Vol.2013-ITS-53 No.2
2013/6/14

ストリーム LDM における地図データのストリーム化機構の設計と評価

伊藤 信一[†1†3]　山口 晃広[†1]　佐藤 健哉[†1†2]　本田 晋也[†1]　高田 広章[†1]

概要：協調 ITS において，各車両に搭載されたセンサや車々間通信により取得した周辺交通情報や地図情報を階層的に管理する Local Dynamic Map（LDM)の検討がすすんでいる．我々は，LDM にデータストリーム管理システム(DSMS)を適用することにより高速な処理を実現するストリーム LDM を提案してきたが，静的な地図データはデータベース上で扱うため，地図データを取得，処理するための遅延時間が性能上のボトルネックとなっていた．本論文では，ストリーム LDM に地図データのストリーム化機構を導入することにより，静的な地図データも DSMS 上で高速に処理することを可能とし，この問題を解決できることを示す．車両走行シミュレーションデータを使った評価により，地図データとの空間演算の平均レイテンシにおいて，従来方式の数 10 倍の性能を得られることを確認した．

Design and Evaluation of Map Data Processing Mechanism for Stream LDM

SHINICHI ITO[†1†3]　AKIHIRO YAMAGUCHI[†1]　KENYA SATO[†1†2]
SHINYA HONDA[†1]　HIROAKI TAKADA[†1]

Abstract: Local Dynamic Map(LDM) has been studied for ITS Cooperative Systems. The LDM is a hierarchical system that manages driving environment recognition information sent from vehicle on-board sensors and through vehicle-to-vehicle communications or map data. We have proposed “Stream-LDM” using Data Stream Management System (DSMS) to realize high response. However, we found processing time for map data on data base to be performance bottlenecks. In this paper, we describe that this problem is resolved by Map Data Processing Mechanism which makes it possible to process static map data in DSMS. By Evaluation using vehicle driving simulation, we confirmed that the average performance of spatial operator for map data to be improved several 10 times as much as the conventional methods.

1. はじめに

近年，協調ITSの普及に伴い，レーダやカメラなど自車センサから得られる情報に加えて，車々間，路車間通信により車両周辺の情報を取得し，衝突防止などの安全運転支援を行なうシステムの研究が実用化に向けてすすんでいる[1][2]．更なる発展として，高精度センサ，地図情報を用いた自動運転システムの研究，開発も活発化しており[3][4]，これらのシステムを効率的に実現するための基盤技術の開発が求められている．

これらのシステムでは，周辺車両，道路状態，交通状況，地図情報など様々なデータが利用される．欧州の標準化組織であるETSIでは，協調ITS の世界標準化を進めており，これら多様なデータを管理するための仕様としてLocal Dynamic Map（LDM）が提案されている[5]．LDMは，車両走行情報のような動的なデータから地図情報のような静的なデータまで，車両走行環境に関するデータを統合的に保持，管理する．安全運転支援などのアプリケーションは，LDMを通じて，車両データ，信号データ，地図データなど必要なデータを取得する．LDMに関するプロジェクトとしては，欧州のSAFESPOTが挙げられる[6]. SAFESPOTでは，PostgreSQL[7]やSQLite[8]といったRDBMSを利用して，LDMを実現している．これに対して，我々は，データストリーム管理システム（DSMS）[9]とリレーショナルデータベース管理システム（RDBMS）とを併用することによって高速なLDMを実現する技術をストリームLDMとして提案してきた[10][11][12]．DSMSでは，あらかじめ登録されたクエリに従って，到着したデータをメモリ上でリアルタイムに処理を行う．この仕組みによって，すべてをRDBMS上で処理を行なう場合と比較して，データベースへのアクセスを減らし，高速化を実現できる．但し，従来手法では，静的な地図データは RDBMS上で扱っていたため，地図データを取得，処理するための遅延時間が性能的なボトルネックとなっていた．LDMでは，車両情報や危険情報のような動的なデータを，その位置情報にもとづき，地図データ上にマッピングすることによって，周辺車両や危険物と自車両との関係を認識し，衝突検知や危険検知などを実現する．この地図データとの演算の高速化がLDMを利用するシステムにおいては必須の課題である. そこで, 本論文では，ストリームLDMに地図データのストリーム化機構を導入することにより，この問題を解決する手段を示す．

本論文の構成は以下の通りである．まず2節でストリームLDMの全体的な構成を述べる．3節で地図データのストリーム化機構の詳細について述べ, 4節でその評価について

†1 名古屋大学大学院情報科学研究科附属組込みシステム研究センター
Center for Embedded Computing Systems, Nagoya University
†2 同志社大学モビリティ研究センター
Mobility Research Center, Doshisha University
†3 日立製作所情報通信システム社
Hitachi, Ltd., Information & Telecommunication Systems Company

述べる．最後に，5節で纏めと今後の課題について述べる．

2. ストリーム LDM

2.1 Local Dynamic Map（LDM）

LDM は，ITS 協調システムで利用することを目的としたデータの集合体である．LDMでは，データの更新頻度に応じて，次のような4階層の階層モデルでデータを管理する．

・第1階層：静的なデータ
　道路形状や交差点等の地図データ
・第2階層：一時的に静的なデータ
　信号や標識の位置等，地図に付随するデータ
・第3階層：一時的に動的なデータ
　交通事故，渋滞/工事，路面状況等
・第4階層：動的なデータ
　車両，歩行者，障害物の位置や速度，信号の状況

