



HAL
open science

Visualisation interactive de données hétérogènes pour l'amélioration des dépenses énergétiques du bâtiment

Benoit Lange

► **To cite this version:**

Benoit Lange. Visualisation interactive de données hétérogènes pour l'amélioration des dépenses énergétiques du bâtiment. Modélisation et simulation. Université Montpellier 2, 2012. Français. NNT: . tel-01108347

HAL Id: tel-01108347

<https://hal.science/tel-01108347>

Submitted on 22 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

ACADÉMIE DE MONTPELLIER

UNIVERSITÉ MONTPELLIER II

Sciences et Techniques du Languedoc

THÈSE

présentée au Laboratoire d'Informatique de Robotique
et de Microélectronique de Montpellier pour
obtenir le diplôme de doctorat

Spécialité : **Informatique**
Formation Doctorale : **Informatique**
École Doctorale : **Information, Structures, Systèmes**

**Visualisation interactive de données hétérogènes pour
l'amélioration des dépenses énergétiques du bâtiment**

par

Benoit LANGE

Version du 16 octobre 2012

Rapporteurs

M. Gilles GESQUIÈRE, Professeur.....LIRIS, Université Lumière Lyon 2, Lyon

M. Jean-Pierre JESSEL, Professeur IRIT, Université Paul Sabatier, Toulouse

Directeur de thèse

M. William PUECH, Professeur.....LIRMM, Université Montpellier II, Montpellier

Co-Directeur de thèse

Mme. Nancy RODRIGUEZ, Maître de conférence LIRMM, Université Montpellier II, Montpellier

Examineurs

M. Bruno JOBARD, Maître de conférence..... LIUPPA, Université de Pau et des Pays de l'Adour, Pau

Mme. Anne LAURENT, Professeur LIRMM, Université Montpellier II, Montpellier

Invité

M. Xavier VASQUES, Manager de l'Innovation Lab IBM, Montpellier

Table des matières

Table des matières	i
1 Introduction	1
1.1 Le projet RIDER	2
1.2 Positionnement de la thèse	3
1.3 Plan du manuscrit	4
I Etat de l'art	5
2 Notions préliminaires	7
2.1 La modélisation d'objets 3D	8
2.1.1 Nuage de points	8
2.1.2 Maillages triangulés	10
2.1.3 Voxel	11
2.1.4 La triangulation (Tesselation)	12
2.1.5 La méthode de Voronoï	15
2.2 Les niveaux de détails	17
2.3 Les architectures parallèles	23
3 Visualisation de données	25
3.1 Introduction	25
3.2 Définitions relatives aux données	26
3.3 Visualisation adaptée à la problématique du bâtiment	27
3.4 Visualisation de données multi-dimensionnelles	31

3.4.1	Visualisation géométrique	31
3.4.2	Visualisation iconique	37
3.4.3	Visualisation hiérarchique	40
3.5	Les moteurs 3D	42
3.6	Synthèse	45
4	Visualisation analytique de données numériques	47
4.1	Introduction	47
4.2	La gestion de données	48
4.3	L'interaction appliquée à la donnée	51
4.3.1	L'interaction matérielle et logicielle	51
4.3.2	Collaboration	53
4.4	Des visualisations multi-vues	55
4.4.1	Des outils d'analyse guidée par l'utilisateur	55
4.4.2	Des visualisations 2D	58
4.4.3	Des visualisations 3D	62
4.5	Synthèse	67
II	Contributions	69
5	Modélisation de bâtiment	71
5.1	Introduction	71
5.2	Paradigme client / serveur	72
5.3	Modélisation de l'espace basée sur des particules	73
5.3.1	Modèle en grille régulière	76
5.3.2	Modèle concentrique	77
5.3.3	Modèle multi-résolutions	79
5.4	Interpolation basée capteur	80
5.4.1	Triangulation de Delaunay	81
5.4.2	Cellules de Voronoï	82
5.4.3	Pondération des particules	83
5.5	Conclusion	84
6	Visualisations mises en place	87
6.1	Introduction	87
6.2	Visualisations issues de l'état de l'art	88
6.2.1	Les graphiques	90
6.2.2	Visualisation par lignes brisées	90
6.2.3	Les graphes radar	91
6.2.4	La visualisation par multiples nuages de points	92

6.2.5	La visualisation par glyphe	92
6.3	Nos visualisations 3D	93
6.3.1	Visualisation basée seulement sur une information spatiale	93
6.3.2	Visualisation volumique	94
6.4	Solutions de multi-résolutions	101
6.4.1	Multi-résolutions pour la transmission d'informations	101
6.4.2	Multi-résolutions pour la visualisation	102
6.5	Outil de fouille de données	104
6.5.1	Modèle d'interaction	104
6.5.2	Extraction de contenu à partir de données réelles	106
6.6	Synthèse	108
7	Résultats et expérimentations	111
7.1	Introduction	111
7.2	Optimisation énergétique pour l'application serveur	112
7.2.1	Mesure du temps d'exécution de notre algorithme	113
7.2.2	Energie dépensée pour le calcul	117
7.3	Les données d'un centre de calcul : Green Data Center (GDC)	119
7.3.1	Analyse de la pièce haute densité	122
7.3.2	Analyse de la pièce basse densité	123
7.4	Les données d'un bâtiment tertiaire	124
7.5	Synthèse	130
8	Conclusions et perspectives	133
8.1	Résumé de nos contributions	133
8.2	Difficultés rencontrées	135
8.3	Limitations	136
8.4	Perspectives	136
	Liste des publications	139
	Annexes	141
	Les graphes	141
	Bibliographie	145

Introduction

Il y a 10 sortes de gens sur Terre :
ceux qui connaissent le binaire,
et les autres.

ANONYME

Sommaire

1.1	Le projet RIDER	2
1.2	Positionnement de la thèse	3
1.3	Plan du manuscrit	4

Le Grenelle de l'environnement¹ s'est fixé comme objectif de réduire la consommation d'énergie de notre habitat de plus d'un tiers à l'horizon 2020. En effet, 43% de la consommation énergétique en France est aujourd'hui issue des bureaux, des usines, des commerces et des logements.

Pour y parvenir, 2 axes ont été choisis concernant les bâtiments :

- un programme de rupture technologique sur le bâtiment neuf,
- un chantier de rénovation énergétique radicale des bâtiments existants.

Dans ce contexte, l'objectif principal du projet RIDER² (Research for IT Driven EneRgy Efficiency) est de développer une plateforme intelligente pour la gestion multi-échelles et

1. <http://www.legrenelle-environnement.fr>
2. www.rider-project.com

multi-standards des énergies du bâtiment. Les travaux de cette thèse se déroulent dans le cadre du projet RIDER. Nous avons conçu et développé un système de visualisation de données multidimensionnelles provenant de capteurs et de système de gestion du bâtiment. En effet, nous croyons que permettre aux utilisateurs de visualiser comment l'énergie est utilisée dans le bâtiment, peut les aider à maintenir leur consommation et à prendre conscience de l'importance d'un usage réfléchi de l'énergie.

1.1 Le projet RIDER

Le projet RIDER a pour objectif de développer un système d'information innovant permettant d'optimiser l'efficacité énergétique d'un bâtiment ou groupe de bâtiments. Ce système doit fournir les fonctions nécessaires à l'obtention d'un niveau d'optimisation supérieur des systèmes de gestion de bâtiment actuels.

Le projet RIDER est mené par un consortium constitué de l'Université de Montpellier 2 – Sciences et Technologies du Languedoc (UM2) au travers des laboratoires LIRMM et IES, du laboratoire PROMES de l'Université de Perpignan Via Domitia (UPVD), ainsi que des partenaires industriels IBM, EDF, Cofely GDF-Suez, Pyrescom, Coronis et leurs partenaires sous-traitants : ASA et Enoleo. Ce projet est également soutenu par la région Languedoc-Roussillon, la ville de Perpignan et l'agglomération de Montpellier, ainsi que par des équipementiers et exploitants de bâtiments publics et privés. Afin d'évaluer les résultats obtenus au travers de ce projet, trois sites pilotes ont été choisis : le bâtiment IBM Montpellier, le théâtre de l'Archipel à Perpignan et le bâtiment Capdeville de Montpellier.

RIDER est conçu autour d'une architecture orientée service. Elle est composée de deux éléments principaux et de divers composants optionnels. Les éléments principaux sont le système de gestion de bâtiment (BMS pour Building Management System) et le noyau RIDER. La solution de BMS est utilisée pour interconnecter les différentes solutions pour gérer le bâtiment au projet RIDER. Les éléments interconnectés peuvent être par exemple des capteurs ou des vannes. Le noyau RIDER, quant à lui, est chargé d'agréger ces différentes données utiles et d'appliquer une série de règles de gestion du bâtiment. Ce composant utilise un entrepôt de données et un moteur de règles. L'entrepôt de données utilise une représentation particulière du bâtiment propre au système RIDER. Des composants peuvent étendre les fonctionnalités du système, par exemple en proposant de nouvelles informations pour améliorer le modèle global de fonctionnement des données. La figure 1.1 est une représentation schématique de RIDER. Certains modules peuvent seulement fournir des informations (par exemple la météo) et ne pas traiter les données fournies par le noyau RIDER.

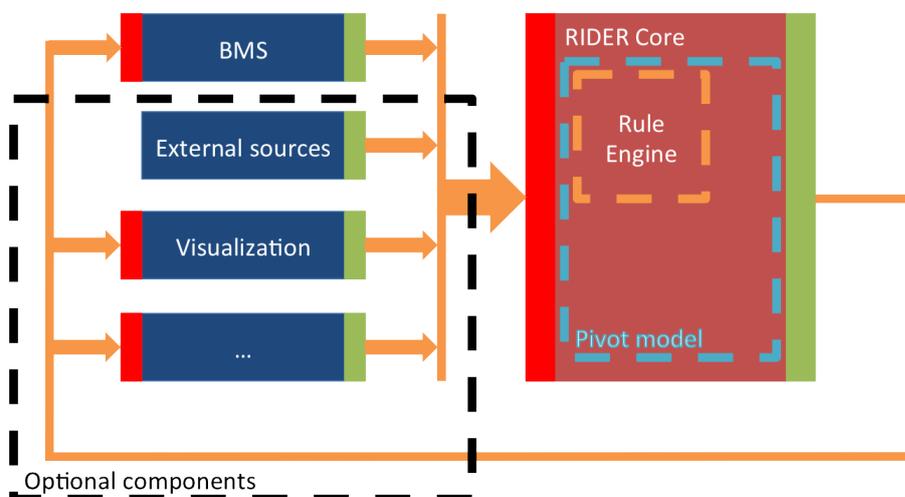


FIGURE 1.1 – Schéma de représentation de RIDER.

Les travaux que nous avons menés dans le cadre de cette thèse concernent la partie visualisation. En effet, les systèmes de gestion du bâtiment (BMS) fournissent des visualisations 2D simples basées sur des courbes, graphiques camembert ou cartes. De plus, ces outils ne sont utilisés que pour ajuster les différents paramètres ou pour fournir un retour pour le gestionnaire du bâtiment. L'utilisateur est passif devant son système et subit les différentes règles mises en place sans pouvoir interagir et alimenter le système. Aucune interface n'est spécifiquement dédiée à la mise à jour ou l'amélioration de la représentation interne des données. Un bâtiment peut-être considéré comme une entité "vivante", qui évolue en permanence. Par exemple, les immeubles de bureaux peuvent voir leur topologie interne évoluer dans le temps, notamment au travers de changements des cloisons. Nous souhaitons donc donner à l'utilisateur la possibilité d'explorer les données en 2D et 3D pour améliorer sa connaissance du système et lui permettre, en outre, de l'enrichir grâce à ses "découvertes" et réflexions.

1.2 Positionnement de la thèse

Dans le cadre de ces travaux, nous nous sommes intéressés à différents verrous scientifiques notamment autour de :

- la gestion des données de grande taille,
- la visualisation interactive,
- les outils d'analyse par la visualisation.

La problématique de gestion de données complexes et hétérogènes dans le cadre du bâtiment est la suivante : un bâtiment est géré par un ensemble de capteurs de différents types. Les capteurs produisent des données de manière régulière mais à des fréquences différentes en fonction du type de données. Si la fréquence de rafraîchissement des données

est élevée, il est nécessaire d'élaborer des méthodes de traitement efficaces pour pouvoir les traiter et visualiser en temps-réel. De plus dans le cadre du projet RIDER, la gestion de données multi-bâtiments est à prendre en compte. Il est donc nécessaire d'élaborer des stratégies efficaces de gestion de ces données.

Le second objectif de ces travaux de recherche est orienté autour des techniques de visualisation. Nous nous attacherons à étudier des méthodes de visualisation utilisables dans le cadre d'un usage grand public, avec une faible courbe d'apprentissage. Les méthodes de visualisation que nous proposons, en plus des techniques classiques 2D, reposent sur une représentation volumique des données.

Enfin, notre système doit permettre de mettre à jour le modèle de données de manière efficace, soit en réalisant des annotations, soit en proposant de nouvelles informations à celui-ci.

1.3 Plan du manuscrit

Ce manuscrit est structuré en deux parties autour de l'état de l'art (chapitres 2 à 4) et des contributions (chapitres 5 à 7).

Le chapitre 2 expose les différentes notions de base nécessaires à la compréhension de ces travaux. Nous allons introduire différents concepts autour de la modélisation 3D et les traitements relatifs à ce type de données.

Dans le chapitre 3, nous présentons différentes méthodes de visualisation de données, notamment la problématique relative aux bâtiments et les méthodes de visualisation de données hétérogènes.

Dans le chapitre 4, nous introduisons la problématique de la gestion de données, ainsi que les méthodes de visualisation interactive et multi-vues.

Le chapitre 5 présente le modèle que nous avons mis en place au travers d'une architecture client / serveur. Nous présentons également les solutions de modélisation autour des capteurs du bâtiment.

Le chapitre 6 décrit les différentes visualisations mises en place dans le cadre de ces travaux : les visualisations 2D, les visualisations 3D multi-résolution et l'outil de fouille de données.

Enfin le chapitre 7 présente les différents résultats que nous avons obtenus au travers de deux jeux de données réelles.

Dans le chapitre 8, nous clôturons ce manuscrit en présentant une conclusion générale à ces travaux ainsi que les différentes perspectives envisagées.

Première partie

Etat de l'art

Notions préliminaires

Nous ne pouvons pas prédire où nous conduira la Révolution Informatique. Tout ce que nous savons avec certitude, c'est que, quand on y sera enfin, on n'aura pas assez de RAM.

D. BARRY

Sommaire

2.1	La modélisation d'objets 3D	8
2.2	Les niveaux de détails	17
2.3	Les architectures parallèles	23

Ce chapitre est consacré à présenter divers concepts nécessaires à la compréhension de ces travaux de recherche. Nous allons ainsi introduire dans la section 2.1 les différentes solutions de modélisation d'objets 3D que nous avons étudiées. La section 2.2 traite de la réduction de complexité des objets afin de permettre leur utilisation massive sur des systèmes informatiques. Enfin, la section 2.3 traite de l'utilisation des architectures parallèles servant à réduire le temps de calcul lié aux traitements et à la visualisation des données.

2.1 La modélisation d'objets 3D

En informatique graphique, l'environnement 3D représenté est communément appelé "scène". La scène est composée d'un ensemble d'objets 3D appelés modèles. Ils peuvent être représentés de différentes manières et possèdent des transformations permettant de les positionner dans le repère global de la scène (rotation, position, échelle). Une des plus grandes barrières à la visualisation fidèle des objets est l'inévitable discrétisation des modèles 3D. En effet, les données continues n'étant pas supportées par les périphériques graphiques, il a été nécessaire de développer des solutions de modélisation discrète. Dans cette section, nous présentons les trois principales approches utilisées pour modéliser des objets 3D puis les deux principales méthodes de transformation d'un type de modèle vers un autre.

2.1.1 Nuage de points

Un objet 3D peut être modélisé par un nuage de points constitué d'une liste de points définis par leurs coordonnées cartésiennes. Cette méthode permet de représenter les surfaces et les volumes. Dans la figure 2.1, nous pouvons voir une représentation du lapin de Stanford¹, à base de nuage de points. Ce modèle est une représentation surfacique, les différents volumes internes (tissus, squelette, etc...) ne sont pas représentés. Pour chaque point, il est possible de définir d'autres attributs, en plus de sa position dans l'espace tels que la normale, la couleur, la vitesse, la taille, ... Un objet 3D peut être représenté par D l'ensemble des sommets de son nuage de points, avec $D = \{d_1, \dots, d_n\}$. $\forall i \in \{1, \dots, n\}$, d_i est un sommet représenté par l'ensemble $\{c_i, e_0, e_1, \dots\}$, où $c_i = \{x, y, z\}$ définit les coordonnées spatiales du point avec $x, y, z \in \mathbb{R}^+$ et chaque e_j un attribut supplémentaire (par exemple la couleur).

1. <http://graphics.stanford.edu/data/3Dscanrep/>

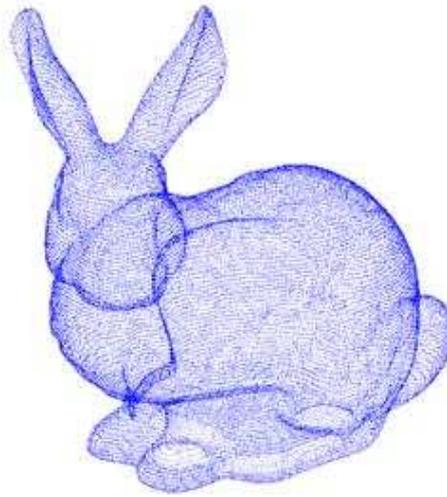


FIGURE 2.1 – Vue surfacique d'un modèle 3D à partir d'un nuage de points.

Systèmes à particules

Parmi les modélisations par nuage de points, les systèmes à base de particules sont des outils très utilisés du domaine informatique graphique, pour simuler des comportements et phénomènes physiques. Les particules sont de petits éléments avec des attributs configurables comme leur localisation, leur couleur, leur vitesse, leur voisinage. A chaque itération des algorithmes de simulation, il est nécessaire de calculer le déplacement des particules et leur impact sur les particules environnantes. Trois grandes classes de systèmes à particules sont utilisées : les particules statiques, les particules pré-calculées et les particules dynamiques.

Les particules dynamiques sont des éléments qui se déplacent dans l'espace et dont les trajectoires dépendent de la vitesse, de l'accélération et de la direction qui leur sont attribuées tout au long de l'exécution. Ce type de particules est largement utilisé pour simuler des comportements de fluides, de fumée ou de feu. Un exemple de représentation d'un fluide est présenté en figure 2.2. Ici, les particules sont utilisées pour simuler les différents comportements des éléments qui composent l'eau, notamment au travers d'une dynamique produite sur les particules.

La dernière approche utilise des particules statiques. Cette solution permet de visualiser de grandes quantités de données sous forme de nuage de points, les particules étant colorées afin de créer une certaine dynamique. Le fait de ne pas calculer les différents déplacements permet de réduire le coût du calcul des collisions.



(a) Rendu d'une simulation de fluide par particules.



(b) Rendu de la même méthode après tessellation de l'eau (voir section 2.1.4).

FIGURE 2.2 – Simulation d'un fluide basée sur une approche à particules

2.1.2 Maillages triangulés

Cette deuxième approche pour la représentation d'objets 3D est la plus répandue en informatique graphique pour créer des modèles. La primitive élémentaire des maillages triangulés est le triangle. L'avantage de choisir le triangle au lieu d'un polygone à plus de 3 cotés est qu'il est situé sur un plan dans l'espace. Les triangles du modèle sont tous reliés entre eux afin de produire un maillage, représentant la surface d'un objet. La figure 2.3 présente une représentation du lapin de Stanford sous la forme d'un maillage triangulé. Comme présenté sur cette figure, la taille des triangles peut être variable. Pour colorer ce type d'objets, deux solutions sont couramment utilisées : soit les couleurs sont appliquées grâce à une cartographie issue d'une image (l'image est appelée texture), soit en colorant directement les sommets. Pour définir un maillage formellement, nous utilisons le fait qu'il soit composé de sommets reliés ensemble par des arêtes. Soit M un maillage, $M = (V, E)$, avec $V = \{v_1, \dots, v_n\}$ un ensemble de sommets, les coordonnées de chaque sommet v_i sont $c = \{x, y, z\}$ avec $x, y, z \in \mathbb{R}^+$. E représente l'ensemble des arêtes.

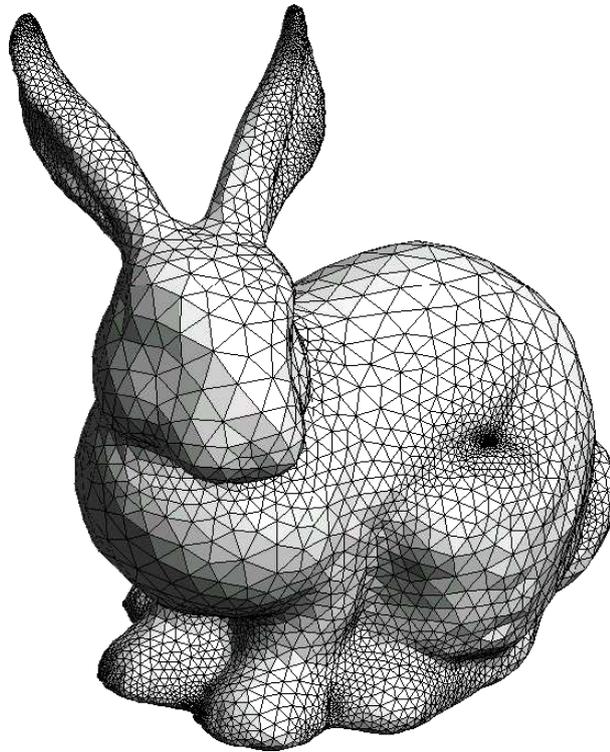
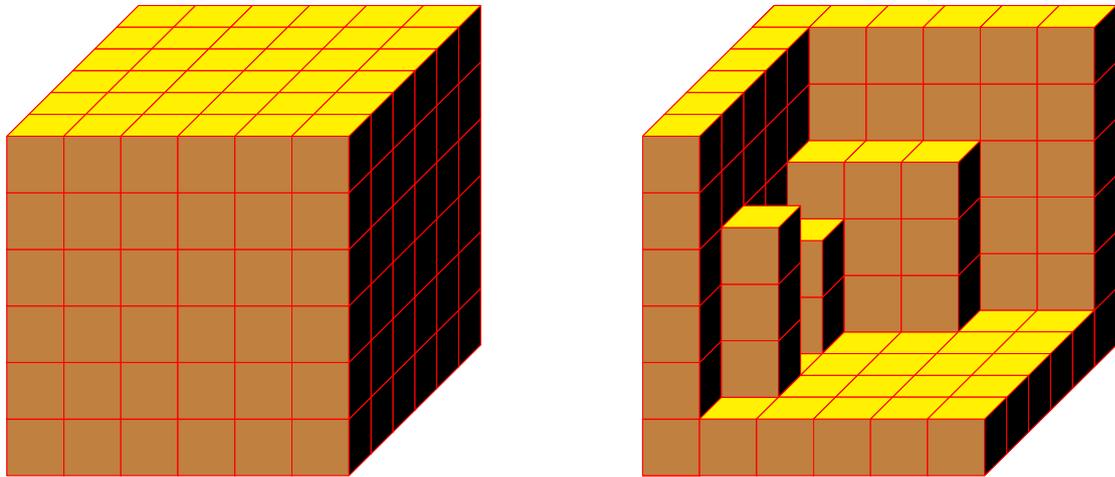


FIGURE 2.3 – Vue surfacique d'un modèle 3D à partir d'un maillage polygonal.

2.1.3 Voxel

Cette méthode de représentation d'objet 3D est celle basée sur des voxels. Le terme voxel signifie VOLumetric piXEL; ainsi, comme pour une image, la méthode par voxels utilise une représentation discrète et régulière des objets 3D. La primitive de base est donc un parallélépipède rectangle, le plus souvent un petit cube. Afin de représenter un objet 3D, il suffit simplement d'empiler ces cubes comme dans un jeu de construction. Cette méthode produit donc un objet avec un aspect crénelé dont la finesse dépend de la résolution de la grille de voxels. Un objet 3D est représenté par $F = \{f_0, \dots, f_n\}$, l'ensemble de voxels du modèle. Un voxel f_i est représenté par $f_i = \{c_i, t_i e_0, e_1, \dots\}$, où $c_i = \{x, y, z\}$ représente les coordonnées spatiales du voxel avec $x, y, z \in \mathbb{R}^+$, $t_i = \{h, l, L\}$ représente les différentes dimensions du voxel avec $h, l, L \in \mathbb{R}^+$ et chaque e_j un attribut supplémentaire (par exemple la couleur). Ce type de primitive peut être utilisé pour la représentation d'objets surfaciques mais également volumiques. Un exemple de représentation surfacique d'un cube est illustré figure 2.4(a); la figure 2.4(b) présente les volumes intérieurs de ce modèle. Grâce à cette méthode, il est possible de produire des rendus de modèles plus complexes notamment dans le domaine médical pour représenter le corps humain, par exemple.



(a) Vue surfacique d'un modèle 3D à partir de voxels.

(b) Vue volumique d'un modèle 3D à partir de voxels.

FIGURE 2.4 – Volume à base de Voxel.

2.1.4 La triangulation (Tesselation)

Comme mentionné précédemment, les triangles représentent la primitive la plus utilisée par les dispositifs graphiques. Il est alors nécessaire de gérer un maillage à partir de n'importe quelle représentation (courbes implicites, points, voxels). Nous nous intéressons particulièrement à la génération du maillage à partir d'un ensemble de points.

Il existe différentes solutions pour recréer un graphe à partir de sommets. Les triangulations aléatoires, par exemple, lient de manière arbitraire les sommets les uns aux autres, tout en évitant les chevauchements d'arêtes. Les méthodes plus classiques créent un maillage triangulé de manière aléatoire, puis calculent le centre de gravité de chaque triangle et enfin vérifient qu'aucun sommet n'est inclus dans ce triangle. Un exemple de triangulation aléatoire est présenté en figure 2.5. En appliquant une autre triangulation, il aurait été possible d'obtenir un tout autre résultat, les contraintes à respecter étant que tous les sommets doivent être reliés, qu'aucun couple d'arêtes ne doivent se croiser et que toutes les primitives résultantes soient des triangles. La triangulation de Delaunay présentée ci-dessous permet, elle, de réaliser une triangulation stable, c'est à dire que pour un jeu de données précis le nombre d'arêtes et de triangles résultants de la triangulation reste invariable.

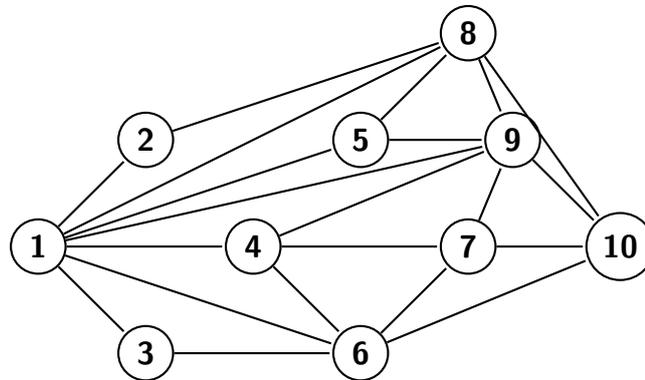


FIGURE 2.5 – Triangulation aléatoire d'une ensemble de sommets.

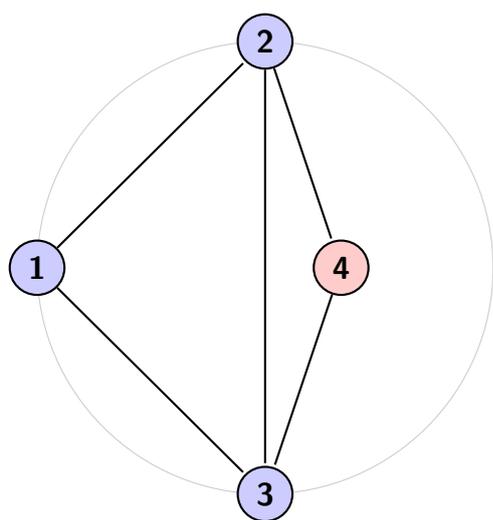
Triangulation de Delaunay

La triangulation de Delaunay est présentée en détail par Avis et Bhattacharya, [Avis et Bhattacharya, 1983]. Le but de cette méthode est de produire un maillage particulier à partir d'un nuage de points. Elle est utilisée depuis de nombreuses années, la première publication scientifique y faisant référence remonte à 1934, [Delaunay, 1934]. Pour réaliser une triangulation, la méthode définit une contrainte : pour chaque triangle formé, le cercle circonscrit construit à partir des trois sommets ne doit contenir aucun autre sommet. La figure 2.6(a) présente le résultat d'une mauvaise triangulation de Delaunay, un des sommets est à l'intérieur du cercle circonscrit, tandis qu'en figure 2.6(b), la triangulation fournit un résultat satisfaisant et aucun triangle ne contient de sommet intérieur. Enfin, la figure 2.6(c) présente le résultat de la triangulation de Delaunay sur l'exemple de la figure 2.5.

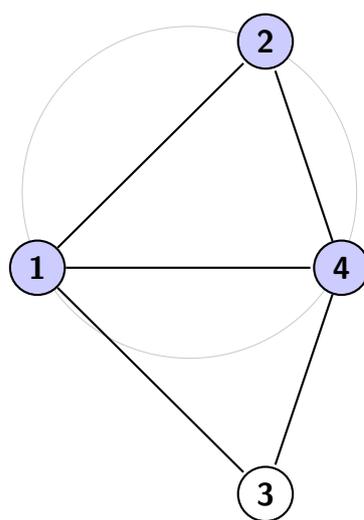
La triangulation de Delaunay est principalement utilisée dans un contexte 2D, mais il est également possible d'étendre le résultat dans un contexte n D. Les cercles sont alors remplacés par des hyper-sphères. Différentes implémentations de la triangulation de Delaunay ont été proposées. Cette solution de triangulation peut être implémentée avec une complexité de $O(n \times \log(n))$ avec n le nombre d'arêtes de la triangulation.

La méthode de Delaunay utilise le principe de tri des angles afin de créer une triangulation de Delaunay. Cette solution permet d'obtenir un meilleur résultat et d'utiliser moins d'espace mémoire que les solutions aléatoires. De plus, la solution proposée est capable de manipuler des nuages de points de différentes tailles.

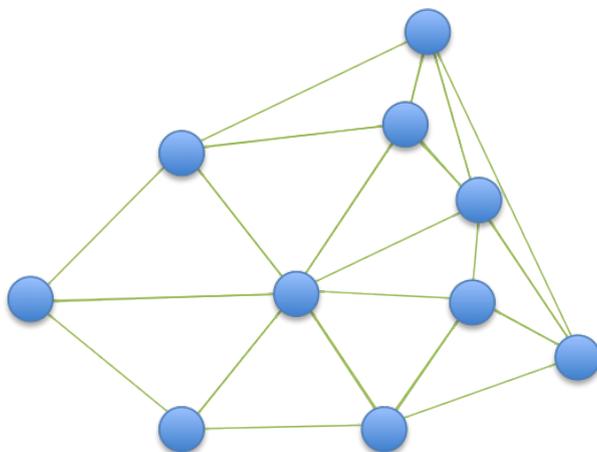
D'autres travaux de recherche se sont intéressés à proposer des méthodes de triangulation efficaces. Par exemple, la librairie CGAL (Computational Geometry Algorithms Library) développée par Fabri *et al.* permet de réaliser une triangulation de Delaunay sur n'importe quel ensemble de sommets, qu'il soit en 2D ou en 3D, [Fabri *et al.*, 1996].



(a) Mauvaise triangulation, le sommet 4 est à l'intérieur du cercle circonscrit.



(b) Triangulation exemplaire.



(c) Triangulation de Delaunay d'une la liste de sommets.

FIGURE 2.6 – Exemple pour la triangulation de Delaunay

Un exemple d'utilisation de la triangulation de Delaunay est illustré dans les outils proposés par Barber *et al.*, [Barber *et al.*, 1996], où les auteurs présentent une solution de calcul d'enveloppe convexe appelé QuickHull. Un objet géométrique est dit convexe lorsque, chaque fois qu'on y prend deux points, le segment qui les joint y est entièrement contenu. Une enveloppe convexe est un ensemble convexe de taille minimale tel que tous les sommets de l'ensemble de départ soit inclus à l'intérieur. Un exemple d'enveloppe convexe est représenté en figure 2.6(c) par la chaîne suivante : $C = \{1, 3, 6, 10, 8, 2\}$. Un exemple d'objet produit par QuickHull est donné en figure 2.7. Il s'agit d'un cube produit par un système de placement par intervalle régulier.

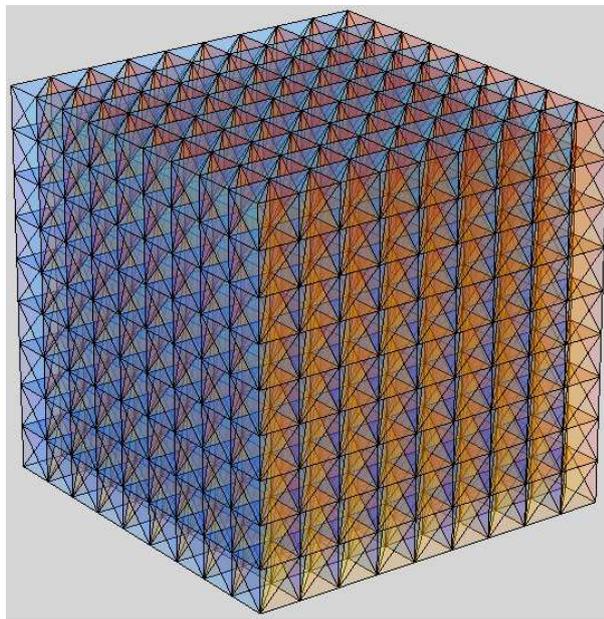


FIGURE 2.7 – Exemple de rendu produit par Qhull (en utilisant la triangulation de Delaunay).

2.1.5 La méthode de Voronoï

La solution duale de la triangulation de Delaunay, ou cellules de Voronoï, permet de subdiviser l'espace d'une manière particulière. Avis et Bhattacharya proposent leur formalisation de cette méthode dans [Avis et Bhattacharya, 1983]. La méthode s'appuie sur un ensemble de sommets appelés sites. L'algorithme de Voronoï cherche à optimiser la segmentation de l'espace par rapport à ces différents sites. Ainsi, chaque point des différentes régions sera assigné au site le plus proche. La figure 2.8 représente la partition de Voronoï pour l'ensemble de sommets que nous avons présenté figure 2.5. Différentes propriétés sont à remarquer :

- chaque arête d'une cellule est en fait une des médiatrices du triangle de Delaunay lui correspondant,
- les sites des cellules de Voronoï correspondent exactement au centre des cercles circonscrits relatifs aux triangles de Delaunay.

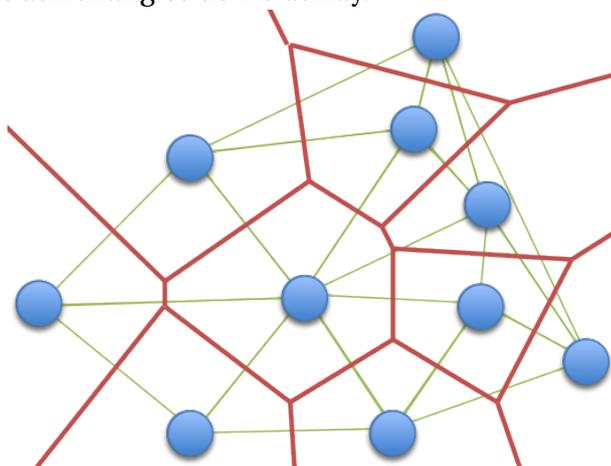


FIGURE 2.8 – Extraction des cellules de Voronoï pour les sommets de la figure 2.6(c), en noir le diagramme de Voronoï, et en orange la triangulation de Delaunay.

Différentes implémentations de cette méthode ont été proposées. Aurenhammer présente une étude sur différentes solutions existantes pour l'extraction de cellules de Voronoï, notamment les approches de planification de mouvement et de clustering, [Aurenhammer, 1991]. Les cellules de Voronoï étant souvent utilisées dans des environnements 2D, l'auteur va plus loin en présentant différentes approches développées pour nD.

Rycroft présente une solution de génération de cellules de Voronoï en 3D, [Rycroft, 2009]. Voro++ permet de calculer l'ensemble des cellules des Voronoï pour un ensemble de points 3D. L'avantage principal de cette solution réside dans le fait que l'espace à diviser n'a pas besoin d'être borné contrairement aux solutions telles que celle de [Barber *et al.*, 1996]. La figure 2.9 présente une représentation de l'algorithme Voro++. Ici, les différentes graines des cellules de Voronoï sont présentées par des sphères métalliques.

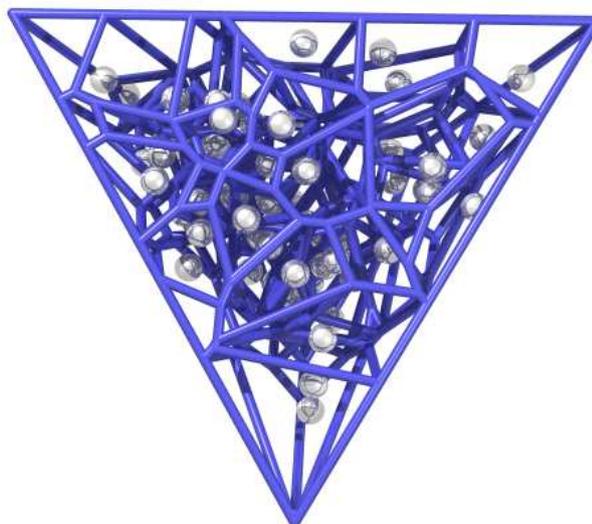


FIGURE 2.9 – Exemple de rendu de la solution Voro++.

Les cellules de Voronoï sont également utilisées en visualisation d'information. Par exemple Balzer et Deussen, [Balzer et Deussen, 2005], proposent une visualisation de données hiérarchiques qui ne sont pas restreinte à l'utilisation de formes rectangulaires.

2.2 Les niveaux de détails

Malgré les avancées technologiques, les capacités des ordinateurs pour le rendu interactif restent toujours fortement limitées comparées à la complexité des objets 3D actuellement produits, notamment par le biais des simulations, de la CAO ou des scanners 3D.

Il est donc nécessaire de réduire drastiquement l'information contenue dans les modèles pour permettre de manipuler ceux-ci de manière interactive. Levoy *et al.* présentent le projet "Digital Michelangelo" dans [Levoy *et al.*, 2000]. Les auteurs cherchent à produire une digitalisation à haute résolution du David de Michel-Ange, via un scanner laser de grande précision. L'objet 3D produit est une représentation numérique de la statue, comprenant 1 milliard de polygones et 7.000 images couleur. Une telle quantité de données a créé de véritables problèmes en terme de stockage mais également de rendu. Il a fallu attendre 2009 pour obtenir un modèle complet de la digitalisation effectuée entre 1997 et 1999. Comme le montre cet exemple, les objets 3D deviennent beaucoup trop complexes pour être utilisables en temps réel, le nombre de primitives étant élevé comparé aux capacités de rendu matériel.

De cette observation est née la technique de simplification de modèle, appelé "niveau de détail" (LOD, Level Of Details). Clark introduit cette méthode dans [Clark, 1976]. Dans son article, il propose d'utiliser différents modèles 3D à des résolutions différentes et de

choisir le modèle à afficher en fonction de sa localisation spatiale sur la scène. La résolution dépend alors de la distance à laquelle se trouve l'objet en fonction du point de vue de la caméra utilisateur. En prenant en compte les limites de la perception visuelle humaine, il est possible de réduire drastiquement les détails d'un modèle 3D lorsque celui-ci est placé suffisamment loin sur la scène. Les approches de simplification ont d'abord utilisé des modèles simplifiés produits à la main par des artistes, avant de devenir presque entièrement automatiques. La figure 2.10 présente le concept de LOD appliqué sur le modèle d'une vache. La résolution de l'objet change. Les détails comme le pis ne sont plus visibles. Différentes solutions existent pour simplifier des objets 3D, qu'ils soient volumiques ou surfaciques.

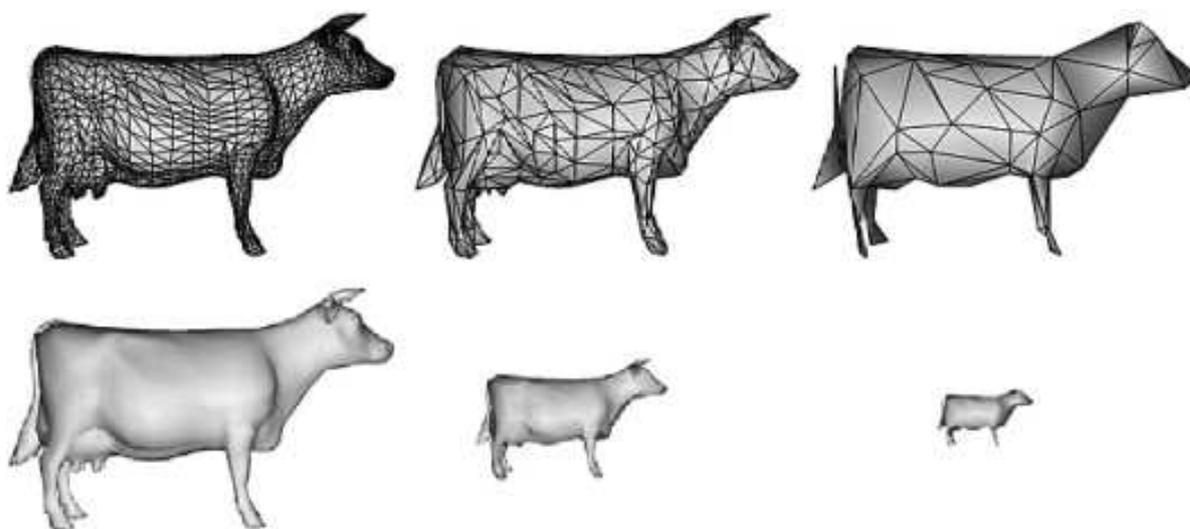


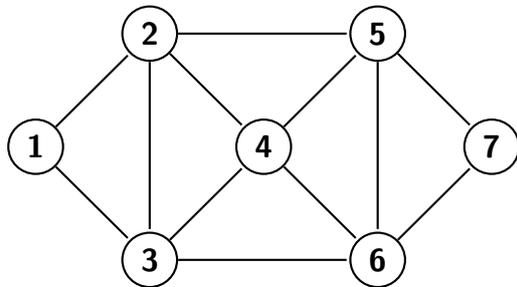
FIGURE 2.10 – Exemple sur un modèle de vache du principe de LOD, [Ribelles *et al.*, 2002].

Simplification de surface

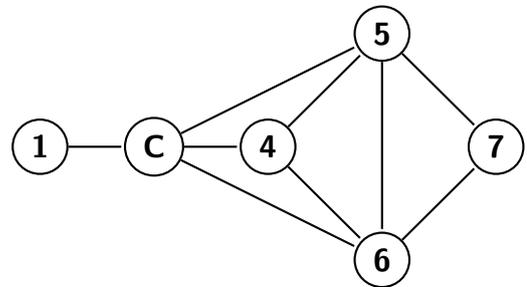
Luebke présente un état de l'art très complet sur les méthodes de simplification, [Luebke, 1997, 2001]. La plupart des méthodes de simplification de modèle surfacique sont basées sur quatre opérateurs de simplification : “edge collapse”, “edge split”, “edge swap” et “vertex split”.

Trois de ces opérateurs sont orientés autour de la simplification d'arêtes. Le premier opérateur “edge collapse” contracte une arête en un seul sommet, comme illustré figure 2.11(b). Le deuxième opérateur “edge split” divise l'arête et rajoute ainsi un sommet et des arêtes (figure 2.11(c)). Le dernier opérateur destiné aux arêtes est le “edge swap” dont l'effet est de changer le sens de l'arête afin de pouvoir appliquer certains types d'algorithmes

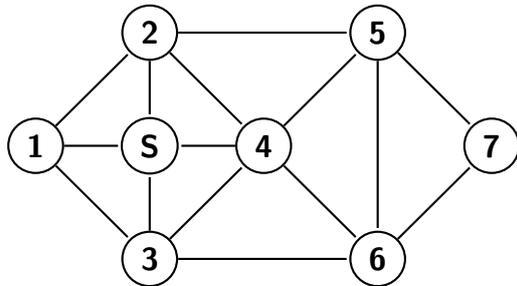
(figure 2.11(d)). Enfin, le but du dernier opérateur “vertex split” est de diviser le sommet et de le remplacer par une arête, comme le montre la figure 2.11(e).



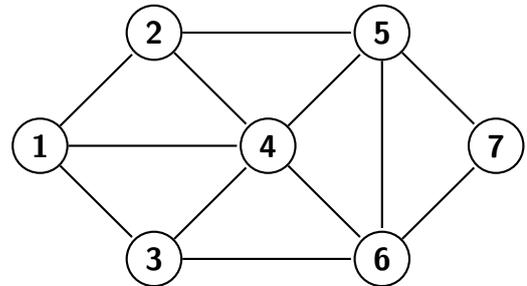
(a) Exemple d'un maillage original.



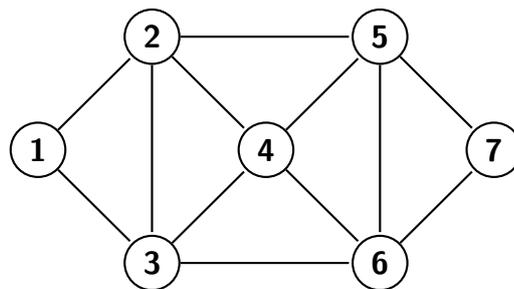
(b) A partir de la figure 2.11(a) l'opérateur de fusion est appliqué sur les sommets 2 et 3 afin d'obtenir le sommet C.



(c) A partir de la figure 2.11(a), la méthode applique sur l'arête 2 et 3 l'opérateur de division et le nouveau sommet est alors S.



(d) A partir de la figure 2.11(a), la méthode applique sur l'arête 2 et 3 l'opérateur de changement de sens.



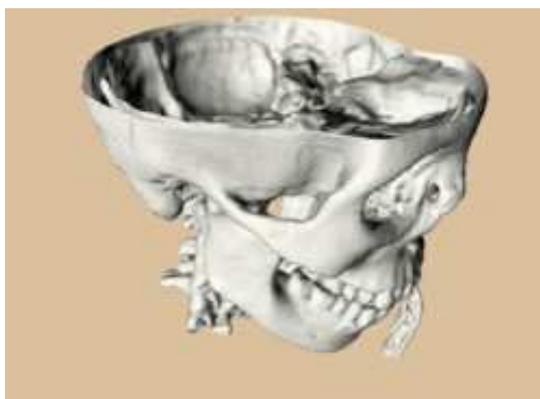
(e) L'opérateur division de sommet permet de transformer un sommet en arête à partir de 2.11(b).

FIGURE 2.11 – Différents opérateurs utilisés pour la gestion de systèmes multi-résolution.

Les algorithmes de simplification peuvent être classés en deux catégories. Les approches locales qui simplifient des primitives du modèles telles que : les arêtes, les triangles où les étoiles (la définition est présentée en annexe). Et les approches globales qui simplifient tout la géométrie du modèle 3D.

Simplification locale

Les méthodes de décimation. Le premier algorithme de simplification de modèle 3D est apparu en 1992, présenté par Schroeder *et al.* dans [Schroeder *et al.*, 1992]. L'algorithme cherche à supprimer des arêtes du maillage afin de réduire la complexité de celui-ci. Afin de ne pas affecter le rendu visuel du maillage, il est nécessaire de mesurer efficacement l'impact de la suppression d'une arête. Ainsi, dans le cadre de l'algorithme présenté, les mesures sont réalisées de manière locale en explorant un voisinage restreint. Le critère mesuré correspond à la courbure topologique de ce voisinage, si la courbure n'est pas importante, il est alors possible de supprimer le sommet sans apporter de déformations importantes au maillage. Kobbelt *et al.* présentent une bibliothèque de décimation dans [Kobbelt *et al.*, 1998], permettant à l'utilisateur de guider la simplification du modèle. Wu et Kobbelt ont ensuite proposé une nouvelle implémentation de décimation basée sur des modèles d'optimisation probabiliste, [Wu et Kobbelt, 2002]. Les figures 2.12(a) et 2.12(b) montrent le résultat d'une décimation sur un maillage de crâne humain, la première figure est composée de 569 000 triangles, tandis que la seconde est composée de 57 000 triangles, le taux de décimation est de 90%.



