

修士論文 2009 年度 (平成 21 年度)

**uZi：虫眼鏡レンズのメタファを用いたユビキタス
コンピューティング環境における情報可視化手法**

慶應義塾大学大学院 政策・メディア研究科
今枝 卓也

修士論文要旨 2009 年度 (平成 21 年度)

uZi：虫眼鏡レンズのメタファを用いた ユビキタスコンピューティング環境における情報可視化手法

論文要旨

近年、ユビキタスコンピューティング環境の発展に伴ない、実世界に係わる様々な仮想空間上の情報が日常の生活空間で増加しつつある。具体的には、センサから得られる情報が挙げられる。無線センサノードを無線ネットワークで接続する事で構築される無線センサネットワーク研究の発展により、研究レベルでの提案から製品化まで含めて様々な無線センサノードが登場している。これらの無線センサノードからのセンサデータを収集する事で様々なアプリケーションが可能になる。例えば、食料品の安全を保証するトレーサビリティシステムや、幼児や高齢者の見守りシステム等である。また、従来からある Web サービスは仮想空間の情報のみを扱うものが大半であったが、時刻表や Twitter のようなコミュニケーションサービスのように、現実空間と密接に関連するサービスが登場してきた。

しかし、これらの現実空間に係わる情報を可視化するためには従来の手法ではユーザに不連続な操作を強制する。例えば無線センサノードから情報を得るためには、従来の手法ではデータベースに直接アクセスしたり、Web インターフェースを操作する等の操作をする。現実空間に関する Web サービスでも同様に、自分の身の回りの情報を得るためにも PC に対する操作を必要とする。このような目の前の情報を得るためにも PC に対する操作を必要とする不連続な操作インターフェースはユーザにとっても不自然であると考えられる。

本稿では、上記の問題を解決するために、携帯端末と AR 技術を用いた情報可視化システムである uZi システムを提案する。携帯端末上で動作する事により、ユーザはどこにいても現実空間に係わる情報サービスを受けられる。また、AR 技術を用いる事で、現実空間と仮想空間を関連づけて情報を取得できる。uZi は現実空間上に存在する様々な情報源やユーザの要求する様々な情報の種類に対応するため、レイヤ構造を持つフィルタリング機構を持つ。uZi を用いる事で、ユーザは連続的な動作で現実空間上の情報を可視化できる。

本稿の手法を評価するために、従来の手法と比較しながらユーザビリティ評価を行い、本稿の手法の有用性を確認した。また、本稿の手法が携帯端末で十分に動作可能であることを示すために、パフォーマンス評価を行った。

キーワード

Augmented Reality, ユビキタスコンピューティング, フィルタ, 可視化

慶應義塾大学 政策・メディア研究科
今枝 卓也

Abstract of Master's Thesis Academic Year 2009

uZi : Information Visualization Techniques in Ubiquitous Computing Environment using Magnifying Glass Lens Metaphor

Summary

In recent years, Information related to the spatial location in the real world has increased as the development of ubiquitous computing environments. In particular, there is much information obtained from the sensor. The development of researches in wireless sensor networks enables various applications. For example, traceability system to ensure safety of food and the system to watch the children and the elderly is proposed by many researchers. Also, the traditional Web services do not deal only with virtual information, but also information related to real space such as timetable of train and communication services such as Twitter.

However, traditional approach to visualize information related to the spatial location in the real world will force users discrete operations. When using conventional methods to obtain information about their surroundings from wireless sensor nodes, for example, users must directly access to databases or operate on Web interface on a PC. It is unnatural for users to use the methods to get information in front of them.

In this paper, in order to solve the above problems, I developed uZi system, visualization system using mobile computer and AR technology. By running on a mobile device, the users can receive information services from anywhere. In addition, using AR technology, the users can associate virtual information with the spatial location in the real space. uZi has layer structured filtering mechanism to deal with various type of information. The users can visualize spatial information with continuous operation.

Keywords

Augmented Reality, Ubiquitous Computing, Filter, Visualization

**Keio University Graduate School of Media and Governance
Takuya Imaeda**

目次

第 1 章	序論	1
1.1	本論文の背景	2
1.2	本論文の目的	3
1.3	本論文の構成	4
第 2 章	現実空間上に関連した情報可視化手法	5
2.1	現実空間に関連した情報	6
2.1.1	センサ	6
2.1.2	情報家電・デバイス	7
2.1.3	Web 上の情報	7
2.2	情報可視化手法	8
2.2.1	PC ベースの情報可視化手法	8
2.2.2	現実空間に即した情報の可視化手法	8
2.2.3	既存手法の問題点	10
2.2.4	虫眼鏡レンズのメタファを用いた情報可視化手法	12
2.3	本章のまとめ	12
第 3 章	AR 技術と携帯端末を用いた情報の可視化	13
3.1	プロトタイプ	14
3.1.1	概要	14
3.1.2	予備実験と考察	17
3.2	uZi システム	18
3.2.1	アプローチ	18
3.2.2	物体情報	19
3.3	本章のまとめ	20
第 4 章	レイヤ構造を持つフィルタリング機構	21
4.1	フィルタリング機構	22
4.1.1	従来の AR アプリケーションにおけるフィルタリング機構	22
4.1.2	レイヤ構造フィルタリング機構	23

4.2	レイヤ	24
4.2.1	入力レイヤ	24
4.2.2	時系列レイヤ	25
4.2.3	関連度レイヤ	26
4.2.4	絞り込みレイヤ	26
4.2.5	出力レイヤ	26
4.3	レイヤ間の連携	27
4.4	本章のまとめ	30
第 5 章	uZi の設計	31
5.1	ハードウェア構成	32
5.2	ソフトウェア構成	32
5.3	本章のまとめ	33
第 6 章	uZi の実装	34
6.1	ハードウェア	35
6.2	ソフトウェア	35
6.3	本章のまとめ	36
第 7 章	評価	37
7.1	定量的評価	38
7.1.1	描画速度評価	38
7.1.2	センサ情報アクセス速度評価	39
7.1.3	uZi 速度評価	40
7.1.4	フィルタリング機構	41
7.2	ユーザビリティ評価	42
7.2.1	第一回ユーザビリティ評価	42
7.2.2	第二回ユーザビリティ評価	43
7.3	本章のまとめ	44
第 8 章	結論	45
8.1	まとめ	46
8.2	今後の展望と課題	46
8.2.1	ビジュアルマーカを用いない物体の認識	46
8.2.2	フィルタ追加機能	46
8.2.3	ビジュアルマーカと物体の紐付けの自動化	47
参考文献		49

図目次

1.1	地図上にマッピングされたセンサ情報	3
2.1	センサノード 左から Mica MoteZ, uPart, SunSPOT	6
2.2	AR アプリケーションの例	9
3.1	ビジュアルマーカ	15
3.2	実行時画面	16
3.3	プロトタイプシステム利用者	17
3.4	スマートフォン 左が iPhone 3GS, 右が Android Dev Phone	19
3.5	uZi の動作の様子	19
3.6	物体にビジュアルマーカとセンサが貼り付けられている様子	20
4.1	従来のフィルタリング機構 各フィルタが並列に存在するため, 再利用性を阻害する.	23
4.2	uZi のフィルタリング機構	24
4.3	フィルタリング機構の構造	25
4.4	テキスト表示フィルタ	27
4.5	グラフ表示フィルタ	28
4.6	抽象表示フィルタ	28
4.7	フィルタ選択画面	29
5.1	ハードウェア構成図	32
5.2	ソフトウェア構成図	33
7.1	センサ情報アクセス速度	40

表目次

3.1	現実空間の認識手法の比較	15
6.1	uZi ハードウェア環境	35
6.2	uZi 実装環境	36
7.1	実験に用いた端末のスペック	38
7.2	iPhone 3GS と Android DevPhone 1 における NyARToolkit 実行速度	39
7.3	センサデータベースのハードウェア・ソフトウェア環境	39
7.4	uZi 実行速度	41
7.5	フィルタのソースコード行数比較	41
7.6	第一回ユーザビリティ評価アンケート結果	42
7.7	第二回ユーザビリティ評価アンケート結果	43

第 1 章

序論

本章では，はじめに本研究の背景と目的について述べ，次に本論文の構成について述べる．

1.1 本論文の背景

近年、ユビキタスコンピューティング研究の発展に伴ない、情報家電やセンサのような情報機器が人間の生活空間に普及しつつある。その結果、仮想空間の情報が現実空間に氾濫するようになってきた。例えば、慶応義塾大学徳田研究室では、SSLab プロジェクト [12] のようなユビキタスコンピューティング環境の実験環境を運用している。これらの実験環境をはじめとするユビキタスコンピューティング環境に存在する情報機器として、建築物や情報機器に埋め込まれたセンサや、環境内に展開された無線センサノードが挙げられる。従来から建築物や家電にセンサを埋め込む事は広く行われてきたが、特定のサービスでしか使用できない事が多かった。ユビキタスコンピューティング環境では、それらのセンサがネットワークに接続され、ユーザは様々な用途にセンサ情報を役立てられる。また、無線通信機能を持つ複数の無線センサノードで無線センサネットワークを構築する事で、環境内に埋め込む手間をかけずにセンサを設置したり、身の回りの物や人にセンサを取り付ける事が可能になる [18]。また、家庭やオフィスのコンピュータネットワークに接続された情報デバイスも普及しつつある。例えばネットワーク接続性を持ったテレビやエアコン等の家電や、デジタルフォトフレームのような小型情報端末が実際に製品として登場している。これらのセンサや情報デバイスから得られる情報は現実空間に密接に関連している。

また、従来から存在する Web サービスの中にも現実空間に関連付けられた情報がある。例えば、電車やバスの時刻表や運行情報のような情報は現実空間上のある地点における情報である。さらに、近年では手軽なコミュニケーションサービスとして Twitter が注目されている。Twitter 上でも自分のいる位置を GPS で知らせたり、「～なう」（今ここにいる）といった形で現実空間と結びつけられる情報がユーザ間でやり取りされる。これらの Web サービスも近年の情報機器の小型化や偏在化と無線ネットワークの展開により、いつでもどこでも利用可能な環境が実現している。

これらの現実空間と関連付けられた仮想空間上の情報を可視化しユーザに提示する事で、ユーザは様々な恩恵を受けられる。センサから得られる情報の場合、アプリケーション例として、食料品の安全を保証するトレーサビリティアプリケーションや、環境の状態や情報機器の使用状況を把握するためのモニタリングアプリケーションが挙げられる。情報デバイスから得られる情報を可視化する事で、ユビキタスコンピューティング環境の発展により複雑化した情報サービスをユーザが理解するための助けとなる事が期待できる。また、Web 上のサービスは Web ブラウザにより可視化される。多くの情報サービスが現在この形で提供されている。しかし、従来使われてきた仮想空間上の情報を可視化する手法を現実空間に関連付けられた仮想空間上の情報に適用すると、ユーザに対して面倒な操作を強要する事になる。具体的には、例えばセンサの情報を得るためには PC での操作を行う必要がある、といった事が起こる。通常、センサから得られた情報はネットワーク上のサーバにあるデータベースに蓄えられる。このデータベースにアクセスするためには、ユーザはコンソールから直接データベースを操作したり、Web インターフェース等を用いて PC 上で数値やグラフを読み取るといった作業が必要になる。この時、例えばユーザが目目の前にあるセンサのデータを調べようとした場合、ユーザは

1. センサの ID を調べる
2. コンソールや Web インターフェースに対してデータの出力要求を出す
3. 出力された数値やグラフを読み取る

という手順を踏む事になる。このような手順は面倒であるとともに、データベースやコンピュータに関する知識が無いユーザにとって敷居が高い。また、Web 上の情報においても、現実空間に関連した情報に関して、PC の Web ブラウザでは現実空間との関わりを把握しづらいという問題がある。特定の駅の時刻表や特定の場所における Twitter のつぶやきを PC のディスプレイ上で文字情報で表現しても、自分と目的の情報の空間的な位置関係がわかりづらい。図 1.1 に示すように、地図上にセンサ情報をマッピングするという手法 [19][8] も用いられるが、この手法ではユーザはセンサや自分のいる位置を意識する必要がある。目の前にあるセンサのデータを調べるためにも地図から特定の位置を読み取る必要があり、ユーザにとって負担を強いる事になる。

