

Une nouvelle méthodologie de géovisualisation pour les entrepôts de données de trajectoires

Sandro Bimonte*, Ali Hassan**, Roland Lenain*

*Université Clermont Auvergne, Irstea, UR TSCF, Centre de Clermont-Ferrand
9, av. Blaise Pascal, 63178 Aubière, France
{sandro.bimonte, roland.lenain}@irstea.fr

**Research & Innovation, Umanis
7, Rue Paul Vaillant Couturier, 92300 Levallois-Perret, France
ahassan@umanis.com

Résumé. Les méthodes de la géovisualisation intégrées dans les systèmes OLAP ne concernent que des données spatiales classiques (points, lignes et polygones). Notre objectif dans cet article est de définir une nouvelle méthodologie d'analyse visuelle qui intègre le space-time cube, cartes animées et le tableau croisé dynamique pour considérer les données des entrepôts de données trajectoires (EDT). En plus des requêtes multidimensionnelles classiques, nous prenons en compte d'autres types de requête : (1) spatio-temporelles multidimensionnelles qui nécessitent la dimension temporelle et la mesure spatiale et (2) spatio-temporelles thématiques multidimensionnelles qui nécessitent aussi des mesures numériques. Ces deux types de requêtes peuvent porter sur une (Mono) ou plusieurs trajectoires (Multi). Nous montrons l'implémentation de notre méthodologie en utilisant le géo-navigateur (Google Earth) et l'extension de tableau croisé dynamique (GeOLAPivot Table).

1 Introduction

Avec l'avancée des techniques d'acquisition des positions géographiques (capteurs, objets connectés, etc.) d'énormes données de trajectoires ont été générées. Ces données de trajectoires sont l'une des sources d'information les plus importantes pour plusieurs applications dans différents domaines comme par exemple, la mobilité (comportement de voyage, modèles de mobilité humaine, etc.), l'environnement, le marketing, l'agriculture (gestion de flotte, de tracteurs, agriculture de précision via les capteurs, etc.), etc.

Le stockage des données spatiales classiques (points, lignes et polygones) dans des entrepôts de données spatiales (EDS) et leur analyse via les systèmes OLAP Spatial (SOLAP) ont été étudiés dans nombreux travaux Alsahfi et al. (2019). Le SOLAP étend OLAP en ajoutant les fonctionnalités des systèmes d'informations géographiques (SIG). Il intègre les concepts de dimension spatiale et mesure spatiale dans le modèle multidimensionnel (Malinowski et Zimányi, 2008). Une dimension spatiale présente un ou plusieurs niveaux ayant un attribut spatial (une géométrie). Une mesure spatiale est une géométrie, et/ou le résultat d'opérations spatiales et/ou des valeurs numériques directement calculées, par exemple, la surface d'une région.

Une nouvelle méthodologie de visualisation pour les EDTs

Les outils SOLAP permettent d'explorer les données à l'aide des affichages tabulaires, graphiques et cartographiques. Ce qui permet d'intégrer des méthodes de géovisualisation dans le processus d'analyse. La géovisualisation est définie comme "*une intégration des approches de la visualisation scientifique, de la cartographie, de l'analyse d'image, de la visualisation d'informations, de l'analyse exploratoire de données et du SIG afin de fournir une théorie, des méthodes et des outils pour l'exploration visuelle, l'analyse, la synthèse et la présentation de données géo-spatiales*" (MacEachren et Kraak, 2001).

Malgré l'importance de la géovisualisation pour analyser les données spatiales, peu de travaux s'intéressent à la visualisation des données spatio-multidimensionnelles. Ils se concentrent sur l'affichage cartographique des données spatiales classiques (Bimonte, 2014).

En effet, les données trajectoires sont des données spatiales complexes car elles sont composées par un ensemble de points avec un ordonnancement temporel. La définition de méthodes de géovisualisation pour les entrepôts de données trajectoires (EDT) est aujourd'hui un défi de recherche majeur et peu exploré. Ainsi, cet article propose une nouvelle méthodologie d'analyse visuelle qui intègre les paradigmes de géovisualisation : "space-time cube", cartes animées, géo-navigateur et "tableau croisé dynamique" (i.e. pivot table) pour l'exploration et l'analyse de données de l'EDT (Figure 1).

Le paradigme de visualisation le plus adopté par les clients OLAP est le tableau croisé dynamique. Il s'agit d'un tableau multidimensionnel qui offre une vue de données imbriquées sur plusieurs niveaux. Le space-time cube est "*une représentation géographique où le temps est traité comme une troisième dimension - axe Z*" (Andrienko et al., 2003). Une carte animée est une carte qui peut être visualisées comme une animation (séquence de cartes) basée sur le temps (DiBiase et al., 1992). Comme défini en (Bimonte et al., 2007), nous confions au "space-time cube" et aux cartes animées l'analyse topologique et temporelle des trajectoires, et nous intégrons dans le tableau croisé dynamique des aperçus de ces cartes. En particulier, nous proposons des nouveaux opérateurs MDX qui permettent d'intégrer une dimension dans une cellule de tableau de pivot, et nous proposons une implantation du space-time cube et des cartes animées dans le géo-navigateur Google Earth pour bénéficier des avantages que celui porte à l'analyse spatio-temporelle (Martino et al., 2009).

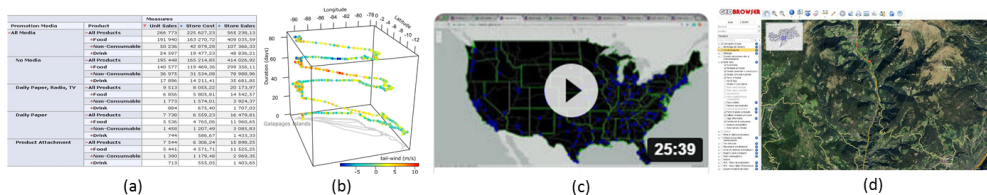


FIG. 1 – Outils de visualisation : a) pivot table, b) space time cube, c) carte animée, d) géo-navigateur.

