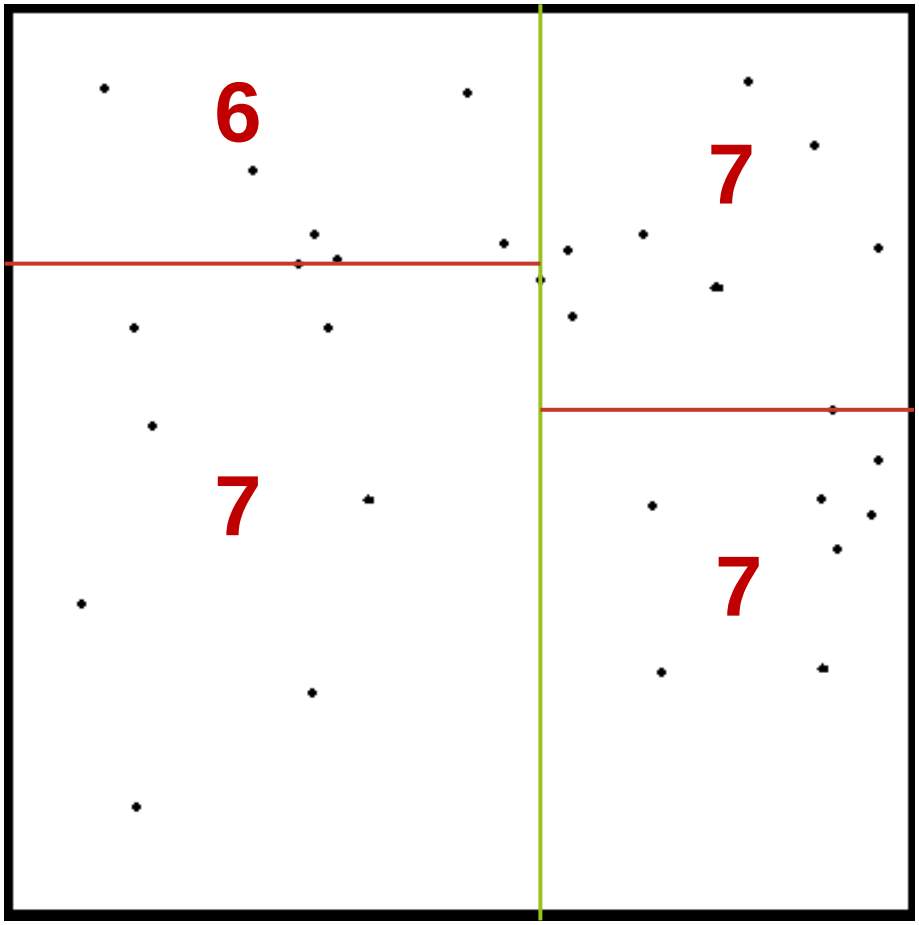
L'ARBRE K-D, CONCRÈTEMENT

Arbre k-d (max 2 points par zone) :



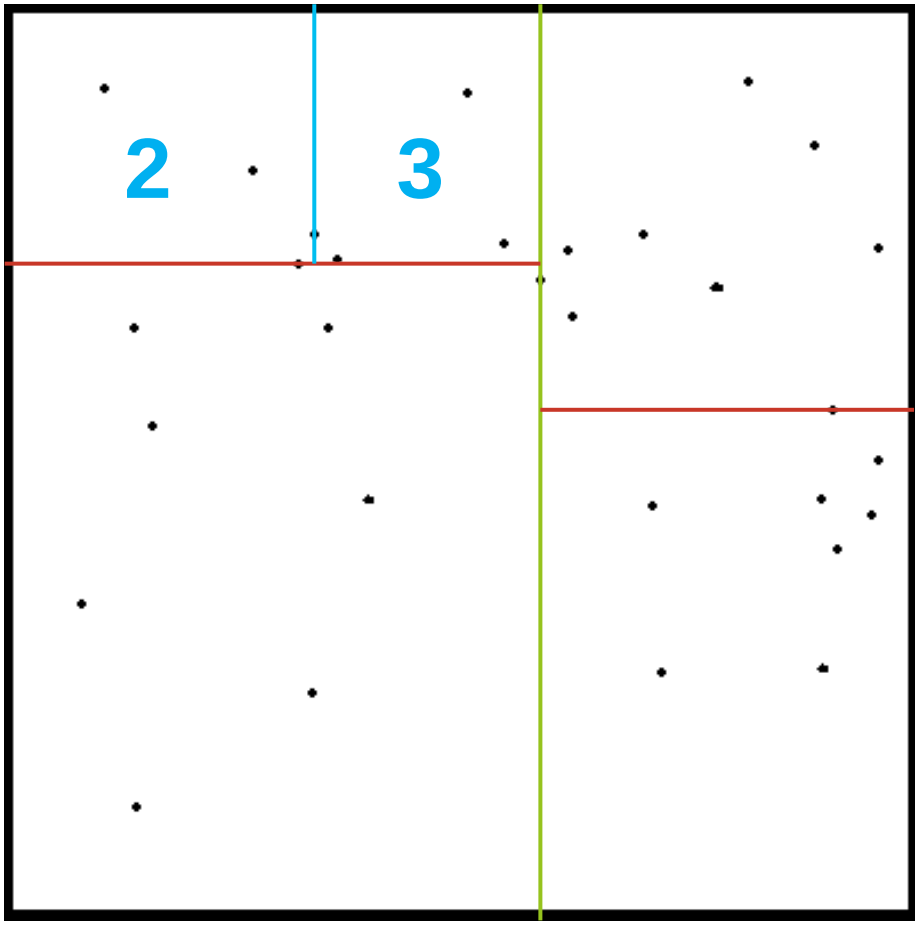
L'ARBRE K-D, CONCRÈTEMENT

Arbre k-d (max 2 points par zone) :



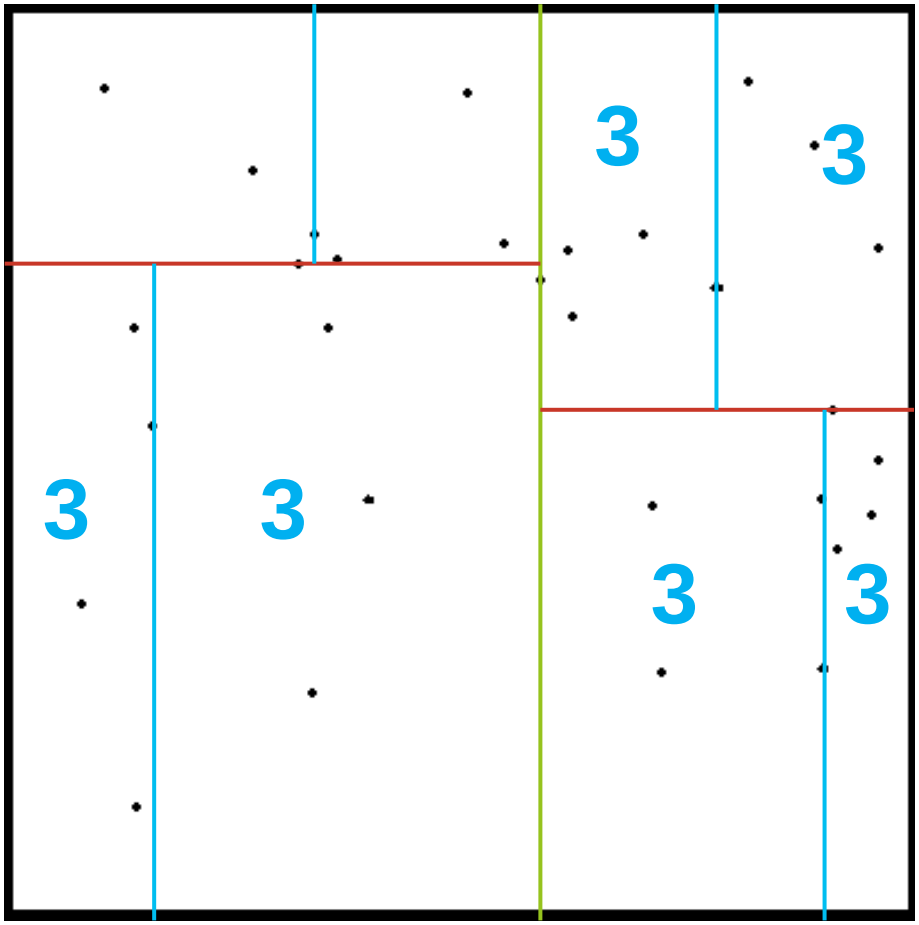
L’ARBRE K-D, CONCRÈTEMENT

Arbre k-d (max 2 points par zone) :



L'ARBRE K-D, CONCRÈTEMENT

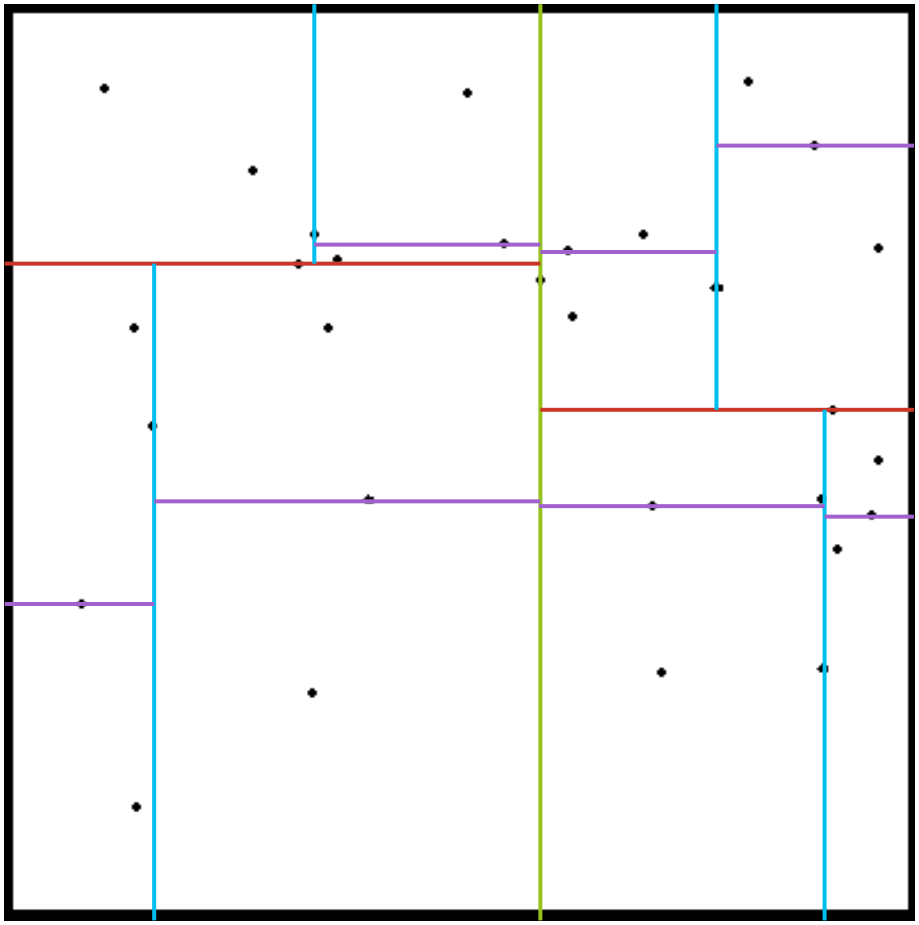
Arbre k-d (max 2 points par zone) :



L'ARBRE K-D, CONCRÈTEMENT

Arbre k-d (max 2 points par zone) :

Fin du découpage presque au même moment pour toutes les zones : graphe équilibré, même performance quelque soit le point recherché.

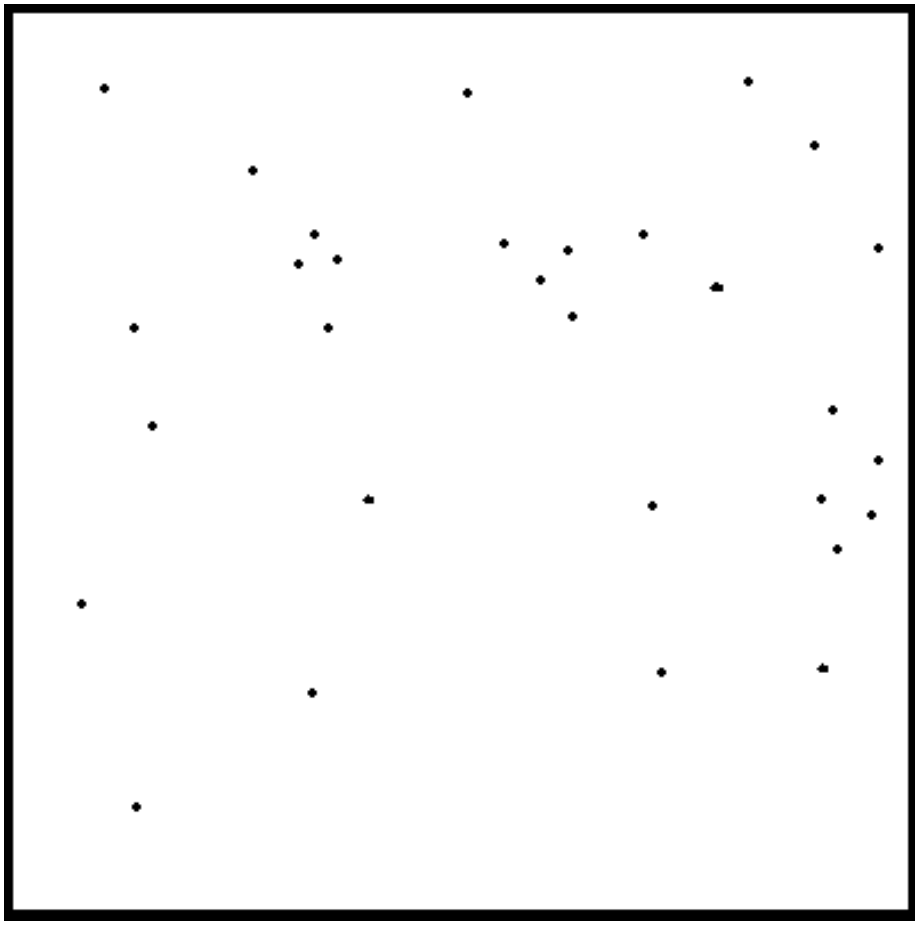


R-ARBRE

- L'idée principale derrière la structure de donnée est de représenter les objets proches **par leur rectangle englobant** au niveau immédiatement supérieur de l'arbre (le "R" de R-arbre est l'initiale de "rectangle"). En d'autres termes, dans un nœud donné de de l'arbre sont stockés les rectangles englobants de chaque sous-arbre dont il est le parent. Sa construction est très basé sur l'**heuristique**.
- Ainsi, lorsqu'on recherche une information spatiale, il suffit de vérifier pour chaque branche si la position recherchée intersecte le rectangle correspondant, et d'ignorer les branches associées à un rectangle pour lequel il n'y a pas d'intersection. Chaque feuille de l'arbre décrit un seul objet, et chaque niveau supérieur décrit un rassemblement d'un nombre toujours plus important d'objets.
- Les R-arbres sont des arbres de recherches équilibrés (toutes les feuilles sont à la même distance de la racine).

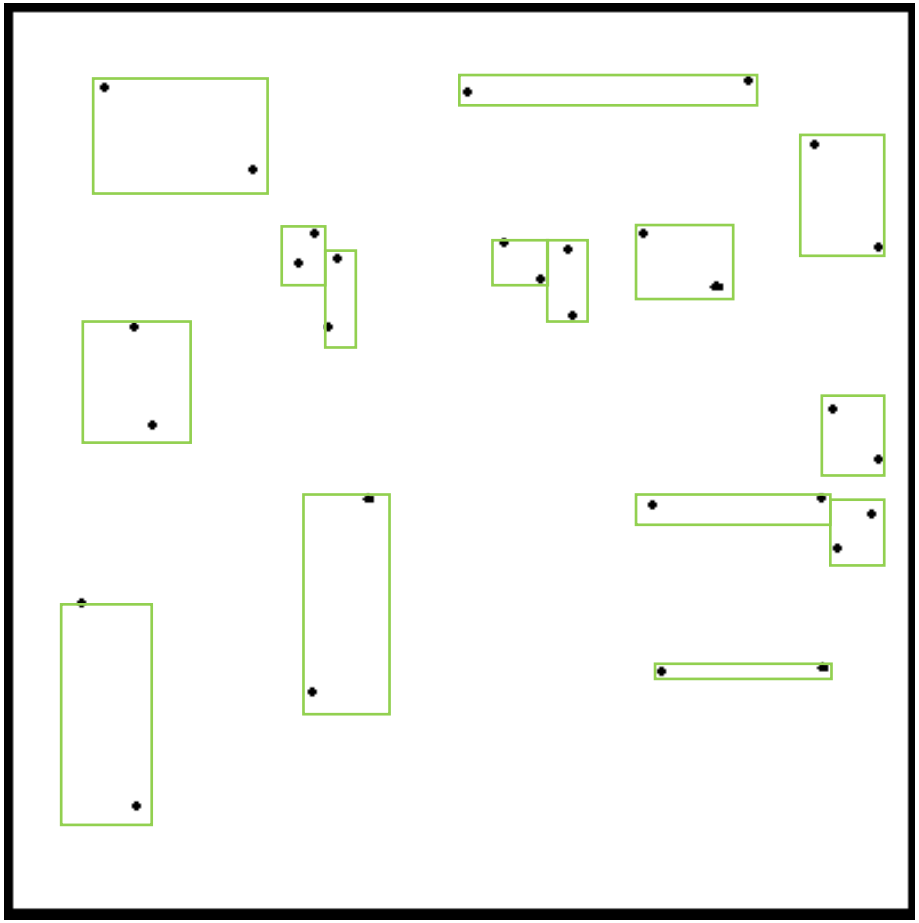
LE R-ARBRE, CONCRÈTEMENT

R-arbre (max 2 points par zone) :



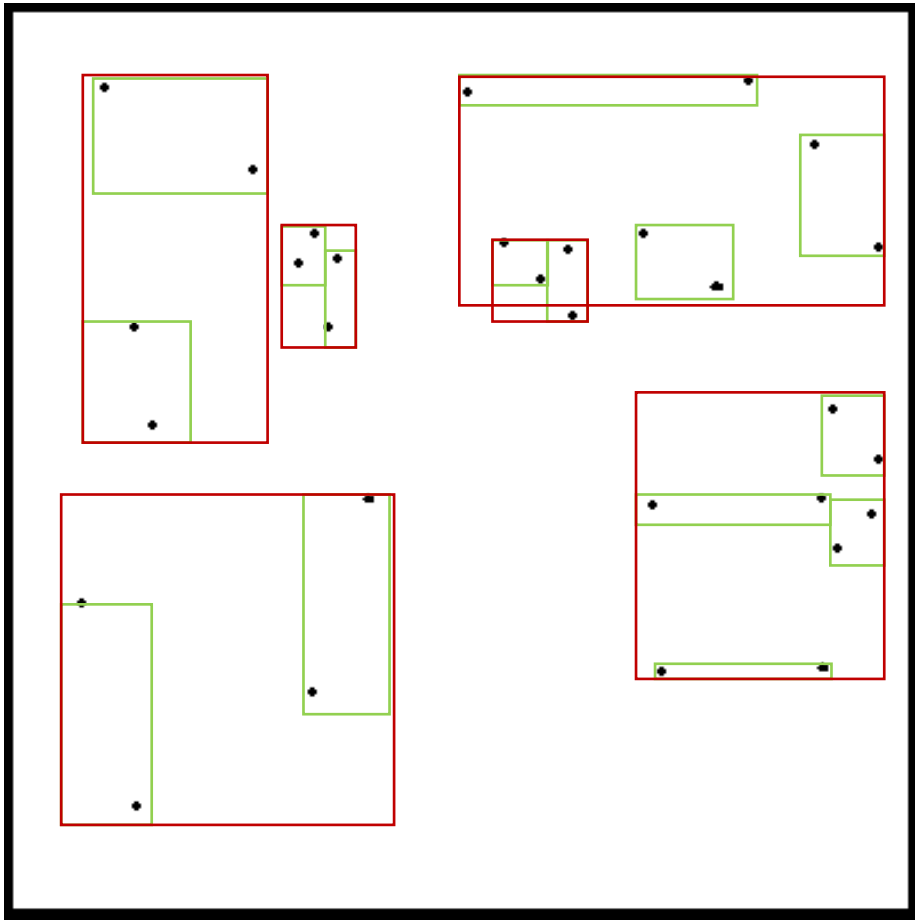
LE R-ARBRE, CONCRÈTEMENT

Arbre k-d (max 2 points par zone) :



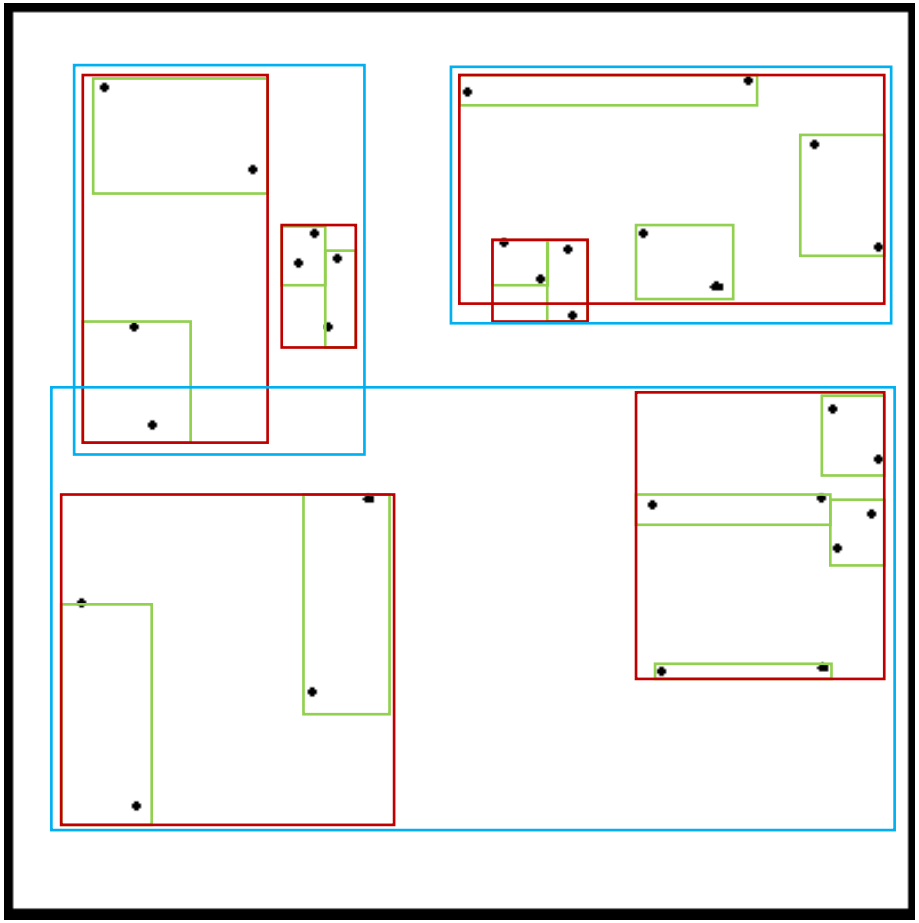
LE R-ARBRE, CONCRÈTEMENT

Arbre k-d (max 2 points par zone) :



