Vol.2012-HCI-148 No.22
2012/6/2

タイルドディスプレイを利用した センサネットワーク情報の可視化

浅川和久† 仙道航† 瀬川典久† 澤本潤†

近年、無線通信機能と計算機能を有した小型センサノードを多数配置することで、温度や湿度、光量など様々な情報を収集するセンサネットワークが注目されている。さらに屋内外を問わずにセンサネットワークを構築できるため、ユビキタスコンピューティングのプラットフォームとして、環境計測システム、セキュリティシステム、 医療・福祉システム、各種トレーサビリティシステムなど幅広い応用が期待されている。
センサネットワークでは、センサノードによるセンサデータの収集のほかに、収集したセンサデータを分析して可視化するためのアプリケーションの開発も重要である。
本研究では、センサネットワークのためのデータベースシステムおよび可視化フレームワークを開発した。また、開発したデータベースシステムおよび可視化フレームワークを利用してセンサデータの可視化を行った。本稿では、タイルドディスプレイに拡張した部分に着目し、その実装方法と実現例について示す。

Visualization of sensor network information Using a Tiled Displays Wall

KAZUHISA ASAKAWA† WATARU SENDO† NORIHISA SEGAWA† JUN SAWAMOTO†

Sensor networks collecting quantitative temperature, humidity, and light data by placing many small sensor nodes equipped with functions of calculation and wireless communication have attracted attention lately. Because these networks can be installed anywhere indoors or outdoors, they are expected to function as a platform of ubiquitous computing providing applications in environmental monitoring, security systems, and healthcare systems. In sensor networks, in addition to collection of sensor data at nodes, development of application programs to analyze and visualize the collected data is important and indispensable. In this study, we have developed a database system and a framework for a visualization program for sensor networks. Furthermore, making use of them, we performed visualization of sensor data. In this paper, we focus on extensions to the Tiled Display Wall, shown for cases and implementation of how to implement them.

1. はじめに

センサネットワークの基本技術が開発され約１０年経ち，現在その技術をいかに一般社会に生かして行くかが重要な課題となっている[1].

現在おこなわれているセンサネットワークの実用化の実証実験は，都市部または安定してネットワークインフラが構築できる地域を対象とすることが多い．しかしながら，中山間地等の条件不利地域が多い東北地域では，携帯パケット，ADSL 等の通信網はもとより，容易に電源を得るのも難しい状況である．また，山間地等の地理的条件から，簡単にセンサネットワークのインフラを構築するのが困難であるのが実情である． しかし，都市部，中山間部といった利用する場所が異なるだけの理由でセンサネットワーク技術の利益が得られないことは避けたいと考えている．

そこで，我々の研究グループでは，山間部でセンサネットワークを有効利用することを考え，さまざまな研究を行っている（図１）．

† 岩手県立大学 ††
Iwate Prefectural University

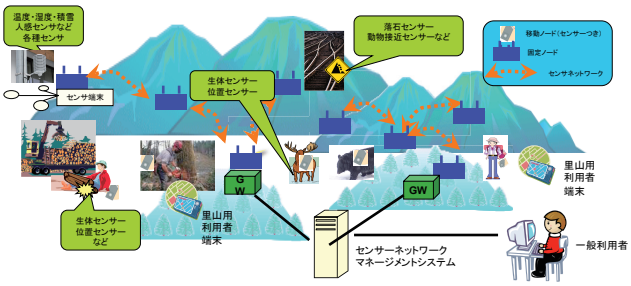

図１ 山間地におけるセンサネットワークの利用

センサネットワークでは、センサノードによるセンサデータの収集のほかに、収集したセンサデータを分析して可視化するためのアプリケーションの開発も重要である。

本研究では、異なるハードウェアベンダーのセンサノードや複数のセンサネットワークを１つのデータベースで管理するためのセンサネットワークのためのデータベースシステムおよび可視化フレームワークを開発した。また、開発したデータベースシステムおよび可視化フレームワークを利用してセンサデータの可視化を行った。

