3U-01

ROSのメッセージ通信を利用したダイナミックマップの検討

Consideration of A Dynamic Map using ROS Communication

青野 朝日 †　石正 幸大 ‡　佐藤 健哉 ‡
Asahi Aono　Yukihiro Ishimasa　Kenya Sato

1　はじめに

近年，知的交通システム (ITS) において，自動車の交通状況や人物の位置などの情報をマップ上に反映させるダイナミックマップの実現が進められている．既存のダイナミックマップを表示する手法として，複数の送信元からデータをデータベースに格納し，アプリケーションがデータベースから複数のデータにアクセスする．アクセス数が少ないと，遅延が小さく問題にならない，今後，コネクティッドビークルと呼ばれる通信機能を搭載した車両の普及より，さらなるデータの増加が予想され，リアルタイム性が欠如してしまう．

そこで本研究では，処理するデータ数が増加してもリアルタイム性を維持するために，送信機能と受信機能を独立させることにより，整合性を気にせず通信を行う，ROS(Robot Operating System)[1] のメッセージ通信をデータ通信に用いたダイナミックマップを表示する手法を提案する．

2　既存システムの問題点

既存システムにおけるデータベースでは，一つの処理 (トランザクション) の実行中は，他の処理は待機状態になってしまう．よって，処理は同時に実行できないため，車両が増加するとそれだけ待機状態の処理は増加してしまう．そのため，対象となるデータが増加するほど，実環境とダイナミックマップから得られる情報を取得する際の遅延が大きくなる．ダイナミックマップは，道路交通における安全性向上や効率化を目的として作られるため，リアルタイム性が要求される．よって，複数データに対して同時に処理を実行する環境が必要である．

3　提案システム

3.1　概要

提案システムでは，道路交通環境におけるダイナミックマップを ROS を用いて構築する．ROS のメッセージ通信を用いることで，データ送信元が増加しても，実環境とダイナミックマップの状況とダイナミックマップから得られる情報の遅延を低減し，リアルタイム性を維持することを目的とする．車両は，車両 ID，位置情報，向き，速度を送信し，センサーは人物の位置情報，識別番号を送信する．サーバはこれらのデータを受け取ってダイナミックマップを作成する．また，既存システムと提案システムを比較するために，車両と歩行者の距離が一定内になると警告を出す衝突判定を行う．

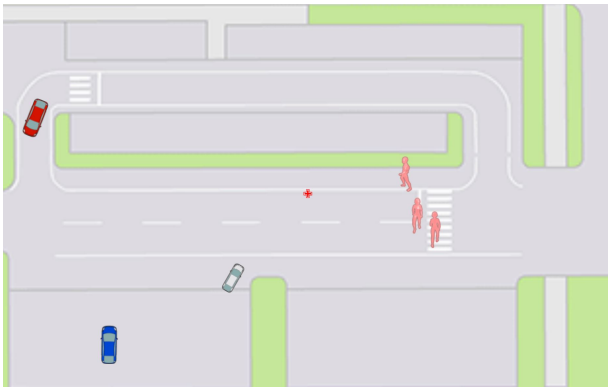

図 1　ダイナミックマップ出力画面

表 1　機器構成

機器構成	
0S	Linux
distribution	Ubuntu14.04
言語	python
ミドルウェア	ROS
接続方法	Wifi

3.2　使用機器

提案システムでは，車両やセンサー、サーバは ROS の互換性が限られるため同一規格のコンピュータを用いる．車両、センサーはこのコンピュータと接続されている．提案システムの機器構成を表 1 に示す．

3.3　システム構成

本提案システムでは，送信ノードから topic にメッセージを発行し，受信ノードでデータを受け取り，ダイナミックマップの描画と衝突判定を行う．既存システムの概要図を図 1 に示し，この提案システムの概要図を図 2 に示す．また，提案システムの送信ノードと受信ノード，及び 3 つのプロセスを以下に示す．

- 送信ノード
  車両やセンサーに積まれたコンピュータで処理するノードで，取得されたデータを topic へメッセージとして送信を行う役割を持っている．
- 受信ノード
  サーバ上で動作するノードで，ダイナミックマップを出力するノードと衝突判定を行うノードが動作している．これらのノードは topic にデータが送信されてくると動作する．