Nous décrivons notre proposition en utilisant un cas d'étude qui porte sur l'analyse des trajectoires des robots mobiles. L'article est structuré de la façon suivante. La section 2 présente l'état de l'art. Nous montrons nos motivations dans la section 3. Avant de détailler notre proposition pour intégrer le space-time cube, les cartes animées et la table pivot dans la section 5, nous montrons les préliminaires dans la section 4. L'implémentation de notre proposition est décrite dans la section 6. Nous concluons avec la section 7.

2 Etat de l'art

L'analyse visuelle des données géographiques se base sur le paradigme "zoom, filtre et détails sur demande" (Shneiderman, 1996). Ce paradigme se base sur la recherche interactive de données selon plusieurs niveaux de granularité. Les cartes interactives sont donc incontournables. Elles jouent le rôle d'interface avec l'utilisateur où elles réagissent à leurs actions sur l'objet géographique en affichant par exemple d'autres informations comme des images, des documents multimédia, etc. Elles peuvent aussi adapter leur contenu et leur affichage aux actions des utilisateurs.

Par ailleurs, la possibilité de visualiser les données avec plusieurs méthodes différentes est un caractère fondamental de la géovisualisation Gutiérrez et al. (2019). Ce qui permet aux utilisateurs de changer le type de visualisation en remplaçant par exemple une carte choroplethe par une carte 3D. Chaque type de visualisation fournit une vision différente des données étudiées ce qui facilite le processus décisionnel.

L'utilité de l'utilisation d'un géo-navigateur (par exemple, Google Earth, Microsoft Virtual Earth) comme méthode de géovisualisation a été montrée dans plusieurs travaux, par exemple, les travaux de Gonzalez-de Santos et al. (2017). Les travaux de (Compieta et al., 2007; Slingby et al., 2008) montrent l'extensibilité de Google Earth qui permet de représenter les données temporelles et 3D, de collecter des informations sur le Web, et d'intégrer des données multimédia avec les données géographiques. Wood et al. (2007) proposent une approche pour intégrer dans Google Earth des méthodes d'analyse spatiales telles que le filtrage sur un critère temporel, spatial et alphanumérique.

L'intégration des méthodes de geovisualisation dans les systèmes SOLAP a été peu étudiée. Certains travaux s'intéressent à la contextualisation des cartes SOLAP à l'aide de données multimédia (Bédard et al., 2006) ou de documents (Pérez-Martínez et al., 2008). Silva et al. (2012) proposent une méthode pour éviter les problèmes d'occlusion visuelle relatifs aux graphiques affichés sur les cartes. Dans le domaine des EDTs, Leonardi et al. (2014) ont proposé un framework pour visualiser les trajectoires des objets qui sont complètement libres ou limités dans leurs mouvements. Dans cette proposition, les trajectoires ne sont pas représentées par leurs points mais par des informations agrégées (durée, distance, début, fin, vitesse moyenne, etc.). Les opérateurs OLAP proposés permettent de présenter la carte comme une grille où chaque cellule représente visuellement des informations, comme par exemple la direction, la distribution géographique ou même les mesures numériques. Par contre, les auteurs ne proposent pas des méthodes pour visualiser les informations temporelles dans la carte.

Plus proche de notre objectif de travail, Stolte et al. (2003) propose l'intégration des cartes SOLAP classiques dans les cellules de la table pivot, mais les auteurs ne proposent pas des opérateurs MDX pour la construction de ce type de geovisualisation. (Bimonte et al., 2007) étend (Stolte et al., 2003) en proposant l'introduction du space-time cube dans une cellule de la table pivot (appelé GeOLAPivot table) (voir Section 4 pour plus de détails), mais ils ne proposent pas d'implantation et ils utilisent le space-time cube seulement pour la visualisation des mesures spatiales géométriques. Dans cet article nous étendons GeOLAPivot table pour la visualisation des données de trajectoires, avec la proposition de l'utilisation d'un geobrowser et des cartes animées.

3 Motivations

Dans cette section, nous décrivons les motivations de ce travail en utilisant un cas d'étude concernant l'analyse des données trajectoires issues de robots mobiles et autonomes. En effet, aujourd'hui de plus en plus de travaux sont menés pour la mise en place de robots autonomes au service de l'agriculture pour réduire les coûts du travail manuel des agriculteurs, pour améliorer leur qualité de vie, ainsi que pour favoriser l'agro-écologie avec des pratiques agricoles ayant des impacts moins importants sur l'écosystème agricole.

Les robots autonomes se déplacent sur les parcelles pour effectuer des tâches techniques simples comme le labour ou le désherbage ou le désherbage mécanique. Ils sont programmés pour effectuer ces tâches en minimisant les déplacements sur les parcelles, via des trajectoires planifiées préalablement, tout en évitant de potentiels obstacles fixes (comme une ornière ou un poteau) ou mobile (être humain, animaux, ou véhicule) nécessitant un écart à la trajectoire prédéfinie. Pour analyser les expérimentations des algorithmes pour le calcul des trajectoires, une possible solution consiste à mettre en place un EDT qui a comme dimensions : (1) le robot, (2) le temps (avec une hiérarchie : seconde, minute, heure, jour), et (3) la location (avec une hiérarchie : parcelle, exploitation agricole). Et comme mesures : (1) la localisation du robot (mesure spatiale représentée par un point), et (2) la vitesse du robot.