研究の中で、得られたセンサの情報が広域にまたがるにつれ、センサの情報を普通のディスプレイで確認するのは非

ⓒ2012 Information Processing Society of Japan

常に困難になっている。そのために、ユーザが広範囲のセンサ情報を容易に確認する手段を提供する必要がある。
そこで､我々はタイルドディスプレイシステム(TDW)の活用を考えた。タイルドディスプレイシステムとは、複数のディスプレイを縦横に連結し、表示するもので、解像度を増加させることが可能である。しかし、複数のディスプレイを連結して 1 枚のディスプレイを構成しているために、ディスプレイ間の連携をとるひつようがある。

本稿では、センサネットワークのデータの管理部および可視化フレームワークについて説明する。そして、タイルドディスプレイに拡張した部分に着目し、その実装方法と実現例について示す。

2. センサネットワークの情報を管理するシステム「SenseDB」

異なるハードウェアベンダーのセンサノードや複数のセンサネットワークを 1 つのデータベースで管理し、情報を可視化するためのシステム「SenseDB」を構築した。

SenseDB は、主にセンサネットワークのデータの管理、及びそのデータの可視化の機能を持つ。

2.1 センサネットワークのデータ管理機能

SenseDB は､センサネットワークからデータベースシステムでは、センサデータの選択、追加、削除、更新に関する操作を RESTful API により利用することができる。
REST（Representational State Transfer）とは、ネットワーク上の Web サービスに適用されるソフトウェアアーキテクチャの一種であり、明示的に HTTP メソッドを使用し、ディレクトリ構造に似た URL を持ち、XML や JSON のいずれか、または両方を転送する。
データベースシステムは一般的な LAMP で構成されている。データベースシステムの構成を表 1 に示す。

表 1 データベースシステムの構成

OS	FreeBSD 7.1-RELEASE-p16 i386
Web サーバ	Apache/1.3.42 (Unix)
Web アプリケーション	PHP 5.2.17
データベース	MySQL 5.5

異なるハードウェアベンダーのセンサノードや複数のセンサネットワークを 1 つのデータベースで管理するためのデータベーステーブルを表 2 のように設計した。ノード番号を管理するためのカラムのほかに、double 型のデータを管理するためのカラムを 10 個用意した。また､座標や日付を管理するカラムにはそれぞれ空間情報に最適な geometry 型、日付に最適な datetime 型を設定した。
値を省略した場合は、デフォルト値を使用され、各カラムに格納できるデータは MySQL のデータ型の仕様に従う。

表 2 データベーステーブル

カラム	型	デフォルト値
ノード番号	double	0
データ 1	double	0
データ 2	double	0
データ 3	double	0
データ 4	double	0
データ 5	double	0
データ 6	double	0
データ 7	double	0
データ 8	double	0
データ 9	double	0
データ 10	double	0
座標	geometry	0,0
日付	datetime	0000-00-00 00:00:00

データベースシステムに蓄積されたセンサデータを扱うための API 群（データベース API）を用意した。プロジェクト作成時にデータベースシステムから発行される、プロジェクト ID、読み込みキー、書き込みキーを利用してセンサデータにアクセスする。プロジェクトとは、センサデータの種類や目的をもとにグループ化したセンサデータの集合のことをいう。センサデータにアクセスするためのメソッドと対応するキーを表 3 に示す。

表 3 センサデータにアクセスするメソッド

メソッド	説明	キーの種類
post	センサデータを追加する	書き込みキー
put	センサデータを更新する	書き込みキー
delete	センサデータを削除する	書き込みキー
get	センサデータを取得する	読み取りキー

2.1.1 POST,PUT,Delete メソッド

センサデータを追加するための URI は、以下となる。

```
POST http://exsample.com/post/[プロジェクトID]/[書き込みキー]/
```

リクエストボディは、以下となる。entry ノードは最大 100 ノードまで指定できる。node はセンサノード番号、data1 から data10 はセンサデータ､pointX と pointY はそれぞれ位置情報の X 座標と Y 座標、date は日付を指定する。

