



Observatoire astronomique  
de Strasbourg



31 mai au 5 août

# Visualisation progressive de données astronomiques en 3D dans un navigateur

**Auteur :**  
Nicolas Adam

**Responsables :**  
André Schaaff  
Pierre Tellier

# Sommaire

- L'Observatoire Astronomique de Strasbourg
- Ma mission
- Les outils utilisés
- Les résultats
- Les problèmes rencontrés
- Le bilan du stage

# L'Observatoire Astronomique de Strasbourg

- **Fondé en 1881**
- **Trois équipes :**
  - L'équipe Hautes Énergies
  - L'équipe Galaxies
  - Le Centre de Données astronomiques de Strasbourg

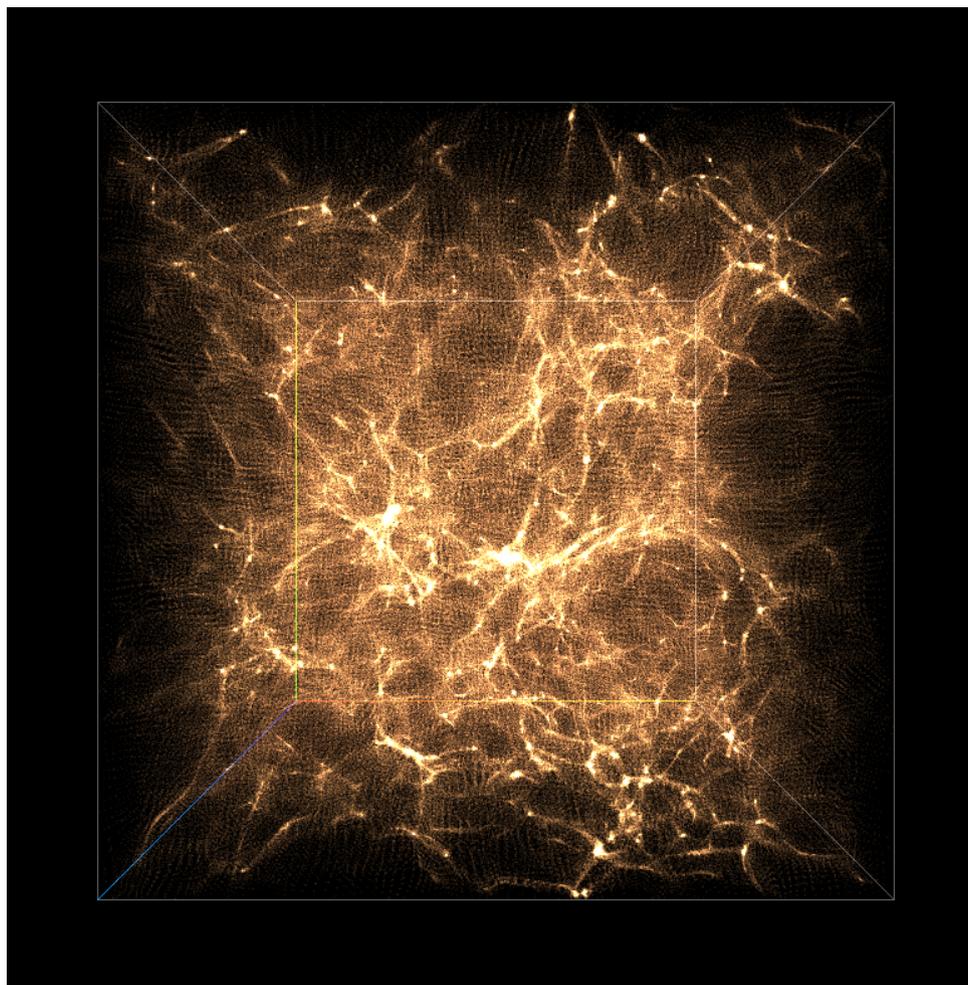
# L'Observatoire Astronomique de Strasbourg

- **Plusieurs missions :**
  - Recherche
  - Enseignement
    - Spécialité astrophysique (master)
  - Service d'observation
    - Développement d'outils et de services pour la communauté :  
Simbad, VizieR et Aladin
  - Diffusion des connaissances
    - Fête de la science, Nuit des étoiles, etc.