(a) Un maillage original de crâne.



(b) Décimation d'un maillage à 90%.

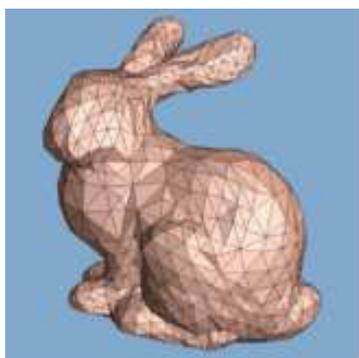
FIGURE 2.12 – Exemple de décimation réalisée sur un modèle de crâne humain, [Schroeder *et al.*, 1992].

L'échantillonnage. Une autre procédure de simplification de modèle 3D est basée sur l'échantillonnage du modèle. Le principe est de déposer des sommets à la surface du maillage de manière aléatoire ou de manière dirigée en fonction de l'algorithme choisi. Il suffit alors de supprimer le maillage original et de re-trianguler le nouvel ensemble de sommets. Cette approche a été explorée dans un premier temps par Turk, [Turk, 1992]. Dans cette méthode, l'utilisateur détermine le nombre souhaité de sommets et ensuite

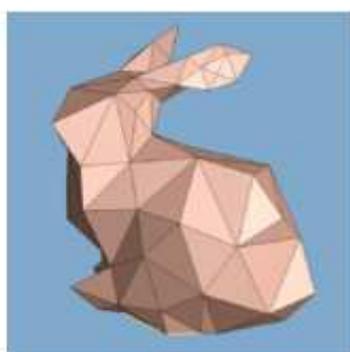
l'algorithme répartit ces sommets sur le maillage de manière aléatoire. Afin de mieux distribuer les sommets, un algorithme de répulsion est appliqué. Une autre approche de ré-échantillonnage est proposée par Krishnamurthy et Levoy, [Krishnamurthy et Levoy, 1996]. Ces auteurs proposent, dans le cadre des objets 3D issus de scanner, de produire un modèle composé de B-spline. Pour filtrer le bruit lié à la méthode de production, le modèle scanné est échantillonné. Un maillage régulier est ainsi produit.

Simplification globale

Subdivision par le bas. Cette approche travaille à partir du modèle appelé "base mesh" : modèle de base, issu d'une simplification maximale du modèle original. Afin de produire le modèle à la résolution désirée, la méthode subdivise les arêtes de ce modèle basique et après un certain nombre d'itérations le résultat obtenu est le modèle original. Cette approche a été proposée par Hoppe, [Hoppe, 1996]. L'auteur utilise une série de fusion d'arêtes afin de simplifier le modèle original pour obtenir le "base mesh". Cette solution permet de réduire le modèle 3D, mais il est également possible de raffiner celui-ci et d'obtenir plus de facettes qu'à l'origine. Un exemple du Lapin de Stanford est présenté figure 2.13(a). Le modèle de base de cet objet est présenté dans la figure 2.13(b) il s'agit d'une version très simplifiée du modèle original du lapin. Enfin trois modèles du lapin à différentes résolutions sont visibles en figure 2.13(c), le lapin le plus proche est celui à la plus haute résolution, tous les détails sont présents tandis que le lapin du fond est à la plus faible résolution et les détails ont quasiment disparu, il ne reste que les parties importantes du modèle.



(a) Maillage du lapin de Stanford après plusieurs itérations de l'algorithme.



(b) Base mesh du lapin de Stanford.



(c) Lapin de Stanford à trois résolutions.

FIGURE 2.13 – Différents résultats de l'algorithme présenté par [Hoppe, 1996].

Simplification dirigée. Cette méthode consiste à contraindre la simplification, notamment en utilisant des enveloppes ou des filtres. L'approche par enveloppe est proposée par Cohen *et al.*, [Cohen *et al.*, 1996]. Cette solution consiste à utiliser deux enveloppes : une interne au modèle et une externe au modèle. Ces deux enveloppes sont les contraintes de simplification du modèle. Ainsi lorsqu'une simplification est appliquée, elle n'est valide que si le résultat est compris entre les deux enveloppes. Un exemple de ces deux enveloppes est présenté en figure 2.14. Une autre méthode de simplification dirigée consiste à utiliser un espace composé d'une grille régulière de voxels. He *et al.* proposent une méthode basée sur la voxelisation de l'espace, [He *et al.*, 1995], la grille de voxels étant utilisée pour définir des clusters de sommets. En fonction de la finesse de la grille de voxels, le modèle sera plus ou moins raffiné.

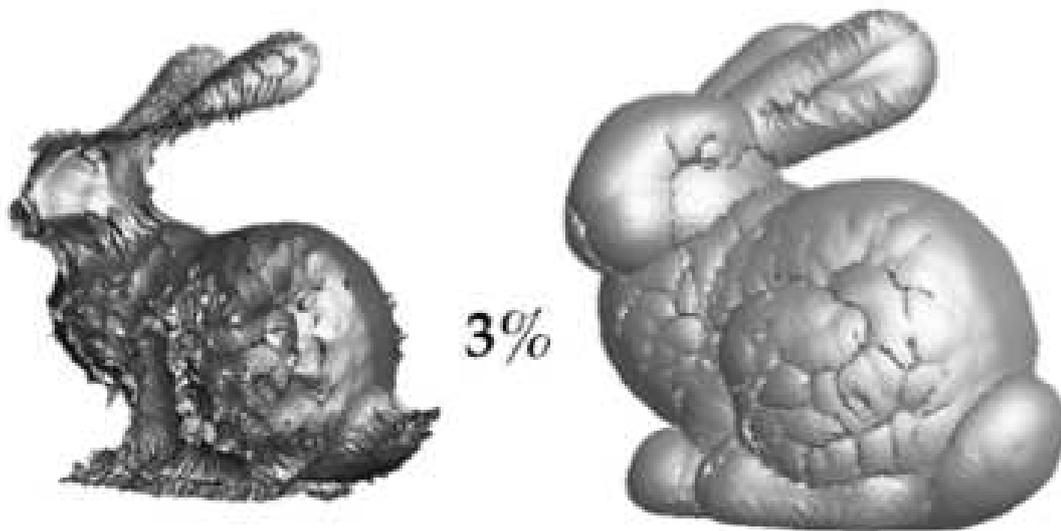


FIGURE 2.14 – Enveloppe générée à partir du lapin de Stanford, [Cohen *et al.*, 1996].

Simplification appliquée aux volumes

Les méthodes de simplification sont généralement appliquées sur des modèles surfaciques. Mais il existe des méthodes applicables sur les modèles basés sur des nuages de points ou de voxels.

Alexa *et al.* proposent une solution de simplification d'un modèle en nuage de points, [Alexa *et al.*, 2003]. Leur méthode de simplification permet de raffiner le modèle et ainsi de diminuer la densité du nuage de points. Cette approche est réversible et permet ainsi d'augmenter la densité du modèle. Malheureusement, cette méthode ne s'applique que

sur des modèles composés de nuages de points surfaciques. De nombreuses solutions de simplification d'un nuage de points sont apparues, nous pouvons citer Pauly *et al.*, [Pauly *et al.*, 2002], mais également Moenning et Dodgso [Moenning et Dodgson, 2004] et Song et Fen, [Song et Feng, 2009]. Les solutions de simplification de volume utilisent généralement une grille de voxels comme présentée dans [He *et al.*, 1995].

L'approche multi-résolution que nous avons introduite ici sera utilisée dans le cadre de nos travaux afin de nous permettre de gérer plus efficacement les visualisations sur des dispositifs ayant des caractéristiques non homogènes.

2.3 Les architectures parallèles

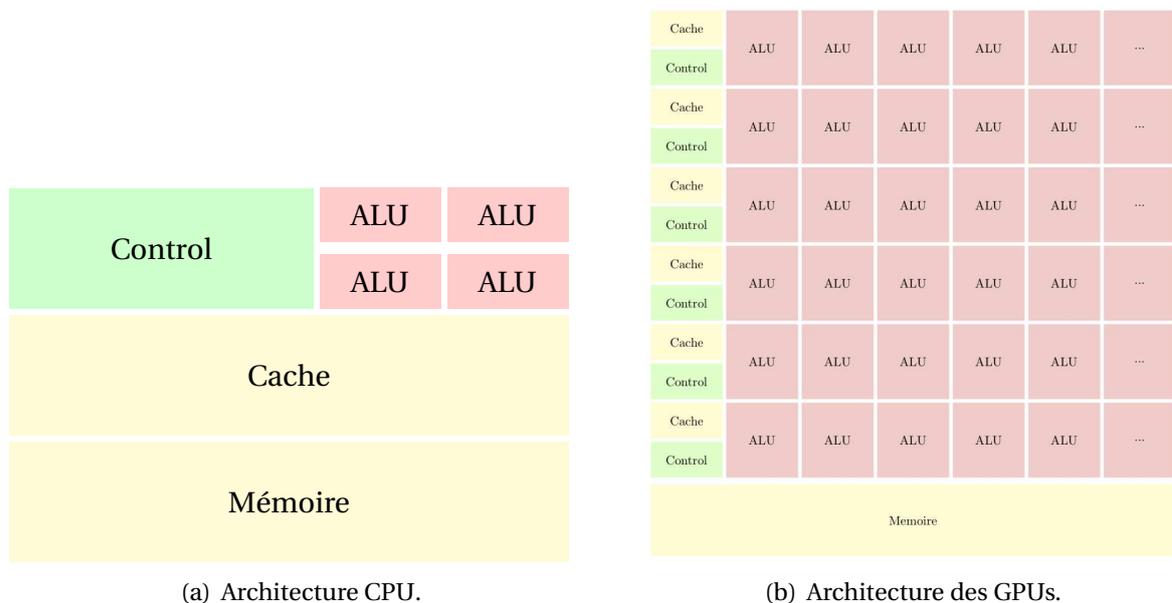


FIGURE 2.15 – Présentation des architectures CPU et GPU.

Les architectures parallèles sont largement utilisées depuis de nombreuses années dans le domaine industriel et scientifique, ces approches permettent d'améliorer les temps de calculs liés aux algorithmes ou données complexes.

Une architecture parallèle possède plusieurs unités de calcul et donne à chacune d'elles une tâche; cette tâche est définie par un sous-algorithme exécuté de manière indépendante et nécessite donc son propre espace de travail en mémoire. Une unité de calcul peut être locale (architecture multi-CPU) ou déportée (plusieurs ordinateurs). A ce jour, les architectures déportées restent chères et complexes à mettre en place.

Des solutions grand public sont progressivement apparues pour réaliser du calcul parallèle. Les éditeurs de cartes graphiques ont proposé des langages de programmation capables d'interagir directement avec l'unité de calcul graphique : cette méthode s'appelle GPGPU (General-Purpose computing on Graphics Processing Units). Les GPU sont composés de centaines d'unités de calcul, comme illustré figure 2.15(b), là où les CPU sont seulement dotés d'une dizaine de ces unités, figure 2.15(a). Malgré tout, les unités arithmétiques (ALU) des GPU sont plus limitées pour certains calculs que celles des CPU.

Des solutions hybrides commencent à émerger, notamment au niveau d'une unification des méthodes de parallélisation, par le biais d'OpenCL, normalisé par le groupe Khronos² [Group *et al.*, 2008]. Ils proposent d'unifier les deux types d'architecture (CPU et GPU) sous un seul langage et ainsi réduire le coût de développement en améliorant les performances. Baumann *et al.* [Baumann *et al.*, 2009] ont présenté la première architecture système proposant d'allier les différents types d'unités de calcul afin d'avoir une capacité de calcul évolutive.

Dans la visualisation de données 2D et 3D, le calcul parallèle est utile pour améliorer le rendu mais également pour le pré-traitement des données. En effet, de nombreuses solutions ont été expérimentées afin d'étudier l'efficacité des solutions à base de calcul parallèle pour les systèmes de visualisation. On trouve ainsi deux approches, celles qui sont orientées autour des pré-traitements et les solutions de visualisation. Dans ces deux cas, différentes solutions matérielles existent soit en utilisant une grappe de machines, soit en utilisant un périphérique adapté tel une carte graphique.

Toutes les notions présentées dans cette section font partie de nos travaux. Nous nous sommes appuyés sur la modélisation par systèmes de particules et par voxels. Enfin, plusieurs tests sur des architectures parallèles nous ont permis d'améliorer notre efficacité énergétique

2. <http://www.khronos.org>

Visualisation de données

L'informatique n'est qu'un outil,
comme un pinceau ou un crayon.

GRIFFO

Sommaire

3.1	Introduction	25
3.2	Définitions relatives aux données	26
3.3	Visualisation adaptée à la problématique du bâtiment	27
3.4	Visualisation de données multi-dimensionnelles	31
3.5	Les moteurs 3D	42
3.6	Synthèse	45

3.1 Introduction

La masse de données générée par l'activité humaine grandit proportionnellement au développement de la démographie et des nouvelles technologies. Par conséquent, la quantité d'informations produites par seconde à l'échelle mondiale peut atteindre plusieurs teras octets. À l'image de ce schéma global, la masse d'informations produites par les systèmes de gestion de bâtiment prend des proportions impressionnantes.

Les informations ainsi générées nécessitent d'être manipulées afin d'être mieux appréhendées. Afin de répondre à ce besoin, différents mécanismes ont été mis en place, no-

tamment des systèmes de visualisation. Dans ce chapitre, nous allons présenter un état de l'art relatif à la visualisation de données orientée autour du bâtiment.

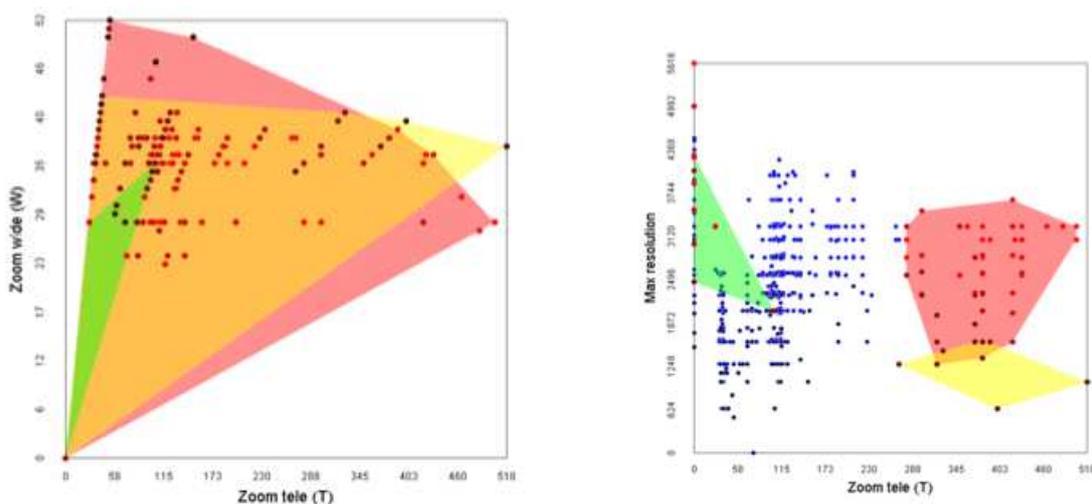
La visualisation de données numériques est un domaine vaste. De nombreuses solutions de visualisation sont apparues au cours du temps, une introduction au domaine de visualisation d'information est proposée par Bertin, [Bertin, 1967]. Ce domaine se base sur des jeux de données : homogènes ou hétérogènes. Nous définissons ces deux méthodologies dans la section 3.2. De plus ces travaux sont orientés autour d'un jeu de données issues du bâtiment, nous présenterons différentes méthodes de visualisation pour ce type de données dans la section 3.3. Dans la section 3.4, nous présentons des approches de visualisation couramment utilisées mais également des méthodes de visualisation 3D. Dans la section 3.5, nous présentons les différents moteurs 3D que nous avons étudiés dans le cadre de la visualisation de données.

3.2 Définitions relatives aux données

Une donnée est la représentation d'une information qui peut être caractérisée par différents attributs qui lui sont propres. Pour un capteur de température par exemple, les différentes caractéristiques peuvent être : la localisation dans l'espace, une date et une valeur de température. Ces différents attributs sont appelés dimensions : une donnée est donc représentée par un agrégat de dimensions. Une notation existe pour les données composées de plus de trois dimensions, le terme nD est utilisé dans ce cas là.

Elmqvist *et al.* présentent un jeu de données composé de 1039 appareils photos, [Elmqvist *et al.*, 2008]. Chacun de ces appareils possède 13 dimensions : une dédiée au nom de l'appareil, les autres dimensions représentent des valeurs : de prix, ouverture focale, etc. Les données qui sont utilisées dans cet exemple sont des données homogènes : pour chaque élément de la base de données, les dimensions sont identiques. Dans la figure 3.1(a), l'auteur nous présente une vue de ce jeu de données, au travers de deux dimensions de zoom. Comme le montre la représentation, trois sous-ensembles de données ont été sélectionnés. Lorsque l'utilisateur effectue un changement de dimensions, figure 3.1(b), alors les données précédemment sélectionnées sont re-localisées. Certains appareils qui semblaient choisis lors de la vue précédente, se retrouvent alors hors de toute sélection. Dans cet exemple, l'utilisateur peut raffiner son choix au travers de la manipulation des différentes dimensions.

Les données hétérogènes sont des données ayant un schéma structurel variable. Par exemple, le type ou le nombre de dimensions peut varier pour deux données. Ainsi, avoir des données non homogènes complexifie l'ajout de nouvelles informations aux modèles : les méthodes automatiques d'intégration s'adaptent difficilement à des schémas structurels variables. Cammarano *et al.* ont proposé une stratégie d'adaptation dans le cadre des



(a) Vue 2D de deux valeurs de zooms.

(b) Une autre vue 2D mêlant zoom et résolution.

FIGURE 3.1 – Vue basée sur le principe des dés présenté dans [Elmqvist *et al.*, 2008].

aggrégateurs de contenu web de type mashup, [Cammarano *et al.*, 2007]. Ces outils produisent et utilisent des données non structurées : leur schéma de données est variable en fonction des contenus des sites web. La solution proposée permet d'intégrer de manière efficace de nouvelles sources de données, mais également de produire une visualisation adaptée. Dans ces travaux, les auteurs ont représenté les informations au travers d'un modèle RDF (Ressource Description Framework). Leur système de visualisation est basé sur le principe de graphe : chaque objet est modélisé par un noeud. Les auteurs représentent les données sous la forme de triplets : (T_i, N_i, E_i) avec $i \in \mathbb{N}$, T_i le type de l'attribut, N_i le nom de l'attribut et enfin E_i la représentation visuelle de cet attribut. La figure 3.2(a) présente une visualisation sous forme de nuage de points, les données sont soit modélisées par un drapeau soit par un cercle en fonction l'information disponible. Sur la figure 3.2(b), les données manquantes sont également visibles, au travers d'une représentation vide des éléments. Ces différents cas présentent des méthodologies de représentation de données hétérogènes.

3.3 Visualisation adaptée à la problématique du bâtiment

Nos travaux de recherche ont pour objectif de produire une solution de visualisation adaptée aux bâtiments. Ainsi, nous nous sommes intéressés aux différentes solutions existantes pour la visualisation de données extraites à partir de capteurs, ainsi que les différentes visualisations des bâtiments. Dans cette section, nous étudions les différentes solutions de visualisation pour le bâtiment.



(a) Vue 2D de deux valeurs de zoom.

(b) Une autre vue 2D mêlant zoom et résolution.

FIGURE 3.2 – Différentes présentations proposées dans [Cammarano *et al.*, 2007].

Un grand nombre d'approches se sont principalement focalisées sur les données thermiques du bâtiment. En effet, ces données permettent de mettre en avant les problèmes éventuels d'isolation. Comme le présente la figure 3.3, au niveau des fenêtres, la température semble la plus élevée comme sur les ampoules des lampadaires. Sur la façade du bâtiment, on peut également analyser le comportement de la température à la surface des différents murs et s'apercevoir d'éventuelles fuites d'air.

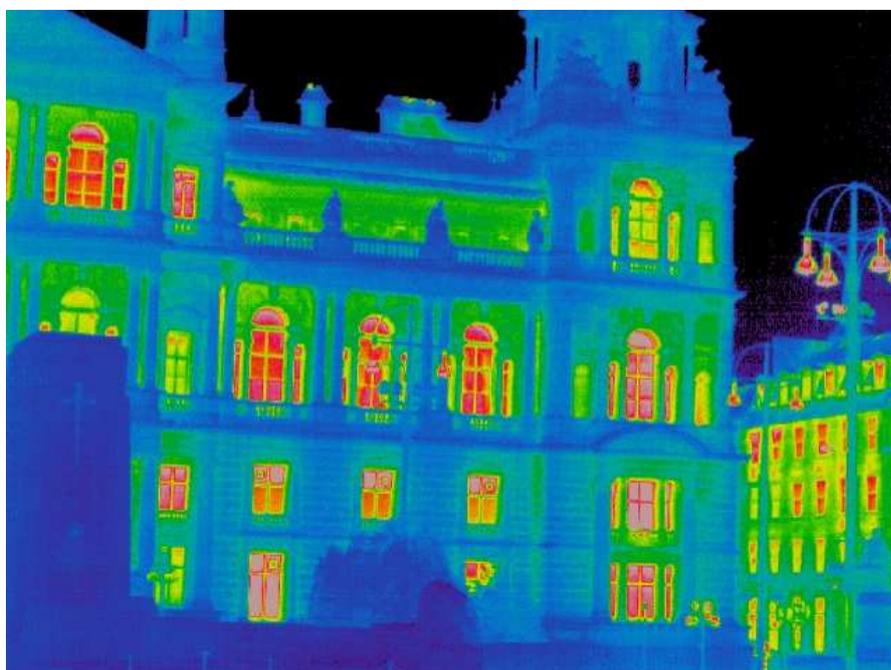


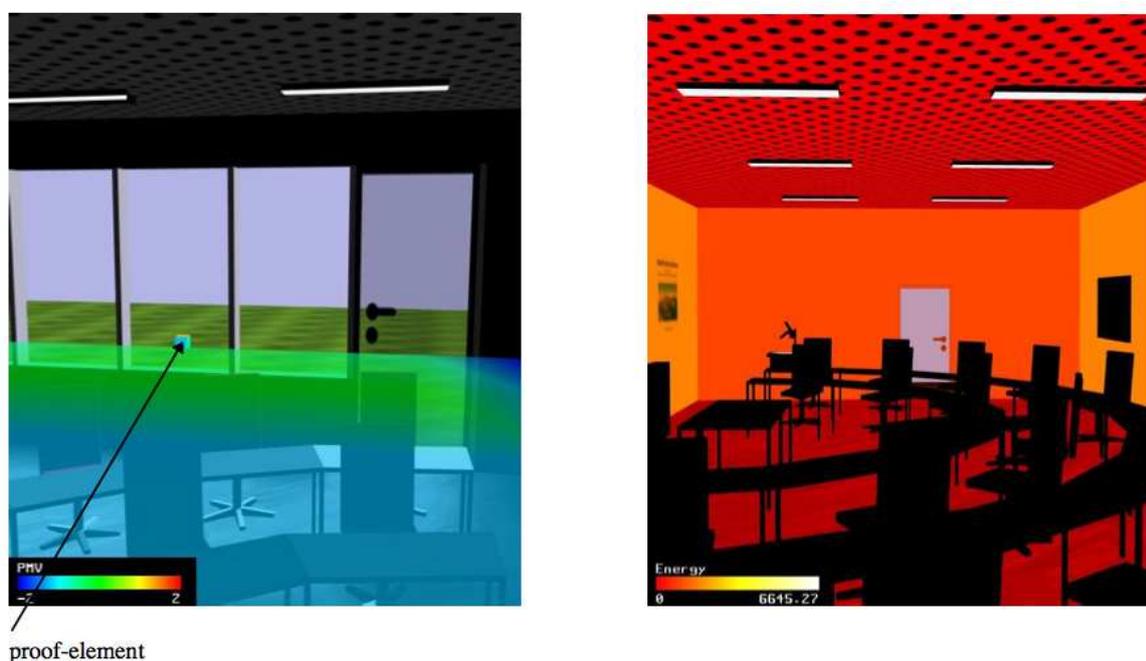
FIGURE 3.3 – Vue infrarouge d'un bâtiment à Glasgow.

L'étude de ces flux de températures est appelée thermographie, et ce domaine permet l'analyse du comportement du bâtiment. Pour produire un tel rendu, la méthode la plus couramment utilisée repose sur l'utilisation d'une caméra thermique avec prise de vues aériennes. Pour réduire les coûts de ces campagnes de mesures, de nouvelles approches se basent sur l'utilisation d'un modèle comportemental du bâtiment utilisant des capteurs de températures. L'extraction de ces modèles comportementaux est une tâche complexe lorsque l'encombrement du bâtiment est pris en compte. La visualisation de ces données peut permettre de valider ou invalider ces modèles comportementaux. Les solutions commerciales de visualisation de ces données restent orientées autour des visualisations 2D classiques comme les courbes et les nuages de points. Ces solutions ne s'attachent pas à avoir une représentation spatiale des données. La compréhension est totalement décorrélée du bâtiment. Il existe une solution de visualisation appelée carte de température qui consiste à prendre en compte les informations spatiales au travers d'une série d'interpolation de mesures. Cette approche ne permet pas de représenter les volumes du bâtiment. Schreyer et Hoque proposent d'utiliser le principe de plug-in de google sketchup afin de combiner un modèle 3D avec des informations cartographiques infra-rouge, [Schreyer et Hoque, 2009].

Dans leurs travaux Schöffel *et al.* s'appuient sur la visualisation du confort, [Schöffel *et al.*, 1999]. Ils ont pour objectif d'améliorer les informations proposées à l'utilisateur. Leur solution repose sur une amélioration de systèmes de simulation thermique existants, un modèle 3D est utilisé comme support d'information. Le modèle thermique de bâtiment produit par leur simulateur est au format STEP (STandard for the Exchange of Product model). Il s'agit d'un format ouvert permettant de modéliser des éléments. Il a pour objectif d'intégrer les éléments de type conception, développement, fabrication et maintenance de l'objet modélisé. Leur modèle physique de confort thermique est basé sur un outil de simulation développé par le CSTB (Centre Scientifique et Technique du Bâtiment¹). Le résultat est une texture appliquée sur le modèle virtuel du bâtiment. Les équipes du CSTB ont également développé un utilitaire de visualisation adaptée à la compréhension de l'utilisateur. Deux approches de visualisation sont proposées à l'utilisateur dans cet outil : la coloration des différents plans du bâtiment et une représentation mobile. La première approche habille les différents murs du bâtiment par des textures générées par les simulations physiques. La seconde solution repose sur l'utilisation d'un élément de petite taille (un cube) qui est texturé en fonction des différentes températures qu'il subit. Dans la figure 3.4(a), nous pouvons voir l'approche de visualisation basée sur le cube ainsi que sur une surface de température dans un environnement. Le cube est texturé en fonction des températures précises. Dans la figure 3.4(b), la solution montre l'application d'une carte globale des températures d'un bureau.

Des solutions de simulation comme COMFIE (moteur de simulation thermique dyna-

1. <http://www.cstb.fr>



(a) Visualisation d'une personne sous forme de cube, avec la surface virtuelle de température.

(b) Visualisation des températures appliquées sur les murs du modèle.

FIGURE 3.4 – Présentation de la solution issue de [Schöffel *et al.*, 1999].

mique multi-zone) sont également dédiées au bâtiment. Salomon *et al.* ont étendu ce moteur de simulation pour réaliser des simulations dynamiques de l'enveloppe du bâtiment dans [Salomon *et al.*, 2005]. COMFIE étendu est associé à d'autres outils de simulations afin de créer un comportement plus réaliste. Ils ont ainsi utilisé un moteur de simulation orienté autour des masques d'ensoleillement, le but est de réaliser un simulateur dynamique qui prend en charge l'évolution de la position du soleil dans le temps. Grâce à ce nouvel environnement, il est possible pour les architectes de mieux connaître le comportement du futur bâtiment qu'ils ont à réaliser.

La visualisation de données issues de capteurs a peu évolué durant ces dernières années. La majorité des méthodes repose sur des approches classiques de visualisation d'information. Buschmann *et al.* présentent une solution de visualisation extensible de réseau de capteurs, [Buschmann *et al.*, 2005]. Les informations mesurées par les capteurs sont envoyées à un système central appelé passerelle. Cette passerelle est un agrégateur de données de capteurs. Cette solution est développée pour être utilisée au travers d'une visualisation extensible (système par couche). Le principe mis en place dans ces travaux repose sur l'utilisation de composants, comme par exemple, une couche de fond permettant

de mettre en avant la zone dans laquelle le capteur se trouve, une couche de relations qui présente les connexités dans le graphe de capteurs notamment pour mettre en avant les éléments proches et une couche de noeuds qui représente les capteurs. Une représentation de leur architecture est présentée en figure 3.5. Comme le présente cette figure, un nœud centralise les données des différents capteurs qui envoient leurs données au travers du protocole TCP/IP. Afin d'analyser les données de manière visuelle, le système crée un rendu à partir des données stockées par ce système et différents composants sont mis en place notamment une visualisation des nœuds, une visualisation par relation, etc. Ce type d'architecture permet donc de visualiser différents contenus en fonction de ce que l'utilisateur désire, de plus le système en couches permet de sélectionner plus simplement les informations cruciales dont l'utilisateur a besoin.

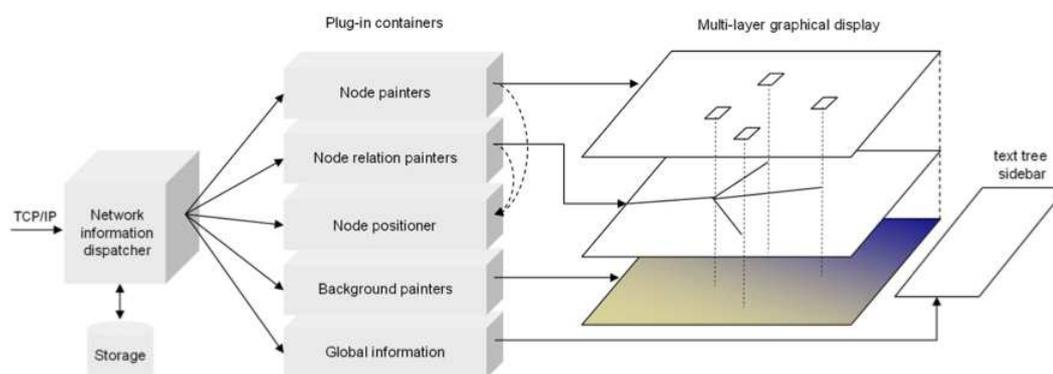


FIGURE 3.5 – Architecture de la solution SpyGlass, [Buschmann *et al.*, 2005]

3.4 Visualisation de données multi-dimensionnelles

La visualisation de données multi-dimensionnelles est une problématique complexe, notamment pour la cartographie de l'information et le nombre de dimensions à visualiser.

Dans cette section, nous présentons différents travaux en matière de visualisation. Nous nous intéressons aux différentes classes de visualisation à savoir les méthodes géométriques, iconiques et hiérarchiques. Cette classification est inspirée de la classification de Chan [Chan, 2006].

3.4.1 Visualisation géométrique

Les méthodes de visualisation géométrique ont pour but de trouver des solutions efficaces pour cartographier les dimensions d'une donnée dans un espace avec peu de dimen-

sions. Dans la majorité des cas, ces solutions utilisent seulement deux ou trois dimensions. Les nuages de points et les coordonnées parallèles (ou lignes brisées) sont les deux principales approches de la visualisation géométrique.

Les nuages de points, plus communément “scatter plot”, sont des représentations 2D ou 3D d’une donnée. Ce type de visualisation projette deux ou trois dimensions sur les axes définis par la vue. Afin de visualiser d’autres dimensions, il est nécessaire de changer les paramètres de la vue et ainsi créer une nouvelle vue. La première extension à cette méthode est l’utilisation d’une matrice de vues qui permet de visualiser sur le même écran les différentes combinaisons de dimensions. Sur la figure 3.6, la solution de représentation géométrique basée sur une matrice de nuage de point est donné, cette approche présente tous les nuages de points possibles avec ce jeu de données. Grâce à cette méthode de visualisation, il devient possible d’identifier des motifs visuels, par exemple une corrélation entre la puissance en nombre de chevaux et le poids. Ce type de visualisation peut très rapidement devenir difficile à comprendre et à exploiter si le nombre de vues est trop important.

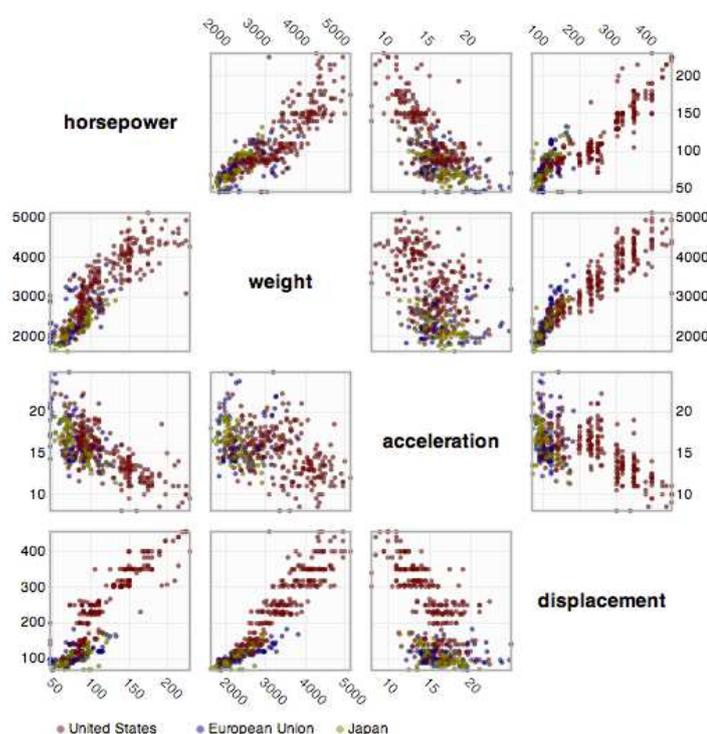
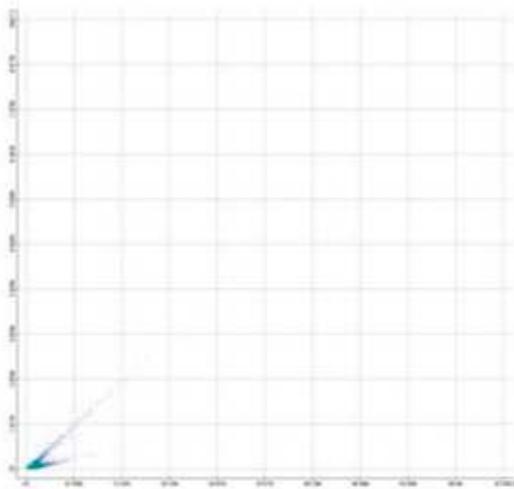


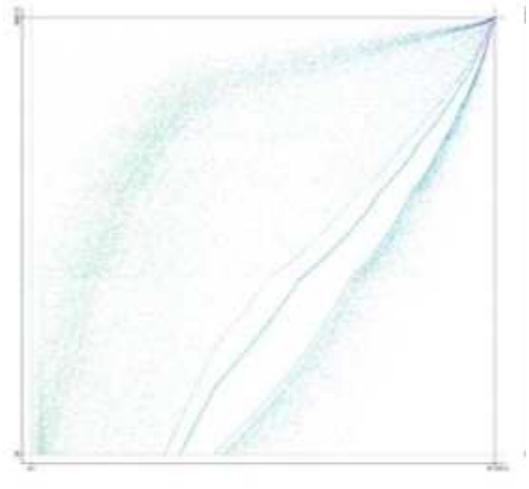
FIGURE 3.6 – Matrice de nuages de points sur des données de véhicules [Keim *et al.*, 2010]

Keim *et al.* proposent une approche améliorée de la visualisation par nuage de points, [Keim *et al.*, 2010]. Le problème de recouvrement de l’information est l’un des inconvénients majeurs de ce type de visualisation, surtout lorsque la quantité d’information est

importante. Les auteurs proposent une solution basée sur une analyse humaine qui permet d'optimiser au mieux le degré de chevauchement des données. La visualisation proposée cartographie l'ensemble des valeurs d'une donnée et l'applique à un pixel de la vue. Les méthodes classiques superposent les données qui doivent être localisées sur le même pixel. La solution de Keim *et al.* propose d'appliquer une distorsion spatiale pour les données superposées. L'utilisateur peut affiner le résultat en appliquant lui aussi des distorsions supplémentaires. Keim *et al.* représentent une visualisation par nuage de point classique avec les différentes occlusions dans la figure 3.7(a). La figure 3.7(b), quant à elle, présente l'amélioration proposée dans leur article ; les données ne se superposent plus, il est donc possible de mieux appréhender le résultat.



(a) Nuage de points brut sans traitement lié à la superposition.



(b) Résultat de la méthode présentée dans [Keim *et al.*, 2010], après plusieurs itérations de distorsion demandées par l'utilisateur.

FIGURE 3.7 – Représentation par nuage de points issue de [Keim *et al.*, 2010].

Une autre approche de visualisation multi-dimensionnelle est proposée par Chan *et al.*, [Chan *et al.*, 2010]. Dans ces travaux, les auteurs cherchent à mettre en avant les corrélations entre les différentes données. Pour cela, ils estiment une vélocité dans le nuage de points. Grâce à cette information, il est alors possible de proposer une méthode de représentation basée sur le principe de flot. La figure 3.8 présente le résultat de cette méthode de visualisation, en donnant un aspect de fluide sur un jeu de données, il est alors possible d'identifier des comportements temporels.

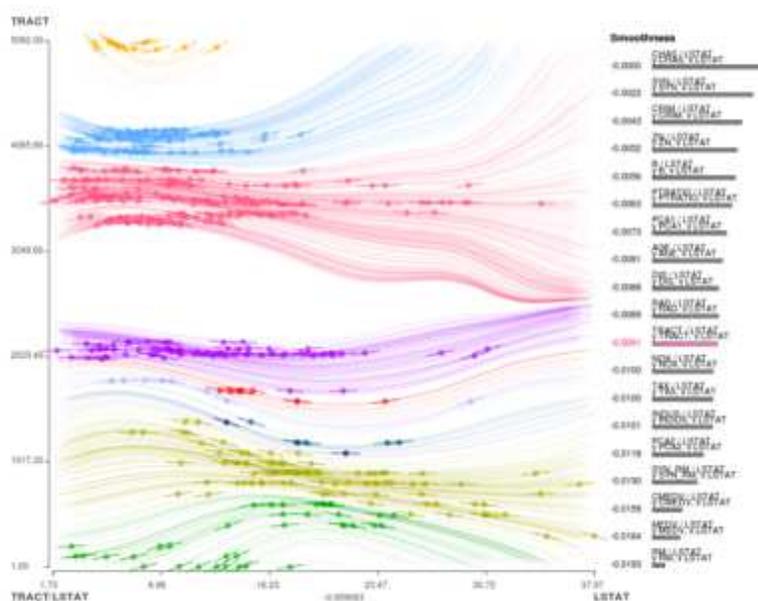


FIGURE 3.8 – Visualisation de flot au travers d'un nuage de points présentée dans [Chan *et al.*, 2010]

Une autre approche de visualisation multidimensionnelle est la réduction de dimensions. Il s'agit d'élaborer des méthodes efficaces de combinaison de dimensions. Il existe deux types d'approches supervisées et non supervisées dans différents domaines comme la fouille de données, l'extraction de contenu et la segmentation d'information. L'approche supervisée est développée autour des connaissances apportées par l'utilisateur. L'approche non supervisée est automatique et basée sur la préservation de propriétés géométriques (distance, angles). Les approches non supervisées sont généralement représentées dans un espace euclidien, mais il est possible de généraliser les méthodes pour obtenir des solutions non euclidiennes. Carter *et al.* présentent dans leurs travaux un framework de réduction de dimensions, [Carter *et al.*, 2011]. Le but de ces travaux est de proposer une approche basée sur des techniques non supervisées de réduction de dimensions. Le principe consiste à apporter une information sémantique spatiale aux différentes données et ainsi appliquer de manière générique une méthode de réduction de dimensions. Enfin, ils appliquent le résultat sur un jeu de données constitué de données médicales, ceci permet d'améliorer le diagnostic final du patient.

L'approche par coordonnées parallèles est une technique très connue et largement utilisée pour la visualisation de données multi-dimensionnelles. Chaque attribut des données est représenté par un axe vertical, dont l'unité est normalisée en fonction des valeurs. Une donnée est représentée par une ligne brisée qui coupe les différents axes. Avec cette méthode, il est possible de comprendre et d'appréhender plus rapidement les corrélations

entre les différentes données, [Bostock et Heer, 2009]. La figure 3.9 présente une implémentation de la représentation par coordonnées parallèles reposant sur la sélection de données sur chaque dimension. Il est alors possible de mettre en avant certaines valeurs ayant une latitude d'évolution faible. D'autres méthodes de visualisation sont également basées sur le concept d'agrégation d'axes, au lieu de représenter les dimensions sur un axe horizontal, elles utilisent une représentation circulaire. Cette méthode est appelée : graphe en étoile ou graphe radar.

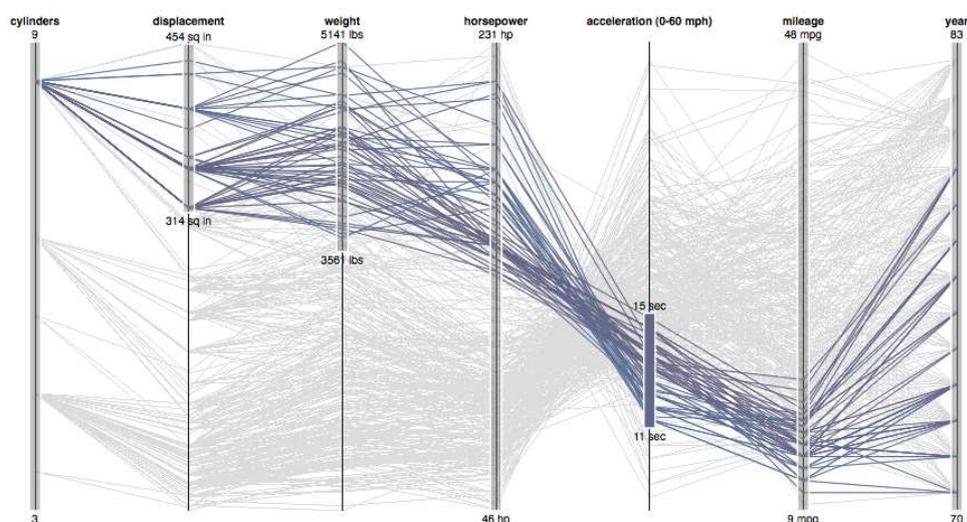
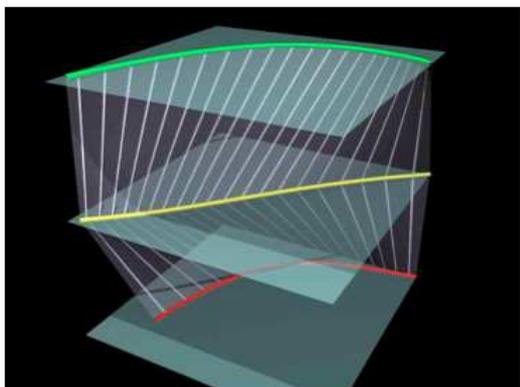


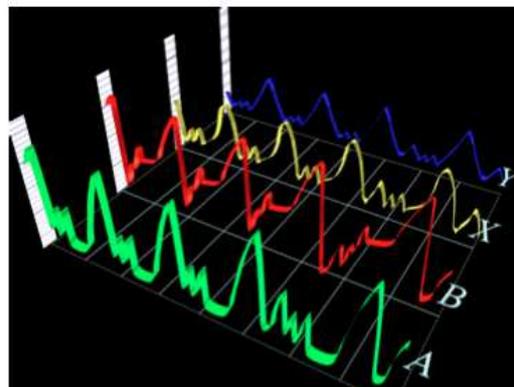
FIGURE 3.9 – Visualisation proposée par Protovis dans le cadre de [Bostock et Heer, 2009], Protovis propose un ensemble de solutions pour la visualisation de données.

Wegenkittl *et al.* présentent trois méthodes basées sur les coordonnées parallèles afin de créer une visualisation multi-dimensionnelle, [Wegenkittl *et al.*, 1997]. L'idée développée ici est de représenter les trajectoires des données. La première méthode repose sur une représentation des coordonnées parallèles de manière surfacique. Cette solution utilise un espace 3D, qui permet une représentation sous forme de surface du jeu de données. La deuxième solution consiste à prendre deux attributs au hasard et de projeter les différentes données tout au long des différents axes. Enfin, la troisième coordonnée peut être sélectionnée par l'utilisateur afin de produire une trajectoire dans l'espace. La troisième solution de visualisation proposée utilise deux plans horizontaux pour cartographier les données. Deux trajectoires sont visualisées en même temps, il est possible d'étendre cette approche en multipliant le nombre de plans utilisés. Au lieu d'utiliser des plans parallèles, il est également possible d'articuler les différents plans de manière différente par exemple de manière orthogonale. La figure 3.10 présente deux exemples de représentations utilisées dans cet article. La figure 3.10(a) présente une vue par coordonnées parallèles, basée

sur un espace 3D. La figure 3.10(b) présente une représentation 3D des données de bourses en utilisant une vue composée de quatre dimensions.



(a) Combinaison de 6 axes sur 3 plans différents



(b) Visualisation de 4 axes en parallèle sur un espace 3D

FIGURE 3.10 – Exemple de visualisation 3D de lignes brisées, [Wegenkittl *et al.*, 1997].

Fanea *et al.* ont développé un mélange de différentes méthodes de visualisation ayant pour base les coordonnées parallèles, [Fanea *et al.*, 2005]. Leur méthode s'inspire des deux méthodes de représentation au travers d'axes horizontaux et circulaires. Les auteurs ont mis en oeuvre le principe des glyphes en étoile. Dans cet article la solution utilisée est basée sur une visualisation tri-dimensionnelle des données au travers d'un graphe. Comme illustré figure 3.11, la représentation est un mélange de coordonnées parallèles et de glyphes en étoile. La visualisation globale des données est obtenue au travers d'une représentation tri dimensionnelle.

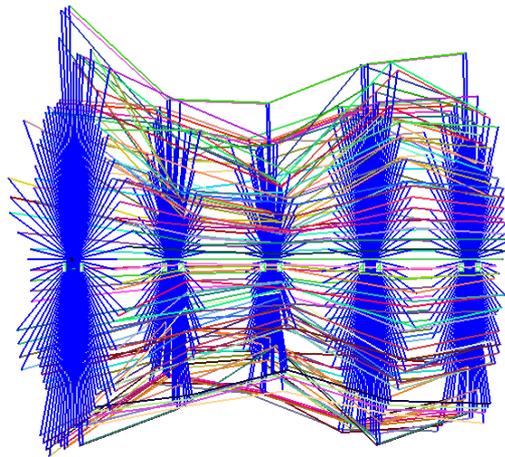


FIGURE 3.11 – Visualisation de données composée d’une méthode de glyphes en étoile et de coordonnées parallèles, [Fanea *et al.*, 2005]

3.4.2 Visualisation iconique

Une autre classe de visualisation que nous avons étudiée concerne la visualisation par icône. Cette représentation génère un symbole sous la forme d’une petite image. Ces symboles sont également appelés glyphes. De nombreuses méthodes de représentation sous forme d’icônes existent : les visages (smileys), les glyphes étoiles, les figures à base de bâton, l’encodage de formes, les textures...

Anderson présente une des premières briques nécessaires à la visualisation iconique, [Anderson, 1957]. L’auteur représente des informations au travers de la génération de petit glyphe. Trouver une visualisation pertinente permet de grandement simplifier la tâche d’extraction de contenu pour l’humain. La solution proposée dans cet article consiste à dessiner un petit symbole dont chaque paramètre est variable en fonction des états des différentes dimensions de la donnée. Concrètement, cette approche consiste à remplacer le traditionnel nuage de points par un nuage de glyphes. Chaque glyphe étant composé : d’un cercle qui peut être localisé en fonction de deux dimensions ainsi qu’une série de “membres”. Il s’agit de petit bâtons placés au-dessus du cercle et dont la hauteur varie en fonction de la valeur dans la dimension donnée. La figure 3.12 illustre ces travaux sur une donnée composée de cinq dimensions. Ainsi, la ligne A présente les différentes représentations de l’icône pour un type de données, la ligne B présente les différentes représentations de cette donnée, avec les différents états possibles. Enfin, cinq exemples de représentation sont donnés.

