

ヘテロジニアスなセンサ環境における 位置取得システムの構築

青木 俊¹ 岩本 健嗣² 由良 淳一² 徳田 英幸¹²

¹ 慶應義塾大学環境情報学部 ² 慶應義塾大学大学院 政策メディア研究科

近年、計算機の小型化が進み、ラップトップのような形だけではなく、家電などへの組み込みや、計算機とセンサ等との組み合わせによって独自の機能を持たせるなど、形を変えて存在するようになっていいる。それらの機器を接続することによって、人々の要求に応じたサービスが提供される環境をユビキタス環境と呼ぶ。ユビキタス環境構築の研究は多くなされているが、その中で、情報世界と物理世界のマッピングは大きな課題の一つである。例えばユーザや物の物理的な位置情報を取り扱うアプリケーションの実現はユビキタス環境に不可欠である。

本論文ではユビキタス環境の中で、ヘテロジニアスなセンサ群の組み合わせによる位置取得システムを提案する。今後構築されるユビキタス環境では、さまざまな種類のセンサが混在すると考えられ、これは位置を取得可能なセンサにおいても例外ではない。しかし、数種類のセンサを包括的に扱えるシステムは少ない、本研究では様々なセンサが混在可能な位置取得システムを構築し、位置に依存したアプリケーション、または位置情報が加わることによって現実世界におけるユーザの行動を支援可能なアプリケーションの容易な実現を目的とする。

Location System under Heterogeneous Sensor Environment

Shun Aoki¹ Takeshi Iwamoto² Junichi Yura² Hideyuki Tokuda¹²

¹ Faculty of Environmental Information, Keio University

² Graduate School of Media and Governance, Keio University

Under ubiquitous computing environment, many applications need to obtain user and device location. For example, when a user moves from a living room to a kitchen, an application must change services to continuously support the user's task such as watching TV.

Many location systems which currently exist realizes high precision. However, when an application which is build for a specific location system moves to another environment that deploys a different location system, the application cannot be used. This is because location systems, which are designed for specific sensors or environments, cannot provide appropriate location information. Moreover, many applications do not need high precision.

This system provides applications with location information to that does not depend on the environment. This system hides the diversity of the sensors by abstracting sensor data, and providing a unified API.

1 はじめに

近年の計算機の小型化、ワイヤレス技術の発展にともない、あらゆる場所に埋め込まれた計算機やセンサがユーザを支援するユビキタス環境 [1] が注目を浴び、その環境を前提とした研究も多くなされている。

例えば、ネットワークに接続されたインターネット冷蔵庫や電子レンジなど、情報家電と呼ばれる機

器が注目されている。これらは今までは単体で動作していたが、ネットワークに接続されることによって冷暖房を家の外から制御するなどの新たな機能が追加され、ユーザを支援できるようになった。

ユビキタス環境では、位置情報を必要とするサービスは増えると予想される。いたるところに PC やセンサが埋め込まれているため、ユーザの現在の位置に依存したサービスが考えられる。ユーザの周囲

に存在するサービスの検索や位置情報取得のためのセンサシステムはユビキタス環境実現のために重要となる。

ユビキタス環境下における位置情報取得システムは、センサのヘテロジニティに対応する必要がある。本稿では、ユビキタス環境内の多様なセンサやセンサシステム、既存のセンサシステムを統合的に扱い、アプリケーションに統一した利用方法を提供するシステムの設計と実装について述べる。

2 背景と目的

2.1 ユビキタス環境

ユビキタス環境の大きな特徴として、空間にPCやセンサが埋め込まれており、ユーザやアクチュエータが小型PCやセンサを持っていることが挙げられる。この特徴を生かすために、今まで独立して動作していたデバイス同士が結びつくことによる協調動作や、異なるデバイスによるサービスの継続性など新たな機能が考えられる。しかし、各空間を構成する機器やセンサ、ユーザが保持する小型PCやセンサはそれぞれ異なる。また、各空間内においても様々なデバイスが混在する。よってそれらの違いを吸収し、統一した利用方法を提供するミドルウェアが必要になり、その技術がユビキタス環境を支える基盤となる。

