
Base de données de capteurs à localisation mobile – Modèle et langage

Iulian Sandu Popa*—**Ahmed Kharrat *** — **Karine Zeitouni*** —
Guillaume Saint Pierre **

** Laboratoire PRISM*

45, avenue des Etats-Unis - 78035 Versailles

{Iulian.Sandu-Popa,Ahmed.Kharrat,Karine.Zeitouni}@prism.uvsq.fr

*** LIVIC, Laboratoire sur les Interactions Véhicules-Infrastructure-Conducteurs*

Bâtiment 824 -14, route de la Minière - 78000 Versailles

Guillaume.Saintpierre@lpc.fr

RESUME. De nos jours, les bases d'objets mobiles constituent un important thème de recherche. Il existe de nombreux travaux sur la modélisation, l'interrogation et l'indexation d'objets mobiles libres ou sur un réseau. Cependant, un objet mobile - tel qu'un véhicule - peut reporter des mesures, relatives à son état ou à son environnement, relevées tout au long de son déplacement par des capteurs embarqués. La gestion de telles données est d'un intérêt primordial pour certaines applications telles que l'analyse du comportement des conducteurs ou la reconstruction des circonstances d'un accident. Mais, cette gestion n'est pas supportée par les approches existantes. Dans cet article, nous proposons un modèle de données et un système adaptés à la gestion de données de capteurs à localisation mobile. Nous donnons ensuite les résultats préliminaires dans un contexte applicatif réel.

ABSTRACT. Nowadays, moving objects databases are an important research topic. A lot of work dealt with modelling, querying and indexing objects that move freely or in networks. However, a moving object - such as a vehicle - could report some measures, related to its state or to its environment, sensed throughout his movement. Managing such data is of major interest for some applications such as analyzing driving behaviour or reconstructing the circumstances of an accident. But, this is not covered by the existing approaches. In this paper, we propose a new data model and a management system that handle mobile location sensor data. We then give preliminary results in a real application context.

MOTS CLE : Bases de données spatio-temporelles, Modélisation, Objets Mobiles, Langage de requêtes, Flux de données de capteurs.

KEYWORDS: Spatiotemporal databases, Modelling, Moving Objects, Query Language, Sensor Data Flows.

1. Introduction

Dans le domaine de la sécurité routière, l'observation du comportement de conduite naturelle (sur route normale pour des trajets habituels) tend à remplacer les seuls tests sur simulateurs ou ceux limités à des circuits dédiés. Connues sous le terme "*naturalistic driving*", ces études passent par une collecte de données sur une grande échelle et sur une période de temps significative (Burns 2005) (Galit 2007). Cela devient possible à moindre coût grâce aux équipements des véhicules modernes. En effet, les multiples capteurs installés sur les véhicules sont accessibles via une interface (Bus CAN) sur laquelle il est possible de brancher un boîtier enregistreur embarqué dans le véhicule. Le bus CAN donne accès à de nombreuses informations dont la vitesse, l'accélération, l'angle au volant, l'action sur les pédales de frein, etc. Le dispositif d'enregistrement peut également recevoir d'autres sources, comme celles d'un capteur GPS ou de radars (donnant la distance aux véhicules avoisinants). On obtient ainsi une base de données détaillée sur la conduite naturelle sur route. Cette base peut fournir des informations précieuses sur l'utilisation et l'utilité des assistants à la conduite (ABS, EPS, etc.) et permet de mettre en évidence des situations de presque-accident (*near-crash*). En outre, selon le principe de boîtes noires dans les avions, elle permet de disposer des informations précédant un accident. Cependant, les études rapportées dans la littérature ont limité la volumétrie des données et les possibilités de leur exploitation. Ainsi, dans la campagne *naturalistic driving* effectuée sur 100 voitures, entreprise par l'administration de sécurité routière américaine (NHTSA), le stockage est limité aux situations de presque-accident. Comme souligné dans leur rapport (NHTSA, 2006), une base de données à grande échelle serait très utile aux chercheurs et ingénieurs pour les études de comportement de conducteurs et contribuera à l'amélioration de l'équipement des véhicules et de l'aménagement routier. Cependant, le défi d'une étude à grande échelle est la gestion et l'analyse d'une grande masse de données à caractère spatio-temporel. Un système de gestion de telles bases de données supportant ce type de données et des requêtes efficaces serait nécessaire. Nous visons à étudier et à développer un tel système de gestion de bases de données. Cette problématique rejoint en partie celle des bases d'objets mobiles. Cependant, des données additionnelles sont associées à l'objet mobile – ici le véhicule – décrivant l'état de l'objet ou des mesures (vitesse, accélération, etc.) relevées tout au long de son déplacement. Ces mesures sont variables dans l'espace et dans le temps. Pour le type d'applications que nous adressons, les mesures sont plus importantes que la simple localisation spatio-temporelle. Or, tous les travaux sur les bases d'objets mobiles ne traitent que de la localisation mobile et ne peuvent se généraliser aux mesures variant le long d'une trajectoire spatio-temporelle. Par ailleurs, bien que ces valeurs correspondent initialement aux flux de données temporels, leur variation dépend davantage de leur localisation sur le réseau que du temps : par exemple, la variation de la vitesse est généralement contrainte par la géométrie de la route et par la vitesse légale. A l'inverse, l'analyse temporelle des données relatives à des trajectoires différentes n'est pas pertinente car, d'une part elles sont asynchrones et

d'autre part, cette comparaison n'a de sens que si ces trajectoires se recouvrent dans l'espace. Par conséquent, il faut capturer la variabilité spatiale de ces mesures et permettre sa manipulation à travers le modèle de données et les requêtes. À notre connaissance, il n'existe aucune proposition de ce type dans l'état de l'art. Néanmoins, parmi les travaux sur les bases d'objets mobiles, celui proposé dans (Güting *et al.*, 2000) constitue une base solide pour la modélisation d'objets mobiles. L'idée de représenter de façon continue la variation temporelle de la localisation et des valeurs scalaires permet une bonne abstraction des objets mobiles. Nous étendons cette approche – comme nous le définirons plus loin – pour capturer la variation spatiale continue des valeurs scalaires. Ensuite, nous étendons l'algèbre proposée avec un nouveau type et des opérateurs associés. Ce type capture la variation de n'importe quelle mesure dans l'espace, dont les données de capteurs mobiles sont un cas particulier.

Nous avons présenté à la conférence SAGEO en 2008 une version préliminaire de ce travail (Sandu Popa *et al.*, 2008a). Cet article apporte des précisions dans l'état de l'art et dans le modèle avec une extension à de nouvelles fonctions d'agrégation retournant des valeurs complexes. Globalement, notre travail apporte les contributions suivantes. Premièrement, nous présentons un nouveau concept de « mesure variable dans l'espace » et montrons son utilité dans le contexte d'une étude en « *naturalistic driving* ». Deuxièmement, nous établissons un modèle de données comme extension de celui proposé par Güting *et al.* (Güting *et al.*, 2006), et implémentons une grande partie de cette extension. Finalement, nous montrons la faisabilité du système par une requête sur le dépassement de vitesse. Au-delà de l'application précitée, le modèle proposé est intéressant pour d'autres applications générant et/ou exploitant des flux de données géolocalisés. C'est le cas du transport ferroviaire (TGV et AGV comprennent de nombreux capteurs embarqués), aérien ou maritime. Les mesures peuvent être observées ou calculées et peuvent être liées à une trajectoire d'un objet ou à un endroit donné. On peut ainsi raisonner sur la vitesse d'un objet mobile, sur la vitesse légale ou sur le dénivelé d'une route ou encore sur l'adhérence à chaque endroit de la route en fonction de la météo. Ainsi, on peut modéliser les données fines sur la mobilité (où, quand et à quelle vitesse) des véhicules (équipés GPS), des biens (étiquetés par de RFID), ou des personnes (GSM/GPRS) et répondre aux besoins d'applications de gestion de flottes ou de trafic routier, de logistique ou de dimensionnement de réseaux mobiles.