データの例

```
<?xml version="1.0"?>
<feed>
  <entry>
    <node></node>
    <data1></data1>
    <data2></data2>
        ...
    <data9></data9>
    <data10></data10>
    <pointX></pointX>
    <pointY></pointY>
    <date></date>
  </entry>
</feed>
```

同様に、センサデータを更新するための URI は、以下とな

る。

```
POST http://exsample.com/put/[プロジェクトID]/[書
    き込みキー]/[センサデータID]/
```

センサデータを削除するための URI は、以下となる。

```
GET http://exsample.com/delete/[プロジェクトID]/[
    書き込みキー]/[センサデータID]/
```

2.1.2 get メソッド

センサデータを取得するための URI は、以下となる。

```
GET http://exsample.com/get/[プロジェクトID]/[読み
    取りキー]/[ページ番号]/[フォーマット]
GET http://exsample.com/get/[プロジェクトID]/[読み
    取りキー]/[検索式]/[ページ番号]/[フォーマット]
```

検索式は、フィールド、演算子、オプションから構成される。

フィールドには、センサデータ ID の id、センサノード番号の node、センサデータの data1 から data10、日付の date を指定できる。

演算子は、”<”、”<=”、”>”、”>=”、”!=”、”=”が指定できる。

オプションとして、複数の検索式を AND で連結するための¥_、センサデータを昇順で取得するための asc、センサデータを降順で取得するための desc、センサデータを 10 件取得するための limit10、25 件取得するための limit25、50 件取得するための limit50、100 件取得するための limit100、センサノードのリストを最新のセンサデータを付加して取得するための nodeList がある。

検索式は[フィールド][演算子][数値]のように構成する。[数値]には 0 から 9 とマイナスを表す-と小数点を表す.が使える。

date を指定した場合の[数値]は 8 桁または 12 桁で指定する。例えば 2011 年 12 月 3 日は、”20111203”、2011 年 12 月 3 日 4 時 56 分は”201112030456”になる。

複数の検索式を構成するには、[フィールド][演算子][数値][_][フィールド][演算子][数値]のように_を使用する。asc または desc を複数指定した場合は最後に指定した asc または desc が有効になる。デフォルト値は desc である。

limit10、limit25、limit50、limit100 を複数指定した場合は最後に指定した limit10、limit25、limit50、limit100 が有効になる。デフォルト値は limit10 である。

nodeList を指定した場合は、その他の検索式は無効になる。

センサデータの出力フォーマットとして xml、csv、json、jsonp、Google Earth や Google Maps で利用できる kml をサポートしている。

フォーマットして XML フォーマットを指定した場合のレスポンスボディは以下となる。

表 4 にそれぞれのノードの意味を示す。

get メソッドを使用した例

```
<?xml version="1.0" encoding="UTF-8"?>
<feed>
   <title>BirdGPS</title>
   <description>
   <![CDATA[
   BirdGPS
   ]]>
   </description>
   <id>1</id>
   <updated>2012-01-18 17:51:30</updated>
   <request>
   <![CDATA[
   http://exsample.com/get/[プロジェクト
ID]/[読み取りキー]/limit10/1/xml
   ]]>
   </request>
   <page>
      <total>2025</total>
      <total_pages>203</total_pages>
      <current_page>1</current_page>
   </page>
   <entry>
      <id>1529</id>
      <node>42</node>
      <data1>0</data1>
      <data2>0</data2>
      ...
      <data9>0</data9>
      <data10>0</data10>
      <point>39.678585,141.16638</point>
      <date>2011-12-20 09:39:23</date>
   </entry>
 </feed>
```

表４　各ノードの説明

ノード	説明
title	プロジェクトの名前
description	プロジェクトの説明
id	プロジェクトの ID
updated	リクエストした日時
request	リクエスト URI
total	センサデータの件数
total_page	ページ数
current_page	ページ番号
id	センサデータ ID
node	センサノード番号
data1 - data10	センサデータ
point	座標
date	日付

