

センサネットワークを用いた移動体監視フレームワークに関する研究

長坂 康史*・千葉 達哉**

(平成22年10月29日受付)

A Study on Framework for Observation of Moving Objects Using Sensor Network

Yasushi NAGASAKA* and Tatsuya CHIBA**

(Received Oct. 29, 2010)

Abstract

For realizing the ubiquitous society, several technologies to observe moving objects have been developed in recent years. A lot of sensor nodes are used and the sensor network is introduced in most cases. The management of the nodes is, however, not easy, because there are lots of nodes. Moreover, not all the networks consist of only one kind of sensors. They make it difficult to handle the data in the sensor network. Therefore, a general-purpose framework for transferring data in the network has been developed. The proposed framework has implemented the functions not only to calculate the location of the moving object but also to predict it in the next unit time with data of near nodes. Performance measurements also have been performed to evaluate the framework. As a result, the effectiveness of the proposed framework has been confirmed because of the small processing time and the good scalability.

Key Words: sensor network, observation of moving object, software framework

1. はじめに

ヒト等の移動体を監視する技術は古くから注目されており、多くの活発な研究が行われている。この移動体監視技術として、近年の情報技術の発展により、GPSを始めとして、Active Badge[1]やマルチカメラシステム[2]など数多くの技術やシステムが開発されている。しかし、監視対象の移動体を検知するための装置、例えば、監視カメラなどは、運用の際に電源やネットワーク等のインフラを必要とするために、屋外などでの運用が難しいという問題があった。

この問題は、センサネットワーク[3]を用いることで解決することが可能であると考えられる。センサネットワークとは、無線通信機能と演算機能、そしてセンサやバッテリーを

備えた小型のセンサノードを多数配置し、そのセンサノード間を相互に接続したネットワークのことである。センサ自体が低コストであることと同時に、通信範囲内にノードがない場合でも、他のノードが通信を中継する機能を持たせることができるため、より広い領域を高密度で網羅することが可能である。しかし、センサネットワークでは、領域内に配置されたセンサノードから大量のセンサデータが発生し、これらのデータを効率よく処理する仕組みが必要である。また、用途に応じて用いるセンサが異なることや、適応事例に対応した専用のシステムを開発しなければならないという課題も存在する。

そこで本研究では、センサネットワークを用いた移動体監視を汎用的に利用することができるフレームワークを提案する。この提案フレームワークは、移動体の位置を検知

* 広島工業大学工学部情報工学科

** 三菱スペース・ソフトウェア株式会社

する機能、センサノードやシンクノードを管理するセンサネットワーク管理機能等の移動体監視に必要な機能だけではなく、複数のセンサノードから収集されるデータから移動体の位置情報へ変換し、その位置情報を用いて移動体の移動方向や移動速度を計算し、移動の予測をするための情報を提供する機能を実装する。本提案フレームワークを用いた監視システムでは、この位置情報と移動体情報、すなわち、移動速度と移動方向を用いて移動体の監視を行う。

2. 提案フレームワークで想定する移動体監視システム

本研究におけるセンサネットワークを用いた移動体監視システムでは、複数のセンサノードを用いて移動体監視を行う。センサネットワークを用いた移動体監視システムの概要を図1に示す。

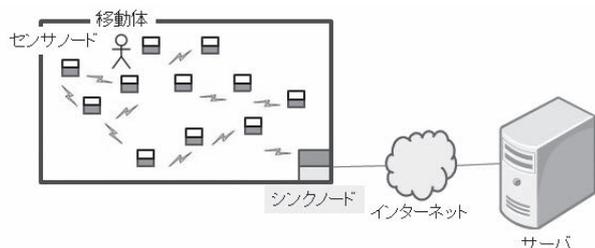


図1 センサネットワークを用いた移動体監視システムの概要図

本研究では、図1に示す通り、複数のセンサノードとそれらセンサノードを管理するシンクノード、そして、システム全体を管理するサーバでの構成を想定している。各センサノードは、自身の持つセンサを用いて移動体の検知を行う。移動体を検知した場合、センサから得たセンサデータをシンクノードに送信する。シンクノードは、受信したセンサデータをサーバへ送信する。そして、サーバではシンクノードから受信したセンサデータを用いて移動体監視を行う。