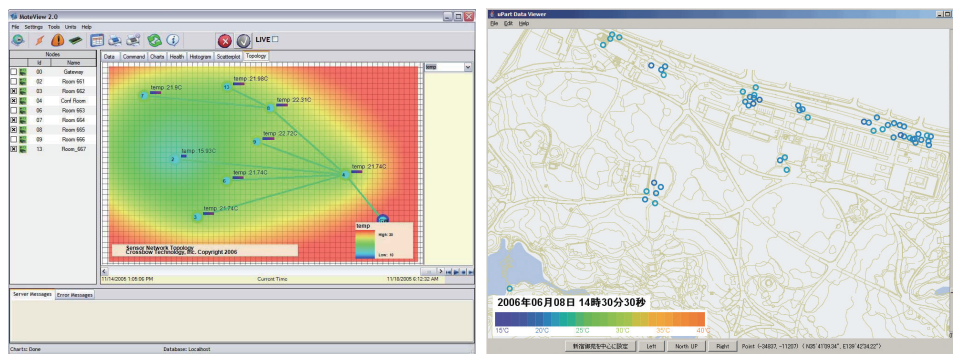


図 1.1 地図上にマッピングされたセンサ情報

1.2 本論文の目的

本稿は、ユーザに対して特殊な知識や不連続な動作を要求しない形で実空間に関連付けられた仮想空間上の情報にアクセスできるようにする事が目的である。そこで、本稿ではこの目的を達成するために、uZi システムを構築する。uZi はユーザの多くが持っていると考えられる携帯電話のような携帯端末をユーザデバイスとし、Augmented Reality (AR) 技術を用いて情報を可視化する。AR 技術とは、現実空間の映像上に仮想空間の情報を投影する技術である。AR 技術を用いる事で、現実空間の映像の特定の位置や物に、それに関連付けられた仮想空間上の情報を描画できる。ユーザは映像を見るだけで自分の周りにある情報を閲覧可能になる。

また、ユビキタスコンピューティング環境では、センサ情報や情報デバイス情報、Web からの情報等、非常に多くの種類の情報が存在する。さらに、それぞれの種類毎に膨大な情報が蓄積される。例えばセンサ情報の場合、通常データベースに過去からの情報が蓄積される。AR 技術を用いて情報を可視化する場合、このような膨大な情報を一度にユーザに提示する事は不可能であるだ

けでなく、ユーザを混乱させてしまう。ユーザが必要としている情報はその時々コンテキストによって変わる。そのため、本稿において目的としているシステムにはユーザが必要な情報を選択できるようにする仕組みが必要である。uZiはこの要求をレイヤ構造を持つフィルタリング機構を導入する事で解決する。センサ情報を取得するフィルタや Web 上の情報を取得するフィルタを切り替える事で様々な情報に対応できるようになる。さらに、これらのフィルタを開発する際の負担を低減するため、フィルタリング機構にレイヤ構造を持たせ、フィルタの再利用性を高める。

1.3 本論文の構成

本稿では、まず第 2 章で本研究の関連研究について述べ、その問題点を指摘し、uZi の機能要件をまとめる。次に、第 3 章で Augmented Reality (AR) と携帯端末を用いた情報可視化手法を述べた後、第 4 章で uZi のフィルタリング機構の仕組みについて説明する。第 5 章では uZi の設計を行い、第 6 章で uZi の実装について述べる。第 7 章で uZi を評価し、第 8 章で本稿をまとめる。

第 2 章

現実空間上に関連した情報可視化手法

本章では，まず現実空間に関連した仮想空間上の情報について考察する．次に，既存の情報可視化手法を挙げ，それらの問題点を指摘した後，本稿における情報可視化手法について述べる．

2.1 現実空間に関連した情報

本節では、本稿で提案するシステムにおいて可視化対象とする現実空間に関連付けられた仮想空間上の情報についてまとめる。

2.1.1 センサ

ハードウェア技術の進歩に伴ない、情報デバイスの小型化が進んでいる。ユビキタスコンピューティング分野では、無線センサノードが登場し、注目を集めている。既に製品として Crossbow の Mote[20] や Sun の SunSPOT[15] 等が登場している。また、より小型のセンサノードの研究・開発も進められており、uPart[2] のように 2cm 未満程度のサイズしか無い物もある。主な無線センサノードを図 2.1 に示す。

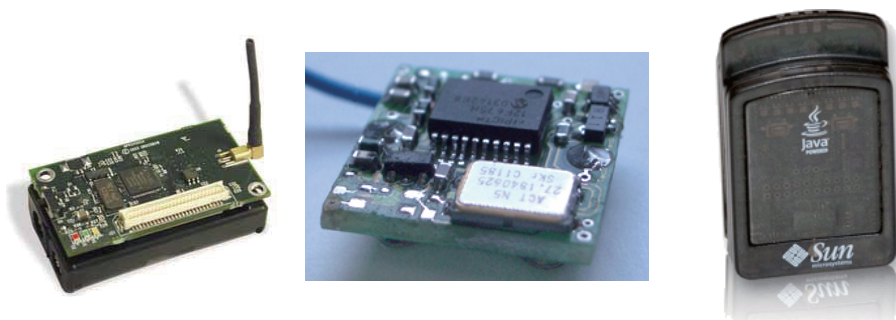


図 2.1 センサノード 左から Mica MoteZ, uPart, SunSPOT

このような小型化されたセンサハードウェアは日常空間へのセンサの埋込や身の回りの物へのセンサの取り付けを可能にする。また、無線センサノードは無線により他の無線センサノードやデータを収集するシンクノードと呼ばれるノードと接続し無線ネットワークを構築する。ネットワークを通じた協調動作により、無線センサノードは単体のセンサではカバーできないような広い地域を監視したり、移動する物体に取り付けるといった事を可能にする。これらのセンサから得られる情報に基づいたアプリケーションとして、環境モニタリングアプリケーションとトレーサビリティアプリケーションが考えられる。

環境モニタリングアプリケーションとは、ある環境内に複数の無線センサノードを配置し、環境の温度や照度等の状態変化を監視するアプリケーションである。室温や照度を監視する事で、環境をペットや観葉植物に最適な状態に保ったり、過去の状態を確認して異常な状態に陥っていなかったか確認できる。また、室外においても環境の調査目的で環境モニタリングを行う事がある。AiryNotes の例に見られるように、都市のヒートアイランド現象や緑化効果等の可視化に用いられる。

トレーサビリティアプリケーションとは、食料品の生産から消費者による消費までの経路をセン

サによって監視し、食料品の安全や品質を確認できるようにするアプリケーションである。従来のトレーサビリティアプリケーションは RFID タグの利用例として挙げられてきた。RFID を用いたトレーサビリティアプリケーションでは、生産地で RFID タグを食品に取り付ける。消費者の手に渡るまでの輸送経路において RFID タグがどの地点を通ったか記録する事で、消費者や食品小売店は食品の生産地を確認できる。この例では、生産地と輸送経路を確認する事で、食品の安全がある程度確保できると考えられる。しかし、実際にどのような状態で保存されたかは RFID だけではわからない。特に輸送上の保存状態による品質の劣化が問題となる高価なワインや食肉は経路確認だけでは不十分となる。例えば、ワインは生産地から消費地まで運ばれる間の揺れや温度によって品質を落としてしまう事がある。特に高価なワインは品質に問題があれば大きく価値を下げてしまうので問題になる。そこで、ワインにセンサを付け、輸送経路でのセンサ情報を蓄積し、後で管理者や消費者が確認できるようにすれば、品質に問題が無いか確認できる。また、食肉の食中毒対策のように食料品が安全な生産地で生産され、保全された状態で輸送された事を示す事で安全である事を保証する例も考えられる。

本稿では、身の回りの様々な物に無線センサノードが取り付けられている環境を想定する。そして、無線ネットワークを介してそれらの情報が取得できるとする。

2.1.2 情報家電・デバイス

ユビキタスコンピューティング環境では、様々な情報家電や AV 機器、コンピュータ等がネットワークに繋がりサービスを提供する。ネットワークに繋がる事で、従来のような単体での機能だけでなく、DLNA を用いたメディアの共有やネットワークからの操作等、総体としてのサービスや機能が多彩になり、ユーザにとってより便利になると考えられる。その一方で、サービス提供の方法が複雑になる事で、ユーザにとってサービスの把握が難しくなるという弊害が起こる。また、無線の活用によりサービスを提供する物がユーザから見え難くなるという問題が起こる。そこで、ユビキタスコンピューティング環境においては、情報機器は自身の機能や提供可能なサービスの情報を持っており、それをユーザに知らせる必要がある。環境内のデバイスの機能やサービスを記述する手法として、USDL[14] というサービス記述言語が提案されている。また、Universal Plug and Play (UPnP) はサービスの記述や発見等のプロトコルを仕様策定している。本稿では、デバイスやサービスの情報を USDL や UPnP のような手法で記述・発見できる環境を想定する。

2.1.3 Web 上の情報

Web 上の情報の中にも現実空間に関連付けられた情報が存在する。例えば電車やバスの時刻表、天気予報の情報である。

コミュニケーションサービスでも現実空間に関連した情報がやり取りされる。Twitter ではユーザが友人に対して一言のみ「つぶやく」事で緩やかな繋がりを成立させている。これらのつぶやきの中でも、「～なう」のように自分の位置を知らせるつぶやきが多く見られる。これは、「今自分は

～にいます」という意味であり，自分の居場所を友人に伝える事で連帯感を生じさせたり，現実空間での出会いを生み出すといった効果が期待できる．例えば以下のようなシナリオが考えられる．

学生である A さんは同じ研究室に所属している B さんに会いに行った．しかし B さんは研究室にはいなかった．そこで，A さんが B さんの Twitter のつぶやきを見てみると，「食堂なう」と書かれており，食堂に居る事がわかった．食堂に行ってみると，無事 B さんに会えた．

このシナリオでもわかる通り，Web 上でのコミュニケーションサービスでも位置情報が重要になったり役立つ場合が存在する．これらの情報を現実空間に関連付けて可視化する事で，ユーザの現在のステータスを示す物として利用できると考えられる．

2.2 情報可視化手法

本節では，まず既存の情報可視化手法について関連研究を分類し，特徴を述べる．次にその問題点を指摘し，本稿で提案する手法と uZi システムについて述べる．

2.2.1 PC ベースの情報可視化手法

センサ情報のようにデータベースに蓄えられる情報の可視化手法として最も素朴な方法はコンソールからデータベースにアクセスする方法である．コンソールからデータベースに対するクエリを文字列として送る事でそのクエリにマッチする結果を得られる．この手法では全ての情報を文字列や数値情報として扱う．データベース上の情報に対しては最も柔軟な運用が可能であるが，データベースに対する操作に関する知識や技術が必要になる．

Web 上の情報に対しては通常の Web ブラウザから見られる HTML によって記述された表示手法が用いられる．センサデータベースが Web インターフェースを持っている場合にもこの手法が用いられる．文字や画像をベースにした可視化であり，グラフ等から視覚的に情報を理解するのに優れる．しかし，コンソールを介したデータベースアクセス手法も Web インターフェースを用いた手法もユーザのいる場所は考慮しない．また，PC を介して操作する必要がある．そのため，例えばユーザのいる場所の近くにある情報を見る場合には，ユーザの位置情報を指定したり，文字列やグラフから自分に関連する部分を抜き出して解釈する等，ユーザにとって煩わしい操作を要求する．

2.2.2 現実空間に即した情報の可視化手法

上記のような現実空間に即さない可視化手法の欠点を補う手法として，地図画像への情報のマッピングや二次元バーコードを用いた可視化手法，AR 技術等がある．

地図画像へのマッピング

特定の場所に係わる情報の可視化手法として、地図画像へのマッピングがある。特定の場所に係わる情報とは、例えばその場所における温度のような情報である。この手法は天気予報や気温分布等の表現に使われる。無線センサネットワークの情報可視化手法としても広く使われており、Crossbow 社の Mote の情報ビューワである MoteView や AiryNote[8] で使われている。

二次元バーコード

また、AiryNotes では携帯電話の二次元バーコード読み取り機能を用いてセンサ情報を可視化する操作にも対応している。新宿御苑で行われた微気象観測に関する実証実験において使用された AiryNotes システムでは、利用者が簡単にセンサ情報を閲覧できるようにセンサを入れておくパッケージに二次元バーコードが印刷されている。利用者は情報を閲覧したいセンサを発見したら、センサパッケージ上の二次元バーコードを携帯電話で撮影する事でセンサ情報を Web ブラウザ上で閲覧できる。この手法は、多くのユーザが持っている携帯電話をユーザデバイスとして採用しており、利用者に特殊な技術や操作を要求する事なく、手軽にセンサ情報へのアクセスが可能になる。

Augmented Reality

現実空間に即した情報の可視化技術として、Augmented Reality (AR) 技術が注目されている。AR 技術とは、現実空間に仮想空間上の情報を投影する技術である。AR 技術を用いる事により、ユーザは仮想空間の情報を現実空間と関連付けて理解できるようになる。図 2.2 に実際に AR 技術が使われているアプリケーションの例を示す。この例では、ロボットの挙動を赤い線で表示する事で、ユーザは実際にロボットを操作しなくてもロボットの動作を確認できる。