以下検索の例を示す。

検索式を利用したセンサデータの取得の例を示す。

data1 の値が 12.3 より大きいものを 50 件 CSV フォーマットで取得する

```
http://exsample.com/get/[プロジェクトID]/[読み取り
    キー]/data1>12.3_limit50/[ページ番号]/csv
```

date が 2011 年 1 月 1 日のものを KML フォーマットで取得する

```
http://exsample.com/get/[プロジェクトID]/[読み取り
    キー]/date>=20110101_date<=20110101/[ページ番
    号]/kml
```

ⓒ2012 Information Processing Society of Japan

date が 2011 年 1 月 1 日 12 時 34 分以降のものを昇順で KML フォーマットで取得する

```
http://exsample.com/get/[プロジェクトID]/[読み取りキー]/date>=201101011234_asc/[ページ番号]/kml
```

3. 可視化フレームワークと可視化コンポーネント

データベース API を利用した可視化機能を提供するための可視化フレームワークを提供する。
図 2 に可視化フレームワークの構成を示す。

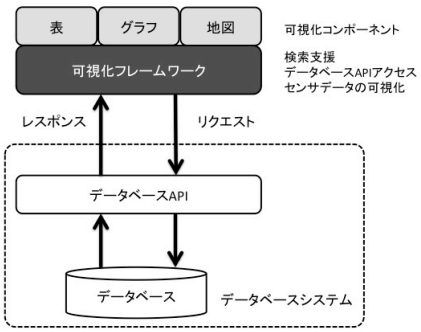

図 2 可視化フレームワーク

可視化フレームワークはデータベース API の仕様をもとに設計されており、検索支援、データベース API アクセス、センサデータの可視化といった機能を持っている。
ユーザは、可視化フレームワークを利用してデータベースシステムにアクセスを行う。
可視化フレームワークは、データベース API にリクエストし、レスポンスとしてセンサデータを XML フォーマットで受け取り、表やグラフといった可視化コンポーネントにセンサデータを描画する。
可視化フレームワークとデータベース API の通信には、Ajax という技術を利用している。
また、ブラウザの互換性に関する問題を解決し、Web 開発の効率を高めるために jQuery[2]という Ajax アプリケーションフレームワークを利用している。
可視化コンポーネントのグラフは、グラフに特化した jqPlot[3]、可視化コンポーネントの地図は、Google Maps API[4]を利用している。図 3 に、可視化コンポーネントを示す。

ユーザは、このインタフェースを利用することで可視化フレームワークが提供する検索支援機能を活用することができる。
検索支援機能は、検索式を入力できる「検索」、地図や折れ線グラフ、棒グラフなどの可視化コンポーネントを指定するための「表示」、
XML や KML、CSV などの出力フォーマットを指定する「フォーマット」、取得件数を指定する「検索」などから構成されている。
また、「検索」は、データベース API の検索式と同じ記述が可能である。