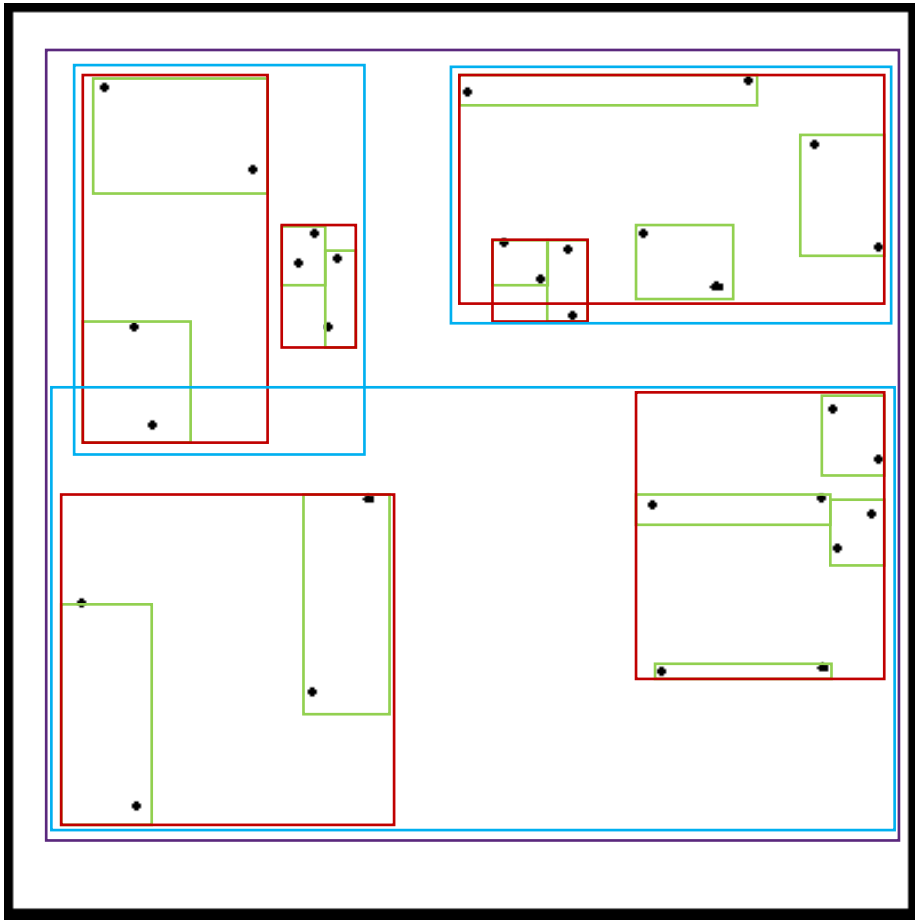
LE R-ARBRE, CONCRÈTEMENT

Arbre k-d (max 2 points par zone) :



LE R-ARBRE, CONCRÈTEMENT

Arbre k-d (max 2 points par zone) :

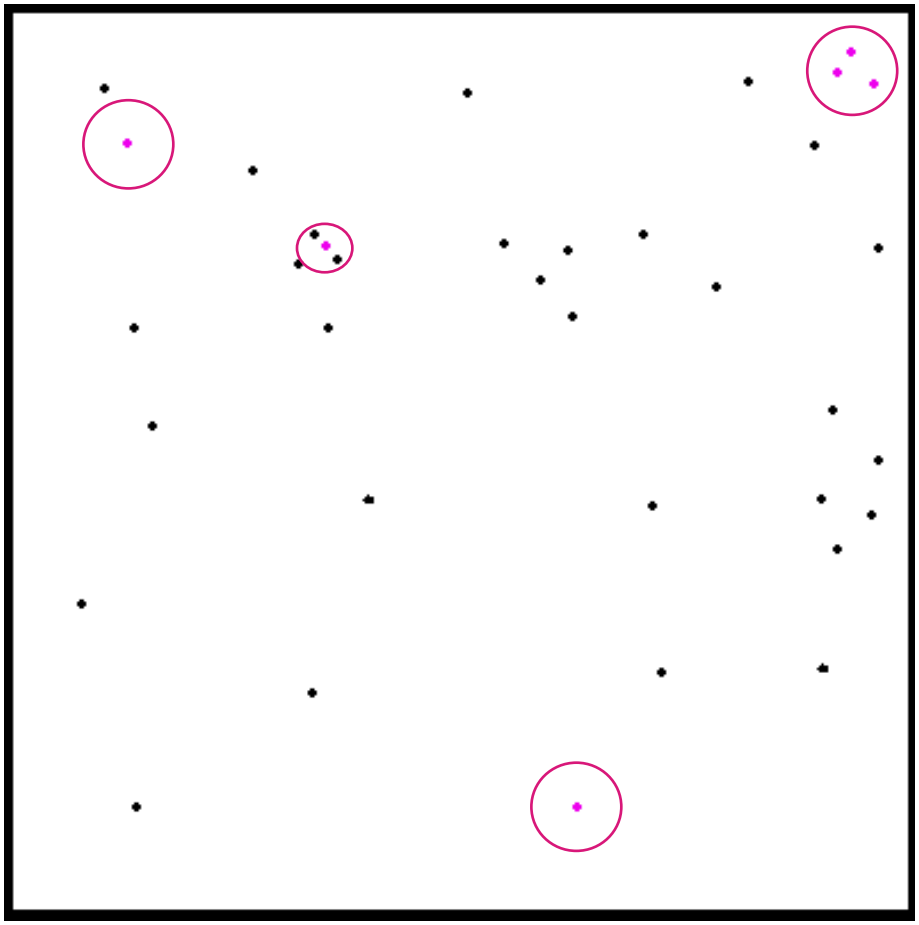


TRAVAUX PRATIQUES : MISES À JOUR D'ARBRES

- Différents arbres réagiront de différentes façons à l'ajout d'objets.
- Savoir si les données seront amenées à être modifiées doit influer sur le type d'arbre choisi.

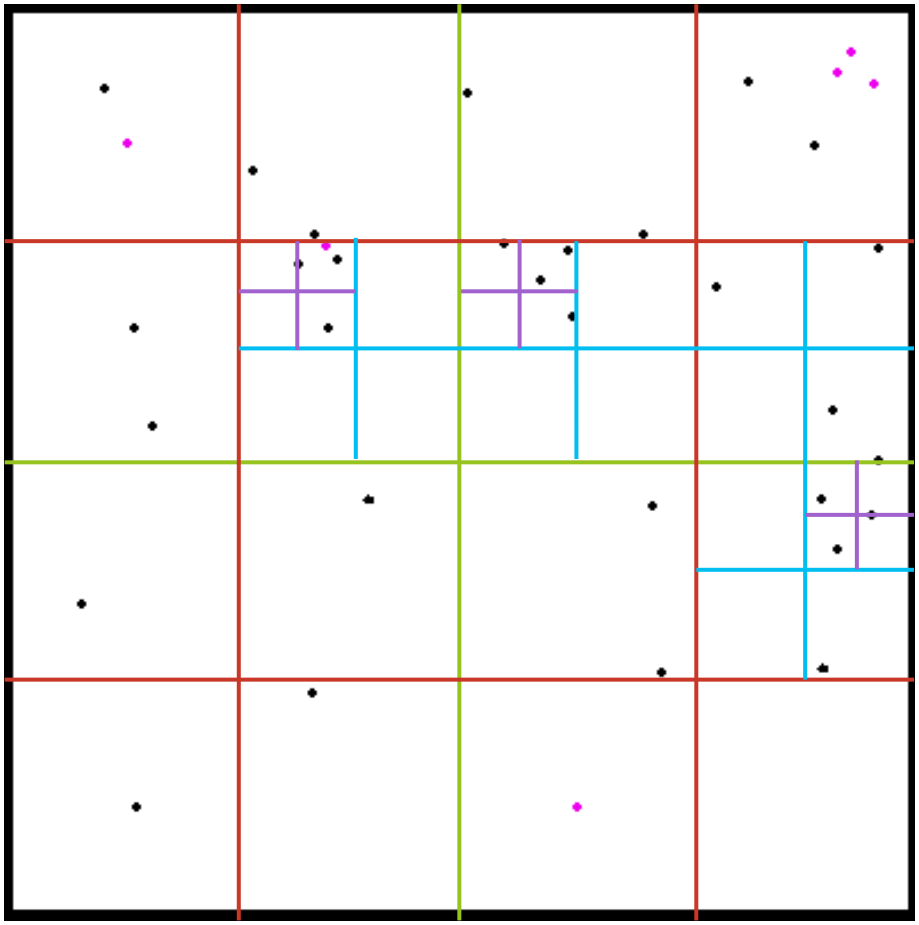
RAJOUT DE POINTS : LE PROBLÈME DES MÀJ

Nouvelles données : 6 nouveaux points



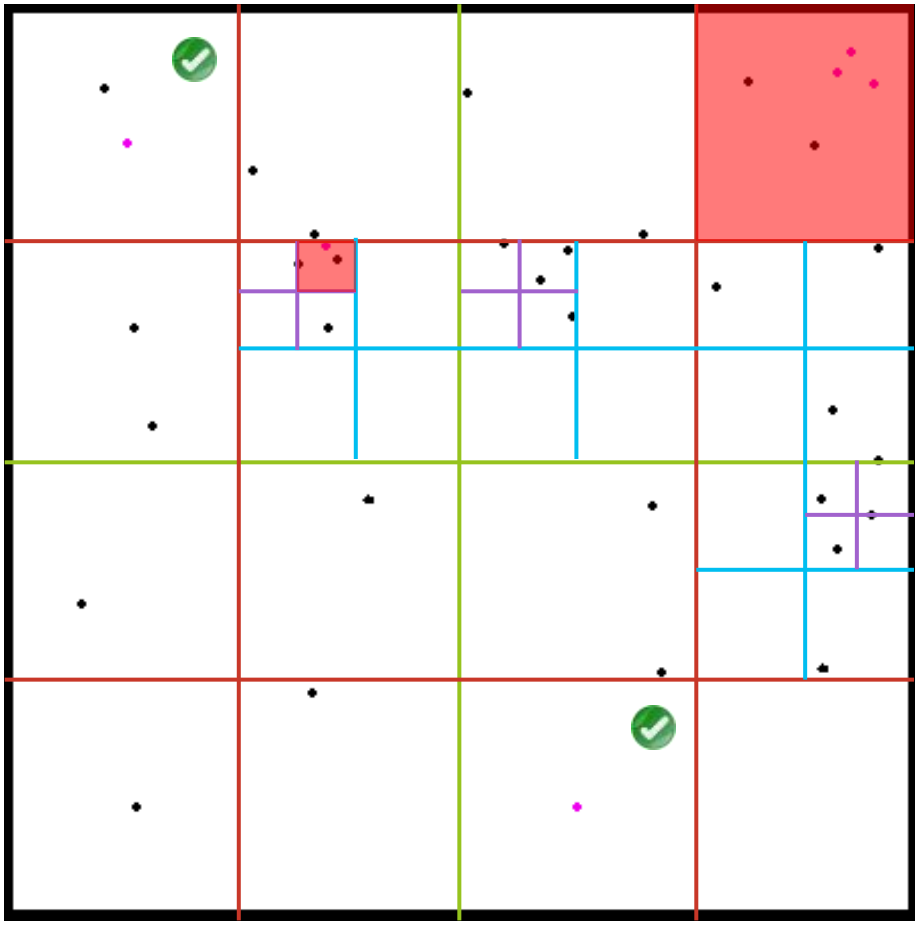
MISE À JOUR DU QUADTREE

Le quadtree adaptatif (max 2 points par zone) :



MISE À JOUR DU QUADTREE

Le quadtree adaptatif (max 2 points par zone) :

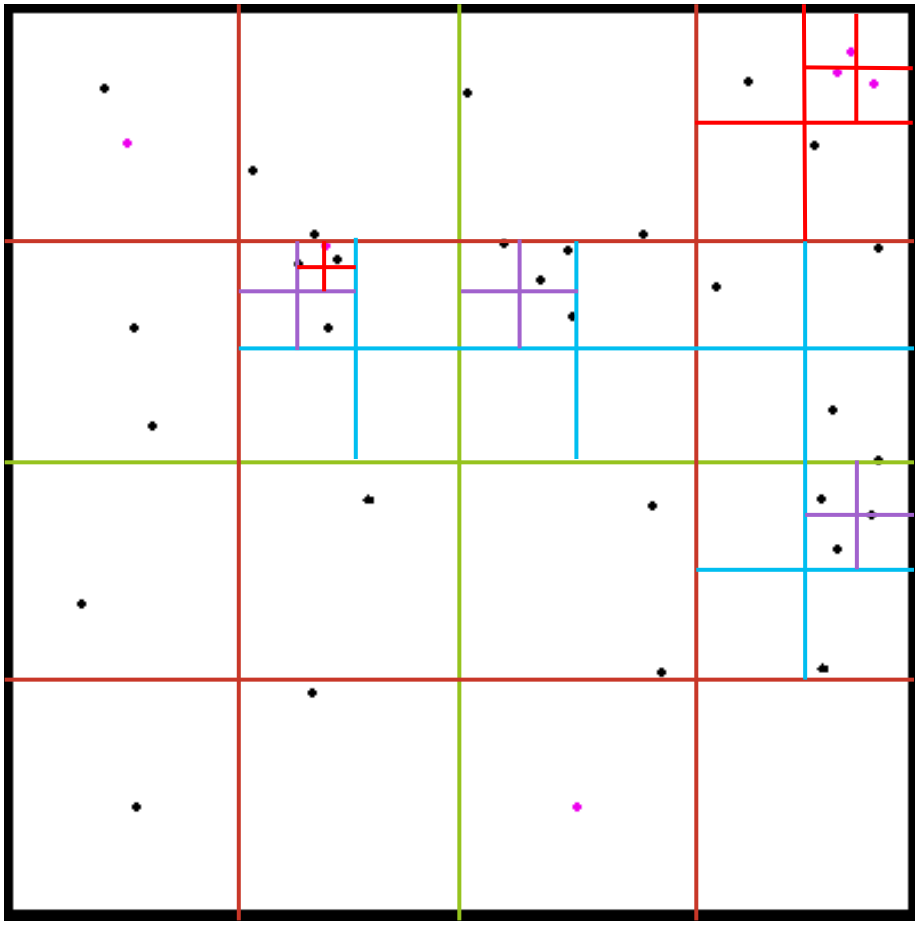


MISE À JOUR DU QUADTREE

Le quadtree adaptatif (max 2 points par zone) :

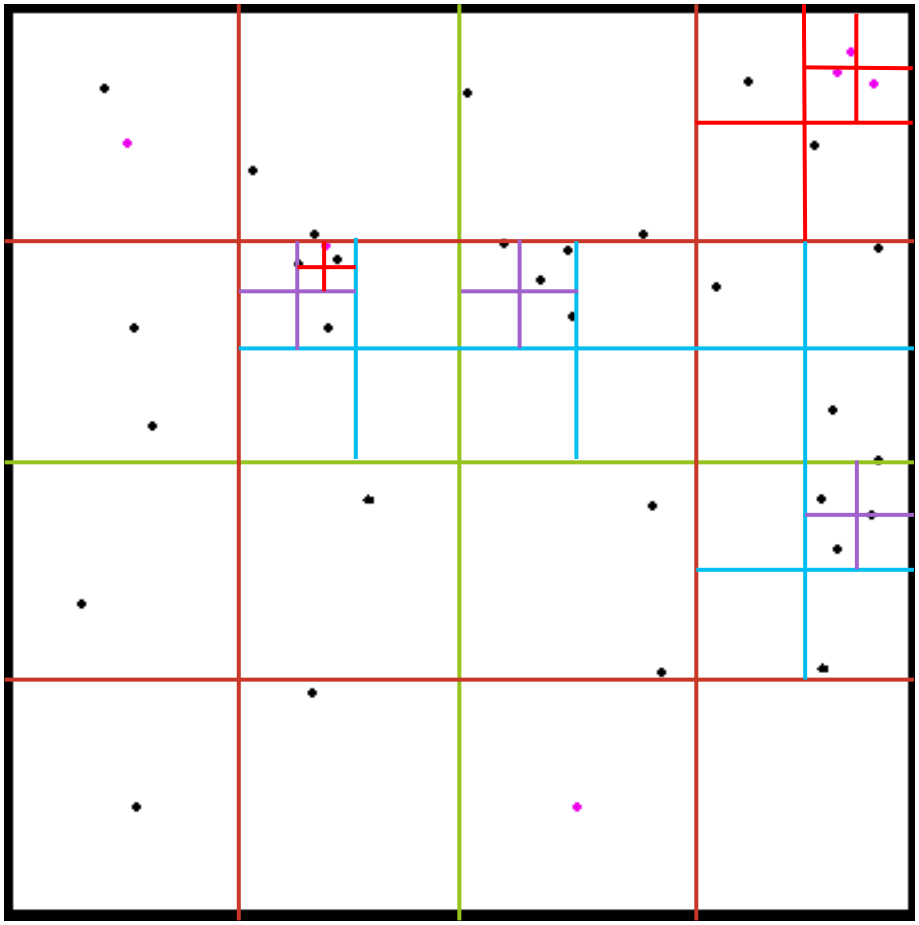
Facile, non ? Aucun découpage à modifier, juste des ajouts à faire.

Pas de changements majeurs dans l'arborescence.



MISE À JOUR DU QUADTREE

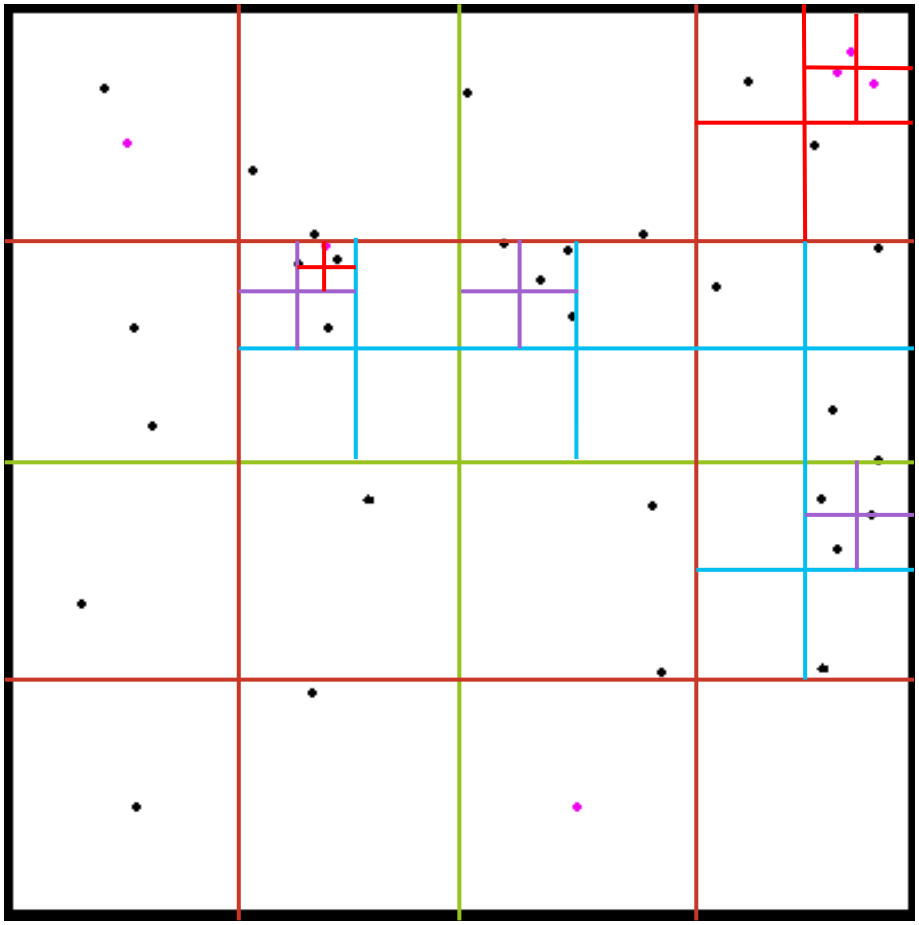
Le quadtree adaptatif (max 2 points par zone) :



MISE À JOUR DU QUADTREE

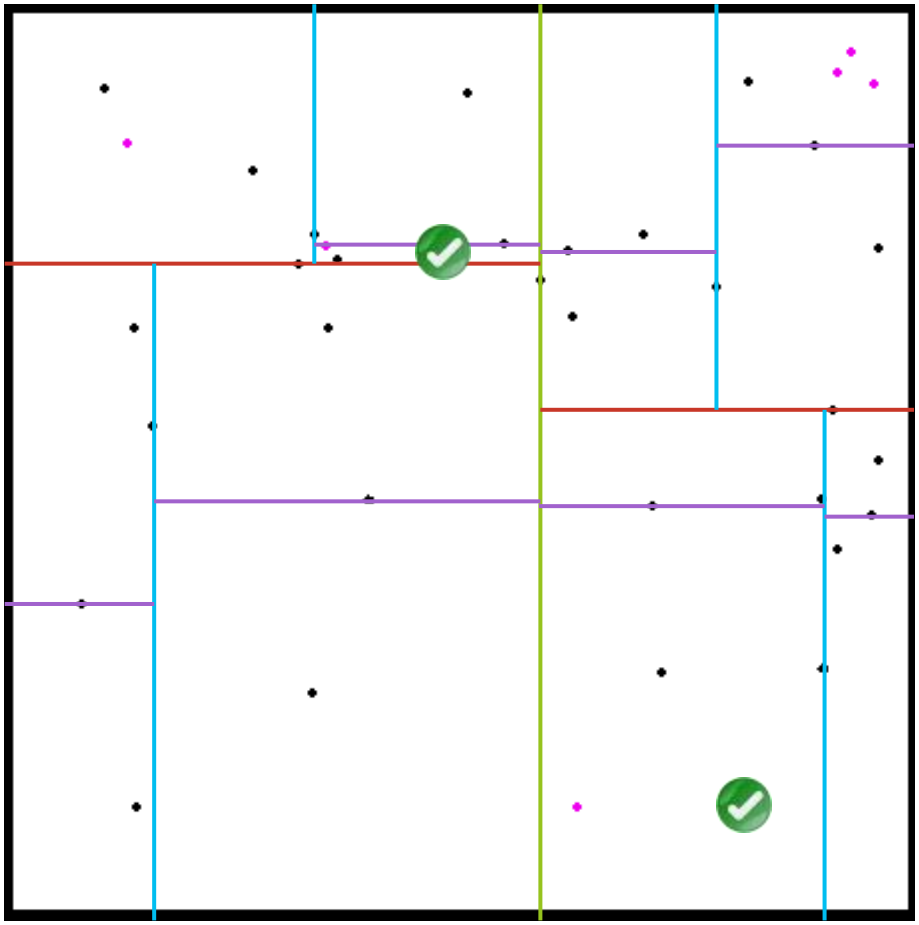
Le quadtree adaptatif (max 2 points par zone) :

Le point est en dehors de l'emprise des données initiales. Il faut **tout** refaire.