Cet EDT permet d'analyser non seulement le comportement du robot, mais également celui d'autres robots susceptibles de parcourir la parcelle, que ce soit à un autre moment ou simultanément. Plusieurs objectifs peuvent être visés par un tel EDT, comme notamment :

1. L'analyse de certaines métriques relatives aux informations fournies par le robot. Celle-ci peuvent être liées au travail effectué (prise d'image, résistance à l'avancement), ou provenir de l'état du robot (adhérence, vitesse, etc...) . Ceci permet d'identifier des zones difficiles dans une parcelle (Deremetz et al., 2017) ou nécessitant une attention particulière pour les passages suivant ou les autres robots.
2. La détection de zone difficile à traverser incluant en particulier les obstacles. Les modifications locales de trajectoire du robot peuvent en effet être interprétées pour identifier des zones comportant des singularités que sont des obstacles.
3. La vérification du comportement d'une flotte de robots par rapport aux paramètres d'une trajectoire prédéfinie ou des mouvements d'un robot leader (Tourrette et al., 2018). L'analyse de trajectoire permet en effet d'évaluer le bon positionnement relatif et donc l'intégrité d'une formation de robots mobiles.

Dans cette modélisation spatio-multidimensionnelle, une trajectoire est décrite par : un ensemble de membres temporels les plus détaillés, la mesure spatiale et les mesures numériques. Un ensemble de faits compose donc la trajectoire. Cette représentation permet de répondre aux requêtes OLAP classiques qui portent sur les mesures numériques associées à chaque point de la trajectoire. Une approche différente qui consiste à stocker dans un fait plusieurs points avec leurs valeurs temporelles (Leonardi et al., 2014) optimise l'espace de stockage des faits, mais rend impossible ce type de requêtes.

Ces analyses sont faites par les experts en robotique qui n'ont pas des compétences ni en OLAP, ni en SIG. Pour cela, il faut que les outils mis à disposition soient faciles à utiliser et à comprendre. Avec cet EDT, il est donc possible de répondre à différents types de requêtes : multidimensionnelles, spatio-temporelles multidimensionnelles (identifiées avec 'ST') et spatio-temporelles-thématiques multidimensionnelles (identifiées avec 'STT'). Ces requêtes

peuvent porter sur une trajectoire (i.e. Mono - identifiées avec '1') ou sur plusieurs trajectoires (i.e. Multi - identifiées avec 'n'). En particulier :

1. **Multidimensionnelles** : il s'agit des requêtes OLAP classiques, comme par exemple "Pour chaque parcelle et seconde, quelle est la vitesse minimale du robot ?"
2. **Spatio-temporelles multidimensionnelles** : ces sont des requêtes qui nécessitent la dimension temporelle et la mesure spatiale.
 - Un exemple de requête ST1 est : "Pour chaque parcelle et jour, quelle est la trajectoire du robot1 ?".
 - Un exemple de requête STn est "Pour chaque parcelle et jour, est-ce qu'il y a deux robots qui se trouvent au même temps dans la même position (collision) ?".
3. **Spatio-temporelles-thématiques multidimensionnelles** : ces sont des requêtes spatio-temporelles qui nécessitent aussi des mesures numériques.
 - Un exemple de requête STT1 est : "Pour chaque parcelle et jour, quelle est la vitesse de la trajectoire d'un seul robot ?"
 - Un exemple de requête STTn est : "Pour chaque parcelle, et jour, est-ce qu'il y des robots de la même flotte qui n'ont pas la même vitesse au même endroit ?".

Si des mesures numériques peuvent être mises en place pour ces requêtes (par exemple une mesure booléenne comme "collision : oui/non"), l'analyse visuelle de toute la trajectoire et des paramètres associés est nécessaire pour les experts en robotique pour comprendre les causes des ces comportements anormaux des robots. En outre, il est aussi nécessaire de pouvoir situer et donc visualiser les trajectoires sur un modèle terrain en 3D pour comprendre son effet sur le comportement des robots (ex : présence d'une pente). Il s'agit donc de contextualiser la visualisation des données de la dimension spatiale comme décrit en (Martino et al., 2009).

La géovisualisation de ces requêtes avec des cartes SOLAP classiques (c.-à-d. cartes thématiques) est inadaptée car :

1. les cartes SOLAP classiques n'ont pas une représentation explicites du temps (pour les requêtes spatio-temporelles). Par exemple dans la carte de Figure 2, il est impossible de déterminer s'il y a eu une collision (requête STn).



FIG. 2 – Cartes thématiques.

2. pour la visualisation de plusieurs paramètres numériques, les cartes SOLAP classiques nécessitent de nombreuses variables visuelles qui compliquent l'affichage et la lecture de la carte. Par exemple, pour visualiser le robot et la vitesse il faut utiliser deux variables visuelles comme par exemple la couleur et le grain, rendant la carte trop compliquée pour les utilisateurs non experts en SIG.

4 Préliminaires : GeOLAPivot Table

GeOLAPivot table est une extension en trois dimensions du tableau croisé dynamique proposé en (Bimonte et al., 2007). Elle permet de répondre au besoin d'une technique visuelle adaptée à la comparaison de mesures spatiales, en fonction de différents membres du même niveau hiérarchique, et de comprendre les relations spatiales/thématiques entre les mesures. GeOLAPivot table combine le concept du space-time cube (Gatalsky et al., 2004) et du tableau croisé dynamique. Elle exploite la troisième dimension pour comprendre comment un phénomène spatial a évolué en fonction d'un autre facteur (comme le temps) en superposant des données sur une carte.

En effet, les cellules de GeOLAPivot table liées aux données spatiales sont des cubes. Elles représentent dans une seule description visuelle les mesures associées à différents membres d'un même niveau de hiérarchie. Un cube peut être tourné pour obtenir le meilleur point de vue. Ainsi, un utilisateur peut tourner le cube autour de trois axes afin d'analyser les données. La base du cube est associée aux dimensions spatiales (X,Y) et sa troisième dimension à une autre dimension alphanumérique. Les mesures spatiales associées au même fait sont représentées par la même couleur. Un exemple est donné dans la Figure 3 montrant le district du nord de l'Italie touché par certaines maladies infectieuses au cours de la période 2001-2003.