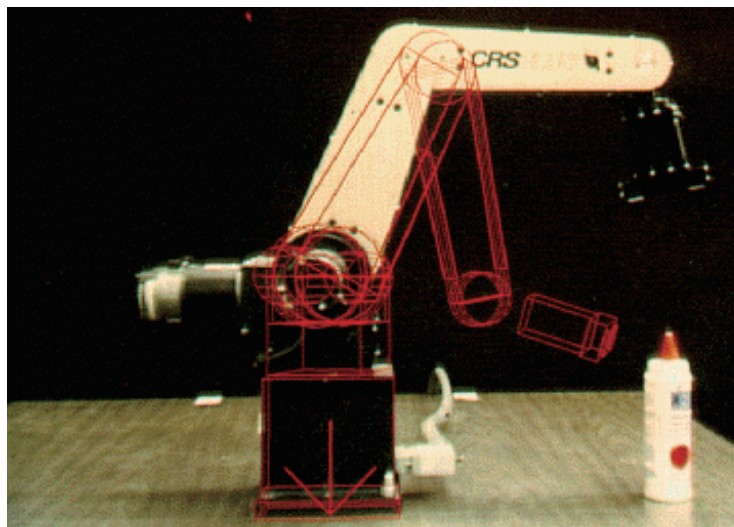


図 2.2 AR アプリケーションの例

AR 技術によってセンサや Web 等の情報を現実空間上に投影する手法としては、既に提案や実用化されているものが存在する。特にセンサの可視化に対しては兵器技術分野において既に様々なシステムが実用化され運用されている。例えば、戦闘機や戦闘ヘリ等のコックピットには、搭乗機の周辺状態をヘッドアップディスプレイ (HUD) やヘッドマウントディスプレイ (HMD) 上に表示するシステムが搭載されている [1]。このような技術は民間分野への転用も進んでおり、既に自動車のフロントガラス上に周辺情報を表示する技術が実用化されている。しかし、これらのシステムは航空機や自動車の周辺という限定的な状況や空間にのみ対応する。今後、人間の生活空間内には様々な情報デバイスが登場すると考えられ、その使用用途も多岐に渡る。人間の生活空間内には様々なセンサやデバイスが偏在するユビキタスコンピューティング環境においては、情報可視化システムは多様化する状況に対応できるシステムである必要がある。

AR 技術を用いた生活空間内における仮想空間の情報可視化システムとしては NaviCam[13] が挙げられる。これは現実空間内に仮想空間の情報をビジュアルマーカの形で埋め込み、カメラでビジュアルマーカを読み取って現実空間の映像と仮想空間の情報を重ね合わせて表示するものである。このシステムでは、現実空間内の本棚やカレンダー情報等、比較的静的な情報を対象としており、常に変化し続ける動的な情報であるセンサ情報は対象としていない。ユビキタスコンピューティング環境での AR 技術の利用例として、慶應義塾大学徳田研究室の u-Photo[16] が挙げられる。これは、写真というメタファを用いて情報機器の操作を可能にするデバイスである。u-Photo は直感的なデバイス操作を目的としており、ユーザは情報機器を撮影した写真によって情報機器を操作できる。

AR 技術を用いてユビキタスコンピューティング環境内のセンサ情報の可視化を行っている研究として SensAR[6] がある。これは AR 技術を用いて、センサから得られた環境の状態をユーザに親しみやすい 3D オブジェクトを用いて抽象的に可視化している。また、都市デザインのためのツールとして、CO センサを可視化する Lens[17] がある。これはセンサから得られた情報を球や煙等様々な表現で可視化するシステムである。都市デザイナーはセンサから得られた情報を自分のいる場所と関連付けられながら確認できる。

AR 技術からヒントを得た新しいインタラクションとして MIT Media Lab の WUW (Wear Ur World) [11] がある。この研究では現実空間と仮想空間の間のインタラクションには断絶があるとし、より直感的な操作と可視化を可能にするデバイスを提案している。具体的には、ユーザはカメラとプロジェクタから成るデバイスを装備し、空間的な指のジェスチャによって操作を行う。これにより、ユーザはキーボードやディスプレイから解放され、現実空間に即した操作や、物の情報を物体上に直接投影する等が可能になる。

2.2.3 既存手法の問題点

既存手法の多くはユーザに「不連続な操作」を要求する。不連続な操作とは、ユーザが目的としている操作とは直接的には関係無いデバイスや情報に対する操作の事である。これに対して、連続的な操作とはユーザが目的としている操作に対して直接関係する操作の集合のみで成り立つ操作を

指す。本稿では、情報可視化に際しては連続的な操作が可能になるシステムが望ましいと考える。コンソールや Web インターフェースの文字や画像を用いた情報可視化手法では、PC 上での操作を要求されるため、ユーザは不連続な操作をしなくてはならない。具体的には、以下のようなシナリオが考えられる。

C さんは自室で観葉植物を育てている。観葉植物周辺の環境管理のためにセンサを配置している。ある日観葉植物の一つが枯れている事に気づき、原因を調べるためセンサ情報が蓄積されているノート PC を立ち上げ、データベースにアクセスした。観葉植物に付けられているセンサの ID を調べ、最近数日間の照度と温度のデータを取得するようクエリを投げた。

D さんは出張先の都市で電車に乗って移動する事にした。ホテルの最寄の駅に着くと、携帯電話を使って現在地と目的地を入力し、時刻表を表示した。時刻表から路線を調べ、発車時刻を調べた。

センサ情報を調べるシナリオでは、センサが目の前にあるにも関わらず、ノート PC を起動し、データベースにアクセスする等、不連続な操作をしている。また、時刻表のシナリオでは、ユーザが駅のプラットフォームにいたとしても、Web 上から時刻表を探すには現在位置や目的地を入力する必要があり面倒である。目の前の電車が目的地まで行くかどうか分からない事がある。

地図画像上に情報を描画する手法は広く使われている。特に位置的に広範囲に分布している情報を取得する際には有効である。しかし、本稿において問題意識としている、自分の目の前にある情報を取得する際には地図と自分のいる位置を確認しながら情報を探す必要がある。つまり、ユーザは自分の目的に合うように地図情報を解釈しなくてはならず、不連続な操作と言える。

二次元バーコードを用いる手法では、一般的に二次元バーコードは Web ページへの誘導手段として用いられる。センサ情報を扱っている AiryNotes も二次元バーコードを同様の手段として用いている。そのため、最終的には Web インターフェースを用いた操作を要求される。また、利用者に対して「携帯電話による撮影」という操作を要求するため、必要としている情報にたどり着くのが困難になる場合がある。例えば、利用者が周辺にあるセンサの中から異常な値を示しているセンサを探したいと思った場合、センサを一つ一つ撮影して回らなくてはならず、面倒である。また、情報の取得自体は Web インターフェースを用いるため、上記と同じ問題が起こり、不連続な操作と言える。

これらの既存の手法では目の前にある物についての情報を取得するにも不連続な操作が要求され、不便である。本稿では、目の前にある物の情報は目の前に出す事で、連続的な操作になると考えた。この要求に対しては AR 技術を使う事で解決できるが、既存の手法ではユビキタスコンピューティング環境の多用な情報源や情報の種類に対応できない。例えば、SensAR や SiteLens では特定のセンサについてのみ可視化を行っており、その他の多様なセンサへの対応は示されていない。ユビキタスコンピューティング環境では今後も様々なセンサやデバイスが登場すると考えられ、センサ情報の可視化手法は未知のセンサに対する対応も示さなければ不十分である。また、これらの手法ではセンサ情報以外の可視化についても考えられていない。

WUW は現実空間と仮想空間の断絶を埋めるための新しいインタラクションとして注目される。しかし、WUW は操作から視覚までを含めたインタラクション全般に対する研究であり、物体は数種類のみしか認識できず、センサやデバイス等の情報への対応も想定していない。また、WUW では指の認識を行うために、ユーザはカメラや小型プロジェクタ、カラーマーカ等の特殊なデバイスを装備しなくてはならず、ユーザに対する負荷が大きい。このようなユーザに対して負荷を与えるような装備は不連続な動作であると考えられる。

2.2.4 虫眼鏡レンズのメタファを用いた情報可視化手法

既存手法における問題の分析から、uZi は AR 技術を用いて情報の可視化を行う。AR 技術を使う事で目の前にある物の情報を目の前に出せるため、ユーザが特殊なデバイス进行操作したり情報を解釈したりする手間が省け、連続的な操作が可能になると期待できる。

上記の既存研究に関連して述べたように、ユーザに負荷を与えるデバイスを装着させる事はそれ自体が不連続な操作と言える。そこで、本稿ではユーザデバイスとしては携帯電話等の携帯端末を用いる。現在、多くの人々は携帯電話を持ち歩いており、携帯電話をユーザデバイスに採用する事で連続的な動作となる事が期待できる。

また、ユーザの多様な要求に応え、ユビキタスコンピューティング環境での未知のデバイスに対応するために、本稿では取得した情報のフィルタリング機構を採用する。ユビキタスコンピューティング環境には、膨大な量と種類の情報が存在する。これらを全て携帯端末のディスプレイ上に表示する事は事実上不可能である。そこで、可視化したい情報毎にフィルタを用意し、フィルタを切り替える事で可視化する情報を変化させる手法によりこの問題を解決する。本稿では、uZi を、携帯端末を虫眼鏡、フィルタを虫眼鏡のレンズと見立て、レンズを切り替える事で様々なユビキタスコンピューティング環境の情報が見られる虫眼鏡システムとしての利用を想定する。

2.3 本章のまとめ

本章では、まず本稿において可視化対象とする情報を定義した。次に、それらの情報を可視化する既存の手法について分類し、その問題点を指摘した。この問題点を踏まえ、本稿で開発する虫眼鏡のメタファによる情報可視化システムである uZi の概観を示した。次章では、プロトタイプの実装を通して AR 技術による情報可視化手法を示し、携帯端末を用いた情報可視化手法について述べる。

第 3 章

AR 技術と携帯端末を用いた情報の可視化

本章では，AR 技術と携帯端末を用いた情報可視化手法について述べる．まず，本稿の有用性を確認するための uZi のプロトタイプについて述べる．次に，プロトタイプ実装の考察を元にして uZi で採用したアプローチを示す．

3.1 プロトタイプ

uZi の実装の前段階として、AR 技術による情報可視化手法の有用性を確認するために uZi のプロトタイプを実装した。本節では、実装したプロトタイプの概要と、プロトタイプを用いた予備実験について述べる。

3.1.1 概要

ユビキタスコンピューティング環境における情報可視化手法として AR 技術を採用したシステムはあまり存在しない。そこで、プロトタイプを実装し、実際に AR 技術で情報の可視化が可能であるか確かめる必要があると考えた。

まず、プロトタイプは実装の簡略化のため、携帯端末ではなく PC 上で実装する事とした。プロトタイプでは、可視化対象の情報としてセンサ情報のみを扱う。これは現実空間に関連付けられた仮想空間上の情報として、最も動的に変化し、従来の手法では可視化が難しいと考えたからである。このプロトタイプを通して、本稿で提案している手法の有用性が確認でき、さらにその実装において発生する問題点を発見できると考えた。

現実空間のオブジェクト認識手法

現実空間上に仮想空間の情報を投影するには、カメラ映像から現実空間上に存在するオブジェクトやカメラの位置情報等、現実空間を認識する必要がある。現実空間を認識する手法としては、

- ビジュアルマーカを用いる手法
- 3D モデルを用いる手法
- 特徴点を抽出する手法

の 3 つがある。以下にそれぞれの特徴を述べる。

ビジュアルマーカを用いる手法は、カメラに映ったビジュアルマーカの形状からカメラとビジュアルマーカ間の位置関係を求める。ARToolkit[9] のようにライブラリとして提供されているソフトウェアが増えた事から、AR 分野では広く使われている。ビジュアルマーカを用いるには予めビジュアルマーカの画像を用意し、ビジュアルマーカと現実空間の間の紐付けを行っておく必要がある。さらに、現実空間にビジュアルマーカを配置するという手間がかかる。しかし、ビジュアルマーカに一意に決定された ID を割り振っておけば、大量にオブジェクトが存在する空間内でも個々のオブジェクトを認識できる利点がある図 3.1 にビジュアルマーカの例を示す。

3D モデルを用いる手法は、予め現実空間内のオブジェクトの 3D モデルを作成しておき、現実空間の映像からそのモデルに一致する物を探し、カメラとオブジェクト間の位置関係を求める [3]。この手法は認識したいオブジェクトの 3D モデルを作成しておく必要があるが、現実空間にビジュアルマーカを配置しなくてもよい。しかし、同じ形をしているオブジェクトは同じ物として認識し

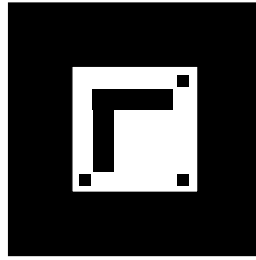


図 3.1 ビジュアルマーカ

てしまう可能性が高い。そのため、商品を管理している倉庫内のように同じ物が大量に存在する空間には対応できない。

特徴点を抽出する手法は、カメラ映像を画像解析して特徴点を抽出し、カメラの移動量からカメラの位置や姿勢を求める手法である [10]。予め用意しておくモデルが存在せず、現実空間を改変する必要もないため、他の手法に比べて人間の生活空間内にも簡単に導入できる。しかし、カメラの位置情報を求める手法のため、カメラがどの程度移動したかはわかっても、現実空間のオブジェクトを認識できない。また、他の手法に比べて計算量の観点から負荷が高い。以上の 3 つの手法の特徴を表 3.1 にまとめた。