MISE À JOUR DE L'ARBRE K-D

Arbre k-d (max 2 points par zone) :

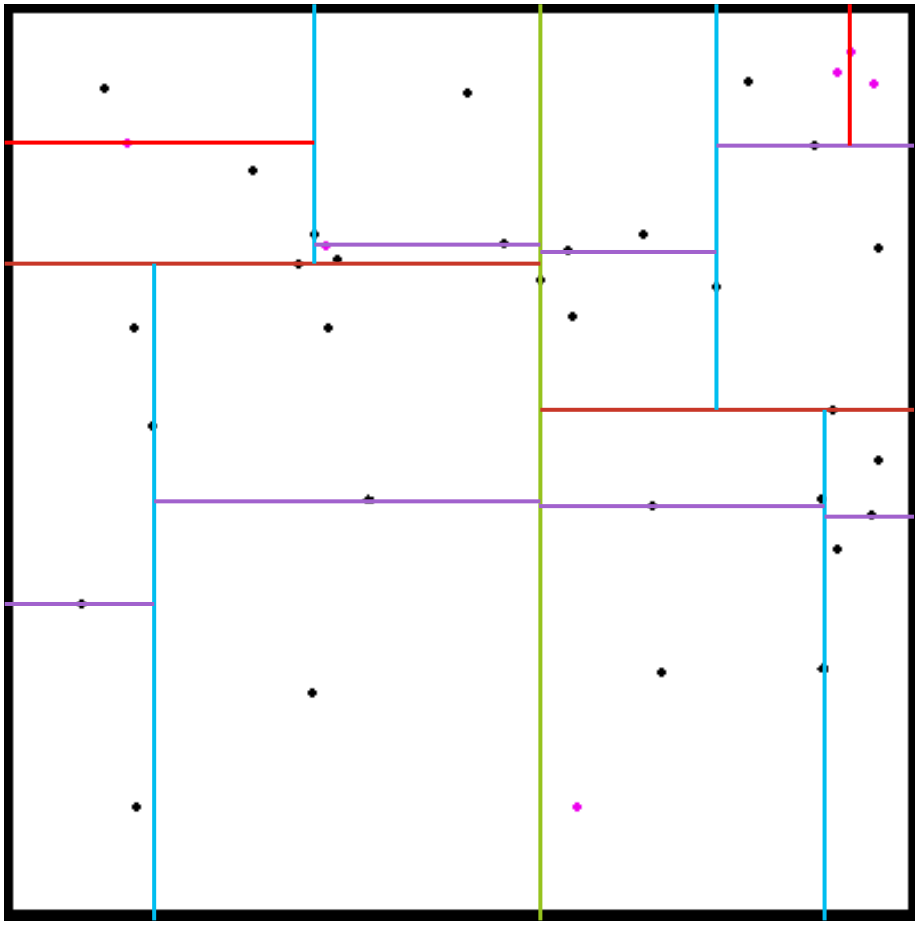


MISE À JOUR DE L'ARBRE K-D

Arbre k-d (max 2 points par zone) :

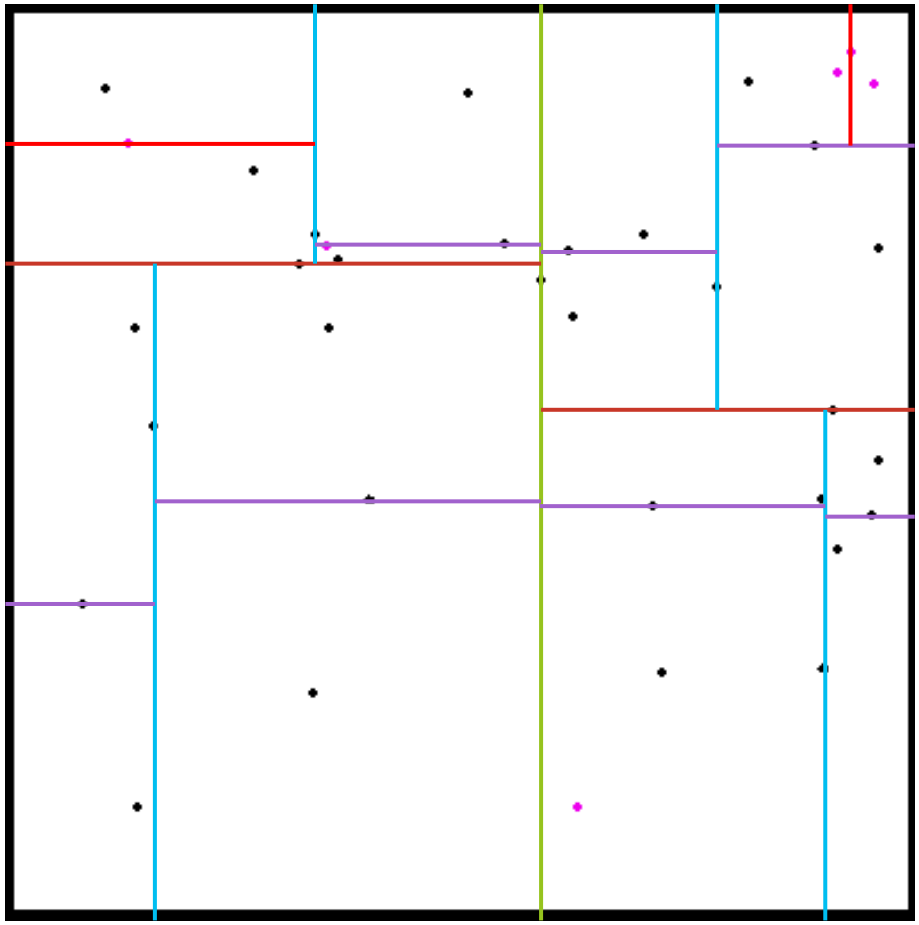
On a réussi à intégrer les points, mais l'arbre n'est plus optimal.

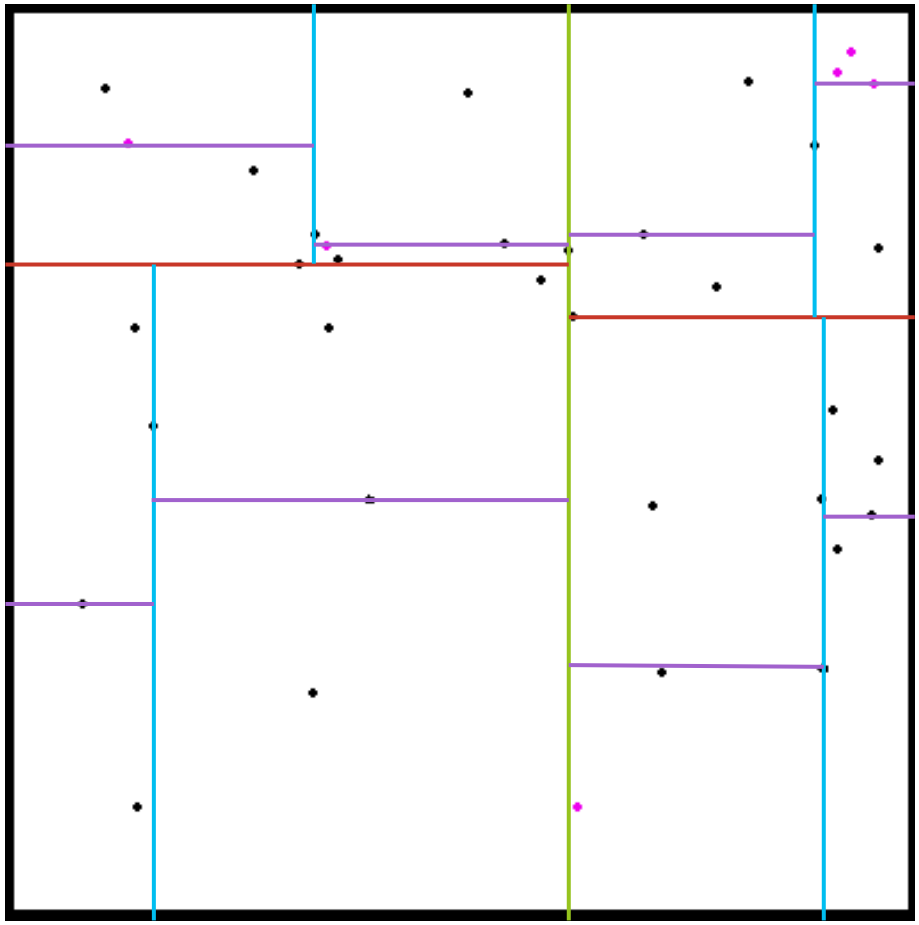
Pas de problème majeur avec des points en dehors de l'emprise initiale.



MISE À JOUR DE L'ARBRE K-D

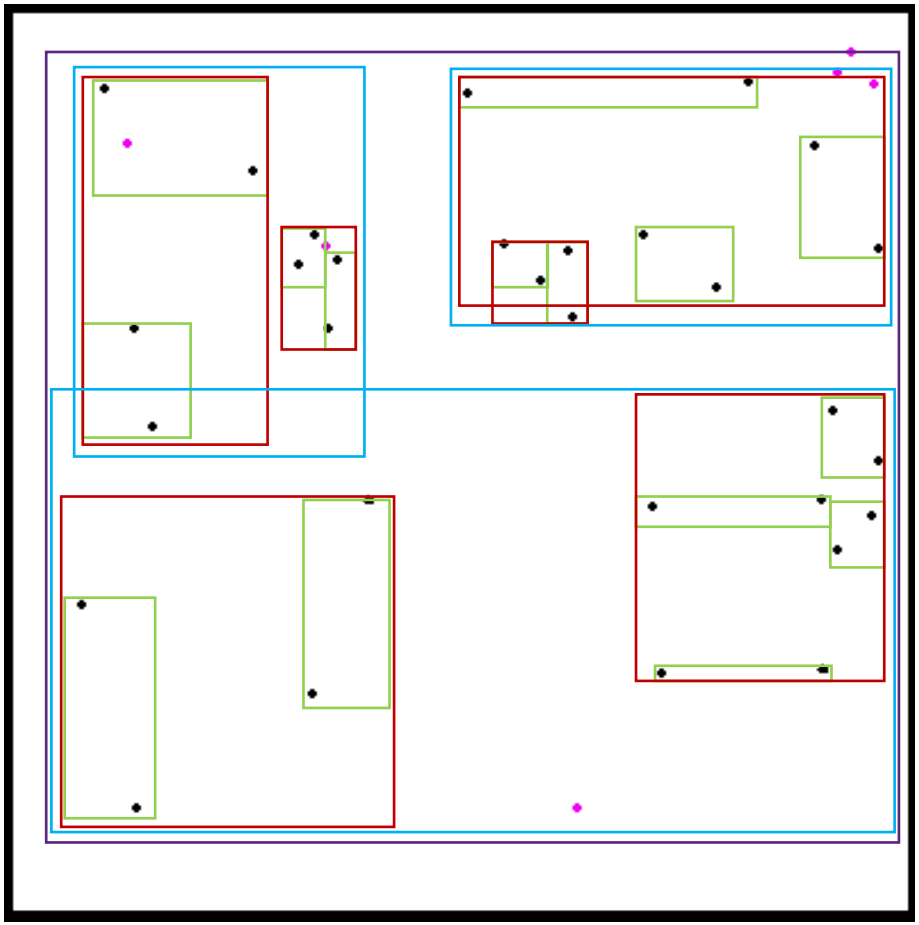
Arbre mis à jour vs Arbre reconstruit de zéro





MISE À JOUR DU R-ARBRE

R-arbre (max 2 points par zone) :

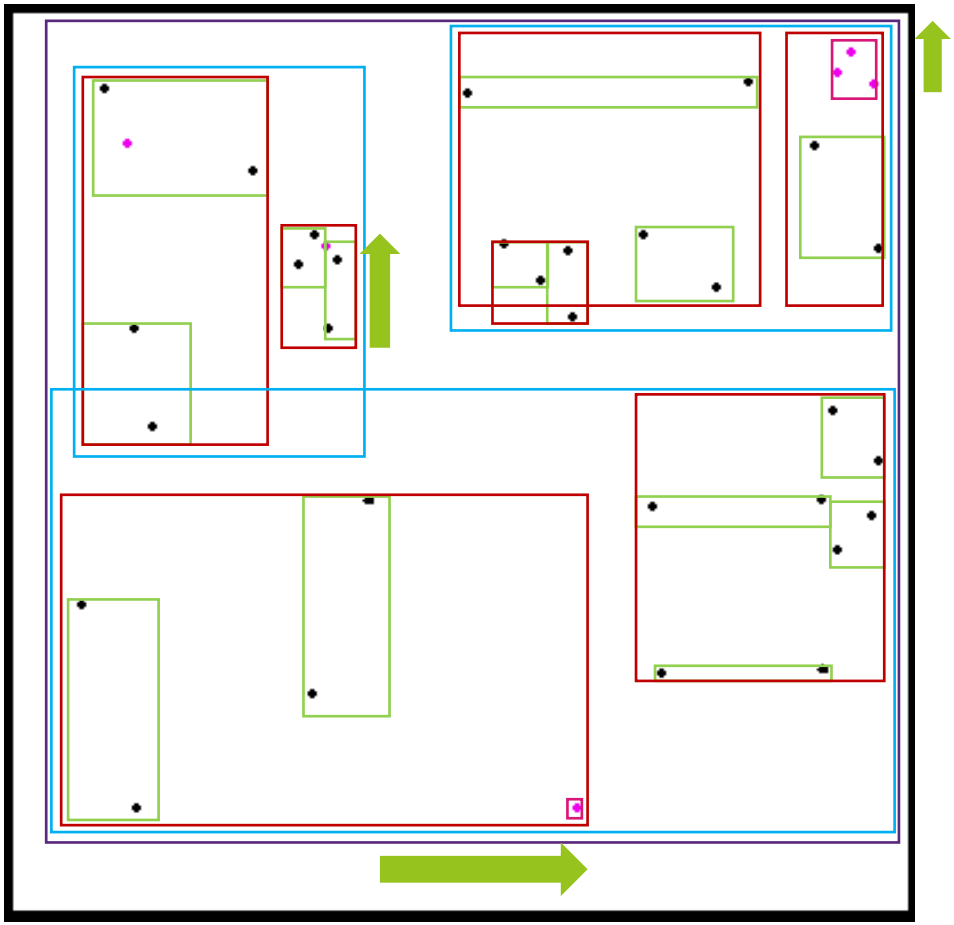


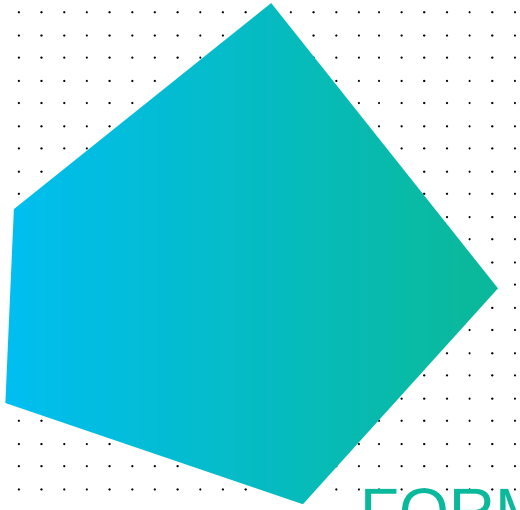
MISE À JOUR DU R-ARBRE

R-arbre (max 2 points par zone) :

La contrainte de 2 points max n'est plus respectée.

On a rajouté les points aux boites auxquelles cela causerait le plus faible agrandissement.





FORMATS 3D EXISTANTS

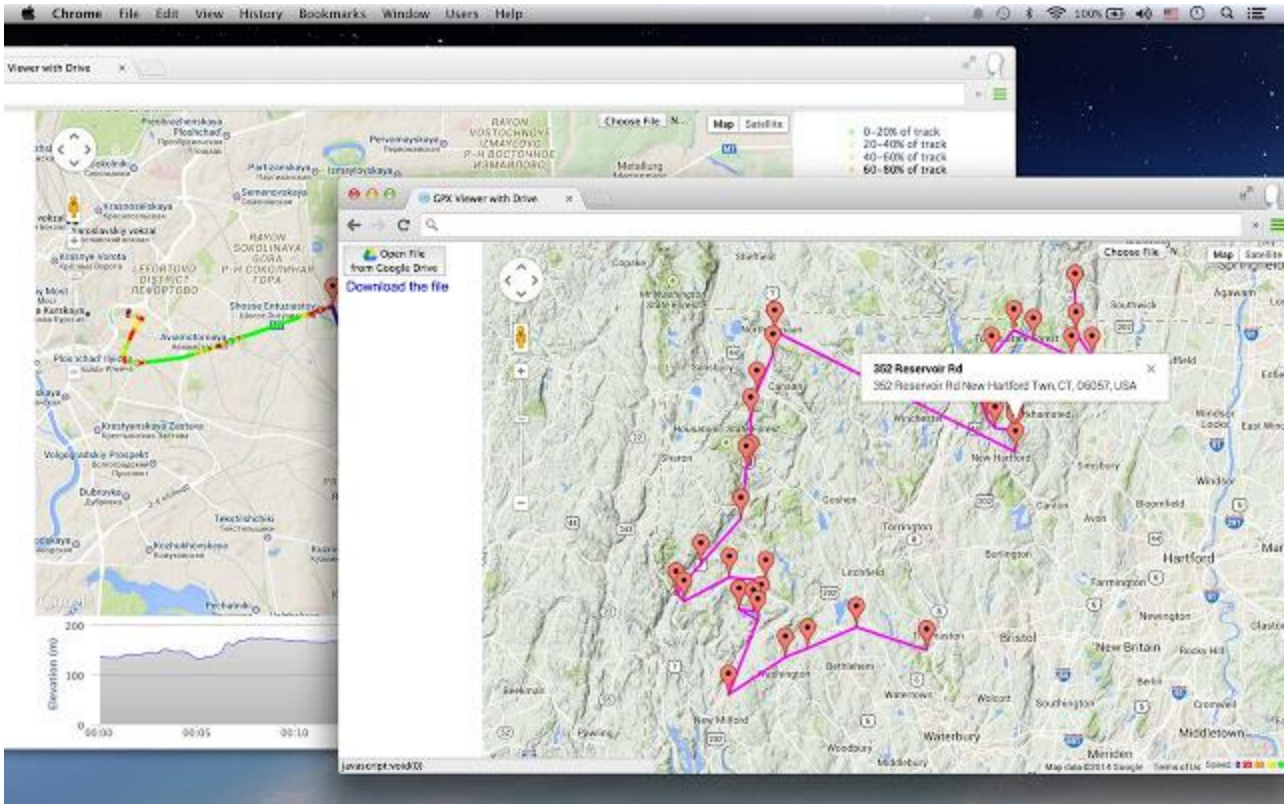
LISTE DE FORMATS OUVERTS

Basés sur XML :

- Collada (COLLaborative Design Activity)
- KML (Keyhole Markup Language), basé sur Collada, célèbre grâce à Google Maps : définit des caractéristiques (description textuelle, images, polygones, marque…) à afficher sur une carte
  - KMZ, du KML zippé
- GML (Geography Markup Language) et tous ses dérivés, dont
  - CityGML
  - InfraGML
  - IndoorGML
  - …

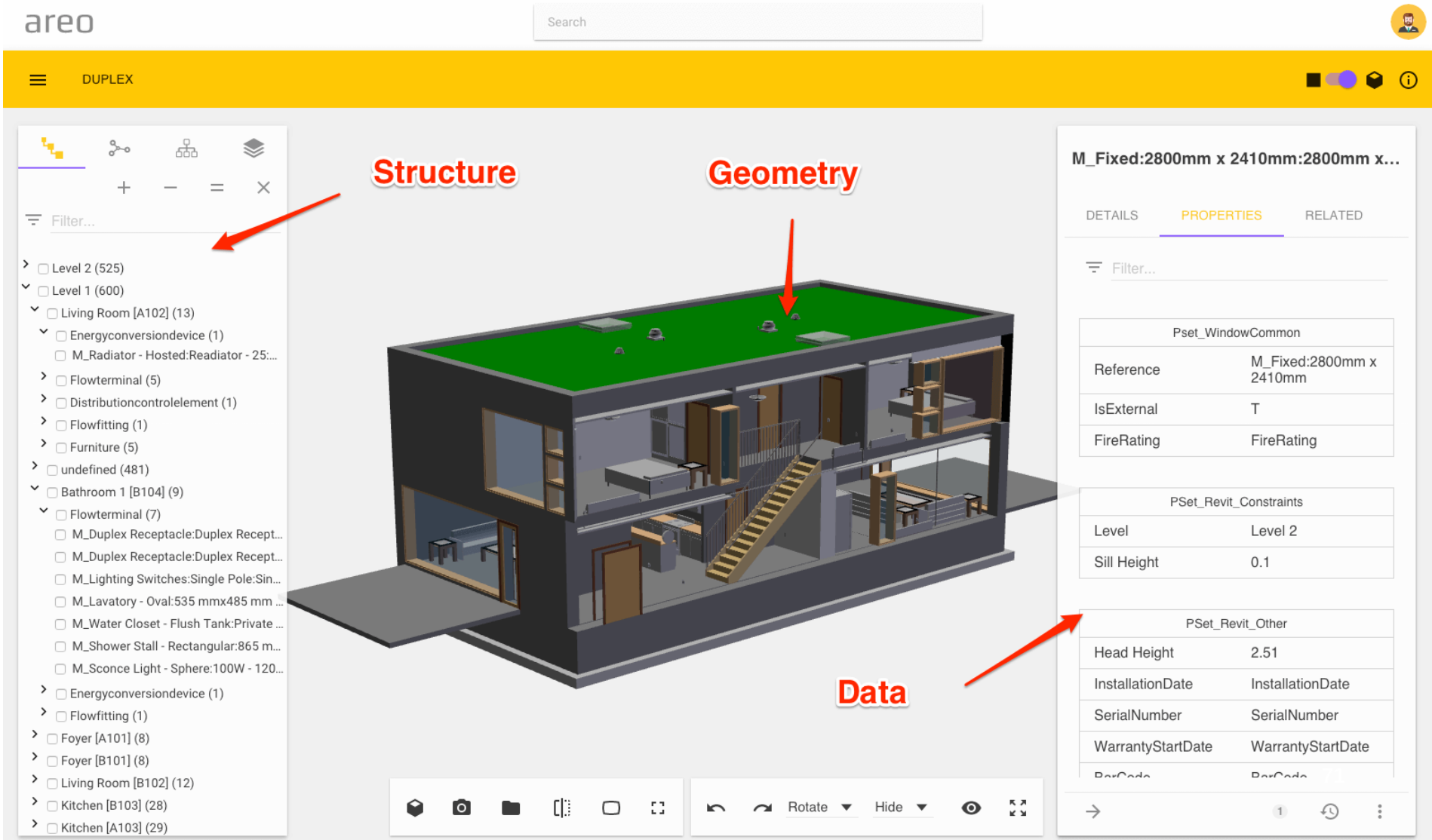
LISTE DE FORMATS OUVERTS

- GPX, pour des traces GPS : suite de points organisés par treks.