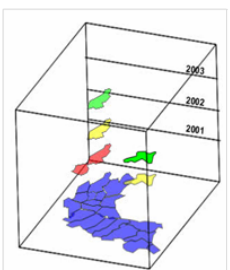
<i>Incidence</i>	<i>Location</i>	<i>Time.All Time-Dept</i>
>6.0	+ Northern Italy	

FIG. 3 – Exemple de GeoLAPivot Table

5 Intégration de space-time cube & cartes animées et pivot table

Nous présentons dans cette section comment utiliser le space-time cube et les cartes animées pour visualiser les résultats des requêtes SOLAP. Ensuite, nous expliquons les opérateurs MDX proposés qui fusionnent les cellules de la table pivot pour construire les trajectoires.

5.1 Space-time cube & cartes animées pour OLAP

Dans cette section, nous décrivons l'implantation du space-time cube & cartes animées avec Google Earth (GE). En particulier nous allons montrer comment visualiser les requêtes définies en Section 3 en utilisant GE avec le space-time cube et les cartes animées.

1. Pour les **requêtes spatio-temporelles**, deux possibilités sont fournies par GE.
 - *Carte animée* avec l'usage de la couleur pour différencier par exemple les robots, et l'animation pour représenter le temps (requête STn) (Figure 5).
 - *Space-time cube* avec l'usage de la couleur pour différencier par exemple le robot et l'axe Z pour le temps (requête ST1) (Figure 4).

Si les cartes animées sont naturellement plus "lisibles", elles ne peuvent pas être intégrées dans les cellules de la table pivot comme un aperçu d'image, tandis que le space-time cube a le pouvoir de représenter le temps de façon statique.
2. Pour les **requêtes spatio-temporelles-thématiques**, deux possibilités sont fournies également par GE :
 - *Carte animée* avec l'usage de la couleur pour différencier par exemple la vitesse des robots, et l'animation pour représenter le temps (requête STTn). Par exemple en Figure 6, il est possible d'analyser si et à quel endroit la flotte de robots ne respecte pas la contrainte d'avoir la même vitesse.
 - *Space-time cube* avec l'usage de la couleur pour différencier par exemple la vitesse et l'axe Z pour le temps (Figure 7) (requête STTn).

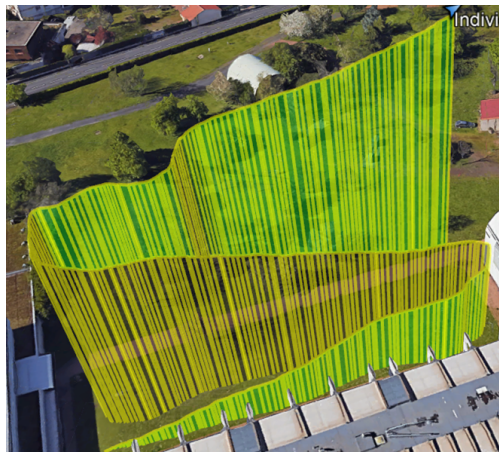


FIG. 4 – Implémentation : ST1 query

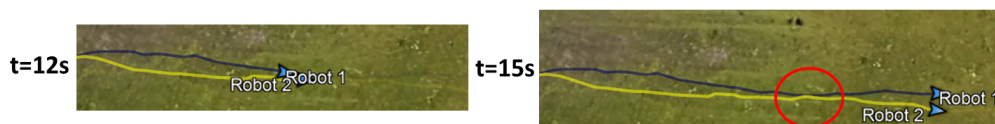


FIG. 5 – Carte d'une requête spatio-temporelle animée (collision, ST2).

Le mapping entre les éléments du modèle multidimensionnel de l'EDT et les géovisualisations du space-time cube et des cartes animées est montré dans le Tableau 1. Nous pouvons remarquer dans ce tableau que la seule différence entre les cartes animées et le space-time cube

Une nouvelle méthodologie de visualisation pour les EDTs



FIG. 6 – Carte d'une requête spatio-temporelle-thématiques animée, STT2.

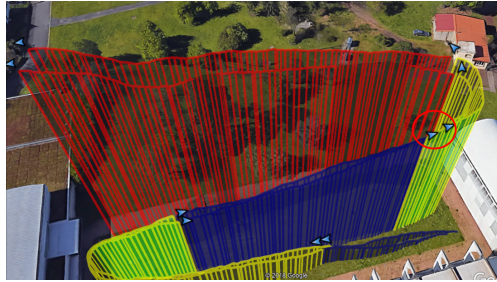


FIG. 7 – Implémentation, STT2.

est la méthode de la présentation du temps. Les cartes animées utilisent l'animation contrairement au space-time cube qui utilise la troisième dimension de l'espace (Z). Dans les requêtes spatio-temporelles, les mesures spatiales sont présentées par les coordonnées géographiques (X :Y) et les membres de dimensions non spatio-temporelles sont distingués par les couleurs.

Concernant les requêtes spatio-temporelles thématiques, les mesures numériques sont présentées par les couleurs. Les données devraient être analysées au niveau le plus détaillé de temps afin de calculer les valeurs des mesures numériques. La mesure spatiale se représente par les coordonnées géographiques (X :Y) pour les requêtes Mono et par un ensemble des coordonnées géographiques pour les requêtes Multi. Comme les couleurs sont utilisées pour présenter les mesures numériques, les dimensions non spatio-temporelles (le type de robot par exemple) ne peuvent plus être présentées.

			Carte animée	Space-time cube	Requête
Spatio-temporelle	Mono	Temps	Animation	Axe Z de l'espace	ST1
		Mesure spatiale	(X :Y)	(X :Y)	
	Multi	Temps	Animation	Axe Z de l'espace	STn
		Mesure spatiale	(X :Y)	(X :Y)	
			Couleur	Couleur	
Spatio-temporelle thématique	Mono	Niveau le plus détaillé de temps	Animation	Axe Z de l'espace	STT1
		Mesure spatiale	(X :Y)	(X :Y)	
		Mesure numérique	Couleur	Couleur	
	Multi	Niveau le plus détaillé de temps	Animation	Axe Z de l'espace	STTn
		Mesure spatiale	Ensemble de (X :Y)	Ensemble de (X :Y)	
		Mesure numérique	Couleur	Couleur	
			-	-	
			-	-	

TAB. 1 – Implémentation.