LDMでは，車両データのような上位階層の動的なデータも，GPSなどから得られる位置情報をもっている．LDMを利用するアプリケーションは，この位置情報によって，動的なデータを，下位階層の静的な地図データと関連付けることによって有意な情報を得ることができる．

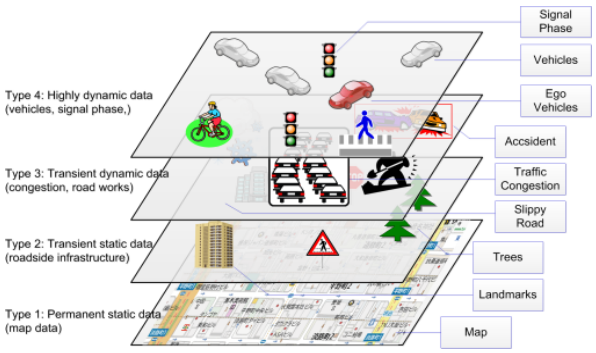

図 1　LDM の階層モデル
Fig. 1　Layers of LDM

2.2 データストリーム管理システム（DSMS）

DSMSとは，時系列に到着するデータをリアルタイムに処理するシステムである．センサや車車間通信で得られるデータは，時々刻々と変化する．このようなデータの処理には，RDBSMSよりDSMSの方が向いている．

DSMS では，入力されたデータを処理する為の演算（オペレータ）が用意されており，それらを組み合わせてクエリを記述する．この点は RDBSMS と同じであるが，DSMS では，クエリは，データ処理の実行前，あらかじめシステムに登録しておき，データが入力される度に登録されたクエリに従って処理を実行し，結果をアプリケーションに通知する．これらの処理はメモリ上で行なわれるため，RDBMS と比較して高速な処理が可能である．

2.3 ストリーム LDM とその課題

我々の提案するストリームLDM は，DSMSとRDBMSを組み合わせて効率的にLDMを実現する技術である．LDMのデータのうち，車両データのような上位階層の動的なデータは，RDBに格納することなく，ストリームデータとして直接DSMSに流して処理を行なう．下位階層の静的な地図データ及びそれに付随するデータは，RDBに格納するが，必要に応じてRDBを検索し，必要な情報を取得する．しかし，ストリームLDM におけるRDBMSとDSMSの分担には課題が残っていた．従来手法では，静的な地図データは，RDBに格納されているため，RDB上でのSQLによる検索，空間演算等を通して動的データとの関連付けを行なう必要がある．このときの演算はRDB上で行なうため，すべてメモリ上で演算を行なうDSMS上での演算と比較してコストが高い．特に，対象となる車両台数が増えるに従って，これらの演算のための遅延時間が増幅され，性能的なボトルネックとなるため，DSMSのメリットが十分生かせないという問題がある．

本研究では，静的な地図データを，自車両位置情報をベースとして，必要な範囲を動的にストリーム化する地図データのストリーム化機構を導入することにより，この問題の解決を図る．図2に，本研究で提案するストリームLDMの構成図を示す．動的な車両データについては，自車センサや車車間通信からデータの到着に従ってストリーム化を行い，DSMSのクエリに車両データストリームとして入力する．静的な地図データについては，地図データのストリーム化機構を通じて，必要な範囲をストリーム化し，DSMSのクエリに周辺地図データストリームとして投入する．本研究では，従来RDB上で行なってきた地図データと動的なデータを関連付けるための空間演算をDSMS上で実現するために，空間演算機能を備えたオペレータである空間演算オペレータを導入する．

上述の通り，ストリームLDMに地図データのストリーム化機構を導入することにより，車両データのような動的なデータに加えて，静的な地図データもDSMSにより統一的に扱えるようになる．これによって，地図データとの演算をDSMS上で高速に実行できるようになる．また，データを利用するクエリの観点から見ると，地図データが格納されているRDBを意識する必要はなくなり，データの位置透過性が高まる，というメリットもある．

地図データをDSMSによって処理する手法は，Traffic Monitoringの分野でも研究されているが[14][15]，これらは，地図データそのものはストリームデータ化するのではなく静的データとして処理している．本研究で提案する方式は，動的なデータと同様，静的な地図データもPUSH型のデータとしてDSMSに投入する，という点に特長がある．我々は，ストリームLDMによる分散処理の研究もすすめているが[16]，分散処理では，DSMSのクエリを複数ノード間で分散実行する．このとき，演算で必要になった時点で地図データを取得するPULL型のデータ入力では，データ取得のた

めの遅延時間が問題になるとともに，地図データベースと接続しているノードでしか地図データを利用できない．本方式では，地図データのストリーム化機構を通じて必要な地図データをあらかじめ投入するため，そのような制限は受けない．分散処理に適した方式である．