2.2 既存の位置取得システム

既存の位置取得システムは多数存在し、固定のセンサ群に依存することにより、高精度を実現している。また、システムを構成するセンサ群を用いて位置情報に伴う他の情報を提供している。既存の位置取得システムの研究の目標は、精度の向上と、位置情報以外の付随情報の提供の二点である。既存の位置システムをユビキタス環境下において使用すると、次のような問題が生まれる。固定センサ群に依存しているため、すでにセンサシステムが設置されている空間に新たなセンサシステムを導入した場合、既存のシステムの特徴や利用方法などが適用できない。この場合、新旧両方のセンサシステムが協調してさらに高い精度が得られるようなシステムである事が望まれる。また、ある位置システムが導入された空間用に構築したアプリケーションを、他の位置システムが導入されている空間で利用した場合、アプリケーションは動作しない。

これらの問題点は上記のユビキタス環境の特徴に対して相反する。固定センサ群への依存はセンサの混在という特徴を無視しており、設置の困難さはユビキタス環境導入の障害となり、拡張性のなさや、システムに依存したアプリケーションの要求は環境の多様性に対して対応不可能となる。

以上の理由で既存の位置取得システムは、ユビキ

タス環境下で利用することは困難である。

2.3 研究目的

ユビキタス環境に適した位置取得システムは、環境内の多様なセンサを利用し、アプリケーションに対して位置情報提供インタフェースを提供する必要がある。本研究では、ユビキタス環境の特徴である機器の混在と、位置情報の重要性、ユビキタス環境に適した位置システムの不足を考慮し、ヘテロジニアスなセンサ環境に対応した位置情報取得システムを構築する。図1にヘテロジニアスな位置センサ環境を示す。この例では数種類のセンサ、センサシステムが存在し、データの表現方法は座標、距離、2値情報と多様である。

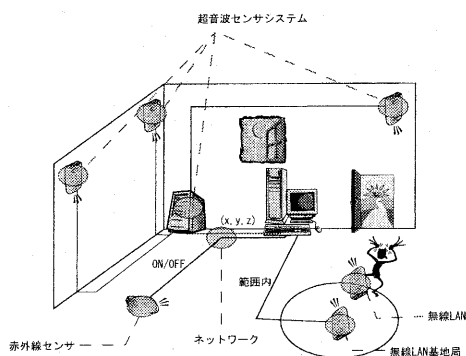


図1: ヘテロジニアスなセンサ環境

本研究では、既存の位置システムの問題を解決し、ヘテロジニアスなセンサ環境において、統一された位置取得方式を提供する。

2.4 想定アプリケーション

以下に本システムを利用するアプリケーション例をあげる。

● パブリックシアター

周囲のスピーカーとディスプレイを用い、ミニシアターを提供するアプリケーションを想定する。自分の部屋のような個人が占有可能な環境であれば部屋全体の空間からデバイスを検索し、公共の場のような多くの人が利用するような環境ではユーザ周辺の狭い範囲でデバイスを検索する必要がある。このように環境に適応して要求粒度を変更すべきアプリケーションは多く存在する。既存の位置取得システムであればこのようなアプリケーションごとの要求粒度とは関係なしに、一定の粒度の位置情報を提供したり、アプリケーションの要求する粒度を満たすことが不可能であれば、新たなシステムの導入が必要となる。ユビキタス環境下において、アプリケーションは環

境に応じて求める粒度を自由に変更できる必要がある。

● 動的ライト ON OFF システム

部屋に人が入るとライトが自動的に点灯するアプリケーションを想定する。アプリケーションは座標のような細かい粒度の位置情報を必要としない。既存のシステムを用いて実現する場合、部屋の範囲と人の座標を比べることにより、部屋内にいるかどうかをアプリケーション自らが判断する手法が考えられるが、これは細かい座標からの再計算が必要となる。