Il est important de mettre en évidence les raisons de l'usage du géo-navigateur Google Earth pour l'implantation de nos méthodes de géovisualisation. Le premier point concerne

l'utilisation des cartes 3D, en effet, même si les cartes SOLAP 2D sont suffisantes pour certaines applications, les fonctionnalités 3D peuvent énormément enrichir les capacités d'analyse spatio-multidimensionnelle, en prenant en compte les modèles numériques de terrain.

Deuxième point, la plupart des outils SOLAP ne peuvent pas intégrer des données externes dans les entrepôts des données spatiales comme dimensions et mesures ou simplement comme des couches multimédia additionnelles visibles dans le client SOLAP, car ils sont basés sur des technologies propriétaires. Ceci est une importante limitation car une analyse spatiale effective implique la comparaison du phénomène avec les éléments situés autour (ex. routes, industries, villes, etc.). Donc, la capacité d'importer des informations spatiales à partir d'autres sources de données est fondamentale.

5.2 Opérateurs MDX CUBEs

MDX est le langage de-facto standard pour les serveurs et les clients OLAP. MDX est un langage d'interrogation pour les entrepôts de données qui prend en compte les concepts de dimensions, hiérarchies, et mesure. La particularité de MDX est qu'il est aussi un langage d'interrogation orienté visualisation. Une requête MDX permet de définir les dimensions et les mesures que doivent être visualisées mais aussi comment elles sont affichées dans le tableau de pivot. Pour cela MDX définit les mots clés ON COLUMNS, ON ROWS et WHERE qui représentent respectivement les colonnes, les lignes et les axes des tableaux de pivot résultats de la requête.

En MDX donc toutes les dimensions et les mesures sont affichées dans les lignes et les colonnes du tableau de pivot. Cela rend donc incompatible MDX avec la paradigme de géovisualisation GeOLAPivot table car ce dernier prévoit que la dimension temporelle apparait à l'intérieure des cellules de la table pivot. Pour cela nous proposons des nouveaux opérateurs MDX, appelés CUBE, qui permettent de modifier la table pivot conformément à GeOLAPivot table. Puisque une carte animée peut être considérée comme un autre type de géovisualisation des mêmes données que le space-time cube, ces opérateurs sont aussi utilisés pour produire une représentation d'une requête MDX comme une carte animée.

L'opérateur ST1CUBE est utilisé pour la géovisualisation de requêtes de type ST1 (monospatio-temporelles-multidimensionnelles). Il est défini formellement de la façon suivante :

$$\text{cube}(\text{Tmembre}_1, \text{Tmembre}_2, \dots, \text{Tmembre}_n, \text{mesure spatiale})$$

où : $\text{Tmembre}_1, \dots, \text{Tmembre}_n$ sont les membres du niveau le plus détaillé de la dimension temporelle. Et, "mesure spatiale" est la mesure représentant les points de la trajectoire. Un exemple de son usage est montré en Figure 8.

Dans l'exemple de Figure 8, le décideur veut cartographier les données trajectoires pour trois secondes et le robot1. En appliquant l'opérateur ST1CUBE, une liste d'identifiants est créée dans une seule cellule. Les trois cellules sont donc fusionnées en une seule cellule dans la table pivot. Le code "3d :fr" est ajouté, et puisque pour ces données une représentation cartographique n'a jamais été affichée, la valeur de la cellule reste alphanumérique. Ensuite, nous avons étendu JRubik pour que les cellules de la table pivot soient interactives en permettant de déclencher la création des trajectoires dans Google Earth et donc leur visualisation. Dans ce processus d'exploration, si le résultat de l'opérateur CUBE a été visualisé, alors la cellule correspondant affiche l'image de GE capturée comme montré en Figure 9. Il est important de

Une nouvelle méthodologie de visualisation pour les EDTs

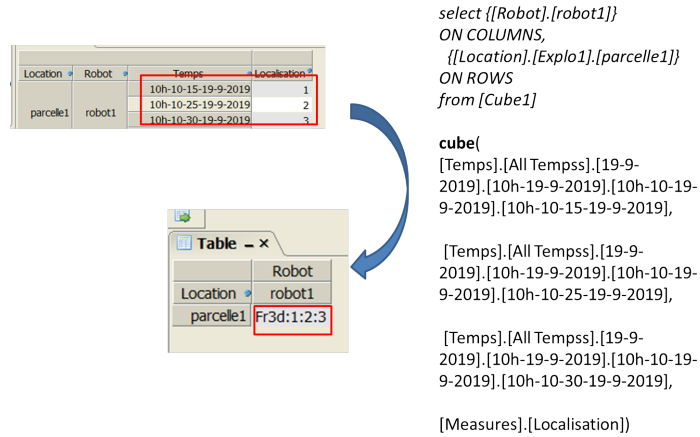


FIG. 8 – Implémentation.

souligner que ce lien dynamique entre la table pivot et GE garanti l'interactivité de l'exploration de données, qui comme décrit en Section 2, est fondamentale et nécessaire pour un correct processus de geovisualisation. En accord avec le paradigme d'analyse visuelle "zoom, filtre et détails sur demande" (voir Section 2), le processus d'analyse est donc le suivant : l'utilisateur explore les données à analyser avec la table pivot dans l'outil OLAP. Ensuite, une fois un jeu de données identifié, il utilise l'opérateur CUBE pour visualiser les données trajectoires dans GE avec le space-time cube et/ou les cartes animées. Ensuite le processus d'exploration recommence et on peut comparer deux space-time cubes simplement en les affichant dans la table pivot.