センサネットワークでは、センサノード同士が通信を行うことで、通信範囲外のノードと通信を行うことが可能である。また、センサノードは自身がバッテリーを搭載しているために運用の際に必要とするインフラが少ない。そのため、センサノードを増やすことで、監視領域を広げることが容易である。さらに、センサの種類を変更、もしくは増やすことにより、監視対象に応じた可変的なシステムを構築することが可能となる。

しかし、センサネットワークは、センサノードの数に比例して送信されるセンサデータの数が増加するため、これらを処理する仕組みが必要である。また、センサの種類に応じて、専用のシステムを構築する必要がある。本研究では、これらの課題を解決するフレームワークを提案する。

3. 提案フレームワーク

3.1 提案フレームワークの概要

提案フレームワークのAPI構造を図2に示す。本提案フレームワークには、移動体の移動先を予測する機能、センサノードやシンクノードを管理する機能等のセンサネットワークを用いて移動体監視を行うためのAPI機能が移動体監視APIとして実装されている。それに加えて、センサノードIDや移動体オブジェクト等のデータを送受信する通信APIや、移動体の位置情報、移動方向や移動速度といった、移動体情報を管理する移動体情報管理APIが実装されている。

また、提案フレームワークでは、監視領域を複数の区域に分割し、それぞれにシンクノードを配置して区域を管理させることで移動体の監視を行う。

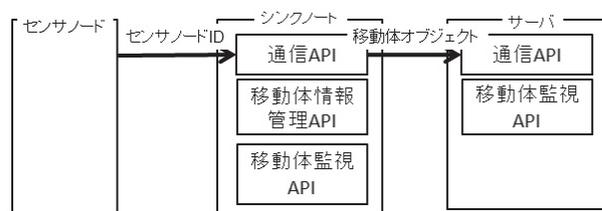


図2 提案フレームワークのAPI構造

3.2 位置情報の処理

提案フレームワークでは、監視領域をローカル座標として取り扱う。さらに、センサノードを格子状に設置し、それぞれに位置情報と関連付けたIDを付加する。これにより、センサノードのIDのみを用いて位置情報を求めることができ、センサの種類に依存しないシステムを構築することが可能となると考える。センサノードIDとその例を図3に、センサノードの配置例を図4に示す。



図3 センサノードIDの例

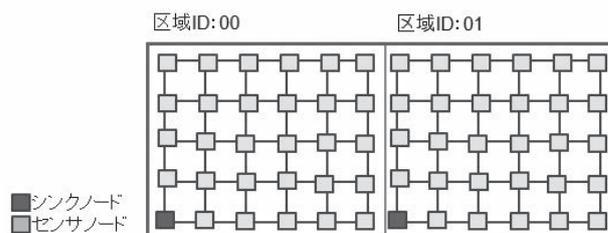


図4 センサノードの配置例

図3に示す通り、センサノードのIDはセンサノードの位置を示す値となっている。内訳は、区域ID (ID)、区域

内の横座標 (X), 縦座標 (Y) となっている。提案フレームワークでは, センサノードが移動体検知時にこのセンサノード ID を送信する。シンクノードでは, この ID から, 移動体の位置を計算する。また, シンクノードを原点として, センサが離れていくほどに, センサ ID の X, Y 座標が1ずつ増加していく。また, シンクノードが管理する区域には ID が付加されており, サーバはこの ID を用いて分割した区域を管理する。例えば, 区域 ID00 においてシンクノードから X 座標が5, Y 座標が2 離れているセンサノードの ID は【00】【05】【02】となる。なお, センサノードをある程度均等に配置することにより, 前述したセンサノード ID から位置情報を求めることが容易となる。

3.3 移動傾向の計算

前述したセンサノード ID より位置情報を計算した後, 提案フレームワークでは移動体の移動方向 (以下, 移動傾向) と移動速度を計算する。なお, 計算される移動方向と移動速度は, 直近の2点間の移動傾向と, 全体の移動傾向の2方向と速度である。その概要を図5に示す。