位置システムには、既に部屋の入り口に設置されている赤外線センサや圧力センサなどを利用して座標から再計算することなく部屋の出入りを監視し、アプリケーションに対して部屋全体という大きな粒度の位置情報を提供することが求められる。また、無線 LAN の範囲など粒度の粗い情報を利用して、求める情報を得ることが可能となれば、新たな位置システムの追加の必要がなくなる。このように、アプリケーションは部屋を一つのリージョンとする粒度を要求し、リージョンに人がいるかないかという目的に直結した単純な情報を得ることが望まれる。

● 仮想空間における遠隔制御を支える機器統合ミドルウェア

遠隔地から仮想空間で視覚的に機器を操作するためのミドルウェアを構築する際、実世界における機器の位置管理が必須になる。このようなミドルウェアはさまざまな環境で使用される。しかし、環境によって位置取得システムが異なる場合、ミドルウェアはそれぞれの位置取得システムに対応したインタフェースを用意する必要がある。位置情報を利用するミドルウェアが多くの環境で使用される場合、それぞれの環境の位置システムを統一的に扱い、インタフェースを統合するシステムの存在は必須となる。本システムはさまざまなセンサ、センサシステムを扱うことにより、環境の混在におけるインタフェースの統合的役割もはたす必要がある。

3 研究概要

ユビキタス環境において位置を取得可能な機器として、既存の位置取得システム、センサ、一般的に位置取得に用いられていないが、副次的な機能として位置取得可能であるものが挙げられる。以降、これらの位置を取得可能なものを位置取得可能機器と呼ぶ。位置取得可能機器には多くの種類が存在する。既存の位置取得システムは ActiveBat[2] や、Cricket[3] など多く開発されており、センサは赤外線センサ、超音波センサ、圧力センサなどが存在し、

疑似センサは無線 LAN 基地局などが存在する。これらの多様な位置取得可能機器を扱い、位置情報を提供可能とする。

本研究では、ユビキタス環境下におけるアプリケーションの位置要求の多様性として、知りたい位置情報の要求粒度の多様性と、位置の要求方法の多様性を扱う。例えば、アプリケーションごとや、同じアプリケーションにおいても環境の違いで、1m という単位で位置を知りたい場合があれば、50cm という単位で位置を知りたい場合もあり、要求粒度が異なる。また、あるセンサの位置を知りたい場合があれば、ある場所に人が入ってきたかを監視したい場合もある。このように要求方法は多様である。本システムにおいて、要求粒度の違い、要求方法の違いを考慮にいったアプリケーションインタフェースを構築する。

3.1 解決すべき問題点

ユビキタス環境に適した位置取得システムを構築するために、位置取得可能機器の多様性、位置要求の多様性を解決する必要がある。

位置取得可能機器の多様性

それぞれの位置取得可能機器は把握する有効範囲が異なり、位置取得可能機器から送られるデータは存在/非存在を表す 2 値情報、距離、座標のデータタイプが存在する。また、データタイプが距離、座標であった場合は、データの精度も位置取得可能機器ごとに異なる。本システムにおいて、さまざまな有効範囲と、データタイプ、データの精度に対応する必要がある。位置取得可能機器の有効範囲を抽象化し、データをデータタイプと精度をもとに本システムにおける表現方法で扱う。

位置要求の多様性

それぞれのアプリケーションは要求粒度が異なり、要求方法が多様である。位置情報取得可能機器と同様に、アプリケーションによる位置要求の多様性においても本システム内で扱う際の方式を決定する必要がある。

3.2 抽象化モデル

本システムでは、位置取得可能機器のさまざまな有効範囲を数種類の基本立体からなる 3 次元空間にマッピングする。また、アプリケーションの要求粒度を立方体の集合からなる 3 次元空間にマッピングする。この 2 つのマッピングによって抽象化を行う。

位置取得可能機器をマッピングするための有効範囲を決定する要素として、指向性、始点の形状、最大距離がある。指向性とは位置取得可能機器が値を取ることができる角度であり、始点の形状とは、位置取得可能機器が表すデータの基準点の形状であり、最大距離とは位置取得可能機器が値を取ることがで

