

ストリームプロセッシングによる車載統合制御システムのための分散型センサデータ処理機構の構築

山田 真大[†] 鎌田 浩典^{†1} 佐藤 健哉[‡] 手嶋 茂晴[†] 高田 広章[†]

[†] 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

[‡] 同志社大学理工学部情報システムデザイン学科

要約 安全運転支援のため車両に搭載されたセンサからの情報に基づき障害物や前方車との衝突危険性を判断し、ドライバへの警告や自動で危険を回避する車載統合制御システムが登場している。特性の異なる複数のセンサデータを融合して利用することで、対象物認識確率の向上が可能である。センサデータを利用する複数のアプリケーションが個別にセンサデータを処理することは可能であるが、対象物の認識などアプリケーションごとに重複する処理も多い。本研究では、複数センサから得られたデータを直接それぞれのアプリケーションが処理するのではなく、走行環境の理解のためにデータの抽象度を向上させることで、アプリケーションの処理を軽減しデータ提供を簡単にするための分散型センサデータ処理機構の構築を行う。実際に、車速センサや車載カメラなどのセンサデータを利用した衝突予防システムを構築し、その実現可能性を探る。

Distributed Sensor Data Processing System with Streaming Processing Model for Vehicle Integrated Control System

Masahiro YAMADA[†] Hironori KAMADA^{†1} Kenya SATO[‡]

Shigeharu TESHIMA[†] Hiroaki TAKADA[†]

[†] Center for Embedded Computing Systems, Nagoya University

[‡] Department of Information Systems Design, Doshisha University

Abstract The vehicle integrated control system recently appears to provide safe driving and danger warning to drivers. The system is capable of determining collision risk with a car and obstacles based on information from on-board sensors. By using a method for conveniently integrating data provided by various kinds of sensors, it is possible to increase probability and robustness for target recognition. In vehicle application implementation, there are duplicate functions in each application for processing sensor data such as statistical calculations, pattern matching algorithms, and so on, due to the reason that individual applications directly receive unprocessed data from each sensor. In this research, we develop the distributed sensor data processing platform for vehicle environment understanding to abstract sensor data and consolidate duplicated functions in applications. In addition, we make a feasibility study to implement the collision avoidance system on the sensor data processing platform with vehicle speed sensor, vision camera, and so on.

1 はじめに

近年、安全運転支援のため車両に搭載されたセンサからの情報に基づき障害物や前方車との衝突危険性を判断し、ドライバへの警告や自動で危険を回避する車載統合制御システムが登場している^{1, 2)}。たとえば、ミリ波とステレオカメラ、ドライバモニタの情報を統合し判断することで、操舵回避に適切なステアリング比にし、サスペンシ

ョンを制御し、ブレーキアシストにより回避操作の支援を行い、衝突が避けられない状況では介入ブレーキを作動させることで衝突衝撃を緩和し被害を軽減するシステムがある³⁾。一般に利用されるセンサは複数の種類があり、性能や価格、利用環境の特性などに違いがあるため、複数のセンサ情報を融合して利用することで、対象物認識の確率を向上させる技術が利用される^{4, 5)}。センサ情報を利用するアプリケーションに関して、衝突予防

¹ 現在 (株) オクトパス, OCTOPATH Corporation

システム以外にも、車両追従システムや車線逸脱防止システム、駐車支援システムなどがある。それぞれのアプリケーションが個別にセンサからの情報をそれぞれ処理することは可能であるが、対象物の認識など重複する機能も多い。また、複数のセンサ情報に対して複数のアプリケーションが同時に処理を行うにはシステムが複雑になり1か所に大きな計算資源が必要となる。

一方、センサ情報を処理する仕組みとして、環境モニタリングへの適応を目的として、STREAM⁶⁾、TelegraphCQ⁷⁾、Aurora⁸⁾、NiagaraCQ⁹⁾、COUGAR¹⁰⁾など、センサから時系列に生成されるストリームデータを処理するためのシステムが複数検討されている。本研究では、ストリームプロセッシングの手法を利用し、センサから得られる情報の抽象度を向上させることで走行環境を統一的に理解し、複数のアプリケーションから利用可能とする車載統合制御システムのための分散型センサデータ処理機構構築を行う。