ユーザが、可視化フレームワークを利用してデータベースシステムにアクセスを行うためのインタフェースを図 4 に示す。

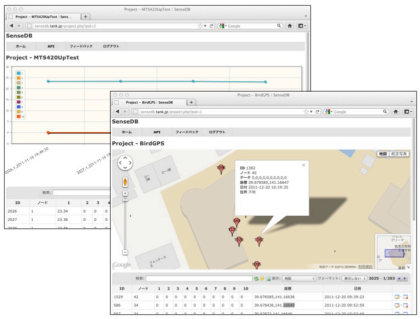

図 3 可視化コンポーネント

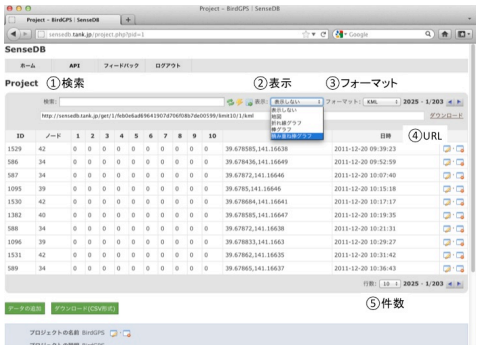

図 4 可視化コンポーネント制御インタフェース

「フォーマット」や「検索」は、データベースシステムにアクセスする際にそれぞれデータベース API の検索式に変換される。

4. SenseDB のタイルドディスプレイへの出力

4.1 岩手県立大学 i-MOS タイルドディスプレイシステム

本システムを、岩手県立大学のタイルドディスプレイシステムに出力した事例について報告する。岩手県立大学地域連携研究センター(i-MOS)には、3 次元実験室が設置されている（図 5）。3 次元実験室には、タイルドディスプレイシステム(TWD)が設置されており、外部の人も利用できる環境になっている[5]。TDW の仕様は、図 6 に示す。

ⓒ2012 Information Processing Society of Japan

図 5 i-MOS 3 次元実験室 TDW [5]

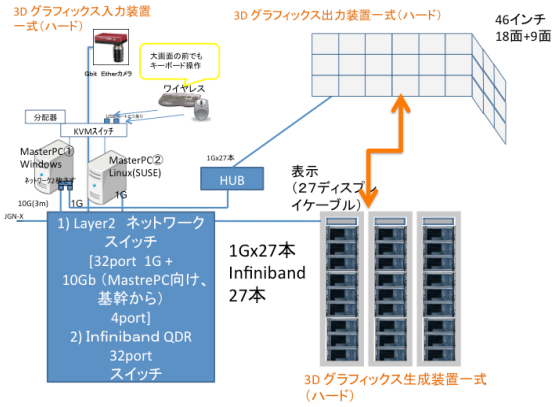

図 6　i-MOS 3 次元実験室構成図

4.2 Sense DB のタイルドディスプレイシステムへの拡張

SenseDB の可視化フレームワークをタイルドディスプレイへ拡張を行った。本拡張は、主に位置情報を元に構成されたセンサ情報を対象に行う。よって、本システムでは具体的に、google map を対象にタイルドディスプレイへの拡張を行った。

実装は、可視化コンポーネントにタイルドディスプレイを追加することで実現した。図 7 にシステムの概要を示す。

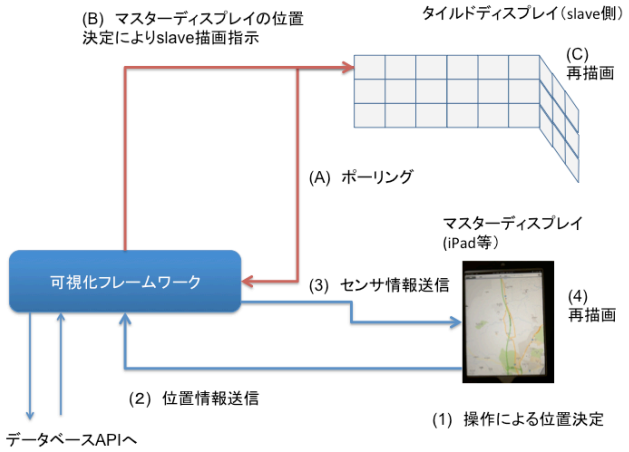

図 7　タイルドディスプレイ API のシステム構成図

利用者が、マスタディスプレイ(iPad 等)で google map を操作する。その情報は、既存の可視化フレームワークによって、データベース API を利用し、マスターディスプレイでセンサ情報の可視化が行われる。

タイルドディスプレイ側は、一定時間毎（現在は 5 秒おき）にポーリングを行い、マスターディスプレイの操作情報を監視し、情報が変化したときに、マスターディスプレイの再描画を行う。

本システムの実装は、Chrome ブラウザの kiosk モードを利用して行った。実装方法は、横山他[6]の WDiM と同様にマスタディスプレイおよびタイルドディスプレイ側は Java Script で構築している。WDiM とは、以下の点が異なる。