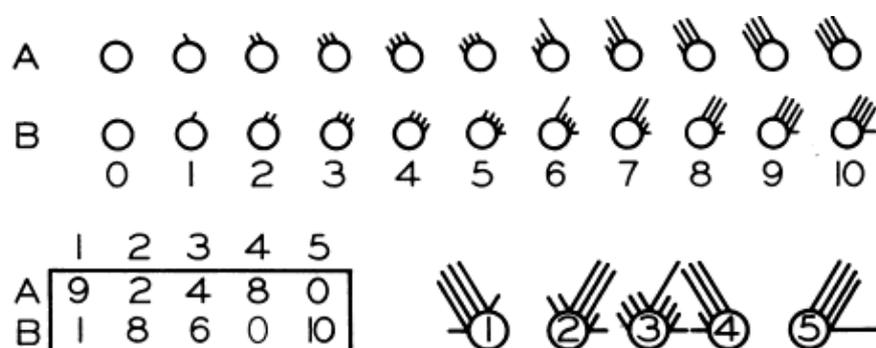


FIGURE 3.12 – La figure est composée des deux éléments A et B. A et B ont différents états représentés en haut de la figure. La visualisation que nous cherchons à produire correspondant au schéma en bas à gauche. Et en combinant les vues A et B, le résultat obtenu est la figure en bas à droite, [Anderson, 1957].

Une nouvelle approche est introduite par Chernoff, [Chernoff, 1973]. La problématique consiste alors en une visualisation de symboles représentés par des visages. Le nombre de dimensions que l'auteur cherche à visualiser est inférieure à 18. Chaque dimension est assignée à une caractéristique du visage telle que la longueur du nez ou la courbure de la bouche par exemple. L'utilisation du visage est un outil mnémotechnique mais également un outil de compréhension rapide de l'information. Grâce à cette méthode, les petites différences de valeurs sont extraites beaucoup plus rapidement par l'utilisateur. Le cerveau humain est utilisé ici comme un ordinateur, il filtre les phénomènes insignifiants et se focalise sur ceux qui sont importants. Si le nombre de dimensions est trop important, il devient alors difficile d'assigner toutes ces caractéristiques à un visage. De plus, les dimensions doivent être assignées correctement afin de ne pas induire en erreur la compréhension de l'utilisateur.

Une autre approche de la visualisation iconique est la création de textures complexes. Pickett et Grinstein présentent une méthode de génération de textures basée sur des données multi-dimensionnelles, [Pickett et Grinstein, 1988]. Ils expérimentent ainsi leur méthode de visualisation sur des données météorologiques issues d'images satellites. Leur approche utilise une série de petit bâtons qui sont associés les uns aux autres pour représenter une forme. Ils assemblent ces différentes formes pour obtenir une représentation des données, ceci crée un rendu de texture pour l'œil humain, qui peut alors être utilisé pour extraire de la redondance ou un effet saillant. Les icônes sont composées de cinq membres (bâtons) qui sont reliés entre eux. La base de la représentation est un des membres qui est appelé corps. Les quatre autres éléments peuvent alors être soit attachés entre eux sous forme d'une chaîne, soit reliés directement au corps. Par cette représentation, il est possible d'obtenir 12 classes différentes de données à visualiser. Un exemple

de résultat proposé par cette solution est donnée en figure 3.13, ainsi, la figure permet de mettre en avant une zone homogène dont le motif semble différent du reste de la figure.

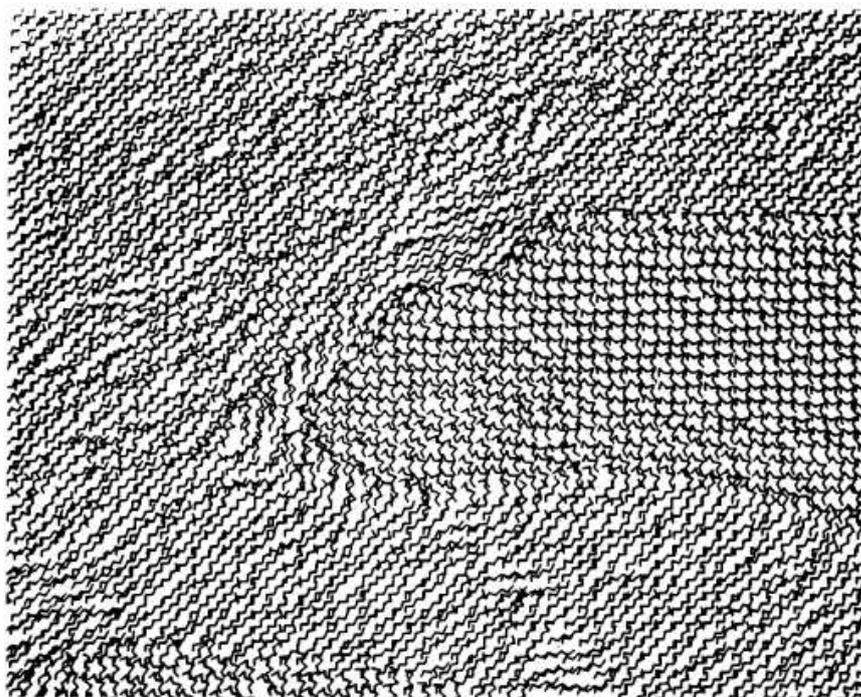


FIGURE 3.13 – Représentation sous forme de texture complexe de données issues d'images multi-spectrales produites par un satellite météo, [Pickett et Grinstein, 1988].

La dernière approche que nous avons étudiée est une représentation 3D basée sur de la modélisation d'icône en 3D. Van Walsum *et al.* proposent le développement d'un framework de visualisation pour la production d'icônes, [Van Walsum *et al.*, 1996]. Les différentes caractéristiques des icônes sont produites en fonction des valeurs des dimensions de chaque donnée. Pour aider dans la génération de visualisation, un langage de description a été mis en place. Cette approche permet de générer une grande variété d'icônes et de configurer celles-ci. Leur framework est également développé pour produire une visualisation à différents niveaux de détails, qui dépend du niveau d'abstraction désiré. Dans la figure 3.14, la solution mise en avant présente différents résultats de données, les différentes formes sont des agglomérats de formes particulières.

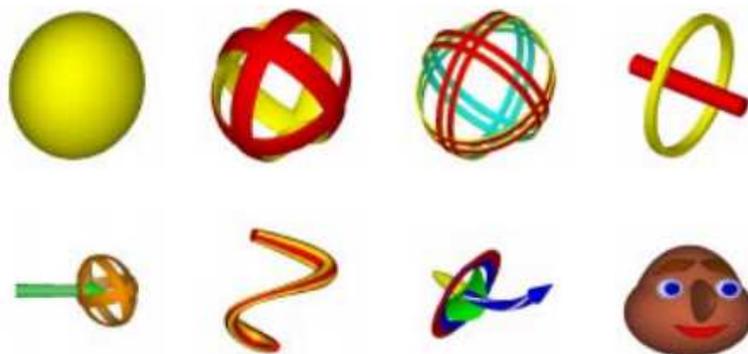


FIGURE 3.14 – Exemple de représentation sous forme d’icônes produites par la méthode [Van Walsum *et al.*, 1996].

3.4.3 Visualisation hiérarchique

Les dernières solutions de visualisation auxquelles nous nous sommes intéressés sont les méthodes de visualisation hiérarchique. L’espace de visualisation est divisé en sous-espaces. L’utilisation de cette méthode a souvent donné lieu à une hybridation des techniques de présentation de l’information afin de proposer un ensemble de vues à l’utilisateur ; il lui est alors possible de mélanger différentes formes de représentations de l’information.

De nombreuses visualisations hiérarchiques se sont focalisées sur des stratégies en arbre. L’approche la plus connue est le “treemap”. Les treemaps sont des visualisations qui utilisent une décomposition hiérarchique de l’écran en fonction des différents attributs des valeurs. La taille des différents éléments est déterminée par la valeur des attributs. La couleur peut également servir à encoder de l’information. Cette représentation est composée par des tuiles, Johnson présente TreeViz une solution de visualisation hiérarchique, [Johnson, 1992]. Il s’agit ici d’une implémentation de la visualisation par treemap. Treeviz permet de représenter l’information en utilisant différents attributs de différentes manières : textuelle, position et représentation en arbre. La taille des différents noeuds dépend des valeurs des attributs. Cette solution utilise l’interaction de l’utilisateur pour extraire des informations utiles. Une représentation du temps est donnée en figure 3.15, les différentes zones voient leur taille évoluer en fonction de l’espace utilisé par les différents fichiers.

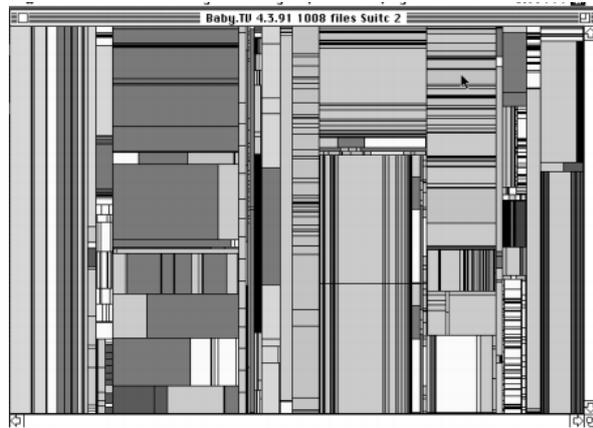
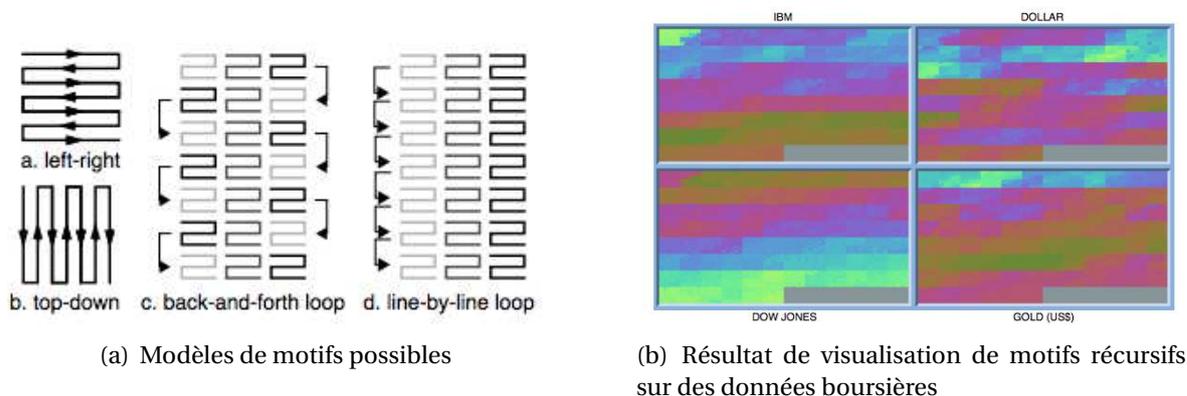


FIGURE 3.15 – Visualisation par Treemap d'un système de fichiers, [Johnson, 1992]

Nous nous sommes également intéressés aux solutions basées sur les motifs à répétition présentées par Keim *et al.*, [Keim *et al.*, 1995]. Les auteurs proposent une méthode de généralisation des méthodes de visualisation orientées pixels. Les approches qui ne sont pas orientées pixels permettent de ne visualiser qu'une quantité limitée d'informations au travers d'une vue, de l'ordre de 100 à 5.000 attributs. Keim *et al.* utilisent chaque pixel d'un écran ; il devient alors possible de créer une vue comportant environ 1.000.000 d'attributs. Ce type de visualisation cartographie chaque dimension de données sur un pixel. Grâce à cette approche, la quantité d'informations qui peut être visualisée dépend du matériel utilisé. L'auteur propose également à l'utilisateur de paramétrer la vue qu'il utilise : différents motifs de cartographie des dimensions ont été mis en place. La figure 3.16(a) présente les différents motifs récurrents utilisés dans le cadre des travaux de cet article. Enfin la figure 3.16(b) présente des exemples de représentation de données issues de la bourse. Ici le modèle récurrent utile est en ligne et permet de mettre en avant une linéarité de l'évolution.



(a) Modèles de motifs possibles

(b) Résultat de visualisation de motifs récurrents sur des données boursières

FIGURE 3.16 – Modélisation d'une visualisation par motif présentée dans [Keim *et al.*, 1995]

D'autres solutions de visualisation existent et sont basées sur des agrégations de vues. Certaines de ces applications sont développées pour un contexte particulier, tandis que d'autres sont développées pour proposer une généralisation d'outils de visualisation. Matkovic *et al.* proposent ComVis comme outil de visualisation, [Matkovic *et al.*, 2008]. Cette solution est basée sur des systèmes multi-vues. Ils cherchent à développer une solution de prototypage rapide pour les visualisations. Chaque vue est paramétrable par l'utilisateur, ceci lui permet de s'appropriier l'outil et son interface, ainsi la phase d'apprentissage se retrouve réduite. Dans cet exemple, la solution est paramétrable par l'utilisateur, un exemple de configuration est proposé en figure 3.17 où différentes représentations sont combinées afin de créer une vue des données. Il est ainsi possible de suivre différentes évolutions au travers de plusieurs vues.

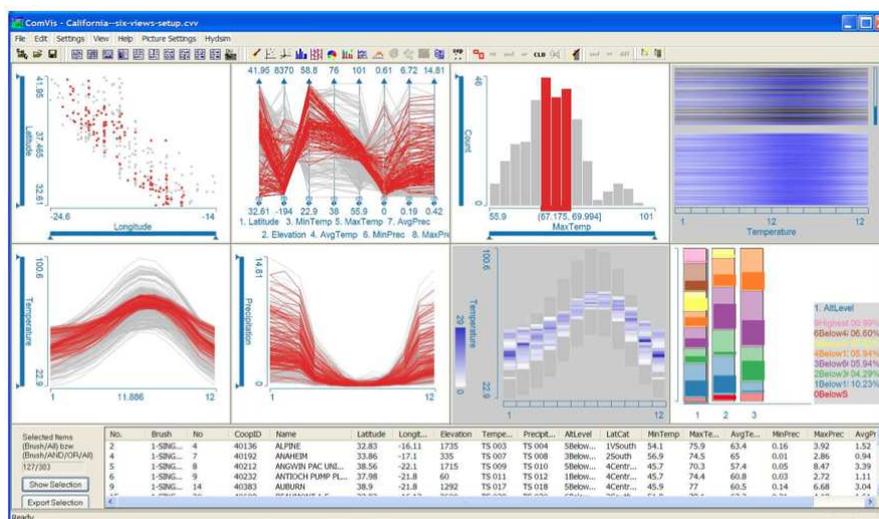


FIGURE 3.17 – Exemple d'une configuration de vue sous ComVis, [Matkovic *et al.*, 2008]

3.5 Les moteurs 3D

Après avoir exploré les différentes méthodes de visualisation, nous pouvons nous intéresser aux méthodes permettant de les mettre en scène : les moteurs 3D. De nombreux moteurs 3D sont utilisés dans le cadre de la visualisation. Dans cette section, nous effectuons un état de l'art de différents moteurs 3D avec leurs caractéristiques, leurs points forts et leurs faiblesses.

La première bibliothèque que nous avons étudiée est OpenGL (Open Graphics Library²). Cette bibliothèque est utilisable sur différents supports de visualisation, que ce soient des environnements Linux, Windows, OSX, mobiles. Elle permet de gérer les diffé-

2. <http://www.opengl.org>

rentes primitives basiques utilisables sur des dispositifs graphiques et de gérer une caméra élémentaire. Cette bibliothèque ne permet pas la gestion de l'environnement au travers d'un graphe de scène³ par exemple. Afin d'obtenir un environnement suffisamment complet, il est nécessaire de rajouter de nombreuses structures de données.

D'autres bibliothèques 3D existent et sont plus complètes. Ces solutions permettent de manipuler plus simplement les objets contenus dans les scènes et même la caméra. De nombreuses bibliothèques existent comme VTK⁴ (Visualization ToolKit), qui est une bibliothèque orientée autour de la visualisation volumique et de fluides, ou des solutions comme OpenSceneGraph⁵ qui est orientée autour de la réalisation d'environnements virtuels. Un autre exemple de solution grand public est Ogre3D⁶. Le tableau 3.1 présente différentes caractéristiques des moteurs 3D. Le critère de scène représente la faculté du moteur à être doté d'une modélisation par graphe de scène. Le critère d'interaction représente les capacités du moteur à gérer des interactions simples : manipulation de l'élément, outil d'exploration du contenu. Le critère de collaboration représente la capacité du moteur à être utilisé par plusieurs utilisateurs simultanément. Le critère de volume représente les capacités d'un moteur graphique à ne pas être simplement efficace avec la gestion de modèle surfacique mais également avec des modèles volumiques.

	Scène	Interaction	Collaboration	Volume
OpenGL	Non	Non	Non	Non
VTK	Non	Non	Non	Oui
Ogre 3D	Oui	Oui	Non	Non
OpenScene Graph	Oui	Oui	Non	Oui

TABLE 3.1 – tableau récapitulatif de différentes bibliothèque de rendu.

De nombreux travaux se sont intéressés à trouver la bibliothèque la plus adaptée au contexte. Cubaud *et al.* ont développé un système d'exploration de bibliothèques. Afin de réaliser cette application, ils ont donc expérimenté différentes solutions pour la visualisation comme par exemple VirTools⁷ et VRML⁸. Virtools est une bibliothèque très complète de visualisation de données développée par Dassault Système⁹. Cette bibliothèque est orientée autour de Windows et ne permet pas d'être utilisée sur d'autres plate-formes. VRML (Virtual Reality Modeling Language) est une méthode de représentation 3D inter-

3. Le graphe de scène structure de manière logique la représentation spatiale d'une scène graphique.

4. <http://www.vtk.org>

5. <http://www.openscenegraph.org>

6. <http://www.ogre3d.org>

7. <http://www.3ds.com/products/3dvia/3dvia-virtools/>

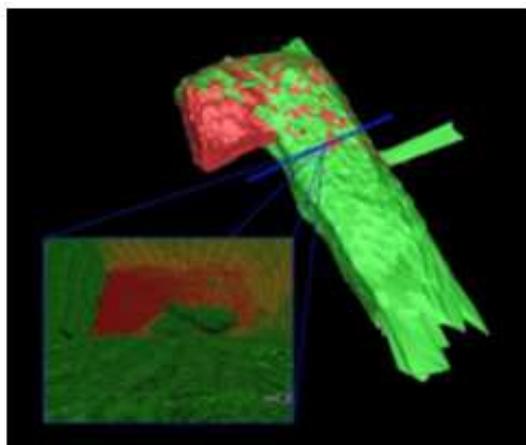
8. <http://www.vrml.org>

9. <http://www.3ds.com/fr/>

prétée. L'environnement 3D est modélisé au travers d'un fichier normalisé et un outil de visualisation produit le rendu. Cubaud *et al.* se sont intéressés à certains critères : l'interactivité, l'animation, la détection de collisions, [Cubaud *et al.*, 2005]. Les solutions commerciales se sont révélées les plus pertinentes mais avec un prix élevé pour réaliser des développements sur le court terme.

Des environnements plus complexes sont utilisables pour réaliser des visualisations efficaces, notamment au travers des mondes virtuels. Ceux-ci sont très largement utilisés dans des domaines particuliers tels que les jeux sérieux ou les jeux vidéos par exemple. Ils sont originalement développés pour des usages ludiques ou assimilés, mais peuvent être également utilisés dans le cadre de visualisation de données.

Farr *et al.* ont développé une visualisation autour de l'environnement OpenSim¹⁰, [Farr *et al.*, 2009]. Cet environnement est un serveur de mondes virtuels libre de droits, qui fournit un environnement immersif 3D pour le client Second Life¹¹. Les auteurs utilisent OpenSim comme une plate-forme collaborative qui peut être manipulée par plusieurs utilisateurs en même temps. Ils ont ajouté également un moteur physique afin de simuler les déplacements de ces éléments. Malheureusement un tel système est limité : le moteur de simulation OpenSim ne permet pas de gérer suffisamment d'éléments car il est contraint à 50. Malgré tout, ce type d'environnement reste très intéressant, notamment pour la visualisation collaborative.



(a) Un exemple de vue proposée dans [Friese *et al.*, 2008] : un explorateur de grotte en 3D.



(b) Modélisation d'une ville au travers d'OpenSceneGraph présentée dans [Fritsch et Kada, 2004].

FIGURE 3.18 – Exemple d'utilisation de moteur de 3D pour des outils de visualisation

10. <http://opensimulator.org>

11. <http://secondlife.com>

Enfin, d'autres solutions sont basées sur des moteurs 3D utilisés dans le domaine des jeux vidéo afin de créer des moteurs de visualisation. Cette approche a été explorée dans [Fritsch et Kada, 2004; Friese *et al.*, 2008]. Comparée aux solutions professionnelles, l'utilisation de moteurs de jeux est moins coûteuse. Les travaux de thèse de Friese étaient orientés sur la reconstruction d'objet 3D à partir d'un scan laser. Dans ces travaux l'auteur a voulu s'affranchir du développement du moteur de rendu, il a préféré utiliser une solution existante. Il a étudié trois approches issues du jeu sérieux afin d'évaluer le meilleur outil pour la visualisation de données. Dans la littérature, des approches ont utilisé le moteur de Quake 3¹², mais également celui d'Unreal Tournament 2004¹³ ou bien celui de FarCry¹⁴. L'utilisation d'un moteur de rendu orienté jeux vidéos a des avantages : de bonnes performances et peu de pré-requis matériels. Mais, ceux-ci ont également des inconvénients : faible documentation, des restrictions logicielles, une durée de vie limitée et peu d'extensibilité. Un exemple de rendu produit par un moteur de jeux est présenté en figure 3.18(a). Fritsch et Kada ont utilisé ce type de moteur afin de réaliser un outil de visualisation de zones urbaines. Ils ont ainsi pu appliquer différentes simulations de flux : touristiques, urgences, trafics. Un exemple de simulation est illustré figure 3.18(b).

3.6 Synthèse

Dans ce chapitre, nous nous sommes intéressés au domaine de la visualisation d'informations. Ce domaine a pour but de représenter des données de manière visuelle. Les données peuvent être multi-dimensionnelles, avec des attributs des différentes catégories. Par exemple, une donnée peut avoir des informations de température, pression, hygrométrie, position, date, ... Nous nous sommes intéressés aux méthodes relatives au domaine de la visualisation d'informations ainsi qu'aux différentes techniques mises en œuvre pour aider à la compréhension des données.

Nous avons passé en revue trois grandes classes de visualisations existantes : hiérarchiques, géométriques et iconiques. Les méthodes géométriques utilisent l'espace de visualisation pour cartographier les données. Ainsi, il est possible de visualiser des données en 2D ou 3D simplement en utilisant les méthodes courantes de représentation. Mais, il est également possible d'extraire des corrélations de ces données pour simplifier le résultat de la visualisation. Les approches basées sur des icônes utilisent une représentation par pictogramme, ceci permet à l'utilisateur de garder une compréhension globale du jeu de données. Si l'attribution des caractéristiques est bien utilisée alors les icônes peuvent être de redoutables outils de communication au travers des capacités humaines. Enfin, les vi-

12. <http://www.idsoftware.com>

13. <http://www.unrealengine.com>

14. <http://www.ubi.com/fr/>

sualisations hiérarchiques utilisent des systèmes de division de l'espace pour représenter des modèles récurrents des données.

Les différentes méthodes de visualisation que nous avons introduites représentent les solutions majeures utilisées dans le cadre de systèmes de visualisation. Les solutions de gestion de bâtiments utilisent une partie des méthodes présentées, notamment les méthodes de visualisation par matrice de nuage de points, par coordonnées parallèles ou par glyphe. Les autres méthodes présentées dans ce chapitre, n'ont pour la plupart pas été industrialisées, en effet celles-ci sont trop complexes à mettre en œuvre dans ce type d'application.

Dans nos travaux de recherche, présentés dans la seconde partie de ce manuscrit, nous verrons comment utiliser et développer ces méthodes de visualisation dans le cadre d'un outil spécifiquement étudié pour analyser la consommation énergétique d'un bâtiment. Les différentes solutions que nous avons présentées sont utilisées dans le chapitre 6.

Visualisation analytique de données numériques

Informatique : alliance d'une science inexacte et d'une activité humaine faillible.

L. FAYARD

Sommaire

4.1	Introduction	47
4.2	La gestion de données	48
4.3	L'interaction appliquée à la donnée	51
4.4	Des visualisations multi-vues	55
4.5	Synthèse	67

4.1 Introduction

Le dernier domaine auquel nous nous sommes intéressés concerne l'analyse visuelle (VA pour Visual Analytics). Ce domaine est présenté dans [Thomas et Cook, 2005], [Keim, 2002] et [Shneiderman, 2001]. Il est défini comme étant la science du raisonnement analytique, facilité par une interface graphique pertinente. Ce domaine de recherche est orienté autour de différentes activités : la gestion de l'information, la détection d'information, la possibilité d'évaluer ces résultats et enfin les actions possibles en utilisant les informations

précédentes. Ce domaine est fortement connexe au domaine IV (pour Information Visualization).

Le domaine VA (Visual Analytics) couvre toute la procédure de la prise de décision en combinant les méthodes de la fouille de données automatique, la visualisation d'information et la cognition humaine. Keim *et al.* présentent différents segments de ce domaine, dans [Keim *et al.*, 2006] et [Keim *et al.*, 2008] : la visualisation d'informations, la gestion des données, l'analyse de la donnée, la perception associée à la cognition humaine, les interactions hommes machines, les méthodes mises en place pour évaluer le contenu de la visualisation. Foley *et al.* définissent une taxonomie du domaine, [Foley *et al.*, 2006], basée sur les travaux de Thomas et Cook, [Thomas et Cook, 2005]. Dans leur article, Foley *et al.* proposent ainsi les principaux segments qui nous ont intéressés. Dans ce manuscrit, nous avons principalement étudié des méthodes de visualisation de ce domaine, des solutions de gestion des données, de l'analyse de ces données et enfin différents moyens d'interaction.

Dans ce chapitre, nous allons présenter différentes solutions d'applications développées autour du domaine de la fouille de données visuelle. Une des problématiques principales de ce domaine est la gestion de données. Ces données sont habituellement massives et complexes à manipuler. Dans la section 4.2, nous présentons les approches mises en place pour gérer ces données. Afin de manipuler ces données, des méthodes d'interactions sophistiquées ont été mises en place. En effet, les solutions traditionnelles ne suffisent plus à manipuler les différentes vues créées. Nous présenterons dans la section 4.3 ces méthodes d'interactions. Dans ce domaine, les différentes approches de visualisation qui sont menées sont orientées autour de systèmes multi-vues ; nous présenterons celles-ci dans la section 4.4.

4.2 La gestion de données

Dans cette section, nous nous intéressons aux différentes solutions utilisées pour gérer les données. Les masses de données produites de nos jours par les différents systèmes informatiques nécessitent des solutions adaptées de gestion. Le but est de permettre de traiter dans un temps raisonnable les différentes requêtes d'un système informatique. Dans cette section, nous abordons les solutions mises en place pour gérer des données complexes.

Une des premières solutions que nous avons étudiées organise les données en utilisant le principe d'agrégation. En effet, il est important d'arranger au mieux les différentes données, notamment en utilisant des solutions de tri. Une visualisation pertinente nécessite de correctement traiter l'information en fonction de différents critères. Ankerst *et al.* travaillent sur la problématique d'agrégation qui permet d'extraire des informations d'une

importante masse de données au travers de différents systèmes de visualisation déjà existants, [Ankerst *et al.*, 1998]. L'approche proposée par Ankerst *et al.* présente une solution d'arrangement des données en fonction de leur similarité. Les données avec une forte similarité sont placées proches les unes des autres. La similarité est évaluée en fonction des caractéristiques de dimensions. La mesure de similarité de cette approche consiste à identifier deux dimensions similaires, c'est à dire qui évoluent peu l'une par rapport à l'autre. Un des problèmes principaux de cette approche est que la similarité n'est pas la même dans tous les cas d'usage. Une agrégation appliquée sur certaines dimensions n'aura pas la même signification pour des experts de domaines différents. L'extraction de contenu est donc une étape importante dans le processus d'analyse des données ; l'extraction de corrélations est cruciale.

La visualisation de flot est une autre discipline, qui nécessite notamment de manipuler de grandes masses de données multi-dimensionnelles. Lors de simulation de fluide, les données produites par les simulateurs sont trop massives et dépassent très rapidement les capacités de stockage des machines standards. L'aspect visuel proposé par une solution de visualisation de fluide est très important, notamment pour la compréhension des résultats. Ilcik propose un kit de composants logiciels de visualisation interactif pour la gestion de données de fluide, [Ilcik, 2008]. Cette méthode est basée sur la visualisation sélective composée d'éléments calculés. L'architecture du kit de composants logiciels proposé est une agrégation de deux briques principales : la brique de calcul et la brique de visualisation. Le module de calcul permet de déterminer les différents mouvements des flux, tandis que la partie visualisation est basée sur une interface multi-vues connectée. Les vues sont une solution hybride de rendu volumique, utilisant différentes techniques de rendu simultanément. Leur kit de composants logiciels permet ainsi un rendu interactif de données massives en utilisant un ordinateur standard.

Borgeat *et al.* ont développé une solution de visualisation de données complexes, [Borgeat *et al.*, 2005]. Les données traitées dans cet article sont issues de capteurs. Ce type de sources de données possède un inconvénient majeur, il s'agit du taux de rafraîchissement qui est trop important. Leur solution est développée autour de deux briques : la première réalise les différents calculs nécessaires ainsi que la gestion des données, l'autre réalise la visualisation. La partie visualisation est un outil destiné aux systèmes à écran large et exploite un modèle multi-résolution : les images de deux vidéo projecteurs sont superposées, comme l'illustre la figure 4.1. L'image centrale est plus petite, le nombre de pixels au centimètre carré est donc plus élevé. La solution de gestion des données est orientée autour des données de modèles 3D. La solution développée dans le contexte de ces travaux propose donc une vue multi-résolution pour des systèmes à écran large. La solution présentée dans cet article est similaire à la méthode précédente, une approche fragmentée est utilisée.



FIGURE 4.1 – Illustration du système de multi-résolution, sur un écran collaboratif, [Borgeat *et al.*, 2005].

La dernière approche que nous avons étudiée concerne la segmentation de l'espace. Nous nous sommes intéressés à ce type de techniques de gestion de l'espace afin de mieux répondre à la problématique de gestion de données. Afin de gérer des données complexes, certaines approches se basent sur ce type de méthodes afin d'ordonner les données. Une des solutions les plus fréquemment utilisées est appelée KD-tree. Le KD-tree fonctionne de la manière suivante, l'algorithme divise un espace de manière à optimiser au maximum les partitions de cet espace. Tobler propose une amélioration de l'algorithme de KD-tree et dont le principe est utilisé dans de nombreuses applications de localisation et simulation, [Tobler, 2011]. Le but est d'élaborer la meilleure solution pour extraire le voisinage de différents points. Cette méthode permet de rapidement localiser des informations au travers d'un espace de grande taille. L'approche présentée ici mélange une segmentation par des plans de manière orthogonale et circulaire comme présenté en figure 4.2. L'évaluation de cette solution montre que le calcul de voisinage de différents points est plus efficace en mélangeant la division de l'espace de manière orthogonale et circulaire.

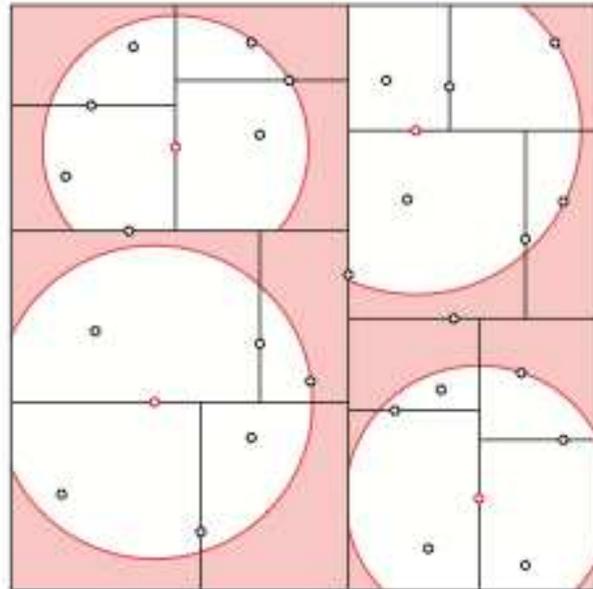


FIGURE 4.2 – Résultat de la division de l'espace par la solution de Tobler, [Tobler, 2011].

En terme de gestion de données, de nombreuses solutions se sont tournées vers une décomposition des traitements : une brique est destinée à réaliser les calculs complexes et la gestion de la donnée, tandis que la deuxième brique réalise la visualisation.

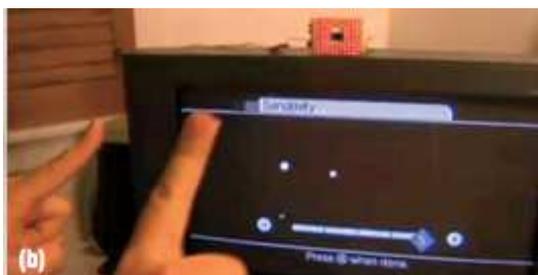
4.3 L'interaction appliquée à la donnée

Afin de manipuler des données complexes, il est nécessaire d'élaborer des stratégies d'interactions intuitives. Le modèle d'interaction nécessaire pour manipuler une donnée est dans de nombreux cas ignoré. Malgré tout, la manipulation de données reste une des parties les plus importantes pour l'extraction de contenu. Dans cette section, nous étudions différentes solutions d'interactions permettant de fouiller dynamiquement les données. Nous nous sommes intéressés à trois approches d'interactions : les dispositifs matériels / logiciels et les outils collaboratifs.

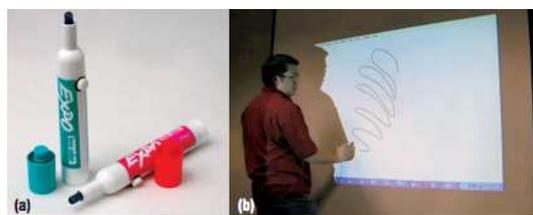
4.3.1 L'interaction matérielle et logicielle

La première solution d'interaction que nous avons étudiée est issue des travaux de Lee présentés dans [Lee, 2008]. Nintendo a présenté la Wiimote en 2006 en s'appuyant sur un nouveau type de périphérique de contrôle. Ce dispositif est basé sur une série de capteurs utilisés afin d'analyser les mouvements produits par un utilisateur. Ainsi, la Wiimote est composée de différents capteurs dont des accéléromètres et une caméra infrarouge. Dans

le cadre de ses travaux, l'auteur a cherché à détourner les usages de cette manette afin de permettre d'explorer de nouvelles méthodes d'interactions. La première solution mise en avant dans ces travaux repose sur l'utilisation de la caméra : des miroirs sont placés aux extrémités des doigts, un projecteur émet un champ infrarouge et la caméra capte la réflexion de ce projecteur. Malheureusement, la caméra ne permet pas de suivre plus de 4 éléments. La solution peut ainsi suivre les déplacements de 4 doigts (limitation de la camera). La deuxième solution présentée permet de réaliser un tableau blanc interactif. Le but de ce dispositif est de permettre de développer une interface collaborative. L'outil est basé sur l'utilisation d'un crayon qui émet de l'infrarouge, les déplacements de celui-ci sont suivis par la caméra et retransmis au système. La dernière méthode mise en place est basée sur la détection de visage : l'utilisateur s'équipe d'un dispositif spécifique équipé de 2 diodes électroluminescentes infra-rouges. La caméra sert à analyser la localisation des diodes électroluminescentes pour détecter la position du visage. La solution obtenue est une méthode de suivi de visage à faible coût. La figure 4.3(a) présente le dispositif mis en place par l'auteur : la grille de diodes ainsi que la Wiimote placée derrière. La figure 4.3(b) présente le stylo développé par l'auteur ainsi que son utilisation sur un tableau blanc interactif.



(a) Interface de suivi de doigts basée sur une Wiimote.



(b) Interface de tableau numérique utilisant une Wiimote.

FIGURE 4.3 – Exemple d'approches utilisant une Wiimote comme moyen d'interaction, [Lee, 2008] .

Han propose une solution qui repose sur la réflexion des rayons infrarouges dans le plexiglass, afin de capter la localisation des doigts, [Han, 2005]. En fait l'auteur propose une solution d'interface multi-points à bas prix. Cette solution repose sur l'utilisation d'une caméra infrarouge et d'un jeu de diodes infrarouges placé autour d'une surface en Plexiglas. Les résultats de cette méthode sont satisfaisants pour créer une interface tactile de grande taille.

Enfin, pour communiquer entre le périphérique et le logiciel, des solutions ont besoin d'être développées, soit au travers d'un pilote qui retransmet les entrées utilisateur cor-

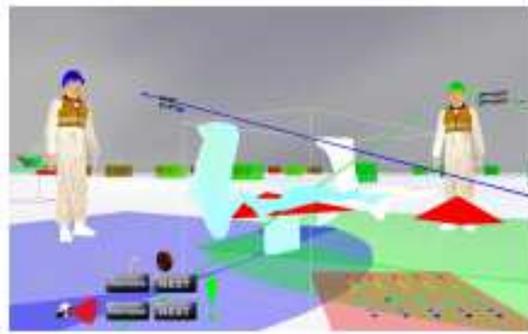
rectes au système, soit au travers d'un logiciel tiers. Kaltenbrunner *et al.* présentent TUIO, un protocole de communication développé pour les surfaces tactiles, [Kaltenbrunner *et al.*, 2005] et [Kaltenbrunner, 2009]. Ce protocole est spécifiquement défini pour être utilisé dans le cadre d'applications développées autour des interfaces tactiles et tangibles. Cette solution a été intégrée dans un kit de composants logiciels de vision par ordinateur pour permettre l'interaction objets / systèmes. Ce protocole est utilisable au travers de n'importe quel langage de programmation et application. Pour ce faire, il suffit simplement d'implémenter la solution d'écoute sur le client. Ainsi, le protocole est orienté autour d'un format spécifique des messages. Cette solution est intégrée au kit de composants logiciels utilisé pour la ReacTable et TDesk.

4.3.2 Collaboration

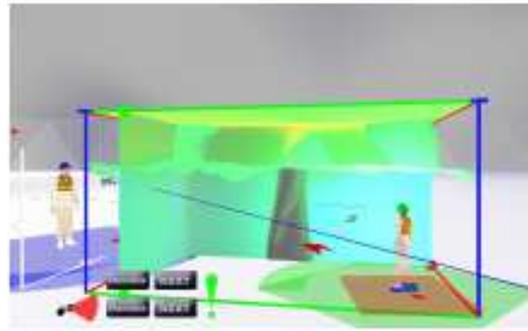
La collaboration entre différents utilisateurs au travers d'un même système informatique est une tâche complexe. En effet, les systèmes d'exploitation actuels sont orientés autour du paradigme WIMP (Windows Icône Menu Pointer) qui est mono-utilisateur. Les visualisations scientifiques nécessitent dans de nombreux cas d'être manipulées par différentes personnes simultanément. Dans cette section, nous allons présenter des approches d'interactions multi-utilisateurs pour la manipulation de vues complexes.

Duval *et al.* présentent une solution collaborative de visualisation scientifique, [Duval *et al.*, 2008]. Dans le cadre de leur application, les utilisateurs doivent partager leurs connaissances découvertes au travers d'une application dédiée. Les auteurs ont développé des outils de partage de contenu : chaque utilisateur contrôle son orientation, sa position dans l'espace et son échelle, et ces différentes informations sont transmises aux autres utilisateurs. Pour assister les utilisateurs dans cette navigation, les auteurs ont mis en place un système de curseur 2D et 3D. Leur solution donne la possibilité d'annoter des données complexes au travers d'une scène partagée par les différents utilisateurs. De plus afin de raffiner ces données, il est possible de réutiliser les données d'un autre utilisateur et de les annoter ensuite. En figure 4.4(a), la solution présentée montre le point de vue global d'un environnement paraphé avec deux utilisateurs qui manipulent un jeu de données au centre. La figure 4.4(b) présente une vue utilisateur de cette méthode de collaboration.

La plupart des systèmes de fenêtrage ne supportent qu'un seul curseur et donc un seul utilisateur, il s'agit du paradigme WIMP. Wallace *et al.* présentent une solution d'interface multi-curseurs et multi-utilisateurs, [Wallace *et al.*, 2004]. La problématique de ces travaux de recherche est illustrée en figure 4.5(a), elle consiste à visualiser et manipuler des informations sur plusieurs machines et d'agréger les différents résultats sur un écran central. Pour résoudre leur problématique, les auteurs ont mis en place un nouveau gestionnaire de fenêtre, spécialement conçu pour leur problématique. L'interface mise en place se concentre sur l'agrégation de vues et permet de gérer un environnement multi-curseurs.



(a) Visualisation globale de la scène.



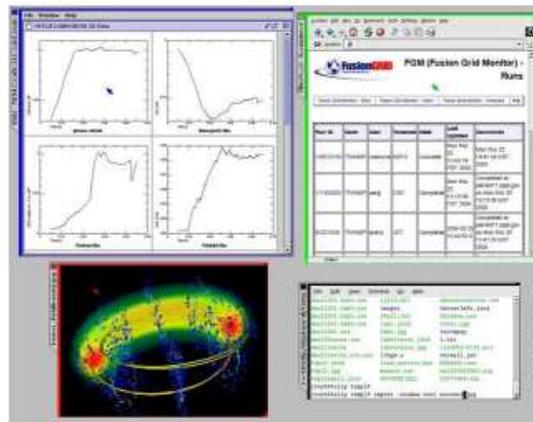
(b) Vue proposée à l'utilisateur.

FIGURE 4.4 – Exemple de rendu de la méthode [Duval *et al.*, 2008].

Cette solution permet à différents utilisateurs d'interagir entre eux au travers de vues différentes, mais au travers d'un même système de fenêtrage. La figure 4.5(b) présente le résultat de la vue centralisée : les différentes applications de visualisation de données utilisées sont agrégées sur le même écran et des couleurs sont utilisées pour savoir quel utilisateur est sur quelle vue.



(a) Un système d'interface multi-vues et multi-utilisateurs.



(b) Interface multi-utilisateurs composée des différentes vues des utilisateurs.

FIGURE 4.5 – Présentation de la méthode mise en place dans [Wallace *et al.*, 2004].

Dans cette section nous avons présenté les systèmes d'interactions de nouvelle génération, ainsi que des solutions multi-utilisateurs. Malgré leur efficacité et leur pertinence, ces différentes approches sont trop peu utilisées.

4.4 Des visualisations multi-vues

Enfin, nous nous sommes intéressés aux solutions d'analyse visuelle au travers de solutions d'analyse de données pour l'utilisateur ainsi que d'interface multi-vues. Dans cette section, nous allons traiter des approches de visualisation 2D, 3D mises en place, mais également de solutions d'analyse visuelle innovantes dont l'extraction de contenu est guidée par les utilisateurs.

4.4.1 Des outils d'analyse guidée par l'utilisateur

Dörk *et al.* proposent une solution de visualisation de données basée sur les relations entre les données, [Dörk *et al.*, 2011]. Les auteurs se reposent sur le principe de une visualisation de relation explicite et implicite. Les relations explicites sont des relations directes entre différentes données, les relations implicites sont des relations dérivées d'autres relations. Leur approche est basée sur deux visualisations : une chronologie et une carte d'intérêt. La carte d'intérêt représente de manière spatiale les données : les liens entre les données sont alors représentés par des flèches. Le style graphique utilisé pour la représentation des données et dépendant de leur redondance. Cette solution de visualisation est représentée en figure 4.6, la méthode représente les liens qui existent entre les philosophes, leur position dans l'espace sert à représenter la chronologie.

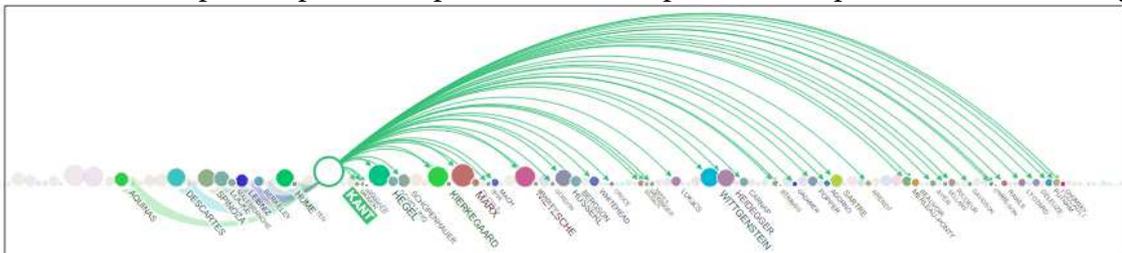


FIGURE 4.6 – Représentation chronologique de philosophes, [Dörk *et al.*, 2011].

Liu *et al.* travaillent sur la problématique de détection d'incident sur les réseaux informatiques. Les auteurs proposent un outil d'aide au diagnostic pour les pannes réseaux. Cette tâche reste une analyse complexe pour les administrateurs réseau, il est donc nécessaire de proposer des outils performants d'analyse. Cette solution est basée sur un outil de visualisation interactif combiné à un outil de fouille de données automatique. L'utilisation des capacités cognitives humaines ainsi que des capacités de traitement d'un ordinateur sont des enjeux majeurs pour proposer des outils d'analyse pertinents. L'utilisateur peut contrôler rapidement des informations produites par la méthode de fouille de données automatique mais également détecter des informations que le système n'aurait pas détecté. Les solutions actuelles développées pour les administrateurs réseau proposent des visualisations brutes des données, notamment des solutions plus adaptées à une utilisation humaine. L'approche développée au travers de ce système est orientée

autour de la multi-résolution : l'utilisateur peut définir un raffinement de la visualisation en fonction de ce qu'il désire comprendre. Pour comprendre cette méthode, la figure 4.7 présente la vue développée dans le cadre de ce projet, elle montre les différentes représentations des noeuds et des sous-réseaux ainsi que les liens entre ces sous-réseaux.

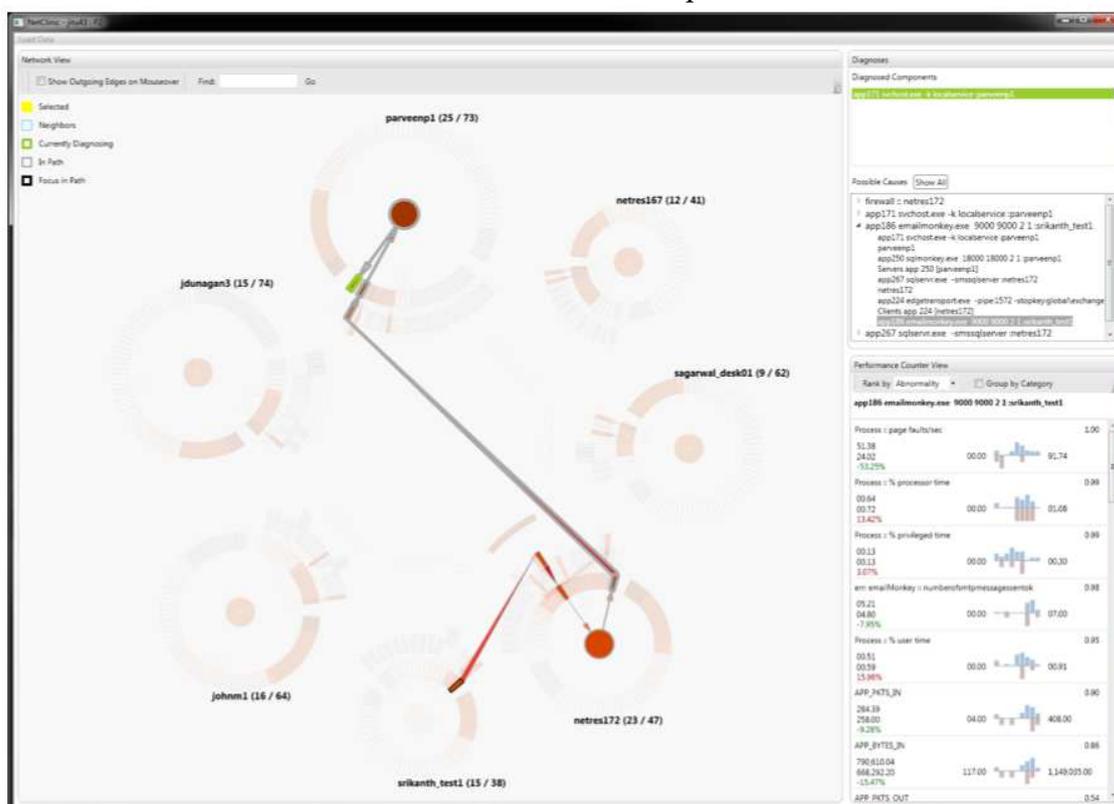


FIGURE 4.7 – Visualisation de données de réseau dans le cadre de l'application NetClinic, [Liu *et al.*, 2010].