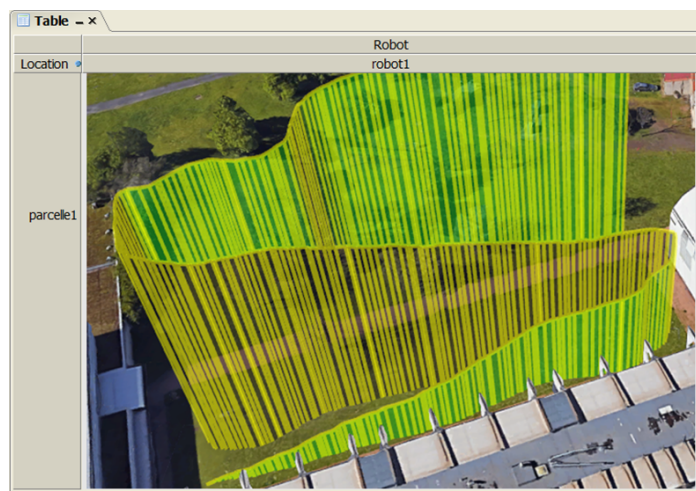


FIG. 9 – Implémentation dans Jrubik, exploration.

L'opérateur STnCUBE (qui correspond aux requêtes multi spatio-temporelles multidimensionnelles) est défini formellement de la façon suivante :

STncube(Tmembre₁, ..., Tmembre_n, Dmembre₁, ..., Dmembre_n, mesure spatiale)

où Dmembre₁, ..., Dmembre_n sont les membres du niveau le plus détaillé d'une dimension "D" non spatiale et non temporelle. Chaque membre de "D" correspond à une trajectoire.

L'opérateur STTICUBE (qui correspond aux requêtes mono spatio-temporelles thématiques multidimensionnelles) est défini formellement de la façon suivante :

STT1cube(Tmembre₁, Tmembre₂, ..., Tmembre_n, mesure spatiale, mesure numérique)

où mesure numérique est une mesure non spatiale associées à la trajectoire.

Enfin, l'opérateur STTnCUBE (qui correspond aux requêtes multi spatio-temporelles thématiques multidimensionnelles) est défini formellement de la façon suivante qui combine les paramètres des opérateurs décrits au dessous :

STTncube(Tmembre₁, ..., Tmembre_n, Dmembre₁, ..., Dmembre_n, mesure spatiale, mesure numerique)

6 Implémentation

Dans cette section nous décrivons l'implémentation de notre proposition. L'architecture de l'outil est montrée en Figure 10. Elle est composée du SGDB spatial Postgis où les données (spatiales et non spatiales) de l'EDT sont stockées. Les données spatiales sont stockées dans une table particulière et seulement les identifiants des coordonnées sont stockés dans la table de faits car les données de type GEOMETRY sont assez volumineuses et peuvent ralentir le calcul des requêtes OLAP. Le serveur OLAP est Mondrian. Mondrian est un serveur OLAP dont la configuration des tables utilisées comme dimensions et faits se fait via un fichier XML comme montrée en Figure 11.

Dans notre approche, chaque ligne de la table de fait représente un point de la trajectoire. Il est donc possible de pouvoir requêter l'EDT à des niveaux de dimensions qui ne sont pas les plus détaillés. Mais les opérateurs CUBEs comme ils ont été définis ne s'appliquent qu'aux membres les plus détaillés de dimensions. Pour prévenir un usage incorrect des opérateurs nous avons introduit des contraintes d'intégrités visuelles (Boulil et al., 2014) qui avertissent l'utilisateur lors de l'interaction avec la table pivot. Ces contraintes sont implémentés en MDX au sein du fichier XML Mondrian (Figure 11). Un exemple est montré en Figure 12.

Le client OLAP utilisé est JRubik. Nous avons étendu les fonctionnalités de JRubik pour permettre l'affichage d'images dans les cellules de la table pivot (Figure 9) et pour l'implémentation des opérateurs CUBEs. En particulier, nous avons implémenté l'opérateur ST1cube via une réécriture de requête. En effet, afin d'obtenir plusieurs mesures dans une cellule, JRubik crée une nouvelle mesure calculée, convertissant les valeurs numériques en chaînes de caractères et ensuite l'opérateur de concaténation est appliqué, comme montré en Figure 8. La conversion en chaîne se fait grâce à la fonction MDX Format([Membre], "") et la concaténation à l'aide de l'opérateur MDX ||. L'implémentation de l'opérateur ST1cube suivi le pattern :

```
with member [Measures].[new] as 'If((([Measures].[mesure spatiale] IS EMPTY), "", (((("Fr3D : " ||
Format([Tmembre1], "")) || " :") || ... || Format(Tmembren), "")))'
select [Dim1], ..., [Dimm] ON COLUMNS, [Dimm+1], ..., [Dimn] ON ROWS
from [Cube] where [Measures].[new]
```

où : $Dim_1 \dots Dim_n$ sont toutes les dimensions non temporelles.

Cette formule s'applique à une mesure calculée qui concatène la mesure spatiale (l'identifiant de la géométrie) avec la valeur temporelle définie par le membre de la dimension temps (T_{membre_i}), pour que Google Earth puisse avoir toutes les informations nécessaires pour la construction et la visualisation de la trajectoire. Un exemple de requête obtenue à partir de la requête de Figure 8 est :

```
with member [Measures].[new] as 'IIf((([Measures].[Localisation] IS EMPTY), "", (((("Fr3d : " ||
Format([Temps].[All Tempss].[19-9-2019].[10h-19-9-2019].[10h-10-19-9-2019].[10h-10-15-19-9-2019],
"")) || " :") ||
Format([Temps].[All Tempss].[19-9-2019].[10h-19-9-2019].[10h-10-19-9-2019].[10h-10-25-19-9-2019],
"")) || " :") ||
Format([Temps].[All Tempss].[19-9-2019].[10h-19-9-2019].[10h-10-19-9-2019].[10h-10-30-19-9-2019], "")))'
select [Robot].[All Robots].[robot1] ON COLUMNS, [Location].[explo1].[parcelle1] ON ROWS
from [Cube1] where [Measures].[new]
```

Un fichier KML (Keyhole Markup Language) est créé à partir de cellules de GeOLAPivot table afin de visualiser les trajectoires dans GE (Figure 13). KML est un langage qui étend le langage XML. Il est destiné à la gestion de l'affichage de données géospaciales dans GE.