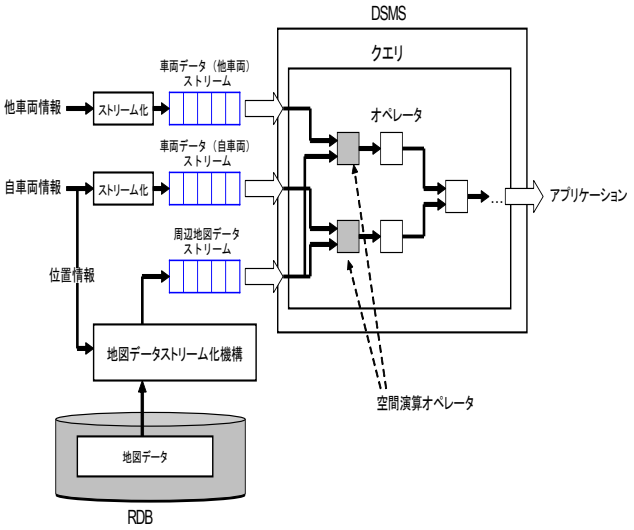

図 2　ストリーム LDM
Fig. 2　Stream-LDM

3. 地図データのストリーム化

3.1 地図データのストリーム化機構

2.3で述べた地図データのストリーム化機構の概要を図3に示す．地図データのキャッシュ機構はキャッシュロジック，地図データキャッシュ，ストリーム出力，というコンポーネントから構成される．地図データのキャッシュ機構では，地図データが格納されているRDBから地図データを入力し，これを内部の地図データキャッシュに一旦格納する．この処理はキャッシュロジックが行なう．キャッシュロジックには，どのタイミングでどのデータをRDBから入力し，キャッシュに格納するか，キャッシュに格納された地図データをどのタイミングでキャッシュから追い出すか，というキャッシュアルゴリズムを実装する．キャッシュに格納された地図データは，ストリーム出力によってストリームデータとして出力される．これらは車両データのような動的データと同様，DSMSの入力となる．

地図データをストリームとして扱う場合，ある時点で必要となる地図データの地図上の範囲はアプリケーションに依存するが，我々がストリームLDMの適用を考えている安全運転支援システムにおいては，必要となる地図データは自車両周辺の限られた範囲である．本研究で示す地図データのストリーム化機構も，これに適したキャッシュアルゴリズムを採用し，自車両周辺の一定範囲の地図データを周辺地図データストリームとして出力する．このために，自車両位置情報がキャッシュヒントとして入力され，地図データをキャッシュする範囲，ストリーム出力する範囲を決めるベースとなる．自車両の走行に伴い，ストリーム出力の対象となる自車両周辺の範囲もシフトする．このとき，新たに自車両周辺範囲に加わった地図データのみをストリームとして出力する．一方，自車両周辺範囲から外れた地図データについては，そのストリームを打ち消すための削除ストリームを出力する．削除ストリームとは，打ち消したいデータのヘッダに削除フラグを設定して，それをストリームとして出力したものである．後述するように，通常，ストリームとして出力された地図データは，クエリでの処理に従って，空間演算オペレータのウィンドウに格納され，ウィンドウの検索を通して他のデータと関連付けられる．よって，地図データのストリーム化機構が一度に出力するストリームは，既に出力してクエリに供給されているストリームとの差分であればよい．この仕組みによって，DSMS上に重複したデータが流れるのを避け，効率的なストリーム処理を可能としている．

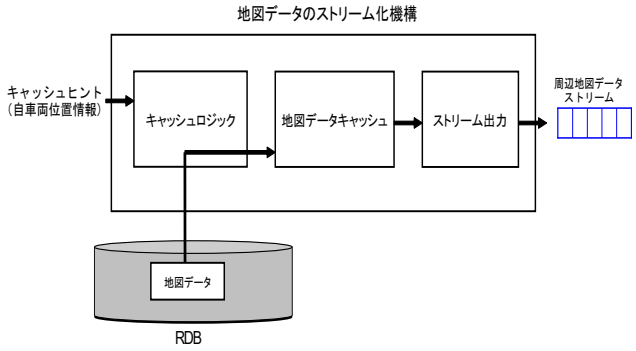

図 3　地図データのストリーム化機構
Fig. 3　Map Data Processing Mechanism

3.2 地図データキャッシュ

地図データキャッシュは，Map と Cache より構成される．図 4 にその図を示す．キャッシュする地図データの実体は Cache に格納されるが，地図上のどの範囲のデータが Cache に格納されているか，という情報は Map が保持する．Map は，地図上の一定範囲の空間をメッシュという領域に分割し，そのメッシュ単位で情報を保持する．Map はそのメッシュ情報の集合である．各メッシュ情報は，そのメッシュ領域に存在する地図データのキーの集合（例えば道路データの場合，そのメッシュ領域に存在する道路データの集合．道路は複数のメッシュ領域に跨ることもあるが，その場合は対応する各メッシュ情報にその道路データのキーが登録される．）及びその領域の地図データがキャッシュされているか否かという情報（CacheIn または CacheOut）をキャッシュの状態フラグとしてもつ．Map の情報はシステム初期化時に RDB に格納された地図データよりあらかじめ作成

しておく．地図上のどの範囲を Map として作成するか，メッシュ領域の単位当たり大きさについては地図データのストリーム化機構のパラメータにより設定できる．各メッシュ領域に存在する地図データのキーをあらかじめ求めておくことにより，実行時に RDB への高速な検索が可能となる．

地図データキャッシュにおける地図データのキャッシュへの出し入れは上記 Map のメッシュ単位で行う．あるメッシュ領域の地図データをキャッシュする場合，そのメッシュ情報のキー集合から，それら各キーに対応する地図データを RDB より取得する．取得した地図データは Cache に格納し，メッシュ情報のキャッシュ状態フラグを更新する（CacheIn）．Cache の実体はハッシュテーブルで，キャッシュする地図データのキーによりキャッシュアドレスを特定し，該当するエントリにデータを格納する．道路データのような地図データでは，1 つのデータが複数のメッシュ領域に跨ることがありうるが，このような場合は，キャッシュに重複して登録はしない．逆に，あるメッシュ領域の地図データをキャッシュから追い出す場合，そのメッシュ情報の状態フラグを更新する（CacheOut）．キャッシュから追い出すべきデータは，キャッシュ格納時と同様にメッシュ情報のキー集合から特定する．但し，1 つのデータが複数のメッシュ領域に跨る場合，すべてのメッシュ領域が CacheOut の状態にならない限り，そのデータはキャッシュから削除しない．