(1) 各ディスプレイのタイル環境変数の設定が不要、ただしベゼル幅の大きさを考慮していない。
(2) センサデータの描画に着目しているために、センサデータとの連携機能が実装されている。
(3) タイルドディスプレイの操作を、マスタディスプレイの操作で行う。

4.3 SenseDB タイルドディスプレイインタフェース

SenseDB における、タイルドディスプレイ可視化コンポーネントのインタフェースの定義を以下に示す。

4.3.1 マスタディスプレイ API

マスタディスプレイを処理するための可視化コンポーネント API は、以下のように定義される。

(A) GET http://[SenDBアドレス]/display/[モード] /master/[プロジェクトID]/

(B) GET http:// [SenDBアドレス]/display/[モード] /master/[プロジェクトID]/#[KMLファイルのURL]

[モード]:タイルドディスプレイで利用するアプリケーション
[プロジェクト ID]:処理するセンサデータベースの ID

本 API における[モード]は、タイルドディスプレイ上で利用するアプリケーションに対応する。現状では、google map(値：map)のみ指定可能である。プロジェクト ID は、センサネットワークから取得された一連のデータが蓄積されているデータベースを指す。

また、#[KML ファイルの URL]を指定することで、google map 上に表示するセンサデータの形式を指定することが可能になる。例えば、

GET http:// [SenDBアドレス]/display/[モード]/master/[プロジェクトID]/# http:// [SenDBアドレス]/get/1/[プロジェクトID]/node=42_limit100/1/kml

と記述することで、センサノードのID42番の１００個のデータを取得するkmlを利用して、タイルドディスプレイのマスタディスプレイにセンサ情報を表示する。

ⓒ2012 Information Processing Society of Japan

4.3.2 タイルドディスプレイ API

タイルドディスプレイを処理するための可視化コンポーネント API は、以下のように定義される。

GET http://[SenDB アドレス]/display/[モード]/slave/[プロジェクト ID]/[読み取りキー]/#[ディスプレイ X 座標],[ディスプレイ Y 座標]

[モード]:タイルドディスプレイで利用するアプリケーション
[プロジェクト ID]:処理するセンサデータベースの ID
[読み取りキー]:センサデータベースにアクセスするためのキー(256bit MD5)

#[ディスプレイ X 座標],[ディスプレイ Y 座標] : ディスプレイの座標

本 API は、マスタディスプレイによって表示された情報を中心とし、その周りの画像をタイルドディスプレイで表示を行うための API である。本 API によって、各タイルドディスプレイを構成するディスプレイ 1 枚毎に URL が付与され、独立して描画することで、高解像度の画像を生成する。

タイルドディスプレイの API は、ディスプレイの解像度、ディスプレイの枚数、設置の組み方に依存せずに動作する。マスタディスプレイの表示部分を中心と定め、それを中心にＵＲＬが決定される。
例えば 3x4 のディスプレイ構成の場合、以下のような#[ディスプレイ X 座標],[ディスプレイ Y 座標]を構成することが可能である。

表 5 [ディスプレイ X 座標],[ディスプレイ Y 座標]の例

2,1	-1,1	0,1	1,1
2,0	-1,0	0,0	1,0
-2,-1	-1,-1	0,-1	1,-1

4.3.3 システム動作例

4.3.1-4.3.2 の API を用い、センサ情報の可視化を行った。盛岡市からの依頼で、盛岡付近を飛行するカラスの位置情報を調査することになった。そこで、本研究で利用する送信ノードをカラスの背中にとりつけ、１週間飛行データを取得することにした。
2011 年 12 月 20 日に、あらかじめ捕獲許可を得て捕獲していたカラス６羽に送信機をとりつけ（図 8,9）、盛岡市内で放鳥し、リアルタイムに追尾を行った[7]。