LISTE DE FORMATS OUVERTS

- **IFC** (ISO 16739), issu du monde du BIM, l'industrie du bâtiment.



LISTE DE FORMATS OUVERTS

- **Nuages de points :**
  - **E57**
  - **LAS**
  - **PTX**
  - **PTS**
  - **PLY**

Différences : fichiers texte ou binaire, points exprimés en coordonnées relatives (par station) ou absolues…

CITYGML

- Basé sur GML, lui-même basé sur XML
- Modèle de données avec bâtiments, voirie, cours d'eau, végétation…
- Format d'échange pour les données 3D urbaines très riche en sémantique
- Gestion des niveaux de détail
- Version 2.0

- Très verbeux, peu performant

CITYGML


Disaster management
Radio network planning
Noise immision mapping
CityGML
for 3d city models
Police simulator
Business development & tourism
Kaufhof
Springfield
Leffers
Münsterplatz
Navigation
Facility management
Urban planning
Architecture

CITYGML : EXEMPLES

- https://www.businesslocationcenter.de/berlin3d-downloadportal/#/export

- https://www.europeandataportal.eu/data/en/dataset/3d-model-den-haag

- https://kartta.hel.fi/3d/

CITYGML : ÉTAT DE LA NORMALISATION

Version 1.0 – OGC 08-007r1

- Développé à l'origine par SIG3D (Allemagne)
- Adopté par l'OGC en aout 2008

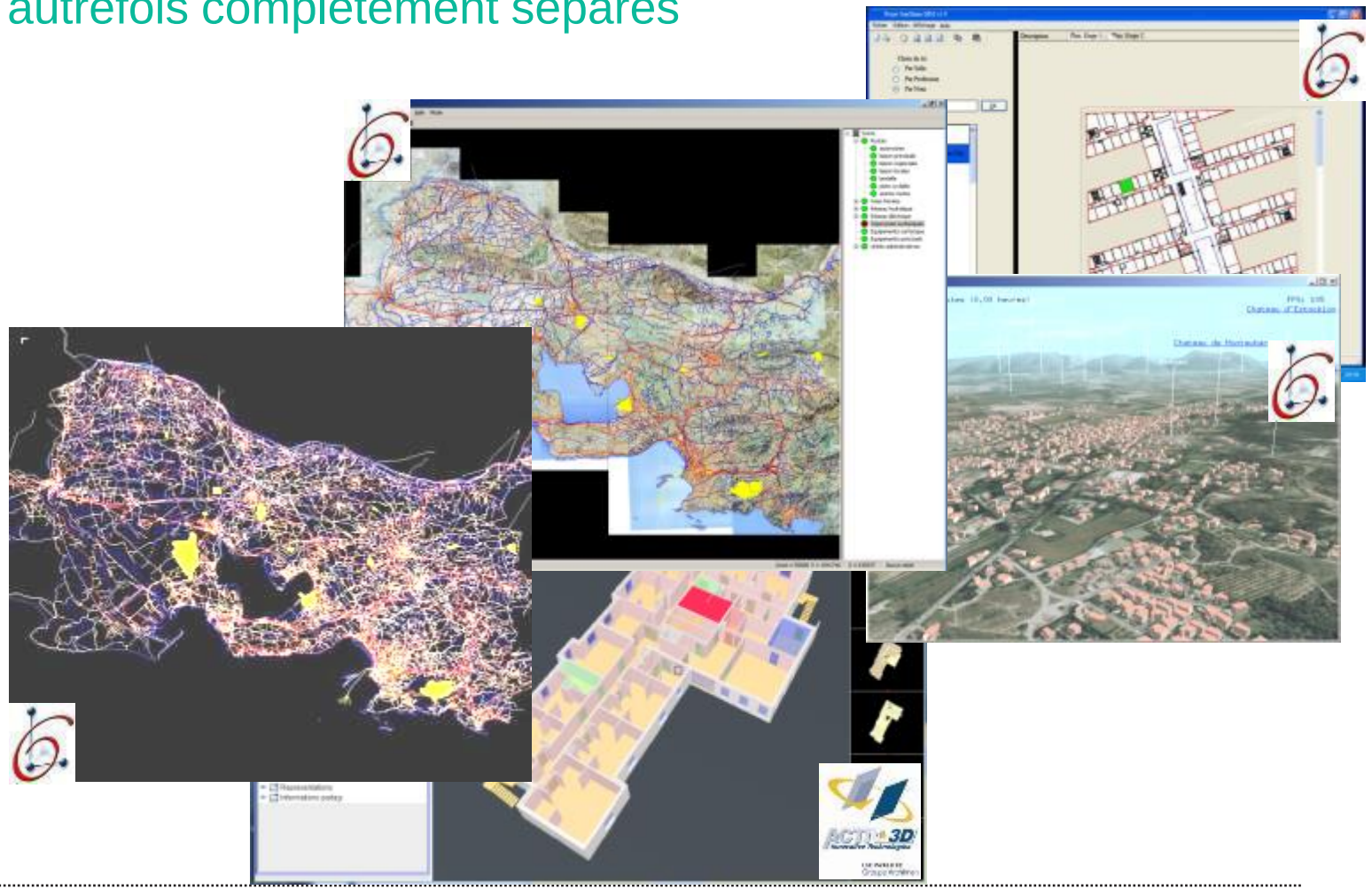
Version 2.0 – OGC 12-019

- Adopté par l'OGC en mars 2012
- Principales évolutions :
  - **Usage étendu du LoD 0 (emprises des bâtiments, par ex.)**
  - **Ajout des ponts et tunnels**
  - **Base pour INSPIRE Building**

http://www.opengeospatial.org/standards/citygml

VILLES VIRTUELLES ET MODÈLES URBAINS

- Les villes virtuelles se retrouvent à l'interface de plusieurs mondes qui étaient autrefois complètement séparés
  - **BIM**
  - **CAO**
  - **SIG**



CITYGML : CARACTÉRISTIQUES CLÉS

Modélisation et représentation 3D d'un environnement urbain :

- bâtiments, ponts, tunnels…
- objets et structures urbaines communes : voirie, végétation, …
- géométrie : surface externe + interne (LoD4)
- topologie (faible en LoD0, riche en LoD4) : partage de primitives ou décomposition en primitives
- Sémantique : classes d'objets et attributs
- gestion des niveaux de détails (LoD)
- gestion des textures
- capacité d'extensions : Application Domain Extension (ADE)

LE STANDARD CITYGML 2.0 (OGC 12-019)

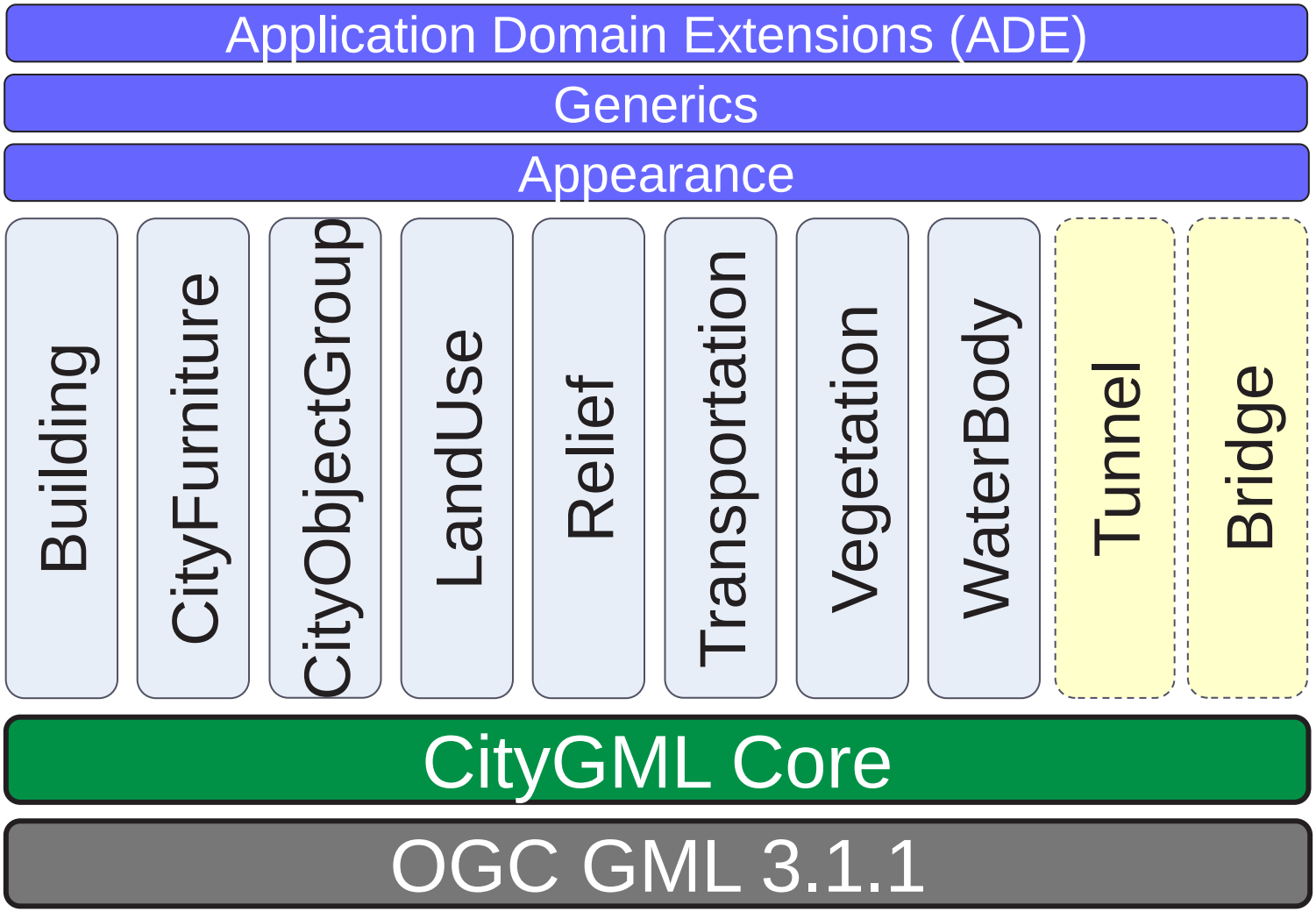
Quasiment 350 pages !

- Mélange d'exigences et de texte descriptif/recommandations

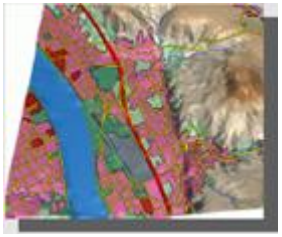
Contenu principal

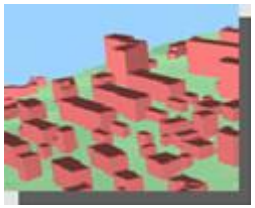
- Chapitres 5 et 6 : description des caractéristiques de CityGML
- Chapitre 7 : description de la modularisation
- Chapitre 8 : modèle spatial : Description du profil d'ISO 19107 utilisé
- Chapitre 9 : modèle d'apparence (textures)
- Chapitre 10 : modèle thématique : Modélisation des objets urbains par module
- Annexe A (normative) : schémas XML
- Annexe B (normative) : suite de tests abstraits pour les instances CityGML
- Annexe C (informative) : listes de codes
- Annexe D (informative) : liste des primitives géométriques GML utilisées
- Annexe E (informative) : niveau de détail des objets et propriétés
- Annexe F (informative) : évolutions de CityGML 2.0
- Annexe G (informative) : jeu de données exemple
- Annexes H et I (informative) : 2 exemples d'ADE : Noise et Ubiquitous Network Robots

CITYGML : MODÉLISATION THÉMATIQUE ET MODULAIRE



NIVEAUX DE DÉTAIL (LEVEL OF DETAIL – LOD)





LoD 0 – modèle régional

- Modèle 2,5D – MNT

LoD 1 – modèle urbain

- Modèle à toits plats

LoD 2 – modèle urbain

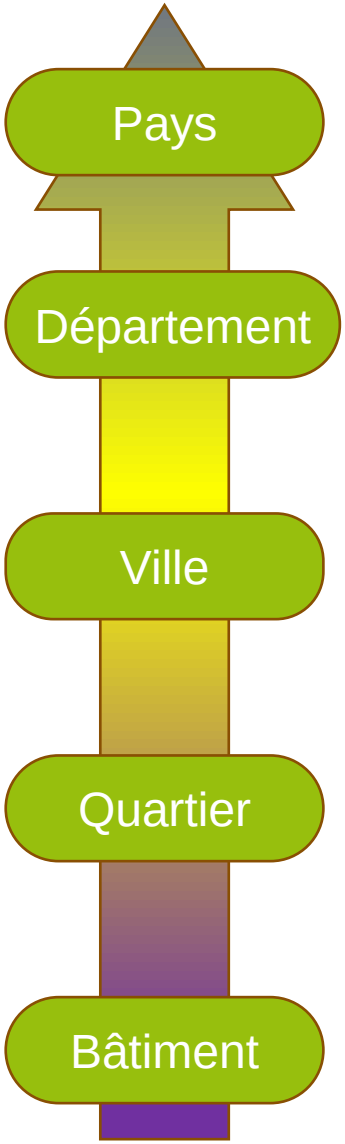
- Textures, structures de toit

LoD 3 – modèle urbain

- Modèle architectural détaillé

LoD 4 – modèle d'intérieur

- Maquette navigable



LES NIVEAUX DE DÉTAIL DANS CITYGML

LOD 0 : modélisation 2,5D

- Occupation du sol, MNT, emprise des bâtiments

LOD 1 : modélisation volumique simple

- Pas de discrimination sémantique des surfaces composant l'objet
- Pas d'objets sémantiques précis

LOD 2

- Structures de toits plus précises
- Différenciation des surfaces sémantiques composant l'objet (mur, toit…)
- Décomposition sémantique, lien vers des objets structurants

LOD3

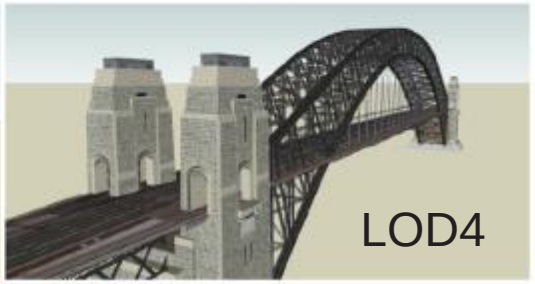
- Modélisation architecturale extérieure dont ouvertures : portes et fenêtres

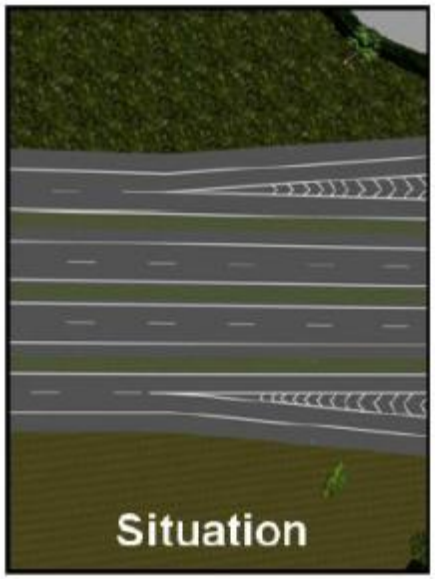
LOD4

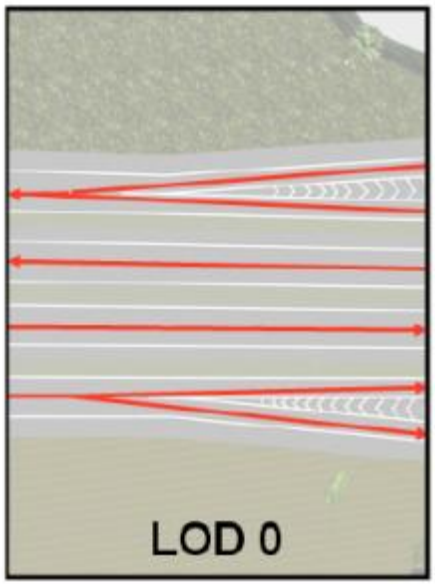
- Modélisation architecturale intérieure