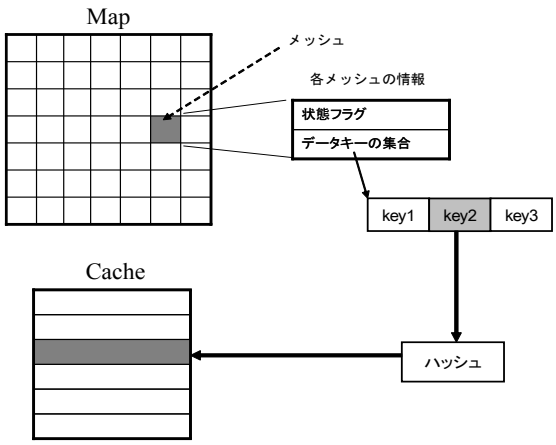

図 4　地図データキャッシュ
Fig. 4　Cache of Map Data

3.3 キャッシュアルゴリズム

前述の通り，本研究では，衝突検知のような安全運転支援システムに適したキャッシュアルゴリズムを採用する．車両のような移動体の位置を基準として地図データをキャッシュする方式としては先行研究[17]があり，LRU のような一般的なキャッシュアルゴリズムに対する有効性が示されている．但し，先行研究では，特定の目的地，経路を想定するが，本研究では，目的地の予測がつかないより一般のケースを考える．

キャッシュ範囲は，自車両が存在するメッシュ位置を中心とした地図上の一定範囲とする．キャッシュの更新タイミングは，自車両の位置があるメッシュから別のメッシュに移動した場合である．図 4 に概念図を示す．この図は，自車両が最初に Map 上の①に相当する位置におり，④に相当する位置まで走行するケースにおいて，キャッシュされる地図データの Map 上の範囲がどのように変わるかを示している．本キャッシュアルゴリズムでは，自車両の存在するメッシュ位置を中心として，その一定範囲にあるメッシュに存在する地図データをキャッシュする．この図では，その範囲を 5×5 メッシュとしている．最初に自車両が①の位置にいるときのキャッシュ範囲が①の位置を中心とする点線で囲んだ範囲．自車両の走行に伴い，この点線で囲んだ範囲，すなわちキャッシュ範囲もシフトする．太線で囲んだ範囲が最新の自車両位置が④にある場合のキャッシュ範囲である．キャッシュの更新は，自車両の初期位置が特定されたタイミング，その後は，自車両の位置があるメッシュから別のメッシュに移動したタイミング，で行なう．このとき，キャッシュ範囲にある地図データを 3.2 で述べたように，そのキャッシュ範囲に相当する Map のメッシュ情報を参照して，RDB より取得し，キャッシュの更新処理を行なう．図 5 でも示した通り，キャッシュ範囲は，前のキャッシュ範囲と重なる部分をもちながらシフトしていくので，キャッシュの更新処理が行なわれるのは，前回キャッシュ範囲との差分に相当する部分である．つまり，新たにキャッシュ範囲に入った地図データが CacheIn の対象となり，キャッシュ範囲から外れたデータが CacheOut の対象となる．

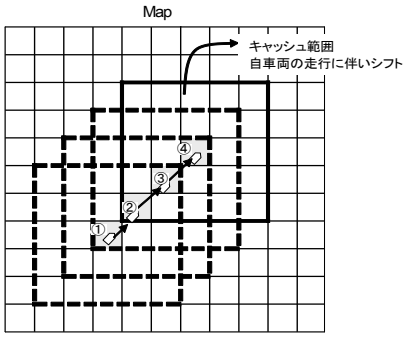

図 5　キャッシュアルゴリズム
Fig. 5　Cache Algorithm

このように，本キャッシュアルゴリズムでは，メッシュ単位でキャッシュ範囲をシフトさせることにより，漸次的にキャッシュを更新する手法をとっており，リアルタイムな処理に適したアルゴリズムとなっている．メッシュサイズの調整によっても，1 回当たりのキャッシュ更新時間を極力減らすことが可能である．