2 統合データ表現

2.1 取得情報

車両に搭載された各種センサや通信手段から得られる情報により、自車周辺にある他の車両や歩行者、障害物などの周辺物体、道路やレーンの走行路を認識することを走行環境理解と呼ぶ。衝突の危険性を判断するためには、対象の物体までの距離と相対速度から算出される衝突までの時間と、物体の横方向の位置情報が必要となる。一般にミリ波やレーザを利用したレーダは対象の物体までの距離や相対速度の検出に優れており、車載ステレオカメラは、物体の形状や横方向の位置情報に関しての検出精度が高い。たとえば、対象の物体に対して、距離、横方向の位置情報、形状、速度を抽出するために特徴の異なる複数のセンサを組み合わせるセンサフュージョン手法が利用される¹¹⁾。本研究においては、CCDやCMOSなどの車載カメラである画像センサ、ミリ波やレーザを利用したレーダによる車間距離センサ、自車に搭載されたGPSや速度・加速度センサ、地図情報、車車間・路車間通信により得られた他車の走行状況などに関する情報の利用を想定する。

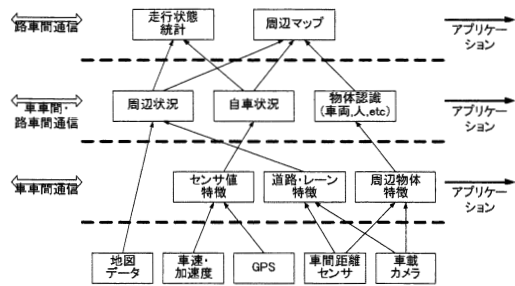


Fig. 1 車載統合制御システムのためのセンサデータ統合例

2.2 データ構成

車載統合制御システムのためのセンサデータ統合例を Fig. 1 に示す。この場合は4つの階層に分けてデータの処理を行う。ここでは階層を上げることを抽象度の向上と呼ぶ。センサから得られたそのままのデータを処理し、抽象度を向上させることで、走行環境の理解を行うモデル構成とする。たとえば、車載カメラから得られた画像データに加えてミリ波レーダから得られた車間距離情報を合わせて判断し周辺物体の特徴を抽出する。この情報をさらに抽象度を向上させ、パターンマッチングを行い車両や人と認識する。一方で、地図データやセンサから得られた周辺状況と合成することで、どの位置に車両や人が存在するか周辺マップ情報を作成することが可能となる。最終的に自車の走行状態を合わせて判断し、衝突の可能性を検知することができる。このようにデータの抽象度を向上させることで、それぞれの情報がより扱いやすい情報となり、アプリケーションに対してのデータ提供も簡単になる。また、他の車両が保持している情報や、インフラの情報を車車間通信や路車間通信を利用し、センサデータそのものではなく、抽象度を向上させた情報を交換することで、通信や計算の負荷を削減することが可能となる。

2.3 車載統合制御システム

車載統合制御システムにおいて利用される場合は複数あり、たとえば、前方の映像を表示するだけのシステムから、この映像において車両や歩行者の表示を強調するシステム、自車の走行状態を統計情報として記録するシステム、周辺の車両や

歩行者の位置情報を提供するシステム、衝突の危険性を判断しブレーキをかけたたり、ステアリングをきったりする衝突予防システムの車載統合制御システムなどが考えられる。それぞれのシステムが必要とするデータは異なり、すべてはセンサから得られるデータをもとに処理した情報が基本となるが、それぞれのシステムが同様の処理を独自で行うことには無駄がある。本管理機構においては内部でデータ処理を行い抽象度を向上させた情報を必要なシステムに提供することが可能である。