Hao *et al.* ont proposé un outil de gestion de centre de calcul et d'analyse de données issues de ces centres, [Hao *et al.*, 2010]. Les centres de calculs sont des équipements complexes nécessitant un grand nombre d'optimisations énergétiques afin de réduire leur empreinte énergétique. Ainsi, pour pouvoir réaliser des optimisations structurelles, des applications dédiées sont développées. Ces applications sont orientées autour d'outils de visualisation. Pour des systèmes critiques, la remontée en temps réel d'informations est un des éléments majeurs pour conserver l'intégrité du système ; dans un cadre énergétique, ceci permet de réduire de manière efficace l'énergie consommée par ce type de système. Les données sont extraites de différents capteurs physiques qui permettent d'analyser le comportement du centre de calcul. Leur solution de visualisation est développée autour de la visualisation par nuage de points. Ils l'ont étendue ainsi afin de résoudre les différents problèmes de superposition. Leur mé-

thode de placement permet de facilement identifier les corrélations entre les différents capteurs. Les auteurs se sont notamment intéressés aux liens entre les données de températures et les systèmes de refroidissement. La figure 4.8 présente une carte des températures d'un centre de calcul. La solution qui est développée est destinée à des plateformes mobiles. En effet, il est important pour ce type de systèmes critiques de permettre la mobilité de ses utilisateurs pour réagir le plus rapidement possible.

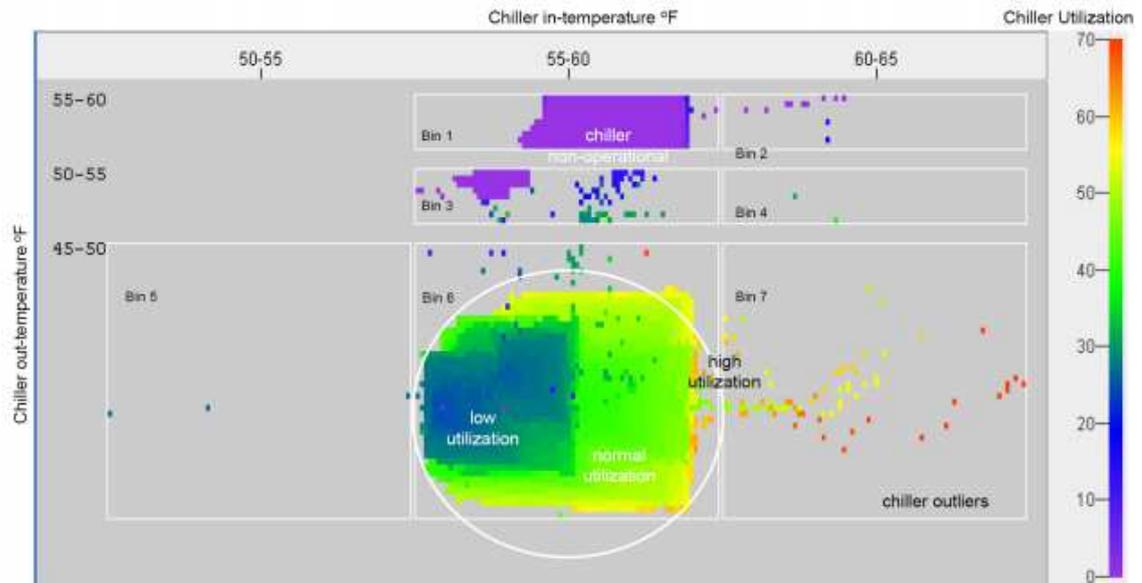
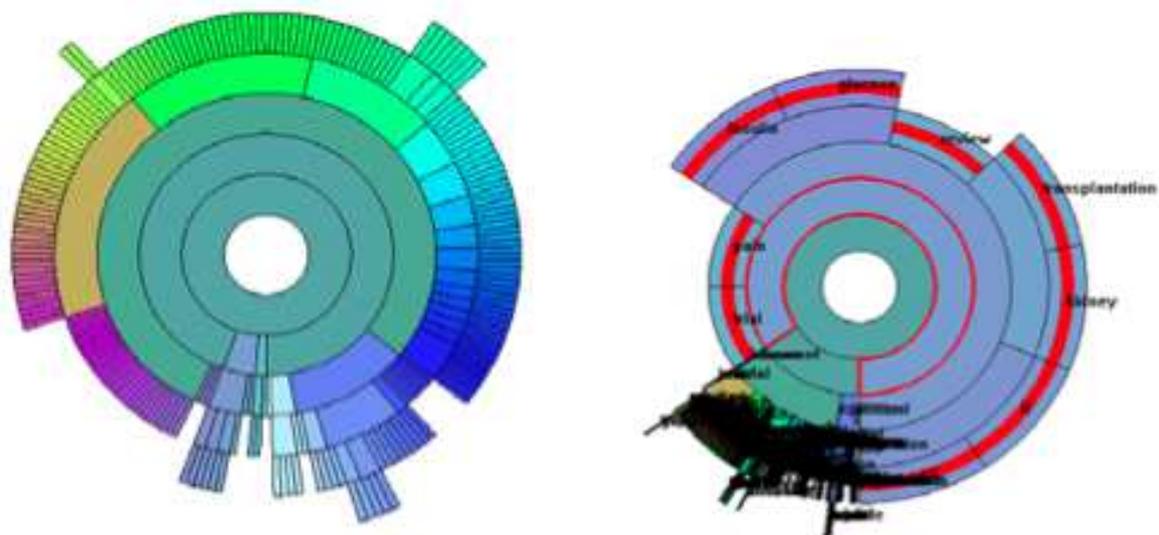


FIGURE 4.8 – Visualisation des données des climatiseurs d'un centre de calcul, [Hao *et al.*, 2010].

Enfin, Yang *et al.* ont développé une solution de visualisation 2D, [Yang *et al.*, 2003]. La méthode de visualisation utilisée est orientée tri, la méthode cherche à ordonner les différentes dimensions de la manière la plus pertinente pour le rendu. Manipuler et organiser les informations d'une visualisation sont des tâches cruciales car elles orientent la compréhension d'un résultat. La méthode développée dans cette approche est un outil de détection d'informations basé sur une hiérarchisation des différentes dimensions. Cette hiérarchie est produite par l'agrégation des dimensions. Une réorganisation automatique est proposée à l'utilisateur, mais il peut malgré tout réorganiser les dimensions en fonction de ce qu'il souhaite et également appliquer différentes transformations sur celles-ci. Leur approche est donc adaptable à différents types de visualisations, ce qui permet de mieux appréhender le contenu des données. L'approche standard de visualisation non traitée par l'utilisateur est présentée en figure 4.9(a). La figure présente la vue raffinée de cette méthode avec une mise en avant de certaines informations par l'utilisateur.



(a) Proposition de vue réalisée par l'approche automatique.

(b) Vue après le raffinement réalisé par l'utilisateur.

FIGURE 4.9 – Visualisation en anneau proposé par [Yang *et al.*, 2003].

4.4.2 Des visualisations 2D

Dans le domaine de l'analyse de données visuelle, de nombreuses méthodes de visualisation utilisées sont issues du domaine de la visualisation d'information. De plus celles-ci sont dans de nombreux cas orientées autour de représentations 2D. Dans cette section, nous allons nous intéresser aux approches multi-vues et 2D.

Roberts propose un état de l'art du domaine IV, basé sur un système de visualisation combinant des approches utilisant des systèmes à fenêtrage multiple, [Roberts, 2007]. Les premières méthodes multi-vues ont été développées afin de combiner différentes solutions de visualisation. Cette approche permet de proposer un outil efficace pour la fouille de données. Ces systèmes multi-vues utilisent des espaces liés. Les liens entre les données correspondent à des corrélations, afin de permettre à l'utilisateur de mieux comprendre les données, notamment au travers de différentes formes de représentation. Les approches multi-vues nécessitent un processus en différentes étapes afin d'être mises en place : stocker et manipuler les données, générer des vues et enfin interagir avec ces données. Le stockage et la manipulation de ces informations nécessitent des traitements particuliers afin de nettoyer le contenu, mais également de fusionner les différents contenus pour permettre l'extraction d'informations. Ce traitement utilise des méthodes issues de la fouille de données (KD pour Knowledge Disco-

very). Après l'étape de fouille de données, il est nécessaire de proposer une vue adaptée aux résultats. Enfin, des solutions d'interactions standards sont mises en place afin de permettre à l'utilisateur de fouiller correctement le contenu des données. Ainsi, la figure 4.10 présente différents points de vue d'un ensemble de données selon plusieurs moyens de représentations. Tout un ensemble de courbes a été mis en place dans cet outil.

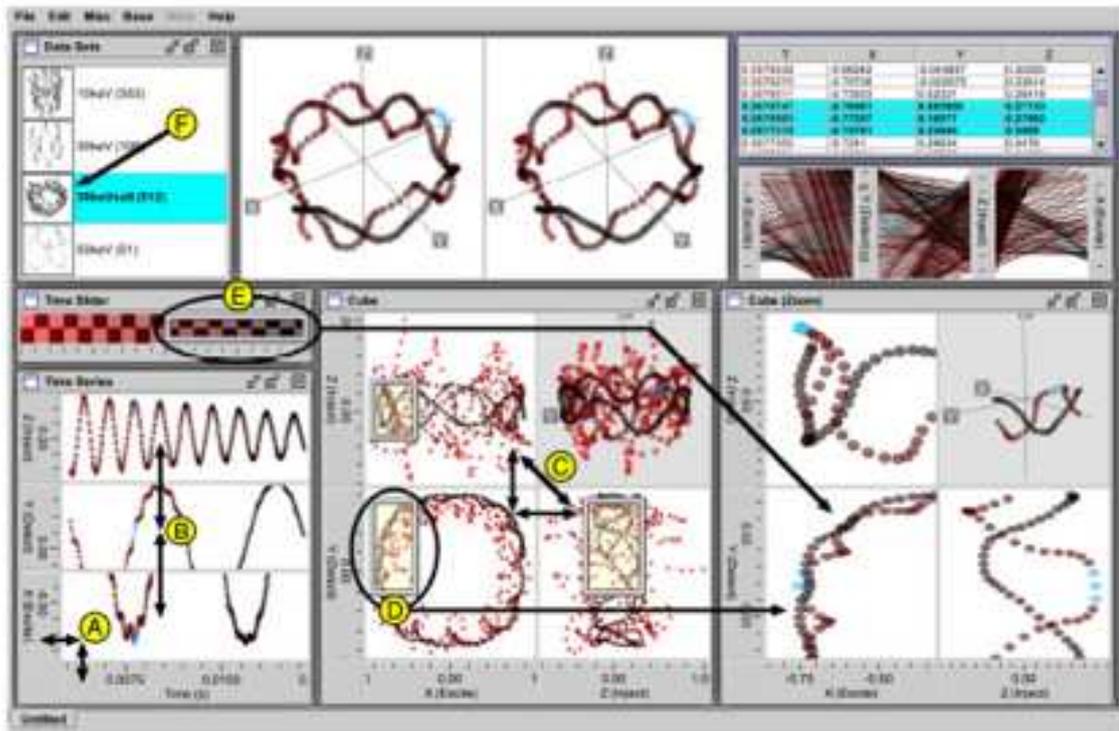


FIGURE 4.10 – Cette solution de visualisation permet de simuler la trajectoire d'un ion, [Roberts, 2007].

Seo et Shneiderman ont développé un outil de visualisation qui permet de générer des visualisations basées sur le classement par caractéristiques, [Seo et Shneiderman, 2005]. La plupart des nouvelles méthodes de visualisation pèchent au niveau de la compréhension utilisateur. Pour pallier ce problème, la solution permet de visualiser des jeux de données au travers de vues 1D ou 2D. Le but est de permettre de découvrir les relations, les agrégats, les valeurs aberrantes ou des caractéristiques particulières sur un jeu de données. Pour faciliter la compréhension, des méthodes de tri de données sont mises en place, leur principe repose sur des fonctions de corrélations, de régression linéaire/quadratique ou d'uniformisation des données. La projection des données au travers des différentes visualisations est définie par l'utilisateur. Les différentes vues proposées lui permettent alors de choisir les différentes classes d'information qu'il souhaite traiter. Grâce à cette méthode, il est possible de paramétrer efficacement la vue, mais également de détecter des nouvelles informations. Dans cet outil, la figure 4.11 présente un exemple de l'inter-

face proposée : les vues disponibles peuvent être paramétrées et ajoutées à la volée. Différentes vues issues de l'état de l'art sont disponibles comme le montre la représentation : les cartes de températures, les nuages de points ou encore la représentation par courbe.

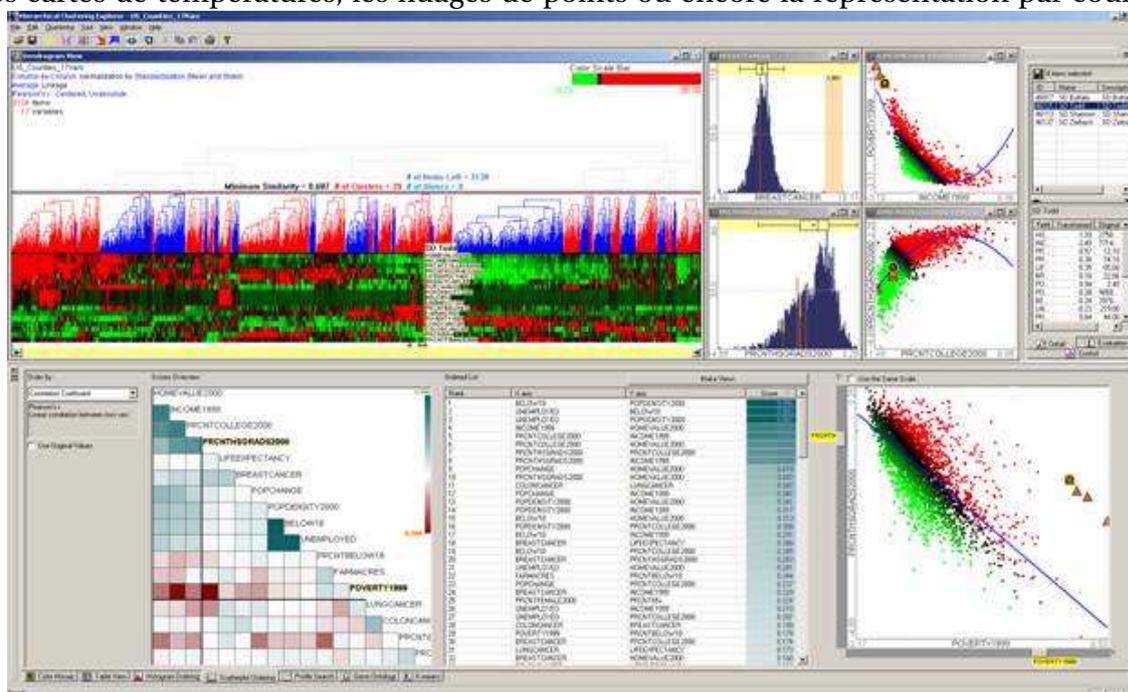


FIGURE 4.11 – La solution proposée a pour but de gérer une interface complète basée sur différentes solutions de visualisation pour permettre d'explorer les données complexes, [Seo et Shneiderman, 2005] .

Plusieurs travaux se sont intéressés à élaborer des extensions des systèmes de visualisation. Ainsi, dans l'article d'Aigner *et al.*, la solution proposée consiste à rajouter la notion de temps qui est une dimension importante dans les jeux de données, [Aigner *et al.*, 2008]. Cette information supplémentaire est partout autour de nous. Les données horodatées font partie des sources de données les plus récurrentes. Dans le cadre de ces travaux, les points de mesure sont issus de capteurs et le taux de rafraichissement est élevé. Les auteurs utilisent ainsi différentes méthodes de visualisation autour de fonctions. Ils ont donc étudié l'importance pour les systèmes de visualisation d'adapter les dimensions en fonction du temps. Des méthodes de type clustering sont mises en place pour fouiller de manière plus efficace les données et donc d'identifier des éléments au travers des données. Enfin ils intègrent l'utilisateur dans la boucle d'analyse pour extraire des informations pertinentes. La figure 4.12 présente les différents résultats de la méthode de visualisation, notamment

l'intégration de deux vues, une à base de courbe et la deuxième basée sur un planning.

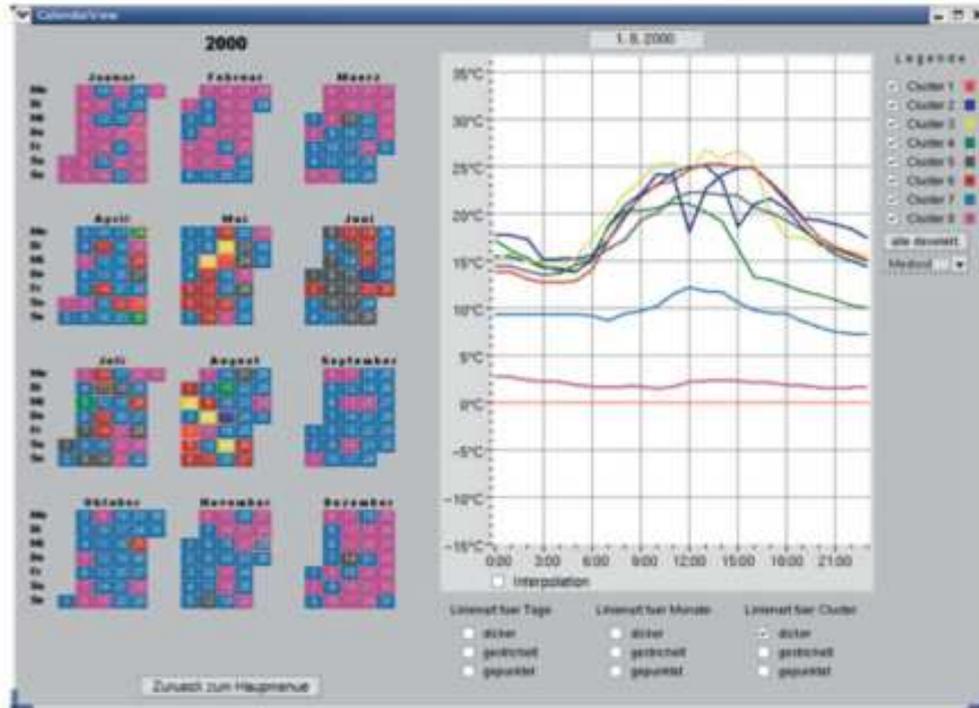


FIGURE 4.12 – Visualisation temporelle proposée, [Aigner *et al.*, 2008].

Migut et Worring présentent une solution d'analyse de données à risque, [Migut et Worring, 2010]. Les données à risques sont issues des domaines où la décision finale est cruciale : il est donc nécessaire de ne pas laisser la décision finale à une machine mais à un humain. Une machine ne doit jamais être responsable de décisions critiques. L'approche proposée dans cet article permet d'utiliser une série de vues interactives en 2D afin de visualiser les différentes informations. Une méthode automatique, basée sur un classifieur, est utilisée pour permettre de classer les données. Des outils de fouille de données par l'utilisateur sont ajoutés à cette vue afin de leur permettre de mieux interpréter le résultat proposé. La visualisation utilisée est basée sur une agrégation d'images afin d'associer les données à une personne physique. Ils couplent ainsi un système de visualisation basé sur des informations numériques, une fouille de données automatique et une image représentative. La figure 4.13 présente une vue du modèle de la solution mise en place, la visualisation est filtrée par l'utilisateur en se basant sur des outils d'exploration, ensuite la vue est connectée à une source de donnée externe. Enfin l'utilisateur peut explorer

ces données au travers d'un outil agglomérant différentes données de différentes sources.

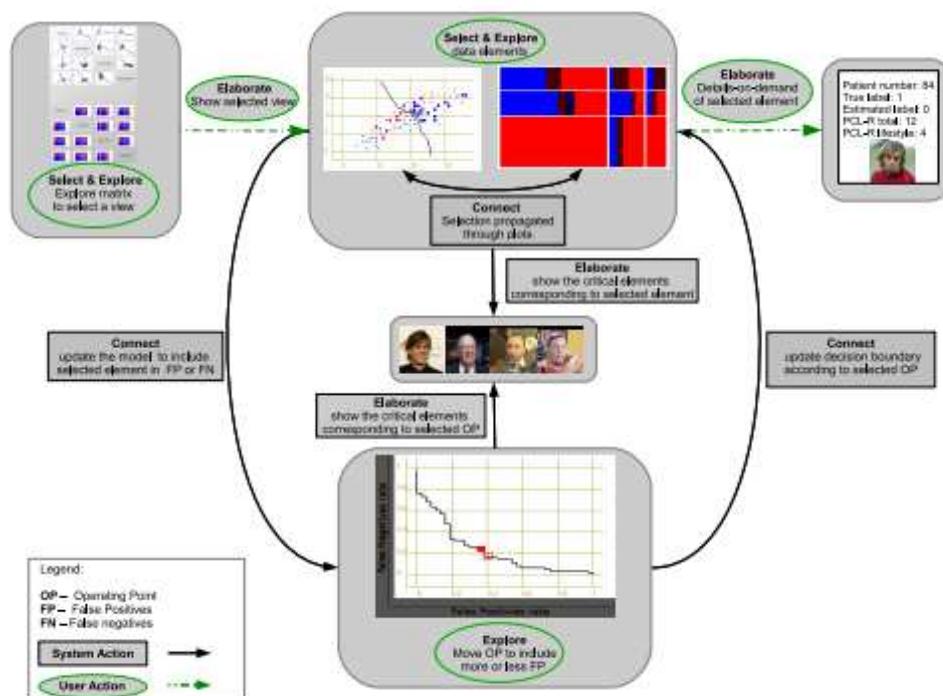


FIGURE 4.13 – Vue du kit de composants logiciels proposé pour la prise de décision basée sur une vue par photo, [Migut et Worrington, 2010].

4.4.3 Des visualisations 3D

Des méthodes se sont également intéressées à des visualisations 3D. Ainsi, dans cette section, nous allons présenter différentes approches liées aux outils de fouilles de données 3D. La première méthode est proposée par Lex *et al.*, [Lex *et al.*, 2010]. Leur travaux ont pour but d'assister les experts en biologie génétique dans l'élaboration d'hypothèses. Les données issues de ce domaine sont massives. La plupart des méthodes de visualisation pour ce domaine sont une représentation sous la forme de tableau. Dans certains cas, il s'agit d'outils de visualisation statistiques. Dans cet article, les auteurs présentent une méthode de visualisation développée pour leur application : Caleydo. Ils représentent les données sous forme de grand graphe combiné avec un ensemble de sous-graphes. Un système appelé Bucket basé sur une agrégation de fenêtres est utilisé. Deux approches de visualisation sont alors mises en place. La première approche utilise une heat map et la seconde approche est basée sur le principe des coordonnées parallèles. Différents outils d'interactions sont alors mis à la disposition de l'utilisateur tel que le zoom, le déplacement et la sélection. De plus, ils ont développé un outil appelé chemin qui est une représentation

visuelle appliquée à une donnée d'un lien entre différentes vues. Ainsi il devient possible de visualiser les liens entre les vues et de conserver l'information au travers des liens. La figure 4.14 illustre un exemple de cette application où 3 types de vues sont présentées : une vue en carte de chaleur, une vue sous forme de courbe et une vue sous forme de graphe. De plus, afin de corrélérer les données, des liens sont représentés entre les différentes vues.

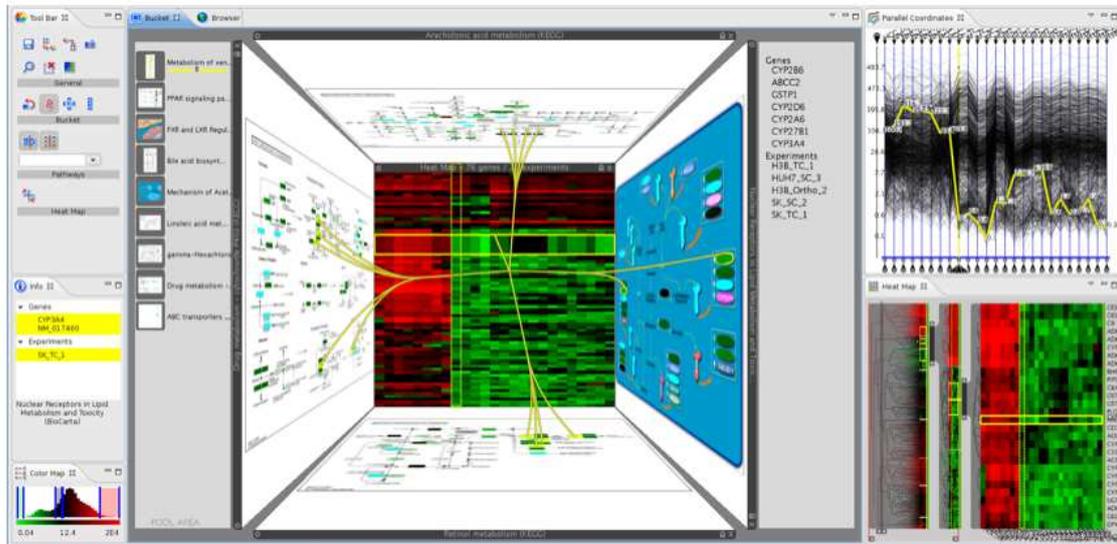


FIGURE 4.14 – Vue du système de bucket de la solution proposée dans [Lex *et al.*, 2010].

Piringer *et al.* ont amélioré un kit de composants logiciels de visualisation complexe de données. L'utilisation de nuage de points pour la visualisation souffre de problèmes de recouvrement : deux données peuvent être localisées sur le même pixel, [Piringer *et al.*, 2004]. Les auteurs se sont principalement intéressés à la mise en place d'une solution permettant de réduire le recouvrement des éléments. Elle consiste à passer vers une visualisation 3D, celle-ci permet alors de supprimer la majorité des problèmes de recouvrement. La partie interaction est réalisée sur la vue 2D et est ensuite appliquée à la vue 3D. Une vue de l'interface est donnée en figure 4.15, dans le cadre de cette solution, trois fenêtres sont dédiées à des données en 2D tandis que la quatrième fenêtre est la représen-

tation globale en 3D. Les vues 2D fonctionnent par coupe afin d'explorer le jeu de données.

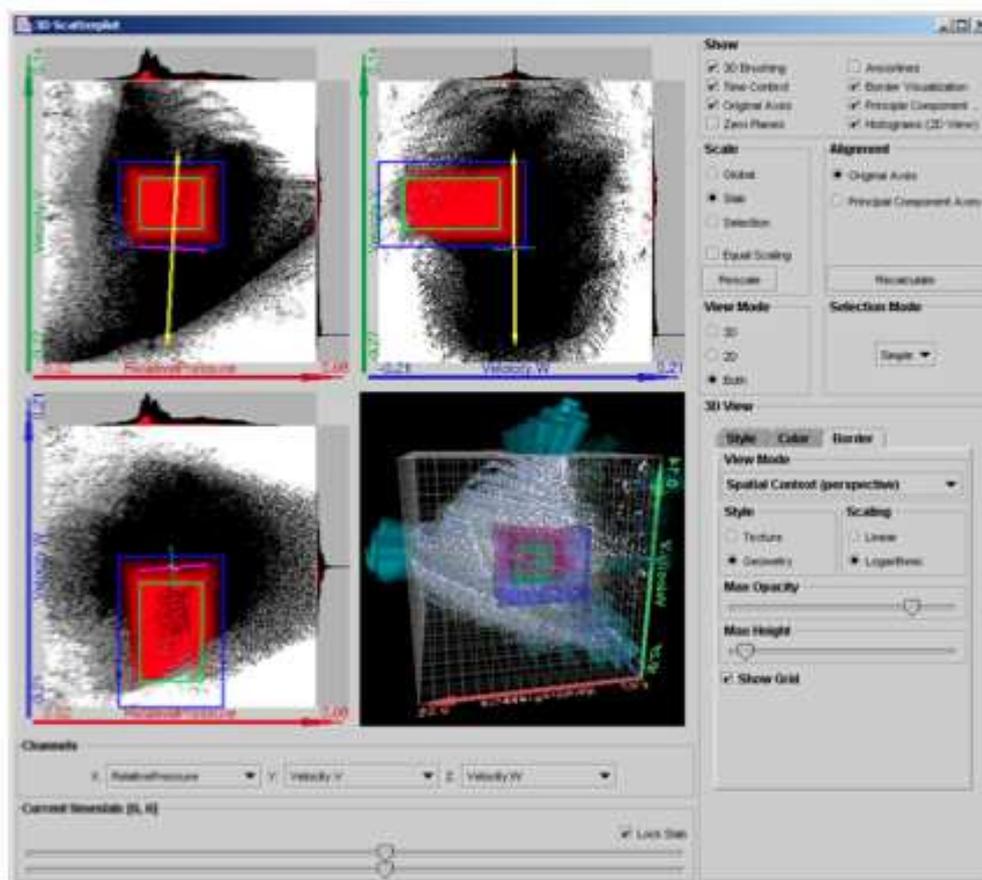


FIGURE 4.15 – Outil de visualisation de données sous la forme de nuage de points 3D, [Piringer *et al.*, 2004].

Les approches de visualisation traditionnelles souffrent de problèmes de lisibilité et d'interaction. L'article présenté par Yang *et al.* repose sur la visualisation multidimensionnelle basée sur les valeurs et les relations des éléments, [Yang *et al.*, 2004]. La solution mise en place est orientée autour de l'affichage par pixels; ils sont utilisés pour représenter les informations de manière condensée sur un écran. Des outils de navigation ont été mis en place pour permettre à l'utilisateur de diriger la représentation en appliquant des distorsions aux représentations. Des outils de raffinement des données ont également été ajoutés au travers de moyens de sélectionner des informations. Avoir un placement des glyphs correct dans un espace peut être un gage de bonne compréhension de ce type de représentation. Ainsi, la méthode proposée ici repose sur une représentation de corrélation par distance. Plus deux glyphs ont une forte corrélation plus ils sont proches. Des interactions standards ont été mises en place afin de permettre à l'utilisateur de naviguer dans cette visualisation comme par exemple

une solution d'exploration de données basée sur des informations spatiales (outil de zoom, déplacement). Un outil automatique se charge d'orienter le contenu des informations afin de permettre à l'utilisateur de réaliser une fouille de données plus efficace. Un rendu de cette méthode de visualisation par glyphe est présenté en figure 4.16.

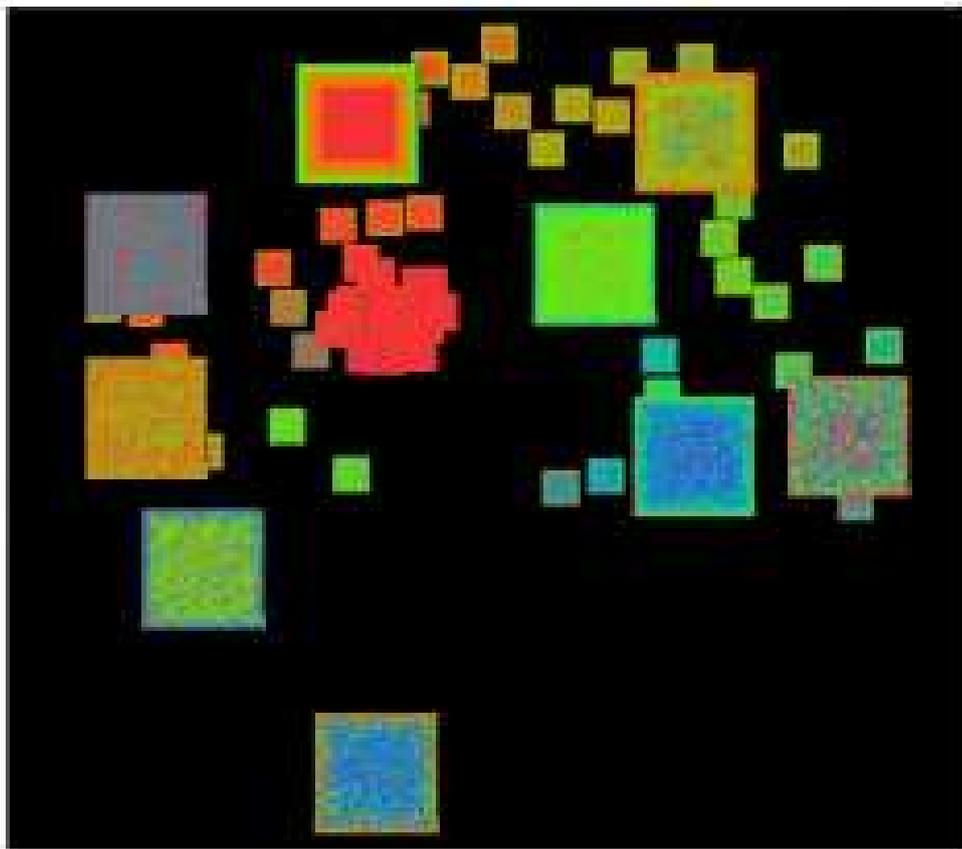


FIGURE 4.16 – Certains glyphes de cette visualisation sont élargis afin d'examiner les détails du contenu proposé, [Yang *et al.*, 2004].

Korsara *et al.* présentent un nouvel outil de visualisation pour la fouille de données, [Kosara *et al.*, 2004]. Leur outil est une composition d'éléments : un moteur de rendu volumique, un système multi-vues et enfin un système d'interaction entre les vues. Leur solution consiste à proposer une méthode de représentation par nuage de points 2D vers une vue 3D. Dans les environnements 2D, il est difficile de mettre en avant les volumes des éléments au travers de ce type de visualisation. En utilisant un paradigme 3D, il devient plus simple d'appréhender le volume de l'objet physique. Dans cette approche, les auteurs définissent ainsi une nouvelle solution de visualisation appelée voxelplot. Cette vue utilise des primitives représentant pour chaque point de vue une visualisation par voxel map. Grâce à cette méthode, ils peuvent utiliser un moteur de rendu volumique pour re-

présenter une vue globale de leurs données. La mise en service de leur visualisation passe par différentes étapes. Les données sont chargées en mémoire afin d'être utilisables dans l'environnement RTVR : il s'agit d'un moteur de rendu volumique, [Mroz et Hauser, 2001]. Le système de Korsara *et al.* repose sur un système multi-vues, ainsi une fois les données chargées, l'utilisateur définit les différentes dimensions qu'il souhaite visualiser pour chaque vue. Malheureusement, la solution mise en place ne permet pas un rendu satisfaisant pour la compréhension humaine, le rendu reste complexe comme le montre la figure 4.17. Le deuxième inconvénient est inhérent à la bibliothèque utilisée : il n'y a pas d'indice de profondeur autre que la transparence. L'utilisateur peut interagir avec le contenu de la vue au travers d'un ensemble d'interactions standards qui sont proposées. Les auteurs se servent d'un panneau de contrôle dédié pour manipuler les différentes informations des vues. Cette application a été expérimentée sur un jeu de données issues de capteurs placés dans un pot d'échappement, la forme du pot est visible en haut à gauche de la figure 4.17.

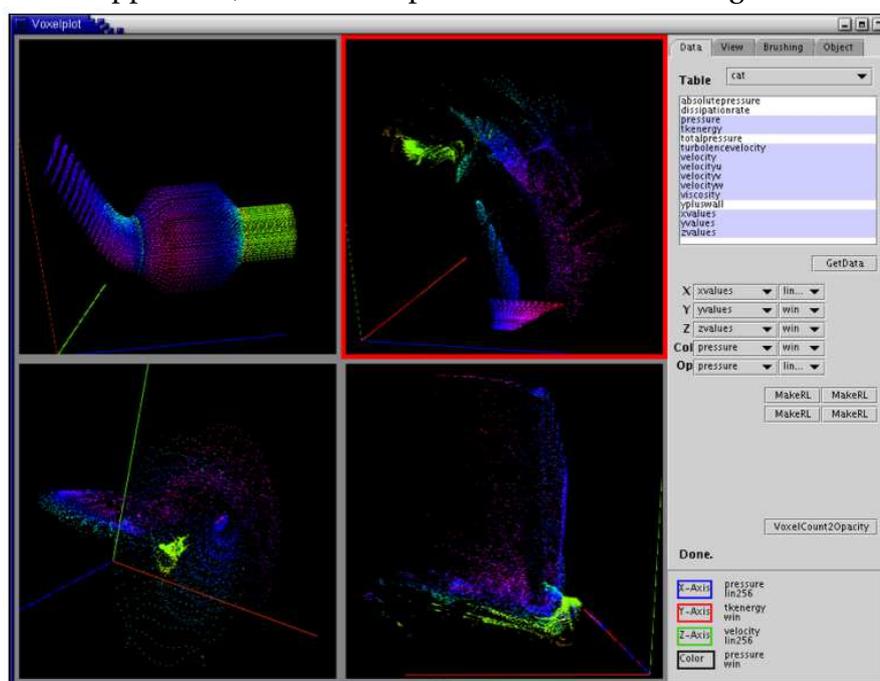


FIGURE 4.17 – Visualisation par Voxelpilot de données issues d'un pot d'échappement, [Korsara *et al.*, 2004].

Les méthodes de visualisation de données doivent permettre de mieux comprendre les visualisations, mais surtout d'extraire de nouvelles informations sur le contenu des données. Ainsi, Fuchs *et al.* proposent une nouvelle approche pour intégrer l'utilisateur dans la fouille de données en développant un outil de visualisation combiné avec des méthodes de fouille automatique, [Fuchs *et al.*, 2009]. Le but est d'utiliser les capacités cognitives humaines pour arriver à résoudre des problèmes

complexes. L'application se sert de l'utilisateur afin de valider les hypothèses produites par la méthode automatique de fouille de données. Une fois générées, les hypothèses seront par la suite visualisées au travers de différentes représentations qui pourront être manipulées par l'utilisateur. L'utilisateur pourra ainsi facilement étudier ses données au travers de différentes vues comme illustré en figure 4.18. Dans cet exemple, les vues agrégées sont des représentations temporelles d'un jeu de données. Il est possible d'observer le comportement de fluide dans le cadre d'une simulation.

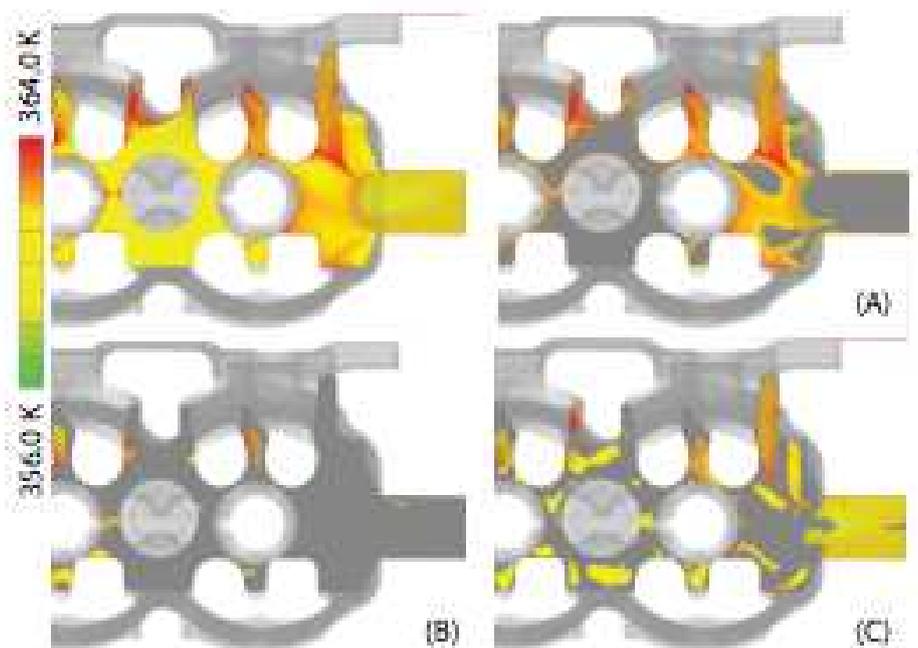


FIGURE 4.18 – Visualisation basée sur la compréhension de données, [Fuchs *et al.*, 2009].

4.5 Synthèse

Dans ce chapitre, nous avons étudié différentes solutions développées autour du domaine de l'analyse visuelle. Le cœur de ces travaux consiste à développer des solutions de visualisation permettant de fouiller efficacement les données. Le fait d'adapter les méthodes de visualisation promet un gage de qualité dans les solutions de fouille de données. Il est nécessaire de proposer des solutions de visualisation les plus simples possibles, afin que tous types d'utilisateurs soient capables de les utiliser. D'autre part, les données fouillées dans le cadre de ce domaine sont très souvent complexes (en terme de dimensions différentes) et de grande taille. Il est alors nécessaire d'adapter les visualisations pour réduire la complexité des vues.

Nous avons également détaillé les méthodes d'interactions pour la fouille de données. Elles sont utilisées afin de fournir un contenu pertinent. Il est nécessaire de développer des outils adaptés, qu'ils soient basés sur des dispositifs particuliers ou sur des paradigmes complexes. Grâce à tout cela, il est alors possible d'extraire des corrélations sur les données et donc de mieux les comprendre.

Enfin, nous avons présenté différentes solutions de gestion de données complexes. Il est à retenir que ces solutions sont dans de nombreux cas dissociées de la partie traitement pour la visualisation. En effet, le traitement des données est consommateur en terme de ressources et ne doit pas ralentir la visualisation. Nous avons donc choisi d'utiliser cette méthode dans les travaux que nous avons menés. De plus afin d'optimiser les manipulations des données, il est nécessaire de trouver des solutions efficace pour l'interaction notamment au travers d'une interface post-WIMP [Van Dam, 1997]. Enfin, les solutions de visualisation adaptées à la découverte d'information doivent permettre de visualiser les données de manière multi-vues, c'est à dire combiner plusieurs solutions de visualisation avec des liens forts entre elles.

Dans la seconde partie de ce manuscrit et en particulier dans le chapitre 6, nous verrons comment nous avons exploité ces différentes méthodes d'analyse visuelle.

Deuxième partie

Contributions

Modélisation de bâtiment

Comme la Hongrie, le monde informatique a une langue qui lui est propre. Mais il y a une différence. Si vous restez assez longtemps avec des Hongrois, vous finirez bien par comprendre de quoi ils parlent.

D. BARRY

Sommaire

5.1	Introduction	71
5.2	Paradigme client / serveur	72
5.3	Modélisation de l'espace basée sur des particules	73
5.4	Interpolation basée capteur	80
5.5	Conclusion	84

5.1 Introduction

Dans le cadre du projet RIDER, présenté dans le chapitre 1 de ce manuscrit, différents composants ont été développés afin de fournir des informations au modèle de données du noyau de l'architecture. Dans ce chapitre nous nous intéressons à la méthode que nous

avons mise en place pour créer la base de notre visualisation interactive de données hétérogènes. Cette visualisation repose sur différentes briques articulées autour d'une plateforme client / serveur. Les différentes briques de cette méthode de visualisation sont : une approche modélisation des données issues du bâtiment et une brique de rendu. Le rendu produit cherche à se rapprocher au plus près de ce que les utilisateurs connaissent en terme de visualisation, plutôt que de réaliser des visualisations trop complexes et donc non efficace comme nous avons pu le montrer dans le chapitre 4. Notre solution de visualisation propose à n'importe quel utilisateur de comprendre correctement les vues après une très courte période d'apprentissage.

Différentes contraintes sont liées au cadre de ces travaux, notamment la simplicité en terme de visualisation et une contrainte de calculs en temps réel. Un système de visualisation interactif est un système principalement contraint par une réactivité de l'ordre de 30 ms, pour une visualisation la contrainte se compte en nombre d'images. Ainsi, comme le souligne Madougou *et al.*, le nombre d'images produites doit être égal ou supérieur à 30 images par seconde, [Madougou *et al.*, 2006]. Il est également nécessaire de trouver des solutions de gestion du modèle de données efficaces. Pour ce faire, nous nous sommes basés sur les solutions fragmentées en une brique gestion de données et une brique visualisation.

Dans ce chapitre, nous nous intéressons donc à la mise en place du paradigme client / serveur en fonction des besoins de l'architecture de RIDER. Nous présentons cette architecture en section 5.2. Ensuite, nous nous intéressons dans la section 5.3 à la modélisation d'espace pour le bâtiment. Et enfin, nous nous intéressons à l'interpolation de cet espace au travers de la section 5.4.

Dans les différentes figures de ce chapitre, nous allons présenter une représentation simplifiée de notre solution de modélisation d'espace, notamment au travers de vues 2D. Nous utiliserons une symbolique bien définie : les triangles représentent les capteurs du bâtiment et les cercles, quant à eux, représentent notre support de visualisation (nous l'appellerons particule). Les différentes couleurs servent à donner des informations relatives à la segmentation de nos vues.

5.2 Paradigme client / serveur

Comme nous l'avons présenté dans l'introduction de ce manuscrit, l'architecture de RIDER est un agrégat de différents composants. Nous avons choisi de développer notre composant de visualisation autour du paradigme client / serveur. Le but est de permettre de déporter une partie des calculs dans un environnement tiers qui est représenté par le serveur. La partie client, quant à elle, est chargée de produire la vue pour l'interface utilisateur.

Notre solution de visualisation est basée sur deux briques logicielles. La partie serveur du système possède l'information du modèle général que nous avons utilisé, ainsi, tous les traitements et calculs lourds sont réalisés sur cette plateforme. La partie client, quant à elle, est destinée à supporter seulement la visualisation de ces données ainsi que l'interaction.

5.3 Modélisation de l'espace basée sur des particules

L'information qui nous est fournie dans le cadre du projet RIDER peut varier en fonction du système de contrôle et régulation du bâtiment (BMS : Building Management System). Il est donc nécessaire de développer un modèle de données capable de traiter efficacement l'hétérogénéité des données.

Dans le cadre de la visualisation, nous avons mis en place un modèle simplifié du modèle pivot (modèle de transfert de données) contenu dans le cœur de RIDER. Ce modèle possède un certain nombre d'attributs relatifs aux bâtiments, mais également des attributs relatifs aux pièces qui sont dans le bâtiment. Chaque pièce possède différentes structures, que ce soit du mobilier ou des capteurs. Ainsi, un bâtiment est composé d'une localisation dans l'espace, d'une enveloppe englobante et d'un agrégat de pièces. Les pièces sont composées d'une position relative au bâtiment, d'une enveloppe englobante, d'un ensemble de points qui représentent la topologie de la pièce et enfin d'un ensemble d'éléments. Ces éléments peuvent être de différents types : soit capteur, soit mobilier. Ils sont représentés par une position dans l'espace et un identifiant. Un élément de type capteur est de plus représenté par une valeur et un type. Enfin l'élément de type équipement est quant à lui représenté par une enveloppe englobante.

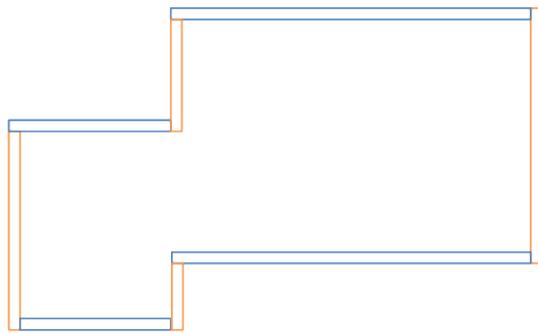
Notre projet est destiné à produire un rendu visuel 3D de ces données. Ainsi, nous avons cherché à extraire la représentation de bâtiment en 3D compatible avec les différents systèmes de gestion du bâtiment. De nombreux travaux de visualisation appliqués au bâtiment utilisent soit des méthodes 2D, soit des environnements 3D avec un modèle de bâtiment sur lesquels les informations mesurées sont appliquées sur l'objet sous la forme de textures. Ces deux types de visualisation ne sont pas suffisamment efficaces : dans le cadre d'une vue 2D, les informations relatives à la hauteur sont perdues, dans le cadre de textures appliquées sur un modèle, l'information relative à l'intérieur des éléments est perdue. De plus, nous souhaitons réaliser une visualisation utilisable sur différents types de plateformes qu'elles soient mobiles, de bureau ou adaptées à un centre de calcul. Pour ce faire, chaque pièce est représentée par un élément appelé espace pour le modèle IFC (Industry Foundation Classes) ou zone homogène dans TrnSys¹. Cet espace, va nous servir de fil conducteur pour la visualisation, notamment au travers d'une méthode de remplissage de celui-ci par des particules.