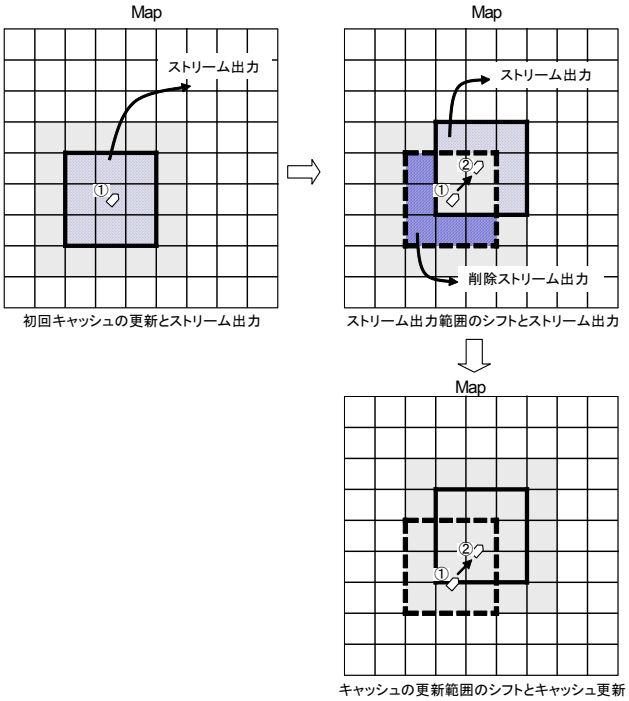

図 6　ストリーム出力
Fig. 6　Stream Output

3.4 ストリーム出力

地図情報キャッシュに格納された地図データは，ストリーム出力によってストリームデータとして出力する．このとき出力する地図上の範囲であるストリーム出力範囲も，キャッシュ範囲と同様，自車両の走行に伴いシフトする．ストリーム出力するタイミングも，キャッシュと同様，自車両の位置があるメッシュから別のメッシュに移動したタイミングで行なう．

自車両の走行に伴い，どのようにストリームを出力するかについて図 6 を用いて説明する．自車両が図 5 と同様，Map 上を左下から右上に向かって走行するケースを考える．キャッシュ範囲は灰色，現在のストリーム出力範囲は太線，過去のストリーム出力範囲を点線で示す．初めに，自車両が初期位置①にあるとき，自車両のキャッシュ範囲の地図データが地図データキャッシュに格納される．このとき，キャッシュ範囲はストリーム出力範囲の周囲 1 メッシュ分余分にとっておく．キャッシュされたデータのうち，ストリーム出力範囲に存在する地図データをキャッシュから取り出し，ストリームとして出力する．次に，自車両のメッシュ位置が①から②に移動したとする．このメッシュ間を移動したタイミングで，ストリーム出力範囲もシフトし，ストリーム出力を行なう．このとき，キャッシュ範囲をストリーム出力範囲の周囲 1 メッシュ分余分にとっているため，自車両がどの方向に移動しても，ストリーム出力範囲のデータはキャッシュに存在することが保証される．すなわち，キャッシュによって地図データの先読み機能を実現している．これによって，ストリーム出力範囲のシフトに伴い，即座にキャッシュからストリームを出力することを可能としている．キャッシュの更新と同様，ストリーム出力についても，1 回のストリーム出力の対象とするデータは，前回ストリーム出力範囲との差分に相当する部分である．新たにストリーム出力範囲に入ったデータをストリームとして出力する．一方，ストリーム出力範囲から外れたデータについては，DSMS 上からストリームを打ち消すための削除ストリームを出力する．最後に，3.3 で述べたように，キャッシュの更新範囲をシフトし，キャッシュの更新を行なう．以降，このサイクルを自車両のメッシュ間の移動に伴い繰り返す．

3.5 空間演算

上述の通り，地図データのストリーム化機構によりストリーム化された地図データは，DSMSのクエリの入力となる．DSMSのクエリでは，これら地図データと車両データのような動的データを，それらのデータが有する位置情報によって関連付ける演算が必要となる．このような演算は空間演算と呼ばれるが，本研究では，このような演算を扱うためにSpatial Joinという空間演算オペレータをDSMSに導入する．

Spatial Joinは，位置情報をもつ2つのストリームを入力し，位置情報がマッチした場合，2つのデータを結合し，結果ストリームとして出力する．一方の入力は，車両データのような動的データ，一方の入力は地図データを想定している．前者を入力1，後者を入力2とすると，入力1のデータは逐次処理されるが，入力2のデータはオペレータのウィンドウに格納される．入力1にデータが入力されたとき，このウィンドウ内のデータとの位置マッチングを行い，マッチしたデータと結合したデータを生成する．このような演算は，具体的には，車両データと道路データを入力し，車両とその車両が走行している道路を結合した車両走行データを生成したい場合に適用できる．

Spatial Join の動作詳細について，図 7 を用いて説明する．

①入力 2 のデータ入力

入力 2 のストリームキューからデータを入力し，ウィンドウに格納する．但し，タプルが削除タプルの場合は，ウィンドウから対応するタプルを検索し，そのタプルをウィンドウから削除する．この処理を入力 2 のキューにデータがある限り行う．

②入力 1 のデータ入力

入力 1 のストリームキューからデータを入力する．

③マッチング及び結合処理の実行

入力 1 から入力したデータの位置座標と最近傍にある位置座標をもつデータをウィンドウから検索し，両者を結合したデータを生成する．

④結果タプルの出力

③で生成したデータを出力用ストリームキューに出力する．③でタプルが生成されなかった場合は何も出力

しない．入力 1 のキューにデータがある場合は，②に戻る．ない場合は終了．

上述空間演算オペレータは，地図データのキャッシュ機構と連動することによって，現在の演算に必要な範囲の地図データのみをウィンドウに保持する．これによって，位置マッチングに要する検索時間が抑えられ，すべてをメモリ上で実行するため，高速な演算を実現できる．