表 3.1 現実空間の認識手法の比較

	現実空間の 改変	事前の モデル作成	個別の オブジェクト認識
ビジュアルマーカ	必要	必要	可能
3D モデル	必要	不必要	形状が違えば 可能
特徴点	不必要	不必要	不可能

本稿では、人間の生活空間内を想定環境としている。そのため、形が同じでも違うオブジェクトが存在する状況がある。例えば商店等では、同じ種類のオブジェクトが大量に存在している状況が考えられる。上記の比較を踏まえ、本稿では形が同じでも個々のオブジェクトを違う物として扱えるビジュアルマーカを用いる手法を使う。また、ビジュアルマーカはシステム利用者のサービス発見の助けとなる効果も期待できる。

センサ情報の抽象化

動的に変化し続けるセンサ情報を文字で表現すると、数字が目まぐるしく変化して読み取りが困難になる。グラフ、絵等の二次元的な表現をすると、三次元である現実空間の映像から浮き上がってしまう。そのため、システム利用者の視覚を混乱させ、円滑な作業進行を妨げてしまう恐れがある。

そこでプロトタイプでは 3D モデルを使って抽象的にセンサ情報を表現する。図 3.2 に実際の実行時の画面を示す。

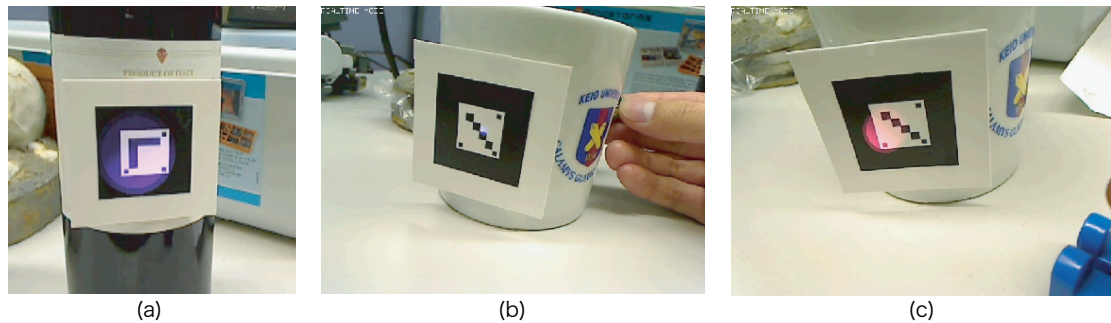


図 3.2 実行時画面

今回のプロトタイプ実装では、球の大きさや色、運動によってセンサ情報を表現している。球の大きさは照度を表現している。センサが照明の下で照らされている時には図 3.2 の (a) のように球が大きく見える。センサを手で隠す等して光を遮ると、(b) のように球が小さくなる。球の色によって温度を表現している。通常の室温程度の温度の時には球の色は青い。熱湯を注いだカップにセンサを近づけて温度を上げると色が赤く変化する。また、球はビジュアルマーカを中心にして公転運動をしている。この公転半径と運動の速さで加速度を表現している。公転半径が小さく、遅く動いている時、加速度センサは小さい値を示している。公転半径が大きく、速く動いている時は、加速度が大きい事示している。

想定環境

プロトタイプ実装では、人間の生活空間内のオブジェクトにビジュアルマーカとセンサが貼り付けられている環境を想定した。センサは uPart[2] を用いる。uPart は

- 照度
- 温度
- 振動（加速度）

の 3 種類の情報をセンシングできる。また、各センサはセンサ情報の蓄積や配信を行うセンサ情報サーバにネットワークで接続されていると想定した。

ユーザはカメラの取り付けられた HMD を装着し、それらが接続された小型 PC を持ち歩くと想定した。実際にシステムを使用しているユーザを図 3.3 に示す。

センサ情報フィルタ機構

現実空間上には大量のオブジェクトが存在し、今後センサが内蔵されたオブジェクトが増大していくと予想される。センサが内蔵されていないオブジェクトでも、無線センサノードを利用する事



図 3.3 プロトタイプシステム利用者

でシステム利用者自身の手でセンサを取り付けられる環境が整備されていくと考えられる [18]. このような環境下で、全てのセンサ情報を HMD 上に投影する事は困難であるだけでなく、システム利用者の視覚情報を混乱させ作業を妨害してしまう結果になりかねない. また、システム利用者の求める情報によって、表示する情報の種類を切り替える必要がある. 例えば、現在のセンサ情報を取得したい場合と、過去の一時点でのセンサ情報を知りたい場合では要求される情報の種類が違う. このような要求に答えるために、プロトタイプ実装ではセンサ情報フィルタ機構を実装した. センサ情報フィルタ機構はユーザの操作に応じて“サングラスのレンズを切り替えるように”表示する情報の種類を切り替える機能である. センサ情報フィルタ機構により、表示する情報を現在と過去で切り替えたり、特定の値を示すセンサのみを表示するといった表示方法ができるようになる.

プロトタイプでは、リアルタイムフィルタ、タイムマシンフィルタ、異常状態検知フィルタの 3 つのフィルタを実装した. リアルタイムフィルタは現在のセンサの状態のみを表示する. タイムマシンフィルタは過去にさかのぼってセンサ情報を表示するフィルタである. システム利用者はマウス操作によって、時間を巻き戻したり、逆に早送りする等して、時間経過によって変化するセンサ情報を閲覧できる. 異常状態検知フィルタは異常状態が検知されたセンサのみを表示する. このフィルタを用いる事で、例えば、食料品が過去に異常な状態に置かれていたかどうか確認でき、食料品の安全性を確認できる.

3.1.2 予備実験と考察

予備実験として、実装したプロトタイプを大学生 5 人に使ってもらい、聞き取り調査を行った. また、慶応義塾大学湘南藤沢キャンパスの研究発表会である SFC Open Research Forum 2007 において、来場者に実際にプロトタイプを使ってもらい聞き取り調査を行った. 頂いたコメントとして、「従来の二次元バーコードや Web インターフェースを使ったシステムよりも理解しやすい」といった肯定的な回答が得られ、プロトタイプの有用性が確認できた. 特に、現実空間に関連付けら

れた仮想空間上の情報の可視化手法としての AR 技術について「見ただけで欲しい情報が取得できる」と有用性がわかった。しかし、「日常的にはこのシステムを使いたいとは思わない」等、否定的な結果も得られた。これは、今回の評価実験ではノート PC や Web カメラ等のような、携帯電話に比べて非常に大きな機器を用いたため、HMD をユーザデバイスとしたハードウェア構成にはユーザに不満があったためであった。プロトタイプからの教訓として使用機器を小型化する必要がある事がわかった。

また、抽象的な可視化手法は情報のおおよその値を読み取ったり、複数の情報を比較する際には有用だが、それだけでは具体的な数値がわからないという問題点があった。予備実験でのユーザからの聞き取り調査でも、数値情報やグラフ情報を表示したいという要望があった。プロトタイプ実装では球のみでセンサ情報を表示しているが、状況に応じてモデルの使い分け等の解決策が必要である事がわかった。また、異常状態検知フィルタでは異常状態にあるセンサのみ画面に情報を表示していたが、「異常状態検知フィルタで異常ではないビジュアルマーカを読み取った時には異常では無い事を示す情報を表示してほしい」という意見も見られた。このコメントから、利用者はビジュアルマーカのある場所には常に情報が表示される事を期待している事がわかった。このような利用者の要求に対しては、プロトタイプのフィルタ設定の改善や、新しくフィルタを用意する等の必要がある。

さらに、プロトタイプの実装から、フィルタ機構の実装には多大な労力がかかる事がわかった。プロトタイプではセンサ情報のみを対象としてフィルタ機構を構築した。リアルタイム、タイムマシン、異常状態検知の 3 つのフィルタを実装したが、これら以外にもフィルタが必要になると考えられる。実際、プロトタイプでもセンサの所有者情報を元にしたフィルタが欲しいといった要望があった。また、ユビキタスコンピューティング環境ではセンサ以外にも様々な情報源からの情報を可視化する必要がある。

3.2 uZi システム

本節では本稿で提案する物体可視化手法を実現する uZi システムについて述べる。

3.2.1 アプローチ

プロトタイプの考察から、HMD を使ったユーザデバイスはユーザへの負担が大きかったり、ユーザの装着時の外観が損なわれてしまうので、予備実験での不満が大きかった。そこで、uZi では携帯端末をユーザデバイスとして採用する。近年、携帯電話等の携帯端末が高機能化し、従来は難しかった画像処理や高速な無線ネットワークへの接続が可能になった。特に、iPhone や Android のようにデバイスの全面がディスプレイになっている携帯電話が登場している。図 3.4 に iPhone と Android 端末を示す。

プロトタイプでは AR 技術による情報可視化手法の有用性が確認できた。そこで、uZi でも情報の可視化は AR 技術を使う。物体の認識手法はビジュアルマーカを使う。ビジュアルマーカは利



図 3.4 スマートフォン 左が iPhone 3GS, 右が Android Dev Phone

用可能なデバイスやサービスの目印となる効果がある。

本稿では、iPhone のような全面ディスプレイの携帯端末を虫眼鏡デバイスとして使う事を提案する。携帯端末のディスプレイに、携帯端末の先にある映像を映し、その場所に関係のある情報を投影する。ユーザはディスプレイを覗き込む事によって、空間的位置関係と関連付けながら情報を確認できる。HMD と違い、現在多くのユーザは携帯電話を日常的に持ち歩いていると考えられるので、携帯端末はユーザへの負担が少ない。携帯端末をかざすだけで情報の取得が可能なので、デバイス装着による外観についても気にする必要がない。図 3.5 に実際の動作の様子を示す。

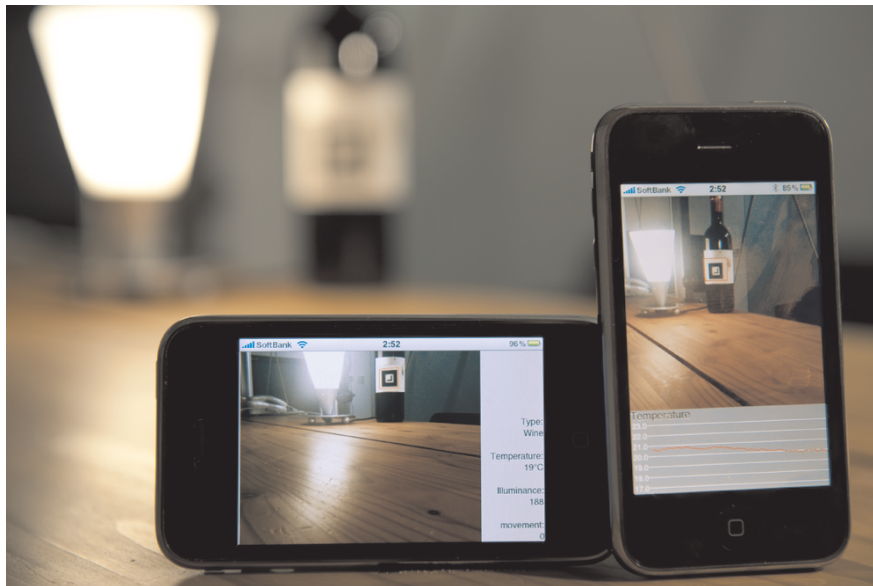


図 3.5 uZi の動作の様子

3.2.2 物体情報

プロトタイプと同様に、各物体にはビジュアルマーカが貼りつけられていると想定する。また、物体によってはセンサも同様に貼りつけられている。対象とする物体としては、具体的には、

- ワイン

- オフィスの机
- バス停

等を想定する。図 3.6 に実際にビジュアルマーカやセンサが貼りつけられている様子を示す。



図 3.6 物体にビジュアルマーカとセンサが貼り付けられている様子

ビジュアルマーカには一意に ID が振られており、物体の基本的な情報が物体データベース上に ID と紐付けられて格納されていると想定する。物体情報は、

- 物体の名称
- 所有者
- センサの ID
- 異常状態の定義

等が記述されていると想定する。

3.3 本章のまとめ

本章では、まず uZi のプロトタイプを実装し、予備実験を行った。その結果、AR 技術を用いた情報可視化手法の有用性が確認できた。また、PC や HMD をユーザデバイスとするシステムはユーザへの負担が大きく、実際にユーザの不満も高い事がわかった。これらの結果から、AR 技術と携帯端末を用いる uZi の有用性が期待できたため、uZi の詳細について検討した。また、uZi で想定する環境について定義した。プロトタイプの実装過程では、フィルタの実装量が膨大になり、開発者の負担が大きい事もわかった。この解決策として、次章でレイヤ構造を持つフィルタリング機構について述べる。

第 4 章

レイヤ構造を持つフィルタリング機構

本章では，uZi のフィルタリング機構について述べる．uZi のフィルタリング機構は，様々なユーザの要求に応えるためにレイヤ構造を持つ．まず各レイヤについて述べ，ついでレイヤ間の連携について述べる．