Le reste de cet article est organisé comme suit : nous résumons l'état de l'art dans la section 2. La section 3 décrit le modèle proposé. Elle présente le nouveau type et les opérations correspondantes, et montre son utilisation pour exprimer des requêtes types. La section 4 décrit notre implémentation. Finalement, la section 5 conclut et trace des perspectives de ce travail.

2. Aperçu de l'état de l'art

La gestion de bases d'objets mobiles a reçu une attention particulière durant ces dernières années en raison des avancées et de la banalisation des technologies mobiles et de géo-localisation, telles que les téléphones cellulaires ou le GPS. De nombreux travaux portent sur la modélisation et le langage. Citons les travaux de la thèse de Yeh (Yeh 1995), ceux en rapport avec le projet Chorochronos (Grumbach *et al.*, 2001, Güting *et al.*, 2000), (Koubarakis *et al.*, 2003) et les travaux de l'équipe de Wolfson (Sistla *et al.*, 1997), (Vazirgiannis et Wolfson 2001), (Wolfson 2002). L'ouvrage de Güting fait une synthèse des avancées dans ce domaine (Güting et Schneider 2005). (Pelekis *et al.*, 2005) résume les modèles spatio-temporels dont ceux des objets mobiles.

Les applications visées impactent le modèle et le langage dans ces propositions. On distingue ainsi deux types d'applications. Les applications de type LBS (*Location Based Services*) requièrent des requêtes continues ou prédictives, basées sur les positions courantes des objets mobiles. Les pionniers sont (Sistla *et al.*, 1997) dont le modèle MOST (*Moving Objects Spatio-Temporal*) décrit des bases de données avec des attributs dynamiques variant continuellement au cours du temps. Ils proposent également un langage nommé FTL (*Future Temporal Logic*) basé sur la logique temporelle pour formuler des requêtes prédictives. Le second type d'applications concerne l'analyse des trajectoires spatio-temporelles complètes, dont les requêtes combinant les intervalles temporels et spatiaux. Les travaux de Güting (Güting *et al.*, 2000) font référence jusqu'aujourd'hui. Différentes implémentations existent, comme dans *SECONDO* (Forlizzi *et al.*, 2000) et dans STAU (Pelekis *et al.*, 2002), puis dans *HERMES-MDC* (Pelekis *et al.*, 2006), (Pelekis *et al.*, 2008). STAU est la première implémentation à se baser sur l'extensibilité objet-relationnelle en proposant une cartouche de données spatio-temporelle pour Oracle. Néanmoins, ces travaux ne prennent pas en considération les mouvements contraints des objets mobiles, tels que les trajectoires des véhicules dans un réseau routier ou des trains sur un réseau ferrés. Cet aspect est essentiel pour un grand nombre d'applications dont celles considérées ici.

En effet, connaissant un réseau, une trajectoire contrainte par ce réseau peut être représentée par ses positions relatives sur les lignes (ex. les routes) de ce réseau. On parle de modèle contraint. Là encore, la proposition la plus complète est celle dans (Güting *et al.*, 2006). Bien que les modèles de trajectoires libres puissent s'appliquer aux trajectoires contraintes, cela n'est pas judicieux pour plusieurs raisons. La première est qu'ils ne capturent pas la relation entre la trajectoire et le milieu contraint où elle évolue alors que cette information est essentielle pour l'analyse. La deuxième est qu'ils sont limités dans la représentation de la trajectoire, estimée par une interpolation linéaire entre les points relevés, alors qu'elle suit la géométrie du réseau dans le modèle contraint. De plus, le modèle contraint permet la réduction de dimensionnalité en transformant l'espace du réseau en 1D par juxtaposition de toutes ses lignes (Pfooser et Jensen 2003). Cela permet de meilleures performances de

stockage et de requêtes qu'avec le modèle de trajectoires libres. Enfin, le modèle contraint permet de générer aisément une description symbolique de trajectoires, comme une séquence de lignes traversées et d'intervalles temporels, moins détaillée mais plus intelligible et plus compacte (Savary *et al.*, 2004).

2.1. Modèle d'objets mobiles

Dans cet article, nous nous intéressons aux bases d'objets mobiles historiques contraints par le réseau. La proposition la plus complète pour modéliser les objets mobiles historiques est, de notre point de vue, celle issue des travaux de Güting (Güting *et al.*, 2006). En effet, elle couvre la modélisation abstraite, le langage et l'implémentation. De plus, elle modélise explicitement les objets mobiles contraints et la position relative au réseau.

Comme nous le verrons plus loin, notre proposition se base sur ce modèle et l'étend avec les données spécifiques aux capteurs mobiles. Par conséquent, nous résumons ce modèle dans le reste de cette section.

\rightarrow <i>BASE</i>	<i>int, real, string, bool</i>
\rightarrow <i>SPATIAL</i>	<i>point, points, line, region</i>
\rightarrow <i>GRAPH</i>	<i>gpoint, gline</i>
\rightarrow <i>TIME</i>	<i>instant</i>
$BASE \cup SPATIAL \cup GRAPH \rightarrow$ <i>TEMPORAL</i>	<i>moving, intime</i>
$BASE \cup TIME \rightarrow$ <i>RANGE</i>	<i>range</i>

Tableau 1: L'ensemble des types définis dans (Güting *et al.*, 2000) et (Güting *et al.*, 2006)

Güting *et al.* proposent une algèbre définie par un ensemble de types spécifiques (Tableau 1) et une collection d'opérations (Tableau 2 et 3) sur ces types (Güting *et al.*, 2000), (Güting *et al.*, 2006). Ces types sont : des types scalaires (*BASE*), des types spatiaux 2D (*SPATIAL*), des types spatiaux liés à un réseau (*GRAPH*), des types scalaires ou spatiaux variants dans le temps (*TEMPORAL*). Des exemples de types sont : *real*, *point* (position absolue en 2D), *gpoint* (position relative au réseau), *gline* (ligne relative au réseau), *moving (point)* (position 2D variant dans le temps), et *moving (gpoint)* (position relative au réseau variant dans le temps). Tous les types de base ont l'interprétation habituelle. Par exemple, si on note avec A_α le domaine de définition pour le type α , alors pour le type *real* ce domaine est : $A_{real} = R \cup \{\perp\}$, où $\{\perp\}$ représente la valeur nulle (ou non définie). Le temps est isomorphe aux

nombres réels. Les données de type *range* sont des intervalles disjoints et sont utilisés pour faire des projections ou des sélections sur les types *moving*.

Les types spatiaux décrivent des entités dans l'espace Euclidien, tandis que, pour les types GRAPH, l'espace est représenté par un réseau. Les types 2D correspondent globalement aux définitions standards (ISO 2003).

Le type GRAPH dépend du réseau sous-jacent. Fondamentalement, le modèle proposé définit un réseau comme un ensemble de routes et de jonctions entre ces routes. Un emplacement dans le réseau est une position relative sur une route. Il est décrit par l'identifiant de la route, un nombre réel donnant la position relative et un côté. Ceci est directement lié au concept de la référence linéaire largement utilisé dans les applications en transport et proposé dans des systèmes comme Oracle (Oracle 2005). C'est ainsi que les types *gpoint* et *gline* sont représentés. Enfin, à partir des types de BASE, SPATIAL et GRAPH, ils dérivent les types temporels correspondants, en utilisant le constructeur *moving*. Les types temporels sont des fonctions ou des ensembles infinis de paires (instant, valeur). Une telle représentation infinie, concevable dans le modèle abstrait, ne peut être implémentée directement. Dans (Forlizzi *et al.*, 2000), une représentation discrète pour ces types est proposée. Nous discuterons plus en détail ce point dans la section 4 de ce papier.