1. <http://www.trnsys.com>

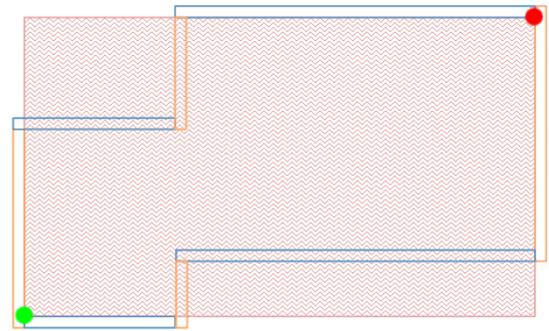
Nous nous sommes intéressés à l'utilisation de particules statiques afin de proposer un rendu volumique multi-résolution pour chaque pièce du bâtiment. Comme nous l'avons vu précédemment, il existe principalement trois méthodes spécifiques d'utilisation de particules. Nous utilisons la méthode statique, moins coûteuse en terme de calcul que les deux autres méthodes. De plus, cette méthode est très efficace pour créer le rendu d'objets 3D de grande taille. L'utilisation des particules nous permet également d'adapter la visualisation au support d'affichage. Chaque particule peut être colorée en fonction de différents paramètres. Ce type de rendu à base de particules est un avantage important en terme de temps de calcul : contrairement aux particules dynamiques le temps de rafraîchissement est largement réduit. Ainsi, la contrainte du taux de rafraîchissement minimum de notre visualisation est respectée. Nos particules vont donc être utilisées comme support de visualisation volumique. Elle seront déposées dans l'espace intérieur de chaque pièce, notamment en fonction de l'équipement compris dans l'espace de la pièce. L'espace d'une pièce est défini dans le modèle IFC comme étant une représentation du volume de chaque pièce, ce volume est contraint par la topologie de la pièce.

Les espaces sont bornés en une forme simpliste représentée par un pavé. Dans notre système, nous avons voulu pallier le problème d'une représentation trop simple en utilisant un système hiérarchique. Les espaces que nous pouvons représenter grâce à ce modèle seront alors plus proches de la réalité. Notre modèle hiérarchique est défini de la manière suivante : nous avons un espace grossier modélisé par l'enveloppe englobante de l'espace et un modèle plus fin représenté par un schéma du modèle de la pièce. L'espace grossier est utilisé pour générer les particules rapidement, il est calculé en utilisant la méthode AABB (Axis Aligned Boundary Box), présenté dans [Van Den Bergen, 1998]. La figure 5.1(b) présente la solution d'enveloppe englobante d'une pièce complexe, pour l'extraire la méthode repose sur une extraction des valeurs minimum (point vert) et maximum (point rouge) de la pièce. La figure 5.1(c) représente plus fidèlement la surface au sol en définissant un ensemble de jalons (points de couleur rouge), qui correspond aux points de déflexion des différents murs, ordonnés dans un sens anti-horaire.

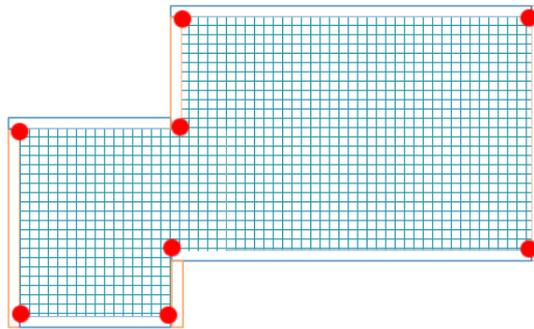
Pour modéliser notre méthode, il est nécessaire de définir certaines informations. Dans notre modèle, une coordonnée spatiale est représentée par $c = \{x, y, z\}$ avec $x, y, z \in \mathbb{R}^+$. Un bâtiment peut être représenté sous la forme suivante : $b_i = \{c_{i_{loc}}, c_{i_{max}}\}$, avec $c_{i_{loc}}$ et $c_{i_{max}}$ les coordonnées extrémales de ce bâtiment, avec $i \in \mathbb{N}$, $1 \leq i \leq NB_{bmax}$ et NB_{bmax} représente le nombre maximum de bâtiments. A partir de ces deux points, il est possible de localiser et reformer l'enveloppe englobante du bâtiment. Chaque bâtiment contient plusieurs salles et donc plusieurs espaces (un par salle). Chaque espace peut être décomposé de la manière suivante : une enveloppe englobante et la liste des différents jalons (les différents sommets du périmètre) de la topologie de la pièce. Un espace est modélisé de la manière suivante : $r_j = \{c_{j_{loc}}, c_{j_{max}}, v_{j_0}, \dots, v_{j_k}\}$, avec $c_{j_{loc}}$ et $c_{j_{max}}$ les points extrémaux de l'espace relatifs aux coordonnées du bâtiment. Enfin, v_{j_0}, \dots, v_{j_k} représentent les différents jalons



(a) Schéma de la topologie d'une pièce.



(b) Schéma de l'enveloppe englobante de la pièce présentée en figure 5.1(a).



(c) Schéma de l'espace de la pièce présentée en figure 5.1(a).

FIGURE 5.1 – Représentation multi-résolution d'une pièce standard.

utilisés pour modéliser la pièce, la valeur k représente le nombre de jalons d'une pièce. Pour l'exemple de la figure 5.1(a) ce nombre est de 8. La valeur de j avec $j \in \mathbb{N}$ est le nombre de pièces, ce paramètre appartient à l'intervalle : $1 \leq j \leq \text{NB}_{\text{rmax}}$, avec NB_{rmax} le nombre maximum de pièces. La figure 5.2 présente le principe de coordonnées dans le bâtiment au travers d'une représentation simple d'un bâtiment.

Une fois le modèle de la pièce défini, il est nécessaire de rajouter les particules pour définir l'espace de la pièce et donc le nouveau support d'information. Ainsi, nous avons mis en place trois méthodes de modélisation d'espace discret : un modèle en grille régulière, un modèle concentrique et enfin un modèle basé sur la division d'espace. De plus, afin d'être compatible avec différents périphériques de visualisation, nous avons étudié le paradigme LOD introduit par Clark [Clark, 1976]. Cette méthode consiste à adapter la résolution d'un objet 3D en fonction de la distance de celui-ci avec la caméra de la scène 3D. Dans notre projet, nous avons utilisé ce paradigme en adaptant la résolution de notre objet 3D en fonction du périphérique. Chaque périphérique possède ses propres caracté-

ristiques et capacités de rendu. Nous introduisons le paramètre ϕ qui permet de gérer les différentes résolutions pour les modélisations proposées.

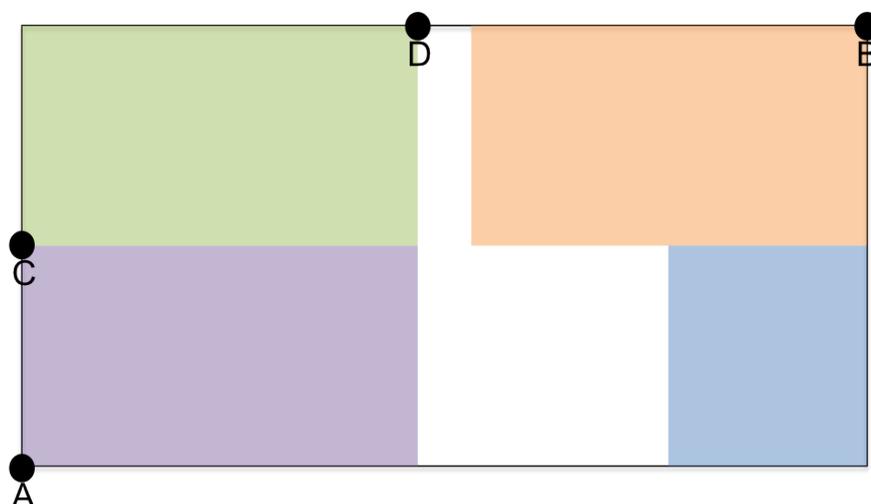


FIGURE 5.2 – Représentation du modèle de coordonnées utilisé par notre visualisation, le point A correspond à la localisation initiale de notre bâtiment et représente le point d'origine du repère, le point B est comme le point A un point extrémal du rectangle formant l'enveloppe englobante du bâtiment. Le point C est le point d'origine de la pièce verte et enfin le point D détermine avec le point C l'enveloppe englobante de la pièce.

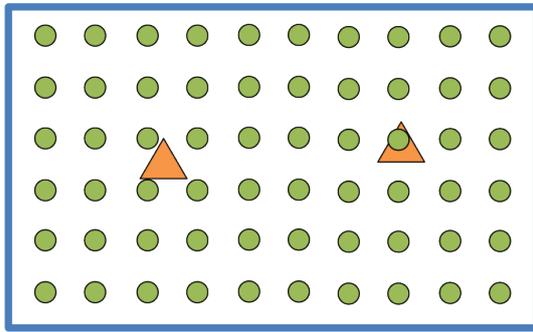
5.3.1 Modèle en grille régulière

Le premier modèle de représentation de volume basé particules utilise une grille régulière pour déposer les particules dans l'espace de la pièce. Ceci nous permet d'avoir une distribution uniforme des particules. La valeur de ϕ est bornée par la plus petite mesure de l'enveloppe englobante de l'espace : $\forall r_j, \phi \leq \min(x_{j_{\max}}, y_{j_{\max}}, z_{j_{\max}})$.

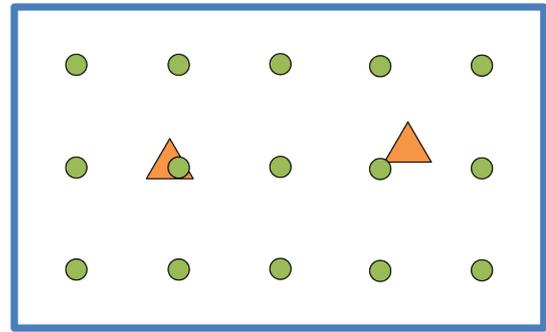
Les figures 5.3(a) et 5.3(b) montrent deux exemples de résolution possible avec un ϕ différent : un est adapté à un périphérique qui supporte peu de primitives graphiques (particules), tandis que le second supporte beaucoup plus de primitives. Ce paramètre nous permet donc d'adapter de manière efficace la résolution désirée pour un périphérique.

L'ensemble de particules généré a été modélisé de la manière suivante : $P = \{p_1, \dots, p_{nbP_j}\}$, avec nbP_j le nombre de particules généré par cette représentation et $nbP_j \in \mathbb{N}$.

Chaque particule est modélisée de la manière suivante : $p_k = \{x_k, y_k, z_k\}$ et $k \leq nbP_j$, avec $0 \leq x_k \leq x_{j_{\max}}$, $0 \leq y_k \leq y_{j_{\max}}$, $0 \leq z_k \leq z_{j_{\max}}$. Les coordonnées des particules sont



(a) Grille régulière avec un ϕ faible (résolution du nuage de particules élevée).



(b) Grille régulière avec un ϕ élevé (résolution du nuage de particules faible).

FIGURE 5.3 – Représentation multi-résolution de l'espace d'une pièce.

calculées grâce à la méthode suivante :

$$\forall k \in \mathbb{N}, k \leq nbP_j, p_k = \begin{cases} x_k = b \cdot \phi \\ y_k = c \cdot \phi \\ z_k = d \cdot \phi \end{cases}$$

avec $b \in \mathbb{N}, 0 \leq b \leq x_{j_{max}}, c \in \mathbb{N}, 0 \leq c \leq y_{j_{max}}, d \in \mathbb{N}, 0 \leq d \leq z_{j_{max}}$ et b, c, d des valeurs d'incrément. Afin de déterminer le nombre de particules, il est nécessaire d'utiliser la formule suivante :

$$\forall r_j, nbP_j = \frac{((x_{j_{max}} + 1) \times (y_{j_{max}} + 1) \times (z_{j_{max}} + 1))}{\phi^3}$$

Grâce à cette méthode, l'espace de la pièce est rempli de particules de manière régulière. Les zones éloignées des capteurs sont également couvertes. Certaines techniques de visualisation que nous verrons plus tard ne pourront être appliquées que sur cette méthode de construction.

5.3.2 Modèle concentrique

Avec ce modèle, nous utilisons un ensemble de sphères centrées sur les capteurs pour produire les particules. Les particules sont déposées à la surface de chaque sphère. Ainsi l'information produite est localisée au plus proche de capteurs. Différents paramètres peuvent varier sur cette méthode : l'espace entre deux sphères et la localisation des particules à la surface de la sphère. Pour notre méthode de visualisation, nous nous sommes servis de l'indice ϕ pour faire varier la distance entre les différentes sphères. Ainsi, plus le nombre de sphères est important, plus le nombre de particules sera élevé. En effet, dans

notre approche, le nombre de particules par sphère reste constant. Pour générer les coordonnées des particules dans l'espace j , nous résolvons l'équation de la sphère en fonction de la localisation des capteurs. Soit nbS_j le nombre de capteur de l'espace r_j . Soit la constante $nbpc$, le nombre de particules par sphère de capteur. Le nombre de particules ainsi généré dépend du nombre de sphères concentriques insérées dans notre pièce.

$$\forall i \in \mathbb{N}, i \leq nbS_j, \forall k \in \mathbb{N}, k \leq nbpc, p_k = \begin{cases} x_k = x_{s_i} + \text{rayon} \cdot \sin \theta \cdot \cos \rho \\ y_k = y_{s_i} + \text{rayon} \cdot \sin \theta \cdot \sin \rho \\ z_k = z_{s_i} + \text{rayon} \cdot \cos \theta \end{cases}$$

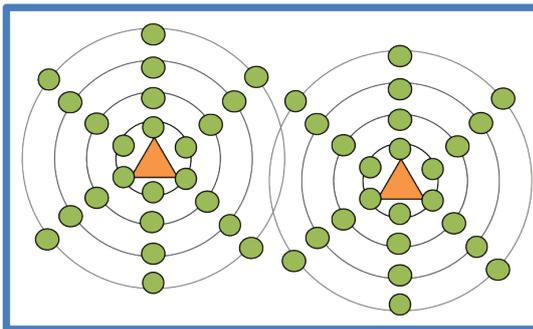
avec

$$\begin{cases} \text{rayon le rayon de la sphère} \\ 0 \leq \theta \leq 2\pi, & \theta \in \mathbb{R} \\ 0 \leq \rho \leq \pi, & \rho \in \mathbb{R} \end{cases}$$

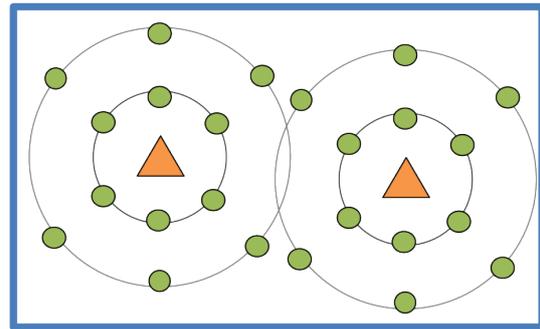
Afin d'avoir une résolution très faible, le paramètre ϕ devra varier de manière à réduire le nombre de couches produites. Afin de réduire la résolution, il faut augmenter la valeur de ce paramètre. Etant donné la topologie d'une pièce, le nombre de particules ne peut être qu'approximatif théoriquement, ainsi pour calculer le nombre maximal de particules, l'inéquation suivante est utilisée :

$$\forall r_j, nbP_j \leq x_{j_{\max}} \times y_{j_{\max}} \times z_{j_{\max}}$$

Une représentation à différentes résolutions est proposée en figures 5.4(a) et 5.4(b). En



(a) Utilisation de sphères concentriques avec un paramètre ϕ faible.



(b) Utilisation de sphère concentrique avec un paramètre ϕ élevé.

FIGURE 5.4 – Représentation multi-résolution de la solution de création de particules par sphère concentrique.

utilisant une méthode de génération statique, le nombre de particules n'est pas optimisé, certaines couches (notamment celles placées proche des extrémités de la pièce) auront une concentration réduite. Nous avons mis en place des méthodes itératives afin de résoudre ce problème. Si une particule se trouve à l'extérieur de l'espace de la pièce, alors il est nécessaire de lui assigner une nouvelle position.

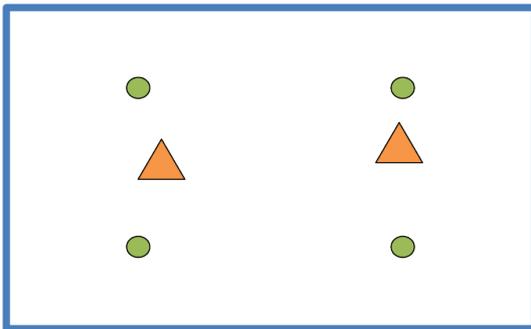
Grâce à cette méthode basée sur un ensemble de sphères concentriques, un espace plus précis est généré aux alentours de capteurs. Mais les techniques de visualisation que nous avons mises en place ne seront pas toujours adaptées à cette méthode de représentation.

5.3.3 Modèle multi-résolution

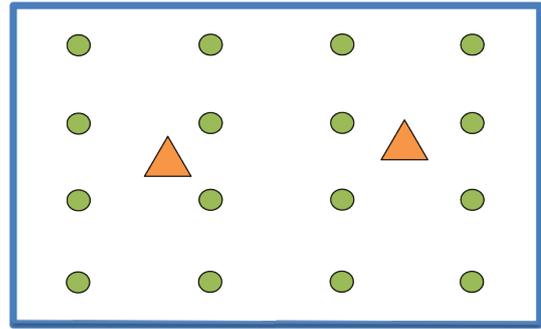
La dernière méthode de construction que nous avons utilisée est basée sur une subdivision de l'espace. Chaque espace est divisé en huit parties équivalentes (en 2D quatre parties équivalentes). Une particule est déposée dans le centre de chaque cellule. Ensuite, chaque cellule est re-divisée en huit parties équivalentes et ainsi de suite. Cette méthode est inspirée des octrees, qui est une méthode de subdivision spatiale. Le nombre de particules dépend de deux paramètres : ϕ et le nombre de divisions de l'espace. Plus la valeur de ϕ est élevé, plus le nombre de particules généré est important :

$$\forall r_j, nbP_j = nbD^\phi$$

avec nbD représentant le nombre de division de l'espace. Les figures 5.5(a) et 5.5(b) présente une schématisation de l'espace d'une pièce par méthode de placement multi-résolution.



(a) Utilisation de sphère concentrique avec un paramètre ϕ faible.



(b) Utilisation de sphère concentrique avec un paramètre ϕ élevé.

FIGURE 5.5 – Représentation de la modélisation hiérarchique du modèle à particules.

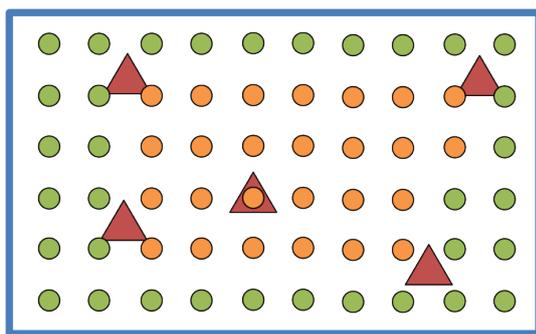
En terme de visualisation cette méthode est très proche de la méthode de grille régulière. Par contre, elle est plus efficace pour localiser les particules dans l'espace en fonction

des capteurs. Le nuage de particules peut avoir une importante granularité et ainsi remplir l'espace de la pièce de manière homogène. La granularité, en terme de particules, sera la même pour toutes les pièces, quelle que soit leur taille.

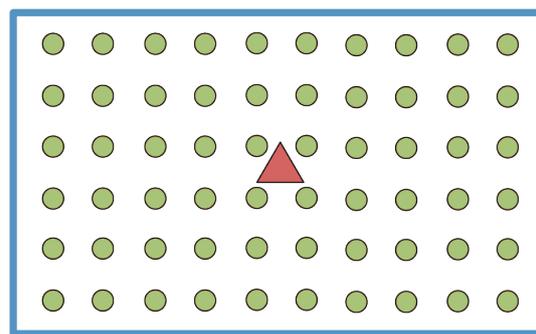
5.4 Interpolation basée capteur

Une fois l'espace discrétisé, il est nécessaire d'interpoler les valeurs de particules en fonction des capteurs. Le principe d'interpolation consiste à résoudre localement un certain nombre d'observations locales. En utilisant une méthode de pondération des particules, notre solution recherche pour chaque particule ses dépendances vis-a-vis des capteurs. Puis notre algorithme calcule pour chaque capteur un coefficient d'influence sur les particules. Le placement des particules est basé sur différentes méthodes mathématiques : la triangulation de Delaunay et l'extraction de cellules de Voronoï.

Afin de rendre notre méthode compatible avec le plus de bâtiments possible, nous avons dû mettre en place une méthode robuste de génération et d'interpolation de particules. Ainsi, il existe deux types de bâtiments, ceux qui sont dotés d'une faible densité de capteurs par pièces (généralement un capteur par pièce) ou des bâtiments avec une nuée de capteurs par pièce (bâtiment critiques). en fonction du type de granularité des capteurs, le résultat de la visualisation sera grandement influencé. Les différentes classes de pièces sont présentées en figure 5.6, il est nécessaire d'identifier deux zones les zones vertes qui nécessitent une interpolation simple des données et les zones jaunes qui nécessitent une interpolation complexe. L'interpolation simple sera réalisée par l'extraction de cellules de Voronoï, tandis que les méthodes complexes reposeront sur une triangulation des capteurs et une fonction d'interpolation.



(a) Schéma d'une pièce composée de plusieurs capteurs.



(b) Représentation typique d'un bureau, ne disposant que d'un seul capteur.

FIGURE 5.6 – Deux exemples de pièces avec des topologies de capteurs différentes.

5.4.1 Triangulation de Delaunay

Dans cette section, nous avons utilisé une méthode basée sur une triangulation afin de pouvoir localiser plus fidèlement les différentes particules dans cet espace. Il est possible d'appliquer différentes méthodes de triangulation pour obtenir un maillage de capteurs comme par exemple une triangulation aléatoire. Mais le résultat n'est pas unique et en fonction des exécutions il est possible d'obtenir différents résultats de maillage. En effet, les méthodes aléatoires choisissent deux sommets sans condition particulière pour définir le maillage. Afin d'avoir une solution de triangulation stable, nous avons utilisé la triangulation de Delaunay qui produit des valeurs de triangulations uniques (nombre de triangles constant [Avis et Bhattacharya, 1983]). Cette méthode fonctionne de la manière suivante : deux sommets sont reliés par une arête si et seulement s'il existe un cercle qui contient seulement et exclusivement ces deux sommets. La figure 5.7 présente le résultat de la triangulation de Delaunay sur un grille régulière, toutes les particules comprises dans l'enveloppe englobante de l'ensemble des capteurs sont sélectionnées (particules jaunes). Il s'agit des particules internes à l'ensemble de capteurs.

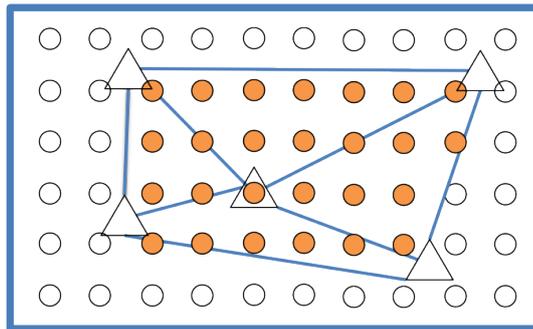


FIGURE 5.7 – Triangulation de Delaunay d'un ensemble de capteurs en 2D.

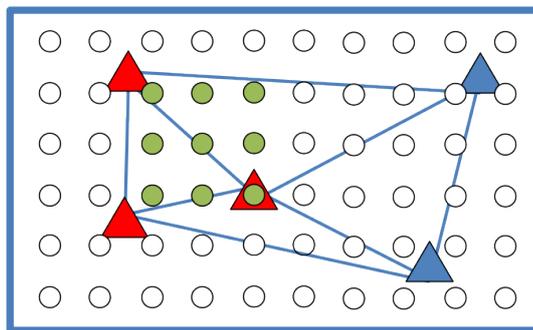


FIGURE 5.8 – Exemple d'enveloppe englobante basée sur la méthode AABB (Axis Aligned Boundary Box).

Une fois la triangulation produite, nous calculons la dépendance des particules vis-à-vis des capteurs. Ainsi, pour chaque triangle formé, il est nécessaire de localiser les particules qui sont incluses à l'intérieur de celui-ci. Ces particules sont extraites à l'aide de l'enveloppe englobante des capteurs, illustrée en figure 5.8. Afin de réduire le coût de calcul en éliminant le plus rapidement les particules non désirées, il est nécessaire de trouver des solutions de réduction d'information, l'enveloppe englobante du triangle de capteurs est calculée, puis une analyse de toutes les particules est effectuée afin d'identifier les particules internes au triangle. En 3D, un tétraèdre est utilisé au lieu de travailler avec un triangle. Pour la localisation, la méthode utilisée consiste à extraire pour chaque triangle son enveloppe englobante. Pour chaque particule incluse dans cette enveloppe, une face du tétraèdre est sélectionnée (en 2D il s'agit d'un segment), et le rayon est lancé partant de la particule vers le dernier sommet du triangle, cette méthode est inspirée des travaux de Snyder sur l'illumination, [Snyder et Barr, 1987]. Si trois rayons entrent en collision avec le tétraèdre, alors la particule est incluse dans le tétraèdre. Afin de déterminer si une particule est incluse dans le triangle, il suffit que deux arêtes soient en collision avec les différents rayons émis depuis la particule. Dans le cadre de la figure 5.9(a), la particule est interne car le nombre d'arêtes en collision est d'au moins deux. Dans la figure 5.9(b), il y a une seule intersection, la particule est donc hors du triangle.

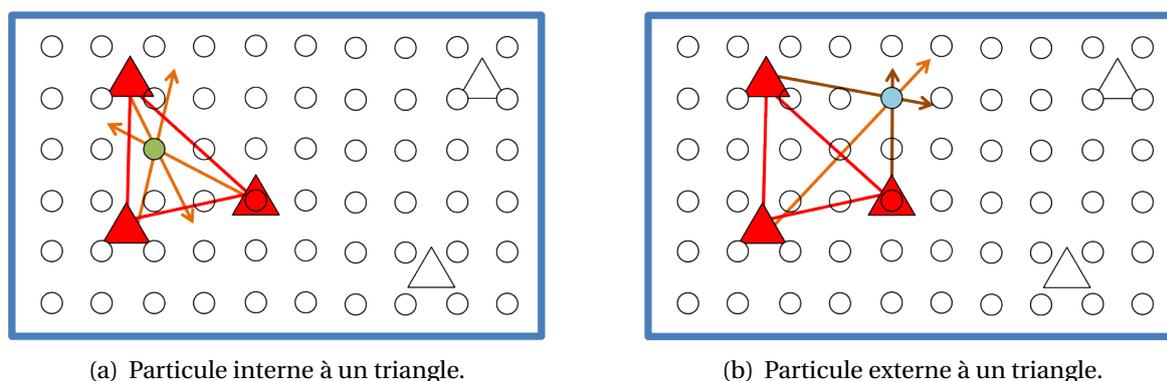


FIGURE 5.9 – Exemple de collision pour deux types de particules.

Grâce à cette méthode il est possible de localiser avec précision les particules vis-à-vis des tétraèdres de capteurs. Le résultat final de la localisation par la triangulation de Delaunay est présenté en figure 5.10, toutes les particules internes se retrouvent avec une relation directe aux différents triangles les englobant.

5.4.2 Cellules de Voronoï

Dans ce paragraphe, nous nous intéressons à la méthode de localisation des particules externes (cf. figure 5.11).

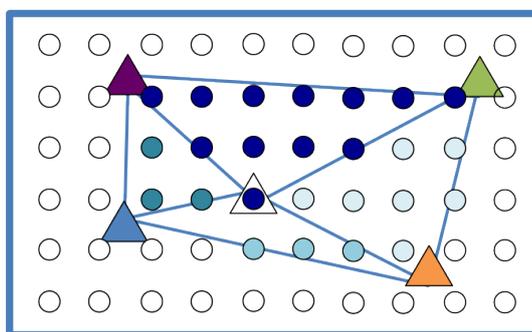


FIGURE 5.10 – Localisation des particules internes au maillage de Delaunay.

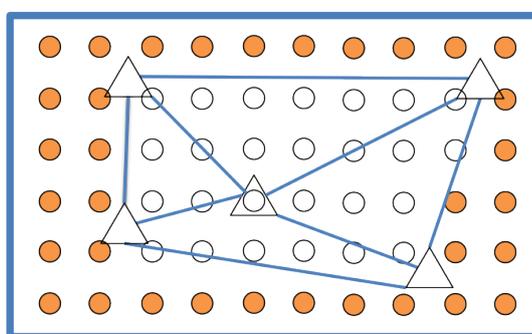
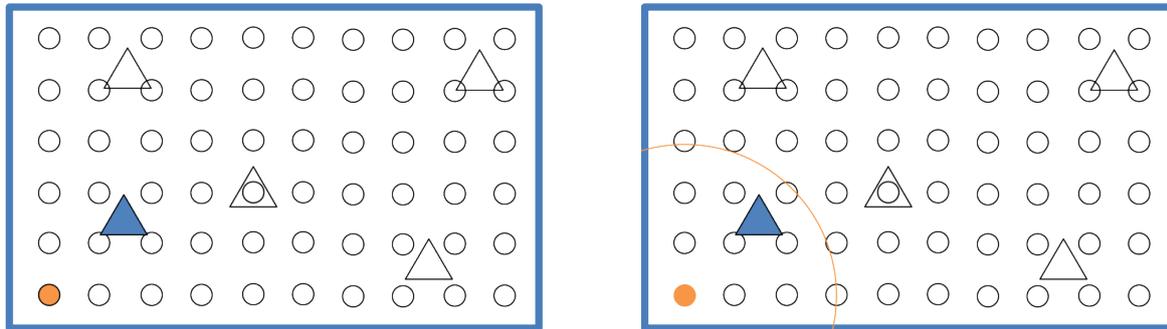


FIGURE 5.11 – Particules externes au maillage de capteurs.

Afin de localiser efficacement ces particules, nous nous sommes servis de la fonction duale de la méthode de Delaunay : les cellules de Voronoï, [Avis et Bhattacharya, 1983]. Pour calculer ces cellules, chaque particule analyse sa distance aux capteurs et extrait le plus proche. Le principe de cette méthode dans un espace discret consiste à localiser chaque particule en fonction de chaque capteur et de sélectionner les plus proches. La figure 5.12 présente la méthode de résolution de l'extraction de cellules de voronoï dans un espace discrétisé.

5.4.3 Pondération des particules

Grâce à ces deux méthodes mathématiques, il nous est possible de déterminer quels capteurs influencent quelles particules. Pour calculer une valeur à chaque particule, nous avons mis en place une fonction d'influence basée sur l'inverse de la distance (particule, capteurs). Soit S^p l'ensemble des capteurs qui influencent la particule p . Soit V_s la valeur du capteur s . Soit W_p la valeur d'une particule basée sur l'inverse de la distance mesurée par la fonction $d(p, s)$. Pour chaque sommet il est nécessaire de pondérer la relation avec la



(a) L'algorithme sélectionne une particule non localisée.

(b) La particule cherche son capteur le plus proche en utilisant une fonction d'intersection entre un cercle et un point.

FIGURE 5.12 – Méthode discrète de la résolution d'extraction de cellules de Voronoï

particule en fonction de la distance totale de tous les sommets en relation. Cette fonction est présentée ci-dessous :

$$\forall p \in P, W_p = \sum_{s_2 \in SP} (V_{s_2} \cdot (1 - \frac{d(p, s_2)}{\sum_{s \in SP} d(p, s)}))$$

5.5 Conclusion

Dans ce chapitre, nous avons étudié une méthode de modélisation que nous avons mise en place pour interpoler les différentes valeurs de l'espace d'une pièce. La méthode retenue repose sur l'utilisation de particules afin de transposer cet espace de manière discret. Nous avons proposé trois différents choix de modélisation de cet espace : par grille régulière, par sphère concentrique et enfin par un schéma de multi-résolution. Une fois le choix du modèle défini nous avons élaboré une méthode générique de localisation de ces particules par rapport aux différents capteurs. Pour cela nous nous sommes reposés sur l'extraction de cellules de Voronoï et son dual la triangulation de Delaunay. Le modèle que nous avons mis en place est compatible avec différentes topologies de pièce. Ici la solution mise en place permet de modéliser un espace dans un environnement, les différents calculs que nous réalisons pour la modélisation sont exécutés sur le serveur de notre outil.

Nous nous réservons de ce modèle de données dans les chapitres suivants, notamment lors de l'évaluation de ces méthodes pour réaliser une solution économique en énergie mais également pour réaliser des supports d'informations. Enfin, nous présenterons une application de cette solution sur des données concrètes issues de plusieurs bâtiments

pilotes. Nous pourrions ainsi mettre en avant les avantages et les inconvénients de cette approche de modélisation.

Ces travaux ont donné lieu à deux publications dont une nationale [Lange *et al.*, 2010] et une internationale [Vasques *et al.*, 2011].

Visualisations mises en place

Un programme est un sort jeté sur un ordinateur, qui transforme tout texte saisi au clavier en message d'erreur.

Anonyme

Sommaire

6.1	Introduction	87
6.2	Visualisations issues de l'état de l'art	88
6.3	Nos visualisations 3D	93
6.4	Solutions de multi-résolutions	101
6.5	Outil de fouille de données	104
6.6	Synthèse	108

6.1 Introduction

Dans ce chapitre, nous nous intéressons aux différentes méthodes d'analyse visuelle que nous avons implémentées dans notre prototype d'outil de visualisation. Nous présentons les différentes méthodes mises en place pour la visualisation et pour la fouille de données. Dans un premier temps, nous étudions les différentes approches de représentation de l'information en 2D ou en 3D, puis nous présentons les solutions que nous avons mises

en place pour améliorer le modèle d'un bâtiment, qui est notre ressource principale dans le cadre du projet RIDER.

Pour permettre d'améliorer le modèle de données issues du projet, il est important de proposer des outils adaptés, ceux-ci dépendant fortement des profils utilisateurs. Ainsi, dans ces travaux, nous nous sommes confrontés à différents types d'utilisateurs finaux : des techniciens du bâtiment, des gestionnaires de bâtiment et des spécialistes de domaines complètement différents. Le premier profil cherche à se rapprocher des interfaces connues de type tableau de bord, les gestionnaires quant à eux préfèrent se focaliser sur des vues orientées gain (énergie, coût, ...). Le dernier groupe est fortement attaché au fait d'avoir une interface abordable, nécessitant un temps d'apprentissage relativement court.

Dans ce chapitre, nous allons nous intéresser aux méthodes de visualisation mises en place dans le cadre de ce projet. La section 6.2 présentera les représentations standards que nous avons dû mettre en place dans le cadre de ce projet industriel, celles-ci couvrent un ensemble de visualisation issues d'outils de gestion de bâtiment. Dans la section 6.3, nous présentons les méthodes de représentation de l'information que nous proposons, ces approches sont basées sur un environnement 3D. Et enfin, dans la section 6.5, nous nous intéressons aux méthodes que nous avons mises en place pour améliorer le modèle du bâtiment, notamment autour de solutions interactives.

6.2 Visualisations issues de l'état de l'art

Le but de ces travaux est de réaliser une solution logicielle afin de visualiser des données de bâtiment. Ainsi, les solutions que nous devons mettre en place sont en partie couvertes par des outils existants. Nous nous sommes donc intéressés à reproduire les méthodes de visualisation 2D existantes, notamment les méthodes géométriques et les méthodes hiérarchiques. Dans le chapitre 4, nous avons présenté le principe des méthodes géométriques et hiérarchiques. Les approches géométriques utilisent différents espaces pour représenter les données, mais plus particulièrement l'espace euclidien. Les méthodes hiérarchiques, quant à elles, sont basées sur l'agrégation de l'information.

Dans un cadre industriel, de nombreux tableaux de bord de gestion de bâtiment utilisent des méthodes classiques (courbes, vu-mètre, graphique camembert, ...) de la visualisation d'informations. Les figures 6.1(a)¹, 6.1(b)² et 6.1(c)³ présentent trois tableaux de bord proposés par des logiciels de gestion de données issues du bâtiment. En général, ce type d'outil n'offre pas de solutions d'optimisation ou d'amélioration du bâtiment. Ce type d'outil permet uniquement d'explorer les données. Dans l'exemple de la figure 6.1(c), il

1. <http://www.johnsoncontrols.fr/>

2. <http://www.energimotique.com/>

3. <http://www.visiomatic.com/>

est possible d'avoir des informations supplémentaires sur la topologie du bâtiment (représentation par pièce), mais il est nécessaire de réaliser un travail long et fastidieux pour produire ce résultat.



(a) Visualisation proposée par Johnson Controls, pour la gestion de données issues du bâtiment



(b) EnerVue : Energy Management Dashboard.



(c) Outil de visualisation basée sur la cartographie présentée dans Visiomatic.

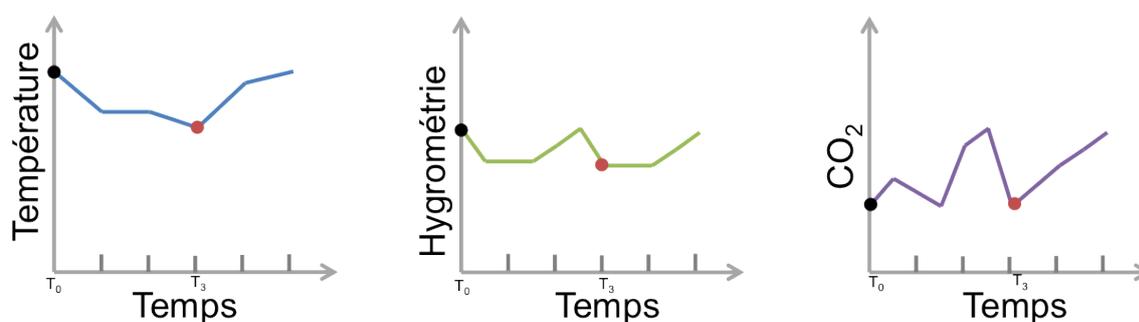
FIGURE 6.1 – Exemple de représentation standard de données basée sur le principe de courbes

Le premier type de visualisation qui nous a intéressé repose sur les méthodes géométriques. Les méthodes de visualisation de données pour le bâtiment sont orientées autour

de ces approches. Nous avons donc développé un certain nombre de ces méthodes de visualisation pour permettre aux utilisateurs de solutions existantes de ne pas être troublés par des approches différentes.

6.2.1 Les graphiques

La première solution que nous avons mise en place est la visualisation de données par courbe. Le modèle RIDER fournit la valeur des capteurs placés dans le bâtiment. Ainsi, notre solution de visualisation consiste à cartographier pour chaque capteur sa valeur en fonction du temps. La figure 6.2 illustre la construction de cette méthode de visualisation pour un ensemble de mesures de capteurs, chaque courbe représente un type de données avec son évolution temporel. Grâce à cette méthode, il est possible de suivre dans le temps (en fonction des données transmises) l'évolution des points de mesure du bâtiment.



(a) Représentation des valeurs d'un capteur de température en fonction du temps.

(b) Représentation des valeurs d'un capteur d'hygrométrie en fonction du temps.

(c) Représentation des valeurs d'un capteur de CO₂ en fonction du temps.

FIGURE 6.2 – Représentation par courbe en fonction du temps de trois types de mesures.

6.2.2 Visualisation par lignes brisées

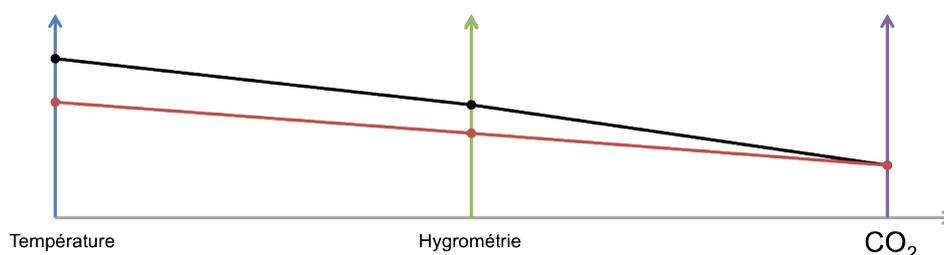


FIGURE 6.3 – Représentation par lignes brisées de trois mesures à deux intervalles de temps différents, en noir nous avons T0 et en rouge T3.

La seconde approche de visualisation que nous avons intégrée dans cet outil est la visualisation par coordonnées parallèles. Cette méthode de visualisation agrège différentes dimensions sur la même vue afin de permettre notamment d'extraire plus facilement les compositions. Les capteurs utilisés dans le bâtiment produisent des mesures comme la température, la pression et l'hygrométrie. Ainsi, dans notre solution de visualisation, les différentes dimensions correspondent aux différentes valeurs mesurées par les capteurs.

Le principe de visualisation par lignes brisées repose sur une série d'axes des ordonnées, qui représentent les différentes dimensions de la donnée. Les différentes valeurs sont cartographiées pour obtenir la visualisation, c'est à dire que chaque point de mesure est placé sur un des axes qui lui est propre, ensuite la solution connecte les différentes valeurs du même intervalle pour obtenir une représentation de la donnée au temps désiré. La figure 6.3 présente un exemple de représentation par ligne brisée, chaque polyligne représente un agrégat de capteurs. Ainsi, dans cette représentation, deux agrégats de capteurs sont visualisés et sont composés d'un capteur de température, d'un capteur d'hygrométrie et d'un capteur de CO₂. Par ce type de visualisation, il est possible de comparer des agrégats similaires. Malheureusement si le nombre de capteurs devient trop important, il devient alors difficile d'extraire du contenu.

6.2.3 Les graphes radar

La dernière solution de représentation géométrique présentée dans notre application est appelée graphe radar. Cette solution consiste à cartographier les différentes dimensions d'une donnée sur une représentation en étoile. Le radar est composé d'un point central et de différents axes correspondants à chaque dimension, l'angle entre chaque axe est déterminé en fonction du nombre de ses dimensions. La figure 6.4 présente le résultat de cette méthode de visualisation sur le même jeu de données que les figures précédentes. Ainsi, chaque triangle représente un agrégat de capteurs.

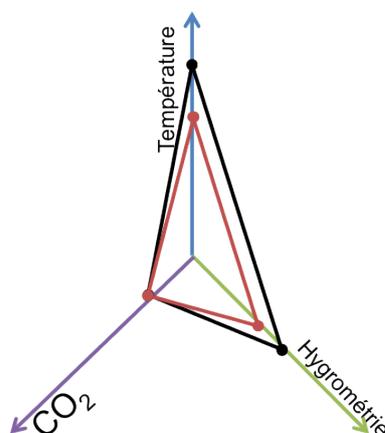


FIGURE 6.4 – Représentation par graphe radar de deux captations à deux intervalles de temps différents. En noir nous avons T0 et en rouge T3.

6.2.4 La visualisation par multiples nuages de points

Les nuages de points sont une méthode très utilisée pour la visualisation de données géométriques. Cette solution est également très largement utilisée pour représenter les données composées de nombreuses dimensions notamment en utilisant le principe d'agglomérat de nuages de points. Le but de cette méthode est de cartographier les données sur plusieurs vues paramétrées sur l'axe des abscisses et l'axe des ordonnées. Chaque figure ainsi produite est représentée par un jeu de deux dimensions. Les données sont alors cartographiées en fonction de leur valeur. Grâce à cette approche, il est possible de corréler différentes dimensions entre elles, mais également de mettre en forme les données pour extraire des informations supplémentaires.

6.2.5 La visualisation par glyphe

La dernière solution, issue de l'état de l'art du domaine IV (Information Visualisation), que nous avons exploitée dans la solution de visualisation pour RIDER est orientée autour de la génération de glyphes. Un glyphe est une image produite à l'aide d'un motif récursif appliqué à une instance des données. Pour cette solution, nous avons mis en place différentes solutions de motifs récursifs, soit basés sur l'agrégation de dimensions, soit sur le tri des dimensions. Chaque pixel du dispositif de visualisation peut-être utilisé pour représenter les différentes données et surtout l'évolution temporelle de celles-ci.

6.3 Nos visualisations 3D

Cette section présente les différents nouveaux types de visualisation que nous proposons dans le cadre de la gestion de bâtiment. Ces méthodes sont orientées autour d'un environnement 3D. Dans cette section, nous présentons des méthodes de visualisation basée sur des informations partielles gérées dans le modèle pivot de RIDER, mais également des méthodes basées sur la visualisation de volume.

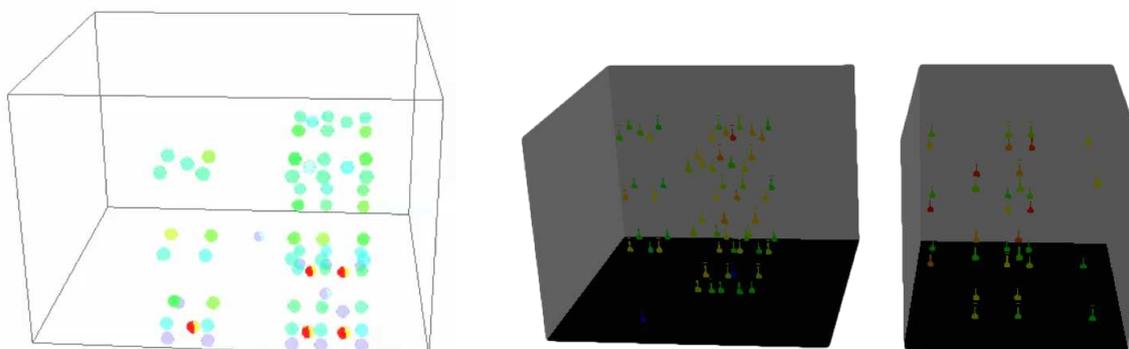
6.3.1 Visualisation basée seulement sur une information spatiale

Au cours de ces travaux, nous nous sommes intéressés à travailler sur des moyens d'utiliser la représentation spatiale des données mais également d'étendre les méthodes de visualisation standards pour répondre au problème de temporalité.

Les données que nous étudions sont issues de capteurs. Les différentes dimensions de ces capteurs ont une forte connotation, il est alors possible de leur attribuer une sémantique visuelle. Une de ces dimensions correspond aux coordonnées spatiales du capteur. Ainsi, il est possible d'utiliser cette information en terme de solution de placement de l'information. Nous cartographions ainsi les données par rapport à leurs coordonnées réelles dans l'espace. La représentation des données est alors basée sur les informations de localisation auxquelles est ajoutée une dimension pour améliorer la compréhension de la donnée. Nous avons étudié trois de ces composantes pour permettre de proposer différentes solutions autour de la représentation par glyphe, sphère et icônes.

Une des solutions de visualisation spatiale 3D est basée sur la représentation par sphère. Des sphères représentent les différents capteurs et sont placées dans l'espace de la pièce. Les capteurs sont alors segmentés équitablement en fonction du nombre de dimensions qu'ils contiennent. Chaque partie de la sphère qui représente une dimension est alors pondérée en fonction de sa valeur. Ici, la figure 6.5(a) présente le résultat de cette méthode de visualisation, ainsi les différentes sphères sont localisées à l'emplacement des capteurs. De plus ces sphères sont colorées en fonction des différents capteurs localisés.

La seconde approche que nous avons mise en place est orientée autour de la représentation iconique. Dans cette méthode, pour chaque capteur il est nécessaire d'assigner une représentation symbolique pour les différents points de mesures. Cette solution permet donc de visualiser les données multi-dimensionnelles de manière à extraire le contenu de celles-ci, en se basant sur des représentations communes. Les différentes icônes utilisées présentent des représentations conventionnelles des types de données. Cette approche repose sur l'utilisation des informations géographiques des capteurs, un exemple de rendu produit par cette méthode est donné en figure 6.5(b), les capteurs présentés ici sont des capteurs de températures.



(a) Représentation d'un jeu de données en utilisant la représentation par sphères.

(b) Représentation d'un ensemble de données basées sur la solution de représentation sémantique.

FIGURE 6.5 – Ensemble de systèmes de visualisation basés sur l'utilisation des informations spatiales issues des données de RIDER.