4.1 フィルタリング機構

本節では uZi のフィルタリング機構について述べる。まず、従来の AR アプリケーションで使われているフィルタリング機構について調査し、その欠点を指摘する。次に従来のモデルの欠点を克服するレイヤ構造モデルのフィルタリング機構を提案する。

4.1.1 従来の AR アプリケーションにおけるフィルタリング機構

AR 技術を情報可視化として採用しているシステムではフィルタリング機構を備えている。例えば、セカイカメラ [4] は現実空間の特定の位置に文字列や画像、音声等のタグを貼りつけられるアプリケーションである。ユーザはセカイカメラを通して現実空間を見る事で、その場所の説明やユーザのコミュニケーションに使える。たくさんのユーザが現実空間にタグを貼りつけすぎると、人が多く集まるような場所はタグでいっぱいになってしまう。これらのタグを全て画面上に表示しようとする、画面がタグで埋まってしまう。現実空間の映像も見えなくなってしまうため、現実空間の映像上に情報を投影する事で仮想空間情報の現実空間における空間的位置関係を関連付けられる AR 技術を使う意味が無くなってしまふ。そのため、セカイカメラではタグの登録時刻に応じて表示するタグを絞り込めるフィルタが実装されている。これにより、古い情報は表示しないようにでき、一度に表示する情報量を制限できる。

Layer[21] は周辺の情報を検索してその情報を可視化する、現実空間の上に仮想空間のレイヤを追加するアプリケーションである。特に施設情報を探す用途を想定しており、例えば現在位置の近くにあるレストランや ATM を検索し、現実空間の映像上にその情報を投影できる。Layer では検索対象を指定するための「コンテンツ」が用意されている。この「コンテンツ」は多数の施設情報から目的の施設情報を絞り込むための一種のフィルタと捉えられる。また、Layer は様々な用途や要求に合わせるため、フィルタを追加できるという特徴がある。これにより、新しい検索対象が登場した際にもフィルタを追加するだけで対応できる。

これらの AR アプリケーションでは、個々のフィルタは個別に実装され、全て並列に扱われる。それぞれのフィルタ間での連系は考えられておらず、似たような機能を持つフィルタがあっても再利用性は低い。uZi のプロトタイプ実装でも従来の AR アプリケーションと同じように全てのフィルタを並列に実装した。その実装経験から、並列したフィルタリング機構の実装はフィルタ追加に手間がかかり面倒になりがちである事がわかった。例えば、プロトタイプでは、過去のセンサ情報で異常値を示すセンサのみを表示する異常状態検知フィルタを実装した。センサのみを考慮した場合、異常状態検知フィルタは対象とする情報が現在の情報の場合と過去の情報の場合がある。この時、現在の情報を対象とした異常状態検知フィルタは現在の情報を取得する部分がリアルタイムフィルタと重複する。また、同様に、過去の情報を対象とする異常状態検知フィルタはタイムマシンフィルタと重複する。さらに、それら 2 つの異常状態検知フィルタは異常状態を検知する部分が重複する。並列したフィルタリング機構では、異常状態検知フィルタは異常状態の定義や取得方法

が同じでも別個に実装する必要がある。異常状態検知フィルタの例で見られたように、フィルタの中の機能には他のフィルタで重複する場合が考えられる。特にユビキタスコンピューティング環境では情報源として様々な物が考えられる。それら全てにリアルタイム、タイムマシン、異常状態検知等のフィルタを実装するのは非常に手間がかかる。そこで、既存のフィルタの中から共通化できる部分のみを取り出して実装し、それぞれのフィルタが連携する事でフィルタの開発を単純にできると期待できる。ソースコードレベルでの共通化を図る事で手間の軽減を実現する事は可能であるが、設計レベルでの強制力が無く、また開発者にフィルタ実装のための特殊な知識が必要になってしまう。uZi プロトタイプやセカイカメラ、Layer 等の従来の AR アプリケーションで採用されているフィルタリング機構の概略を図 4.1 に示す。

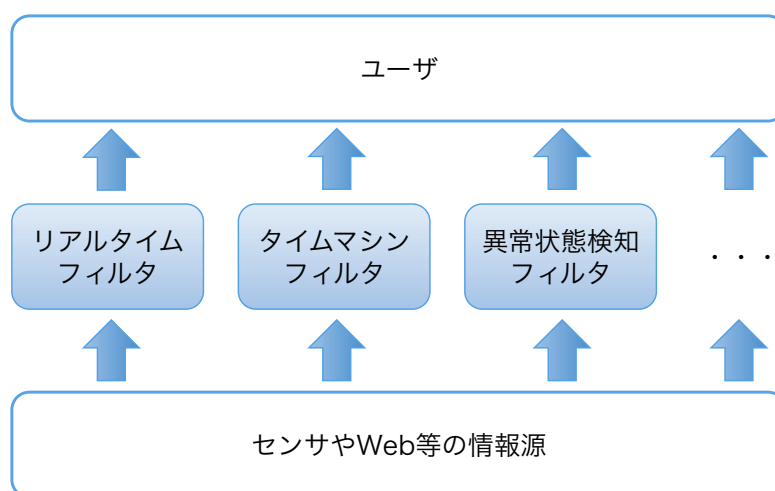


図 4.1 従来のフィルタリング機構 各フィルタが並列に存在するため、再利用性を阻害する。

4.1.2 レイヤ構造フィルタリング機構

プロトタイプの実装と既存 AR アプリケーションにおけるフィルタリング機構の考察から、フィルタの中には機能の重複する物があるにも関わらず、既存のフィルタの実装方法は再利用性が低い事がわかった。uZi では個々のフィルタは最低限の機能のみを実装する事とした。単純なフィルタ同士が連携する事で複雑な機能を提供する。ユーザは複数のフィルタを選択し連携させる事で、システムに要求する情報を伝える。例えば、ユーザがセンサ情報を情報源としたい場合にはセンサ情報取得フィルタを選択する。センサ情報取得フィルタは特定の種類のセンサから情報を取得する機能しか持たない。センサ情報取得フィルタから得られたセンサ情報は、次のフィルタに送られる。ユーザが次のフィルタとしてリアルタイムフィルタを選択した場合、センサ情報の中から現在時刻に最も近い情報のみを選び、次のフィルタに送る。さらに異常状態検知フィルタにこの情報が送られると、センサ情報が予め定義された異常状態になっているかどうか調べ、異常状態と検知されれば次のフィルタに送られる。このようにして、次々にフィルタを通すことで、最終的にはユーザの

要求する情報のみを出力する。

フィルタの中には役割が競合するものがある。例えば、最新の情報を取得するリアルタイムフィルタと過去の情報を取得するタイムマシンフィルタを同時に使用する事はできない。uZi のフィルタリング機構はこのようなフィルタ間の関係を表現するため、レイヤ構造を持つ。uZi では、各フィルタは必ず何かのレイヤに所属する。例えばセンサ情報フィルタであれば入力レイヤ、リアルタイムフィルタであれば時系列レイヤに所属し、同じレイヤに属するものは同時には使用できない事とした。また、フィルタ同士の連携を無制限に許してしまうと、ユーザのフィルタリング機構への理解を妨げてしまう恐れがある。フィルタリング機構をレイヤ構造にする事で、ユーザは単純な各レイヤ間の関係のみに注目すれば良くなる。前述の従来の AR アプリケーションのフィルタリング機構との比較のために、uZi のフィルタリング機構の概略を図 4.2 に示す。

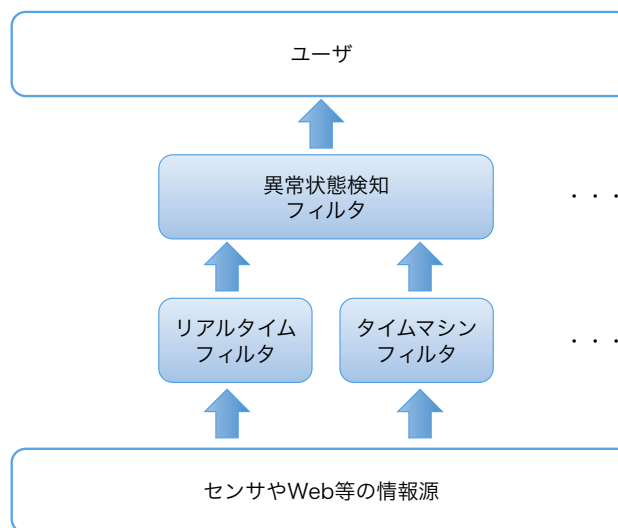


図 4.2 uZi のフィルタリング機構

4.2 レイヤ

本節では各レイヤとそのレイヤに所属するフィルタについて説明する。uZi の各レイヤとフィルタは、想定するシナリオやアプリケーションから決定した。uZi のフィルタリング機構の構造を図 4.3 に示す。

4.2.1 入力レイヤ

入力レイヤは大元となる情報源を指定するレイヤである。以降のレイヤは入力レイヤで得られた情報を元にフィルタリングを行う。uZi システムはビジュアルマーカの ID に紐付けられた物体情報を入力レイヤに送る。



図 4.3 フィルタリング機構の構造

SunSPOT フィルタ

uZi では無線センサネットワークとして SunSPOT の使用を想定している。SunSPOT フィルタは SunSPOT のネットワークから得られた情報を取得する。システムから渡された物体情報に載っている SunSPOT の ID を元に、SunSPOT ネットワークから該当 ID のセンサ情報を取得する。過去から現在までの一定期間内に取得できた SunSPOT のセンサ情報を次のフィルタに渡す。実際の実装においては、現在のところ携帯端末から SunSPOT ネットワークの情報にアクセスする手段が無いため、一度センサ情報をセンササーバに蓄えてから携帯端末に送る。

Twitter フィルタ

Twitter フィルタは Twitter のつぶやきを取得する。対象とするアカウントは物体情報に載っている所有者のアカウントとする。Twitter のタイムラインには自分のつぶやきの他、友人のつぶやきや自分宛のダイレクトメッセージが載っている。

時刻表フィルタ

時刻表フィルタは Web 上の時刻表情報や運行情報を取得する。運休や遅延等の状態が異常状態に該当する。

4.2.2 時系列レイヤ

時系列レイヤは情報の時刻を元にフィルタリングする

リアルタイムフィルタ

リアルタイムフィルタは現在時刻に最も近い情報のみを次のフィルタに送る。入力元が SunSPOT フィルタの場合、最後に送られたセンサ情報のみになる。Twitter フィルタの場合、該当アカウントの最新のつぶやきを取得する。

タイムマシンフィルタ

タイムマシンフィルタは過去一定期間の情報を取得する。情報を取得する期間はユーザが決定できる。

4.2.3 関連度レイヤ

関連度レイヤは自分との関連の度合いによって情報を絞り込むレイヤである。

自分フィルタ

自分フィルタは自分の所有物の情報のみを取得する。Twitter フィルタの場合、自分のつぶやきのみを取得する。

友達フィルタ

友達フィルタは自分と自分の友人の所有物の情報を取得する。Twitter フィルタの場合、自分と自分がフォローしているアカウントのつぶやきを取得する。

4.2.4 絞り込みレイヤ

絞り込みレイヤは前のレイヤから取得された情報を特殊な条件で絞り込んで次のレイヤに送る。

異常状態検知フィルタ

異常状態検知フィルタは前のレイヤから送られてきた情報の中から、異常な値を示している物のみを残す。異常状態はオブジェクトデータベース上の物体情報で定義される。

4.2.5 出力レイヤ

出力レイヤは前のレイヤで得られた情報を画面に描画する。プロトタイプでは、情報は球で抽象的に表現されていた。抽象的な表現は複数のセンサ情報を比較する際に一覧性に優れる事がわかったが、具体的な情報がわからない事に対して被験者から不満が聞かれた。そこで、uZi では画面への描画もフィルタとして実装する事で、ユーザの好みの表現方法を選択できるようにした。

テキスト表示フィルタ

詳細情報フィルタは前のフィルタから得られた情報を数値や文字列によって表現する。ユーザがセンサの具体的な情報や Twitter のつぶやき、時刻表の時刻等を見たい時に使う。図 4.4 で実際の画面を示す。



図 4.4 テキスト表示フィルタ

グラフ表示フィルタ

グラフ表示フィルタは情報をグラフで表現する。特にタイムマシンフィルタと連携する事で、過去の情報を一覧できる。図 4.5 にグラフ表示フィルタの実際の画面を示す。

抽象表示フィルタ

抽象表示フィルタはプロトタイプでの描画手法と同じように、情報を抽象的に表現する。図 4.6 に実際の画面を示す。

4.3 レイヤ間の連携

各レイヤの中から選択されたフィルタは他のレイヤのフィルタと連携する事で、情報源からの情報の取得からディスプレイへの表示までを行う。ユーザは図 4.7 に示すフィルタ選択画面から使用したいフィルタを選択できる。レイヤの中でユーザに選択されたフィルタが無い場合、そのレイヤは無視され、全ての情報が次のレイヤに渡される。出力レイヤに選択されたフィルタが存在しない