À partir de l'identifiant de la géométrie, les coordonnées géographiques (X :Y) sont extraites de la base Postgis. Et, à partir de la valeur temporelle, la valeur de la troisième dimension (Z) est calculée où $Z=0$ pour le premier point de la trajectoire. Ensuite, cette valeur augmente avec le temps. Les trois valeurs (X,Y,Z) sont stockées dans une balise <gx :coord> à laquelle la valeur temporelle est associée dans une balise <when>. Une trajectoire est présentée dans la balise <gx :Track>. Elle est associée dans la balise <Placemark> à un style de visualisation (cf. Figure 13). Pour une requête STT1 (ou STTn) où une trajectoire peut être divisée en plusieurs morceaux ayant des couleurs différentes, chaque sous-trajectoire est présentée par un <Placemark>. Le dernier point de la première sous-trajectoire est le premier point de la sous-trajectoire suivante.

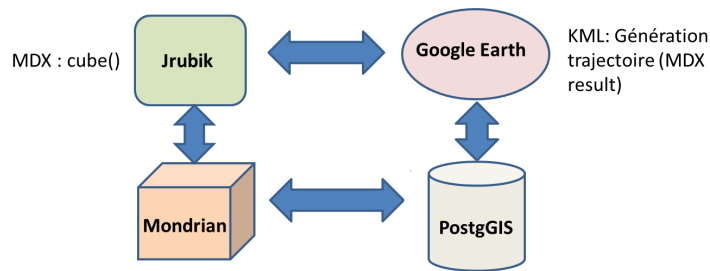


FIG. 10 – Architecture.

7 Conclusion

Dans cet article, nous avons proposé une méthodologie de géovisualisation pour les entrepôts de données de trajectoires. Cette méthodologie propose l'utilisation de plusieurs types de visualisation de données de trajectoires : space-time cube, cartes animées et le tableau croisé dynamique. Les cartes animées peuvent représenter le temps par l'animation et le space-time

```

- <Schema name="EDT">
- <Cube name="Cube1">
  <Table name="fait"/>
  + <Dimension name="Location" foreignKey="id_location">
  + <Dimension name="Temps" foreignKey="id_temps">
  + <Dimension name="Robot" foreignKey="id_robot">
  <Measure name="Localisation" column="localisation" visible="false" aggregator="min"/>
  <Measure name="Vitesse" column="vitesse" aggregator="avg" formatString="##.##"/>
  - <CalculatedMember name="trajf" visible="true" dimension="Measures">
    <Formula> Iif ([Measures].[Localisation] IS EMPTY, "", (CAST([Measures].[Localisation] AS String) || "*" ||
    [Temps].CurrentMember.Name)) </Formula>
  </CalculatedMember>
  - <CalculatedMember name="traj" visible="true" dimension="Measures">
    <Formula> Iif([Temps].CurrentMember.Level.Name="Second" and
    [Robot].CurrentMember.Level.Name="Robot" and [Location].CurrentMember.Level.Name="Parcelle"
    and [Measures].[Localisation] > 0, CAST([Measures].[trajf] AS String),"Operateur cube non
    applicable") </Formula>
  </CalculatedMember>
</Cube>
</Schema>

```

FIG. 11 – Mondrian

Location	Temps	Robot	traj
parcelle1	10h-10-15-19-9-2019	+All Robots	Operateur cube non applicable
	10h-10-25-19-9-2019	+All Robots	Operateur cube non applicable
	10h-10-30-19-9-2019	+All Robots	Operateur cube non applicable

FIG. 12 – Contraintes.

The image shows a 3D visualization of a trajectory in Google Earth, with a red mesh representing the trajectory and yellow lines indicating specific points. A red box labeled "Visualisation dans Google Earth" points to the 3D view. A red box labeled "Fichier KML" points to a code editor showing KML XML code. Arrows labeled "Points de trajectoires" point from the 3D view to the KML code.

FIG. 13 – KML et Google Earth.

cube représente le temps en utilisant la troisième dimension (Z). Pour les requêtes qui nécessitent la dimension temporelle et la mesure spatiale (spatio-temporelles), nous avons proposé d'utiliser les couleurs pour distinguer les trajectoires différentes. En ce qui concerne les requêtes qui nécessitent des mesures numériques (spatio-temporelles-thématiques), nous avons utilisé les couleurs pour représenter les valeurs des mesures.

Afin d'implémenter notre proposition, nous avons utilisé une extension du tableau croisé dynamique (GeOLAPivot Table) qui permet de visualiser les cartes dans les cellules de la table pivot. Un ensemble d'opérateurs MDX : ST1CUBE, STnCUBE, STT1CUBE et STTnCUBE ont été proposés pour fusionner plusieurs cellules (points d'une trajectoire) de la table pivot en une seule. Autrement dit, ces opérateurs reconstruisent les trajectoires. Ensuite, ces trajectoires

sont transformées en un format KML afin de les présenter en Google Earth.

Cette formalisation a été mise en oeuvre pour l'analyse de comportements robotiques en agriculture. Ceci est en effet un point essentiels pour le développement d'engins autonomes sur des parcelles agricoles, puisqu'il permet aux utilisateurs de vérifier le bon fonctionnement des robots et d'anticiper d'éventuelles difficultés requérant une intervention humaine et/ou une modification du réglages des lois de commandes.

Nos travaux futurs sont : la validation de notre méthode avec un jeu de données réel, l'implémentation des autres opérateurs MDX CUBE et l'étude des différentes méthodologies de stockage des données trajectoires.