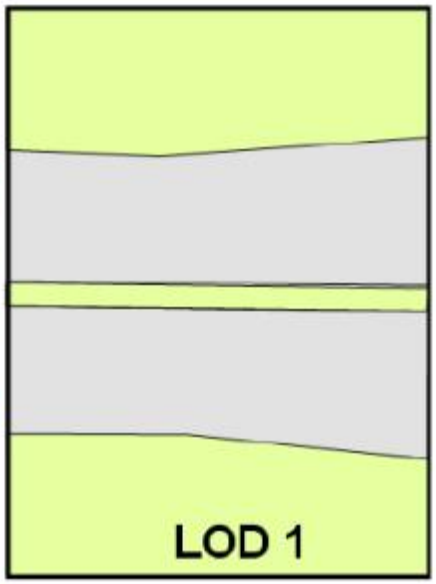
QUELQUES EXEMPLES

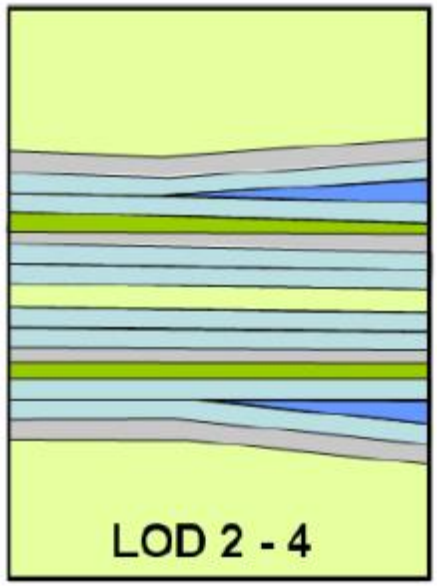
- Ponts
- Routes



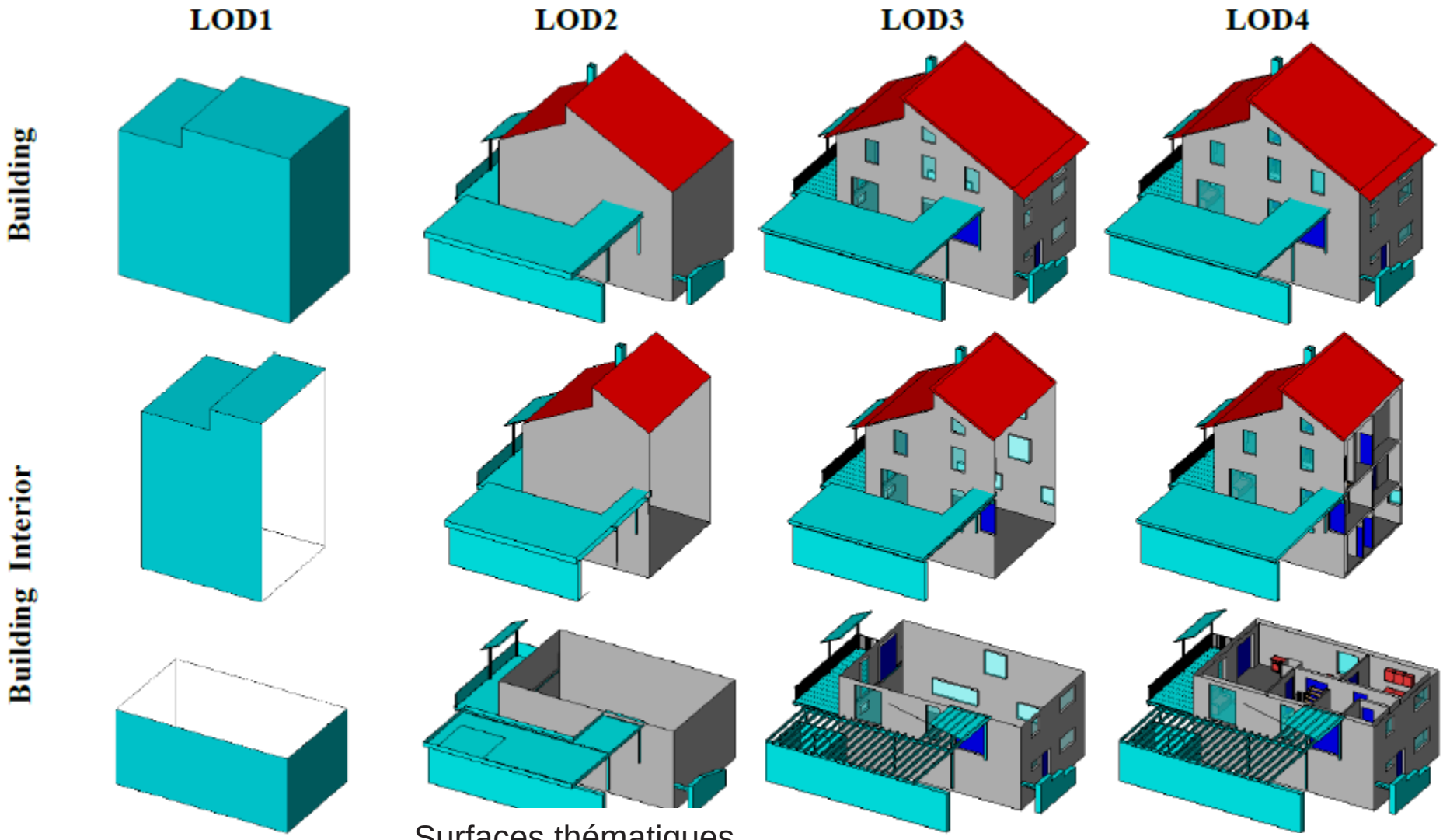






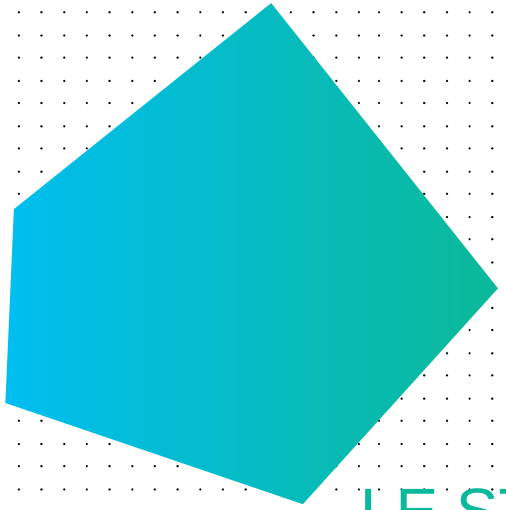


DE LOD1 À LOD4



ET ENFIN

- **I3S**
- **3D Tiles**

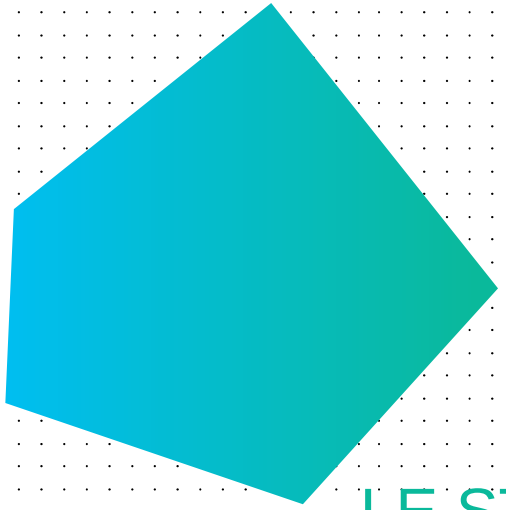


LE STANDARD 3DTILES

Aspects normatifs

STATUT DE 3D TILES

- 3D Tiles *Tuiles tridimensionnelles*
- Standard communautaire OGC
  - Spécification disponible sur le site de l'OGC : http://docs.opengeospatial.org/is/18-053r2/18-053r2.html
  - Publiée le 31 janvier 2019.
- Aussi disponible sur Github : https://github.com/AnalyticalGraphicsInc/3d-tiles
  - Attention, cette version peut être en avance sur le standard approuvé
- Créé par AGI (Analytical Graphics Inc)
  - Spécification 3DTiles (Github)
    - **Ouverte**
  - Outil de préparation et d'hébergement des *caches* 3Dtiles : Cesium Ion
    - **Propriétaire**
  - Visualisateur CesiumJS
    - **Ouvert**
    - **Prise en main aisée**



LE STANDARD 3DTILES

Format

APERÇU DE LA SPÉCIFICATION

- Tuiles (*tiles*) et jeux de tuiles (*tilesets*)…

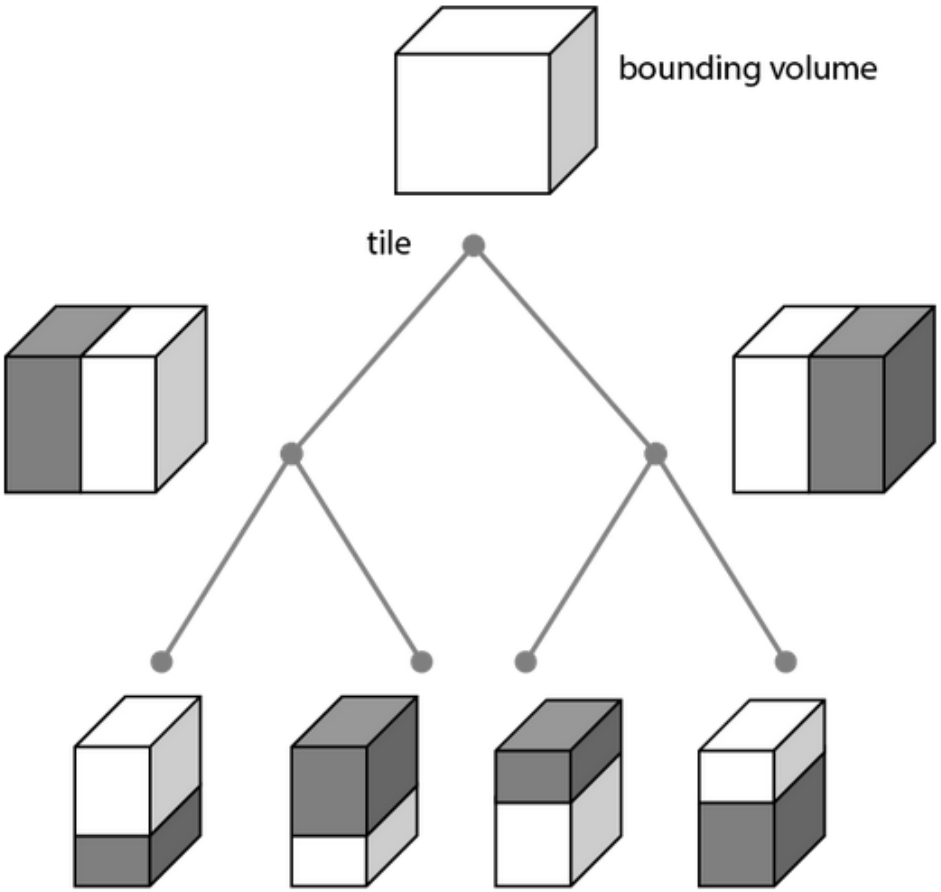
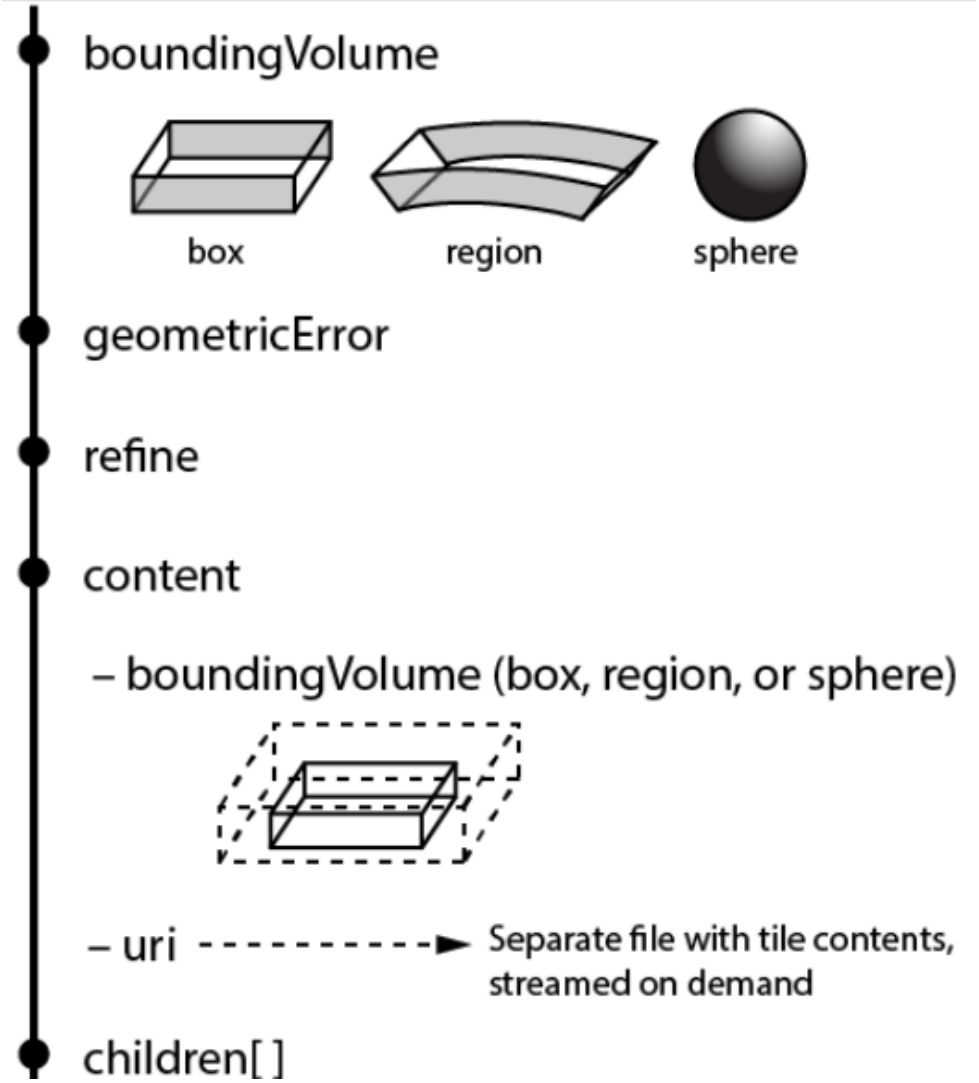


Figure 1: A sample 3D Tiles bounding volume hierarchy

Table of Contents

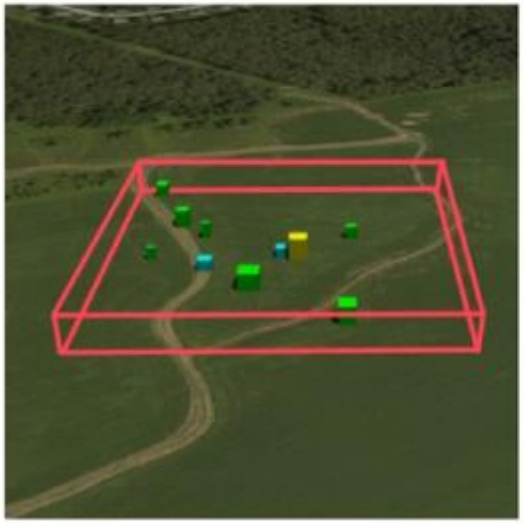
PROPRIÉTÉS D'UNE TUILE

- Volume englobant
- Erreur géométrique
- Raffinage
- Contenu
- Tuiles filles



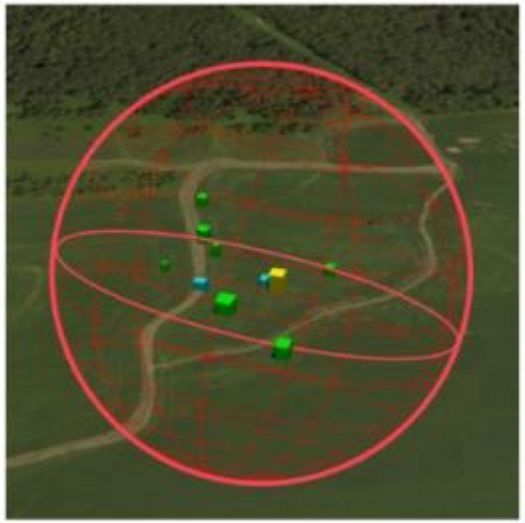
PLUSIEURS FORMES DE VOLUMES ENGLOBANTS

Bounding Box



```
"boundingVolume": {
   "box": [
     0,   0,   10,
     100, 0,   0,
     0,   100, 0,
     0,   0,   10
   ]
}
```

Bounding sphere



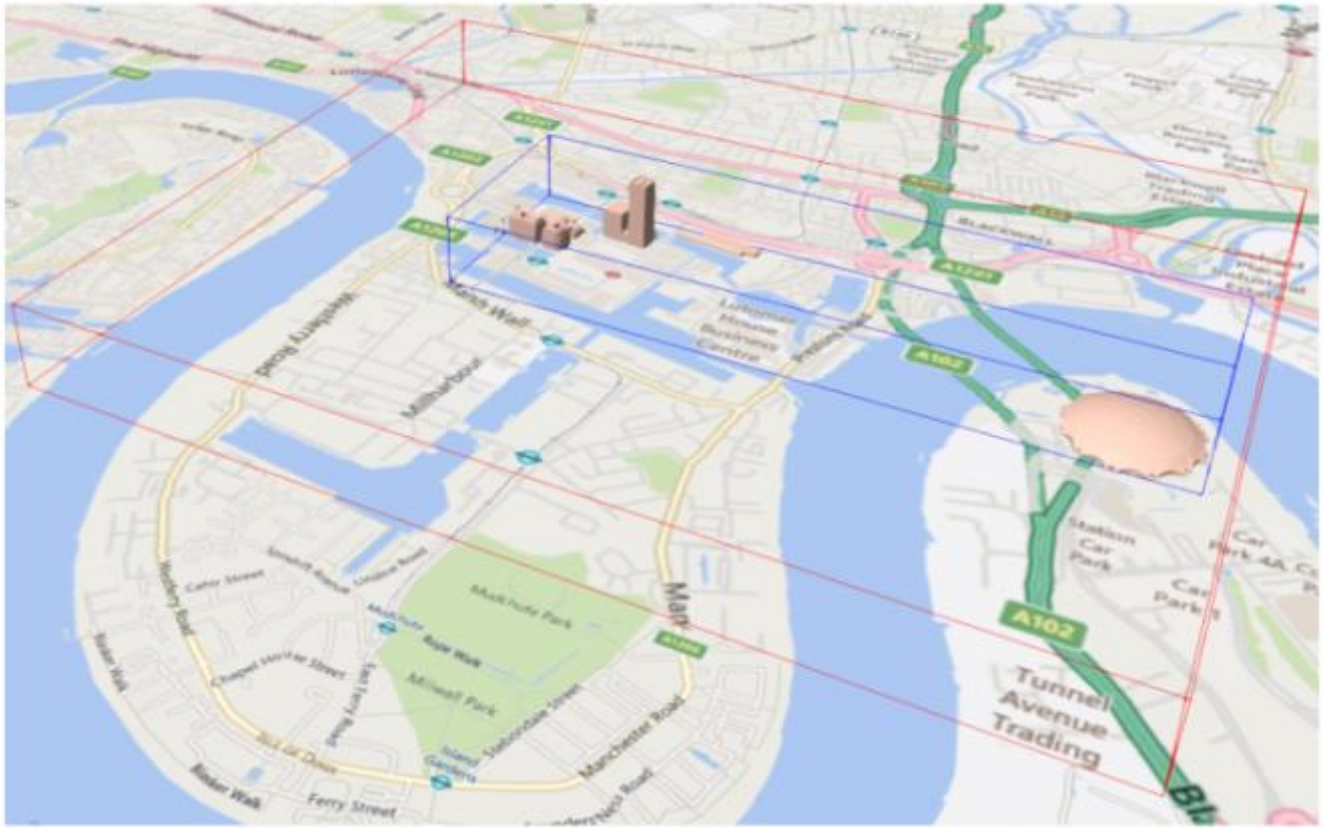
```
"boundingVolume": {
   "sphere": [
     0,
     0,
     10,
     141.4214
   ]
}
```

Bounding region

```
boundingVolume": {
  "region": [
    -1.3197004795898053,
    0.6988582109,
    -1.3196595204101946,
    0.6988897891,
    0,
    20
  ]
}
```

PLUSIEURS TYPES DE VOLUMES ENGLOBANTS

- Volume du jeu de tuiles
- Volume du contenu du jeu de tuiles
- Volume de requête du visualisateur (*viewerRequestVolume*)
  - https://www.youtube.com/watch?v=PgX756Yzjf4



PARAMÈTRE D'AFFICHAGE DU NIVEAU SUIVANT

- Plusieurs niveaux de zoom dans une même vue
- Variable d'erreur géométrique *Geometric Error*
  - En mètres
  - Basé sur mesures comme densité de points, taille de la tuile…
- Calcul de *Screen-Space Error*
  - En pixels
  - Evalue le niveau de détail de la vue courante
  - Si supérieur au maximum, on déclenche l'affichage du niveau suivant

- A votre avis…
  - *Une valeur haute de geometricError induit un raffinage plus ou moins agressif ?*

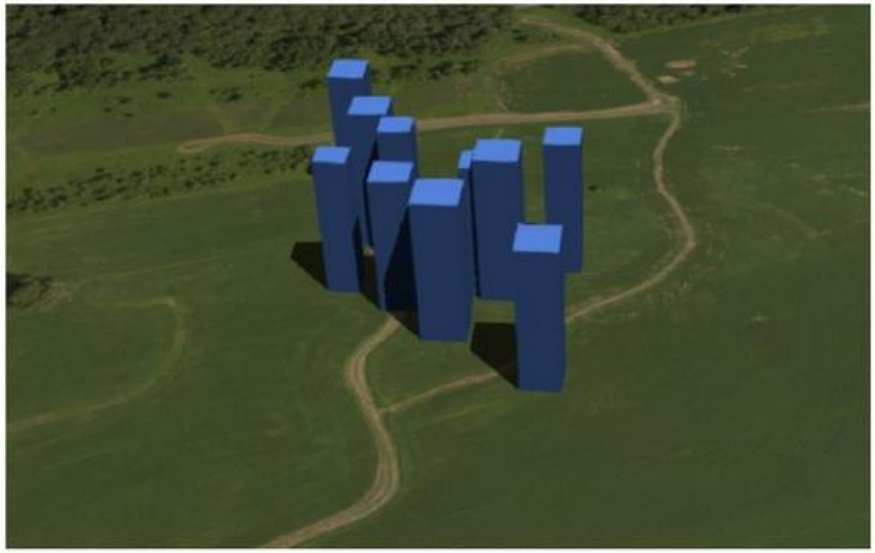
PARAMÈTRE D'ERREUR GÉOMÉTRIQUE

- *Une valeur haute de geometricError induit un raffinage plus agressif*
  - *Tuiles filles chargées et rendues plus tôt*
- *GeometricError décroît de la tuile-racine aux tuiles-feuilles*

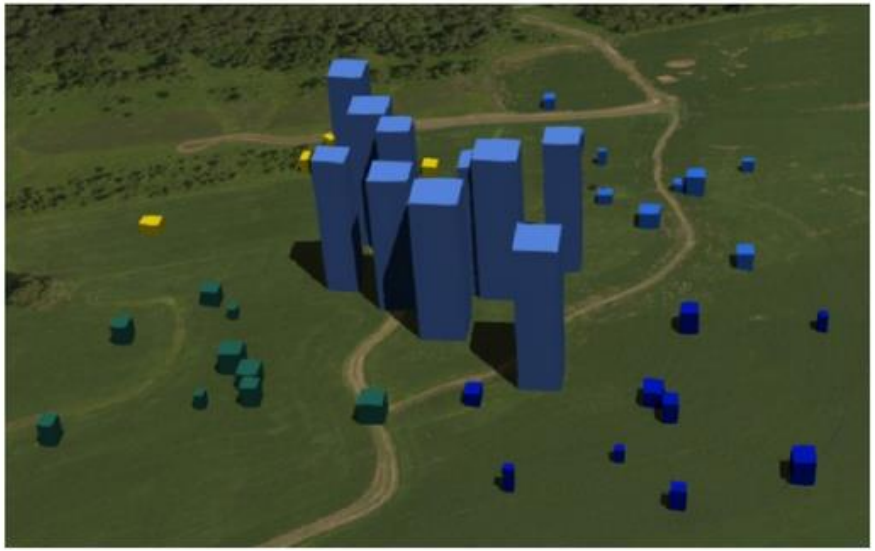
RAFFINAGE (*REFINEMENT*)