# Ma mission

- **Application de visualisation**
  - Déjà existante
    - Affichage et manipulation de données
  - Mais limitée (RAM, carte graphique)
    - Gère 2 millions de points (80 Mo de données)
  - Doit pouvoir gérer de gros volumes de données
    - De l'ordre du téraoctet (4096 au cube)

# Ma mission



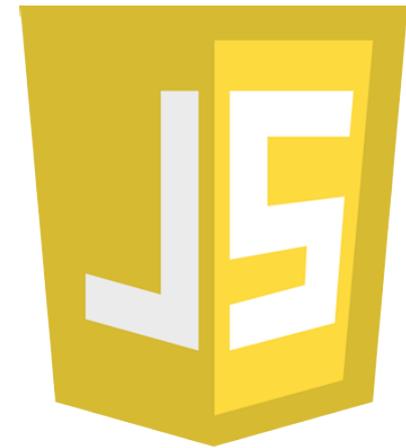
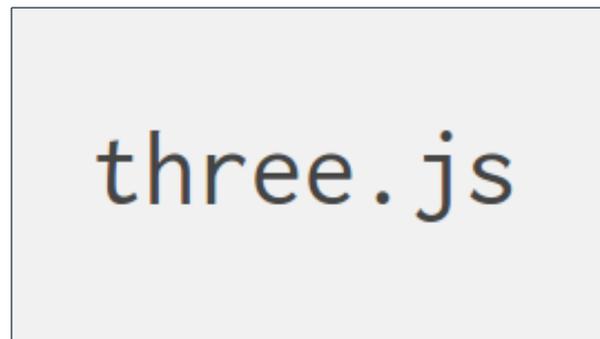
2 millions de points

# Les outils utilisés



mongoDB

CSS



# Les outils utilisés



**Robomongo**

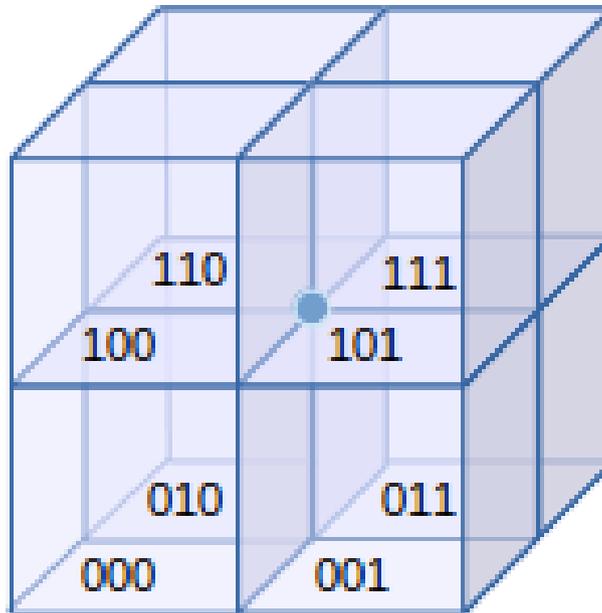


**GitLab**

# Les résultats – côté serveur

- **Création de l'octree**
  - Partitionnement récursif d'un espace tridimensionnel
  - Indexation des octants
    - Identifiant unique pour chaque octant
    - Position représentée par une suite de nombres binaires
  - Contient des données dégradées

# Les résultats – côté serveur



Indexation du premier niveau d'un octree

# Les résultats – côté serveur

- **Insertion dans mongoDB**
  - Chaque document représente un nœud ou une feuille
  - Pas de référence entre les documents
- **Requête sur la base de données**
  - Requête par regexp
    - `db.collection.find({id : /^011001/}, {points : true})`
  - Renvoie tous les points d'un nœud donné