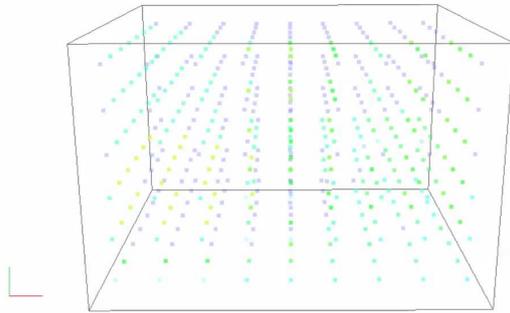
6.3.2 Visualisation volumique

Enfin, nous nous sommes intéressés à la visualisation volumique de données. Comme nous l'avons présenté dans le chapitre 5, nous avons mis en place un système de modélisation des espaces au travers d'un ensemble de particules. Dans cette section, nous nous intéressons à la représentation de cette information par notre système de visualisation au travers de différentes représentations volumiques.

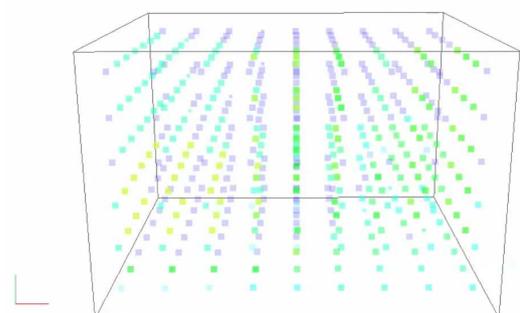
Visualisation basée sur des primitives simple

La première méthode que nous avons mise en place repose sur l'utilisation de la primitive point des dispositifs graphiques. Ainsi, chaque particule est représentée par un point. Cette méthode de représentation est présentée en figure 6.6(a), il est possible de s'apercevoir que différents problèmes sur ce type de méthode émergent. Ainsi, il est complexe d'appréhender le volume de la pièce sur une représentation statique. Ici les différentes particules sont colorées en fonction des capteurs qui leur ont été assignés.

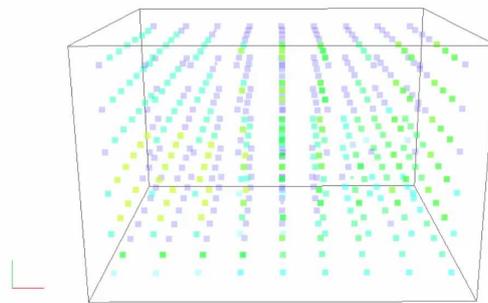
Cette représentation reste complexe à manipuler pour des usagers non avertis. Une des notions complexes à appréhender avec ce type de visualisation est la mise à jour de données. En effet, sur une large structure de ce type, le moindre changement est complexe à visualiser. Il est donc nécessaire de mettre au point différents outils de saillance de l'information. Pour ce faire, nous utilisons deux attributs supplémentaires pour la représentation des particules : la taille et la transparence. La méthode repose sur l'historisation des données sur le client et fonctionne de la manière suivante : les particules qui sont mises à



(a) Visualisation des particules dans une pièce au format brut (RAW).



(b) Visualisation des particules dans une pièce au format brut (RAW) avec mise en avant de la mise à jour (par le déplacement et un jeu de transparence), $T = 0$.



(c) Visualisation des particules dans une pièce au format brut (RAW), $T = 1$.

FIGURE 6.6 – Ensemble de visualisations basées sur des primitives élémentaires.

jour régulièrement voient leur fonction de transparence se réduire et leur taille augmenter. Tandis que celles qui sont moins mises à jour voient leur taille diminuer et leur fonction de transparence augmenter. Cette solution de représentation est donnée en figure 6.6(b), qui représente l'état initial de notre vue. Après plusieurs itération (mise à jours), le résultat obtenu est donnée en figure 6.6(c) : certaines particules ont rétréci, voir disparu.

Visualisation structurée

La deuxième approche que nous avons étudiée cherche à simplifier le rendu de notre visualisation, en effet afficher toutes les primitives du nuage de particules est trop coûteux pour des dispositifs plus légers. De plus cette solution de représentation peut également engendrer des difficultés de compréhension. Ainsi, nous avons mis en place différentes solutions de rendu afin de permettre de simplifier la visualisation d'un point de

vue complexité et compréhension. Pour ce faire, nous avons utilisé une méthode basée sur l'extraction de régions d'intérêts. Ces régions sont des zones homogènes de valeurs. Dans notre méthode de modélisation, ces régions d'intérêts sont des particules ayant une valeur proche. Nos données représentent ainsi un ensemble de composantes.

Nous nous sommes intéressés à deux solutions d'extraction de classes d'équivalences : la première méthode est basée sur des espaces englobants, tandis que la deuxième solution est basée sur la théorie des graphes.

La première approche repose sur l'extraction de particules ayant une valeur similaire. La similarité est mesurée en fonction d'un δ défini. Si deux particules ont un écart de valeur inférieure à δ alors les deux particules appartiennent à la même classe d'équivalence. La deuxième approche repose sur la théorie des graphes, une introduction à la théorie des graphes est présentée en annexe. La méthode calcule l'ensemble du graphe de voisinage des particules et ensuite un algorithme d'extraction de composantes connexes en fonction de la valeur des particules est appliqué afin de segmenter ces particules en composantes. Ces deux méthodes permettent de créer plusieurs ensembles de particules appelés composantes. L'étape suivante est la visualisation. Ici, différentes approches ont été proposées : l'utilisation d'enveloppes englobantes et des méthodes de triangulation.

La méthode de visualisation par enveloppe englobante repose sur l'utilisation des AABB. Pour cette méthode, on extrait les coordonnées maximales et minimales d'une composante. Dans notre visualisation, la seule information que nous visualisons est l'enveloppe englobante de chaque composante. Ainsi, en fonction de la méthode d'extraction de composante utilisée, la visualisation est plus ou moins complexe. La figure 6.7 présente les deux résultats de cette méthode au travers du même jeu de données. Le résultat montre que la solution basée sur les graphes, crée plus de composantes que la méthode basée sur une valeur de similarité. Par contre, le résultat est plus précis pour la méthode basée sur les graphes, des zones d'intérêts apparaissent.

Le problème de l'approche par les graphes est que le résultat crée bien plus de recouvrements visuels entre les composantes que la méthode basée sur une mesure de similarité. Malgré tout, la similarité crée également des recouvrements, notamment à cause de la méthode de représentation utilisée. Pour palier ce problème, nous avons mis en place une solution de réduction de complexité de compréhension de la vue en utilisant des méthodes de retriangulation : le *marching cube*. Le *marching cube* est un algorithme qui permet de créer un objet polygonal à partir d'un nuage de points 3D, [Lorensen et Cline, 1987]. Dans notre méthode, nous appliquons pour chaque composante une méthode de triangulation. Cette méthodologie nous permet de créer une représentation plus structurée des composantes. Comme le montre la figure 6.8, les données issues de l'extraction de composante se prêtent relativement bien à cet usage.

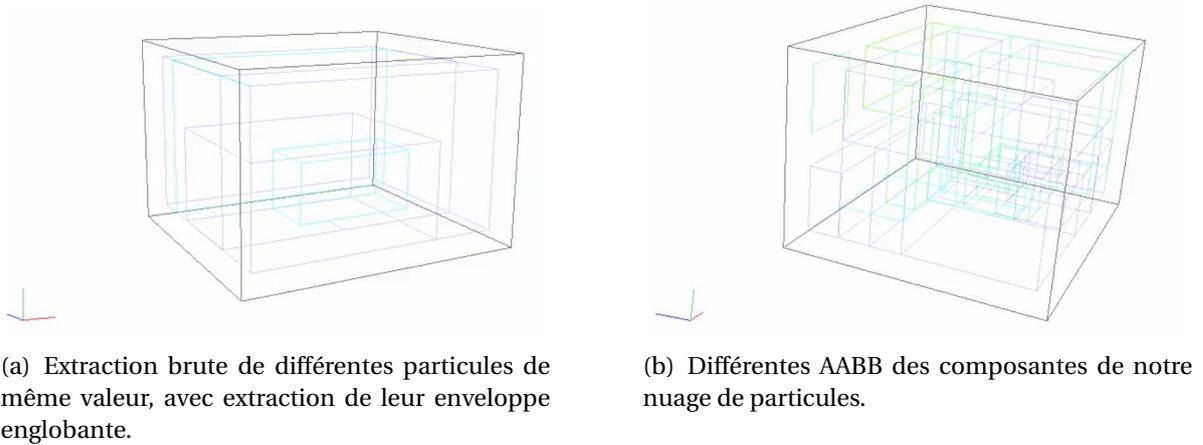


FIGURE 6.7 – résultat de l'extraction d'enveloppe englobante des deux méthodes d'extraction de composante.

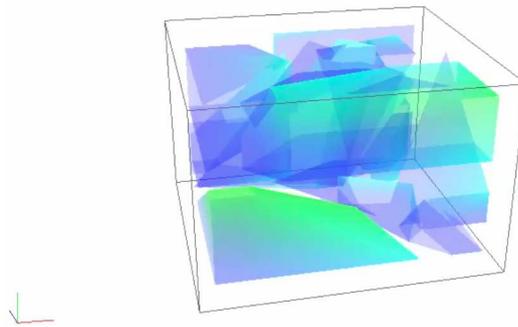


FIGURE 6.8 – Méthode de visualisation, basées sur la triangulation des composantes connexe.

Visualisation par voxel

Pour produire une visualisation de volume pertinente, nous utilisons une représentation par voxel. Grâce à cette solution il est possible de réaliser une visualisation volumique qui prend en compte toute la pièce. Les voxels sont répartis dans toute la pièce en fonction de la localisation des particules. En fonction de la modélisation de l'espace des pièces, les voxels vont contenir plus ou moins de particules. Grâce à cette solution et contrairement à la visualisation simple de particules, il est possible de mieux appréhender le contenu de la visualisation est ainsi de permettre d'extraire plus simplement des informations relatives à ces données. Malheureusement, la solution par voxel ne nous permet pas d'avoir un rendu continu du volume de données. Il est donc nécessaire de développer des solutions alter-

natives. Comme le présente la figure 6.9(b) l'exploration des données internes à la pièce est inaccessible, il est nécessaire de définir des outils d'explorations de la représentation interne à la pièce. Nous avons utilisé une fonction de transparence sur les voxels afin de mettre en avant certaines informations de la pièce comme le montre la figure 6.9(a). Malheureusement, cette méthode fait clairement apparaître les différences entre les voxels.

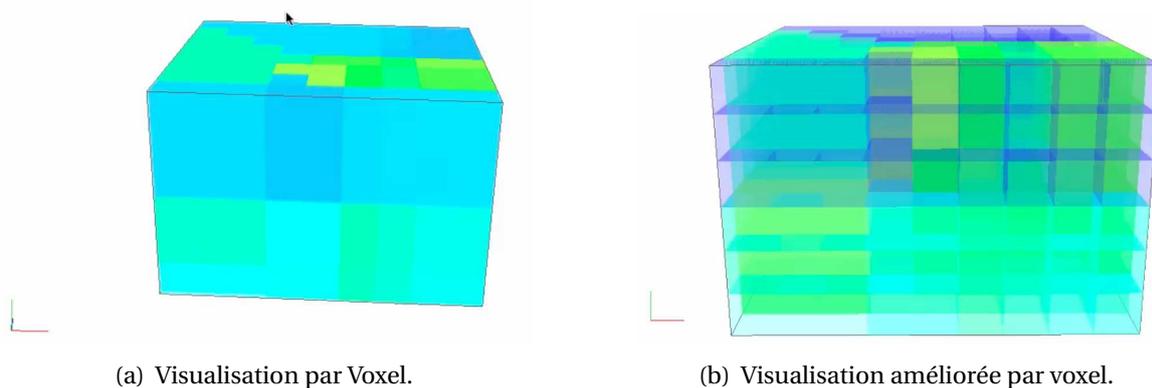


FIGURE 6.9 – Ensemble de visualisation localisée à l'aide d'information spatiale utilisée dans le cadre du projet RIDER.

Visualisation basée shaders

Afin de proposer de nouvelles formes de représentations de notre visualisation par voxel, nous avons mis en place plusieurs shaders. Les shaders sont des programmes informatiques dont le but principal est de calculer des effets de rendu à partir du dispositif graphique de l'ordinateur. Il existe ainsi trois types de shaders : pixel, sommet et géométrie. Le pixel shader est utilisé pour directement interagir avec les différents pixels de la scène au travers de la couleur notamment. Les shaders de sommets, eux, sont utilisés pour manipuler les informations relatives aux sommets envoyés au dispositif graphique. Ils peuvent ainsi agir sur la position, la couleur et les coordonnées de textures du sommet. Enfin, le dernier shader manipule la géométrie de la scène. Il permet ainsi de générer de nouvelles primitives. L'utilisation de cette technique permet de créer des modèles 3D plus complexes et modulaires. Pour notre outil de visualisation, nous avons mis en place quatre techniques autour des shaders afin d'améliorer le rendu graphique de notre solution.

Le premier shader que nous avons utilisé repose sur une fonction de lissage des couleurs entre chaque voxel de l'image. Afin de lisser la représentation de notre visualisation, cette méthode interpole les valeurs des sommets des voxels en fonction de ce qui l'entoure. Ainsi, on obtient un rendu volumique lissé de nos voxels. La sensation de créneaux a disparu. Un exemple de rendu produit par cette méthode est donné en figure 6.10(a).

Comme nous pouvons le voir, la visualisation créée donne un aspect plus correct comparé aux méthodes précédentes.

Le deuxième shader utilisé a pour but de mettre en avant les structures dominantes en produisant un effet concave ou convexe en fonction du voxel. Le shader est basé sur un jeu de lumière qui va permettre de créer les différents effets : convexe ou concaves. En fonction de la valeur d'un seuil, un rendu plus ou moins concave est appliqué au voxel. Ainsi, cette méthode fait ressortir des valeurs élevées et basses comme le montre la figure 6.10(b).

Le troisième shader se base sur l'extraction des isosurfaces. Il s'agit de courbes de niveau dans un espace 3D. Par cette méthode, des zones homogènes et connexes sont mises en avant. Cette approche repose sur la définition d'une fonction de seuil qui correspond à l'ensemble de valeurs que l'utilisateur souhaite mettre en avant. Dans le cadre d'une carte de température, l'utilisateur peut désirer visualiser les températures critiques de son environnement, ainsi il appliquera un seuil sur les températures élevés, un exemple de rendu produit par cette méthode est donnée en figure 6.10(c). A partir de cette approche, il est donc possible de mettre en avant des structures en utilisant les couleurs originales des voxels. La figure 6.10(d) met en avant l'isosurface des données de températures, notamment celles qui sont élevées.

Une extension aux lignes brisées

La dernière approche de visualisation que nous allons présenter a été développée afin d'analyser les informations de l'entrepôt de données. En effet, suite à différents problèmes relatifs aux données, nous avons mis en place un système d'analyse visuelle basé sur les coordonnées parallèles. Nous avons étendu l'approche standard dans le but de réaliser une vue 3D des lignes brisées.

Cette méthodologie repose sur l'utilisation des coordonnées parallèles avec une relation de temporalité qui est mise en avant. Les différentes périodes de capture sont agrégées dans la largeur afin de créer une vue surfacique des données. Cette méthodologie est présentée figure 6.11, les pièces du bâtiment sont représentées par une surface composée des différentes données des capteurs de la pièce. Cette approche surfacique permet de créer un point de vue des données d'une pièce et ainsi d'identifier rapidement les profils et les motifs récurrents. Les évolutions temporelles peuvent être extraites plus simplement et des compositions entre les capteurs peuvent être proposées.

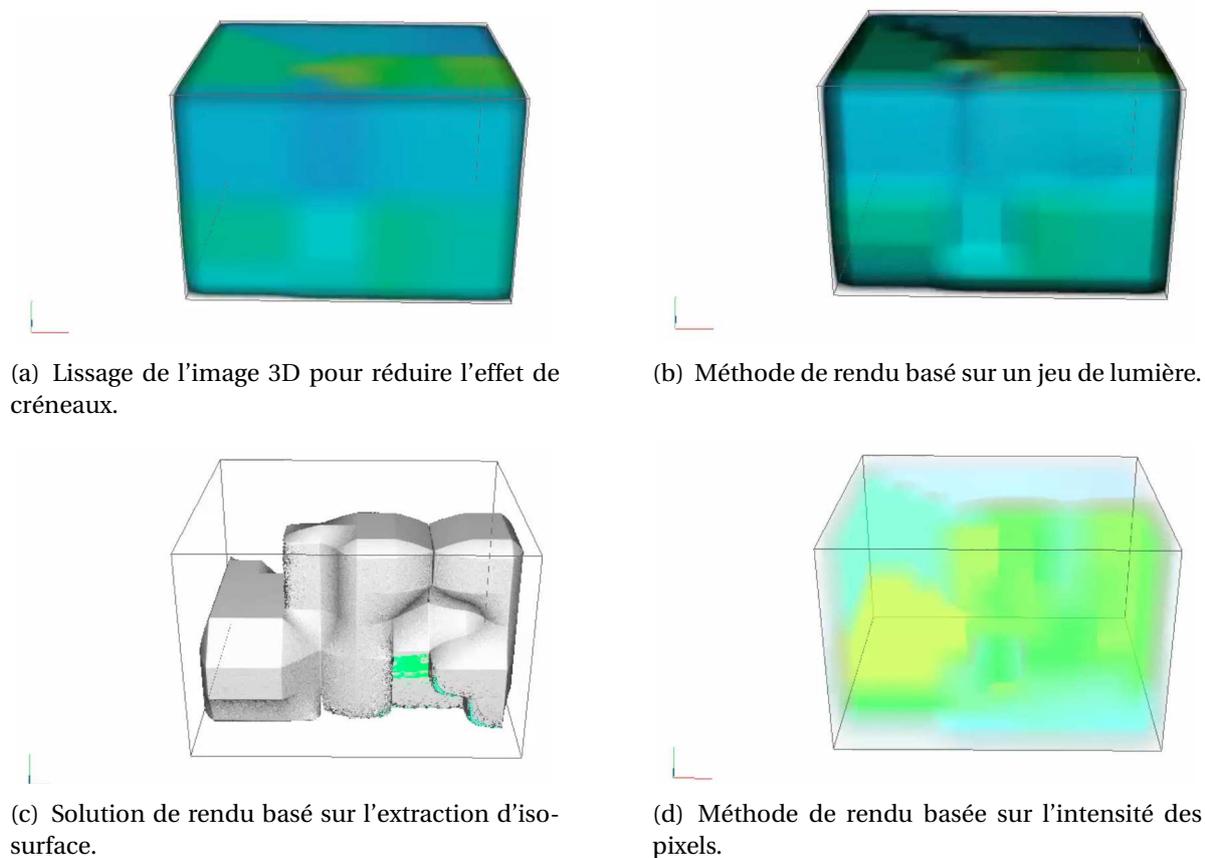


FIGURE 6.10 – Ensemble de shaders mis en place pour améliorer notre rendu pour la visualisation.

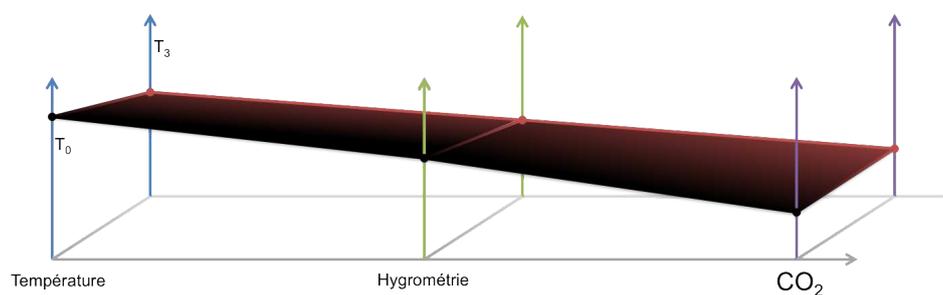


FIGURE 6.11 – Représentation par lignes brisées étendues de deux captations à deux intervalles de temps différents. En noir nous avons T_0 et en rouge T_3 .

Les différentes méthodes que nous avons présentées dans ce document ont été mises en place pour comprendre plus simplement un jeu de données. Nous utilisons ces diffé-

rentes méthodes dans le chapitre suivant afin d'extraire différents contenus à partir d'un véritable jeu de données⁴. Nos différentes solutions de visualisation ont pour but de fonctionner sur différents supports matériels de visualisation. Ainsi, nous nous sommes intéressés aux approches de multi-résolution.

6.4 Solutions de multi-résolution

L'objectif du projet est de réaliser une architecture multi-échelles d'un point de vue matériel c'est à dire qu'elle peut être exploitée sur un centre de calcul pour les données d'une ville ou sur un simple ordinateur pour gérer un bâtiment. Par rapport à la visualisation, nous avons également travaillé sur la gestion de rendu en fonction du matériel disponible. En effet, le dispositif de visualisation utilisé limite la quantité d'informations utiles.

Notre solution est développée autour du paradigme client / serveur. Un serveur crée le nuage de particules et celui-ci est ensuite transmis au client. Pour créer un système multi-résolution, nous avons proposé une solution adaptative en fonction du client lors de la génération du contenu, puis une solution de mise à jour multi-résolution et enfin une adaptation de la visualisation sur le client.

6.4.1 multi-résolution pour la transmission d'informations

La première solution de multi-résolution est axée sur l'identification des dispositifs de visualisation. Notre projet a pour but d'être utilisé par différentes solutions de visualisation de données comme par exemple : un téléphone portable ou une salle immersion. Pour ce faire, nous nous appuyons sur le paradigme client / serveur que nous avons mis en place. Chaque client s'identifie au serveur en transmettant son identifiant, qui correspond aux capacités de rendu du dispositif. Il est alors possible pour le serveur d'adapter le nombre de particules en fonction du dispositif client. Comme nous l'avons vu dans le chapitre précédent, ce coefficient intervient lors de la génération des particules au travers du facteur ϕ .

Une fois les particules générées, il faut transmettre les informations aux clients. Pour ce faire nous avons utilisé un protocole réseau bien défini. La partie la plus consommatrice en terme de quantité d'informations envoyée concerne les particules. Nous avons donc proposé une solution reposant sur la mise à jour progressive des données en fonction du support de communication utilisé afin de ne pas saturer la bande passante. Cette solution permet donc de donner une impression de mise à jour progressive de l'ensemble des particules. La figure 6.12(c) présente le principe de cette diffusion d'information, la valeur du capteur mis à jour est envoyée, puis son voisinage direct et enfin un voisinage étendu. par

4. L'ensemble de ces méthodes est présenté dans la vidéo à l'adresse suivante : <http://benoitlange.fr/these/videos/visus.mp4>

cette méthode, il est possible de simuler un comportement de flux dans une structure qui n'était pas destinée à cela. Cependant, comme le montrent les figures 6.12(a) et 6.12(b) la mise à jour des données donne un effet de fluide plus convaincant sur des données simulées que sur des données réelles.

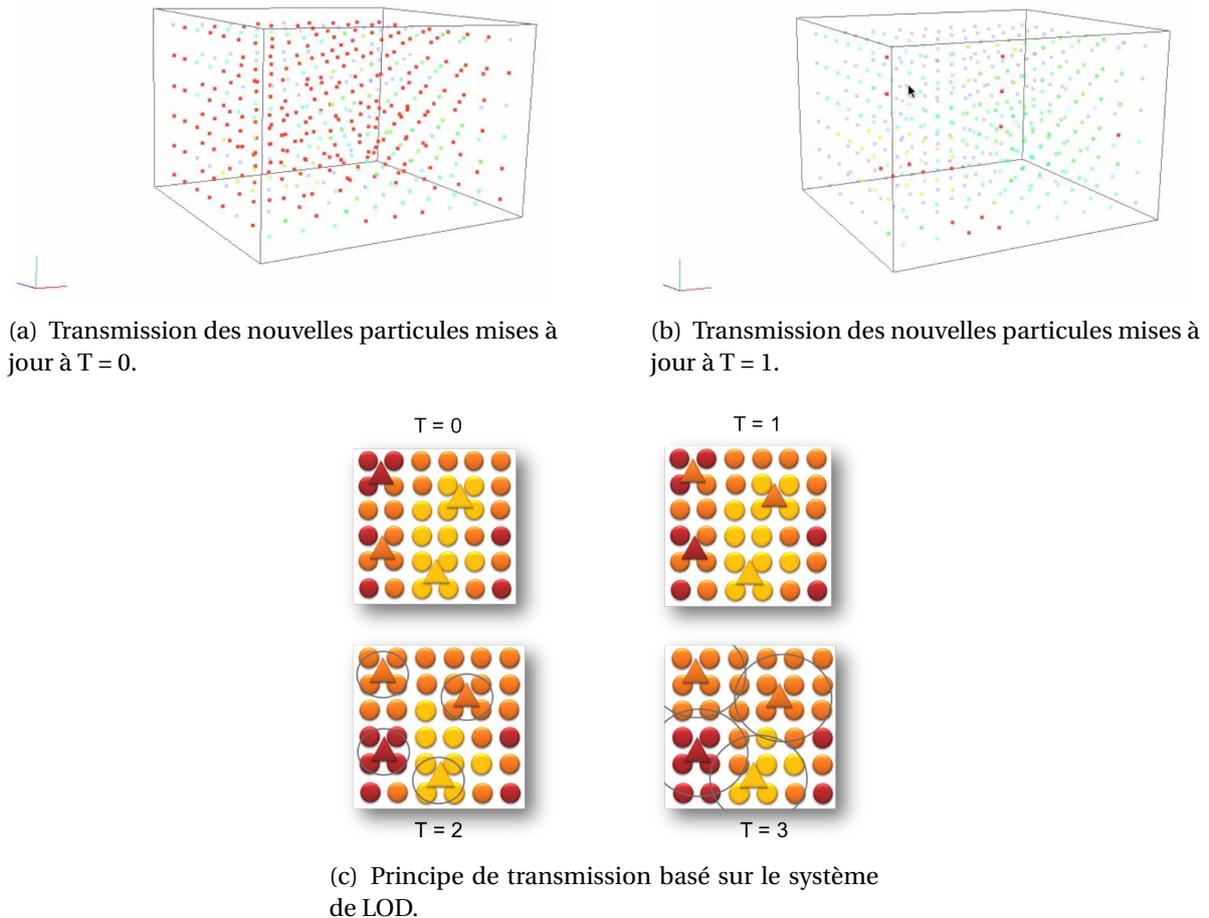


FIGURE 6.12 – Principe de fonctionnement du LOD pour la transmission avec schéma d'explication et résultat obtenu.

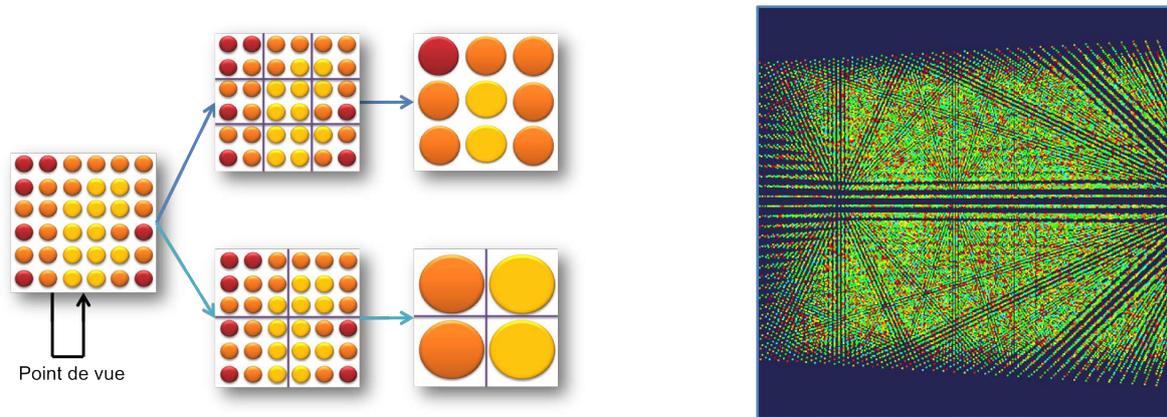
6.4.2 multi-résolution pour la visualisation

Nous nous sommes également intéressés à la simplification de scène dans le cadre de contenu issu du bâtiment. Dans cette section, nous allons présenter les différentes approches multi-résolution mises en place dans le cadre de ces travaux. Nous avons mis au point deux méthodes de simplification de nuage de particules afin de produire des visua-

lisations plus légères et ainsi créer le rendu de plusieurs pièces, voir du bâtiment tout en conservant le maximum d'informations.

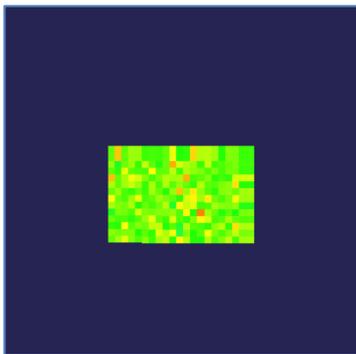
La première méthode que nous avons mise en place repose sur l'agrégation de particules. Pour cette méthode, chaque espace (pièce) est divisé de manière régulière afin d'obtenir un espace composé de sous-espaces. Chaque sous-espace possède une ou plusieurs particules. On agrège ainsi les différentes valeurs des particules et on assigne à notre espace cette valeur. On obtient ainsi un espace clusterisé. Grâce à cette approche, il est possible de gérer plusieurs niveaux de détails : soit pour des périphériques différents, soit pour des données placées plus ou moins loin dans la scène par rapport à la caméra. Cette partie des calculs est gérée par le client, ainsi il peut dynamiquement produire à la volée une vue adaptée au contexte. Le principe de la méthode est donné en figure 6.13(a). Comme il est possible de le voir, cette méthode permet de gérer différentes solutions de visualisation en fonction du contexte de la caméra. Nous avons utilisé cette méthode sur un jeu de données aléatoire. La figure 6.13(b) présente une représentation d'un espace basée sur des particules avec une pondération aléatoire des particules. La figure 6.13(c) présente le résultat de la clusterisation de notre pièce ; un lissage moyen a été appliqué sur cette vue en raison du tirage aléatoire des différentes valeurs. Enfin les figures 6.13(d) et 6.13(e) présentent le résultat de clusterisation à d'autres étapes, la surface de la pièce semble ainsi totalement lissée. Les informations importantes (valeurs élevés, valeurs faibles) ont disparues.

La seconde méthode de multi-résolution s'intéresse à extraire les informations importantes des données. Dans le cadre d'un centre de calcul, l'important est d'extraire les valeurs de température trop basses ou trop haute par exemple. Extraire ce type d'informations est relativement important pour des systèmes critiques. Une panne peut avoir des conséquences matérielles et humaines importantes. Il est alors crucial de permettre d'extraire rapidement des informations pertinentes des données en se basant notamment sur l'extraction de points caractérisés. La méthode utilise le principe de parcours de voisinage. Pour ce faire, il est nécessaire de calculer le voisinage de chaque particule. La méthode fonctionne de la manière suivante : pour chaque particule restante dans la liste de particules, le K-voisinage est exploré en fonction de la distance de l'espace par rapport à la caméra, la notion d'extraction de voisinage est traitée dans les annexes. Parmi cet ensemble, les particules critiques sont extraites. Ainsi, un nouvel espace est défini contenant moins d'informations, mais disposant des informations cruciales de la visualisation. Dans la figure 6.14(a)n, nous présentons le schéma de représentation de cette méthode, différents rendus peuvent être extraits d'un ensemble de données, notamment en explorant une certaine profondeur dans le graphe de données mais également en sélectionnant les données importantes à extraire. Un aperçu de ces résultats est donné en figure 6.14(c) : après simplification du nuage de points, le résultat obtenu met en avant les points critiques (points rouge).

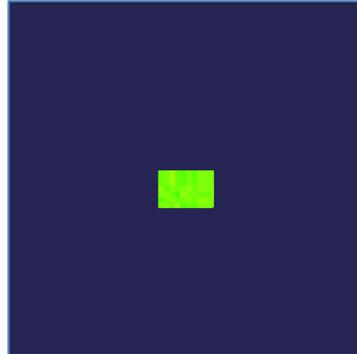


(a) Schéma de la méthodologie pour l'agrégation de particules.

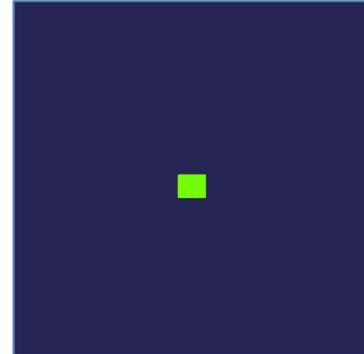
(b) Espace original d'une pièce.



(c) Agrégation de l'espace présenté en figure 6.13(b) (Distance=1).



(d) Agrégation de l'espace présenté en figure 6.13(b) (Distance=3).



(e) Agrégation de l'espace présenté en figure 6.13(b) (Distance=5).

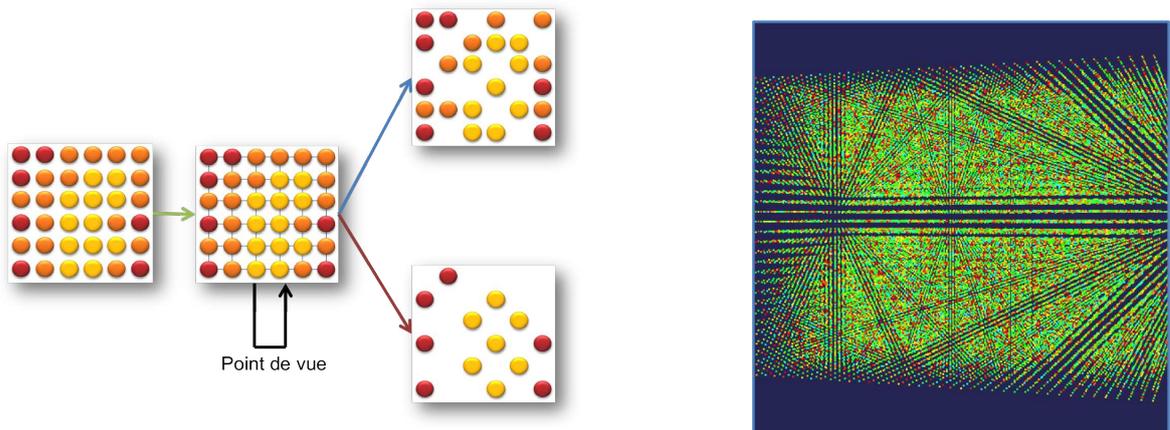
FIGURE 6.13 – Niveau de détails basé sur l'agrégation d'informations.

6.5 Outil de fouille de données

La dernière partie que nous allons étudier dans ce chapitre traite des différentes méthodes utilisées pour fouiller les données. La solution que nous avons proposée repose sur les modèles d'interaction usuelle, mais également sur les solutions d'extraction de contenu.

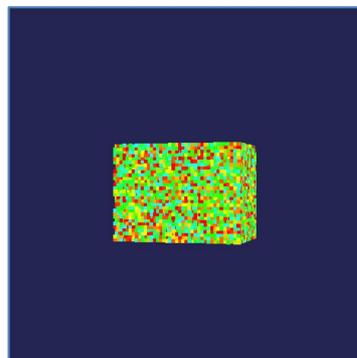
6.5.1 Modèle d'interaction

Pour manipuler les différentes vues, il est nécessaire de développer des outils efficaces pour manipuler l'information. Nous avons donc mis en place les interactions basiques



(a) Schéma de la méthodologie d'extraction de particules pour la mise en avant des points d'intérêts.

(b) Espace original d'une pièce.



(c) Extraction de points d'intérêts de l'espace présenté en figure 6.14(b) (Distance=1).

FIGURE 6.14 – Niveau de détails basé sur l'extraction de points d'intérêts.

de manipulation d'interface qui sont : le panning, le zoom, le filtre et la sélection. Ces méthodes d'interactions communes à de nombreux systèmes sont présentes dans notre solution. Notre jeu de données est basé sur des structures réelles. Il est alors nécessaire de pouvoir naviguer physiquement au travers de cette visualisation. Grâce aux outils de panning et zooming, il est possible de manipuler cette vue. Grâce aux solutions de filtrage, il est plus simple de réduire le nombre de données à visualiser sur un affichage. Il est alors possible de sélectionner seulement les points que l'on souhaite visualiser : soit dans un soucis de réduction de données, soit pour mettre en avant certaines parties importantes de la donnée. Un exemple de notre solution de filtrage est présenté en figure 6.15, cette solution permet de mettre en avant des objets dans notre vue⁵. Cette méthode permet une extraction du contenu au travers de différentes manipulations de l'utilisateur.

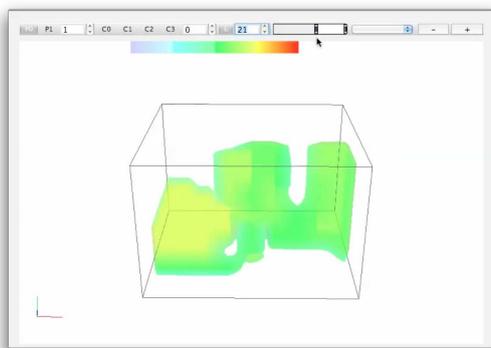


FIGURE 6.15 – Filtre appliqué sur un espace, seulement les valeurs les plus élevées sont conservées.

6.5.2 Extraction de contenu à partir de données réelles

Notre méthodologie de visualisation est orientée pour permettre de soumettre de nouveaux contenus aux modèles globaux de RIDER. Comme pour les différents composants développés autour de ce projet, il est important de donner la possibilité de modifier la structure des données et de proposer de nouveaux contenus au système. Dans cette partie, nous nous intéressons aux méthodes mises en place pour proposer ces nouveaux contenus. Deux approches ont été développées au cours de ces travaux : une approche basée composition et une approche basée sur la création de nouveaux éléments dans les pièces.

Une composition est une relation existant entre deux notions qui sont dépendantes l'une de l'autre. D'un point de vue statistique, il s'agit d'étudier l'intensité des liaisons qui peuvent exister entre deux variables. Dans le cadre de ces travaux, les compositions entre des capteurs sont appelées capteurs intelligents : smart sensors. Ce type de capteurs vir-

5. Une vidéo de présentation de cette méthode est disponible à l'adresse suivante : <http://benoitlange.fr/these/videos/filtre.mp4>

tuels est défini par l'IES (Institut d'Electronique du Sud) et un des exemples est le suivant :

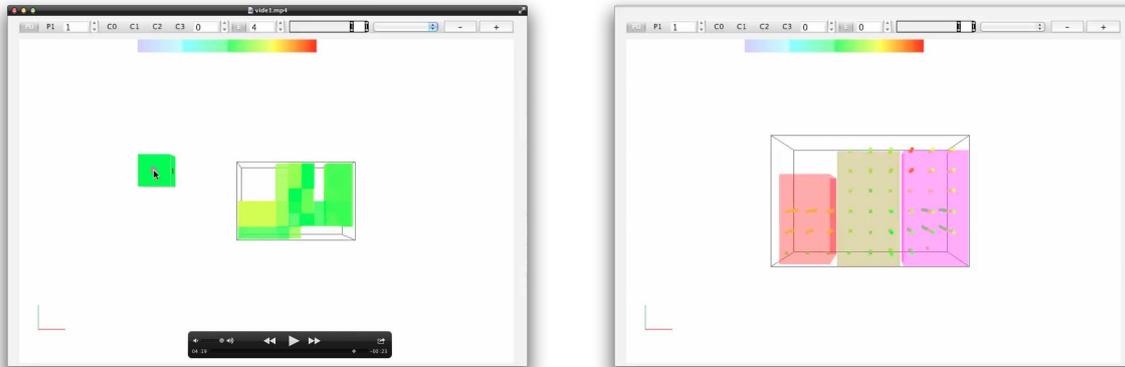
$$V_{\text{confort}} = V_{\text{température}} \times 50\% + V_{\text{humidité}} \times 35\% + V_{\text{pression}} \times 15\%.$$

Pour obtenir la valeur confort, il est nécessaire d'avoir les valeurs de température, humidité et pression atmosphérique. Grâce à ce type de composition, il est alors possible de créer des visualisations supplémentaires en fonction des différentes données disponibles. Il est donc possible d'expérimenter des compositions simples : deux éléments ; ou des compositions multiples pour permettre d'extraire du contenu sur des données. L'approche par composition permet de détecter des événements du bâtiment qui n'étaient pas déclarés précédemment.

La deuxième approche pour améliorer le modèle est basé sur la mise à jour de celui-ci par l'utilisateur et l'administrateur. Ainsi, nous fournissons des outils efficaces d'ajout d'informations pour le modèle. Les bâtiments sont des structures vivantes qui évoluent dans le temps. Ils sont construits autour de modèles numériques complexes qui peuvent ne pas être mis à jour régulièrement et qui souffrent également de problème d'accès (panne serveur par exemple). Les problèmes d'accès sont dû à des changements de propriété du bâtiment ou à des infrastructures trop anciennes (pas de plan numérique). De plus, les bâtiments sont utilisés par les personnes et évoluent dans le temps : par exemple des cloisons peuvent être déplacées, de nouveaux bureaux peuvent être créés et certaines cloisons peuvent disparaître. Seules les solutions de contrôle du bâtiment restent fixes tels que les capteurs et les appareils de climatisation. Le second problème de ce type de données est la cohérence entre la réalité et les informations du bâtiment. En effet, des données peuvent ne pas être renseignées. Ainsi, il est important de donner la possibilité de mettre à jour le modèle de données, mais également d'ajouter des informations manquantes.

Le modèle d'interactions que nous avons mis en place sur le modèle de bâtiment permet de modifier la topologie du bâtiment et également son contenu. Nous avons un outil basé sur un pavé extensible afin de permettre d'interagir avec le modèle de bâtiment. Grâce à ce pavé, il est alors possible de creuser, diviser des espaces, agrandir des espaces, mais également d'ajouter du contenu. En fonction de la méthode qui lui est appliquée, il est alors possible de mettre à jour le modèle du bâtiment. La figure 6.16(a) présente notre outil qui est composé d'un cube mobile et avec des capacités d'annotation. Comme l'illustre la figure 6.16(b), il est possible de localiser avec précision des éléments dans un espace ainsi que de l'annoter.⁶

6. Une vidéo de présentation de cette méthode est disponible à l'adresse suivante : <http://benoitlange.fr/these/videos/contenu.mp4>



(a) Outil d'annotation de modèle 3D.

(b) Résultat d'un espace annoté.

FIGURE 6.16 – Outil d'ajout de contenu sur le modèle de bâtiment existant.

6.6 Synthèse

Dans ce chapitre, nous avons étudié différentes solutions de visualisation que nous avons mises en place pour fouiller les données issues du bâtiment. Dans un premier temps, ces méthodes de visualisations reprenaient les standards des solutions de gestion de bâtiment, au travers de vues 2D. Notre contribution s'articule autour d'une architecture client / serveur. Le serveur est utilisé pour modéliser le bâtiment, tandis que le client est utilisé à des fins de visualisation de ce modèle. Nos différentes solutions de visualisation reposent soit sur une utilisation brute des données issues de notre modèle soit de données traitées. Le but de notre approche est de proposer une visualisation intuitive pour un client lambda afin qu'il puisse interpréter de manière efficace les données du bâtiment.

Enfin, des approches orientées autour des niveaux de détails ont été développées pour permettre de visualiser les données de manière efficace sur les différents périphériques. D'autres approches qui nous ont intéressées permettent de fournir des données au système global de RIDER. Le but est de permettre au système global d'améliorer le rendu de l'information. Les informations de bâtiment ne sont pas toujours suffisamment renseignées, il est alors nécessaire de rajouter de l'information manquante. Notre solution permet en plus de rajouter des informations qui ne sont pas disponibles dans les modèles numériques de bâtiment. Cet outil permet à l'administrateur du bâtiment de mettre facilement à jour la topologie de celui-ci. En effet, un bâtiment est très souvent vivant : la topologie de celui-ci ne fait qu'évoluer. Cet outil est également disponible pour les utilisateurs du bâtiment. Grâce à leur connaissance, il leur est alors possible d'ajouter des informations plus locales sur le bâtiment, notamment sur le contenu des différentes pièces.

Le chapitre suivant présente les résultats obtenus par la visualisation pour améliorer deux modèles de bâtiments existants.

Ces différents travaux ont donné lieu à des publications dans des conférences internationales : [Lange *et al.*, 2011a], [Lange *et al.*, 2011c] et [Lange *et al.*, 2011b].

Résultats et expérimentations

Si les ouvriers construisaient les bâtiments comme les développeurs écrivent leurs programmes, le premier pivot venu aurait détruit toute civilisation

G. WEINBERG

Sommaire

7.1	Introduction	111
7.2	Optimisation énergétique pour l'application serveur	112
7.3	Les données d'un centre de calcul : Green Data Center (GDC)	119
7.4	Les données d'un bâtiment tertiaire	124
7.5	Synthèse	130

7.1 Introduction

Dans ce chapitre, nous nous intéressons aux différents résultats que nous avons obtenus au travers de l'intégration de nos travaux sur des données réelles. Dans le cadre de ce projet, nous avons eu accès à deux jeux de données. Contrairement à ce qui était proposé à l'origine du projet, seul un petit ensemble d'éléments est disponible. Dans la section

7.2, nous présentons les différents résultats d'exécution de l'application serveur et nous proposons la meilleure architecture matérielle afin de réduire l'énergie consommée par cette application. Dans la section 7.3, nous présentons une analyse d'un centre de calcul possédant une forte densité de capteurs. L'analyse va nous permettre d'extraire certaines informations relatives à ce centre de calcul. Enfin, dans la section 7.4, nous analysons le comportement d'un bâtiment tertiaire au travers d'un jeu de données composé d'un petit ensemble de capteurs par pièces.

Dans les différentes représentations que nous allons trouver dans ce chapitre, les vues sont basées sur un gradient de couleur qui va du bleu vers le rouge en passant par le vert. Ainsi pour des valeurs faibles, la couleur utilisée tend vers le bleu tandis que pour les valeurs élevées, la couleur tend vers le rouge.

7.2 Optimisation énergétique pour l'application serveur

Les différentes données que nous avons sont issues de capteurs réels placés dans le bâtiment, nous allons donc réaliser différents traitements pour proposer des améliorations de contenu autour de ces modèles de faible granularité.

La mise en place d'un système intelligent de réduction d'énergie repose également sur un système peu coûteux en énergie. C'est-à-dire qu'il est indispensable de réduire au maximum le coût énergétique d'une solution de supervision, ceci afin de rendre optimale la solution. Notre premier effort repose sur la réduction de l'énergie consommée par notre solution et notamment sur la partie hautement consommatrice : le serveur.

Nous allons au travers de notre visualisation, utiliser les particules pour représenter l'espace des pièces. Comme nous l'avons montré précédemment, l'algorithme d'extraction de la zone d'influence des capteurs peut être représenté par l'algorithme 1.

Dans le cadre du projet RIDER, les solutions que nous proposons doivent également prendre en compte le paradigme d'économie d'énergie. Ainsi, tous les algorithmes doivent avoir une empreinte énergétique réduite. Pour réduire les coûts énergétiques liés à notre algorithme, nous avons évalué la meilleure méthodologie d'exécution de celui-ci.

Nous avons donc étudié le rendement énergétique lors de l'exécution de notre algorithme au travers de différentes méthodologies basées sur : un compilateur libre, un compilateur propriétaire et différentes méthodes de calcul parallèle. Notre algorithme a, quant à lui, été développé en C afin de réduire au minimum les coûts de calcul éventuels, mais également afin d'être compatible avec les méthodes de calcul parallèle sur GPU.

Les expérimentations énergétiques ont été réalisées sur deux approches de modélisation de notre espace par : grille régulière et sphères concentriques. Dans un premier temps,

```

S = AddSensors();
P = AddParticles();
T = TetrahedronExtraction(S);
for all T such as  $t_i$  do
  for all P such as  $p_j \in \text{AABB}(t_i)$  do
    if  $\text{Locate}(p_j, t_i) \geq 3$  then
       $p_j.\text{sensors.add}(t_i)$ 
       $P = P - p_j$ 
    end if
  end for
end for
for all P such as  $p_i$  do
   $p_i.\text{distance} = +\infty$ 
  for all S such as  $s_j$  do
    if  $p_i.\text{distance} > \text{distance}(p_i, s_j)$  then
       $p_i.\text{distance} = \text{distance}(p_i, s_j)$ 
       $p_i.\text{sensors.add}(s_j)$ 
    end if
  end for
end for
ParticlesPonderation();

```

Algorithm 1 Algorithme de localisation de particules.

nous avons étudié la consommation énergétique pour notre application serveur compilée en utilisant le compilateur GCC, mais également ICC, puis nous avons évalué les méthodes de calcul parallèle sur CPU, sur GPU, sur HPC (High Performance Computing) et enfin en hybridant CPU et GPU.

Afin d'évaluer différentes conditions d'utilisation de notre algorithme, nous avons utilisé un paramètre noté ϕ . Ce paramètre est défini dans le cadre d'une utilisation multi-résolution : pour différents dispositifs de visualisation. Nous l'avons fait varier de 0,01 à 5 (0,01 définit une haute résolution, tandis que 5 définit une résolution de faible granularité) ; la valeur d'usage est 1.