- D’un niveau au suivant…
  - Remplacement
  - Ajout
  - Combinaison des deux

Parent Tile

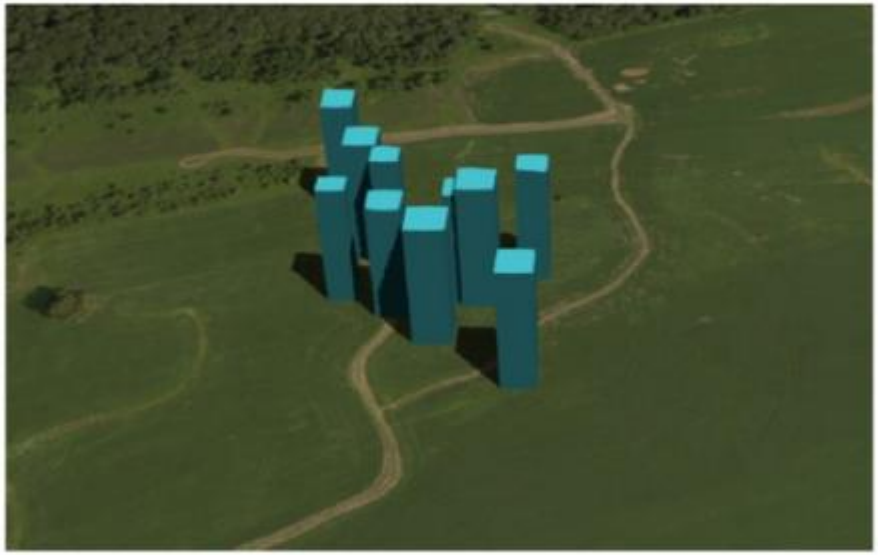


Refined



Parent Tile

Refined



AUTRES PROPRIÉTÉS D'UNE TUILE

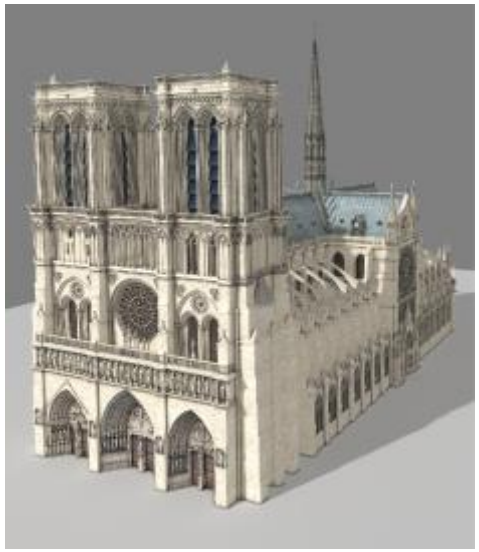
Properties	Type	Description	Required
boundingVolume	object	A bounding volume that encloses a tile or its content. Exactly one box, region, or sphere property is required.	Yes
viewerRequestVolume	object	A bounding volume that encloses a tile or its content. Exactly one box, region, or sphere property is required.	No
geometricError	number	The error, in meters, introduced if this tile is rendered and its children are not. At runtime, the geometric error is used to compute screen space error (SSE), i.e., the error measured in pixels.	Yes
refine	string	Specifies if additive or replacement refinement is used when traversing the tileset for rendering. This property is required for the root tile of a tileset; it is optional for all other tiles. The default is to inherit from the parent tile.	No
transform	number [16]	A floating-point 4x4 affine transformation matrix, stored in column-major order, that transforms the tile's content--i.e., its features as well as content.boundingVolume, boundingVolume, and viewerRequestVolume--from the tile's local coordinate system to the parent tile's coordinate system, or, in the case of a root tile, from the tile's local coordinate system to the tileset's coordinate system. transform does not apply to geometricError, nor does it apply any volume property when the volume is a region, defined in EPSG:4979 coordinates.	No, default: [1,0,0,0, 0,1,0,0, 0,0,1,0, 0,0,0,1]
content	object	Metadata about the tile's content and a link to the content.	No
children	array[]	An array of objects that define child tiles. Each child tile content is fully enclosed by its parent tile's bounding volume and, generally, has a geometricError less than its parent tile's geometricError. For leaf tiles, the length of this array is zero, and children may not be defined.	No
extensions	object	Dictionary object with extension-specific objects.	No
extras	any	Application-specific data.	No

EXEMPLE D'ENCODAGE DE TUILE

```
{
  "boundingVolume": {
    "region": [
      -1.2419052957251926,
      0.7395016240301894,
      -1.2415404171917719,
      0.7396563300150859,
      0,
      20.4
    ]
  },
  "geometricError": 43.88464075650763,
  "refine" : "ADD",
  "content": {
    "boundingVolume": {
      "region": [
        -1.2418882438584018,
        0.7395016240301894,
        -1.2415422846940714,
        0.7396461198389616,
        0,
        19.4
      ]
    },
    "uri": "2/0/0.b3dm"
  },
  "children": […]
}
```

FORMATS DE TUILES

- *Batched 3D Model (*.b3dm)*
  - *agrégation* hors-ligne des modèles 3D hétérogènes
  - Optimise le rendu et l'interaction à la volée



- *Instanced 3D Model (*.i3dm)*
  - *Adapté pour des jeux de données homogènes*
  - *Instances identiques*

- Nuages de points (**.pnts*)

- Tuiles composites (*.cmpt)
  - Agrège des tuiles des différents formats ci-dessus

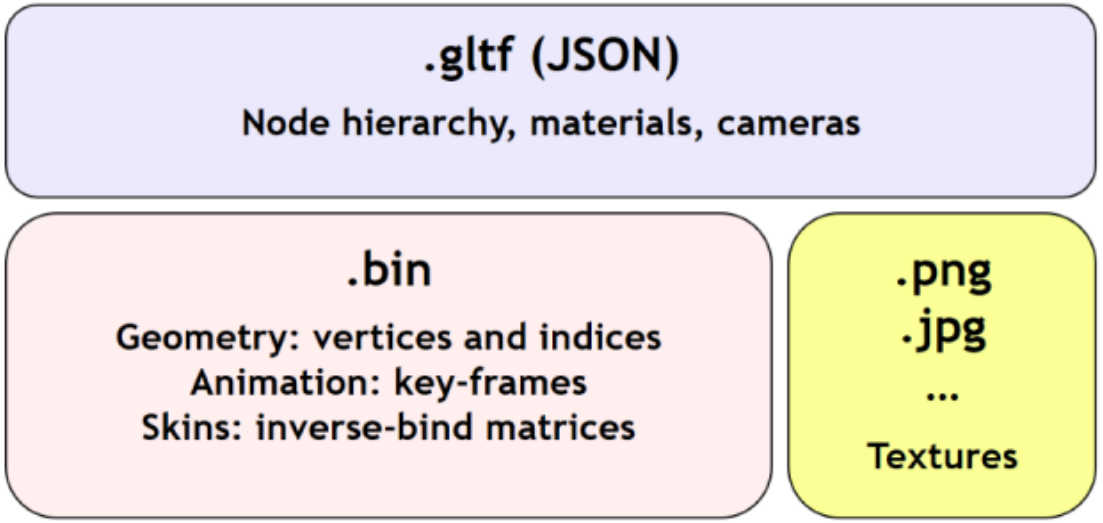
TRANSFORMATIONS

- Optionnel
- Matrice de transformation affine 4x4
- Par exemple
  - Tuile dans un système de coordonnées local
  - Adaptation du **format glTF** qui inverse y et z

```
[
 1.0,  0.0,   0.0, 0.0,
 0.0,  0.0, -1.0, 0.0,
 0.0,  1.0,   0.0, 0.0,
 0.0,  0.0,   0.0, 1.0
 ]
```

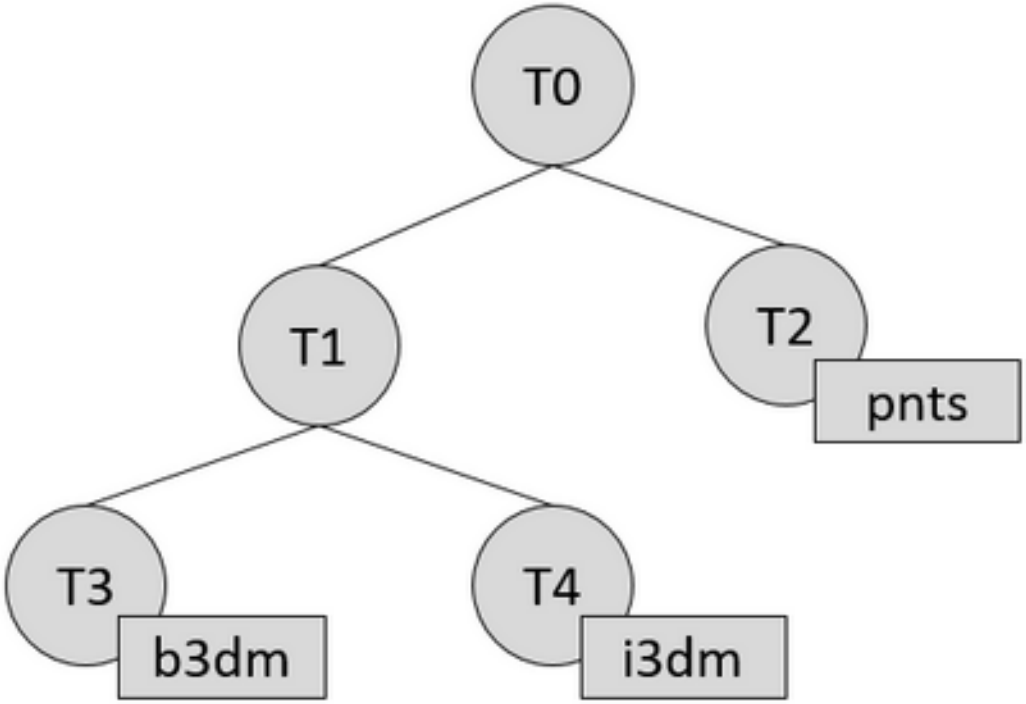
ENCODAGE : FORMAT GLTF

- *Graphic Library Transmission Format*
- Format interopérable et extensible pour la transmission et le chargement de contenu 3D
- glTF contient :
  - Une description de la scène en JSON facile à parser
  - Des fichiers binaires représentant la géométrie, les animations, etc, stockés de manière à être chargés directement par les GPU



EXEMPLE : TRANSFORMATION DES TUILES FILLES

- TO: [T0]
- T1: [T0][T1]
- T2: [T0][T2]
- T3: [T0][T1][T3]
- T4: [T0][T1][T4]



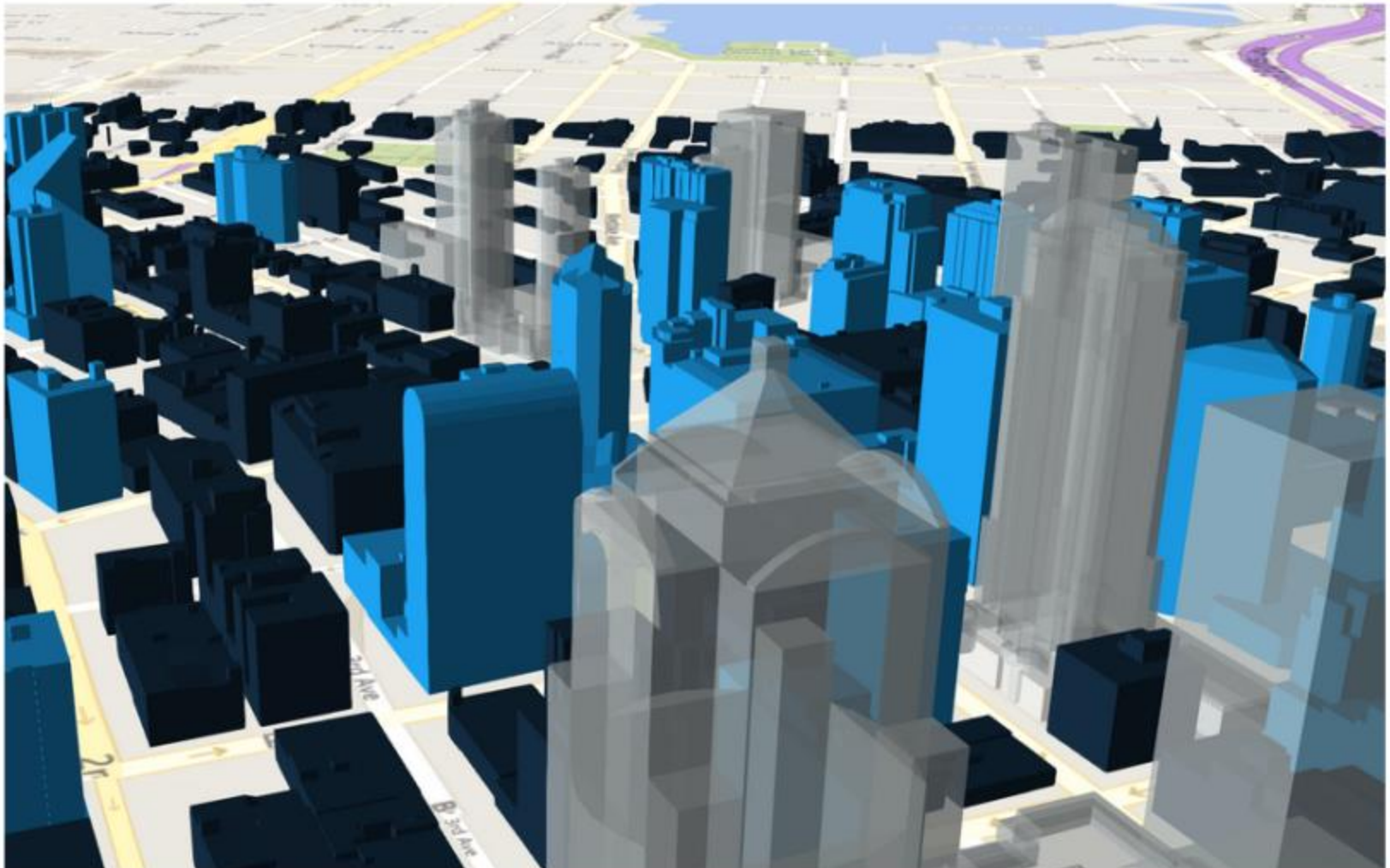
PROPRIÉTÉS D'UN JEU DE TUILES

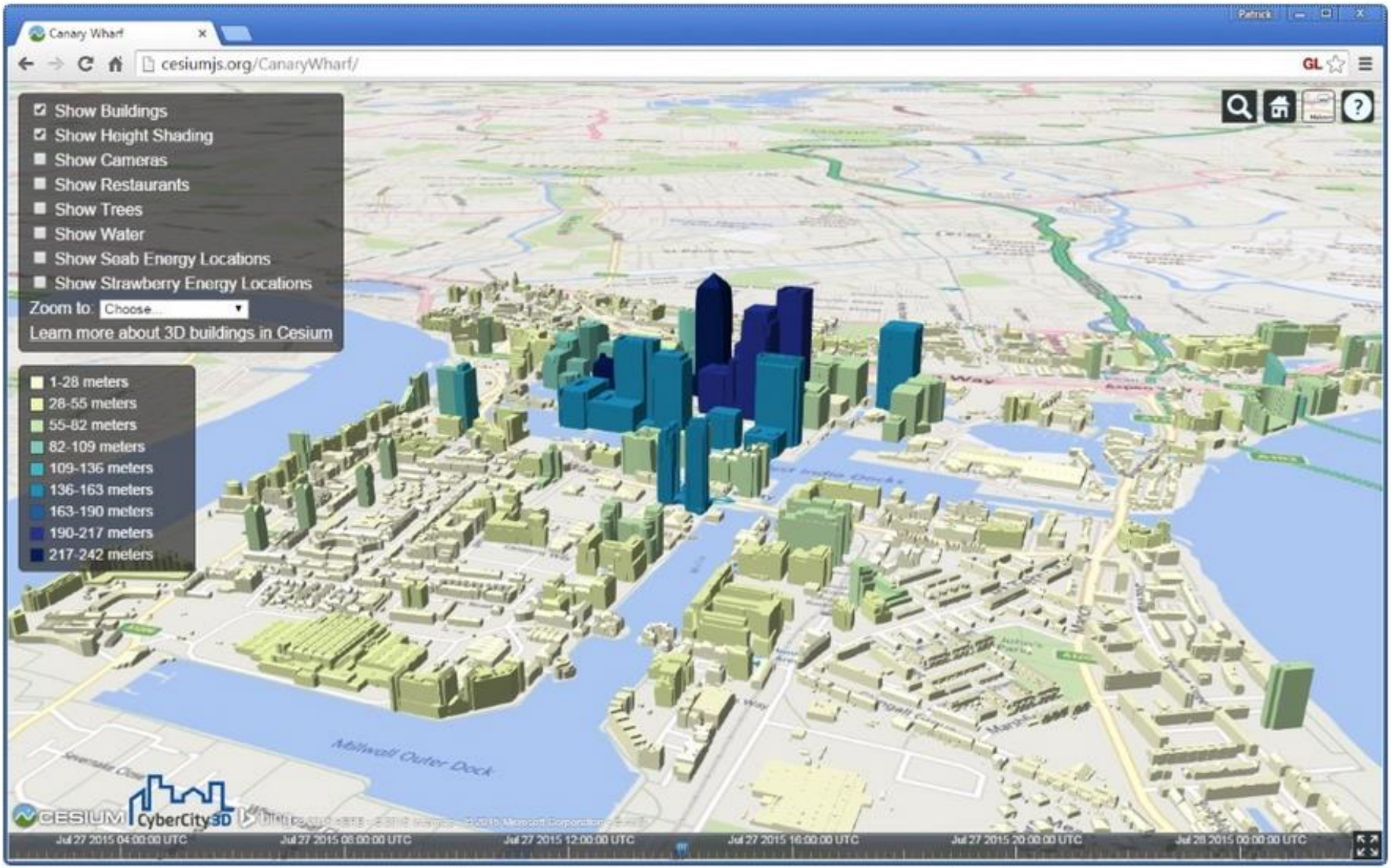
Properties	Type	Description	Required
asset	object	Metadata about the entire tileset.	Yes
properties	any	A dictionary object of metadata about per-feature properties.	No
geometricError	number	The error, in meters, introduced if this tileset is not rendered. At runtime, the geometric error is used to compute screen space error (SSE), i.e., the error measured in pixels.	Yes
root	object	A tile in a 3D Tiles tileset.	Yes
extensionsUsed	string [1-*]	Names of 3D Tiles extensions used somewhere in this tileset.	No
extensionsRequired	string [1-*]	Names of 3D Tiles extensions required to properly load this tileset.	No
extensions	object	Dictionary object with extension-specific objects.	No
extras	any	Application-specific data.	No

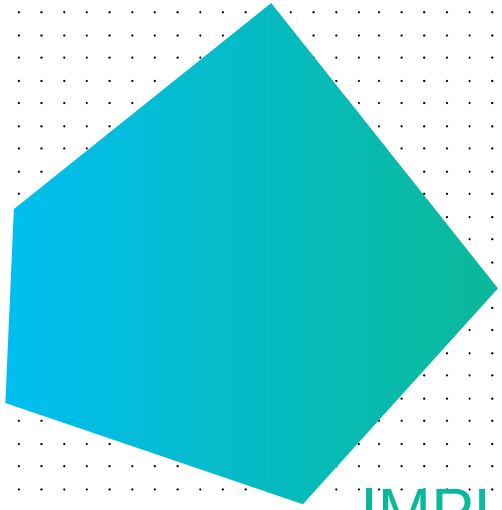
LE STYLE ADAPTE L'AFFICHAGE D'UN JEU DE TUILES

- Expressions modifiant l'affichage des entités
- Par exemple adaptent leur couleur en fonction de leurs propriétés
- Une tuile sans entités peut être considérée comme une entité

```
{
    "show" : "${Area} > 0",
    "color" : {
        "conditions" : [
            ["${Height} < 60", "color('#13293D')"],
            ["${Height} < 120", "color('#1B98E0')"],
            ["true", "color('#E8F1F2', 0.5)"]
        ]
    }
}
```




Canary Wharf
cesiumjs.org/CanaryWharf/
Show Buildings
Show Height Shading
Show Cameras
Show Restaurants
Show Trees
Show Water
Show Saab Energy Locations
Show Strawberry Energy Locations
Zoom to: Choose...
Learn more about 3D buildings in Cesium
1-28 meters
28-55 meters
55-82 meters
82-109 meters
109-136 meters
136-163 meters
163-190 meters
190-217 meters
217-242 meters
Millwall Outer Dock
CESIUM
CyberCity3D
Jul 27 2015 04:00:00 UTC
Jul 27 2015 08:00:00 UTC
Jul 27 2015 12:00:00 UTC
Jul 27 2015 16:00:00 UTC
Jul 27 2015 20:00:00 UTC
Jul 28 2015 00:00:00 UTC