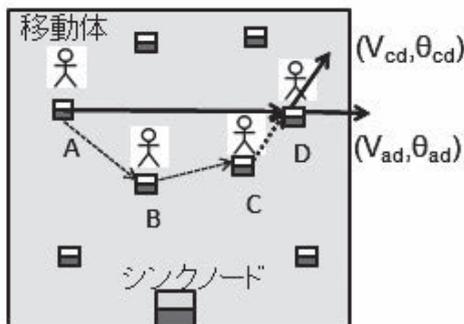


図5 移動傾向の計算

図5では, 移動体がセンサA, B, C, Dの順番に通過したと仮定している。この場合, 直近の2点の移動傾向 (V_{cd} と θ_{cd}) と, 全体の移動傾向 (V_{ad} と θ_{ad}) が計算される。提案フレームワークでは, この移動傾向を用いて移動体監視を行う。

3.4 移動体オブジェクト

提案フレームワークでは, 移動体情報を移動体オブジェクトとして捉え, センサノードから受信したデータを用いてこれを更新する。移動体オブジェクトに関するクラスを図6に示す。

図6に示す通り, 移動体クラスは MovingObject として表記され, その中には移動体 ID (movingObjectID) と位置情報オブジェクト (PositionInfo) として, 移動体を検知した最初の点 (first), 移動体を最後に検知した直近の2点 (previous, last) が, 移動傾向オブジェクト

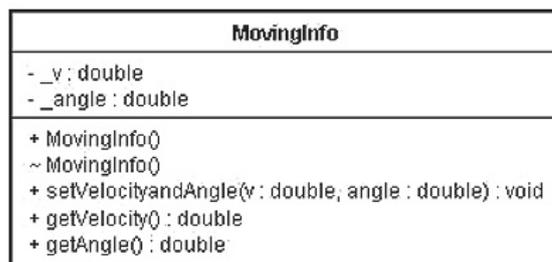
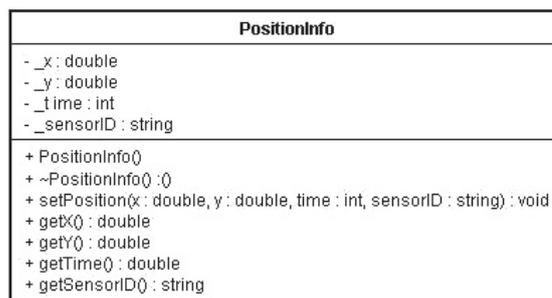
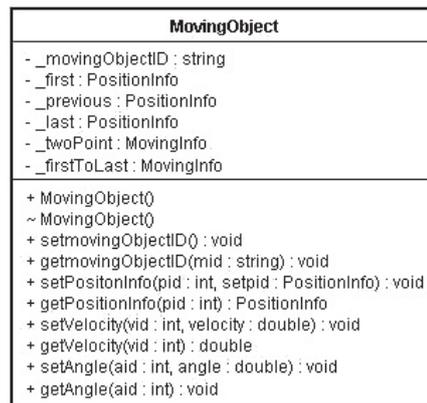


図6 移動体オブジェクトのクラスの概要

(MovingInfo) として直近の2点間の移動方向と移動速度 (twoPoint) と, 全体の移動方向と移動速度 (firstToLast) が格納されている。移動体の位置情報クラスである PositionInfo には移動体を検知した位置 (x, y) と時間 (time), センサノードの ID (sensorID), 移動体の移動傾向クラスである MovingInfo には移動体の移動速度 (v) と移動方向 (angle) が格納されている。

3.5 提案フレームワークを用いたシステムの概要

提案フレームワークを用いたシステムの概要を図7に示す。

シンクノードでは, センサノードから受信したデータを用いて移動体情報を計算し, その移動体情報を用いて自身の管理する区域で移動体監視を行う。ここでの移動体監視とは, 移動体情報を用いて一秒後の移動体の位置を計算することである。移動体監視を行った結果, シンクノードが管理する区域を移動体が超えると判断した場合, 移動体オブジェクトをシリアライズしてサーバへ送信する。サーバは受信した移動体オブジェクトをデシリアライズした後, その移動体オブジェクトを用いて監視領域全体についての移動体監視を行う。