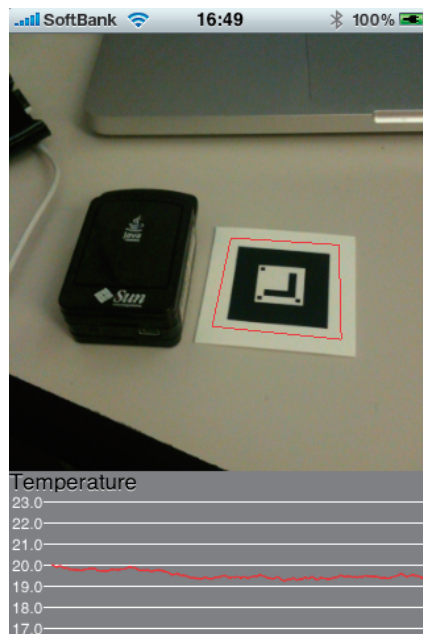


図 4.5 グラフ表示フィルタ



図 4.6 抽象表示フィルタ

場合には描画処理は行われない。

各レイヤのフィルタは前のレイヤから情報を受け取り、その情報を操作した後に次のレイヤに渡す。入力レイヤはセンサ情報やデバイス情報、Web 上の情報等、様々な情報源に対応する必要がある。入力レイヤで情報源の種類によらず統一的に情報を記述できなければ、その後のレイヤは機能が重複していても情報源に応じてフィルタを実装しなくてはならない。そこで、uZi では、入力



図 4.7 フィルタ選択画面

レイヤで得られた情報は XML で記述し，その XML を時系列レイヤ以降に送る．
実際の XML をソースコード 4.1 に示す．

ソースコード 4.1 フィルタ間通信で用いられる XML 例

```
<element name="sunspot">
  <timestamp type="longlong">123456789</timestamp>
  <sunspot_id type="int">0</sunspot_id>
  <illuminance type="int">19</illuminance>
  <temperature type="int">10</temperature>
  <accelerometer type="double">1.0</accelerometer>
</element>
<element name="sunspot">
  <timestamp type="longlong">123456790</timestamp>
  <sunspot_id type="int">1</sunspot_id>
  ...
</element>
```

XML で記述された情報を受け取った各フィルタは，それぞれの方針に従って不要な要素を削除する．最終的に，出力レイヤの XML が到達する時にはユーザが必要な情報のみが残る事になる．物体情報に予め所有者や異常状態の定義を記述しておく事で，各フィルタは情報源の種類によらず統一的に要素の値の検証や削除が行える．

4.4 本章のまとめ

本章では，uZi で採用したレイヤ構造を持つフィルタリング機構について述べた．従来 AR アプリケーションのフィルタリング機構は全てのフィルタを並列に扱う．並列型のフィルタリング機構では，フィルタ間に重複する機能があっても，そのフィルタを再利用する余地が少なく，フィルタ開発者の実装量は膨大になる．ユビキタスコンピューティング環境では大量のフィルタが必要になる可能性があり，このようなフィルタリング機構は現実的ではない．そこで，uZi はフィルタリング機構をレイヤ構造とする事で，フィルタの再利用性を上げ，開発者の負担を低減する．uZi のフィルタリング機構には入力レイヤ，時系列レイヤ，関連度レイヤ，絞り込みレイヤ，出力レイヤの5つのレイヤがあり，それぞれのレイヤに複数のフィルタが所属している．各レイヤは XML 形式のデータを通信し，連携する事でフィルタリング機構を実現する．次章では，これまでに示してきた手法を実現する uZi システムを設計する．

第 5 章

uZi の設計

本章では，現実空間に関連する仮想空間の情報を可視化するシステムである uZi を設計する．まずハードウェアの設計をし，次にソフトウェアを設計する．

5.1 ハードウェア構成

図 5.1 に uZi のハードウェア構成を示す。

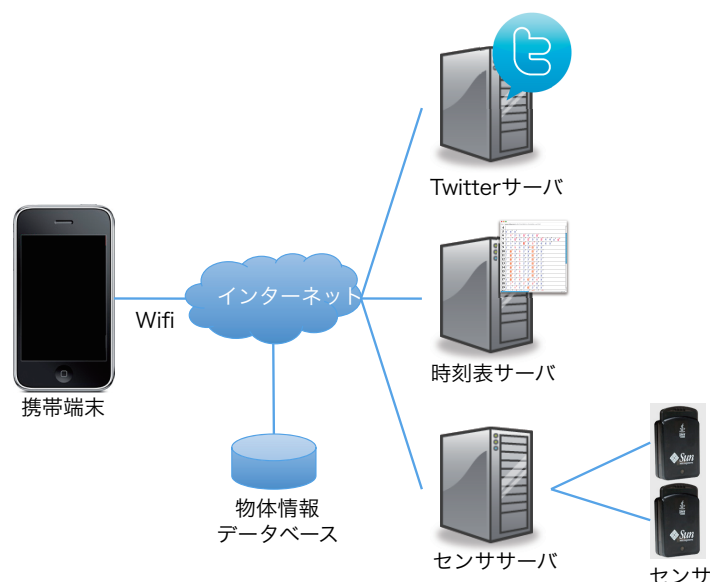


図 5.1 ハードウェア構成図

uZi はスマートフォンのような携帯端末をユーザデバイスとして用いる。携帯端末の想定として、まず周囲の映像を解析する必要性から内蔵カメラを持つ。また、uZi は虫眼鏡のような使用方法を想定しており、映像を大きく表示する必要性から、近年のスマートフォンの多くが採用している全面ディスプレイを持つとする。また、操作インターフェースとしてタッチパネルディスプレイを想定する。

ネットワーク接続性としては、携帯端末は無線 LAN を介してインターネットに接続する。ネットワーク上には、SunSPOT の情報を蓄積、配信する SunSPOT サーバを想定する。また、Web 上の Twitter サービスからつぶやきを、各公共交通機関や乗り越し案内サービスから時刻表や運行情報をそれぞれ取得できると想定する。また、ネットワーク上には物体情報データベースを想定する。このデータベースには、センサや情報家電等の現実空間上の物体に関する情報が蓄積される。オブジェクトデータベースに格納される物体情報はビジュアルマーカの ID と紐付けられている。

5.2 ソフトウェア構成

図 5.2 に uZi のソフトウェア構成図を示す。

携帯端末内蔵のカメラから映像を取得する。カメラから得られた映像は ARToolkit やその派生ライブラリ等によって解析され、ビジュアルマーカの位置を OpenGL 座標系として画面描画モ

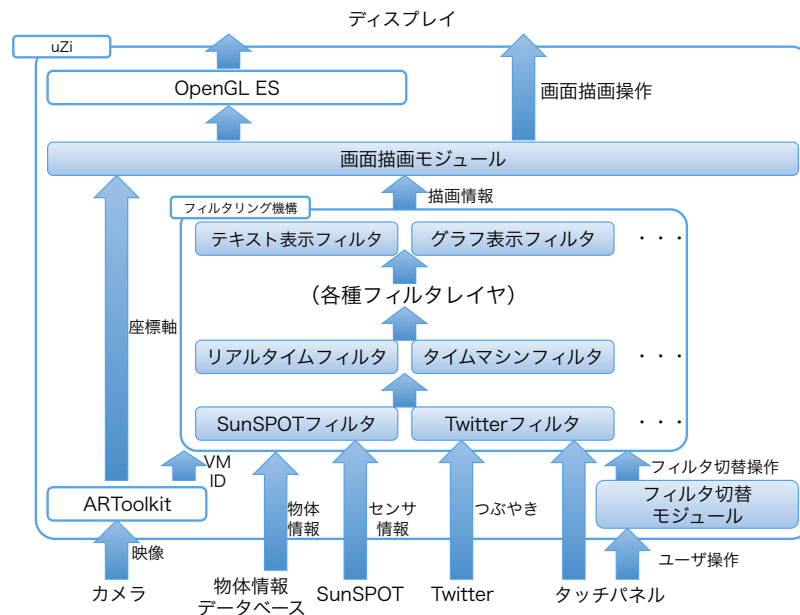


図 5.2 ソフトウェア構成図

ジュールに渡す。また、ビジュアルマーカの ID をフィルタリング機構に渡す。フィルタリング機構ではフィルタの管理を行う。ユーザはタッチパネルからの操作によって使用するフィルタの選択ができる。また、タイムマシンフィルタでフィルタリングする時刻の決定のようなフィルタに対する操作もタッチパネルを介して行う。SunSPOT フィルタや Twitter フィルタ、時刻表フィルタ等の入力フィルタは外部と直接通信して目的の情報を取得する。また、ARToolkit から得られたビジュアルマーカの ID も入力フィルタに渡される。入力フィルタで取得した情報は以降のフィルタに次々と渡され、最終的に出力フィルタに渡される。抽象表示フィルタ等の 3D オブジェクトの描画が必要になる出力フィルタを用いる場合は OpenGL ES を介して画面への描画を行う。それ以外の文字列表示や画像表示等の画面描画処理はプラットフォーム依存の表示手法を使う。

5.3 本章のまとめ

本章では、携帯端末と AR 技術を用いた情報可視化システムである uZi を設計した。まずハードウェアと uZi が接続するサーバ群について述べ、次にソフトウェアを設計した。次章では、本章の設計を元に uZi の実装環境について述べる。

第 6 章

uZi の実装

本章では，現実空間に関連する仮想空間上の情報を可視化する uZi の実装について述べる．まずハードウェアの実装について延べ，次にソフトウェアの実装について述べる．

6.1 ハードウェア

uZi のハードウェア環境を表 6.1 に示す.

表 6.1 uZi ハードウェア環境

デバイス	Apple iPhone 3GS
CPU	ARM Cortex-A8 600MHz
GPU	PowerVR SGX
メインメモリ	256MB
ディスプレイ	3.5 インチ 320x480 ドット
カメラ	300 万画素オートフォーカスカメラ

uZi のユーザデバイスは Apple 社の iPhone 3GS[7] を用いた. カメラは iPhone 3GS 内蔵の物を使った. iPhone 3G に比べ, iPhone 3GS はメモリに余裕があり, カメラの性能も上がっているため, uZi のような AR 技術を用いたシステムには最適であると考えられる.

また, 対象とするセンサは SunSPOT とした. SunSPOT は Sun 社から販売された無線センサノードであり, 他のセンサノードに比べて開発が容易であり, D/A コンバータを備えており拡張性に優れる. 今回の実装では SunSPOT から

- 照度
- 温度
- 加速度 (3 軸)

を取得し, 可視化する.

6.2 ソフトウェア

uZi ソフトウェアの実装環境を表 6.2 に示す.

ユーザデバイス上で動作する OS は iPhone OS 3.1 とした. プロトタイプは Windows での実装だったため, C++ を用いて実装した. iPhone ではフレームワークの制約上 Objective-C による記述を必要とする. そのため, 今回の実装では Objective-C と C++ の共存のため, 言語は Objective-C++ とした.

iPhone OS 3.1 から, iPhone 上での AR アプリケーションの動作が可能になった. 通常, iPhone の AR アプリケーションはセカイカメラや Layer 等のように, GPS とコンパスを用いる物がほとんどである. uZi ではプロトタイプと同様にビジュアルマーカを使って物体認識を行う. プロトタイプでは ARToolkit を利用してビジュアルマーカの認識を行っていた. しかし, 本稿執筆時点では iPhone 上で ARToolkit が利用できなかった. そこで, 今回の実装では ARToolkit で使われて

表 6.2 uZi 実装環境

実装マシン	Macbook (Mac OS X 10.6 SnowLeopard)
言語	Objective-C++
コンパイラ	gcc 4.2
ライブラリ	NyARToolkit (iPhone 環境に移植)
IDE	Xcode 3.2
対象 OS	iPhone OS 3.1

いるアルゴリズムの別の実装である NyARToolkit を iPhone OS 向けに移植した。NyARToolkit を使った理由は、オリジナルの ARToolkit よりも高速である事と、C++ で書かれた実装が存在したためである。

uZi では、フィルタ間では XML で記述されたデータの受け渡しが行われる。XML 形式の文字列を通信データとして用いる事により、フィルタ間通信の実装依存性が抑えられ、新しいフィルタの追加に際して有利になると考えられる。しかし、全てのデータを文字列で通信すると、解析に負荷がかかってしまう。そこで、今回の実装では、携帯端末でも軽快に動作させるために、文字列ではなく DOM オブジェクトをフィルタ間通信のデータとする事とした。XML データを予め解析済みの DOM オブジェクトとしておく事で、処理負荷を低く抑えられると期待できる。

6.3 本章のまとめ

本章では携帯端末と AR 技術を用いた情報可視化システムである uZi の実装について述べた。uZi はユーザデバイスとして iPhone 3GS を用いる。また、NyARToolkit を iPhone OS 向けに移植し、iPhone 3GS 上でのビジュアルマーカベースの AR アプリケーションを実現した。次章では、実装した uZi を評価する。