IMPLÉMENTATIONS DE 3DTILES

QUELQUES IMPLÉMENTATIONS RÉFÉRENCÉES

- https://www.youtube.com/watch?v=KoGc-XDWPDE

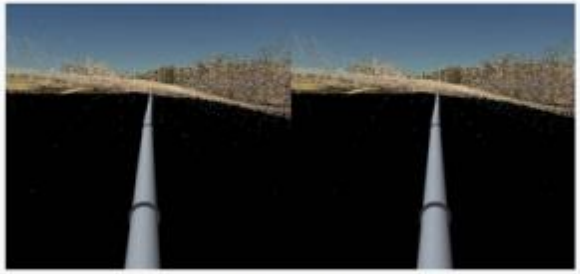
Cesium ion converters

Cesium

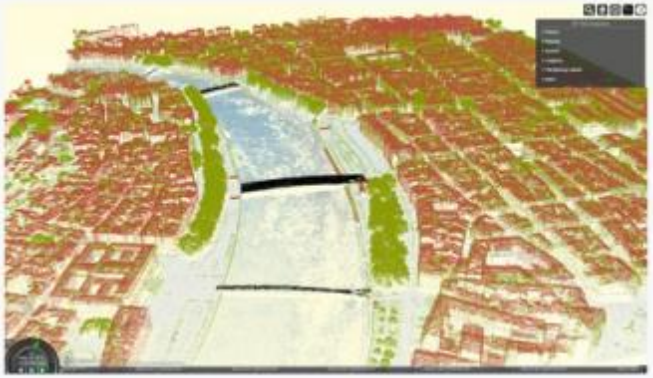
Vricon

Federal Office of Topography

swisstopo

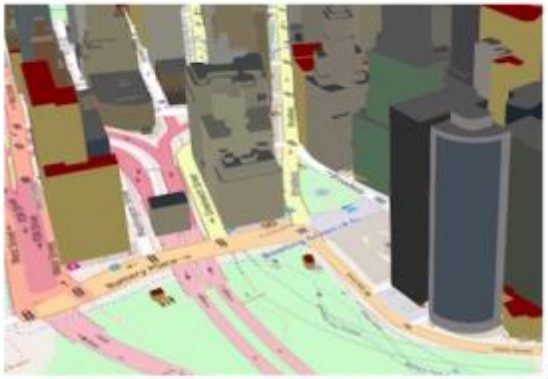


VirtualGIS



LOPoCS and py3dtiles

iTowns 2

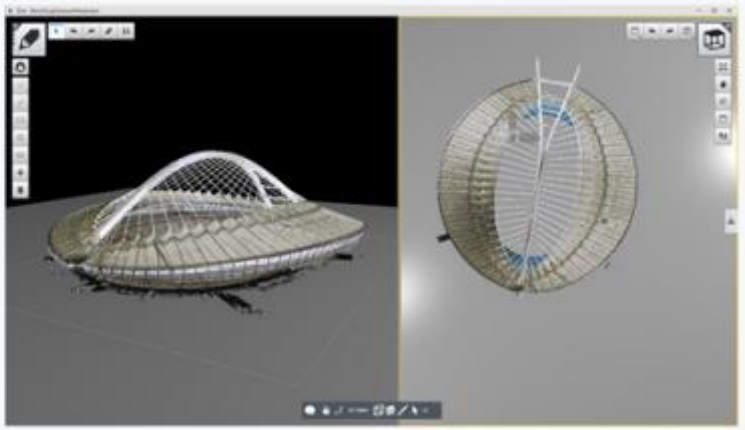


osm-cesium-3d-tiles

geopipe

3D Digital Territory Lab

Bentley ContextCapture

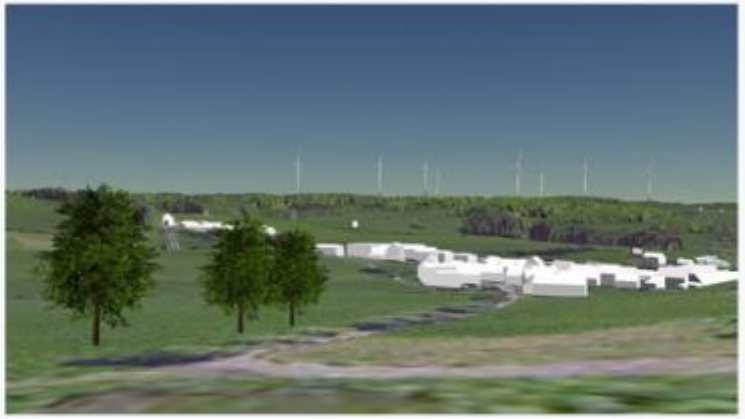


Bentley MicroStation (in progress)

CyberCity3D

virtualcitySYSTEMS

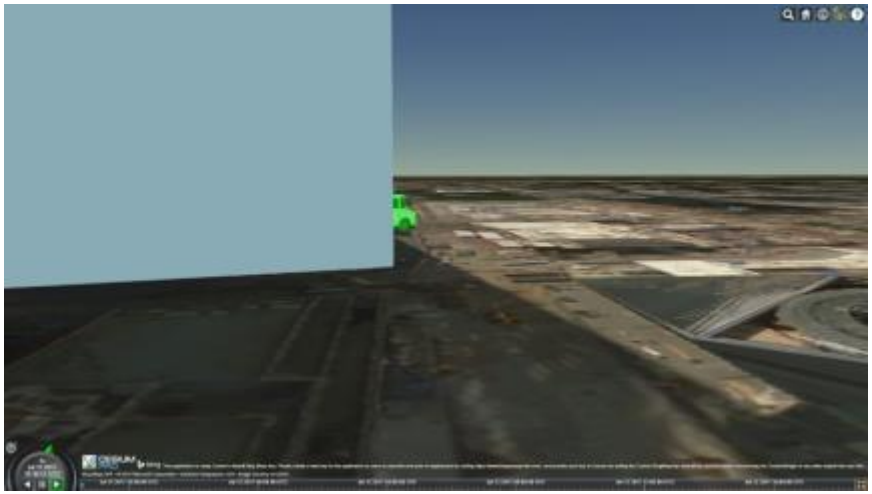
Cityzenith

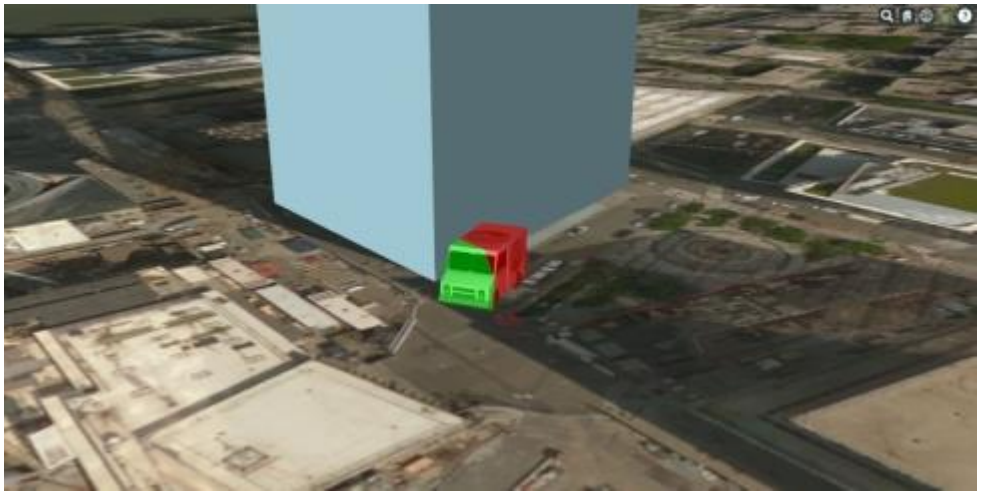


Fraunhofer

PERSPECTIVES

- Prise en compte du temps
  - Illustrer l'évolution temporelle du modèle
  - Analyser les différences
- Langage de style permettant des analyses
  - Exemple : couleur adaptée à l'ombre
    - **"color" : "isVisibleFrom(${anotherFeature.location}) ? color('green') : color('red')"**

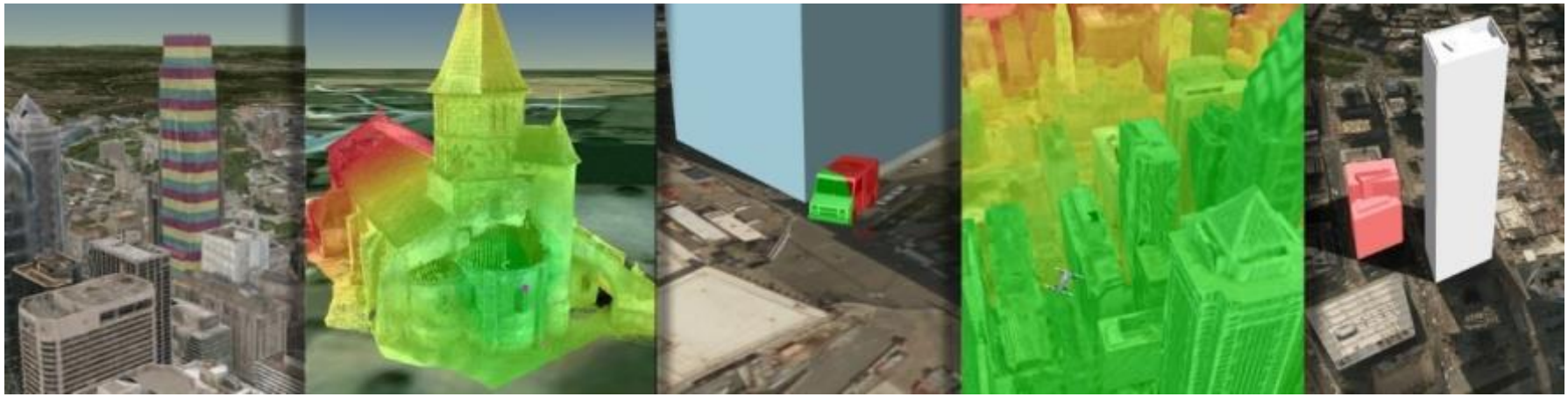




PERSPECTIVES

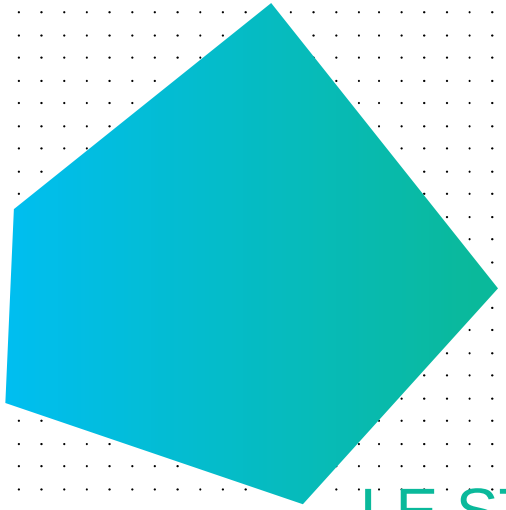
- Tuiles vecteur 3D pour la classification hétérogène
- Améliorations diverses : compression, schémas de tuilage

- Source : https://cesium.com/blog/2017/07/12/the-next-generation-of-3d-tiles/
  - https://www.youtube.com/watch?time_continue=6&v=IkNNNeisgHo



CONCLUSION

- 3DTiles est un standard communautaire OGC approuvé, dont la spécification détaillée est finalisée et disponible sur leur site ainsi que sur Github.
- La solution d'implémentation présentée par l'auteur n'est pas libre : les outils développés par AGI sont propriétaires, ainsi que les techniques auxquelles ils font appel.
- Par exemple, l'algorithme de tuilage n'est défini nulle part dans les spécifications, mais nécessaire pour créer un jeu de données optimisé
- Le visualisateur Cesium est ouvert, pas la préparation des données. D'autres solutions existent pour implémenter le standard.



LE STANDARD I3S

ASPECTS NORMATIFS

STATUT D'I3S

- I3S : Indexed 3D Scene Layer, *couche de scène 3D indexée*
- Standard communautaire OGC, proposé en février 2017, approuvé en aout 2017 en version 1.0.
- Spécification développée par Esri.
- Spécification disponible sur le site de l'OGC : http://docs.opengeospatial.org/cs/17-014r5/17-014r5.html
- Aussi disponible sur Github : https://github.com/Esri/i3s-spec
  - Attention, cette version peut être en avance sur le standard approuvé

CONTENU DU FORMAT

GÉNÉRALITÉS

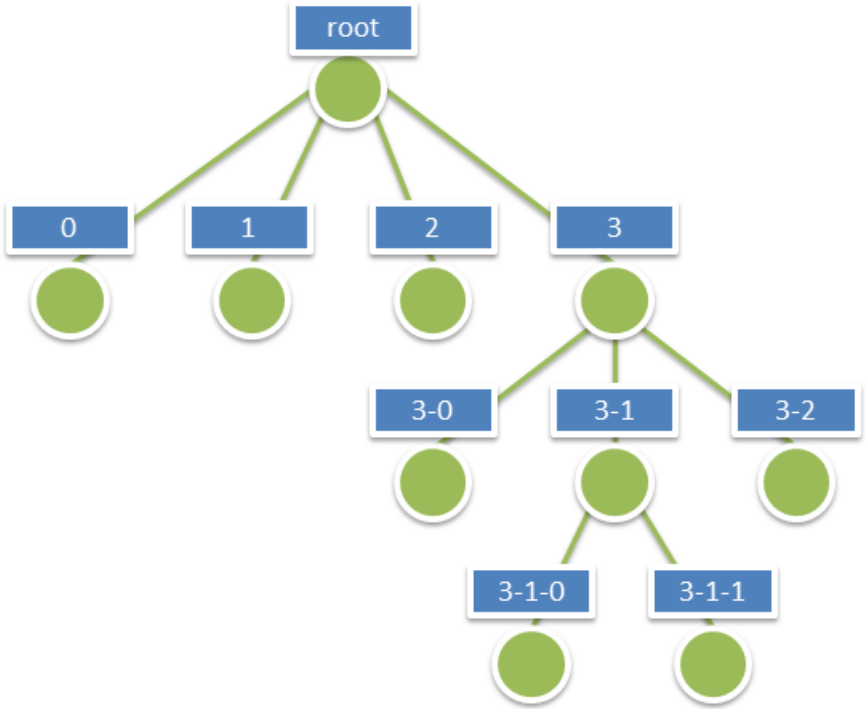
- **Modèle ouvert** de conteneur pour des données 3D géographiques **hétérogènes**
- Pas de limite de volumétrie → besoin de bien optimiser ses données
- Permet à la fois la consommation **locale** et **depuis un serveur**
- Intègre différents types de données :
  - Objets 3D
  - Maillages
  - Ponctuels.
- Il est prévu d'ajouter à ces types les **Lignes**, **Polygones** et **Nuages de Points**, mais ils sont encore en Beta du côté d'ESRI et leurs implémentations sont lacunaires.

STRUCTURE

- I3S fonctionne avec un système hiérarchique à base de nœuds, ces derniers pouvant être représentés mentalement comme similaires à des tuiles
- Chaque nœud contient des sous-nœuds.
- Ces nœuds sont indexés et peuvent être identifiés soit par **un entier**, soit par **un *treekey***, une chaine de caractères permettant d'identifier rapidement la position du nœud dans la hiérarchie
  - Par exemple, le treekey 2-0-1 comprend trois nombres, il est donc au 4e niveau de profondeur et a pour nœud parent le nœud 2-0 qui lui-même a pour parent le nœud 2, qui sera lui-même un enfant du nœud racine, appelé « root ».

STRUCTURE

Exemple de treekeys



STRUCTURE

- Le format I3S modélise l'information sur les nœuds à l'aide d'un ensemble de ressources :
  - Documents d'index de nœuds (*Node Index Documents*),
  - Données d'entités (*Feature Data*),
  - Géométrie,
  - Attributs,
  - Textures,
  - Descripteurs partagés,
- Ces ressources représentent l'ensemble des caractéristiques ou des éléments de données d'un nœud donné et sont toujours attachées à un nœud.
- Le document d'index du nœud est une ressource légère représentant un nœud, sa topologie dans l'arbre et inclut des références à d'autres sous-ressources.
- Parmi les informations sur le nœud, figure le volume englobant, qui sert notamment à déterminer si le client doit ou non afficher le nœud ; ce volume englobant peut être **soit une sphère soit un parallélépipède rectangle**.

STRUCTURE

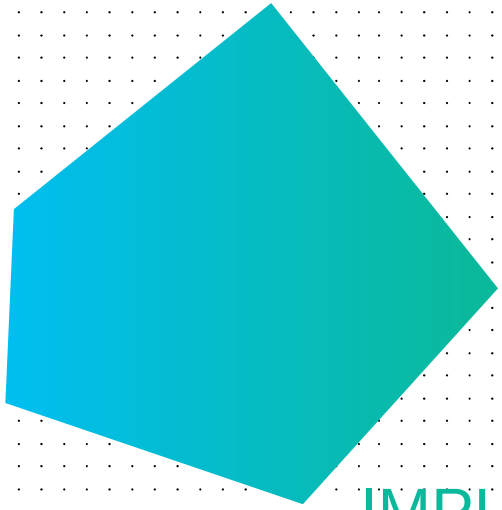
- Le standard n’impose pas d’arbre de données particulier
- Supporte aussi bien **les partitions régulières de l'espace** (par exemple, les quadrees et les octrees) ainsi que **les partitionnements liés à la densité** (par exemple, les R-arbres, les arbres k-d).
- C’est à l’utilisateur de traiter ses données.

NIVEAUX DE DÉTAIL (LOD, LEVEL OF DETAIL)

- Système un peu similaire aux niveaux de tuiles d'une pyramide 2D.
- Les LoD ont une architecture arborescente, le nœud le plus bas contient l'entité originale dans son intégralité au plus haut niveau de détail, et à mesure que l'on remonte l'arborescence, le niveau de détail diminue.
- Cette diminution peut se faire par
  - sous-échantillonnage des textures,
  - généralisation des entités ou des mailles,
  - agglomération de plusieurs entités,
  - découpage de certains gros objets (rivières ou lignes de côtes par exemple),
  - suppression de certains objets.
- Chaque niveau de détail d'un objet contient des informations de qualité, comme la taille maximale que l'objet peut occuper à l'écran ou un encadrement de son échelle. Au moment du rendu, le client peut continuer à descendre l'arborescence jusqu'à trouver un objet d'une qualité suffisante.

FLEXIBILITÉ

- Nombreux choix d'implémentations sont disponibles, entre différents types de données ou au sein d'un même type :
- La subdivision des tuiles en 3D est adaptable : quadtrees, octrees et R-arbres sont autorisés, par exemple.
- Les volumes englobants peuvent être implémentés comme des sphères englobantes (*Minimum Bounding Sphere*, MBS) ou des parallélépipèdes englobants (*Oriented Bounding Box*).
- La structure arborescente des nœuds peut être au choix complète, avec des pointeurs vers les parents, enfants, et fratrie, ou minimale, avec simplement les index qui devront être traités ensuite par le client mais de taille fixe, plus adaptée à un accès par pagination.
- Les données de géométrie peuvent être en JSON ou en format binaire.
- Les niveaux de détail peuvent être liés à différentes métriques : la taille de l'objet sur l'écran, son échelle, ou encore la distance entre l'objet et la caméra.



IMPLÉMENTATIONS

IMPLÉMENTATIONS HORS ESRI

- ContextCapture, par Bentley
- Vricon, par Vricon
- Pix4D, par Pix4D
- Melown Photogrammetry, par Melown

- Ces implémentations sont commerciales et propriétaires
- Parfois partielles (ne gèrent que certains types d'objets)

IMPLÉMENTATIONS ESRI

Scene Layer Types	ArcGIS Enterprise			ArcGIS Pro		
	Publishing	Consuming	First Released	Publishing	Consuming	First Released
3D Object	✓	✓	10.5	✓	✓	1.4
Integrated Mesh		✓	10.5	✓	✓	1.4/2.1*
Point	✓	✓	10.5	✓	✓	1.4
Point Cloud		✓	10.5.1	✓	✓	2.0
Building Scene Layer		✓	10.7	✓	✓	2.2

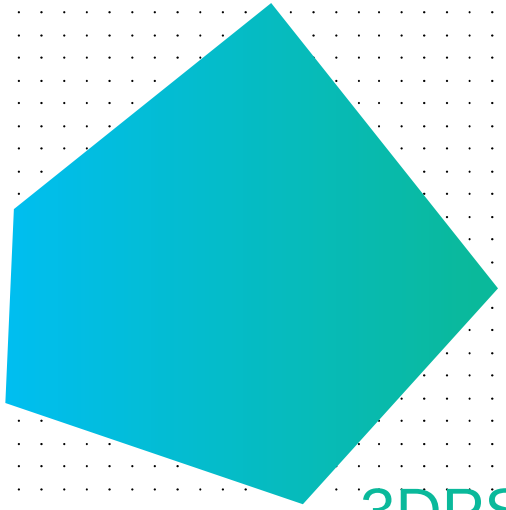
Scene Layer Types	ArcGIS Online		ArcGIS API for Javascript			ArcGIS Runtime		
	Publishing	Consuming	Publishing	Consuming	First Released	Publishing	Consuming	First Release
3D Object	✓	✓		✓			✓	100.2
Integrated Mesh		✓		✓			✓	100.2
Point	✓	✓		✓				
Point Cloud		✓		✓				
Building Scene Layer		✓		✓				

Scene Layer Types	ArcGIS Earth			Esri City Engine		
	Publishing	Consuming	First Released	Publishing	Consuming	First Released
3D Object		✓	1.6	✓		2017.1
Integrated Mesh		✓	1.6			
Point						
Point Cloud						
Building Scene Layer						

Source :
Github I3S

CONCLUSION

- I3S est un standard communautaire OGC approuvé, dont la spécification détaillée est finalisée et disponible sur leur site ainsi que sur Github.
- Comme pour 3D Tiles, si le format est publié librement, l'implémentation n'est pas libre : les outils développés par Esri sont propriétaires, ainsi que les techniques auxquelles ils font appel.
- Par exemple, l'algorithme de tuilage n'est défini nulle part dans les spécifications bien qu'il soit indispensable pour implémenter proprement I3S.
- Cette remarque est modérée par l'environnement dans lequel ce standard sera supposément utilisé ; pour SPP, l'intérêt d'I3S par rapport à 3D Tiles réside dans sa capacité à s'intégrer aisément avec une architecture Esri dans laquelle la compatibilité sera facilitée et les outils de toute façon non libre.
- Utilisation possible mais souci d'interopérabilité cependant.