きる最大の距離である。この3つの要素からマッピングする基本立体を決定する。これらの基本立体がマップされる3次元空間を DataAbstractionArea と呼ぶ。このマッピングによって、様々な位置取得可能機器の有効範囲を抽象化し、一つの仮想空間で表現する。位置取得可能機器の範囲をマッピングした基本立体の上で、データを扱う。

また、本システムはアプリケーションに対して、一定の粒度の立方体で構成された3次元空間を提供する。この立方体の粒度はアプリケーションによって自由に決定出来る。この3次元空間を ReqCubeArea と呼ぶ。アプリケーションは、DataAbstractionArea の構成を気にすることなく、この ReqCubeArea の粒度を単位として自由に位置を取得できる。ReqCubeArea における単位立方体のことをリージョンと呼ぶ。

図2に本研究における抽象化手法を示す。

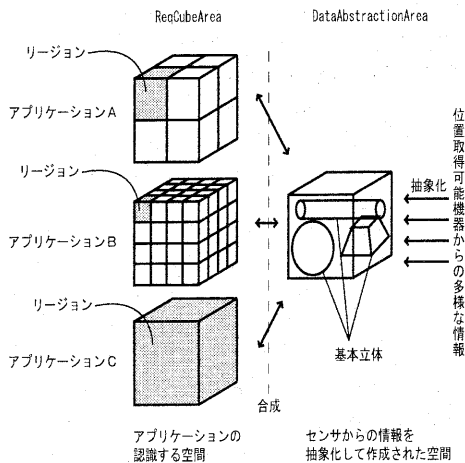


図2: 抽象化概念図

DataAbstractionArea 内では、有効範囲をいくつかの基本立体で表している。アプリケーション A, B, C に対して、各要求粒度で構成された立方体の集合で ReqCubeArea を構築している。この DataAbstractionArea と ReqCubeArea を合成することにより、本システムでは位置取得可能機器の有効範囲をアプリケーションの要求粒度へと変換する。

ヘテロジニアスなセンサ環境において、ReqCubeArea 内のリージョンの粒度に対して DataAbstractionArea 内の各位置取得可能機器の有効範囲が充分であるとは限らない。その場合、位置が単一のリージョンに決定できないため、該当するリージョンのすべてを通知する。

3.3 要求方法の違いへの対応

アプリケーションは位置を要求する際、さまざまな要求方法が考えられる。本システムでは、アプリケーションに対して、さまざまな要求方法に対応した API を提供することにより対応する。

4 設計

本章では、本システムを利用する位置情報取得可能機器に必要とされる情報、本システム内での位置取得可能機器のデータの扱い方を述べたあと、本システムのシステム構成を述べ、最後にシステムの動作例をあげる。

4.1 位置取得可能機器の必要情報

本システムへ接続される位置取得可能機器は本システムが Character-Info, Establish-Info と呼ぶ2つの情報を明示する必要がある。この2つの情報は、位置取得可能機器が本システムに接続する際に通知する。

Character-Info

Character-Info は位置取得可能機器の始点の形状、最大距離、指向性、データタイプから構成され、必要があればデータ精度も含む。始点の形状は点、円、長方形、最大距離は長さ、指向性は角度、データタイプは2値情報、距離、座標、データ精度は長さで表される。Character-Info は基本立体を決定する要因として利用される。

Establish-Info

Establish-Info は位置取得可能機器の設置の際の情報であり、始点の頂点座標情報と向き情報から構成される。これらを数値で指定する。この Establish-Info をもとに、本システムは基本立体を DataAbstractionArea にマッピングする。

4.2 位置取得可能機器抽象化方法

本システムでは、位置取得可能機器が最初に本システムに接続される際に通知される Character-Info と Establish-Info を用いて、初期化作業として次のことを行う。

- 位置取得可能機器の有効範囲をもとに基本立体を生成
- DataAbstractionArea 内に基本立体をマッピング
- 基本立体を複数のオブジェクトに分割

この作業によりデータを統一的に扱うことを可能とする。次に基本立体の生成方法、基本立体の分割方法について述べる。

4.2.1 基本立体の決定

本システムでは Character-Info 中の始点の形状、最大距離、指向性から基本立体を決定する。本システム内では、始点の形状を、点、円、長方形に限定する。各情報と、基本立体の形状を図3に示す。本