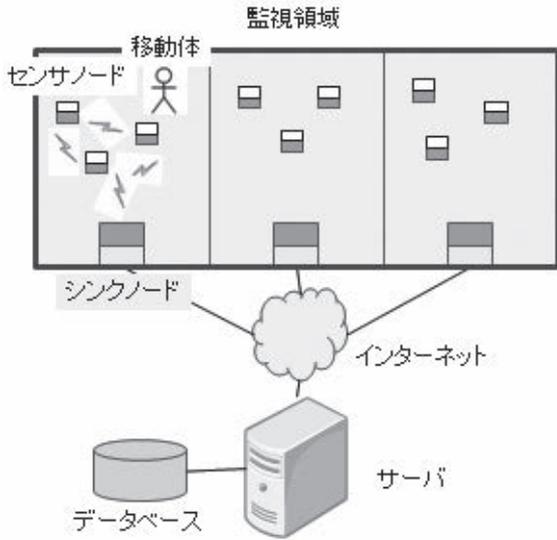


図7 提案フレームワークを用いたシステムの概要

りに、コストが低く、なおかつ消費電力が少ないという特徴を持つ、センサネットワークを構築するための短距離無線プロトコルの一つである。

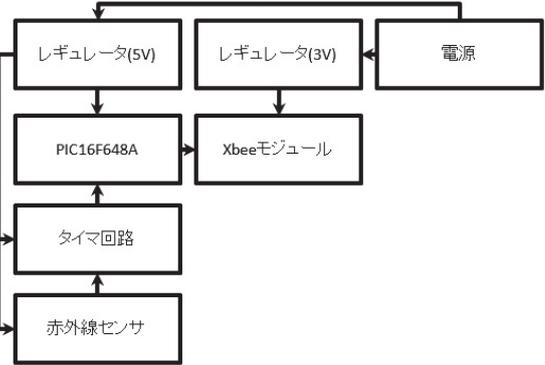


図8 センサボードブロック図

4. 性能評価

提案フレームワークの性能評価を行うために、フレームワークを用いてシステムを実装し、シンクノードではセンサノード数の増加と処理時間の関係を、サーバではシンクノード数の増加と処理時間の関係を測定した。なお、測定項目をフレームワークの各機能の処理時間とシステム全体の処理時間とした。また、測定にあたり、ノード数を仮想的に3, 6, ..., 48台と増加させることで測定を行った。

4.1 測定環境

表1にそれぞれセンサノード、シンクノード、サーバの仕様を示す。性能測定は、サーバ1台、シンクノード1台、センサノード3台で行った。

表1 性能一覧

サーバ	CPU	Intel® Xenon® 2.66 GHz
	メモリ	256 MB
シンクノード	CPU	Intel® Celeron® 1.73Ghz
	メモリ	2 GB
センサノード	OS	Scientific Linux CERN
	CPU	Intel® Celeron® 1.73Ghz
	メモリ	2 GB
サーバ	OS	Ubuntu release 9.10
	CPU	PIC16F648A
	メモリ	256 byte
センサノード	OS	なし

4.2 センサボード

性能評価を行うに当たり、センサボードを作成した。そのセンサボードのブロック図を図8に示す。センサボードには電源とCPU、赤外線センサ、タイム回路、そして通信のためにXbeeと呼ばれるZigbeeと呼ばれるプロトコルを用いるための無線通信モジュールを搭載している。なお、Zigbeeとは、通信速度が低速で伝送距離も短い代わ

4.3 結果と考察

性能評価を行った結果を図9、および図10に示す。図9はノード数の増加による、フレームワークが持つ機能の処理時間への影響を示すものである。また、図10はノード数の増加による、システムの総処理時間とフレームワークの総処理時間への影響である。なお、測定結果はセンサノード、シンクノード間とシンクノード、サーバ間の通信を100回ずつ行った値の平均である。