Prédicats	isempty =, ≠, intersects, inside <, ≤, >, ≥, before touches, attached, overlaps, on_border, in_interior
Opérations ensemblistes	intersection, union, minus crossings, touch_points, common_border
Agrégations	min, max, avg, center, single
Opérations numériques	no_components, size, perimeter, duration, length, area
Distance et direction	distance, direction
Opération de bases	and, or, not

Tableau 2 : Classes d'opérations sur les types non-temporels (Gütting *et al.*, 2000)

Un ensemble d'opérations sont définies sur les types de données ci-dessus. Pour éviter la prolifération des opérations, le même opérateur s'applique en l'adaptant sur plusieurs types. Un ensemble d'opérations non-temporelles est d'abord défini comme dans Tableau 2, puis un processus dit de *lifting* permet de générer les opérations temporelles correspondantes. Ainsi, toutes les opérations non-temporelles sont prolongées sur les types temporels. Enfin, des opérations spécifiques sont ajoutées pour gérer les types temporels (Tableau 3). Dans le contexte des trajectoires

contraintes par le réseau (Güting *et al.*, 2006), quelques nouvelles opérations, comme la distance, ont été adaptées pour les types *gpoint* et *gline* (distance par la route). De nouvelles classes d'opérations sont également ajoutées pour analyser l'interaction entre l'espace du réseau et l'espace 2D, ainsi que des opérations spécifiques au réseau comme le parcours de graphe.

Projection dans le domaine/codomaine	deftime, rangevalues, locations, trajectory routes, traversed, inst, val
Interaction avec le domaine/codomaine	atinstant, atperiods, initial, final, present at, atmin, atmax, passes
When	when
Lifting	(toutes les opérations du Tableau 1 étendues aux types temporels)
Opérations mesurant le changement	derivative, speed, turn, velocity

Tableau 3: *Classes d'opérations sur les types temporels (Güting et al., 2000)*

Des travaux plus récents ont porté sur les capteurs agiles (Servigne *et al.*, 2007) ou encore sur l'extension à la mobilité de la spécification SOS de l'Open Geospatial Consortium (OGC) du service web pour l'échange de données de capteurs (Stasch *et al.*, 2008). Toutefois, ces travaux ne couvrent pas la problématique générale de modélisation et de gestion d'objets mobiles munis de mesures. De plus, dans les travaux mentionnés ci-dessus, les données sont modélisées comme des séries temporelles où la localisation est considérée comme un attribut parmi d'autres. A la différence de ces travaux, nous proposons un modèle qui offre une vue à la fois temporelle et spatiale des mesures, décrivant des profils temporels et spatiaux, ainsi qu'un langage pour gérer ces profils.

3. Modèle proposé

3.1. Motivation et exemples

Nous présentons ici une application qui a motivé notre travail. Un scénario de requêtes permet une illustration tout au long de cet article.

L'INRETS (Institut National de Recherche sur les Transports et leur Sécurité) a développé un Dispositif de Recueil de données Comportementales (DIRCO) pour des campagnes de *naturalistic driving*. Il s'agit d'un dispositif d'enregistrement embarqué et connecté au bus CAN. Il accède aux mesures : vitesse du véhicule,

vitesse individuelle de chaque roue, accélération longitudinale, odomètre, angle du volant, pédale de freinage (0/1), ABS (0/1), etc. Le DIRCO offre la possibilité de relier d'autres sources, comme un GPS, une centrale inertielle donnant l'accélération 3D et l'angle 3D du véhicule ou des caméras¹. Le DIRCO acquiert chaque flux de données comme une séquence temporelle. Les données venant d'une source sont stockées dans un fichier spécifique et chaque enregistrement est un tuple : $(t_i, \alpha_i^1, \alpha_i^2, \dots, \alpha_i^n)$ où t_i est l' $i^{\text{ième}}$ instant α_i^k est l' $i^{\text{ième}}$ valeur fournie par le $k^{\text{ième}}$ capteur. La particularité du DIRCO est qu'il permet une acquisition à une fréquence très élevée pouvant aller jusqu'à des cycles de 10ms. Les flux de données de sources différentes sont asynchrones.

Bien qu'il puisse fonctionner comme une boîte noire pour des véhicules afin de reconstituer les circonstances d'un accident, le DIRCO est principalement un outil de recherche qui peut aider à analyser le comportement de conduite, la sécurité du véhicule et de diagnostiquer des difficultés liées à l'infrastructure routières. Sa mémoire flash de 16GB permet l'acquisition de données, hors caméras, pendant plusieurs mois. Un scénario simple est d'équiper plusieurs véhicules, comme des autobus ou des voitures personnelles, avec le DIRCO, de récupérer et de centraliser ces données, puis de les analyser afin d'identifier des modèles comportementaux de conduite.

Ce type d'approche se généralise en vue de l'évaluation des systèmes d'aides à la conduite apparus récemment. Que ce soit les systèmes déjà bien connus comme les GPS ou les régulateurs de vitesse, ou bien des systèmes plus expérimentaux comme la détection d'obstacle, tous nécessitent une évaluation précise, longue et exhaustive de leur impact sur la conduite. La commission Européenne finance depuis 2008 des projets à grande échelle visant à évaluer des technologies matures appartenant à la catégorie des systèmes de « transport intelligent ». La particularité de ces projets est de recourir à un recueil systématique des données de conduite à l'aide d'un boîtier comparable au DIRCO développé à l'INRETS. Parmi ces projets figure le projet euroFOT (<http://www.eurofot-ip.eu/>) où des centaines de sujets seront étudiés pendant au moins un an, et les données seront ensuite analysées pour évaluer les systèmes étudiés afin d'en tirer des conclusions quand à leur généralisation aux prochaines générations de véhicules.

Au vu de ces informations, on comprend aisément l'intérêt qu'il y a à développer un SGBD adapté aux caractéristiques de ces données, que ce soit par rapport au volume de stockage, ou par rapport au caractère géolocalisé et horodaté. Les différents types de systèmes étudiés induisent une grande variabilité dans les méthodes d'analyse et impliquent parfois une grande précision des informations nécessaires (situations de presque-accident par exemple). Certains indicateurs peuvent certes être calculés avec des systèmes de gestion de base de donnée classiques, mais souvent au prix d'une programmation lourde et d'un temps de

¹ Les flux vidéo sont stockés à part et leur gestion ne fait pas partie du périmètre de cet article

calcul prohibitif. En outre, aucun système ne semble à l'heure actuelle capable de gérer simplement des profils de vitesse (ou toute autre information) mesurés à des temps et des positions différentes mais sur un même axe routier. Il s'avère pourtant qu'un grand nombre de questions associées à l'évaluation des systèmes d'aide à la conduite nécessitent ce genre d'approche.

Afin d'illustrer l'apport de notre modèle dans ce contexte, on se réfèrera tout au long de l'article aux requêtes typiques suivantes :

- Q1. Quel est le profil d'accélération le long d'un segment de route donné pour un trajet donné ?*
- Q2. Quelle est la différence entre le profil de vitesse et la vitesse légale le long d'un segment de route ?*
- Q3. Combien de fois le système ABS s'est-il activé pour un trajet donné ?*
- Q4. Quels sont les trajets où la vitesse excède un profil indiqué de vitesse (par exemple la vitesse légale) par une certaine valeur et quelle est la différence ?*
- Q5. Quel est le rapport entre la vitesse et le régime du moteur pour un trajet donné ?*
- Q6. Quel est le profil moyen d'accélération pour tous les véhicules qui traversent une certaine section de route (courbe) ?*
- Q7. Calculer le profil maximal de la vitesse pratiquée par tous les véhicules qui passent par la section de route indiquée.*
- Q8. Trouver le profil moyen de la vitesse réellement pratiqué (85^{ème} centile) sur une route avant et après l'installation d'un radar automatique.*
- Q9. Quel est le profil moyen de la consommation instantanée de carburant sur une route avant et après l'installation d'un aménagement routier (dos d'âne par exemple).*
- Q10. Quel est le profil minimal et maximal de consommation instantanée de carburant sur une route, et quelle est sa différence avec le profil du conducteur étudié.*

Modéliser des séquences temporelles est faisable en utilisant des fonctions sur le temps (Güting *et al.*, 2000), mais cela n'est pas utile pour le type d'analyse ci-dessus. En effet, les enregistrements par trajet sont relevés à des moments différents et la comparaison entre ces enregistrements n'a de sens que s'ils étaient mesurés au même endroit. Ce qui importe n'est pas l'instant auquel a été relevée la mesure, mais plutôt l'endroit où elle a eu lieu sur la route. La notion de profil spatial (vitesse, accélération) traduit la relation entre la mesure et l'espace. Or, cette notion de profil ne pourrait pas être définie dans le modèle de Güting, ni dans aucun autre modèle. Il est donc nécessaire de définir un nouveau modèle de données.