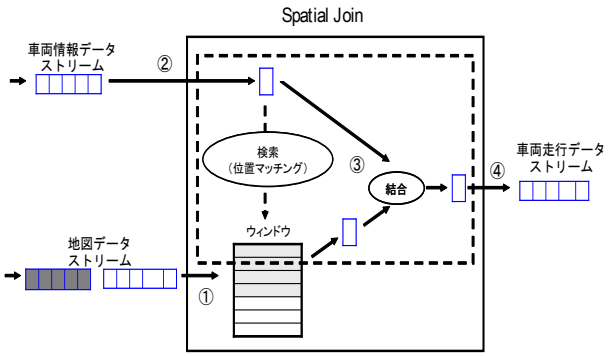

図 7　空間演算オペレータ

Fig. 7　Spatial Operator

4. 評価

4.1 評価方法

上述地図データのストリーム化機構を導入したストリームLDMを実装し，評価を行なった．本評価の目的は，地図データのキャッシュ機構とそれと連動する空間演算オペレータの基本的な性能を明らかにし，地図データをDSMS上で処理することの有効性を確認することにある．従来手法に対する効果を明確にするために車両データのような動的データと地図データとの最も基本的な演算である位置マッチングと結合演算の性能を評価した（図8）．これは，時系列に入力される車両データに対して，それと位置がマッチする道路を地図データベースから取得し，両者を結合した車両走行データを生成するものである．これをストリームLDMにより実現した．これは，地図データのストリーム化機構と3.5で述べた空間演算オペレータ（Spatial Join）から成るクエリをDSMSで実行することにより可能である．一方，従来手法との比較評価のため，これをRDBMSにより実現した場合の性能も取得した．RDBMSとしては，PostgresSQLを使用した．PostgresSQLは，位置マッチングに相当する空間演算機能を備えており，この処理をSQLによって実行できる．

車両データは，車両の識別子，現在の車両の走行速度等，位置座標等の属性，道路データは，道路の識別子，道路の長さ，位置座標等の属性をもつが，本評価で使用するのは，各々の識別子，位置座標情報のみである．車両データは，PreScan[18]を用いて車両走行のシミュレーションデータを作成し，あらかじめCSVファイルより出力したものを使用した．このCSVファイルから，車両データを読み込むことによって車両データの入力を実現した．CSVファイル内の車両データにはタイムスタンプにより昇順に並んでおり，一度の入力では，同一タイムスタンプをもつデータを入力する．車両データとしては，自車両データと他車両データがある．あるタイムスタンプをもつデータは，すなわち，同一時刻のデータは，自車両データは1つしかないが，他車両データは他車両の台数分ある．

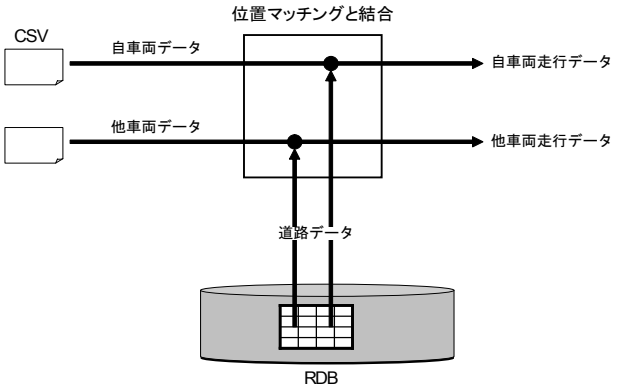

図 8　性能評価の対象処理

Fig. 8　The processing that is targeted for evaluation

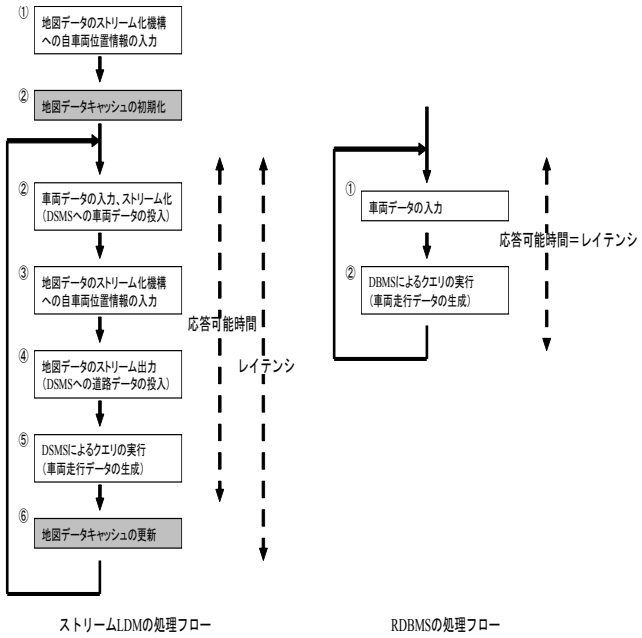

図 9　処理フロー

Fig. 9　Processing Flow

道路データは，地図データとしてRDBに格納する．地図データのソースとしては，OpenStreetMap[19]を利用した．OpenStreatMapの道路データを独自変換ツールを使って我々の定義したフォーマットに変換し，PostgreSQLのデータベース上に登録した．この地図データは，地図データのキャッシュ機構により必要に応じてストリーム化され，DSMSに投入される．

本評価において，ストリームLDMを使用した場合とRDBMSを使用した場合の各々の処理フローを図9に示す．いずれの処理においても，車両データの入力をトリガーとして処理が続行される．前述の通り，一度に入力する車両データは同一タイムスタンプをもつデータであり，その単位で，すなわち時刻毎にクエリ実行等含めた一連の処理を行なう．その処理が終わると，次のタイムスタンプの車両データを入力し，それをすべての車両データを読み終わるまで繰り返す．本評価では，車両データの入力を開始して，クエリの実行が完了するまでの時間（すなわちアプリケーションに結果が応答可能となるまでの時間）を応答可能時間，一連の処理が完了し，次の入力が可能となるまでの時間をレイテンシとして測定した．