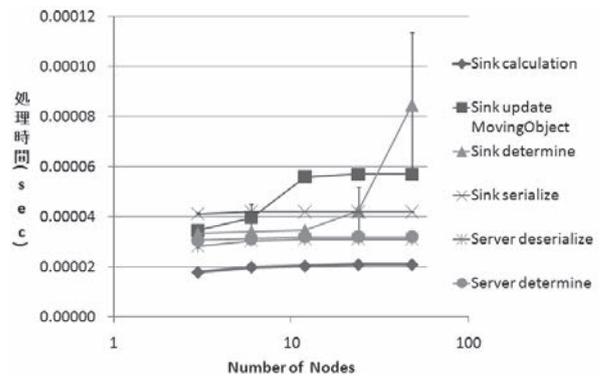


図9 ノード数の増加によるフレームワークが持つ機能の処理時間への影響

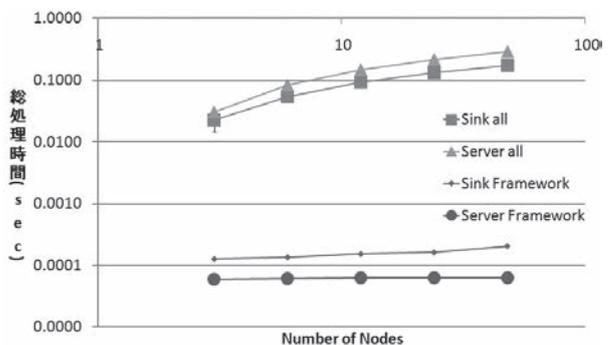


図10 ノード数の増加によるシステムの総処理時間とフレームワークの総処理時間への影響

図9中の各線は、凡例において、上からシンクノードにおける移動傾向の計算、移動体オブジェクトの更新、移動体監視、移動体オブジェクトのシリアルライズ、サーバにおける移動体オブジェクトのデシリアルライズ、移動体監視となっている。図9より、フレームワークが持つ機能の処理時間は、ノード数の増加による処理時間への影響は少なく、処理時間は非常に緩やかな増加をしていることが分かる。

また、図10中の各線は、凡例において、上からシンクノードにおける総処理時間、サーバにおける総処理時間、シンクノードにおけるフレームワークの総処理時間、サーバにおける総処理時間となっている。図10より、ノード数と比較してシステムの総処理時間が増加していることが分かる。これは、ノード数の増加により、受信側でデータ受信の待ち時間が発生しているためであると考えられる。

図10ではノード数の増加によるフレームワークの総処理時間への影響も示している。図より、提案フレームワークの総処理時間はノード数の増加に関わらずシンクノード、サーバともに緩やかに増加していることが分かった。この結果より、提案フレームワークを導入することによる影響はノード数が増加していくごとに非常に小さくなって行くことが分かる。

5. まとめ

本研究では、センサネットワークを用いて移動体監視を行うためのフレームワークを提案し、その有効性を検証した。提案フレームワークは移動体監視の機能だけではなく、センサノードが送信したセンサノードIDを移動体の位置

情報へ変換し、その位置情報を用いて移動体の移動速度と移動方向を計算する機能を実装した。提案フレームワークは、センサノードのIDから位置を検出することにより、センサの種類を問わず位置情報を取り扱うことが可能となる。また、移動体をオブジェクトとして捉え、移動体情報をこのオブジェクトに格納することにより、移動体情報の管理を容易に行うことが可能となる。

また、性能評価より、提案フレームワークはノード数の増加に対して有効であることが分かった。しかし、システムの総処理時間はノード数の増加と比例して増加していた。これは、ノード数の増加による、データ受信待ち時間が発生しているためであると考えられる。今後の課題として、この待ち時間の減少が挙げられる。

参考文献

- [1] R. Want, A. Hopper, V. Falcao, and J. Gibbons. "The Active Badge Location System," in ACM Transactions on Information Systems, pp.91-102, 1992.
- [2] R. Collins, et al. "A system for video surveillance and monitoring: VSAM final report", Technical report CMU-RI-TR-00-12, Robotics Institute, CMU, May 2000.
- [3] 戸辺義人, "無線センサネットワークの技術動向", 電子情報通信学会論文誌 (B), vol. J90-B, no.8, pp.719-771, Aug, 2007.
- [4] 千葉, 長坂: 「センサネットワークを用いた移動体監視システムのための汎用フレームワークに関する研究」, FIT2009 第8回情報科学技術フォーラム (2009)