† 同志社大学 理工学部 情報システムデザイン学科
‡ 同志社大学大学院 理工学研究科 情報工学専攻

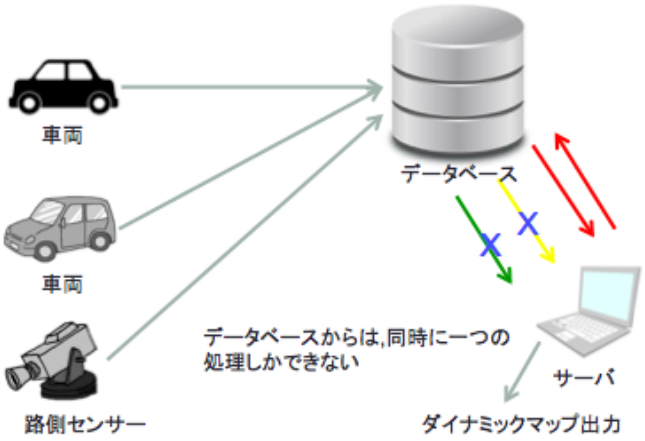

図 2　既存システム

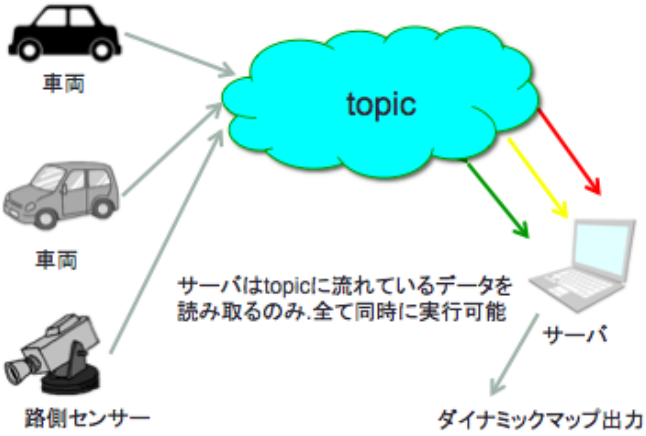

図 3　提案システム

- ROS のメッセージ通信
  各車両とセンサーは，必要な情報を取得すると，これらのコンピュータで動作している送信ノードが topic 上にメッセージを発行する．サーバは無線通信を介して，サーバ上で動作している受信ノードがこのメッセージを受信することにより通信を行う．送信側は送信，受信側は受信のみの処理を行う．
- ダイナミックマップ
  サーバ側でマップウィンドウを出力し，受け取りノードがデータを受け取るたびに，そのデータを元に，対応するウィンドウの位置に画像を出力する．topic の値が書き換えられるとそのたびに画像の位置を更新する．受け取りノードはあらかじめ待機状態にしておき，topic に値が送信されてくると画像を更新する関数を呼び出す．図 1 にダイナミックマップの出力画面を示す．
- 衝突判定
  サーバ内で衝突判定を行うためのノードを作成しておく．このノードでは，topic から値を受け取ると，現在ダイナミックマップ上に表示されている画像の位置とその受け取った値を比較し，一定の距離になると警告を表す画像をウィンドウ上に表示する．対象は，車両と車両，車両と人物の距離判定を行う．

3.4　ダイナミックマップの作成手順

(1) 車両は位置情報，速度，向き，センサーは，人の位置情報を取得する．位置情報は，緯度，経度を使用する．
(2) それぞれの機器から指定された無線通信を介して，topic にメッセージを発行する．
(3) サーバは topic からメッセージを受け取る．
(4) 受け取ったメッセージを元に，ダイナミックマップ上に画像を表示する．また，同時に衝突判定を行う．
(5) topic の値が書き換えられたら，ダイナミックマップ上の画像を更新する．

4　評価

図 2 と図 3 で表した，既存方式と提案システムにおいて，送信元からデータを送信し，サーバ側で衝突判定が行われるまでの時間を比較した．

データ送信元が少ない場合，両者ともあまり時間的な差は見られなかった．しかし，送信元の数が増加すればするほど既存方式はデータの整合性を取るために処理が増加し，時間的な差が見られ始めた．一方，提案システムは，送信元が増加してもあまり変化は見られなかった．

5　考察

今回の提案手法を用いることで，データ受信における処理の待ち時間が小さくなり，送信元の数が増加しても処理の面では，ダイナミックマップ表示の際に起こる時間的な差に影響を与えていない．よって，複数データにおけるデータ通信は ROS のメッセージ通信を用いる方が適している．.

6　まとめ

本研究では，ダイナミックマップにおける送信側 (車両やセンサー) と受信側 (サーバ) のデータ通信に ROS のメッセージ通信を用いた．提案システムでは，既存システムにおける複数データに対する処理を行った際に起こる時間的な差を解消し，リアルタイム性の維持を検討した．データ一つ一つに整合性を取らず，送信する側は送信，受信する側は受信のみの処理を行うことでデータ送信元が増加しても，アプリケーションは複数データをまとめて処理するので実環境とダイナミックマップの時間的な差を抑えることが可能である．

今後は，位置情報だけでなく，多様な事象をダイナミックマップに反映させ，より実環境に近い状況で評価を行っていく．

参考文献

[1] ROS，http://www.ros.org/