システムで扱う位置取得可能機器は、範囲をこの形状に抽象化する。

始点の形状	点	円	長方形
基本立体			
位置取得可能機器例	・超音波センサ ・赤外線センサ ・無線LAN基地局	・焦電型 赤外線センサ ・既存の位置提供システム	・圧力マット ・既存の位置提供システム

最大距離：L
指向性： θ

図3: 基本立体の形状

4.2.2 データタイプの取り扱い

本システム内で扱うことが可能なデータタイプは2値情報、距離、座標である。距離、座標の場合はデータ精度を明示する必要がある。データタイプが2値情報の場合は基本立体を一つのオブジェクトとして、データタイプが距離の場合は基本立体を始点からデータ精度で分割した複数のオブジェクトとして、データタイプが座標の場合は基本立体をデータ精度を一辺とする立方体で近似した複数のオブジェクトとして扱う。最終的にすべてのデータタイプにおいて、データは各オブジェクト毎の存在/非存在を表す2値情報で表される。図4にオブジェクト分割の概念図を示す。

データタイプ	ON or OFF	距離	座標
オブジェクト分割方法			
位置取得可能機器例	・赤外線センサ ・圧力センサ	・超音波センサ ・レーザー	・既存の位置取得システム

データ精度：A

図4: オブジェクトの分割

以降このオブジェクト一つを単位オブジェクトと呼ぶ。

本システムでは、単位オブジェクトに分割することにより異なるデータタイプの位置取得可能機器を

統一的に扱うことを可能とする。

4.3 システム構成

本システムのモジュール構成を図5に示す。

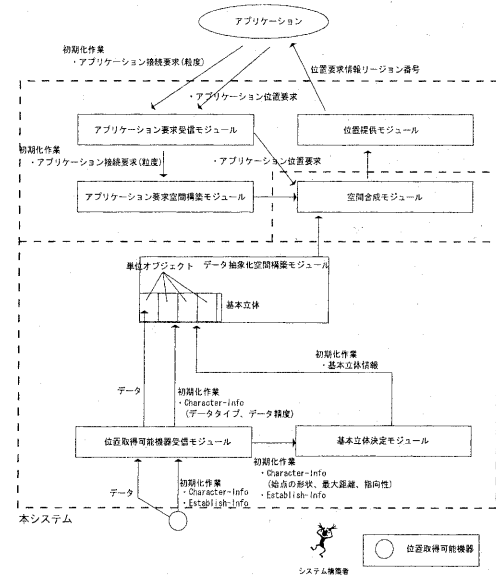


図5: システム構成図

各モジュールを DataAbstractionArea の構築方法, ReqCubeArea の構築方法, 両空間の合成方法より説明し、最後にアプリケーションインタフェースについて述べる。

4.3.1 DataAbstractionArea の構築

本システムにおいて、位置取得可能機器からのデータは位置取得可能機器受信モジュールで受け付ける。位置取得可能機器が本システムに最初に接続する際、Character-InfoとEstablish-Infoを通知する。Character-Infoから、始点の形状と、最大距離、指向性を基本立体決定モジュールへと送り、基本立体を決定する。また、同時に送られるEstablish-Infoをもとに、基本立体をDataAbstractionAreaのどこにマッピングするかを決定し、データ抽象化空間構築モジュールにおいてDataAbstractionArea内に基本立体を形成する。Character-Infoのうち、データタイプとデータ精度はデータ抽象化空間構築モジュールに通知され、基本立体を単位オブジェクトに分割する。

以降、位置取得可能機器からのデータは、単位オブジェクトごとの2値情報で表される。

4.3.2 ReqCubeArea の構築

アプリケーションは本システムに接続する際、アプリケーション要求受信モジュールに要求粒度を通

知する。アプリケーション要求粒度はアプリケーション要求空間構築モジュールに通知され、粒度を一辺の長さとする立方体の集合で表される、ReqCubeArea を構築する。