第 7 章

評価

本章では，uZi の評価を行う．定量評価として，まず画像取得から画面への描画までの速度を評価する．次にセンサ情報を蓄積するデータベースへのアクセス速度を評価した．レイヤ構造を持つフィルタリング機構の評価のために，実装の容易さについて従来のフィルタリング機構との比較を行った．また，被験者に実際に uZi を使ってもらい，ユーザビリティ評価を行った．最後に，実験結果からの考察を行う．

7.1 定量的評価

本節では uZi を定量的に評価する。

7.1.1 描画速度評価

uZi はビジュアルマーカを用いて物体認識を行うため、画像処理を行う。画像処理は比較的負荷のかかる処理のため、低スペックなマシンが多い携帯端末で実際に実用的な速度で動作するか確かめる必要がある。uZi の実装では、実際のカメラ位置と映像の追従性として 10fps 程度の速度が必要と考えた。

まず、ユーザデバイスの機器選定のための実験を行った。ユーザデバイスの機器候補として、スマートフォンプラットフォームの中で注目されている iPhone と Android の端末から、iPhone 3GS と Android Dev Phone 1 を選定した。比較のために実験で用いた iPhone 3GS のスペックと Android Dev Phone 1 のスペックを表 7.1 に示す。

表 7.1 実験に用いた端末のスペック

	iPhone 3GS	Android Dev Phone 1
CPU	ARM Cortex-A8 600MHz	ARM11 528 MHz
GPU	PowerVR SGX	CPU
メインメモリ	256MB	192MB
ディスプレイ	3.5 インチ 320 x 480 ドット	3.2 インチ 320 x 480 ドット
カメラ	300 万画素オートフォーカスカメラ	320 万画素オートフォーカスカメラ
OS	iPhone OS 3.2	Android 1.6
開発言語	Objective-C++	Java

実験のためのソフトウェアとして、カメラから取得した映像を解析し、ビジュアルマーカを認識して 3D オブジェクトを表示するソフトウェアを iPhone と Android 向けにそれぞれ実装した。AR ライブラリとしては各プラットフォーム向けに ARToolkit の派生ライブラリである NyARToolkit を移植して使用した。実験に NyARToolkit を利用した理由としては、C++ と Java の実装があったため、比較が容易だったためである。実験内容は、両端末で実験用ソフトウェアを同じ環境下で動作させ、その動作速度を計測した。実験前の予測としては、ガベージコレクション等のメモリ管理の特性から、Java 版 NyARToolkit は C++ 版より若干遅いと予想された。結果を表 7.2 に示す。

iPhone 3GS では 10fps 以上の十分な速度が得られ、実用上問題無い事がわかった。一方、Android Dev Phone 1 では平均しても 1~2fps 程度の速度しか出なかった。ゲーム等の OpenGL ES を使った他のアプリケーションは比較的高速に動作している事、NyARToolkit を PC 上で動作

表 7.2 iPhone 3GS と Android DevPhone 1 における NyARToolkit 実行速度

端末	1 フレームあたりにかかる時間 (ミリ秒)	fps (1 秒あたりのフレーム数)
iPhone 3GS	66.80	14.97
Android Dev Phone 1	657.14	1.52

させた場合も十分な速度が出ている事から、この実験で低速になってしまった理由はカメラからの画像取得自体が低速であるためと考えられる。以上の結果から、uZi の実装は iPhone 3GS で行う事とした。

7.1.2 センサ情報アクセス速度評価

次に、センササーバ上に蓄えられているセンサ情報へのアクセス速度を計測した。通常、無線センサネットワークは数カ月から数年程度の運用が行われる。この間に蓄えられるセンサ情報は膨大であり、その中からセンサ情報を取得するにはある程度の時間がかかると予想される。uZi はカメラから取得した映像の上に情報を投影する。そのため、情報の取得に大幅な遅延がすると、画面に映されている映像と情報の間に齟齬が発生してしまう。この実験では、新宿御苑で行われた AiryNotes の実証実験を参考にして、センサ情報をデータベース上に蓄積した。1 ヶ月間に 100 個のセンサを動かし続け、それらから得られた情報全てがデータベースに蓄積されたと仮定した。無線センサノードからのセンサ情報の送信は 1 分間に 1 回とした。センサの示す値は、実際にセンサから取得した値を元にしてランダムに生成した物を使用した。センサデータベースとして使用したハードウェア・ソフトウェア構成を表 7.3 に示す。

表 7.3 センサデータベースのハードウェア・ソフトウェア環境

CPU	AMD Duron 807MHz
メモリ	768MB
ハードディスク	1TB
OS	Ubuntu Server Edition 9.10 (Linux Kernel 2.6)
データベースソフトウェア	MySQL 5.1

実験内容としては、このようなデータベースに対して、ランダムに決定された時間に保存されたセンサ情報を取得するクエリを 1 万回出し、その結果が得られるまでの時間を計測して平均を出した。情報取得の対象とするセンサの個数は、1 個、10 個、100 個とし、それぞれの時間を計測した。iPhone 3GS 上の NyARToolkit は上記の実験の結果 15fps で動作する事がわかっているため、少なくとも 66 ミリ秒以内に情報を取得できなければ、センサデータベースへのアクセスが速度上の

ボトルネックとなってしまう。実験の結果を図 7.1 に示す。

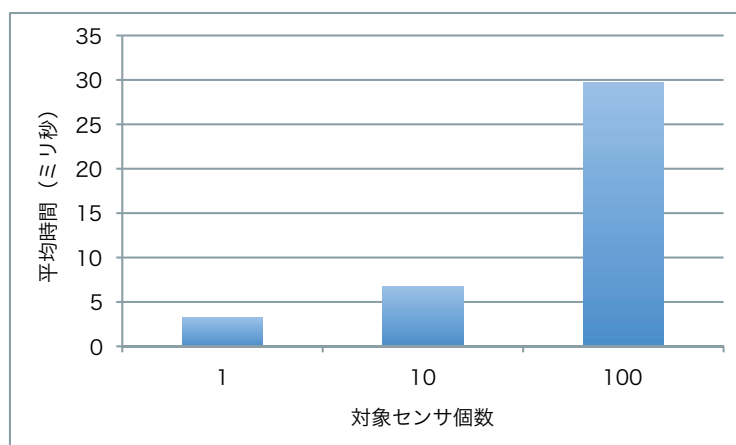


図 7.1 センサ情報アクセス速度

実験の結果から、データベース上のセンサ情報へのアクセスはいずれも 30 ミリ秒以内に終わっており、比較的高速に行える事がわかった。つまり、iPhone 3GS での動作に関しては 100 個のセンサを情報取得対象とする場合でも実用上問題無い事がわかった。実際の運用では、uZi はビジュアルマーカを認識したセンサに対してのみ情報の取得を行う。uZi は iPhone 3GS で運用される事から、一度にカメラに映るビジュアルマーカは高々 10 個程度と考えられるため、センサの取得時間にはかなりの余裕があると考えられる。さらに、今回の実験で用いたセンサデータベースのハードウェア構成は 1 世代以上前の物を用いている。また、データベースの構成についても特に高速化のための工夫は行っていない。そのため、より高速なハードウェアを使ったり、センサ情報に特化したデータベースを作成する等の工夫により、データベースへのアクセス速度はより高速になると考えられる。ただし、センササーバからの情報の取得は Wifi を経由して行うため、実際の運用では無線ネットワークの遅延によってより低速になる可能性がある。

7.1.3 uZi 速度評価

そこで、センサデータベースへのアクセス速度に関する実験を踏まえ、実際に uZi を運用した際の実行速度を計測した。この実験では、実際にビジュアルマーカを読み取り、センサ情報を取得してその情報を描画するまでの時間を計測した。なお、入力フィルタとして SunSPOT フィルタ、時系列フィルタとしてリアルタイムフィルタ、出力フィルタとしてテキスト表示フィルタと抽象表示フィルタを選択し、その他のフィルタは無効にした。情報取得の対象とするセンサは 1 個のみとした。時系列フィルタにリアルタイムフィルタを使用するため、センササーバではデータベースへのアクセスは発生しない。結果を表 7.4 に示す。

この結果から、テキスト表示フィルタと抽象表示フィルタの両方で実用上十分な速度が得られる事がわかった。また、テキスト表示フィルタよりも抽象表示フィルタの方が高速に動作する事がわ

表 7.4 uZi 実行速度

フィルタ	1 フレームあたりにかかる時間 (ミリ秒)	fps (1 秒あたりのフレーム数)
テキスト表示フィルタ	80.08	12.49
抽象表示フィルタ	67.13	14.90

かった。抽象表示フィルタは OpenGL ES を直接操作しているため高速に動作する。一方、テキスト表示フィルタは Cocoa Touch フレームワークを介して文字列の表示を行っているため、比較的低速な動作になる。そのため、FPS に差が出たと考えられる。

タイムマシンフィルタを用いる場合には、この結果にデータベースへのアクセス時間が追加される事になる。この場合でも、テキスト表示フィルタ、抽象表示フィルタともに実用上問題無い事がわかった。

7.1.4 フィルタリング機構

従来の AR アプリケーションにおけるフィルタリング機構では、全てのフィルタが並列に扱われていた。このようなフィルタリング機構では重複する機能があってもフィルタの再利用性は失われてしまい、開発者に多大な労力を課す事になってしまう。uZi では、レイや構造を持つフィルタリング機構を採用する事により、フィルタの再利用性を高めている。そこで、フィルタを並列に扱うプロトタイプと、フィルタをレイや構造で扱う uZi でコード量が増えるか調べた。比較するフィルタとしてリアルタイムフィルタと異常状態検知フィルタを選んだ。まずリアルタイムフィルタのコード量を比較した。次に、それぞれのコードに異常状態検知フィルタを追加した時の増加するコード量を調べた。比較対象としたコードは、同一人物が記述し、ほぼ同じ動作をする。ただし、uZi のコードではセンサを扱うコードの他にフィルタの管理のためのコードが追加されている。表 7.5 はその結果である。

表 7.5 フィルタのソースコード行数比較

	リアルタイムフィルタ のみの場合 (行数)	異常状態検知フィルタ を追加した場合の増加量 (行数)	合計 (行数)
プロトタイプ	306	434	740
uZi	497	110	607

リアルタイムフィルタのみの場合、プロトタイプの方が uZi よりもコード量が少なく済む。これは、uZi ではフィルタをレイや構造で管理する必要があるため、その分コード量が多くなるためである。これらのコードに異常状態検知フィルタを追加すると、プロトタイプの方が uZi に比べ約 4 倍のコードが必要になった。これは、uZi では異常状態の判別をするコードのみを追加したのに

対し、プロトタイプではセンサ情報の取得や描画処理等のリアルタイムフィルタと重複する機能も記述しなければいけなかったためである。今回の比較ではフィルタ実装の一例を比較しただけであるが、uZi はフィルタを追加した時のコード量が少なく済む事がわかった。この例ではリアルタイムフィルタと異常状態検知フィルタのみを扱ったが、Twitter や時刻表等の情報源を増やしたり、所有者によるフィルタリング等の操作を増やすに従って、プロトタイプは uZi に比べコード量が膨大な量になっていくと考えられる。

7.2 ユーザビリティ評価

本稿の提案手法を評価するために、実際に被験者に uZi を使用してもらう事でユーザビリティ評価を行った。

7.2.1 第一回ユーザビリティ評価

まず最初に情報取得手法としての uZi の肉体的、精神的負荷の観点から評価した。uZi の他に PC と Web ブラウザと、携帯電話と QR コードを用いたユーザインタフェースを比較手法として用意した。被験者には、センサ情報、Twitter のつぶやき、交通機関の時刻表を上記インターフェースを用いて取得してもらった。被験者は 11 人で、年齢は 10 代後半から 20 代前半だった。実験環境は室内とし、取得対象とする情報を記載した紙を実験環境内に複数配置した。まず、被験者には実験の趣旨を説明し、取得対象とする情報を指示した。被験者はその情報の記載してある場所まで行き、上記インターフェースのいずれかを使用して情報を取得する。この操作を全てのインターフェース、対象情報に対して行い、最後にアンケートに回答してもらった。アンケート結果を表 7.6 に示す。

表 7.6 第一回ユーザビリティ評価アンケート結果

設問	PC と Web インターフェース	携帯電話と QR コード	uZi
実験において最も負荷が高かった	4	7	0
実験において最も負荷が低かった	1	0	10

アンケートの結果から、実験において uZi は最も負荷が低いという結果が得られ、uZi の有用性が確認できた。しかし、被験者からのコメントとして、表示される情報量が少ないという意見が得られた。これは、uZi は現実空間の映像上に情報を重ね合わせるため、他の 2 手法に比べると情報の表示領域が狭くなってしまうためだと考えられる。今後の課題として、情報の表示手法や抽象的表現の改善が挙げられる。