3 分散型センサデータ処理機構

3.1 ストリームプロセッシング

センサデータベースでは、センサごとのデータの独立性、センサデータの融合や抽象化の必要性、厳密なトランザクションが不要となる。前述したように主に環境モニタリングを目的としたセンサ情報処理のためのセンサデータベースが研究されている。従来のデータベースにおいては、入力から得られたデータを保存し、クエリやトランザクションに応じてデータの処理を行うため、保持しているデータは唯一であり非常に重要な役割を担うが、センサデータの場合、最新のデータの入手が容易であり、それぞれのデータに対して時間的同期を考慮した処理が必要となる。また、このときの実時間性が重要となる場合も多い。そのため、センサデータをストリームデータとしての処理が効率が良い。時間とともに変化する事象に関する最新のセンサデータをストリームとして扱い、本論文ではこのセンサデータの処理をストリームプロセッシングと呼ぶ。Fig. 2にセンサデータのストリームプロセッシングによるモデルを示す。入力データストリームを受け、逐次的にデータ処理ボックスで処理を行いながら、最終的に処理結果を出力する。必要であれば、処理結果としての履歴データを保持することも行う。

3.2 ストリーム演算

センサデータ処理機構で扱うデータは、ストリームデータとしてモデル化される。Fig. 3にストリーム演算のモデルを示す。それぞれのタプルは同一のスキーマを持つ属性値の集合からなり、ストリームデータは時系列のタプルの集合で構成され、ストリーム演算が実施される。一般にストリーム演算は一つ以上のストリームデータを入力として受け取

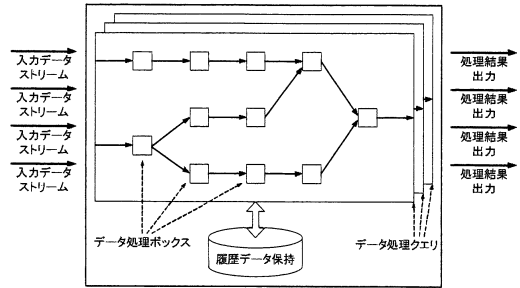


Fig. 2 ストリームプロセッシングモデル

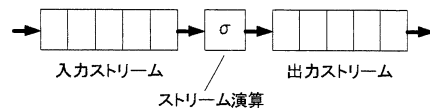


Fig. 3 ストリームデータ

りストリームデータに対する演算を行う。演算結果は再びストリームデータとして出力ストリームに流される。ストリーム演算は、一般にFilter, Join, Aggregateなどの要素で構成される。Aggregateは入力ストリームに複数のタプルが到着した場合に複数のタプルを引数にとり指定された関数を実行し出力を行う要素、Joinは2つの同一スキーマを持つ入力ストリームを受け取り一つのストリームとして出力する要素、Filterは条件式を設定し、入力ストリームのそれぞれのタプルに対しその条件が成立する場合だけ、出力ストリームへ流す要素である。

3.3 クエリ

センサデータ処理機構におけるクエリは、一般に継続型クエリである場合が多い。継続型クエリは、どこからストリームデータを受け取り、どのようなストリーム演算を行い、どこへストリームデータを出力するかという情報をデータストリーム処理システムに一度登録することで、あとはそのクエリ情報に従いストリーム演算によりデータ処理を続ける。データ駆動型で、要求のたびクエリを発行しなくてよいため、その都度クエリを出す場合と比較してオーバーヘッドを削減できる。クエリの記述はXML形式で記述するものやSQL Likeで記

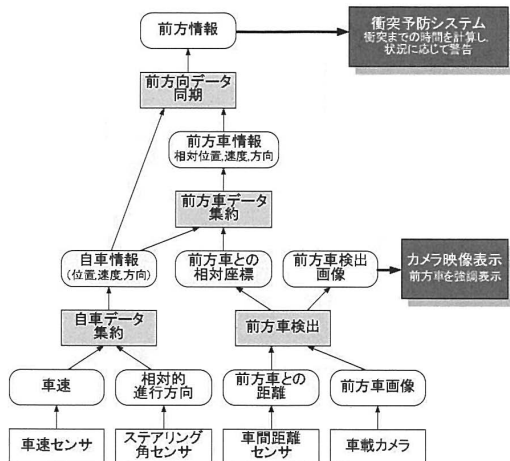


Fig. 4 実装システムのデータストリーム

述するものがある。

4 実装システム

4.1 Borealis

分散型センサデータ処理機構においては、クエリを複数のハードウェアやプラットフォームに分散した状態で実行を行う。一般にノードと呼ばれる単位のプロセスを所有し、ノードごとに固有のクエリ情報を保持し、その情報に従いストリーム演算を実行するものである。本研究では、ブランダイス大学、ブラウン大学、マサチューセッツ工科大学で開発された分散型データストリーム処理機構を実現したシステムである Borealis¹²⁾ を利用した。Borealis は Linux で動作し、アプリケーションは C++ 言語で記述する。ソースコードが公開されており、ユーザ定義演算というストリーム演算の拡張を自由に行うことが可能で、車両固有のストリーム処理を行うのに適している。