3.2. Introduction de nouveaux types de données

Tout comme le modèle algébrique de (Güting *et al.*, 2006) décrit ci-dessous, notre modèle intègre un type spatio-temporel pour modéliser la trajectoire de l'objet, ainsi qu'un type temporel pour modéliser les données générées par les capteurs. Le type temporel correspond à une fonction du temps vers les types de base (*real*, *int*). Il exprime ainsi la variabilité, du point de vue temporel, des mesures de capteurs.

Seulement, le point de vue temporel seul ne suffit pas pour modéliser les données de capteurs mobiles, car les mesures sont souvent étroitement liées à l'espace. Pour être complet, le modèle doit décrire en plus de l'évolution temporelle, l'évolution spatiale des mesures.

A cet effet, nous étendons le modèle de (Güting *et al.*, 2006). Nous introduisons un nouveau concept décrivant le profil des mesures dans l'espace. L'idée est de décrire l'évolution d'une mesure dans l'espace. Ce concept se décline en deux catégories : *SVARIANT* pour décrire le profil dans un espace libre, et *GARIANT* pour le profil le long du réseau.

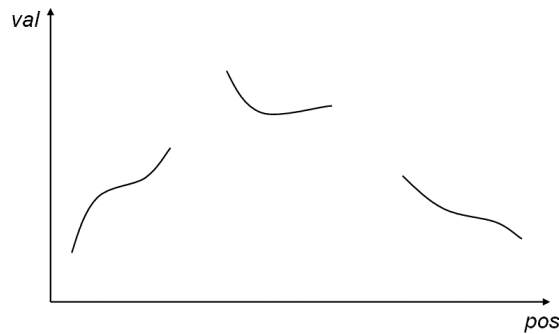


Figure 1. Exemple de profil spatial d'une valeur réelle

Nous proposons deux nouveaux constructeurs de types appelés *gmoving* et *smoving* :

$BASE \rightarrow SVARIANT$	<i>smoving</i>
$BASE \rightarrow GARIANT$	<i>gmoving</i>

Le définitions de ces constructeurs sont données ci-dessous en utilisant les notations de (Güting *et al.*, 2000) :

Définition 1 : Soit α un type de *BASE* ayant le domaine A_α . Alors le domaine de définition pour *smoving*(α) est défini comme suit: $A_{smoving(\alpha)} =$

$\{f \mid f : \bar{A}_{point} \rightarrow \bar{A}_\alpha \text{ est une fonction partielle } \wedge \Gamma(f) \text{ est fini}\}$, où $\bar{A}_\beta = A_\beta \setminus \{\perp\}$ et $\Gamma(f)$ désigne l'ensemble de composants continus maximaux de f .

Définition 2 : Soit α un type de *BASE* ayant le domaine A_α . Alors le domaine de définition pour $gmoving(\alpha)$ est défini comme suit: $A_{gmoving(\alpha)} = \{f \mid f : \bar{A}_{gpoint} \rightarrow \bar{A}_\alpha \text{ est une fonction partielle } \wedge \Gamma(f) \text{ est fini}\}$, où $\bar{A}_\beta = A_\beta \setminus \{\perp\}$ et $\Gamma(f)$ désigne l'ensemble de composants continus maximaux de f .

Étant donné que cet article se focalise sur la mobilité contrainte, nous ne détaillerons dans la suite que la deuxième catégorie de types. Chaque valeur f du domaine de $gmoving(\alpha)$ est une fonction décrivant l'évolution dans l'espace du réseau (*GRAPH*) à une valeur de *BASE*. Figure 1 présente un profil spatial d'une valeur réelle pour une route donnée. L'abscisse représente la position relative sur la route qui peut varier entre 0 et 1.

La condition “ $\Gamma(f)$ est fini” signifie que f est composée d'un nombre fini de composants continus. Le constructeur *gmoving* décrit un ensemble infini de paires (*position, valeur*), où la *position* est un *gpoint*. Par exemple, le profil dans la Figure 1 comprend trois composants continus. Cette condition est nécessaire pour que le modèle soit implémentable.

3.3. Introduction de nouvelles opérations

Tout comme la définition du type, nous utilisons l'algèbre d'opérateurs Gütting *et al.* (Gütting *et al.*, 2006) comme point de départ. En introduisant des nouveaux types, nous devons (i) prolonger les opérations existantes et (ii) étendre le modèle à de nouvelles opérations.

L'idée de prolonger les opérations existantes s'inspire du processus de *lifting temporal*, décrit dans la section 2. Celui-ci permet de générer à partir d'une opération de signature $\alpha_1 \times \alpha_2 \times \dots \times \alpha_n \rightarrow \beta$, l'opération temporelle équivalente de signature : $\alpha'_1 \times \alpha'_2 \times \dots \times \alpha'_n \rightarrow moving(\beta)$, où $\alpha'_i \in \{\alpha_i, moving(\alpha_i)\}$. Chacun des arguments peut devenir temporel rendant le résultat également temporel. Nous empruntons ce principe pour générer les opérations équivalentes variant dans l'espace. Nous proposons ainsi un *lifting spatial* pour les opérations *non-gvariant non-temporelles*. L'opération générée par ce *lifting* devient disponible pour des signatures $\alpha'_1 \times \alpha'_2 \times \dots \times \alpha'_n \rightarrow gmoving(\beta)$, ou $\alpha'_i \in \{\alpha_i, gmoving(\alpha_i)\}$.

Par ailleurs, nous avons défini de nouvelles opérations sur les types *GVARIANT*, et des interactions entre ces types et le réseau. Le tableau 4 ci-dessus décrit d'une manière non exhaustive les principales opérations ainsi définies. Nous donnons dans ce paragraphe quelques explications supplémentaires sur certaines opérations. D'autres opérations sont introduites avec les requêtes dans la section suivante. On distingue cinq classes d'opérations. Les quatre premières représentent des opérations classiques, c'est à dire celles qui prennent à l'entrée un ou plusieurs objets (valeurs) conformément à la signature de l'opération et qui retourne un objet (une valeur). La

dernière classe comprend des opérations agrégats, c.à.d. qui retourne un seul résultat basé sur un groupe d'objets (analogue aux agrégats dans le modèle relationnel).

La première fait la projection dans le domaine du réseau ou dans le domaine des valeurs, appelé codomaine. Ainsi, **trajectory** retourne la trajectoire pour la partie continue. L'opération **rangevalues** fait la projection dans le codomaine et retourne un ou plusieurs intervalles de valeurs. Les opérations **val** et **pos** retournent respectivement la valeur ou le point sur le réseau pour un type *ingpoint*, lequel est défini par la paire (*gpoint*, *valeur*). Quant à la deuxième classe d'opérations, elle concerne les interactions entre les nouveaux types et les domaines (réseau) et le codomaine (valeurs). Elles réalisent des sélections ou des clipping en fonction de critères sur l'un ou l'autre des axes de variation (réseau ou valeurs). Ainsi, **present** est un prédicat qui vérifie si l'objet en entrée est défini sur un point donné du réseau ou sur un intervalle spatial donnée. Enfin, le prédicat **passes** vérifie si l'objet passe par les valeurs fournies en paramètre.

Dans la troisième classe d'opérations, nous incluons les opérations algébriques de base ('+', '-', '.', '/') dans les opérations non-gvariant non-temporelles. Par conséquent, elles deviennent sujettes au *lifting* temporel et spatial. Nous utilisons des fonctions nommées, c'est-à-dire **sum**, **sub**, **mul** et **div**, comme pour toutes les opérations définies. Ces opérations sont utiles à l'analyse fine des mesures de capteurs. Par exemple, elles permettent de calculer la différence entre les vitesses pratiquées de deux objets mobiles sur la partie commune de leurs trajectoires, ou encore de retourner la différence entre les vitesses des objets mobiles et la vitesse légale. Ces opérations prennent en entrée deux fonctions du même type (*temporel* ou *gvariant*) et calculent une fonction résultat dont le domaine de définition est l'intersection des domaines des fonctions paramètres. Pour la division, les parties où l'opération n'est pas définie, sont également éliminées du domaine de la fonction résultat.