7.2.1 Mesure du temps d'exécution de notre algorithme

Notre première expérimentation consiste à évaluer le nombre de particules produites ainsi que le temps nécessaire pour dérouler l'algorithme. Les résultats sont présentés dans le tableau 7.1. Dans ce tableau, la première ligne représente le temps d'exécution né-

cessaire pour les différentes représentations, tandis que la deuxième ligne représente le nombre de particules générées.

	0,01	0,05	0,25	0,5	1	2	5
RG			39,288	4,763	0,550	0,034	0,002
RG taille			6481032	780912	90552	9884	840
SE	52,395	10,512	2,116	1,073	0,548	0,280	0,071
SE taille	8330976	1668563	335851	169333	86006	44260	19370

TABLE 7.1 – Résultat des différentes exécutions pour deux types de modélisation (RG pour grille régulière, SE pour méthode par sphères concentriques).

Le premier constat à faire grâce à ce tableau est que la méthode de particules par grille régulière est plus coûteuse en terme d'espace mémoire que la solution par sphère concentrique. En effet, sur l'intervalle $0,01 \leq \phi \leq 0,05$, l'espace mémoire nécessaire est trop important, il est alors impossible de générer toute la structure sans stocker de l'information sur le disque dur. Ensuite, nous pouvons constater que le nombre de particules générées par la solution sphérique est inférieur à la méthode par grille régulière sur l'intervalle $0,25 \leq \phi \leq 1$. Ceci est dû au nombre de particules de cette méthode. Enfin, sur l'intervalle $1 \leq \phi \leq 5$, le nombre de particules générées par la grille régulière est plus faible, le temps de calcul est alors réduit par rapport à la méthode sphérique. Pour un coefficient ϕ de 1, alors le temps d'exécution de la méthode est quasiment similaire. Une représentation du tableau précédent est donnée en figure 7.1.

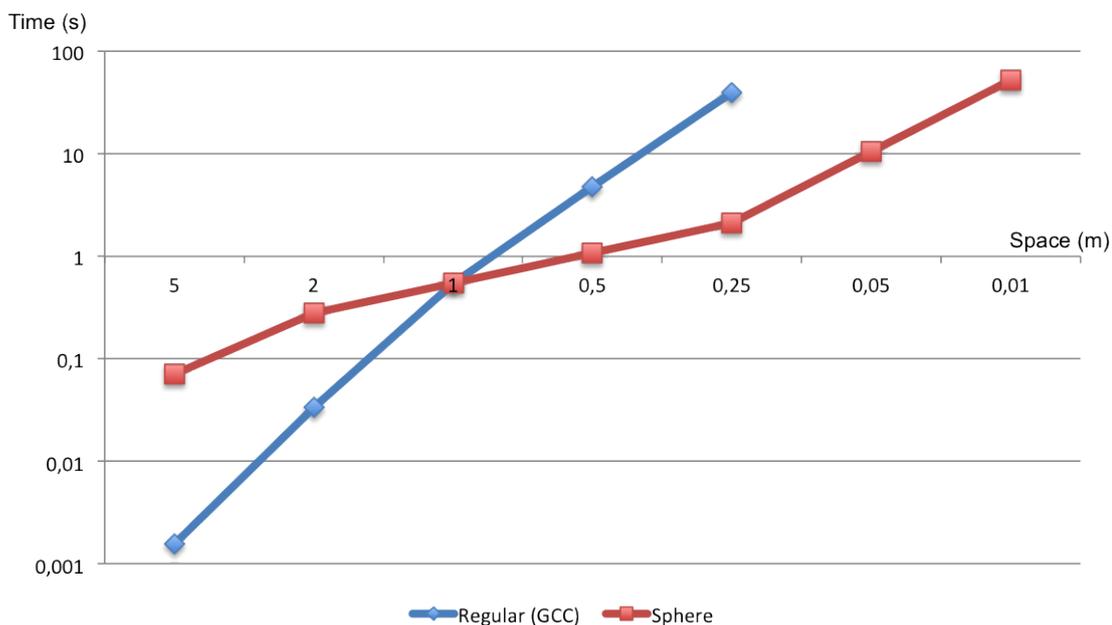


FIGURE 7.1 – Temps de calcul pour différentes résolutions en fonction du modèle de représentation utilisé.

Nous avons également évalué différentes méthodes de calcul parallèle pour déterminer quelle solution est optimale en terme de temps de calcul. Nous avons donc évalué une méthode basée sur un ordinateur standard au travers de la bibliothèque OpenMP, ensuite sur la même machine nous avons appliqué une méthode de calcul parallèle sur carte graphique au travers de la bibliothèque OpenCL. Enfin, nous avons utilisé une unité de calcul de type HPC. Pour réduire le nombre de combinaisons de résultats possibles, nous avons utilisé la grille régulière comme support d'algorithme. Les différents temps d'exécution sont présentés en figures 7.2 et 7.3. Lorsque l'algorithme utilise une faible résolution, les solutions de calcul parallèle coûtent plus cher que la méthode sans calcul parallèle. En effet, le fait d'utiliser une solution de parallélisation a un coût d'exécution supplémentaire.

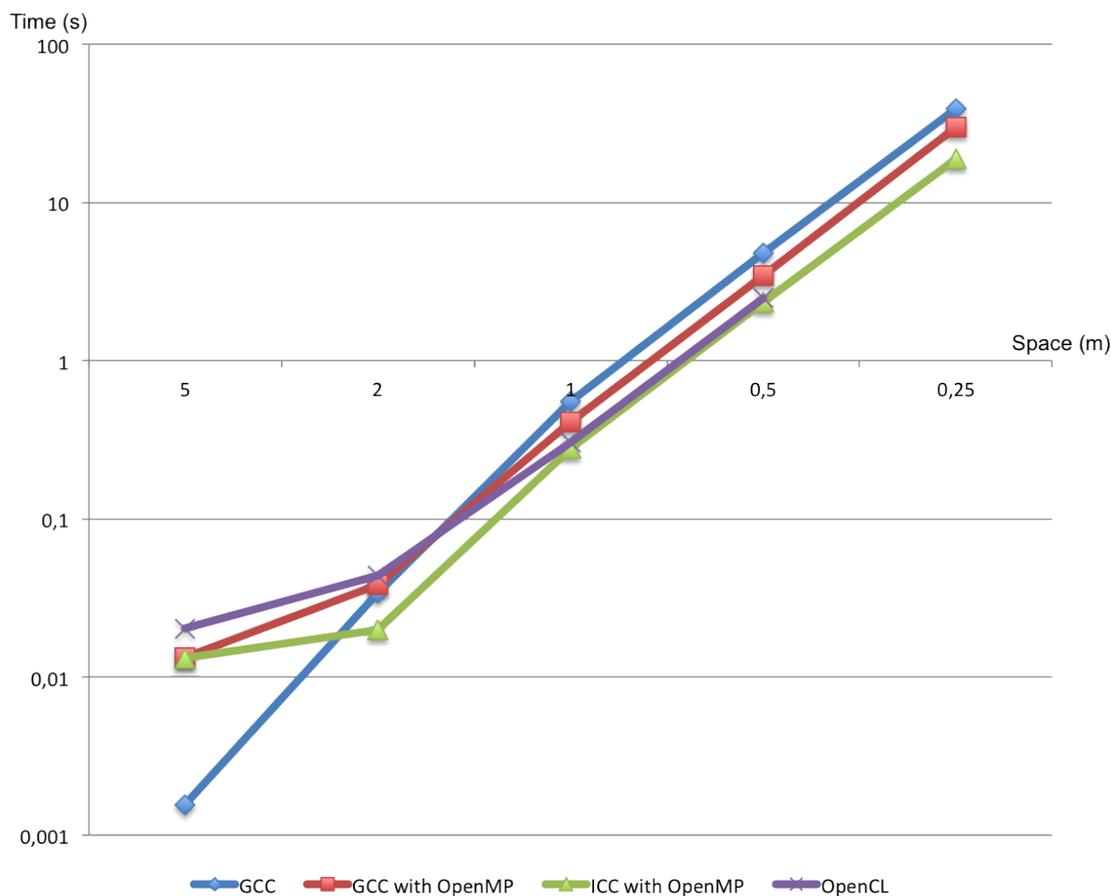


FIGURE 7.2 – Temps d'exécution sur grille régulière en utilisant OpenMP, ICC et OpenCL.

La méthode qui déporte les calculs sur GPU n'est rentable que sur des résolutions fines. Malheureusement la capacité en mémoire des cartes graphiques ne permet que de stocker qu'un nombre très limité de particules. Enfin sur la figure 7.2, nous voyons que la solution qui utilise ICC de manière parallèle est la solution la plus efficace en terme de temps de calcul. Sur la figure 7.3, nous voyons que la méthode par HPC est la plus efficace lorsque l'on effectue un calcul parallèle. La méthode hybride souffre, quant à elle, d'un manque d'efficacité : les problèmes sont dûs aux différents allers retours des données entre la mémoire centrale et la mémoire graphique.

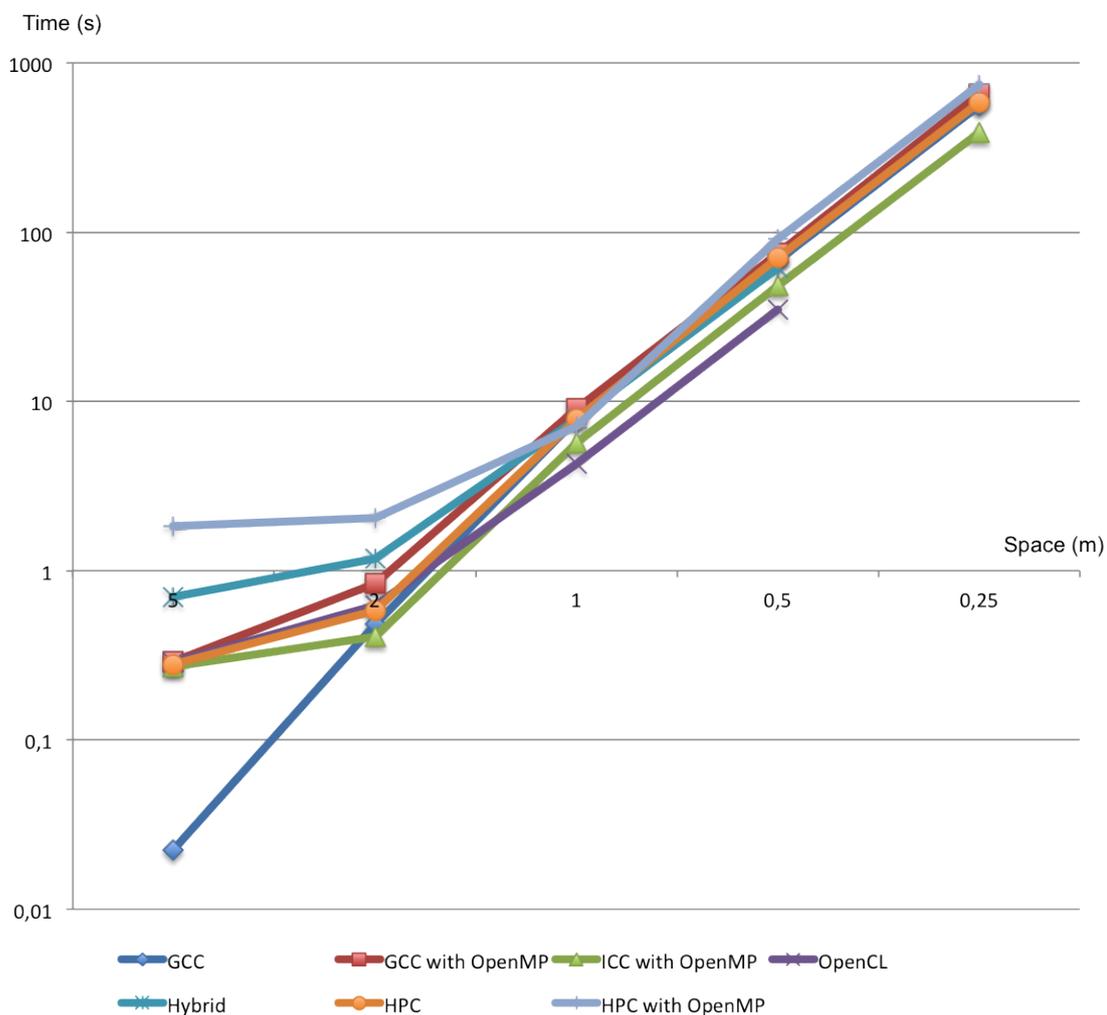


FIGURE 7.3 – Temps d'exécution sur grille régulière en utilisant une architecture hybride et un HPC.

7.2.2 Énergie dépensée pour le calcul

Nos dernières mesures se sont intéressées à l'énergie consommée par ces différentes méthodes. Nous souhaitons évaluer le coût énergétique introduit par notre méthode de calcul afin de ne pas dépenser le capital énergétique gagné lors de nos optimisations. Afin d'évaluer l'énergie maximale consommée, nous utilisons la formule suivante :

$$E = P \times T$$

avec E l'énergie mesurée en joules, P la puissance consommée en watts et T le temps d'exécution de notre algorithme. Les différentes mesures effectuées sont représentées dans la figure 7.4.

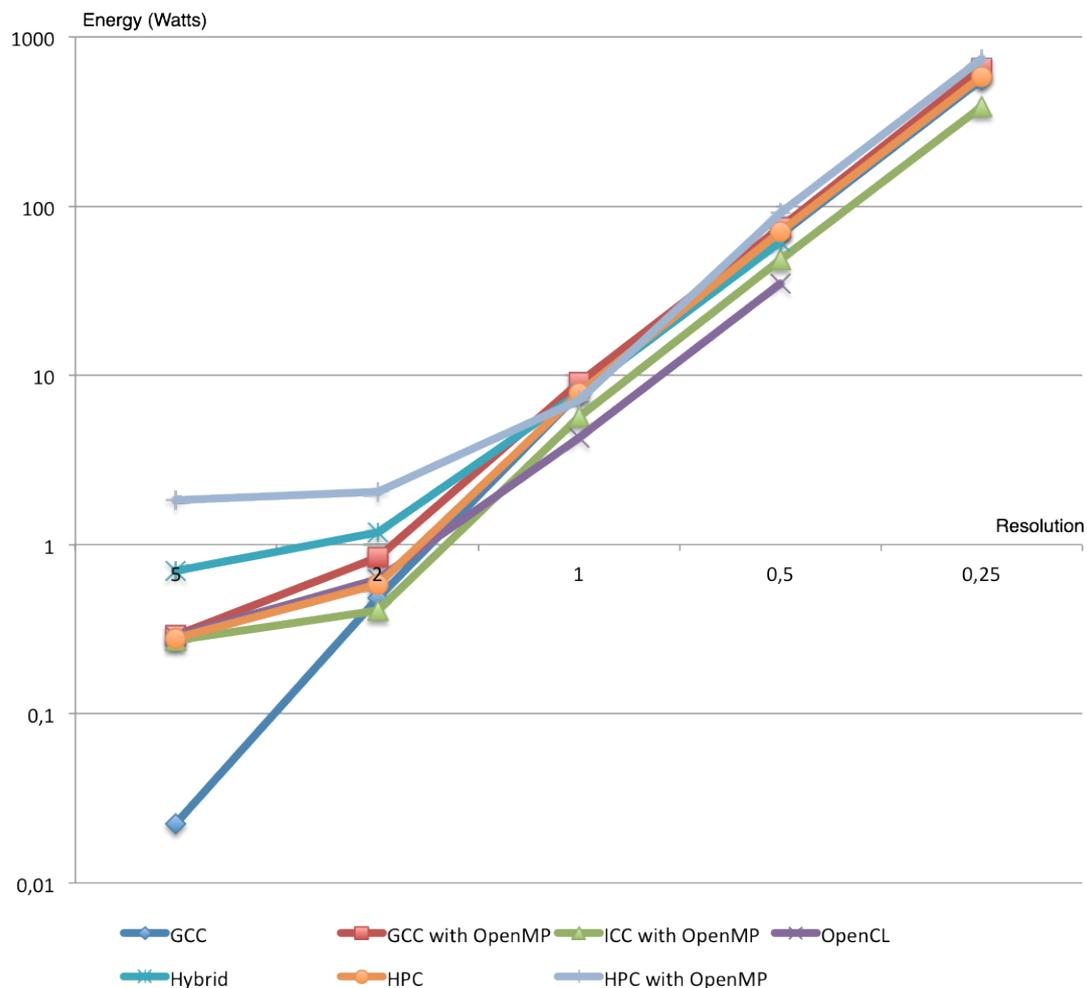


FIGURE 7.4 – Énergie mesurée sur les différents systèmes.

Afin d'économiser de l'énergie pour notre application serveur, il est nécessaire d'adapter l'architecture matérielle à la complexité du modèle. En effet, pour créer des rendus de faible résolution, un ordinateur standard avec une approche sans calcul parallèle est bien plus optimisé pour réduire le coût énergétique de notre solution. Pour une résolution élevée, il est nécessaire d'utiliser une solution de calcul parallèle. En effet, le résultat énergétique est inférieur aux solutions standards. Afin de réduire encore plus le coût énergétique,

une approche par GPU est recommandée, malheureusement ce type d'approche reste limitée en terme de mémoire disponible.

Enfin sur l'approche énergétique, des solutions de types centres de calcul sont inadap- tées. En effet, leur consommation énergétique est supérieure dans quasiment toutes les résolutions.

7.3 Les données d'un centre de calcul : Green Data Center (GDC)

Au sein d'IBM Montpellier, différents centres de calcul sont présents. Un de ces centres de calcul est particulier dans le sens qu'il a été construit afin d'être économique en énergie. Un centre de calcul écologique est un centre de calcul possédant un PUE faible. Le PUE est donc l'unité de mesure de consommation énergétique des centres de calcul. La mesure de PUE est effectuée en divisant la puissance énergétique en entrée par la puissance informatique, une représentation de cette méthode de calcul est présentée en figure 7.5. Ainsi, plus la mesure de PUE est faible, plus le centre de calcul est efficace énergiquement.

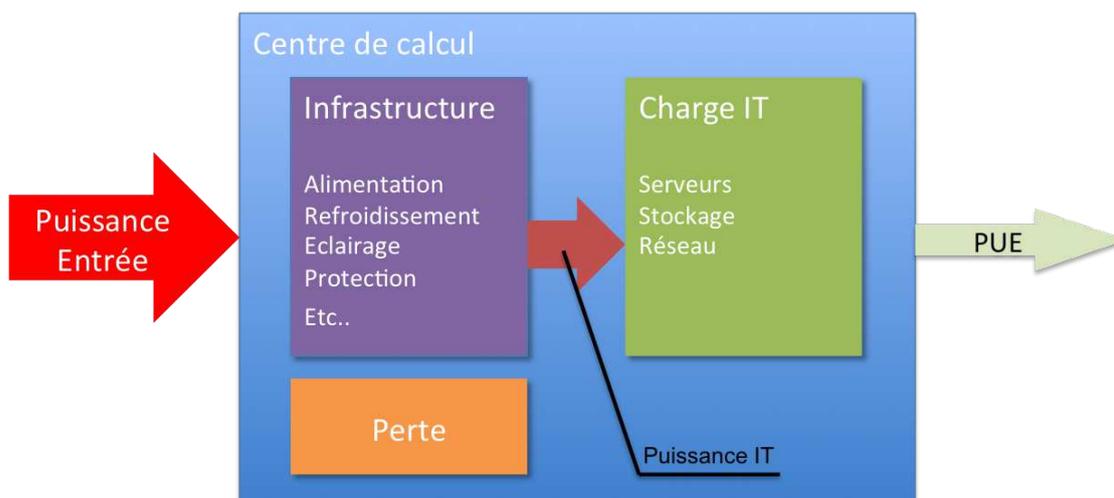


FIGURE 7.5 – Le PUE est l'unité de mesure des centres de calcul. Cette mesure est réalisée en divisant la puissance d'entrée par la puissance informatique utile.

Le centre de calcul écologique d'IBM Montpellier est construit pour atteindre un PUE très faible. Cette vitrine technologique est composée de deux structures : une partie haute

densité et une partie faible densité. Dans le cadre du centre de calcul, la densité est évaluée en fonction du nombre d'unités de calculs au centimètre carré. La salle composée de machines haute densité est refroidie de manière spécifique. Différents boîtiers de refroidissement sont mis en place et sont basés sur des solutions économiques en énergie. Il n'est alors plus nécessaire d'utiliser de solution de climatisation générale. La salle basse densité, quant à elle, repose sur le cloisonnement des différentes allées. Les machines aspirent de l'air frais depuis la face avant et au niveau du sol, qui est ensuite rejeté par l'arrière. Plus l'air est chaud, plus il s'élève et il se réchauffe en plus au contact des lames situées sur son passage. La figure 7.6 présente la structure du centre de calcul, grâce à ces différentes optimisations énergétiques, ce centre de calcul possède un PUE proche de 1.5.

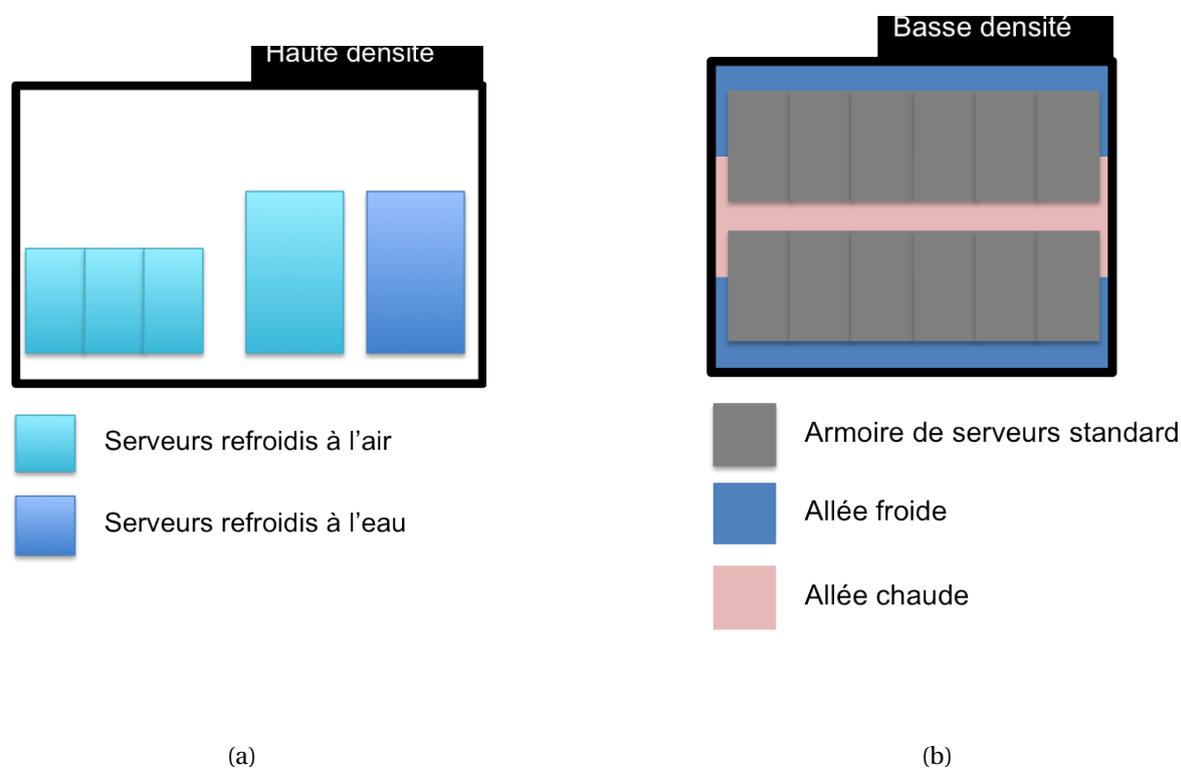
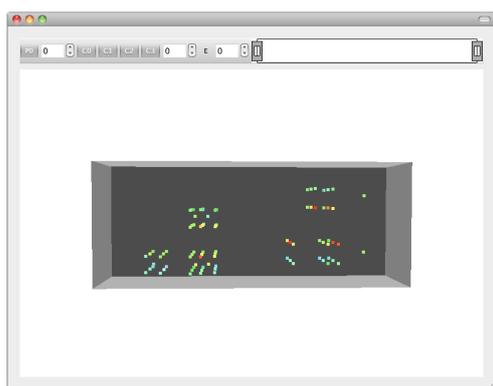


FIGURE 7.6 – Représentation schématique du centre de calcul écologique d'IBM, Montpellier.

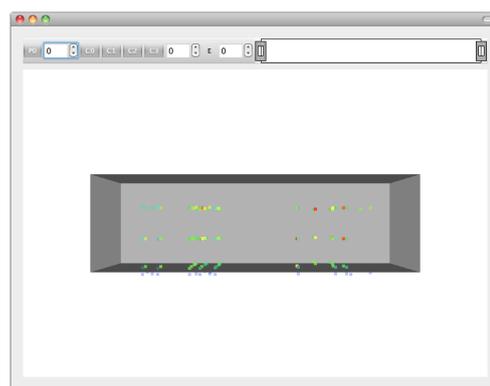
Afin de contrôler le bon fonctionnement de ce centre de calcul, différents capteurs sont placés dans les pièces afin de mesurer les caractéristiques critiques de celles-ci. Pour ce faire, quatre types de capteurs sont mis en place : température, pression, hygrométrie et niveau de batterie des onduleurs. La structure de données des capteurs est composée de peu d'éléments : la localisation (x, y, z) , la valeur de captation, le type de capteurs et les bornes supérieures et inférieures. Les données du centre de calcul sont seulement com-

posées des informations des capteurs. En effet, les informations à propos des systèmes de climatisation (crac), la localisation des serveurs, la localisation des allées, ainsi que la topologie réelle des deux pièces ne sont pas disponibles. Ces différentes informations seront à raffiner grâce à notre solution.

Les données du centre de calcul sont composées de quatre types de capteurs : température, pression, hygrométrie et niveau de batterie. Ces différentes valeurs sont mesurées afin de gérer le comportement du centre de calcul et surtout de relever des incidents. Ces différents capteurs sont également localisés, la solution de gestion du bâtiment (BMS) stocke donc leur position spatiale. Les données sont également composées d'informations relatives à la topologie globale des deux salles réunies. Les figures 7.7(a) et 7.7(b) présentent deux points de vues des données qui nous ont été fournies ; comme nous pouvons le voir les capteurs sont disposés dans une pièce globale qui est une agrégation des deux pièces (haute densité et faible densité).



(a) Vue de dessus des données fournies par IBM au sujet du centre de calcul écologique.



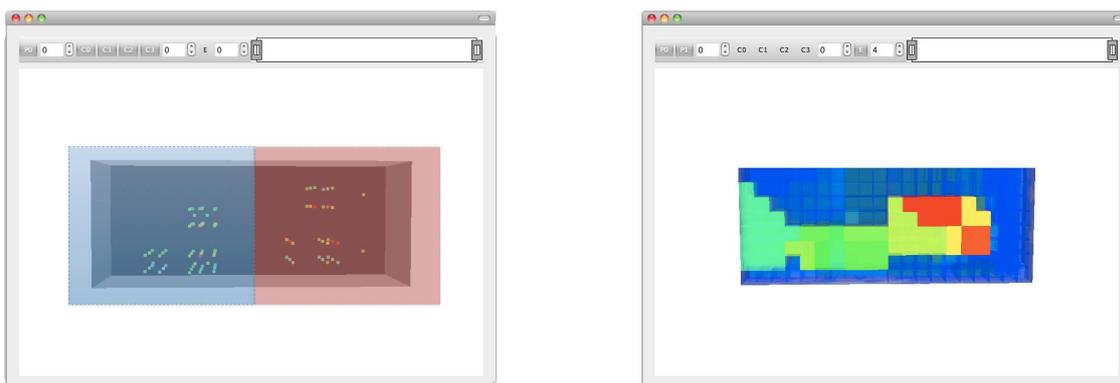
(b) Vue de face de ces mêmes données.

FIGURE 7.7 – Vue brute des données du GDC.

Comme nous pouvons le voir au travers de cette visualisation, beaucoup d'informations sont manquantes. Il est donc nécessaire, voir important, d'améliorer le modèle de données que nous possédons. Ici, nous allons montrer comment améliorer ce modèle au travers d'une visualisation volumique.

Afin de segmenter les différentes pièces, nous utilisons les différentes informations de localisation des capteurs. Ainsi, comme le montre la figure 7.8(a), deux paquets de capteurs sont clairement présents, il est alors possible de segmenter les deux pièces grâce à nos outils. Afin d'identifier les pièces (haute densité et basse densité), nous allons nous baser sur une vue globale volumique de ces pièces. La figure 7.8(b) présente une vue globale des

données au travers d'une représentation volumique (rappelons que la figure 7.6 illustre ce que nous connaissons des données). Nous allons nous servir de l'information relative à la segmentation des allées sur la partie faible densité pour identifier les pièces. Il s'agira de notre hypothèse de base que nous chercherons à valider tout au long de cette analyse.



(a) Possibilité de segmentation des deux pièces afin d'extraire une première information topologique.

(b) Vue volumique des deux pièces réunies.

FIGURE 7.8 – Méthode de visualisation améliorée des données brutes et segmentation de l'espace du GDC.

7.3.1 Analyse de la pièce haute densité

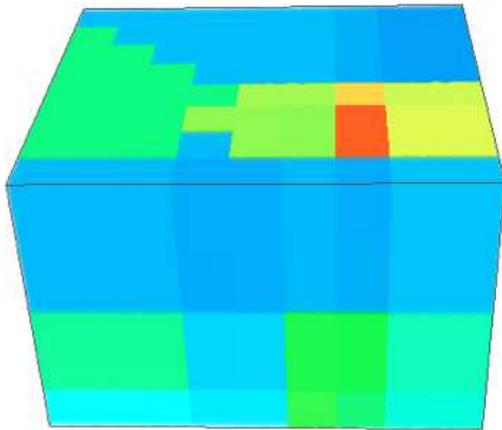
Sur la partie haute densité, nous allons utiliser différents filtres et visualisations afin d'identifier des informations adaptées à nos connaissances.

En utilisant la solution à base de voxels, (figure 7.9(a)) aucune information permettant d'augmenter le modèle n'est visible. Lorsque l'on utilise une des méthodes basées sur les shaders, différentes structures commencent à apparaître, comme illustré en figure 7.9(b). En utilisant la visualisation par température de la salle, il est possible d'extraire des informations : en filtrant les données nous obtenons la figure 7.9(c) où deux masses émergent. Grâce à nos connaissances du terrain, il est possible d'identifier l'origine de ces blocs : il s'agit des emplacements des deux unités de calcul refroidies soit par air soit par eau.

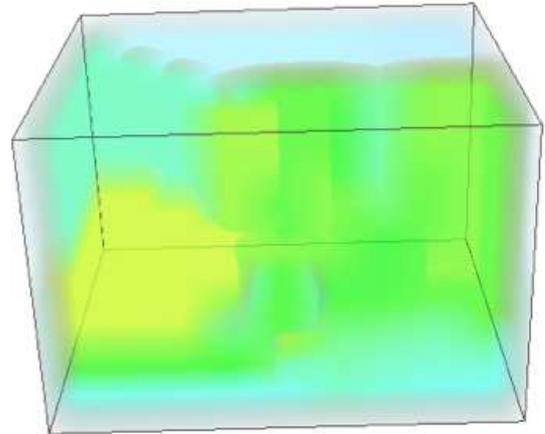
En analysant les parties chaudes, il est possible de se rendre compte qu'une dernière masse est présente. Il s'agit de la partie du centre de calcul variable, cette partie est modifiée en fonction des usages et des besoins clients.

Grâce à notre outil de soumission de contenu, il est alors possible de localiser le mobilier particulier de cette pièce et ainsi d'améliorer le modèle de la pièce. Ces informations

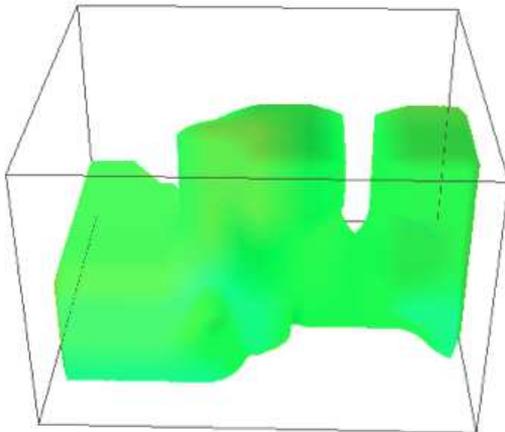
pourront ensuite être réutilisées pour raffiner le modèle physique du bâtiment et optimiser l'utilisation des pièces. Ces optimisations peuvent être proposées au modèle afin d'améliorer le rendu, comme illustré en figure 7.9(d).



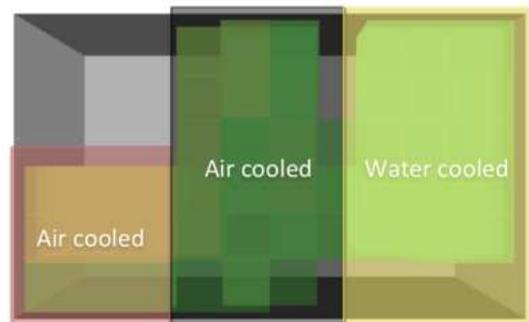
(a) Visualisation volumique de la pièce haute densité en utilisant des voxels non transparents.



(b) Visualisation volumique de la pièce en utilisant les shaders.



(c) Visualisation volumique de la pièce en filtrant les données sur les températures les plus élevées.



(d) Informations extraites à partir de notre visualisation.

FIGURE 7.9 – Différentes vues de la pièce haute densité.

7.3.2 Analyse de la pièce basse densité

Sur la partie basse densité, nous avons utilisé le même processus. Grâce aux particules interpolées, il est possible d'avoir un point de vue relatif à l'état de la salle de calcul. Ainsi,

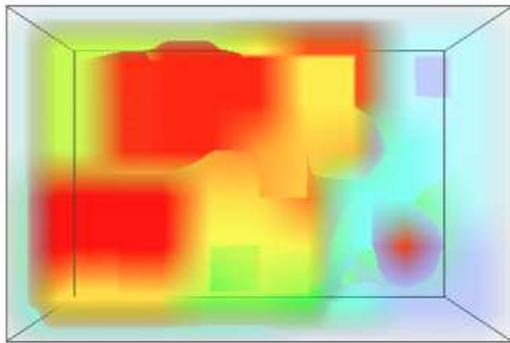
nous cherchons également à détecter des informations en fonction des données connues. Pour ce faire, nous allons nous intéresser principalement aux températures, les autres variables ne fournissent pas d'informations suffisamment significatives. En appliquant la même méthode sur la partie faible densité, la figure obtenue est la 7.10(a). En filtrant les données, trois régions d'intérêts sont mises en avant. La figure 7.10(b) présente les températures les plus élevées de la salle, la figure 7.10(c) montre les températures basses de la salle et enfin, le dernier filtre utilisé en figure 7.10(d) extrait les températures les plus faibles de la pièce. Comme nous l'avons présenté précédemment (cf figure 7.6), 3 zones étaient connues dans le modèle original. Nous avons retrouvé ces trois zones, mais également une quatrième.

En visitant le centre de calcul, nous nous sommes aperçus que ces températures de faible intensité correspondent à la partie CRAC de ce centre de calcul. En effet, la pièce d'extraction d'air est également surveillée par des capteurs. Nous avons donc pu soumettre ces différents contenus au modèle global de RIDER et ainsi améliorer le modèle du centre de calcul. Le modèle précédent peut être amélioré et permet de proposer une nouvelle vue du centre de calcul comme illustré en figure 7.11.

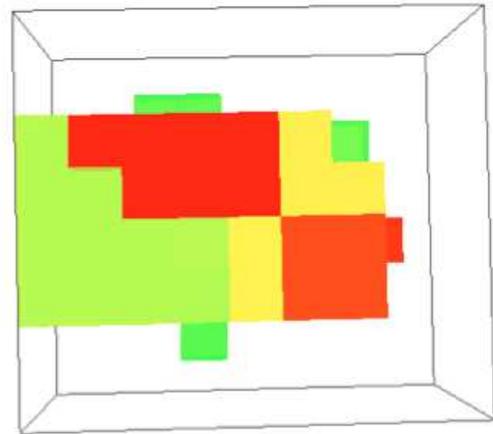
Une seconde utilisation de notre outil permet d'extraire un autre type d'informations. En utilisant la temporalité de notre visualisation, il est possible de mettre en avant les zones qui varient dans le temps. Ainsi, nous avons pu constater que sur la partie haute du centre de calcul, certaines lames de serveurs chauffaient plus que les autres. En prenant en compte la dynamique des fluides (les flux d'air chaud montent), il est nécessaire de réorganiser l'armoire afin de réduire l'énergie nécessaire pour la refroidir. En effet, la température au sommet de l'armoire est plus élevée qu'en bas : le refroidissement étant réalisé par le sol. Identifier des lames de calculs les plus utilisées est recommandé afin de proposer un placement optimale de celles-ci.

7.4 Les données d'un bâtiment tertiaire

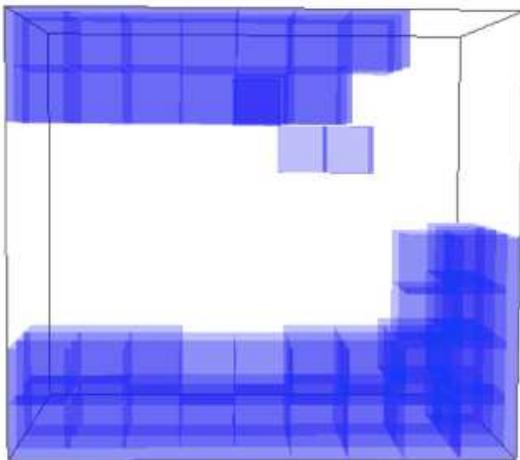
Notre deuxième jeu de données est issu de capteurs localisés dans une série de bureaux. Chaque bureau est équipé d'un cluster de capteurs : généralement deux mesures. Certaines pièces ont été sur-instrumentées et possèdent alors un capteur supplémentaire. Les informations que nous avons au sujet de ces données sont donc les valeurs des capteurs, leur localisation (identifiant du bureau) et enfin le plan des bureaux. La figure 7.12 présente le plan des bureaux, ainsi que les différents types de capteurs présents. Contrairement aux données précédentes, les pièces ne disposent pas de plusieurs capteurs de même type hormis le couloir. Ce plan est segmenté en quatre régions, comme le montre la figure 7.13. Il est composé de bureaux avec une utilisation régulière du lundi au vendredi (zone rouge), il est également composé d'un couloir (zone orange), de salles de réunion (zone noire) et de salles de formation (zone verte). Les salles de formation et réunion sont utili-



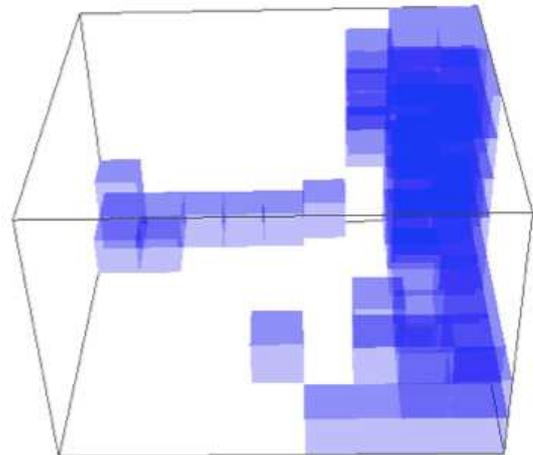
(a) Vue de dessus de la partie basse densité du centre de calcul.



(b) Vue de dessus des températures élevées de la zone faible densité.



(c) Filtre sur les températures basses de la zone basse densité (vue de dessus).



(d) Filtre sur les températures très basses de la zone basse densité (vue à 45°).

FIGURE 7.10 – Différentes vues de la pièce haute densité.

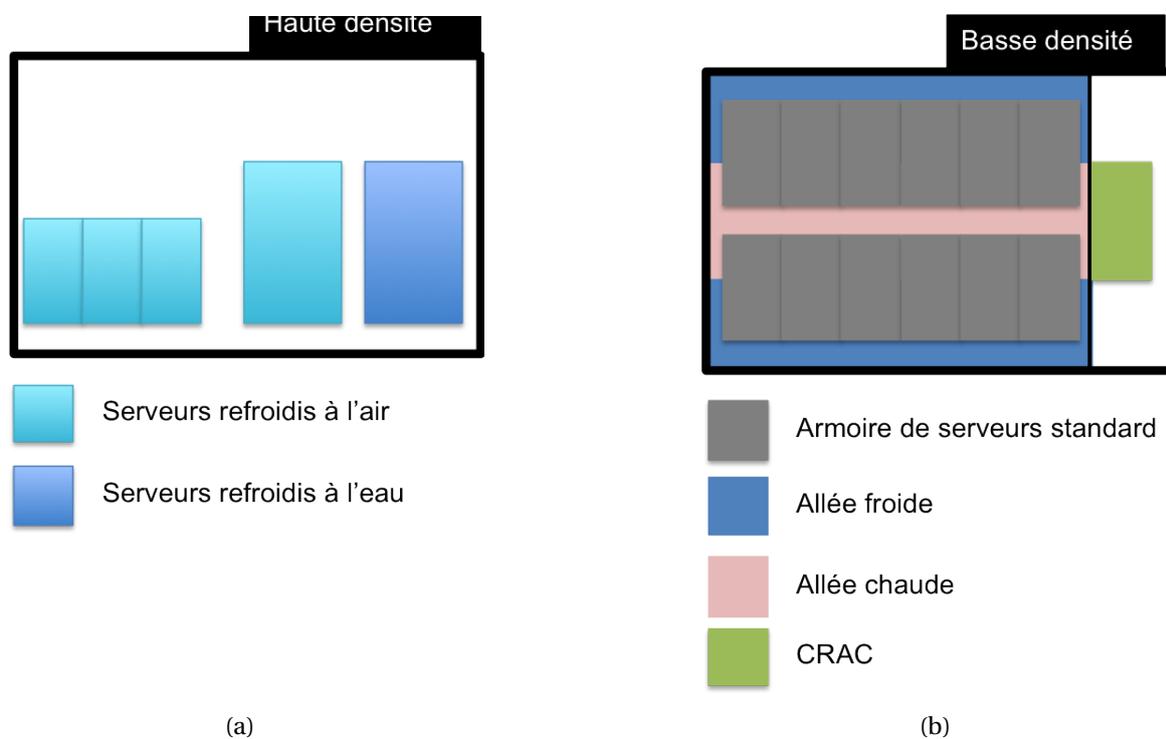


FIGURE 7.11 – Représentation améliorée du schéma du centre de calcul.

sées ponctuellement. La zone entourée de bleu possède en plus un capteur d'hygrométrie. Les données de ce bâtiment sont accessibles par deux méthodes : un service web proposé par un des partenaires du projet ou un entrepôt de données, où sont stockées toutes les informations.



FIGURE 7.12 – Schéma du bâtiment fourni par le système de gestion du bâtiment.

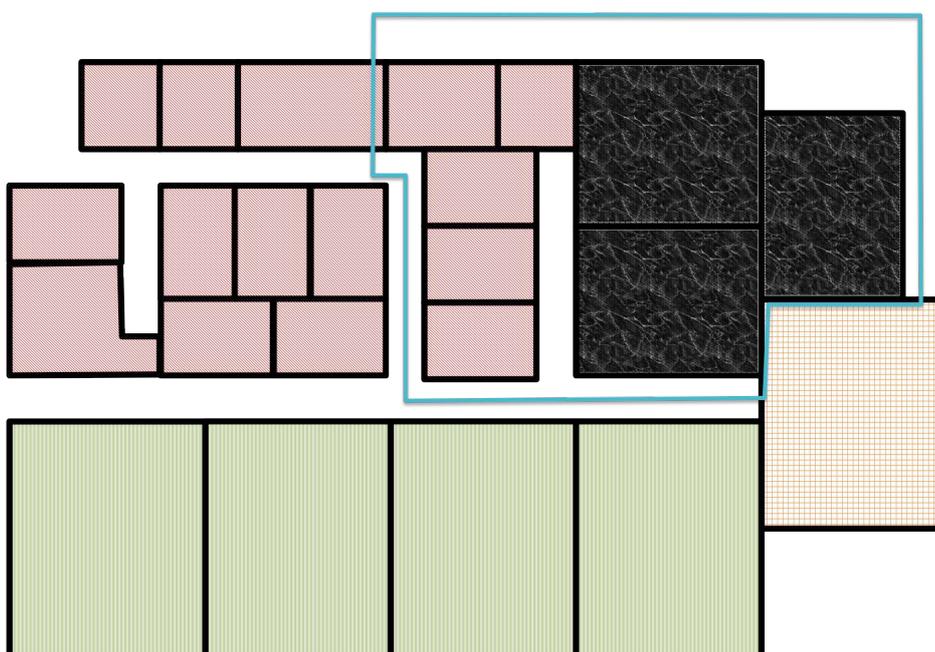
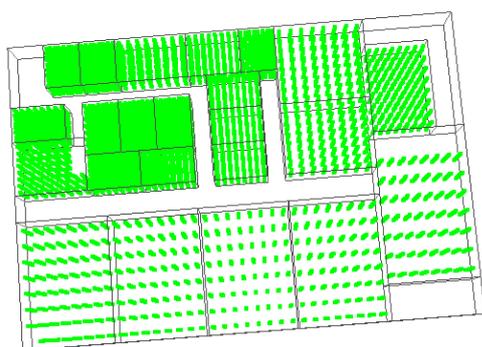
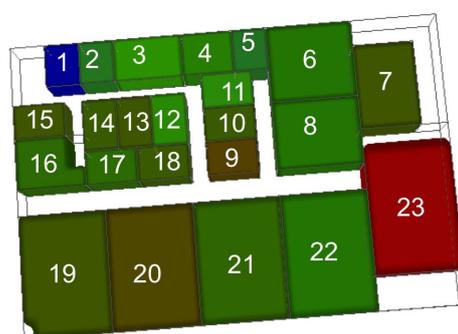


FIGURE 7.13 – Représentation du bâtiment en utilisant notre format. Quatre types de salles sont mis en avant : la zone rouge représente les bureaux, la zone noire les salles de réunions, la zone orange est un couloir et la zone verte représente les salles de formations.

La première étape que nous avons dû mettre en place a été de modéliser les données en utilisant notre modèle de bâtiment. Nous avons ensuite pu explorer les données en utilisant notre visualisation. Dans cette analyse, nous avons pu utiliser toutes les valeurs de capteurs car le contenu était pertinent. Afin de permettre de visualiser correctement les données, il est nécessaire de normaliser l'échelle de valeur de la visualisation, en effet, comme le montre la figure 7.14(a), en utilisant l'échelle standard des données, les différentes pièces semblent en parfait état de fonctionnement sans variabilité perceptible.



(a) Visualisation des données de température du bâtiment sans normalisation.

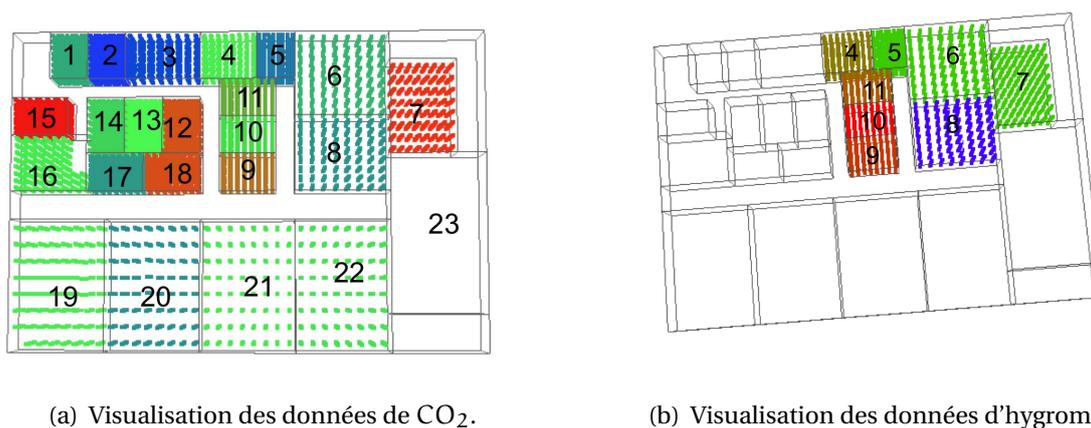


(b) Visualisation des données de température en utilisant les informations de minimum et maximum.

FIGURE 7.14 – Vue des données des capteurs de températures.