以下に本評価の評価環境を示す．

(1) CPU Intel Core2 Duo 3.0GHz
(2) メモリ 1024MB
(3) OS Linux2.6
(4) RDB PostgreSQL 8.4.8

4.2 評価結果

ストリーム LDM において，地図データキャッシュ機構から，自車両周辺のどの範囲のデータをストリームとして出力するか，すなわちストリーム出力範囲は，地図データキャッシュ機構のパラメータとして指定できる．その範囲を 25m×25m，50m×50m，100m×100m としたときの測定結果を図 10 に示す．その範囲に対応するメッシュは，図 6 と同様 5×5 メッシュとしている．車両データは，自車両と他車両 3 台の走行をシミュレーションしたデータを使用した．図 9 に示したように，処理の各繰り返しにおいてレイテンシと応答可能時間を測定し，その平均値，最大値を示した．平均レイテンシと最大レイテンシに大きな差があるが，これは，地図データキャッシュの更新処理に起因する．3.3 で述べた通り，地図データキャッシュの更新は毎回発生する訳ではなく，自車両のメッシュ位置に変化が生じた場合に発生する．この発生頻度は小さいため，平均レイテンシには大きな影響を与えないが，最大レイテンシは，このキャッシュ更新時間によって決定付けられる．ストリーム出力範囲の大きさが増えれば，更新時にキャッシュするデータ量も増えるため，キャッシュ更新時間も増える．そのため，最大レイテンシは，ストリーム出力範囲の大きさに従って増加する．これに対して，応答可能時間は，クエリの実行が完了するまでの時間であり，キャッシュ更新時間を含まない．3.4 で述べた通り，地図データのストリーム化機構は，地図データキャッシュにデータを先読みしておくことにより，即座に必要なデータをクエリに供給できる．従って，応答可能時間はほぼクエリの実行時間で決まり，平均レイテンシと最大レイテンシに大きな差はなく安定した値となっている．

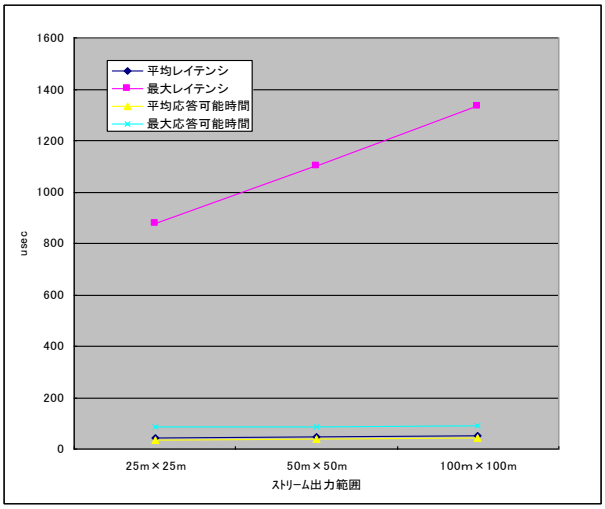

図 10 ストリーム LDM の性能①
Fig. 10 Performance of Stream LDM ①

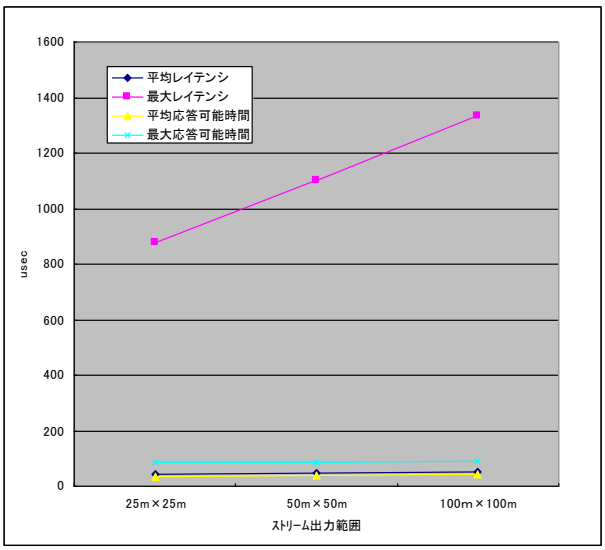

図 11 ストリーム LDM の性能②
Fig. 11 Performace of Stream LDM ②

次に，ストリーム出力範囲を一定（100m×100m）とし，他車両の車両台数を 3 台から次第に増加させていったときの測定結果を図 11 に示す（車両台数が 3 台のときのストリーム LDM の性能は図 10 の 100m×100m のときの測定と同一）．また RDBMS により実行したときの結果とストリーム LDM との比較を図 12 に示す．ストリーム LDM では，平均レイテンシは，車両台数に比例的に増加する．これは，DSMS での空間演算を各車両データに対して実行するためである．これに対して，最大レイテンシは，車両台数にあまり影響を受けない．最大レイテンシを決める主要因のキャッシュの更新処理は，車両台数の影響を受けないためである．RDBMS では，平均レイテンシ，最大レイテンシとも車両台数に比例的に増加する．RDBMS では，入力された各車両に対して SQL により空間演算を実行する．各車両毎に空間演算を実行する点はストリーム LDM と同様であるが，空間演算を DSMS 上で実行した場合と RDB 上で実

行した場合の演算の単価の違いにより，車両台数の増加に従ってストリーム LDM との性能差は著しくなる．

この結果から，従来 DSMS により行なっていた車両データと地図データとの演算をストリーム LDM により行なうことにより，高速化が図れることが確認できた．特に，従来方式の課題であった車両台数が増えたときの遅延時間の問題は著しく改善されることが実証できた．