La quatrième classe d'opérations adresse la même catégorie des types, c'est-à-dire des types temporels et *gvariant* de BASE pour calculer une valeur numérique. Il s'agit des fonctions : **mean**, **min**, **max** et **no_transitions**. Chacune de ces opérations prend comme entrée une fonction du temps ou de l'espace et retourne une valeur réelle représentant l'agrégat de la fonction d'entrée. Leur intérêt est de permettre de calculer une moyenne ou une valeur extrême pour n'importe quelle mesure, et ce sur un intervalle temporel ou spatial.

La dernière classe d'opérations concerne les opérations d'agrégation. Les opérations agrégats retournent comme résultat un seul objet à partir d'un ensemble d'objets du même type (cf. figure 2). Elles définissent ainsi des agrégations d'un groupe d'objets.

Certains de ces agrégats retournent un objet de même type comme le profil moyen (**avg_agg**), le profil minimal (**min_agg**) et maximal (**max_agg**), ou encore le profil résultant du cumul (**min_agg**). L'agrégat **count_agg** retourne le nombre de profils sur le domaine de définition sous forme d'un objet de type *moving(int)* ou

$gmoving(int)$. Enfin, la fonction percentile calcule le profil au-dessous duquel se trouve un certain **pourcentage** de profils de l'ensemble donné en paramètre. Le domaine de définition de la fonction résultat est l'union des domaines des fonctions agrégées. L'utilité des fonctions agrégats est montrée par les requêtes Q6 à Q10.

Classe d'opération	Opération	Signature
Projection dans le domaine et codomaine	trajectory	$gmoving(\alpha) \rightarrow gline$
	rangevalues	$gmoving(\alpha) \rightarrow range(\alpha)$
	pos	$ingpoint \rightarrow gpoint$
	val	$ingpoint \rightarrow \alpha$
Interaction avec le domaine et codomaine	atpos	$gmoving(\alpha) \times gpoint \rightarrow ingpoint$
	atgline	$gmoving(\alpha) \times gline \rightarrow gmoving(\alpha)$
	present	$gmoving(\alpha) \times gpoint \rightarrow bool$
		$gmoving(\alpha) \times gline \rightarrow bool$
	at	$gmoving(\alpha) \times \alpha \rightarrow gmoving(\alpha)$
		$gmoving(\alpha) \times range(\alpha) \rightarrow gmoving(\alpha)$
	atmin	$gmoving(\alpha) \rightarrow gmoving(\alpha)$
atmax	$gmoving(\alpha) \rightarrow gmoving(\alpha)$	
passes	$gmoving(\alpha) \times \beta \rightarrow bool$	
Opérations algébriques de base	sum, sub, mul, div	$moving(\alpha) \times moving(\alpha) \rightarrow moving(\alpha)$ $gmoving(\alpha) \times gmoving(\alpha) \rightarrow gmoving(\alpha)$
Calculs	mean[avg], min, max	$moving(\alpha) \rightarrow real$
		$gmoving(\alpha) \rightarrow real$
	no_transitions	$moving(\alpha) \rightarrow int$ $gmoving(\alpha) \rightarrow int$
Agrégats d'ensemble d'objets	min_agg, max_agg, sum_agg, avg_agg	$\{moving(\alpha)\} \rightarrow moving(\alpha)$
		$\{gmoving(\alpha)\} \rightarrow gmoving(\alpha)$
	count_agg	$\{moving(\alpha)\} \rightarrow moving(int)$ $\{gmoving(\alpha)\} \rightarrow gmoving(int)$
percentile	$\{moving(\alpha)\} \times real \rightarrow moving(\alpha)$	
	$\{gmoving(\alpha)\} \times real \rightarrow gmoving(\alpha)$	

Tableau 4: Exemples d'opérations pour les nouveaux types

L'ensemble d'opérations proposé n'est qu'une base, toutefois riche, de fonctionnalités. D'autres opérations pourront être rajoutées pour satisfaire des besoins spécifiques d'application. Grâce aux progrès dans l'extensibilité des SGBD,

ces types et ces opérations définies de manière abstraites, peuvent aisément être intégrées au SGBD. Il devient alors possible de les exploiter à travers le langage standard SQL. Il reste ensuite à résoudre les problèmes d'optimisation de requêtes pouvant se poser. C'est exactement le schéma que nous avons suivi pour mettre en œuvre notre serveur de données pour des objets et des capteurs mobiles.

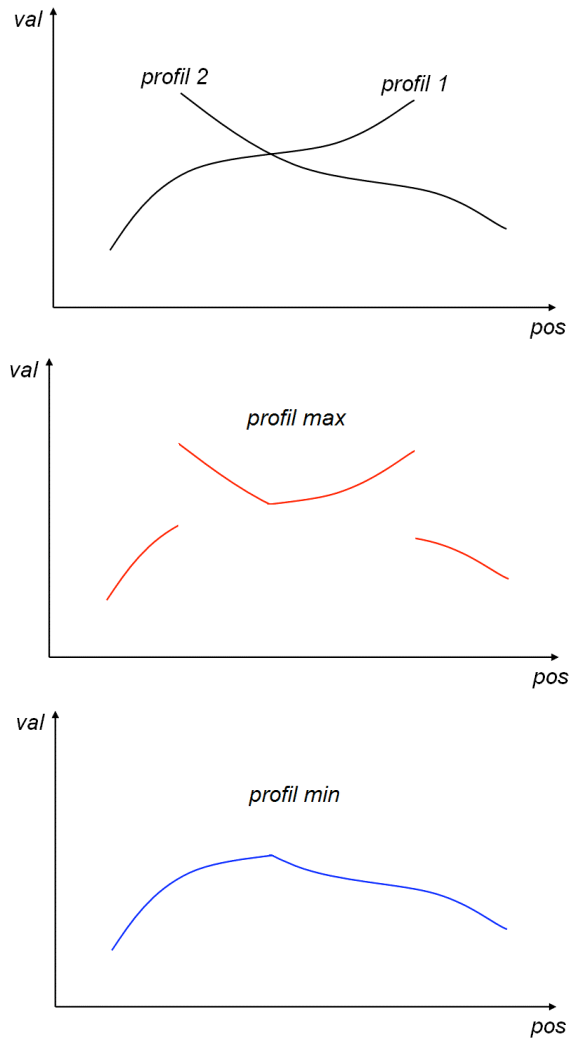


Figure 2. Exemple d'utilisation de la fonction `max_agg` (deuxième graphique) et `min_agg` (troisième graphique) sur deux profils (premier graphique)

3.4. Exemples de requêtes

Le grand intérêt d'utiliser les capacités d'extension du SGBD est d'intégrer facilement les nouveaux types et les opérations dans l'interface SQL standard (voir section 3.1). Les exemples se basent sur une relation contenant l'information sur des véhicules de schéma suivant :

```
vehicle_trip(mo_id: int, trip: moving(gpoint), g_speed:
gmoving(real), t_speed: moving(real), g_acceleration:
gmoving(real), t_acceleration: moving(real), g_RPM:
gmoving(real), t_RPM: moving(real), g_odometer: gmoving(real),
t_odometer: moving(real), g_ABS: gmoving(bool), t_ABS:
moving(bool), g_breakSwitch: gmoving(real), t_breakSwitch:
moving(real))
```

En plus de la trajectoire spatio-temporelle « trip », la relation contient des données de capteurs reportant la vitesse, l'accélération, le RPM, l'odomètre et l'état de l'ABS et de la pédale de frein. Ces données sont modélisées par des fonctions de l'espace (préfixées avec *g_*) et par des fonctions du temps (préfixées avec *t_*). Les paramètres sont préfixés avec le symbole "&" et pourraient être, soit donnés par l'utilisateur au moment de l'exécution, soit issus de calculs préalables.

Q1: *Quel est le profil d'accélération le long d'un segment de route donné pour un trajet donné ?*

```
SELECT atgline(g_acceleration, &aGline)
FROM vehicle_trip
WHERE mo_id = &anID
```

L'opération **atgline** permet de retourner le profil de l'accélération longitudinale restreint au sous-espace indiqué par la géométrie de la route *aGline* donné en paramètre.