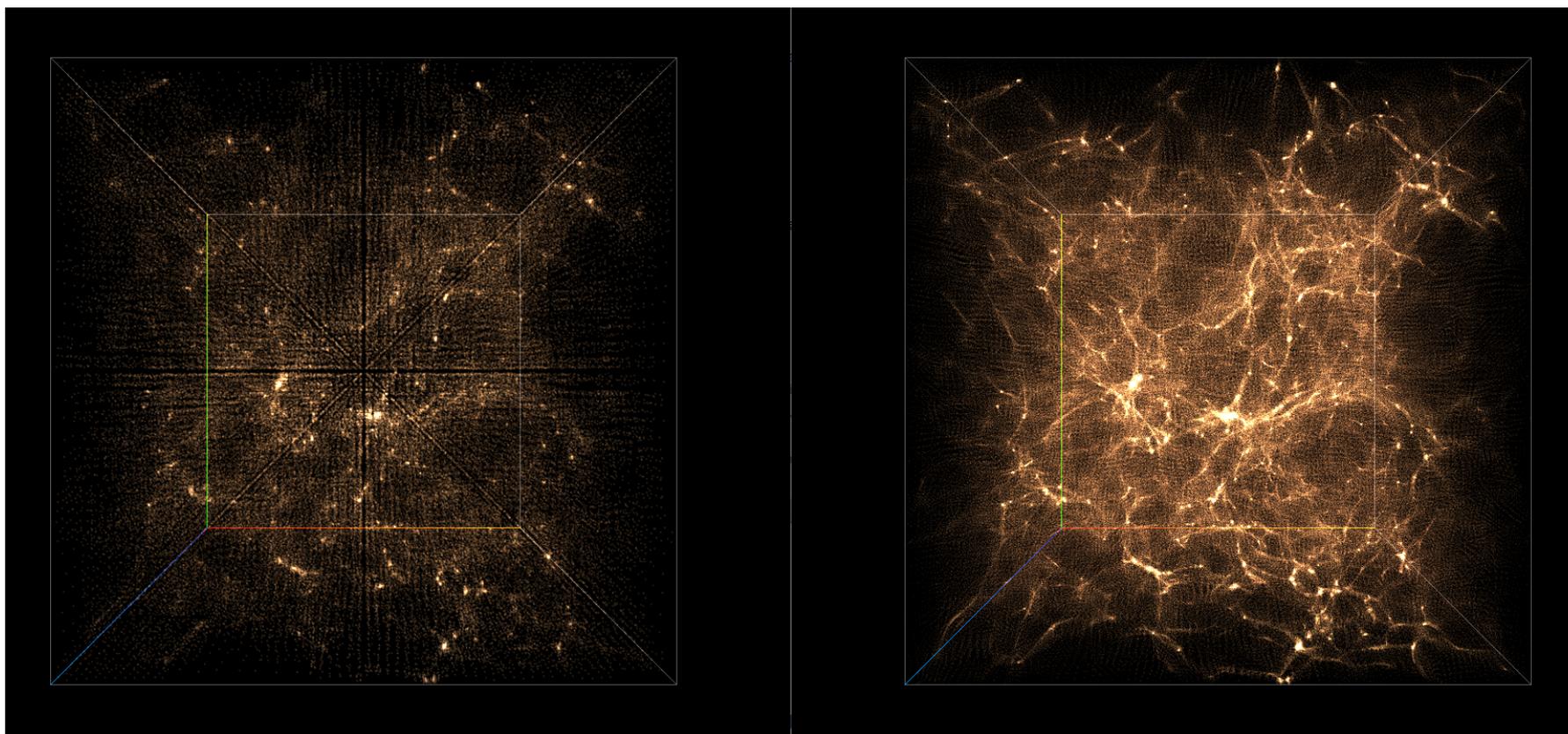
# Les résultats – côté client

- **Requêtes AJAX**
  - Chargement des données
- **Octants chargés lors du déplacement de la caméra**
  - L'octant courant plus les octants voisins
  - 27 requêtes, une par octant chargé

# Les problèmes rencontrés

- **Les données finales**
  - Problème de création de l'octree
    - Manque de RAM
    - Résolu par l'utilisation du *File System*
  - Problème dans l'affichage des points
    - Chargement peu performant
    - Choix d'une profondeur de 7 pour les octants
    - Choix du mode de dégradation des données

# Les problèmes rencontrés



Données dégradées à gauche, non dégradées à droite

# Les problèmes rencontrés

- **Les performances des requêtes**
  - *Wrap* des documents par Mongoose
    - Résolution par l'ajout de la méthode *lean*
    - Pour 2 millions de points :
      - Erreur après 30 secondes sans *lean*
      - Valeurs retournées en 5 secondes avec *lean*
  - Optimisation avec les requêtes d'agrégation
- **Code initial compliqué**
- **Gestion de la RAM par node.js**

# Le bilan

- **Stage très intéressant et instructif**
- **Technologies actuelles**
- **Partie réflexion importante**
  - Choix des technologies, des méthodes
  - Résolution de problèmes
- **Grande autonomie**
  - Mais réunions avec intervenants du projet

**Merci de votre attention !**

Avez vous des questions ?