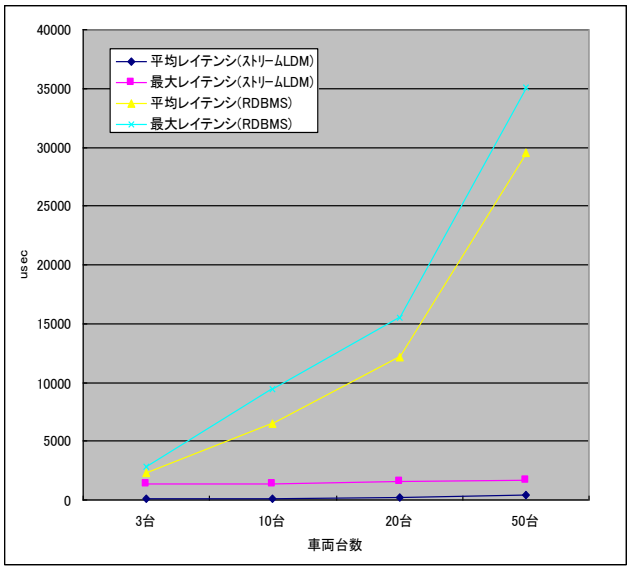

図 12　ストリーム LDM と RDBMS の性能比較
Fig. 12　Performance comparison between Stream-LDM and RDBMS

5. 纏めと今後の課題

本論文では，地図データのストリーム化機構を導入したストリームLDMを提案した．地図データのストリーム化機構では, 従来RDBMSによって処理してきた地図データをキャッシュメカニズムを通して必要なタイミングで必要な範囲をストリーム化することにより，DSMS上での地図データの効率的な演算を実現する．本方式を実装した評価結果より，地図データと車両データの基本的な演算性能において, 従来方式と比較して著しい改善が図れることを示した．

今後の課題としては，各種アプリケーションにおいて有効性を検証し，それらの特性に応じた拡張を検討する予定である．

謝辞　本研究の一部は，SCOPE（12180615）と科研費（25240007）の助成を受けたものである．

参考文献

1) 先進安全自動車(ASV) 推進計画報告書(オンライン), http://www.mlit.go.jp/jidosha/anzen/01asv/resourse/data/asv4pamphlet_seika.pdf (2011).
2) 小林雅文，大田利文，鎌田邦廣, ”DSSS の実用化に向けた研究開発”，自動車技術，Vol.64, No.9, p.43-48(2010).
3) Google’s Self-Driving Car, http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works
4) Toyota’s Semi-Autonomous Car, http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/toyota-semi-autonomous-lexus-car-will-keep-you-safe
5) ETSI TR 102 863 (V1.1.1), Intelligent Transport Sys-tems (ITS): Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM) Rationale for and Guidance on Standardization, (2011).
6) SAFESPOT Core Architecture: LDM API and Usage Reference, D.7.3.1 Annex 2, (2010).
7) SQLite, http://www.sqlite.org/.
8) PostgreSQL, http://www.postgresql.org/.
9) Babcock, B., Babu, S., Datar, M., Motwani,R. and Widom, J. : Models and issues in data stream systems, Proceedings of the twenty-_rst ACM SIGMOD-SIGACT-SIGART symposium on Princi-ples of database systems, pp.1-16 (2002).
10) 佐藤健哉，山田真大，島田秀樹，金榮柱，本田晋也，高田広章: Local Dynamic Map (LDM) の設計と評価，問題点とその対策，第 10 回 ITS シンポジウム 2011 予稿集，pp,85-90, (2011).
11) 山口晃広，本田晋也，佐藤健哉，高田広章: 車載システム向けデータストリーム管理システムにおけるクエリ自動構築手法，第 11 回情報科学技術フォーラム講演論文集，Vol.2, pp.7-14, (2012).
12) 山田真大，鎌田浩典，手嶋茂晴，高田広章，佐藤健哉：データストリーム管理機構を利用した車載データ統合モデルの提案と評価，　自動車技術会論文集，　Vol.41, No.2,pp.419-424, (2010).
13) 勝沼聡，山口晃広，熊谷康太，本田晋也，佐藤健哉，高田広章: 車載組込みシステム向けデータストリーム管理システムの開発，電子情報通信学会論文誌 Vol. J95-D,No. pp.-, (2012).
14) S. Geisler, C. Quix, and S. Schiffer. A Data Stream-based Evaluation Framework for Traffic Information Systems. In IWGS, pp. 11-18, 2010.
15) K.Patroumpas and T.Sellis. Event processing and real-time monitoring over streaming traffic data. W2GIS'12 Proceedings of the 11th international conference on Web and Wireless Geographical Information Systems, p116-133,2012
16) 佐藤健哉，Cloudia:Car-to-X データ統合プラットフォーム～安全運転支援のための分散ストリーム処理の実現～，第 3 回名古屋大学組込みシステム研究センターシンポジウム講演予稿集 pp.57-70, 2012/10.
17) 佐藤健哉，最所圭三，福田晃：放送により配信される位置依存情報のキャッシュ方式，情報処理学会論文集，Vol.41, No9,pp.2434-2444, (2000).
18) PreScan,http://www.tass-safe.jp/prescan/index.html.
19) OpenStreetMap,http://www.openstreetmap.org/.