Q2: *Quelle est la différence entre le profil de vitesse et la vitesse légale le long d'un segment de route ?*

```
SELECT sub(g_speed, &legalSpeed)
FROM vehicle_trip
WHERE inside(trajectory(&legalSpeed), trajectory(g_speed))=1
```

La différence entre deux fonctions décrivant des profils de mesures est calculée en utilisant l'opération **sub**. Un prédicat indexé tel que **inside** pourrait accélérer le temps de réponse de la requête. Cette opération a deux paramètres *gline* et vérifie si le premier est inclus dans le deuxième. Afin d'obtenir la projection dans l'espace d'un profil de données capteur, on peut utiliser l'opération **trajectory**. C'est une opération de type projection dans domaine.

Q3: *Combien de fois le système ABS s'est-il activé pour un trajet donné ?*

```
SELECT no_transitions(g_ABS) / 2
```

```
FROM vehicle_trip
WHERE mo_id = &anID
```

Cette requête illustre simplement l'utilisation de la fonction **no_transitions** qui est applicable aux types discrets de *BASE* (*bool*, *int*) et qui compte le nombre de transitions pour une fonction discrète donnée.

Q4: *Quels sont les trajets où la vitesse excède un profil indiqué de vitesse (par exemple la vitesse légale) par une certaine valeur et quelle est la différence ?*

```
SELECT mo_id, sub(g_speed, &legalSpeed)
FROM vehicle_trip
WHERE intersects(trajectory(&legalSpeed),
                trajectory(g_speed)) = 1
AND max(sub(g_speed, &legalSpeed)) > &threshold
```

Il y a deux nouvelles opérations dans cette requête. D'abord, le prédicat **intersects** est similaire à **inside**, la seule différence étant qu'il recherche seulement une intersection entre les deux *gline* et pas une inclusion. Après, l'opération **max** est une agrégation d'une fonction. Elle vérifie si la valeur maximale assumée par la fonction indiquée comme paramètre est au-dessus d'une certaine valeur seuil. Comme dans la requête précédente, le paramètre pour **max** est représenté par la différence entre la vitesse pratiquée et les profils légaux de vitesse.

Q5: *Quel est le rapport entre la vitesse et le régime du moteur pour un trajet donné ?*

```
SELECT div(t_speed, t_RPM)
FROM vehicle_trip
WHERE mo_id = &anID
```

Cette requête montre l'utilité des opérations algébriques de base pour comparer des profils temporels du même objet mobile. Le profil obtenu par le rapport entre la vitesse du véhicule et le régime du moteur peut être utilisé pour détecter le comportement concernant le passage des vitesses d'un certain conducteur.

Q6: *Quel est le profil moyen d'accélération pour tous les véhicules qui traversent une certaine section de route (courbe) ?*

```
SELECT avg_agg(g_acceleration)
FROM vehicle_trip
WHERE inside(trajectory(&aCurve), trajectory(trip))=1
```

Nous déterminons avec cette requête le profil moyen d'accélération de tous les véhicules qui traversent la section de route indiquée. La fonction **avg_agg** génère un nouvel objet *gmoving(real)* à partir de l'ensemble d'objets de même type, passe comme paramètre, c.à.d. tous les tuples de la table qui correspondent au prédicat dans la clause WHERE.

Q7: *Calculer le profil maximal de la vitesse pratiquée par tous les véhicules qui passent par la section de route indiquée.*


```
SELECT max_agg(atgline(g_speed, &aRoad))
FROM vehicle_trip
WHERE intersects(trajectory(g_speed), &aRoad) = 1
```

D'abord nous trouvons tous les trajets qui intersecte la route indiquée. Pour ces trajets, nous sélectionnons par la fonction **atgline** la partie du profil de la vitesse qui correspond à la route. Ensuite, nous agrégeons les profils ainsi obtenus pour obtenir le profil maximal, par la fonction agrégat **max_agg**.

Q8: Trouver le profil moyen de la vitesse réellement pratiqué (85^{ème} centile) sur une route avant/après l'installation d'un radar automatique.

```
SELECT percentile(atgline(g_speed, &aRoad), 85)
FROM vehicle_trip
WHERE intersects(trajectory(g_speed), &aRoad) = 1
AND inst(initial(trip)) < &installationDate
```

La requête présentée trouve le profil moyen avant l'installation du radar automatique. Une requête similaire doit être posée pour trouver le même profil après l'installation du radar. Comme pour la requête *Q7*, nous filtrons les trajets en retenant seul ceux qui ont une intersection avec la route indiquée et qui débutent avant la date d'installation du radar. Pour cela nous employons la combinaison des fonctions **inst** et **initial** qui retournent la date de début d'un trajet. Finalement, on applique la fonction **percentile** sur l'ensemble de profils sélectionnés. Le deuxième paramètre de cette fonction représente le n^{ème} centile.

Comme nous pouvons le voir, le nouveau type et les opérations sont nécessaires pour exprimer ces requêtes.

4. Implémentation

Actuellement, le support des données spatio-temporelles dans les SGBD existants est limité. Cependant, la plupart des SGBD offrent aujourd'hui des possibilités d'extension pour satisfaire les besoins d'un certain domaine d'application. Plutôt que de développer un prototype partant de rien, nous avons choisi de mettre en œuvre le modèle proposé sous un système offrant cette extensibilité, en l'occurrence, le SGBD Oracle. Ainsi, tous les types sont implémentés comme des nouveaux types d'objets sous Oracle 11g. Les opérations sont implantées en Java et stockées comme un paquetage dans la base de données.

Pour tous les types *moving* et *gmoving*, nous adoptons la représentation par morceau comme proposé dans (Forlizzi *et al.*, 2000) et l'étendons aux nouveaux types. Chaque morceau (unité) peut être une valeur constante, une fonction linéaire du temps ou de l'espace, ou même une fonction plus complexe d'interpolation. Par exemple une unité pour un *gmoving(real)* contiendra les prochains attributs : (*a*, *b*, *c*, *pos1*, *pos2*), alors qu'une unité pour un *moving(real)* contiendra (*a*, *b*, *c*, *t1*, *t2*). Les paramètres *a*, *b*, *c* représentent les coefficients d'un polynôme quadratique, tandis

que $(pos1, pos2)$ et $(t1, t2)$ représentent un intervalle spatial ou temporel qui donne la longueur de l'unité. Pour calculer une valeur à une position donnée, on localise d'abord l'unité correspondante, c.-à-d., là où l'intervalle inclut la position indiquée, puis on calcule la valeur comme $a*pos2+b*pos+c$.

Le développement du système présenté suit trois principales étapes:

- implémentation de l'algèbre (les nouveaux types et opérations)
- création de nouveaux types d'index pour les prédicats spatial, temporel et sur les valeurs de capteurs
- visualisation graphique des résultats de requêtes

Complémentaire à ces directions, il y a un travail important concernant le passage entre les données brutes, comme enregistré par DIRCO, et les données modélisées, comme stocké dans la base de données. En effet, nous pouvons prendre comme exemple les données de GPS. Le GPS produit des positions bidimensionnelles (latitude, longitude), alors que notre modèle considère un positionnement relativement à un réseau routier. Ainsi, un *map matching* (Brakatsoulas *et al.*, 2005) doit être effectué afin de transformer les positions 2D en positions sur un graphe. Ceci est assez compliqué, car la structure du réseau est relativement complexe et la précision du GPS est relativement basse.

4.1. Traitement des fonctions d'agrégation

Contrairement aux fonctions portant sur un ou deux profils, les fonctions d'agrégation est fait sur un ensemble contenant un nombre important de profils, ce qui peut générer un morcellement du profil résultat en un nombre élevé de petites unités et une dégradation de performances. Prenons l'exemple dans la Figure 3. Le premier graphique montre deux profils avec leur décomposition en unités et le deuxième représente le profil maximum de ces profils. Nous remarquons que les unités des deux profils n'ont pas la même répartition dans l'espace. Les intervalles spatiaux des unités des deux profils se recouvrent partiellement. Ceci est courant, car le découpage en unités est propre à chaque profil et dépend fortement de la variabilité de la mesure observée, du moment du temps de l'observation et de la fréquence de recueil des valeurs envoyées par le capteur.