Références

- Alsahfi, T., M. Almotairi, et R. Elmasri (2019). A survey on trajectory data warehouse. *Spatial Information Research*.
- Andrienko, N., G. Andrienko, et P. Gatalisky (2003). Visual data exploration using space-time cube. In *Proceedings of the 21st International Cartographic Conference*, pp. 1981–1983.
- Bimonte, S. (2014). A generic geovisualization model for spatial OLAP and its implementation in a standards-based architecture. *Ingénierie des Systèmes d'Information* 19(5), 97–118.
- Bimonte, S., A. Tchounikine, S. D. Martino, et F. Ferrucci (2007). Supporting geographical measures through a new visualization metaphor in spatial OLAP. In *Proceedings of the 9th International Conference on Enterprise Information Systems.*, pp. 19–26.
- Boulil, K., S. Bimonte, et F. Pinet (2014). Spatial OLAP integrity constraints : From uml-based specification to automatic implementation : Application to energetic data in agriculture. *Journal of Decision Systems* 23(4), 460–480.
- Bédard, Y., M.-J. Proulx, S. Rivest, et T. Badard (2006). *Merging Hypermedia GIS with Spatial On-Line Analytical Processing : Towards Hypermedia SOLAP*, pp. 167–187.
- Compieta, P., S. D. Martino, M. Bertolotto, F. Ferrucci, et T. Kechadi (2007). Exploratory spatio-temporal data mining and visualization. *Journal of Visual Languages & Computing* 18(3), 255 – 279.
- Deremetz, M., R. Lenain, B. Thuilot, et V. Rousseau (2017). Adaptive trajectory control of off-road mobile robots : A multi-model observer approach. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 4407–4413. IEEE.
- DiBiase, D., A. M. MacEachren, J. B. Krygier, et C. Reeves (1992). Animation and the role of map design in scientific visualization. *Cartography and geographic information systems* 19(4), 201–214.
- Gatalisky, P., N. Andrienko, et G. Andrienko (2004). Interactive analysis of event data using space-time cube. In *Proceedings. Eighth International Conference on Information Visualization, 2004. IV 2004.*, pp. 145–152.
- Gonzalez-de Santos, P., A. Ribeiro, C. Fernandez-Quintanilla, F. Lopez-Granados, M. Brandstoecker, S. Tomic, S. Pedrazzi, A. Peruzzi, G. Pajares, G. Kaplanis, M. Perez-Ruiz, C. Valero, J. del Cerro, M. Vieri, G. Rabatel, et B. Debilde (2017). Fleets of robots for environmentally-safe pest control in agriculture. *Precision Agriculture* 18(4), 574–614.

- Gutiérrez, F., N. N. Htun, F. Schlenz, A. Kasimati, et K. Verbert (2019). A review of visualisations in agricultural decision support systems : An hci perspective. *Computers and Electronics in Agriculture* 163, 104844.
- Leonardi, L., S. Orlando, A. Raffaetà, A. Roncato, C. Silvestri, G. Andrienko, et N. Andrienko (2014). A general framework for trajectory data warehousing and visual OLAP. *GeoInformatica* 18(2), 273–312.
- MacEachren, A. M. et M.-J. Kraak (2001). Research challenges in geovisualization. *Cartography and Geographic Information Science* 28(1), 3–12.
- Malinowski, E. et E. Zimányi (2008). *Advanced Data Warehouse Design - From Conventional to Spatial and Temporal Applications*. Data-Centric Systems and Applications. Springer.
- Martino, S. D., S. Bimonte, M. Bertolotto, et F. Ferrucci (2009). Integrating google earth within OLAP tools for multidimensional exploration and analysis of spatial data. In *Proceedings of 11th International Conference on Enterprise Information Systems.*, pp. 940–951.
- Pérez-Martínez, J. M., R. B. Llavori, M. J. A. Cabo, et T. B. Pedersen (2008). Contextualizing data warehouses with documents. *Decision Support Systems* 45(1), 77–94.
- Shneiderman, B. (1996). The eyes have it : A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages, Boulder, Colorado, USA, September 3-6, 1996*, pp. 336–343. IEEE Computer Society.
- Silva, R. A., J. Moura-Pires, et M. Y. Santos (2012). Spatial clustering in SOLAP systems to enhance map visualization. *IJDWM* 8(2), 23–43.
- Slingsby, A., J. Dykes, J. Wood, M. Foote, et M. Blom (2008). The visual exploration of insurance data in google earth.
- Stolte, C., D. Tang, et P. Hanrahan (2003). Multiscale visualization using data cubes. *IEEE Trans. Vis. Comput. Graph.* 9(2), 176–187.
- Tourrette, T., M. Deremetz, O. Naud, R. Lenain, J. Laneurit, et V. De Rudnicky (2018). Close Coordination of Mobile Robots Using Radio Beacons : A New Concept Aimed at Smart Spraying in Agriculture. *International Conference on Intelligent Robots and Systems*.
- Wood, J., J. Dykes, A. Slingsby, et K. Clarke (2007). Interactive visual exploration of a large spatio-temporal dataset : Reflections on a geovisualization mashup. *IEEE Transactions on Visualization and Computer Graphics* 13(6), 1176–1183.

Summary

The geovisualization methods integrated in OLAP systems concern only classical spatial data (points, lines and polygons). Our goal is to define a new visual analysis methodology that integrates the space-time cube, animated maps, and the pivot table to consider the data from Trajectory Data Warehouse (TDW). In addition to traditional multidimensional queries, we consider other types of queries: (1) spatio-temporal multidimensional queries that require the temporal dimension and a spatial measure; and (2) spatio-temporal thematic multidimensional queries that require also numerical measures. These two types of queries can relate to one (Mono) or several trajectories (Multi). We show the implementation of our methodology using the geo-browser (Google Earth) and the extension of pivot table (GeOLAPivot Table).