4.3.3 空間の合成

アプリケーションの位置要求ごとに、空間合成モジュールで ReqCubeArea と DataAbstractionArea を合成する。アプリケーションが位置情報を要求した場合、合成した空間をもとにリージョン番号を用いて位置情報をアプリケーションに通知する。

4.3.4 アプリケーションインタフェース

アプリケーションは本システムに位置の要求を出すとき、以下の要求をすることが可能である。実装言語である Java を用いた API 利用例と共に以下に挙げる。

本システムへの接続要求

```
StartCube sc = new StartCube(int accuracy);
```

要求データ粒度と共に、本システムへ接続する。

センサ名から位置の要求

```
sc.getDevicePosition(String sensorName);
```

センサの名前を指定し、現在どのリージョンにいますかを要求する。

センサ名から周囲のデバイスを検索

```
sc.getArroundDevice(String sensorName);
```

センサ名を指定し、その周囲の他の要求対象物を検索する。

リージョン番号から周囲の対象物を検索

```
sc.getArroundDevice(int resionNo[][][]);
```

リージョンを指定し、そのリージョンを含む周囲のリージョンから要求対象物を検索。

リージョン番号からリージョン内の対象物を検索

```
sc.getDeviceName(int resionNo[][][]);
```

リージョンを指定し、そのリージョン内の要求対象物を検索。

基準となるリージョン番号の設定

```
sc.setBasicPoint(int resionNo[][][]);
```

基準となるリージョンを指定し、以降そのリージョンからの相対リージョン番号で位置情報を取得する。

4.4 システム動作例

本システムはアプリケーションの要求粒度から構成される DataAbstractionArea と、Character-Info と Establish-Info から構成される ReqCubeArea の二つの仮想空間を構築する。本節ではこの仮想空間の構築手順を、アプリケーションによる利用方法に基づいて述べる。

ReqCubeArea の構築

アプリケーションは本システムに接続要求を出すと同時に粒度を通知する。本システムは、アプリケーションより取得した粒度情報から、空間をリージョンに分割し、ReqCubeArea を構築する。

アプリケーションの利用方法

アプリケーションは本システムが提供する API に基づいて位置情報を要求する。例えば特定のユーザの位置を知りたい場合は、アプリケーションは本システムが提供する API に基づいて対象ユーザが持っているセンサの名前を本システムに要求する。本システムは DataAbstractionArea 内の単位オブジェクトの中から、要求するセンサのデータがマッピングされた単位オブジェクトを検索する。その後、ReqCubeArea 内のどのリージョンに目的の単位オブジェクトが存在するかを計算し、アプリケーションにリージョン番号として通知する。単位オブジェクトが複数のリージョンにまたがっている場合は、対象となるすべてのリージョン番号をアプリケーションに通知する。

5 実装

本実装は、Windows2000 上で、J2SDK1.4.0 を用いて行った。センサ情報を抽象化する基本立体は、直方体に限定して行った。さらに DataAbstractionArea と、ReqCubeArea を視覚的に確認するための GUI を Java3D ライブラリ [4] を用いて実装した。

ReqCubeArea の実装

本システムは、アプリケーションが要求する粒度に基づいて空間を立方体に分割し、各立方体を一つのリージョンとして構築する。さらに、各立方体ごとにハッシュテーブルを用意し、すでに存在するセンサ情報に基づきデータの格納場所を用意し、データをセットする。新たにアプリケーションが接続する度にこの作業を繰り返す。

DataAbstractionArea 構築の実装

センサ情報を表現する基本立体を、始点の形状を長方形、最大距離を部屋の長さ、指向性を 0 度に限定して DataAbstractionArea を構築した。また、データタイプは 2 値情報に限定した。センサが本システムに接続される度に、基本立体を DataAbstractionArea に追加する。アプリケーションが追加される時と同様に、追加の際にすでに存在するアプリケーションのハッシュテーブルにセンサデータ領域を確保し、データをセットする。新たにセンサが追加される度にこの作業を繰り返す。