Nous présentons dans la Figure 4 un exemple d'utilisation du découpage régulier pour le calcul des fonctions d'agrégation sur deux profils spatiaux. Le premier graphique représente la décomposition des profils par les intervalles réguliers (représenté par les droites verticales en pointillés). Pour chaque intervalle, on calcule ou on extrapole d'abord les valeurs aux positions limites de l'intervalle. Ainsi, pour le premier profil (en rouge), on trouve les valeurs v_1^1 et v_2^1 pour le premier intervalle, v_2^1 et v_3^1 pour le deuxième intervalle et v_3^1 et v_4^1 pour le troisième intervalle. A partir de ces valeurs calculées pour tous les profils, on applique la

fonction agrégat scalaire correspondante (ex. l'agrégat max pour **max_agg**) sur les valeurs pour générer les valeurs du profil résultat à ces mêmes limites d'intervalles.

Les propriétés algébrique, distributive et holistiques des fonctions agrégats scalaires sont conservées pour les nouvelles fonctions, ce qui permet une implémentation incrémentale pour les fonction **max_agg**, **min_agg**, **count_agg** et **sum_agg**. Quant à la fonction **avg_agg**, elle se base sur le calcul en parallèle de **sum_agg** et de **count_agg** et calcule leur rapport. Finalement, la fonction analytique **percentile** ne peut être évaluée progressivement. Nous calculons d'abord tous les points d'intersection de tous les profils avec les intervalles réguliers. Pour chaque intervalle, nous choisissons la paire de points qui domine une partie de l'ensemble complet des points, égale au pourcentage donné comme deuxième paramètre de **percentile**.

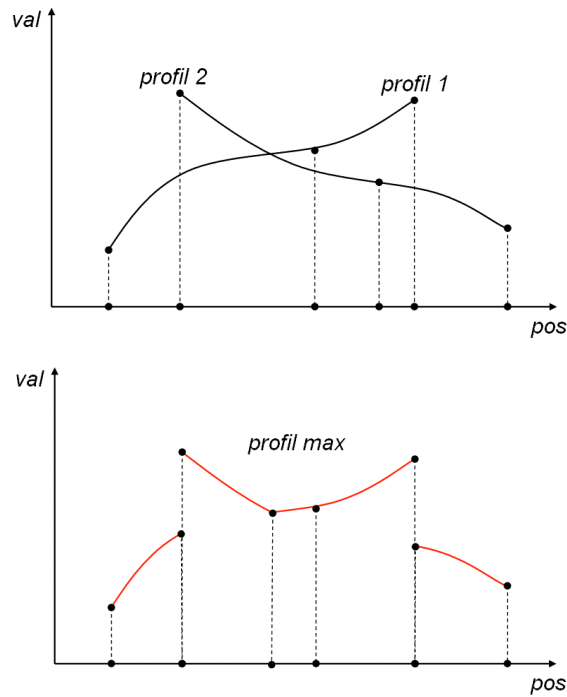


Figure 3. Exemple de fragmentation après l'utilisation de la fonction **max_agg** (deuxième graphique) sur deux profils (premier graphique)

Globalement, cette approche pour l'implémentation des fonctions d'agrégation produit des résultats approximatifs, mais elle offre en contrepartie une bonne optimisation de ce type d'opérations extrêmement coûteuses. L'analyse de la qualité du résultat en fonction de la granularité du découpage fait partie des travaux futurs.

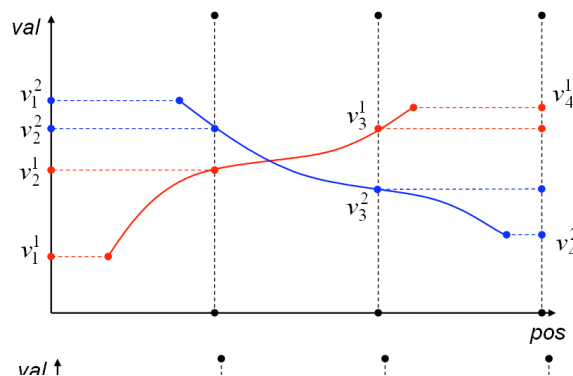


Figure 4. Exemple de calcul de la fonction `max_agg` (deuxième graphique) et `min_agg` (troisième graphique) sur deux profils (premier graphique) en utilisant un découpage régulier

4.2. Visualisation des résultats

L'état actuel du développement, bien que loin d'être complet, a permis de valider l'approche présentée. En effet, nous disposons actuellement d'un nombre suffisant d'opérations implémentées pour pouvoir tester plusieurs requêtes répondant au scénario d'application présenté dans l'introduction de cet article. Nous avons utilisé des jeux de test réels, provenant de recueils effectués par le LIVIC en marge de l'expérimentation du système LAVIA, et d'autres synthétiques générés par une extension du générateur de Brinkhoff (Brinkhoff 2002). Les jeux de test synthétiques ont servi principalement à évaluer le passage à l'échelle des opérateurs indexés (Sandu Popa et Zeitouni, 2008). Dans la suite, nous donnons une des requêtes SQL testées et présentons le résultat graphique obtenu.

Q9: "Où est-ce qu'un conducteur donné dépasse la limite de vitesse ?"

```
SELECT trajectory(greaterThan(g_speed, &legalSpeed)
FROM vehicle_trips
WHERE id = &aDriver;
```

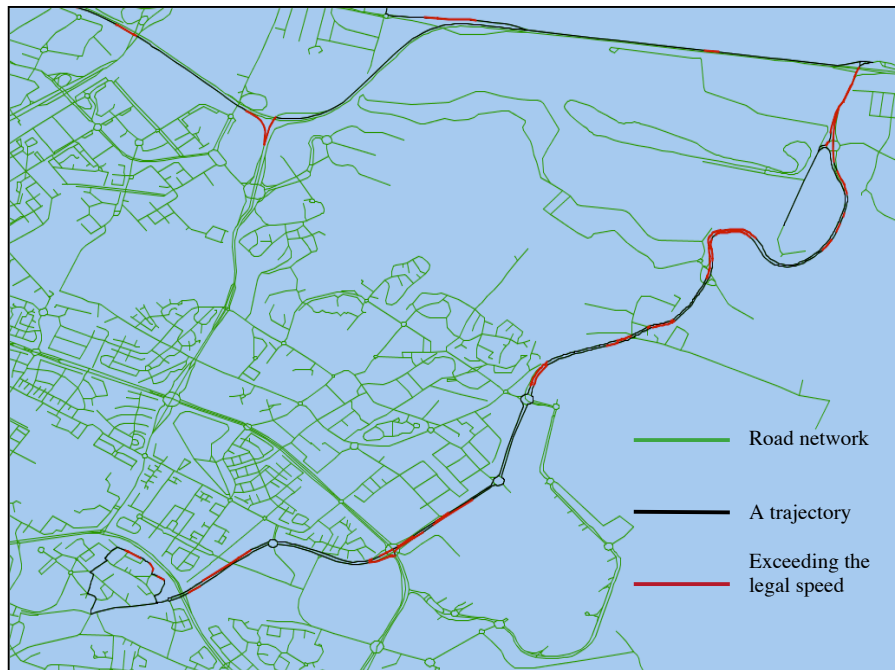


Figure 5. *Comparaison entre la vitesse pratiquée et la vitesse légale pour un trajet*

Cette requête est une simplification de *Q4* du scénario de requêtes dont le résultat peut être illustré aisément sur une carte. La Figure 5 représente un résultat visuel pour une trajectoire réelle d'un véhicule et une carte qui a des informations de limites de vitesse. La visualisation est faite sous Oracle *Map Builder*, un outil offert par Oracle qui permet créer des cartes 2D. Dans la figure, les lignes vertes représentent le réseau, les noires donnent la trajectoire, alors que les rouges indiquent les sections de route où le véhicule se déplace avec une vitesse excédant la vitesse légale.