4.2 アプリケーション実装

分散したストリームデータ処理を行なうアプリケーションとして次の 2 つを設計した。

(1) 衝突予防システム

(2) カメラ映像表示

(1) は、周辺データ層にアクセスし、前方車と自車のデータから衝突の危険性を判断するアプリケー

ションである。(2) は、前方車検出を行ったあとのカメラ映像を表示するアプリケーションである。実装システムの構成を Fig. 4 に示す。設定したセンサは、車載カメラ、車間距離センサ、ステアリング角センサ、車速センサである。車載カメラから得られた前方車画像¹⁴⁾、および、車間距離センサより得られた前方車との距離情報から、前方車画像のなかから前方車を検出する。一方、車速センサから得られた車速データ、ステアリング角センサから得られた操舵角から自車データを集約する。このような構成を採ることで、情報の取得頻度や情報の抽象度に応じた柔軟なアクセスが可能となる。前方車検出された画像をもとの車載カメラの映像に合成し、カメラ映像表示のアプリケーションを実現する。また、自車データ、前方車データとの相対座標から前方車情報を計算し、自車データと同期を取り前方情報とする。この情報をもとに衝突までの時間を計算し状況に応じて警告を行う衝突予防システムのアプリケーションの実現を行なう。ここでは、定義したこれらの階層間のデータの変換処理は Borealis のストリーム演算を使用することで実現を行った。

4.3 ユーザ定義演算

階層化を行ううえで各データ間の変換においてストリームデータ処理が必要となるが、Borealis に標準で用意されているストリーム演算に加えて、前方車認識などストリームデータに対する固有の処理は別途追加する必要がある。今回、OpenCV の認識機能をユーザ定義演算として実装することで前方車認識機能の追加を行った。その前方車検出も含め、以下のユーザ定義演算を追加した。

(1) 前方車検出

(2) 自車データ集約

(3) 前方車データ集約

(4) 前方向データ同期

(1) は前方車画像と前方車との距離を取得し、前方車検出を行い、前方車と自車の位置偏差と前方車検出画像を出力するストリーム演算である。(2) は車速、操舵角を入力として受け取り、自車の座標、車速、方向を計算するストリーム演算である。(3) は (1)、(2) の出力ストリームを受け取り、前方車

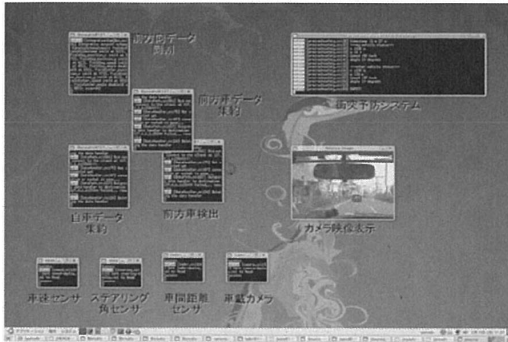


Fig. 5 評価環境実行画面

の座標，車速，方向を計算するストリーム演算である。(4)は(2)，(3)の出力ストリームを受け取り，前方方向に対する情報を集約するストリーム演算である。

5 評価

5.1 評価環境

分散型センサデータ処理機構の評価環境実行時の画面を Fig. 5 に示すとともに，評価環境を Table 1 に示す。

Table 1 評価環境

項目	仕様
CPU	PentiumD CoreDuo 2.8GHz
Memory	2GByte
OS	Ubuntu (Linux) 8.04

5.2 評価結果

ストリーミング処理を行なう際の各センサデータのサイズを Table 2 に示す。前方車検出後のカメラ画像は入力カメラ画像のデータと同じである。入力データ生成周期，および，それぞれの計算を行なう際の同期周期を Table 3 に示す。