空間合成の実装

構築した二つの空間を合成することにより、基本立体と各立方体が交差しているかを判断し、交差し

ている立方体を位置提供リージョンとみなす。
一連の動きを表した GUI の様子を図 6 に示す。

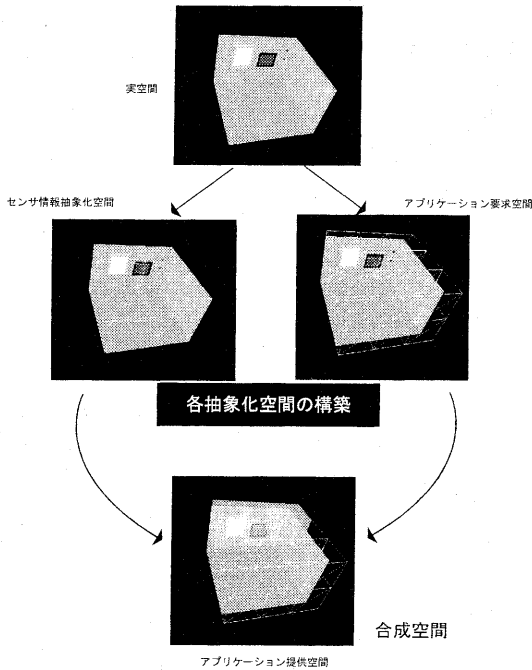


図 6: プロトタイプ実装の GUI

本実装によりアプリケーションの要求粒度に対応した形で位置情報を提供可能とし、それを視覚的に確認可能となった。

6 評価

アプリケーション要求から、その要求に適した形で位置情報を計算するまでの時間を把握するための定量的評価を行った。本システムが利用するセンサの個数、アプリケーションの要求粒度を変化させることにより、これらの値と本システムの負荷の関係測定した。

6.1 測定環境

測定を行ったハードウェアおよびソフトウェアの環境を表 1 に示す。

表 1: 測定環境

CPU	PentiumIII 1200MHz
主記憶	768MB
OS	linux 2.4.18-0v13
JavaVM	J2SDK1.4.0

6.2 センサ数とアプリケーション粒度における交差判定時間

本システムでは、ReqCubeArea を構成する立方体と、DataAbstractionArea 内の基本立体の交差判定を行い、アプリケーションに提供する位置情報のリージョンを計算する。この交差判定の実行速度を決定する要因として、以下の要素が考えられる。

- ReqCubeArea の粒度
- 位置取得可能機器の数
- 基本立体の形状
- 基本立体を占める単位オブジェクトの数

本測定では、アプリケーションの要求粒度と、センサ、センサシステム数における実行速度の変化を測定した。

実際には 540cm x 540cm x 360cm の直方体の空間において、ReqCubeArea の粒度を 60cm, 80cm, 100cm, 120cm, センサノード数を 1 から 4 と変化させて行った。表 2 に測定結果を示す。

本測定により、要求粒度、センサ個数が増加することにつれて、交差判定時間が増加していることがわかる。空間を構成するリージョン数は要求粒度に対して三次関数的に増加するので、要求粒度の細かさは、リージョン数において大きく反映される。今後、空間合成における交差判定の計算方法を工夫する必要がある。

60 センチ単位で位置を取得できれば、人の動き、ものの動きを判断する際、実世界の動きを判断可能である場合が多い。以上のことをふまえて、測定結果よりユビキタス環境下において利用可能であると考える。

7 関連研究

既存の位置取得システムを、センサのヘテロジニティに対応しているか、していないかに分類し、前者の例として、ActiveBat と Cricket, 後者の例として POIX を挙げる。

7.1 The Active Bat Location System

ActiveBat[2] はオフィスなどの環境を想定した超音波センサを用いた位置システムである。位置を知りたい対象に超音波を発信する小型デバイスを取り付け、天井に設置された複数の超音波受信機からの距離より位置を算出する。誤差は 10 センチ以内で取得可能であり、これらの位置情報を用いて、ユーザに位置適応的なサービスを提供する。

しかし、Active Bat は、天井に複数の超音波センサを設置する必要があり、設置の際に多大な労力を必要とする。また、Active Bat は特定の超音波センサに特化してつくられており、他のセンサの利用は考慮に入れていない。したがって、アプリケーションは ActiveBat に依存した形で書く必要がある。