5. Conclusion et axes de recherche

L'exploitation des capteurs embarqués à bord des véhicules amène à des nouvelles applications, lesquelles font émerger de nouveaux problèmes de recherche. Nous nous sommes intéressés ici au problème de modélisation et de requêtes. Nous avons souligné les limitations des travaux existants portant sur les objets mobiles. Le nouveau type d'applications concerne des objets mobiles équipés de capteurs produisant des flux de données additionnels et qui sont fortement liés à la trajectoire de l'objet. Un SGBD capable de gérer de manière unifiée les données concernant l'objet mobile et les flux de données correspondant est d'une grande utilité pour ces applications.

La contribution de cet article est de proposer un modèle pour un tel SGBD en étendant un système existant pour les objets mobiles. Nous avons analysé d'abord les limitations de la modélisation des données de capteurs. En effet, les modèles existants permettent de représenter les flux de données d'un point de vue temporel. Nous avons montré que ces mesures dépendent autant, sinon plus, de l'espace et qu'une représentation par référence à l'espace est nécessaire. Par conséquent, nous avons étendu les types du SGBD avec des fonctions qui décrivent l'évolution des mesures dans l'espace. Nous avons également proposé un ensemble d'opérations dans la perspective du système enrichi. Nous avons introduit la notion de *lifting spatial* inspirée par l'idée de *lifting temporel* existante. Nous avons redéfini toutes les opérations temporelles et modifié la sémantique de certaines d'entre elles pour les nouveaux types. Enfin, nous avons proposé de nouvelles opérations, ainsi que des agrégats sur les nouveaux types. Une illustration d'utilisation du SGBD est donnée par des exemples de requêtes impliquant les nouveaux types et les opérations définies. Le prototype actuel comprend une implémentation partielle de l'algèbre comme une extension du SGBD Oracle par la technique de *cartridge*.

Nous avons pu montrer une application réelle d'analyse de conduite naturelle en liaison avec les systèmes de transports intelligents (Sandu Popa *et al.*, 2008b) et les résultats sont prometteurs pour une application à plus grande échelle.

Le type d'application présenté concerne généralement des jeux de données très volumineux. La suite naturelle de ces travaux est la proposition de techniques d'indexation des nouveaux types. Nous avons exploré une technique d'indexation se ramenant à des arbres B+ et l'avons implémenté dans le *cartridge*. Les résultats obtenus montrent qu'une telle optimisation passe à l'échelle et est absolument nécessaire (Sandu Popa et Zeitouni, 2008). Néanmoins, ceci ne représente qu'un travail préliminaire et une base pour les travaux plus complets sur l'optimisation de requêtes pour notre système.

Remerciements

Nous aimerions remercier le laboratoire de LIVIC, unité mixte de l'INRETS et du LCPC, pour avoir fourni des cartes et des jeux de données réels. Ce projet est soutenu par le financement d'une allocation doctorale de la Région Ile-de-France.

Références

- Brakatsoulas S., Pfooser D., Salas R., Wenk C.: *On Map-Matching Vehicle Tracking Data*. VLDB 2005: 853-864
- Brinkhoff, T. A framework for generating network-based moving objects. *GeoInformatica*, 6(2): 153-180, 2002.
- Burns, P.C. *International Harmonized Research Activities – Intelligent Transport Systems (IHRA-ITS) working group report*. 19th International Technical Conference on the Enhanced Safety of Vehicle (ESV), 2005.
- Forlizzi L., Güting R.H., Nardelli E., and Schneider M., *A Data Model and Data Structures for Moving Objects Databases*. ACM SIGMOD: 319-330, 2000.
- Galit T., *Analysis and Modeling of Driving Behavior Using In-Vehicle Data Recorder Data*, 11th World Conference on Transportation Research, 2007.
- Grumbach S., Rigaux P., Segoufin L., *Spatio-Temporal Data Handling with Constraints*, *GeoInformatica*, 5(1), 2001
- Güting R.H., Böhlen M.H., Erwig M., Jensen C.S., Lorentzos N.A., Schneider M., and Vazirgiannis M., *A Foundation for Representing and Querying Moving Objects*. ACM Transactions on Database Systems, 25(1): 1-42, 2000.
- Güting R.H., Schneider M., *Moving Objects Databases*, Morgan Kaufmann, 2005
- Güting R.H., Almeida V.T., and Ding Z., *Modeling and Querying Moving Objects in Networks*. VLDB Journal 15(2): 165-190, 2006.
- INRETS: <http://www.inrets.fr/ur/livic/>
- ISO 19107:2003, Geographic Information – Spatial Schema, WG 2.

- Koubarakis M., Pernici B., Schek H.J., Scholl M., Theodoulidis B., Tryfona N., Sellis T., Frank A.U., Grumbach S., Güting R.H., Jensen C.S., Lorentzos N., Manolopoulos Y., and Nardelli E. (Eds.), *Spatio-Temporal Databases: The CHOROCHRONOS Approach*. Springer-Verlag, Lecture Notes in Computer Science 2520, 2003.
- NHTSA (2006). The 100-Car Naturalistic Driving Study, Phase II – Results of the 100-Car Field Experiment. Report No. DOT HS 810 593
- Oracle Database Data Cartridge Developer's Guide, 10g Release 2, 2005.
- Pelekis N., *STAU: A Spatio-Temporal Extension to ORACLE DBMS*, PhD Thesis UMIST, Department of Computation, 2002.
- Pelekis N., Theodoulidis B., Kopanakis I., Theodoridis Y.: *Literature Review of Spatio-Temporal Database Models*, The Knowledge Engineering Review Journal, 19(3), 235-274, June 2005.
- Pelekis N., Theodoridis Y., *Boosting Location-Based Services with a Moving Object Database Engine*, Proceedings of MobiDE, 2006.
- Pelekis N., Frenzos E., Giatrakos N., Theodoridis Y.: *HERMES: Aggregative LBS via a Trajectory DB Engine*, SIGMOD 2008: 1255-1258.
- Pfoser D. et Jensen C.S., *Indexing of Network-constrained Moving Objects*. ACM-GIS, 25-32, 2003.
- Sandu-Popa I., Kharrat A., Zeitouni K., CALM : Un système de gestion de données de CApteurs à Localisation Mobile, Conférence Spatial Analysis and GEomatics, SAGEO 2008, Montpellier, France, June 2008.
- Sandu Popa I., Zeitouni K., Implementation and Optimization of a Mobile Location Sensor Database, Rapport PRISM #2008/3, <http://www.prism.uvsq.fr/rapports/bin/abstract.php?id=502>
- Sandu Popa I., Zeitouni K., Dupin F., Saint-Pierre G., Glaser S., Using in-Vehicle Sensor Data for Naturalistic Driving Analysis, 15th World Congress on Intelligent Transportation Systems (ITS'2008), 2008.
- Savary L., Wan T., Zeitouni K., "Spatio-Temporal Data Warehouse Design for Activity Pattern Analysis", Int. DEXA Workshop on Geographic Information Management, 2004.
- Servigne S., Devogele T., Bouju A., Bertrand F., Gutiérrez C., Laucius S., Noël G., Ray C., Gestion de masses de données temps réel au sein de bases de données capteurs, Conférence Spatial Analysis and GEomatics, SAGEO 2007, Clermont-Ferrand, France, Juin 2007.
- Sistla P., Wolfson O., Chamberlain S., Dao S., *Modeling and Querying Moving Objects*, Proc. 13th Int'l Conf. on Data Engineering (ICDE13), Birmingham, UK, 1997.
- Stasch C., Bröring A., Walkowski A. C., *Providing Mobile Sensor Data in a Standardized Way - The SOSmobile Web Service Interface*, 11th AGILE International Conference on Geographic Information Science 2008.
- Vazirgiannis. M. and Wolfson. O.: A Spatiotemporal Query Language for Moving Objects. Proc. of SSTD: 20-35, 2001.

24 Revue. Volume X – n° x/année

Wolfson, O.: *Moving Objects Information Management: The Database Challenge*. NGITS: 75-89, 2002

Yeh T-S., Modélisation de la variabilité des entités dans un Système d'Information Géographique, Doctorat de l'Université Pierre et Marie Curie (Paris VI), Janvier 1995.