車間距離センサ→前方車検出通信時間，前方車検出を行なうための処理時間，前方車検出→前方車データ集約通信，前方車データ集約処理時間，前方車データ集約→前方向データ同期通信時間，前方向データ同期処理時間，前方向データ同期→衝突予防システム通信時間を計測した。通信時間とは，データ入力からノードまでの通信時間と Borealis

Table 2 データサイズ

データ	サイズ
車速センサ	8Byte
ステアリング角センサ	8Byte
車間距離センサ	8Byte
車載カメラ	230,400Byte/frame
前方車情報	44Byte

Table 3 データ入力および計算周期

データ	周期
車速センサ	11ms
ステアリング角センサ	11ms
自車データ集約	11ms

データ	周期
車間距離センサ	11ms
車載カメラ	33ms
前方車検出	33ms

データ	周期
自車データ集約	11ms
前方車検出	33ms
前方車データ集約	33ms

データ	周期
自車データ集約	11ms
前方向データ同期	33ms
前方データ同期	33ms

オーバーヘッドを加算したものである。評価結果を Table 4 に示す。センサ入力からカメラ映像表示アプリケーションまでの通信時間および処理時間の合計が 45.5ms，センサ入力から衝突予防システムアプリケーションまでが 63.2ms であった。

6 関連研究

ロボットが利用者に対してサービスを提供するため，利用者の位置や状況，行動の意味に関する環境情報を構造化し理解するための研究が行われている [10]。この研究は，室内外におけるロボットサービスに必要な人や物の位置を計測・蓄積・構造化すると共に，様々なロボットサービスにおい

Table 4 通信時間および処理時間

項目	時間
センサ入力→ →前方車検出通信	22.9ms
前方車検出処理	15.0ms
前方車検出→ →カメラ映像表示通信	7.6ms
前方車検出→ →前方車データ集約通信	16.0ms
前方車データ集約処理	0.0ms
前方車データ集約→ →車方向データ同期通信	7.5ms
前方向データ同期処理	0.0ms
前方向データ同期→ →衝突予防システム通信	1.8ms

で汎用的に利用可能とする共通プラットフォームである。ロボットが人にサービスを提供するための一般的な枠組みとして、位置情報の計測・蓄積・構造化技術をシームレスに扱う4層構造化モデル(センサデータ層、セグメント層、プリミティブ層、サービス・アプリ層)を提案している。

一方、本研究においては、階層を限定することなく、利用するアプリケーションに応じて階層を設定し、アプリケーションは最上位の階層以外に、下位の階層のデータも直接利用可能としている。このことは、車載用途で採用される組込みシステムの計算資源やメモリを効率的に利用することができる。また、他のノード(車両)との階層ごとのデータ交換も可能となっている。たとえば、抽象化を行う前の画像データを他のノードから得ることが可能であり、また、意味づけする前のオブジェクトの位置情報のみを得ることができる。この機構により、より柔軟なアプリケーションの実現が可能となる。

7 まとめ

安全運転支援のための車載統合制御システムでは、センサからのデータを利用する複数のアプリ

ケーションが個別に処理することは可能であるが、対象物の認識などアプリケーションごとに重複する部分も多い。本研究では、各種センサから得られたデータをそれぞれのアプリケーションが直接処理するのではなく、走行環境の理解のためにデータの抽象度を向上させることで、アプリケーションに対してのデータ提供を簡単にする方法の検討を行った。この方法を利用することで、他の車両が保持している情報や、インフラの情報を車車間通信や路車間通信を通して抽象度を向上させた情報を交換することで、通信や計算の負荷を削減することも可能となる。ここでは、ストリームプロセッシングを行うソフトウェアである Borealis を利用し、PC 上に分散型データストリーム処理機構を構築した。車速、ステアリング角、車間距離、車載カメラの映像といったセンサデータを入力とし、前方車強調カメラ映像表示アプリケーション、および、衝突予防アプリケーションを実装し、システム内の通信時間、処理時間を計測することで、車載統合制御システムのための分散型データストリーム処理機構の実現可能性を探った。