表 2: 測定結果 (ms)

要求粒度 (cm)	120	100	80	60
センサ個数 (個)	(5x5x4)	(6x6x4)	(7x7x5)	(10x10x7)
1	2.05	3.9	6.1	14.8
2	2.05	8.8	5.25	22.2
3	3.35	12.15	9.15	24.7
4	4.65	13.15	15.5	54.3

7.2 Cricket

Cricket[3] は超音波と無線を用いた、センサのヘテロジニティに対応していない位置取得システムである。屋内での位置取得を想定している。ActiveBat との大きな違いはユーザが送信機を持ち、これによりプライバシーの保護が可能となることである。

Cricket beacon と呼ばれる機器を建物内に複数配置し、無線を超音波を定期的に発信する。Cricket compass と呼ばれる機器を小型機器につけ、compass の位置を取得する。無線は同期を取るために用い、無線を受信してから超音波を受信するまでの時間差によって、距離を求める。また、超音波センサを複数用いることにより、向き情報も取得できる。

想定アプリケーションとして、Cricket compass から距離と方向を指定し、その範囲内にあるデバイス一覧を取得する ViewFinder、位置情報と向き情報を利用して、ユーザに建物内の道案内をする WayFinder などがあげられる。

しかし、ユーザが送信機を持っていることにより、送信機の向きと天井に設置される CricketBeacon の配置により位置情報が取れなくなる場合がある。また、ActiveBat と同様に、設置の困難さがあげられる。

7.3 POIX

POIX[5] は XML で記述されたインターネット上における位置情報の交換フォーマットである。位置の表現だけではなく、位置を中心として様々な情報を包括的に提供可能である。例えば、対象が移動体であった場合、移動手段、移動速度、移動方向などが提供される。位置は、緯度、経度、平面誤差、高度、高度誤差で表現される。

しかし、想定以外のデータで表現されるセンサを追加することが不可能となるため、緯度、経度などで表すことができない位置情報は対象外となる。ユビキタス環境下において、緯度、経度で表すことができない位置取得可能機器は多数存在する。また、閉鎖的な空間を想定していないので、ユビキタス環境における位置記述フォーマットとしては不十分である。また、アプリケーション要求の多様性は想定していない。

7.4 まとめ

既存の位置取得システムは、特定のセンサ群に依存しているか、複数のセンサデータに対応しているか、対応しているデータタイプが少なかつたりアプリケーション要求の多様性を視野にいれていないシステムが多い、今後のユビキタス環境の発展を視野にいれた、センサデータのヘテロジニティ、アプリケーション要求のヘテロジニティの両方に対応した位置取得システムは少ない。

8 まとめ及び今後の課題

本稿ではユビキタス環境下の閉鎖的空間における位置情報取得システムを提案した。ヘテロジニアスなセンサ環境を扱うための抽象化モデル、アプリケーションが環境非依存に構築可能とするためのアプリケーション要求方式について述べた。これにより、ユビキタス環境に適した位置取得システムが構築できる。

今後の課題としては、より多くのセンサ情報を表現可能な、抽象化基本図形の更なる考察を行う。また、位置情報と共に利用される機会の多い、対象物の動きに関する情報を位置情報と共に本システムにおいて利用可能とする。

参考文献

- [1] Weiser, M.: The Computer for the Twenty-First Century, *Scientific American*, pp. 94-10 (1991).
- [2] Andy, W., Alan, J. and Andy, H.: A New Location Technique for the Active Office, *IEEE Personal Communications*, Vol. 4, No. 5, IEEE, pp. 42-47 (1997).
- [3] Nissanka, B. P., Anit, C. and Hari, B.: The Cricket Compass for Context-Aware Mobile Applications, *Proc. 7th ACM MOBICOM* (2001).
- [4] Microsystems., S.: Java 3D(TM) API Home Page. <http://java.sun.com/products/java-media/3D/>.
- [5] MOSTEC: POIX Point Of Interest eXchange language. <http://mostec.aplix.co.jp>.