Notre premier intérêt s'est focalisé sur le capteur de température. Ainsi en normalisant les données sur les maxima et minima, le résultat obtenu est présenté en figure 7.14(b). Nous pouvons voir au travers de cette figure, que différentes informations apparaissent dans ce résultat. La zone 23 (représentant le couloir) est la zone la plus chaude. La première hypothèse que nous pouvons réaliser est : au moment de la mesure, une forte source de chaleur est présente, ou les halles de couloirs sont trop chauffées. Le deuxième point d'intérêt est présent dans la zone 1 qui est la plus froide : deux explications sont possibles : la pièce se trouve à proximité d'une porte de sortie d'urgence et l'isolation est mal réalisée, la seconde hypothèse est que l'activité de ce bureau est très faible. D'autres salles sont remarquables comme les pièces 7, 9, 20. Ces zones semblent avoir une activité anormale comparée aux pièces similaires, par exemple pour la pièce 20, les pièces similaires sont 19, 21 et 22. Une première analyse plus approfondie et temporelle va nous permettre de mieux comprendre ces informations. La zone 23 est en permanence la zone la plus haute en terme de température, pourtant la température ressentie est une des plus faibles du bâtiment. Ainsi, nous avons étudié le placement des deux capteurs de température de cette

zone. Il s'avère que ceux si sont placés soit derrière un écran de télévision allumé, soit au niveau d'une bouche de soufflage d'air chaud. Il est alors important de tirer une conclusion sur ces données : les mesures effectuées ne sont pas fiables, afin de pouvoir exploiter au mieux ces données, il est nécessaire de relocaliser ces capteurs.

(a) Visualisation des données de CO₂.

(b) Visualisation des données d'hygrométrie.

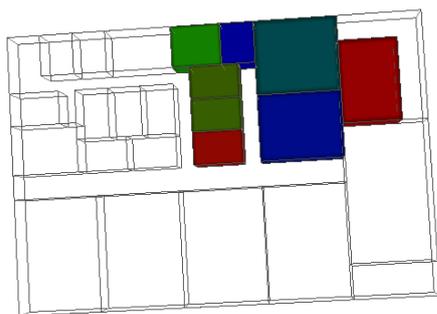
FIGURE 7.15 – Vue de deux types de capteurs.

La seconde analyse que nous avons réalisée repose sur les mesures de CO₂. Sur la figure 7.15(a), différents points remarquables apparaissent. Ainsi, nous nous sommes intéressés aux zones 1, 2, 3, 5, 7, 15, 12, 18, 9. En combinant ces deux vues, il est possible de déterminer que la salle 20 est occupée par des personnes : le taux de CO₂ ainsi que la température sont élevés par rapport aux pièces similaires : 19, 21 et 22, la salle 19 semble être alors influencée par l'élévation de température de la salle 20. Deux questions se posent : les capteurs sont-ils trop près l'un de l'autre, ou l'isolation entre les deux pièces est mal réalisée. La salle 7 semble également occupée, le taux de CO₂ ainsi que la température sont élevés dans cette zone comparée aux pièces voisines qui ont le même profil. Enfin les différents bureaux avec un fort taux de CO₂ et également des températures élevées sont identifiés comme étant occupés par des personnes.

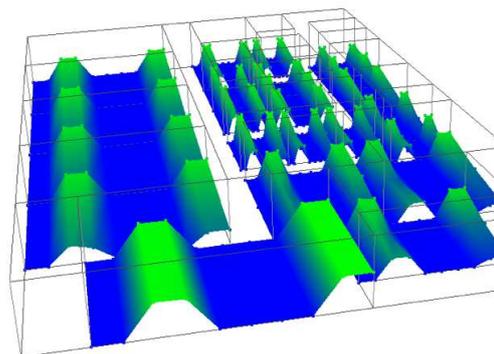
Enfin, nous utilisons la dernière source d'information des données : les capteurs d'hygrométrie. En étudiant la représentation présente en figure 7.15(b), seule une petite zone est instrumentée, ainsi, les salles de réunion n'apportent pas d'information supplémentaire. Par contre, les bureaux ayant un taux d'hygrométrie très élevé peuvent être considérés comme occupés.

Grâce à cet outil, il est possible d'identifier l'activité humaine des différentes salles en analysant leur profil, des questions d'isolation sont levées et seule une visite sur place peut permettre de répondre à ces interrogations. La conclusion que nous pouvons émettre repose sur l'utilisation des bureaux, il est nécessaire de corréliser les informations de tem-

pérature, CO₂ et hygrométrie pour régler l'activité des zones d'un point de vue humain. Cette approche est expérimentée en figure 7.16(a). Les différentes remarques permettent de montrer l'occupation des différentes zones.



(a) Visualisation de corrélation entre les données de CO₂, température et hygrométrie.



(b) Visualisation à base de surface des données, apparition de "trous" dans les données.

FIGURE 7.16 – Vue de données corrélées et visualisation sous forme de plans brisés.

Afin de mieux appréhender les données, nous nous sommes également intéressés à la visualisation temporelle. Nous avons donc utilisé notre solution de visualisation temporelle basée sur l'extension des lignes brisées pour comprendre et analyser le comportement des pièces. En utilisant cette méthode, différentes informations émergent, comme le montre la figure 7.16(b) : des creux dans la base de données sont visibles. Ainsi, en analysant les informations, nous pouvons voir que le système était éteint à ce moment-là, et aucune information n'a été enregistrée. Afin d'évaluer cette défaillance, nous nous sommes basés sur les informations relatives à la période de Noël qui correspondait à une extinction globale de tous les systèmes et nous avons comparé les données lors de ces manques d'informations. Ainsi nous avons pu mettre en avant que durant certaines périodes, les données n'étaient pas utilisables. Nous avons donc réalisé un outil de fouille de données visuelles qui permet de connaître les données utilisables dans le cadre d'incidents.

7.5 Synthèse

Dans ce chapitre, nous avons présenté les différents résultats pratiques que nous avons obtenus au cours de ces travaux. Nous avons présenté trois expérimentations : une analyse

d'architecture, une analyse de données issues d'un centre de calcul et enfin une analyse de bâtiment tertiaire.

Dans le cadre de l'analyse d'architecture, nous avons voulu évaluer notre solution de modélisation et déterminer quelle architecture matérielle est nécessaire afin de ne pas surcharger significativement le coût énergétique d'une solution de gestion de bâtiment. Ainsi, il apparaît que selon le bâtiment géré, la solution à mettre en place varie. Pour une faible granularité de données, il est nécessaire d'utiliser une architecture de type ordinateur standard, tandis que pour une visualisation plus fine, il est souhaitable énergiquement d'utiliser un ordinateur standard avec une méthode de calcul parallèle.

Ensuite, nous nous sommes intéressés à l'analyse des données du centre de calcul d'IBM Montpellier. Nous avons exploré les différentes visualisations proposées au cours de ces travaux pour améliorer le modèle de données fourni par le projet RIDER. Ainsi grâce à une forte granularité des capteurs, il nous a été possible d'extraire un contenu précis de la topologie interne à ce centre de calcul.

Enfin, nous avons également étudié un jeu de données issues d'un bâtiment tertiaire. Malheureusement l'instrumentation de ce bâtiment est très faible et seules des informations relatives à la vie du bâtiment sont remarquables.

Un des problèmes majeurs de nos travaux a été causé par les problèmes d'accès aux données. En effet, celle-ci n'ont été accessibles que tardivement et de manière non automatique dans un premier temps (données du centre de calcul). Pour les données du bâtiment, différents problèmes sont apparus notamment au niveau du temps de réponse du système, la solution de base de données utilisée répondait de manière lente.

Ces différents travaux ont été présentés lors d'une conférence internationale [Lange *et al.*, 2011d] et également dans un chapitre de livre [Lange *et al.*, 2012].

Conclusions et perspectives

L'informatique, ça fait gagner
beaucoup de temps... à condition
d'en avoir beaucoup devant soi!

MIREILLE SITBON

Sommaire

8.1	Résumé de nos contributions	133
8.2	Difficultés rencontrées	135
8.3	Limitations	136
8.4	Perspectives	136

8.1 Résumé de nos contributions

De nos jours, l'économie d'énergie est devenue un enjeu crucial. Les bâtiments où nous travaillons et qui nous servent d'habitation ont été identifiés comme étant une source importante de perte énergétique. De ce constat a émergé le projet RIDER (Research for IT Driven EnERgy efficiency) lequel a pour objectif de développer un système d'information innovant permettant d'optimiser la consommation énergétique d'un bâtiment ou d'un groupe de bâtiments. Ce système est basé sur des composants logiciels, notamment un composant général de modélisation du bâtiment, un composant orienté fouille de don-

nées et un composant de visualisation. Chacun de ces composants est destiné à améliorer le modèle de données du noyau RIDER.

Cette thèse porte sur le composant visualisation que nous avons conçu et développé pour le système RIDER. Plusieurs axes ont été traités concernant la modélisation et la visualisation des données provenant des capteurs du bâtiment, ainsi que l'interaction aussi bien avec l'utilisateur qu'avec le modèle de données RIDER. Dans ce manuscrit nous avons présenté le modèle que nous avons défini pour représenter l'espace intérieur des bâtiments et qui sert de base aux différentes techniques de visualisation de données hétérogènes offertes par le module. L'objectif de l'ensemble de techniques est double : d'une part, permettre à l'utilisateur d'explorer au mieux les données et d'autre part, permettre la mise à jour du modèle du noyau grâce aux informations découvertes par l'utilisateur par le biais des diverses vues. Les jeux de données disponibles sur les pilotes du système RIDER nous ont permis de tester et évaluer la performance des solutions développées. Ces jeux de données proviennent du centre de calcul vert d'IBM et d'un bâtiment industriel et contiennent des mesures issues de capteurs de température, hygrométrie, pression, niveau de batterie des onduleurs, CO₂.

Nos travaux se sont structurés autour de deux axes principaux :

- **Un ensemble de techniques de visualisation interactive de données multidimensionnelles** ; nous avons intégré dans notre module un certain nombre de méthodes de visualisation couramment utilisées dans les systèmes de gestion de bâtiments, notamment la présentation sous forme de courbes, de graphes radars, de nuages de points... Nous avons également mis en place des techniques plus sophistiquées telles que la représentation par volume de l'espace pour permettre aux utilisateurs de mieux appréhender les données et réaliser des diagnostics. L'utilisateur interagit avec le système grâce aux techniques traditionnelles (sélection, zoom, panning). Mais nous proposons également un système d'annotation lui permettant de rajouter des commentaires et de garder la trace de ses analyses. Cette information permettra ensuite de mettre à jour le modèle de données RIDER.
- **Un modèle client / serveur** ; ce modèle nous permet tout d'abord de rendre notre système suffisamment générique pour supporter plusieurs clients d'affichage : un mur d'écrans, un ordinateur personnel, une tablette, un smartphone... De plus le serveur filtre l'information de manière à adapter l'affichage en fonction du dispositif du client. Le serveur reçoit les données à des fréquences variables et envoie aux clients une interpolation de celles-ci pour donner une impression de continuité. En outre, la plupart des calculs sont déportés sur le serveur, ce qui permet d'obtenir un rendu fluide sur n'importe quel type de client.

Dans un premier temps et étant donné le contexte de nos travaux, nous avons comparé diverses architectures matérielles afin d'évaluer leur coût en termes énergétiques. Nous avons ensuite traité les deux jeux de données dans notre système afin d'extraire le

maximum d'information de ceux-ci. Le premier jeu de données contient l'information de capteurs placés dans une salle du centre de calcul IBM à Montpellier. La granularité de capteurs dans la salle est forte, ce qui a permis de compléter le modèle RIDER et de proposer, en outre, des modifications structurelles à effectuer afin d'améliorer le rendement énergétique du centre. Avec le second jeu de données, celui provenant d'un bâtiment tertiaire, nous avons été confrontés à une faible granularité de capteurs. L'information collectée permet cependant, d'améliorer le modèle du temps d'occupation des espaces (bureaux, salles de réunion, . . .) pour ce bâtiment, au sein du système RIDER.

8.2 Difficultés rencontrées

Nous avons rencontré plusieurs difficultés lors de ces travaux de recherche, concernant principalement l'accès aux données. En effet, les solutions de gestion de bâtiment ne sont pas encore développées pour être accessibles à distance. Il est donc impératif de se déplacer sur site pour récupérer les données. Le vrai problème réside dans l'impossibilité de recueillir l'ensemble d'informations nécessaires à une évaluation efficace de nos résultats. En effet, pour les zones dont la densité de capteurs est faible, il est difficile d'obtenir des informations complètes, seules les informations relatives à la vie du bâtiment (occupation, horaires de présence) étant disponibles.

De plus, les autres composants développés dans RIDER n'ont pas encore pu donner lieu à des expérimentations. Seuls des résultats théoriques sont disponibles à propos de la modélisation physique et du confort des bâtiments. Un effort de normalisation des données de capteurs est également à envisager. Les fabricants de systèmes de gestion de bâtiments préfèrent actuellement utiliser leur propre méthode de description de capteurs au lieu de s'appuyer sur des solutions telles que SensorML.

Malgré ces difficultés, l'objectif que nous nous étions fixé concernant la visualisation a été atteint. Nous proposons un système ouvert, multiplateforme, facilitant l'analyse des données multidimensionnelles. En effet, notre visualisation a été expérimentée par les différents membres du consortium de RIDER et a été perçue comme étant intuitive et facilitant le diagnostique. Nous avons également montré que l'amélioration du modèle du bâtiment grâce à la visualisation est possible par l'intégration de données appropriées provenant des utilisateurs. Nous avons, en outre, réalisé un comparatif entre diverses architectures matérielles de manière à évaluer la meilleure solution pour l'implémentation de notre composant, en termes de consommation énergétique.

8.3 Limitations

Pour les raisons citées précédemment concernant les autres composants RIDER, nous avons fait le choix de développer un système indépendant mais ouvert. Nous savons cependant que notre module étant un prototype, son passage au stade de produit commercial ainsi que son intégration dans RIDER vont demander quelques ajustements. A l'heure où ce manuscrit est écrit, le développement du système RIDER continue. Le module de communication entre composants est déjà mis en place mais n'est pas entièrement fonctionnel. En effet, aucun standard n'existe à l'heure actuelle pour communiquer avec un BMS. La solution envisagée repose sur un format XML descriptif. D'autre part, le modèle de données n'est pas encore normalisé ; l'utilisation d'un modèle existant serait à mettre en place. Les dernières discussions se tournaient vers le modèle IFC malgré sa complexité. Comme le montre la figure 8.1¹ de nombreux éléments du bâtiment sont modélisés comme par exemple les câbles, la tuyauterie... Ces éléments ne sont pas référencés sur les systèmes de gestion de bâtiments traditionnels.

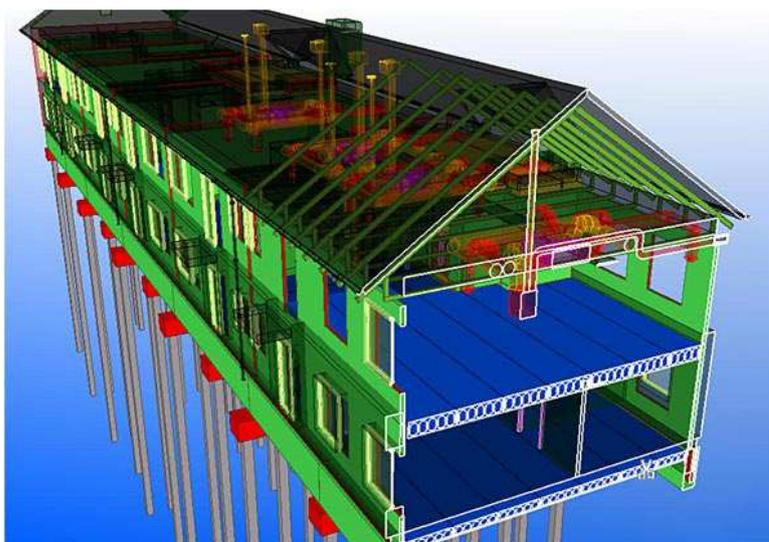


FIGURE 8.1 – Exemple de bâtiment représenté à l'aide d'un modèle IFC

8.4 Perspectives

Dans le cadre d'un projet ambitieux tel que RIDER, un certain nombre de pistes pourraient être explorées pour améliorer l'expérience de l'utilisateur et la qualité des informations présentées. Voici quelques pistes parmi les plus prometteuses :

- **Une solution de réalité virtuelle** ; l'utilisation de lunettes spéciales ou d'une application pour smartphone pourrait permettre d'explorer les données du bâtiment en

1. <http://www.tekla.com>

même temps que le bâtiment lui-même. En effet, grâce à la géolocalisation et à l'utilisation de caméras, il serait possible de se promener physiquement à l'intérieur du bâtiment en utilisant le module de visualisation. Ceci permettrait d'améliorer l'expérience d'utilisation et d'identifier d'éventuels problèmes de localisation de capteurs ou d'isolation du bâtiment.

- **Des méthodes de compression** ; nous nous reposons sur une architecture client / serveur qui, intrinsèquement, utilise une quantité très importante de bande passante pour communiquer les informations entre le serveur et les clients. Des solutions de compression pourraient être étudiées pour réduire le volume de données envoyées par le serveur en fonction du type de technologie réseau entre le client et le serveur.
- **Un modèle physique** ; le modèle physique que nous avons mis en place est rudimentaire et basé sur l'interpolation des données des capteurs. L'utilisation de véritables modèles physiques adaptés à chaque bâtiment permettrait d'obtenir des résultats visuels plus précis.

Pour conclure, cette thèse a permis de confirmer l'intérêt de produire une visualisation interactive et adaptée aux utilisateurs. Des travaux de recherches dans un contexte collaboratif avec des industriels à également permis de donner un cadre théorique à ces travaux tout en gardant une approche pragmatique.

Liste des publications

Chapitre de livre

- **B. Lange, N. Rodriguez, and W. Puech**, How to optimize a building through a visualization software, INTECH, 2012.

Internationales

- **B. Lange, N. Rodriguez, W. Puech, H. Rey, and X. Vasques**, A 3d particle visualization system for temperature management, SPIE Electronic Imaging 2011, Visualization and Data Analysis 7868, 7868- 27, 2011.
- **B. Lange, N. Rodriguez, W. Puech, H. Rey, and X. Vasques**, Visualization assisted by parallel processing, SPIE Electronic Imaging 2011, Parallel Processing for Imaging Applications 7872, 7872-10, 2011.
- **B. Lange, N. Rodriguez, W. Puech, H. Rey, and X. Vasques**, 3d visualization of particle system with extracted data from sensor, LACCEI ,2011.
- **B. Lange, N. Rodriguez**, Lod + : augmenting lod with skeletons, ACM SIGGRAPH 2010 Posters,2010.

Nationales

- **X. Vasques, T. Possompès, H. Rey, M. Le Touzé, N. Auboin, E. Passot, and B. Lange**, Analysis and knowledge discovery from sensors data to improve energy efficiency, CIB, 2011.
- **B. Lange, N. Rodriguez, W. Puech, H. Rey, and X. Vasques**, Visualisation 3D d'un système de particules issu de capteurs de température, CORESA, 2010.

Communications orales

- **B. Lange, N. Rodriguez, and W. Puech**, Visualisation interactive de données hétérogènes pour l'amélioration des dépenses énergétiques du bâtiment, Big Data Mining and Visualization, 2012.
- **B. Lange, N. Rodriguez, and W. Puech**, What happens when software watch behind the screen ?, Journée sur l'analyse vidéo de l'être humain, GDR-ISIS, 2012.
- **B. Lange, N. Rodriguez, and W. Puech**, Client / server level of detail data transmission, CGI, 2011.
- **B. Lange, N. Rodriguez, and W. Puech**, Visualisation interactive de données hétérogènes pour l'amélioration des dépenses énergétiques du bâtiment, De l'acquisition à la compression des objets, GDR-ISIS, 2010.
- **B. Lange, N. Rodriguez, and W. Puech**, Visualization assisted by parallel processing, HPC@LR conference, 2010.

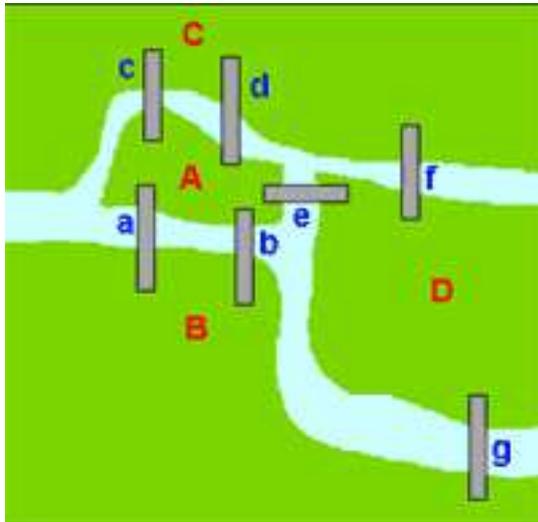


Les graphes

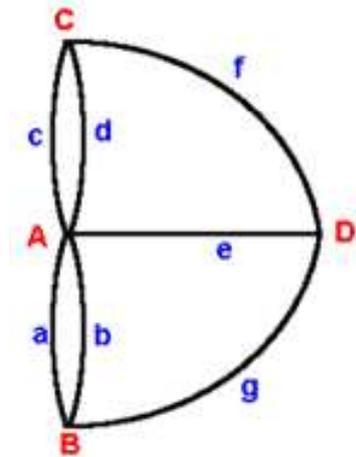
La théorie des graphes permet de modéliser une grande variété de problèmes. Les origines de cette théorie remontent à 1736 lorsqu'Euler résout le problème des ponts de Königsberg (Russie) : est-il possible de parcourir la ville, en revenant au point de départ, en passant une fois et une seule sur chacun des ponts de la ville? Le schéma de la ville est présenté figure 8.2(a), ainsi que la représentation du problème par Euler en figure 8.2(b). L'ensemble des ponts et des îles forment un graphe dans cet exemple. Un graphe est composé d'un ensemble de sommets (îles) reliés par des arêtes (ponts).

Définition d'un graphe. Un graphe noté $G = (V, E)$ est représenté par $V = \{v_1, \dots, v_n\}$ son ensemble de sommets et $E = \{e_1, \dots, e_m\}$ son ensemble d'arêtes.

Définition d'une arête. Pour une arête : $e = \{v_i, v_j\} \in E$, nous pouvons donc dire que v_i et v_j sont les extrémités de l'arête e et que v_i est un voisin de v_j (et vice-versa).



(a) Schéma de la ville.



(b) Graphe proposé par Euler.

FIGURE 8.2 – représentation de la problématique de Königsberg, Russie.

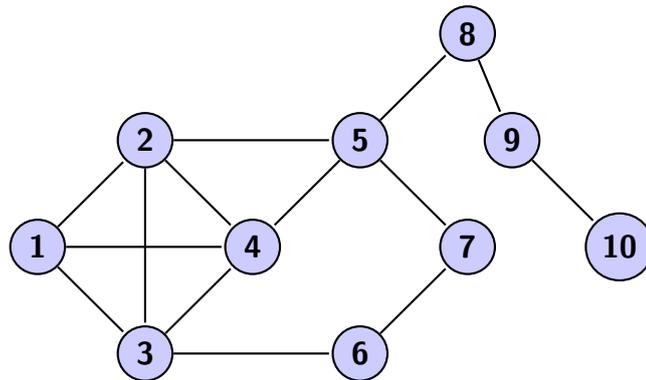


FIGURE 8.3 – Représentation d'un graphe, G, quelconque.

Définition d'un graphe simple. Un graphe simple est un graphe sans boucle et sans arêtes multiples. Une boucle est une arête partant et arrivant au même sommet. Les arêtes multiples sont des arêtes qui relient toutes les deux mêmes sommets.

Définition d'un sous-graphe. $G' = (W, F)$ est un sous-graphe de G si il contient un sous-ensemble de ses sommets ($W \subseteq V$) et un sous-ensemble de ses arêtes ($F \subseteq E$). Dans la figure 8.4, la composante bleue est un sous-graphe de G .

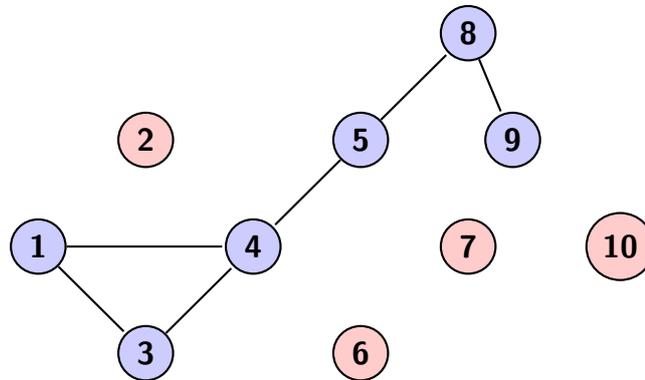


FIGURE 8.4 – Les sommets bleus forment un sous-graphe de G , appelé G' .

Définition d'une chaîne. Une chaîne est une séquence finie d'arêtes consécutives. La longueur d'une chaîne est le nombre d'arêtes qui la constituent.

Définition d'un cycle. Un cycle est une suite d'arêtes consécutives (chaîne) dont les deux sommets extrémités sont identiques.

Définition de la connexité dans un graphe. Un graphe est dit connexe s'il est possible de relier deux sommets quelconques par une chaîne

Définition d'une composante connexe. On appelle composante connexe, un sous graphe qui est connexe. Il peut y avoir plusieurs composantes connexes dans un graphe. Un exemple de deux composantes connexes est donné en figure 8.5.

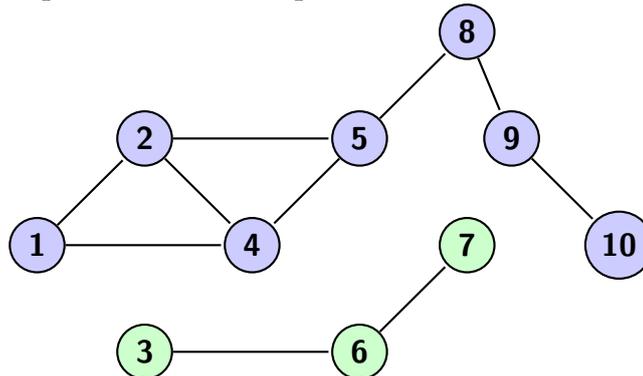


FIGURE 8.5 – G'' (bleu) et G''' (vert) sont deux sous-graphes connexes de G .

Définition d'un arbre. Un arbre est un graphe connexe sans cycle, composé de noeuds et de feuilles. Les noeuds sont des sommets possédant des fils tandis que les feuilles n'en possèdent pas.

Définition d'une étoile dans un graphe. Une étoile est un arbre à n sommets avec 1 noeud et $n - 1$ feuilles.

Définition de voisinage. Dans un graphe, deux sommets sont voisins s'il existe une arête les reliant. Notons k -voisinage de i l'ensemble des sommets qui sont atteignables depuis i en parcourant une chaîne de longueur inférieure ou égale à k .

La figure 8.6(a) illustre une représentation du 1-voisinage du sommet 4 : les sommets en bleu sont ses voisins directs.

Un exemple de k -voisinage est présenté en figure 8.6(b). Les sommets bleu clairs sont les voisins directs de 4, tandis que les sommets bleu foncés sont ses voisins indirects.

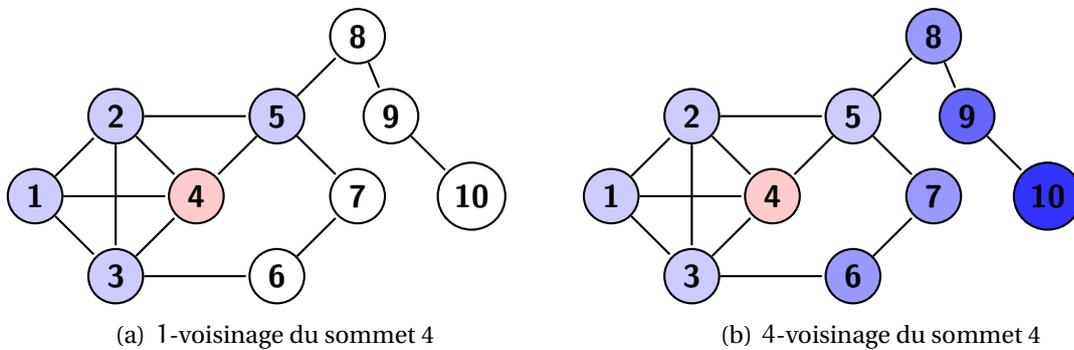


FIGURE 8.6 – Visualisation de deux exemples de voisinage dans un graphe.

Bibliographie

- W. Aigner, S. Miksch, W. Muller, H. Schumann et C. Tominski : Visual methods for analyzing time-oriented data. *Visualization and Computer Graphics, IEEE Transactions on*, 14(1): 47–60, 2008. Cité pages 60 et 61.
- M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin et C.T. Silva : Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, pages 3–15, 2003. Cité page 22.
- E. Anderson : A semigraphical method for the analysis of complex problems. *Proceedings of the National Academy of Sciences of the United States of America*, 43(10):923, 1957. Cité pages 37 et 38.
- M. Ankerst, S. Berchtold et D.A. Keim : Similarity clustering of dimensions for an enhanced visualization of multidimensional data. *In Information Visualization, 1998. Proceedings. IEEE Symposium on*, pages 52–60. IEEE, 1998. Cité page 49.
- F. Aurenhammer : Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23:345–405, September 1991. ISSN 0360-0300. Cité page 16.
- D. Avis et B. K. Bhattacharya : Algorithms for computing D-dimensional voronoi diagrams and their duals. *Advances in Computing Research*, 1:159–180, 1983. Cité pages 13, 15, 81 et 83.
- M. Balzer et O. Deussen : Voronoi treemaps. *Information Visualization, IEEE Symposium on*, 0:7, 2005. ISSN 1522-404x. Cité page 17.
- C. B. Barber, D. P. Dobkin et H. Huhdanpaa : The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22:469–483, December 1996. ISSN 0098-3500. Cité pages 15 et 16.

- A. Baumann, P. Barham, P.E. Dagand, T. Harris, R. Isaacs, S. Peter, T. Roscoe, A. Schüpbach et A. Singhanian : The multikernel : a new os architecture for scalable multicore systems. *In Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, pages 29–44, 2009. Cité page 24.
- J. Bertin : *Sémiologie graphique : les diagrammes, les réseaux, les cartes*. Mouton, Paris, 1967. Cité page 26.
- L. Borgeat, G. Godin, F. Blais, J-A. Beraldin, P. Massicotte et G. Poirier : Visualizing and analyzing large and detailed 3d datasets. 2005. Cité pages 49 et 50.
- M. Bostock et J. Heer : Protovis : A graphical toolkit for visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1121–1128, 2009. Cité page 35.
- C. Buschmann, D. Pfisterer, S. Fischer, S. P. Fekete et A. Kröller : Spyglass : a wireless sensor network visualizer. *SIGBED Rev.*, 2:1–6, January 2005. ISSN 1551-3688. Cité pages 30 et 31.
- M. Cammarano, X. L. Dong, B. Chan, J. Klingner, J. Talbot, A. Halevy et P. Hanrahan : Visualization of heterogeneous data. *In In InfoVis 2007*, pages 1200–1207, 2007. Cité pages 27 et 28.
- K.M. Carter, R. Raich et W.G. Finn : Information-geometric dimensionality reduction. *Signal Processing Magazine, IEEE*, 28(2):89–99, 2011. Cité page 34.
- W.W.Y. Chan : A survey on multivariate data visualization. Rapport technique, HKUST, 2006. Cité page 31.
- Y.H. Chan, C.D. Correa et K.L. Ma : Flow-based scatterplots for sensitivity analysis. *In IEEE Symposium on Proc. VAST'10*, 2010. Cité pages 33 et 34.
- H. Chernoff : The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973. ISSN 0162-1459. Cité page 38.
- J. H. Clark : Hierarchical geometric models for visible surface algorithms. *Commun. ACM*, 19:547–554, October 1976. ISSN 0001-0782. Cité pages 17 et 75.
- J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks et W. Wright : Simplification envelopes. pages 119–128, 1996. Cité page 22.
- P. Cubaud, A. Topol et J. Dupire : Using Game Engines for non 3D Gaming Applications. *In CGAMES'05, Int. Conf. on Computer Games, Angoulême, France*, pages 304–307, January 2005. Cité page 44.

- B. Delaunay : Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1-2, 1934. Cité page 13.
- M. Dörk, S. Carpendale et C. Williamson : Edgemaps : visualizing explicit and implicit relations. *In VDA2011 : Conference on Visualization and Data Analysis*, page 78680G. IS&T/SPIE, 2011. Cité page 55.
- T. Duval, C. Fleury, B. Nouailhas et L. Aguerreche : Collaborative exploration of 3d scientific data. *In Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 303-304. ACM New York, NY, USA, 2008. Cité pages 53 et 54.
- N. Elmqvist, P. Dragicevic et J.D. Fekete : Rolling the dice : Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, pages 1141-1148, 2008. ISSN 1077-2626. Cité pages 26 et 27.
- A. Fabri, G.-J. Giezeman, L. Kettner, S. Schirra et S. Schönherr : The cgal kernel : A basis for geometric computation. 1148:191-202, 1996. ISSN 0302-9743 (Print) 1611-3349 (Online). Cité page 13.
- E. Fanea, S. Carpendale et T. Isenberg : An interactive 3d integration of parallel coordinates and star glyphs. *In Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 149-156. IEEE, 2005. ISBN 078039464X. Cité pages 36 et 37.
- W. Farr, P. Hut, J. Ames et A. Johnson : An experiment in using virtual worlds for scientific visualization of self-gravitating systems. *Journal of Virtual Worlds Research*, 2(3), 2009. ISSN 1941-8477. Cité page 44.
- J. Foley, S. Card, D. Ebert, A. MacEachren et B. Ribarsky : Visual analytics education. *In Visual Analytics Science And Technology, 2006 IEEE Symposium On*, pages 209-211. IEEE, 2006. Cité page 48.
- K.I. Friese, M. Herrlich et F.E. Wolter : Using game engines for visualization in scientific applications. *New Frontiers for Entertainment Computing*, pages 11-22, 2008. Cité pages 44 et 45.
- D. Fritsch et M. Kada : Visualisation using game engines. *Archiwum ISPRS*, 35:B5, 2004. Cité pages 44 et 45.
- R. Fuchs, J. Waser et M. E. Gröller : Visual human+machine learning. *IEEE TVCG*, 15(6): 1327-1334, octobre 2009. ISSN 1077-2626. Cité pages 66 et 67.
- Khronos OpenCL Working Group *et al.* : The opencl specification. *A. Munshi, Ed*, 2008. Cité page 24.

- J.Y. Han : Low-cost multi-touch sensing through frustrated total internal reflection. *In Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118. ACM, 2005. Cité page 52.
- M. C. Hao, R. K. Sharma, D. A. Keim, U. Dayal, C. D. Patel et R. Vennelakanti : Application of visual analytics for thermal state management in large data centres. *Comput. Graph. Forum*, 29(6):1895–1904, 2010. Cité pages 56 et 57.
- T. He, L. Hong, A. Kaufman, A. Varshney et S. Wang : Voxel based object simplification. *In Proc. SIGGRAPH Symposium on Interactive 3D Graphics*, pages 296–303, 1995. Cité pages 22 et 23.
- H. Hoppe : Progressive meshes. *computer graphics. SIGGRAPH'96 Proceedings*, pages 99–108, 1996. Cité page 21.
- M. Ilcik : A framework for global scope interactive visual analysis of large unsteady 3d flow data. *CESCG*, 2008. Cité page 49.
- B. Johnson : Treviz : treemap visualization of hierarchically structured information. *In Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 369–370. ACM, 1992. Cité pages 40 et 41.
- M. Kaltenbrunner : reactivation and tuio : a tangible tabletop toolkit. *In Proceedings of the ACM international Conference on interactive Tabletops and Surfaces*, pages 9–16. ACM, 2009. Cité page 53.
- M. Kaltenbrunner, T. Bovermann, R. Bencina et E. Costanza : Tuio - a protocol for table based tangible user interfaces. *In Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005)*, Vannes, France, 2005. Cité page 53.
- D.A. Keim, G. Andrienko, J.D. Fekete, C. Görg, J. Kohlhammer et G. Melançon : Visual analytics : Definition, process, and challenges. *Information Visualization*, pages 154–175, 2008. Cité page 48.
- D.A. Keim, M. Ankerst et H.P. Kriegel : Recursive pattern : A technique for visualizing very large amounts of data. *In Proceedings of the 6th conference on Visualization'95*, page 279. IEEE Computer Society, 1995. Cité page 41.
- D.A. Keim, M.C. Hao, U. Dayal, H. Janetzko et P. Bak : Generalized scatter plots. *Information Visualization*, 9(4):301–311, 2010. Cité pages 32 et 33.
- D.A. Keim, F. Mansmann, J. Schneidewind et H. Ziegler : Challenges in visual data analysis. *In Information Visualization, 2006. IV 2006. Tenth International Conference on*, pages 9–16. IEEE, 2006. ISBN 0769526020. Cité page 48.

- Daniel A. Keim : Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8:1–8, January 2002. ISSN 1077-2626. Cité page 47.
- L. Kobbelt, S. Campagna et H.P. Seidel : A general framework for mesh decimation. *In Graphics Interface*, pages 43–50. CANADIAN INFORMATION PROCESSING SOCIETY, 1998. Cité page 20.
- R. Kosara, G. N. Sahling et H. Hauser : Linking scientific and information visualization with interactive 3d scatterplots. *In In Proceedings of the 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, pages 133–140, 2004. Cité pages 65 et 66.
- V. Krishnamurthy et M. Levoy : Fitting smooth surfaces to dense polygon meshes. *In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 313–324. ACM, 1996. Cité page 21.
- B. Lange, N. Rodriguez et W. Puech : Client / server level of detail data transmission. *CGI*, 2011a. Cité page 109.
- B. Lange, N. Rodriguez et W. Puech : *How to optimize a building through a visualization software*. INTECH, 2012. Cité page 131.
- B. Lange, N. Rodriguez, W. Puech, H. Rey et X. Vasques : Visualisation 3d d'un système de particules issu de capteurs de température. *Coresa*, 2010. Cité page 85.
- B. Lange, N. Rodriguez, W. Puech, H. Rey et X. Vasques : A 3d particle visualization system for temperature management. *SPIE Electronic Imaging 2011, Visualization and Data Analysis 7868*, 7868-27, 2011b. Cité page 109.
- B. Lange, N. Rodriguez, W. Puech, H. Rey et X. Vasques : 3d visualization of particle system with extracted data from sensor. *LACCEI*, 2011c. Cité page 109.
- B. Lange, N. Rodriguez, W. Puech, H. Rey et X. Vasques : Visualization assisted by parallel processing. *SPIE Electronic Imaging 2011, Parallel Processing for Imaging Applications 7872*, 7872-10, 2011d. Cité page 131.
- J. C. Lee : Hacking the nintendo wii remote. *Pervasive Computing, IEEE*, 7(3):39–45, 2008. Cité pages 51 et 52.
- M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade et D. Fulk : The digital michelangelo project : 3D scanning of large statues. pages 131–144, juillet 2000. Cité page 17.

- A. Lex, M. Streit, E. Kruijff et D. Schmalstieg : Caleydo : Design and evaluation of a visual analysis framework for gene expression data in its biological context. *In Pacific Visualization Symposium (PacificVis), 2010 IEEE*, pages 57–64. IEEE, 2010. Cité pages 62 et 63.
- Z. Liu, B. Lee, S. Kandula et R. Mahajan : Netclinic : Interactive visualization to enhance automated fault diagnosis in enterprise networks. *IEEE VAST*, pages 131–138, October 2010. Cité page 56.
- W. E. Lorensen et H. E. Cline : Marching cubes : A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21:163–169, August 1987. ISSN 0097-8930. Cité page 96.
- D. Luebke : A survey of polygonal simplification algorithms. *IEEE Computer Graphics and Applications*, (21):24–35, 1997. Cité page 18.
- D. P. Luebke : A developer's survey of polygonal simplification algorithms. *IEEE Computer Graphics and Applications*, 21:24–35, 2001. Cité page 18.
- S. Madougou, V. Gouranton et E. Melin : Vers une gestion efficace de données géoscientifiques complexes pour la visualisation sur grappe de pc. *AFRV*, 2006. Cité page 72.
- K Matkovic, W Freiler, D Gracanin et H Hauser : Comvis : a coordinated multiple views system for prototyping new visualization technology. *in Proceedings of the 12th Intern. Conference Information Visualization (IV 2008)*, pages pp.215–220, 2008. Cité page 42.
- G. Migut et M. Worring : Visual exploration of classification models for risk assessment. *In Proceedings of the IEEE Conference on Visual Analytics Science and Technology 2010*, 2010. Cité pages 61 et 62.
- C. Moenning et N.A. Dodgson : Intrinsic point cloud simplification. *Proc. 14th GrahiCon*, 14, 2004. Cité page 23.
- L. Mroz et H. Hauser : Rtvr-a flexible java library for interactive volume rendering. *In Visualization, 2001. VIS'01. Proceedings*, pages 279–564. IEEE, 2001. Cité page 66.
- M. Pauly, M. Gross et L.P. Kobbelt : Efficient simplification of point-sampled surfaces. *In Visualization, 2002. VIS 2002. IEEE*, pages 163–170. IEEE, 2002. Cité page 23.
- R.M. Pickett et G.G. Grinstein : Iconographic displays for visualizing multidimensional data. *In Proc. IEEE Conf. on Systems, Man and Cybernetics, IEEE Press, Piscataway, NJ*, volume 514, page 519, 1988. Cité pages 38 et 39.
- H. Piringer, R. Kosara et H. Hauser : Interactive focus+ context visualization with linked 2d/3d scatterplots. *In Coordinated and Multiple Views in Exploratory Visualization, 2004. Proceedings. Second International Conference on*, pages 49–60. IEEE, 2004. Cité pages 63 et 64.

- J. Ribelles, A. López, O. Belmonte, I. Remolar et M. Chover : Multiresolution modeling of arbitrary polygonal surfaces : a characterization. *Computers & Graphics*, 26(3):449–462, 2002. Cité page 18.
- J.C. Roberts : State of the art : Coordinated & multiple views in exploratory visualization. *In Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV'07*, pages 61–71, 2007. Cité pages 58 et 59.
- C. H. Rycroft : Voro++ : a three-dimensional voronoi cell library in c++. *Chaos* 19, 2009. Lawrence Berkeley National Laboratory. Cité page 16.
- T. Salomon, R. Mikolasek et B. Peuportier : Outil de simulation thermique du bâtiment, comfie. *Journée thématique SFT-IBPSA, La Rochelle*, 2005. Cité page 30.
- F. Schöffel, S. Herkel et J. Dionisio : Interactive three-dimensional visualisation of thermal comfort. 1999. Cité pages 29 et 30.
- A. C. Schreyer et S. Hoque : Interactive 3d visualization of building envelope systems using infrared thermography and sketchup introduction. *InfraMation*, 2009-029, 2009. Cité page 29.
- W. J. Schroeder, J. A. Zarge et W. E. Lorensen : Decimation of triangle meshes. *SIGGRAPH Comput. Graph.*, 26:65–70, July 1992. ISSN 0097-8930. Cité page 20.
- J. Seo et B. Shneiderman : A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4(2):96–113, 2005. Cité pages 59 et 60.
- B. Shneiderman : Inventing discovery tools : Combining information visualization with data mining. *In Klaus Jantke et Ayumi Shinohara, éditeurs : Discovery Science*, volume 2226 de *Lecture Notes in Computer Science*, pages 17–28. Springer Berlin / Heidelberg, 2001. Cité page 47.
- J. M. Snyder et A. H. Barr : Ray tracing complex models containing surface tessellations. *SIGGRAPH Comput. Graph.*, 21:119–128, August 1987. ISSN 0097-8930. Cité page 82.
- H. Song et H.-Y. Feng : A progressive point cloud simplification algorithm with preserved sharp edge data. *The International Journal of Advanced Manufacturing Technology*, 45 (5-6):583–592, November 2009. ISSN 0268-3768 (Print) 1433-3015 (Online). Cité page 23.
- J.J. Thomas et K.A. Cook : *Illuminating the path : The research and development agenda for visual analytics*. IEEE Computer Society, 2005. ISBN 0769523234. Cité pages 47 et 48.
- R. F. Tobler : The rkd-tree : An improved kd-tree for fast n-closest point queries in large point sets. *in Proceedings of Computer Graphics International 2011 (CGI 2011)*, 2011. Cité pages 50 et 51.

- G. Turk : Re-tiling polygonal surfaces. *COMPUTER GRAPHICS-NEW YORK-ASSOCIATION FOR COMPUTING MACHINERY*, 26:55–55, 1992. Cité page 20.
- A. Van Dam : Post-wimp user interfaces. *Communications of the ACM*, 40(2):63–67, 1997. Cité page 68.
- G. Van Den Bergen : Efficient collision detection of complex deformable models using aabb trees. *In J. Graphics Tools*, 1998. Cité page 74.
- T. Van Walsum, F.H. Post, D. Silver et F.J. Post : Feature extraction and iconic visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 2(2):111–119, 1996. Cité pages 39 et 40.
- X. Vasques, T. Possompès, H. Rey, M. Le Touzé, N. Auboin, E. Passot et B. Lange : Analysis and knowledge discovery from sensors data to improve energy efficiency. *CIB 2011*, 2011. Cité page 85.
- G. Wallace, P. Bi, K. Li et O. Anshus : A multicursor x window manager supporting control room collaboration. *Computer Science Report No. TR-0707-04, Princeton University*, 2004. Cité pages 53 et 54.
- R. Wegenkittl, H. Loffelmann et E. Groller : Visualizing the behaviour of higher dimensional dynamical systems. *In Visualization'97., Proceedings*, pages 119–125, 1997. Cité pages 35 et 36.
- J. Wu et L. Kobbelt : Fast mesh decimation by multiple-choice techniques. *In Proceedings of 7th International Fall Workshop on Vision, Modeling, and Visualization*, pages 241–248, 2002. Cité page 20.
- J. Yang, A. Patro, S. Huang, N. Mehta, M.O. Ward et E.A. Rundensteiner : Value and relation display for interactive exploration of high dimensional datasets. *In Proceedings of the IEEE Symposium on Information Visualization*, pages 73–80. Citeseer, 2004. Cité pages 64 et 65.
- J. Yang, W. Peng, M.O. Ward et E.A. Rundensteiner : Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. 2003. Cité pages 57 et 58.

Abstract

Energy efficiency is become major issue. Buildings from any country have been identified as gap of energy, buildings are not enough insulated and this leak represents a major part of energy expenditure. RIDER (Research for IT Driven EnERgy efficiency) has emerged from this statement. This project aims to develop a new kind of IT system to optimize energy consumption of buildings. This system is based on a component paradigm, is composed by a pivot model, a data warehouse with a data mining approach and a visualization tool.

In this manuscript, our focus was on the visualization part of the project. This manuscript is composed in two parts: state of the arts and contributions. Basic notions, a visualization chapter and a visual analytics chapter compose the state of the art. In the contribution part, we present : the data model used in this project, the visualization tool proposed and conclude with two experimentations on real data.

Keywords: *Energy efficiency, Visualization, Visual Analytics, Green Data Center, Delaunay, Voronoï, parallel computing*

Résumé

De nos jours, l'économie d'énergie est devenue un enjeu crucial. Les bâtiments où nous travaillons et qui nous servent d'habitation ont été identifiés comme étant une source importante de perte énergétique. De ce constat a émergé le projet RIDER (Reasearch for IT Driven EnERgy efficiency) lequel a pour objectif de développer un système d'information innovant permettant d'optimiser la consommation énergétique d'un bâtiment ou d'un groupe de bâtiments. Ce système est basé sur des composants logiciels, notamment un composant générale de modélisation du bâtiment, un composant orienté pour la fouille de données et un composant de visualisation. Chacun de ces composants est destiné à améliorer le modèle de données du noyau RIDER.

Cette thèse porte sur le composant visualisation que nous avons conçu, développé et intégré au système RIDER. Plusieurs axes ont été traités concernant la modélisation et la visualisation des données provenant des capteurs du bâtiment, ainsi que l'interaction aussi bien avec l'utilisateur qu'avec le modèle de données RIDER. L'objectif étant de permettre la mise à jour du modèle du noyau grâce aux informations "découvertes" par l'utilisateur grâce aux diverses vues. Les jeux de données disponibles sur les pilotes du système RIDER nous ont permis de tester et évaluer la performance des solutions développées.

Mots clefs : *Economie d'énergie, Visualisation, Analyse Visuelle, Green Data center, Delaunay, Voronoï, calcul parallèles*