（左）図 8 送信ノードをカラスに装着
（右）図 9　カラスに装着した送信ノード

広域センサネットワークから得られた環境情報を、いわてものづくり・ソフトウェア融合テクノロジーセンター(i-MOS)の３次元実験室で可視化を行った（図 10）。得られたセンサの情報を、タイルドディスプレイの１８面に google map の連結表示することで、多人数での詳細なデータの議論が可能になった（図 11，12）。

以下の図に、実装例を示す。

図 10　i-MOS　３次元実験室でのセンサ情報の可視化

ⓒ2012 Information Processing Society of Japan

図 11　カラスの軌跡データのタイルドディスプレイでの表示

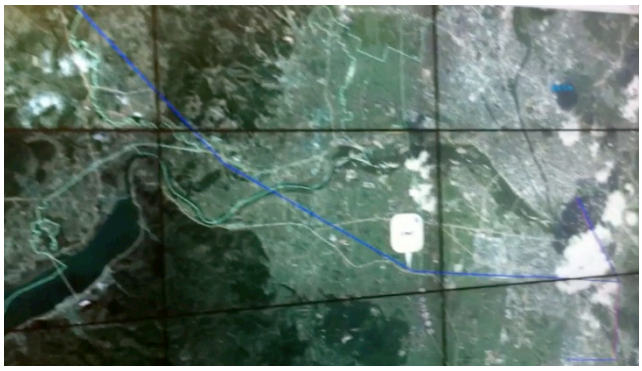

図 12　タイルドディスプレイによる衛星写真とカラスの軌跡データの合成

5. おわりに

私たちは、異なるハードウェアベンダーのセンサノードや複数のセンサネットワークを 1 つのデータベースで管理するためのセンサネットワークのためのデータベースシステムおよび可視化フレームワークを開発した。
また、可視化フレームワークにブラウザによる分散レンダリング機能を追加することにより、タイルドディスプレイを利用した広大な画面での可視化を可能にした。
本研究で開発した開発環境は、異なるメーカーのセンサノードや複数のセンサネットワークで利用できるように設計しているため、実用性が高く、様々な場所や場面での活用を想定している。
今後の研究課題は、本研究で開発した開発環境を利用して、新しいセンサネットワークアプリケーションを構築することである。また、可視化コンポーネントの拡張として、google earth API への適用も考えている。

謝辞　本研究は、JST 研究成果最適展開支援プログラム(A-STEP)「広域センサネットワークの構築と広域センサ情報の可視化　（課題番号：231Z04303）」の支援を受けています。

参考文献

1)　里山での活動を支援するセンサネットワーク環境構築の提案”,瀬川典久, 澤本潤, 松原和衛, 出口善隆, 大石明広, 山本信次, 東淳樹, 青井俊樹, 信学技報, vol. 109, no. 131, USN2009-23, pp.141-146(IEICE USN 研究会京都), (2009)
2) jQuery: The Write Less, Do More, JavaScript Library, http://jquery.com/, 2012.
3) jqPlot Charts and Graphs for jQuery, http://www.jqplot.com/, 2012.
4) Google Maps API ファミリー, http://code.google.com/intl/ja/apis/maps/, 2012.
5) i-Mos 施設概要, http://i-mos.iwate-pu.ac.jp/detail/index#3dlabo
6) 横山 昌平, 石川 博, "Web ブラウザによる超高解像度可視化基盤の開発", 情報処理学会論文誌, Vol.52, No.1, pp.56-67, 2011.1.
7) Norihisa Segawa, Jun Sawamoto, Kazuhisa Asakawa, Mitsuhiro Kashiwada, Atsuki Azuma, Hirokazu Takahashi, Chiaki Nishi,Kenichi Tokita, Kiyokazu Fujinaga, Masato Yazawa, Haruo Tamaki, “Construction of Position Tracking System of Birds using MAD-SS Sensor Network”, IPSN 2012 workshop, Mobile Sensing, http://research.microsoft.com/en-us/um/beijing/events/ms_ipsn12/posters.html

ⓒ2012 Information Processing Society of Japan