7.2.2 第二回ユーザビリティ評価

次に，uZi の情報表現の評価を行った．評価実験の対象は年齢 20 歳から 25 歳までの 7 人とした．被験者には，実験環境に設置されたセンサ 10 個の数値を読み取ってもらった．被験者は，従来の携帯電話による二次元バーコード撮影と Web インターフェースを使ったセンサ情報の取得と，uZi のリアルタイムフィルタと異常検知フィルタ，抽象表示フィルタでセンサ情報の取得ができる．実験環境は室内とし，センサの横に二次元バーコードとビジュアルマーカを貼り付けた．被験者が携帯電話で二次元バーコードを撮影すると，携帯電話の Web ブラウザにその時のセンサ情報についての HTML ページを表示する．また，uZi でビジュアルマーカを覗き見ると，そのビジュアルマーカに対応したセンサ情報が表示される．参加者の内，3 人はセンサを使った環境モニタリングシステムを扱った経験があった．参加者には 5 分程度かけて携帯電話と uZi の簡単な説明をした後，携帯電話と uZi でセンサ情報を取得してもらった．実験後，参加者に uZi システムの使いやすさについての質問をアンケート形式で 5 段階評価で答えてもらった．アンケートの結果を表 7.7 に示す．

表 7.7 第二回ユーザビリティ評価アンケート結果

	1	2	3	4	5	平均
異常検知を行う際には uZi を使いたい	0	0	1	4	2	4.1429
uZi はセンサ情報を 可視化する際には有用である	0	0	0	2	5	4.7142
uZi はデータベースを 直接操作するよりも使いやすい	0	1	0	0	6	4.5714
Web インターフェースよりも uZi を使いたい	0	0	3	2	2	3.8571
二次元バーコードと携帯電話を用いて センサ情報を取得するよりも uZi を使いたい	0	0	0	1	6	4.8571

アンケートの結果から，参加者の多くは，uZi が使いやすいと感じた事がわかった．しかし，センサに関するシステムを扱った経験のある被験者 3 人については，uZi よりも Web インターフェースを好む事がわかった．これは，遠方にあるセンサ等の情報を取得するには PC とネットワークを通じた作業の方が労力が少ないという観点からの結果であった．uZi は遠方の情報を取得する用途は想定していないため，このような要求については従来の情報取得手法を併用する必要がある．ただし，二次元バーコードと Web インターフェースを使った操作に対しては，uZi の方がより好評であった．

また、コメントとしては「二次元バーコードと Web インターフェースを使った情報の読み取りより uZiの方が存在を意識しなかった」や「使用用途によっては二次元バーコードよりも理解しやすいと感じた」といった回答が得られ、uZiの有用性が確認できた。しかし、物体の認識にビジュアルマーカを貼り付けなければいけない事に対し、環境の景観を損ねたり、ビジュアルマーカがカメラに映るように意識しなくてはならないという不満がある事が分かった。さらに、ビジュアルマーカを貼り付ける事への困難さも指摘された。環境によっては認識対象となる物体は膨大な量になる事が考えられる。例えば、コンビニやスーパーマーケットのような大量の商品が存在する場所である。このような環境で物体全てにIDの重複が起きないようにビジュアルマーカを貼り付けるのは困難である。

7.3 本章のまとめ

本章では、uZiの評価を行った。まず、uZiは携帯端末上での動作を前提としている事から、携帯端末上での描画速度とデータベースへのアクセス速度についてパフォーマンスを評価した。次に、フィルタリング機構のフィルタの再利用性について、実際のソースコードを元に評価した。最後に、実際にユーザにuZiを使用してもらい、ユーザビリティ評価を行った。いずれの評価においても、uZiは実用上問題無く動作する事と、本稿の目的を達成できている事が確認された。次章では、本稿をまとめ、最後に今後の課題と展望を述べる。

第 8 章

結論

本稿では，現実空間に関連した仮想空間上の情報を可視化する uZi を設計・実装した．本章では，今後の展望と課題について述べ，本稿をまとめる．

8.1 まとめ

本稿では、現実空間に関連した仮想空間上の情報を可視化する uZi を設計し、実装した。情報の可視化手法としては AR 技術を用いて、現実空間の映像に情報を重ね合わせて表示する。これにより、ユーザは現実空間の物体や空間的位置関係と関連付けながら仮想空間上の情報を確認できる。特に、センサや情報家電等の現実空間と密接に関係するデバイスの情報や、Twitter や時刻表等の現実空間に関連する Web 上の情報の可視化に有効であると考えられる。

本稿では、まず uZi のプロトタイプを実装し、予備実験を行った。その結果、PC や HMD を用いるハードウェア構成はユーザの負担や不満が大きい事がわかった。そこで、uZi はユーザの多くが持ち歩いていると考えられる携帯端末をユーザデバイスとして用いた。また、実装の経験から表示する情報を絞り込むフィルタの実装に多大な労力がかかる事がわかった。uZi では、フィルタの再利用性を高めるために、フィルタリング機構にレイヤ構造を採用した。

これらの手法や設計を元に uZi を実装し、定量的、定性的に評価実験を行った。情報の取得から画面への描画までの速度やデータベースへのアクセス速度を計測し、実用上問題無い事を確認した。また、フィルタリング機構の有用性を確認するためにプロトタイプと uZi のソースコードの比較を行い、開発者の負担を低減できる事がわかった。定性的評価として、実際に uZi を被験者に使ってもらった事でユーザビリティ評価を行った。その結果、uZi が現実空間に関連した仮想空間上の情報を可視化する際に有用である事を確かめられた。

8.2 今後の展望と課題

本節では uZi の実装や評価から分かった課題と、それに基づいた今後の展望を整理する。

8.2.1 ビジュアルマーカを用いない物体の認識

ユーザビリティ評価において、ビジュアルマーカを貼り付ける事への不満がある事がわかった。環境への景観を損ねるという不満に対しては、不可視マーカ [22] を使うという解決方法がある。

8.2.2 フィルタ追加機能

uZi の実装の結果、フィルタリング機構にレイヤ構造を採用する事でフィルタの再利用性が高められる事がわかったが、現在 uZi はユーザがインストール後にフィルタを追加できない。今回の実装では Objective-C++ でフィルタの実装を行った。そこで、フィルタを Javascript のようなスクリプト言語で実装できるようにすれば、ユーザが自由にフィルタを追加できる余地が生まれると考えられる。しかし、携帯端末のような低い処理性能でもリアルタイムに実行可能になるように注意しなくてはならない。今後の展望としては、処理性能を落とさないようにフィルタの追加を可能にする事が考えられる。

8.2.3 ビジュアルマーカと物体の紐付けの自動化

ユーザビリティ評価において、被験者から環境内に膨大な物体がある場合にはビジュアルマーカと物体情報のビジュアルマーカの紐付けを行うのは困難という指摘を頂いた。コンビニやスーパーマーケットでは数千品目以上の商品が扱われ、これら全てにユーザが手作業でビジュアルマーカを貼り付ける事は不可能であるように思える。

しかし、近年、トレーサビリティに関する研究や製品開発が進んでいる。特に食の安全に対する意識の高まりとともに食品のトレーサビリティを確保する要求は高まっている。近年注目されている RFID の代表的な使用例としてもトレーサビリティアプリケーションが挙げられる。また、Fedex 社の運送サービスでは荷物にセンサを取り付け、運送中の荷物の状態を確認できるサービスが始まっている [5]。今後、同様にセンサを使ったトレーサビリティ確認サービスは増大していくと思われる。このようなサービスで、センサにビジュアルマーカを貼り付けておく事で、ユーザが手作業でビジュアルマーカを貼り付ける必要はなくなる。

謝辞

本研究の機会を与えてくださり，絶えず丁寧なご指導を賜りました，慶應義塾大学環境情報学部教授徳田英幸博士に深く感謝致します。また，貴重なご助言を頂きました慶應義塾大学環境情報学部准教授高汐一紀博士に深く感謝致します。

また，慶應義塾大学徳田・高汐・中澤研究室の諸先輩方には，貴重な御助言を頂き，多くの議論の時間を割いて頂きました。特に，政策・メディア研究科特別研究助教青木崇行氏には，本論文の執筆にあたってご指導を頂きました。また，環境情報学部講師中澤仁博士，政策・メディア研究科特別研究助教由良淳一氏，政策・メディア研究科特別研究助教伊藤昌毅には，本研究をすすめるにあたって多くの励ましとご指導を頂きました。ここに，深い感謝の意を表します。

最後に，研究生活を支えてくれた家族，ともに論文の執筆を行い，多くの研究生活をともに過ごした小川正幹氏，唐津豊氏，天野雅哉氏，また研究の日々を共に過ごした ACE 研究グループ，その他多くの友人に深く感謝し，謝辞と致します。

2010 年 2 月 12 日

今枝 卓也

参考文献

- [1] E.C. Adam. Fighter cockpits of the future. In *Digital Avionics Systems Conference, 1993. 12th DASC., AIAA/IEEE*, pp. 318–323, Oct 1993.
- [2] Michael Beigl, Christian Decker, Albert Krohn, Till Riedel, and Tobias Zimmer. uparts: Low cost sensor networks at scale. 2005.
- [3] Gabriele Bleser, Harald Wuest, and Didier Stricker. Online camera pose estimation in partially known and dynamic scenes. In *ISMAR '06: Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 56–65, Washington, DC, USA, 2006. IEEE Computer Society.
- [4] Tonchidot Corporation. Sekai camera. <http://sekaicamera.com/>.
- [5] FedEx. Senseaware. <http://www.senseaware.com/>.
- [6] Daniel Goldsmith, Fotis Liarokapis, Garry Malone, and John Kemp. Augmented reality environmental monitoring using wireless sensor networks. *Information Visualisation, International Conference on*, Vol. 0, pp. 539–544, 2008.
- [7] Apple Inc. iphone. <http://www.apple.com/jp/iphone/>.
- [8] M. Ito, Y. Katagiri, M. Ishikawa, and H. Tokuda. Airy notes: An experiment of microclimate monitoring in shinjuku gyoen garden. In *Networked Sensing Systems, 2007. INSS '07. Fourth International Conference on*, pp. 260–266, June 2007.
- [9] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *IWAR '99: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, p. 85, Washington, DC, USA, 1999. IEEE Computer Society.
- [10] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *ISMAR '07: Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 1–10, Washington, DC, USA, 2007. IEEE Computer Society.
- [11] Pranav Mistry, Pattie Maes, and Liyan Chang. Wuw - wear ur world: a wearable gestural interface. In *CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pp. 4111–4116, New York, NY, USA,

2009. ACM.
- [12] Tadashi Okoshi, Shirou Wakayama, Yousuke Sugita, Soko Aoki, Takeshi Iwamoto, Jin Nakazawa, Tomohiro Nagata, Daichi Furusaka, Masayuki Iwai, Akihiko Kusumoto, Noriyuki Harashima, Nobuhiko Nishio, Yoshito Tobe, Yasushi Ikeda, and Hideyuki Tokuda. Smart space laboratory project: Toward the next generation computing environment. In *in International Workshop on Networked Appliances*, 2001.
 - [13] Jun Rekimoto and Katashi Nagao. The world through the computer: computer augmented interaction with real world environments. In *UIST '95: Proceedings of the 8th annual ACM symposium on User interface and software technology*, pp. 29–36, New York, NY, USA, 1995. ACM.
 - [14] Luke Simon, Ajay Mallya, Ajay Bansal, Gopal Gupta, and Thomas D. Hite. A universal service description language. In *ICWS '05: Proceedings of the IEEE International Conference on Web Services*, pp. 823–824, Washington, DC, USA, 2005. IEEE Computer Society.
 - [15] Inc. Sun Microsystems. Sun spot 無線センサーネットワークデバイス. <http://jp.sun.com/products/software/sunspot/>.
 - [16] Genta Suzuki, Shun Aoki, Takeshi Iwamoto, Daisuke Maruyama, Takuya Koda, Naohiko Kohtake, Kazunori Takashio, and Hideyuki Tokuda. u-photo: Interacting with pervasive services using digital still images. In *Pervasive*, pp. 190–207, 2005.
 - [17] Sean White and Steven Feiner. Sitelens: situated visualization techniques for urban site visits. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pp. 1117–1120, New York, NY, USA, 2010. ACM.
 - [18] Takuro Yonezawa, Hiroshi Sakakibara, Kengo Koizumi, Shingo Miyajima, Jin Nakazawa, Kazunori Takashio, and Hideyuki Tokuda. upackage - a package to enable do-it-yourself style ubiquitous services with daily objects. In *UCS*, pp. 240–257, 2007.
 - [19] クロスボー株式会社. Moteview. <http://www.xbow.jp/moteview12.html>.
 - [20] クロスボー株式会社. センサーネットワーク mote official page. <http://www.xbow.jp/motemica.html>.
 - [21] 株式会社システム・ケイ. Layer. <http://layar.jp/>.
 - [22] 祐介中里, 誠之神原, 直和横矢. 不可視マーカを用いたウェアラブル ar システムの実環境における実証実験 (テーマセッション (2))(ユビキタスコンピューティングのためのパターン認識・メディア理解, 一般). 電子情報通信学会技術研究報告. PRMU, パターン認識・メディア理解, Vol. 104, No. 524, pp. 7–12, 20041210.