謝辞

本研究を進めるに当たり、貴重なご意見を頂いたトヨタ自動車株式会社 BR 制御ソフトウェア開発室の方々をはじめ、走行データの提供をして頂いた名古屋大学情報科学研究科武田研究室各位、ご協力下さった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のメンバー各位に厚く御礼申し上げます。

参考文献

- 1) 浅沼 信吉, 加世山 秀樹, 安全運転支援のための車両予知・予測技術のとりまく状況, 国際交通安全学会誌, Vol.31, No.1, pp.56-61, 2006.
- 2) 西垣戸 貴臣, 大塚 裕史, 坂本 博史, 大辻 信也, 予防安全の高度化を実現するセンサーフュージョン技術, 日立評論, Vol.89, No.08, pp.72-75, 2007.
- 3) 藤田 浩一, 宇佐見 祐之, 山田 幸則, 所 節夫, 衝突危険性のセンシング技術, 自動車技術, Vol.61, No.2, pp.62-67, 2007.
- 4) 二宮 芳樹, 加藤 武男, 小島 祥子, 複数センサ情報の融合による走行環境認識技術, R&D Review of Toyota CRDL, Vol.36, No.3, pp.27-34, 2001.

- 5) 山田 憲一, 画像センサとレーザーダの融合処理による前方車両認識, 自動車におけるエレクトロニクス革新, (株) エヌ・ティ・エス, 2004.
- 6) Arvind Arasu, Brian Babcock, Shvsnath Babu, John Cieslewicz, Mayur Datar, Keith Ito, Rajeev Motwani, Utkarsh Srivastava, and Jennifer Widom, "STREAM: The Stanford Stream Data Manager", Stanford University Technical Report, 2004. <http://dbpubs.stanford.edu/pub/2004-20>
- 7) Sirish Chandrasekaran, Owen Cooper, Amol Deshpande, Michael J. Franklin, Joseph M. Hellerstein, Wei Hong, Sailesh Krishnamurthy, Samuel Madden, Vijayshankar Raman, Frederick Reiss, and Mehul A. Shar, "TelegraphCQ: Continuous Dataflow Processing", Proceedings of International Conference on Management of Data, pp.668-668, 2003.
- 8) Daniel J. Abadi, Don Carney, Ugur Cetintemel, Mitch Cherniack, Christian Convey, Stangdon Lee, Michael Stonebraker, Nesime Tatbul, and Stan Zdonik, "Aurora: a new model and architecture for data stream management", The VLDB Journal, Vol.12, No.2, pp.120-139, 2003.
- 9) Jianjun Chen, David J. DeWitt, Feng Tian, and Yuan Wang, "NiagaraCQ: A Scalable Continuous Query System for Internet Databases", SIGMOD Record, pp.379-390, 2000
- 10) Yong Yao, Johannes Gehreke, "The Cougar Approach to In-Network Query Processing in Sensor Networks", SIGMOD Record, Vol.31, No.3, pp.9-18, 2002.
- 11) 西垣戸 貴臣, 坂本 博史, 大塚 裕史, 大辻 信也, 予防安全の高度化を実現するセンサーフュージョン技術, 日立評論, Vol.89, No.08, pp.72-75, 2007.
- 12) Daniel J. Abadi, Yanif Ahmad, Magdalena Balazinska, Ugur Cetintemel, Mitch Cherniack, Jeong-Hyon Hwang, Wolfgang Lindner, Anurag S. Maskey, Alexander Rasin, Esther Ryvkina, Nesime Tatbul, Ying Xing, and Stan Zdonik, Proceedings of the 2nd Biennial Conference on Innovative Data Systems Research, 2005.
- 13) Shuichi Nishio, Norihiro Hagita, Takahiro Miyashita, Takayuki Kanda, Noriaki Mitsunaga, Masahiro Shiomi, and Tatsuya Yamazaki, Robotic Platforms Structuring Information on People and Environment, Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.2637-2642, 2008.
- 14) Nobuo Kawaguchi, Kazuya Takeda, and Fumitada Itakura, Multimedia Corpus of In-Car Speech Communication, Journal of VLSI Signal Processing Systems, Vol.36, Issue 2-3, pp.153-159, 2004.