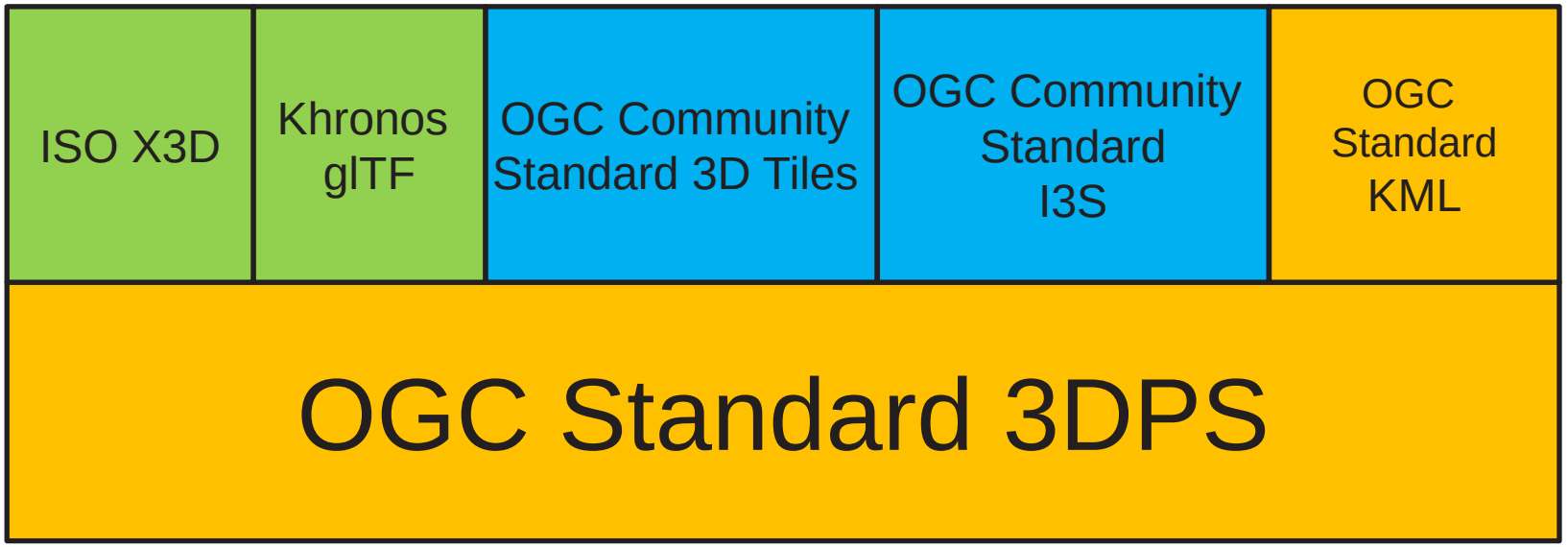


3DPS : INTEROPERABILITÉ ENTRE I3S ET CESIUM

LISTE DE FORMATS OUVERTS

La vision :

Pour les images, on ne parle pas beaucoup des formats de données eux-mêmes, on sélectionne seulement les formats appropriés (jpeg, png, etc.). L'idée est d'avoir la même chose en 3D.



CAS D'UTILISATION

- Plusieurs villes publient les besoins de chauffage des bâtiments comme un service aux citoyens (à l'avenir). Les données sont constituées d'un modèle urbain 3D et d'un label énergie (kWh/m^2.a) par bâtiment.

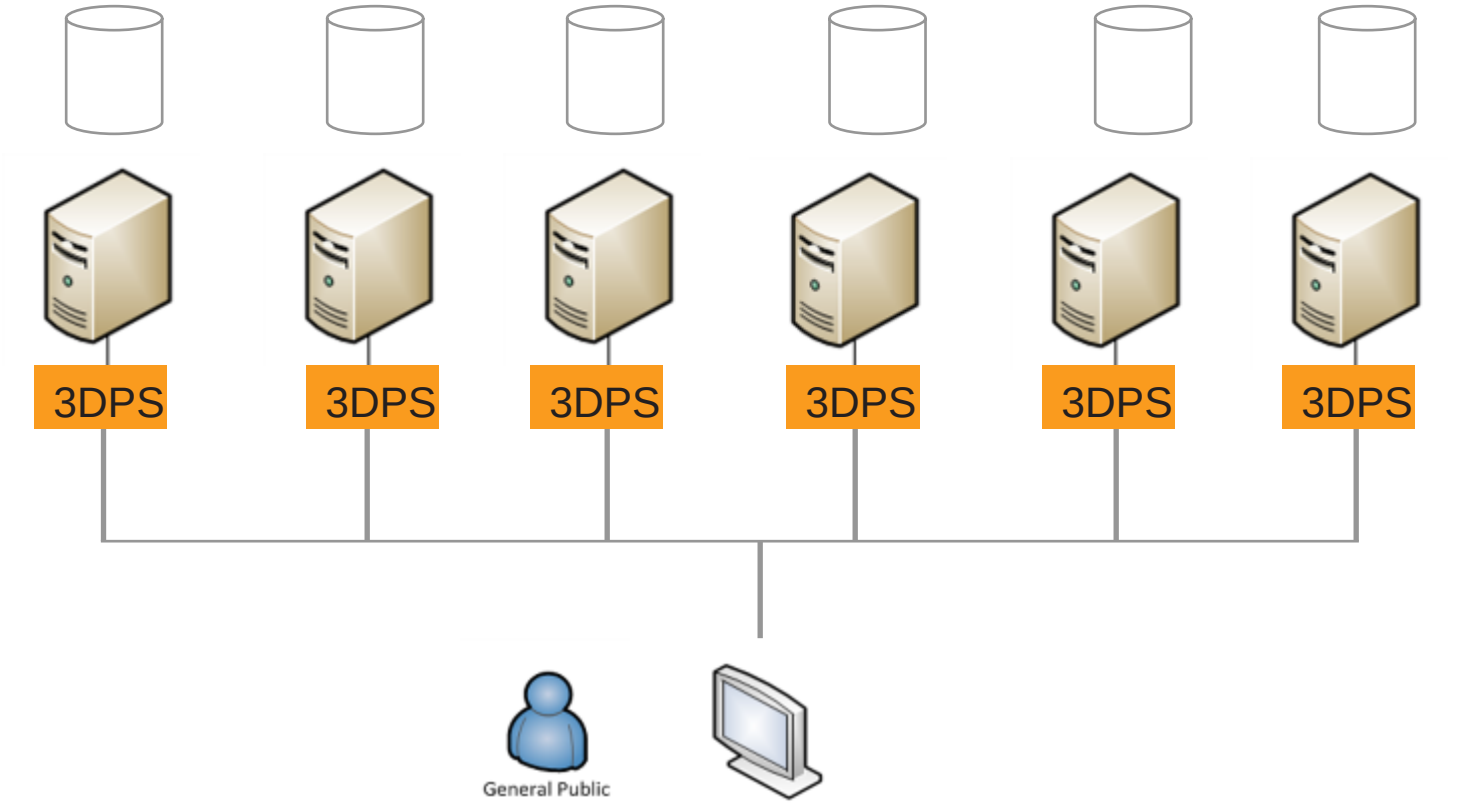
- Le fournisseur de données veut partager les données, mais il veut aussi d'avoir le contrôle des données et du serveur (= modèles 3D distribués).

- Le citoyen veut jeter un coup d'œil à son bâtiment en tapant une adresse et ne se soucie pas du tout des formats de données ou des URL spécifiques pour trouver un modèle.

CAS D'UTILISATION

Données fournies par le LGL BW, demande de chauffage basée sur le Monthly Energy Balance Model, simulée avec SimStadt.

GESTION DE DONNÉES DISTRIBUÉES



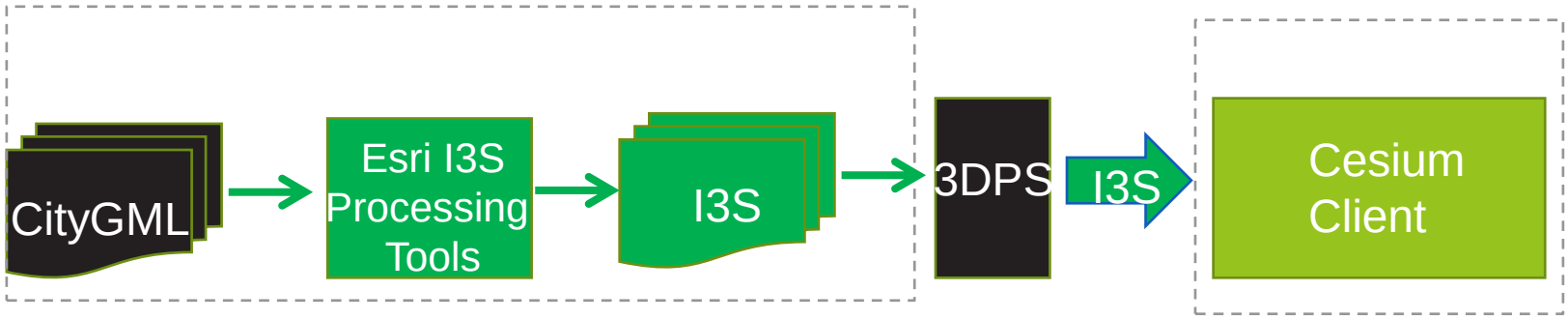
3DPS EN BREF

Représentation: 3DP (3D Portrayal) = WVS (View) + W3DS services

- **GetResourceById**
- **GetView service (selon zone (boundingBox), offset, viewpoint, LoD, format) : résultat image (JPEG ou PNG) => visualisation par humain / exploitation visualisateur ou client léger (rendering)**
- **GetScene service (selon zone (boundingBox), offset, viewpoint, LoD, format) : résultat vecteur (X3D, KML ou VRML) => exploitation client léger (rendering)**
- **Tests de mise en œuvre avec 3D Tiles et I3S (Esri)**

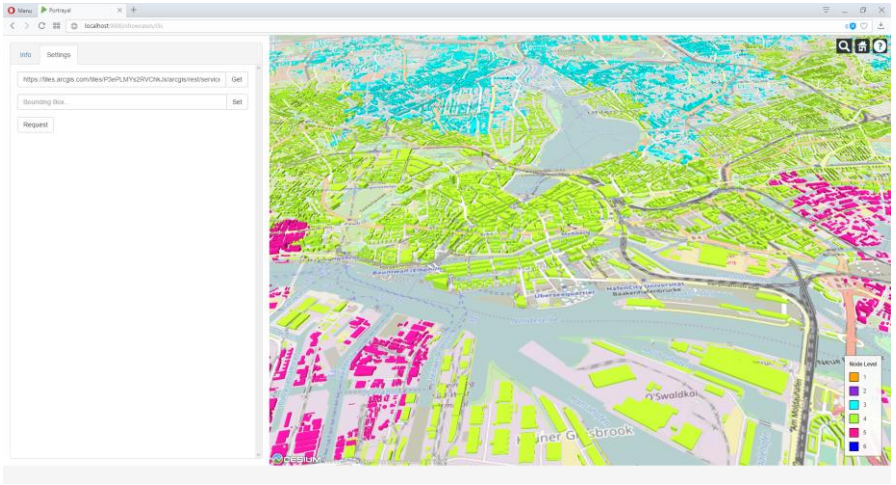
EXPÉRIMENTATIONS

- **Utiliser Cesium Client, afficher 3D Tiles – c'est fait**
- **Utiliser Cesium Client, afficher i3s – c'est fait**
- **Utiliser Cesium Client**
  - **interroger la région spatiale**
  - **les service de rendu délivre l'information**



EXPÉRIMENTATION 1 : I3S DANS CESIUM

Aperçu des technologies utilisées :



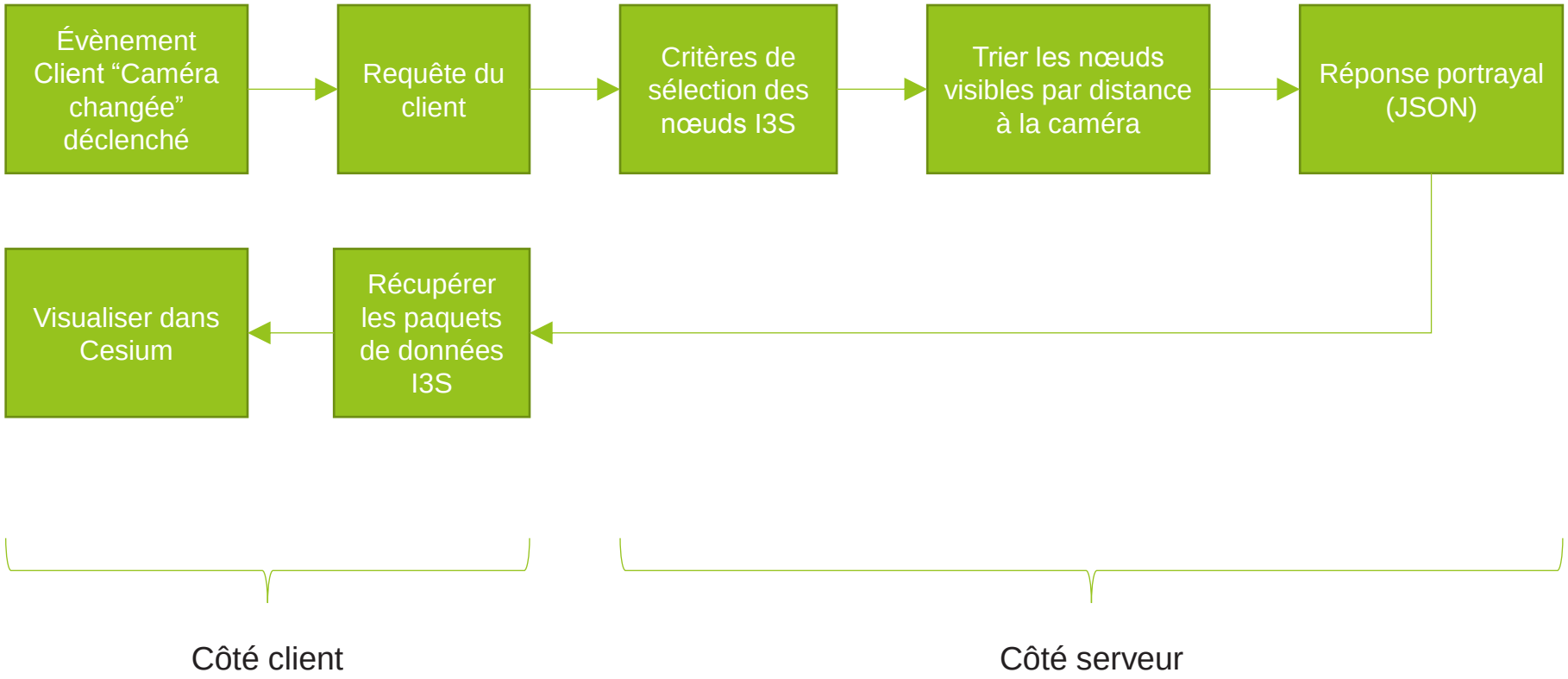
Client:

- Cesium
- JavaScript
- Bootstrap
- jQuery

Server:

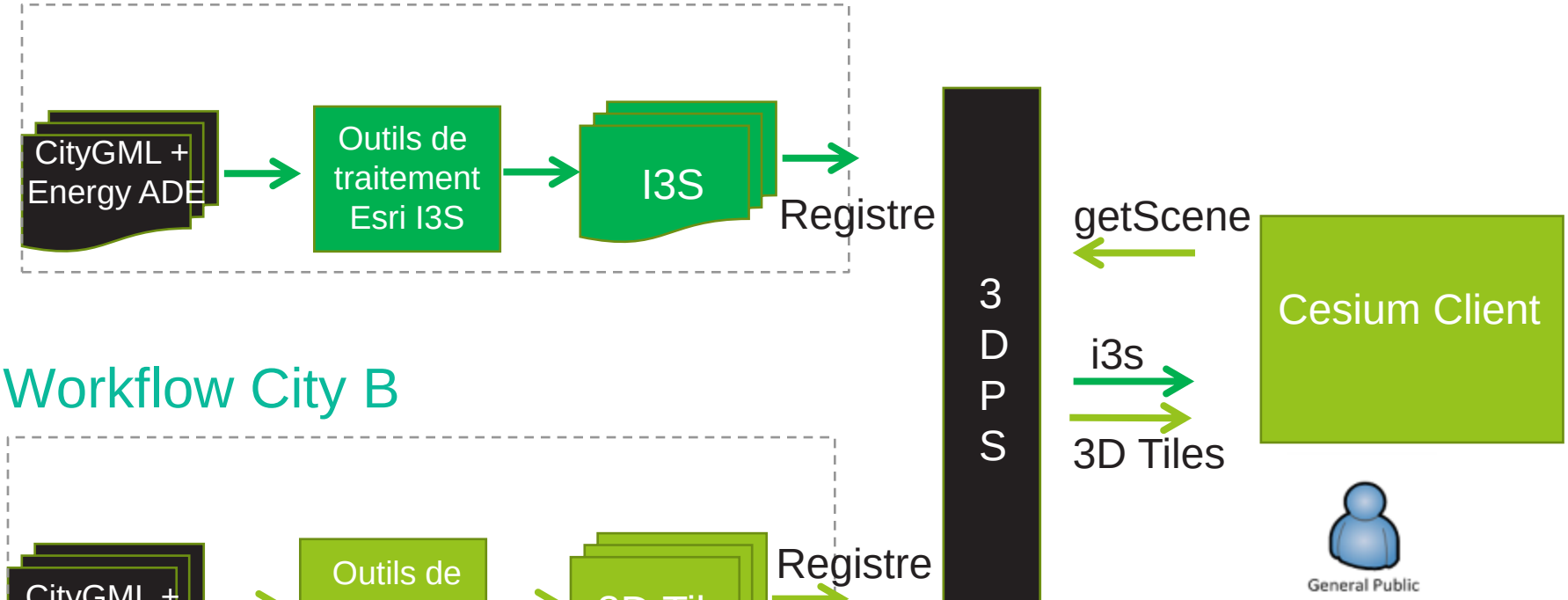
- Play Framework
- Java

RÉSUMÉ DES COMMUNICATIONS



RÉSUMÉ

Workflow City A

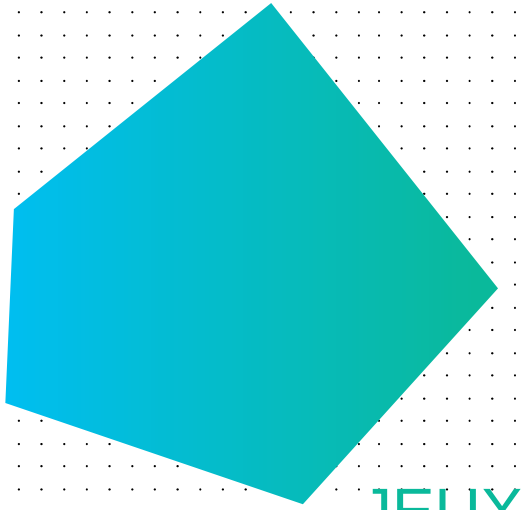


- Workflow City B

PROPOSITIONS D'EXTENSION DANS 3DPS 1.

- **cullingvolume**
  - 24 valeurs
  - 6 plans
  - 4 valeurs par plan (normale au plan + distance du plan à l'origine)
- **camera**
  - 6 valeurs
  - position de la caméra
  - direction de la caméra
- **frustum**
  - 3 valeurs
  - distances du plan d'écrêtage (near, top, right)
- **drawingbuffer**
  - 2 valeurs
  - largeur et hauteur de la fenêtre Césium en pixels
- **time**
  - 1 valeur
  - heure de la requête (en temps epoch)

RAPPORT D'INGÉNIERIE SUR L'INTEROPÉRABILITÉ ET LES PERFORMANCES DE 3D TILES ET I3S



JEUX DE DONNÉES

Conversions depuis CityGML

ORIGINE DES DONNÉES

- 1,1 million de bâtiments sur New York en LoD 1 et 2, sans textures :
  - https://cesiumjs.org/NewYork/index.html
- 500 000 bâtiments à Berlin en LoD 2, avec textures
  - https://www.businesslocationcenter.de/berlin3d-downloadportal/

- Données converties depuis CityGML vers 3D Tiles et I3S via des outils ad hoc, pour pouvoir tester avec les mêmes données initiales.

CONVERSION DES DONNÉES : CITYGML → 3DTILES → CESIUM

- Outil (propriétaire) développé par AGI pour l'occasion.
- 1. Découpage du CityGML et conversion en objets glTF (GL Transmission Format, basé sur JSON) distincts, avec changement d'EPSG (3DTiles requiert du WGS84), puis collage des objets au terrain (pour éviter qu'ils ne flottent dans les airs ou s'enfoncent dans le sol).
- 2. Tuilage
- 3. glTF → 3DTiles, avec optimisations (mémoire, taille des données, atlas de textures).

CONVERSION DES DONNÉES : CITYGML → GDB → I3S

- 1. Conversion des tuiles CityGML en Geodatabase via FME.
- 2. Fusion des gdb
- 3. Outil I3S de géotraitement pour transformer la gdb et *Scene Layer Package I3S* (SLPK), qu'on peut directement fournir à ArcGIS online.

- Contenu I3S conçu pour pouvoir être consommé par 3DPS, puis servi à des clients non-Esri. Succès avec Cesium.

COMPARAISON ENTRE LES DEUX FORMATS

3D Tiles vs I3S

PERFORMANCES

- Structures de données cohérentes : conversion possible entre les différents formats 3D pour un affichage performant côté client.
- Les aspects mémoire ont été pris en compte, et, pourvu que les données aient été correctement optimisées, les deux standards permettent de visualiser des volumétries de données très importantes, de types hétérogènes.
- Ces formats ayant été développés et massivement utilisés depuis bien avant leur avancement comme standards communautaires, les principaux points de blocage potentiels ne proviennent pas de leurs performances, mais de leurs implémentations.

ADAPTABILITÉ

- Possibilité d'adapter les ADE CityGML dans 3DTiles.
- Les ADE sont des extensions qui ne font pas partie du standard originel, mais sont des implémentations de fonctionnalités spécifiques à un domaine (modélisation des informations de bruit, de consommation d'énergie…).
- Pour les adapter au format 3DTiles, il est nécessaire d'en connaitre le schéma XML complet.

- Autre aspect potentiellement intéressant, la notion de temps : 3DTiles permet de renseigner les paramètres temporels (dates de validité par ex.), pas I3S.

LIMITATIONS

- Création des données conditionnée à des outils propriétaires.
  - Calcul de l'erreur géométrique non inclus dans la spec alors que c'est un point central de l'implémentation, pour 3D Tiles
  - Mainmise d'Esri sur les outils I3S.
  - Limites dans l'optimisation, le tuilage…
- Cesium Ion pour 3D Tiles : hébergement, transformation, mais surtout outils d'analyse de performances, calculs de visibilité, etc. Système d'abonnement : probabilité que tous les traitements ne soient pas automatisés.

INSTITUT NATIONAL
DE L'INFORMATION
GÉOGRAPHIQUE
ET FORESTIÈRE

© IGN

MERCI DE VOTRE ATTENTION

eden.ign.fr/20052019