

修士論文 2009 年度 (平成 21 年度)

実世界オブジェクト選択の為の
スイング動作に関する研究

慶應義塾大学大学院 政策・メディア研究科
河田 恭兵

kyo@ht.sfc.keio.ac.jp

修士論文要旨 2009 年度 (平成 21 年度)

実世界オブジェクト選択の為のスイング動作に関する研究

論文要旨

近年、実世界上のオブジェクトを選択する行為に対する重要性が高まっている。モバイル端末が急速に普及し、また多くのオブジェクトがネットワークに関連付けられるようになった。さまざまな家電製品がネットワークに繋がり、ポスターなどの情報媒体には Web サイトなどのアドレスが記載されている。こうした状況で、ユーザはモバイル端末 1 つで複数の情報家電を操作したり、情報媒体に関連付けられたネットワークデータにアクセス出来るようになった。しかし、実世界に存在するオブジェクトからネットワーク上のオブジェクトを利用する際、異なる空間に存在する 2 つのオブジェクトを頭の中で結びつけることが難しい。例えば、同じ種類のネットワークライトが複数あった時、ネットワーク上のリストから一番右側に設置されているライトを選び操作することは困難である。この問題を解決するため、実世界のオブジェクトに直接インタラクトすることで直感的に選択を行う手法が盛んに研究されている。しかし、実世界でインタラクションを行うためにそれぞれの手法には利用しにくいシチュエーションが存在する。例えば、QR コードを用いた手法はカメラで撮影するインタラクションを用いるため、離れた位置からでは利用出来ない。また、利用状況によっても選択手法に求められる要件が大きく異なってくる。

本論文では、パブリック空間において利用しやすい選択手法の要件として 1) 環境側にデバイスを設置しない事、2) 距離的な制約を解消する事、3) インタラクションを誘発する事の 3 つを挙げた。多くのオブジェクトが存在する空間では、オブジェクト選択のためにデバイスを設置するのは現実的ではない。また、移動が困難な状況が考えられるため、移動することなくその場でオブジェクトを選択出来るべきである。最後に、空間におけるオブジェクトの利用を活発にするため、選択行為が周囲のユーザに対して視覚的に見えるべきである。

これらの要件を満たす選択手法として、本論文ではモバイル端末を振る動作を用いた選択手法を提案する。各オブジェクトにモバイル端末をどう振るか示したコードを貼付ける。ユーザはオブジェクトに貼られたコードを目視し、モバイル端末でコードを再現することで選択が可能となる。本論文では、本手法の設計を行うとともに、振る動作の使いやすさを決定する要因を分析し、設計したコードの体系化を行う。

キーワード

振る動作、オブジェクト選択、ヒューマンコンピュータインタラクション、実世界指向インタフェース、ユビキタスコンピューティング、モバイル端末

慶應義塾大学大学院 政策メディア研究科

河田 恭兵

Abstract of Master's Thesis Academic Year 2009

A Study of “Swing Interaction for Real World Object Selection”

Summary

In recent years, importance to the action of selecting real world objects has been increasing. Mobile terminal has spread very rapidly, and many objects came to be related to the network. For example, various home appliance is connected to the network, Web Address are listed on information medium such as posters. In such circumstances, the user came to be able to operate one or more information appliances, access to the network data which has been related to the information medium by a single mobile terminal. However, when using an object that exists on the network from the real world objects, it is difficult to link two objects that exist in a different space in the head. For instance, when there are two or more same kind network lights exist on the network, it must be difficult to choose to operate the light placed at the most right from the list. To solve this problem, the technique to intuitively select the real world object by directly interact to it has been studied actively. However, for each approach to make interaction in the real world, there is situation that the approach is hardly handled. For example, because the QR code based techniques uses the interaction to take a picture, it is not available at a distance. Moreover, the requirement to the selection method greatly differs at the use situations. In this paper, requirement for the easily accusable selection technique being used in the public space are the following: 1) Not to place the device at the environmental side, 2) To eliminate the distance constraints, 3) To induce the user's interaction. Space where a lot of objects exist, it is not realistic to set up a device for the object selection. Moreover, for there may be a situation the user is hardly move over, the object must be selected with out moving. Finally, in order to boost the use of objects in the space, it is necessary the selection act to be visual by the surrounding users.

To satisfy these requirements, in this paper proposes a selection technique using the action of swinging a mobile terminal. Attach a code to show how to swing a mobile terminal to each object. The user sees the code attached to the object, and the user can reproduce the code on the mobile terminal. In this paper, design this method while analyzing the factor to determine the easiness of the swing operation and schematizing the designed code.

Keywords

Swing Interaction, Object Selecting, Human Computer Interaction, Real World Oriented Interface, Ubiquitous Computing, Mobile Device

**Keio University Graduate School of Media and Governance
Kyohei Kawada**

目次

第 1 章	序論	1
1.1	本論文の背景	2
1.2	本研究の目的	3
1.3	本論文の構成	3
第 2 章	実世界オブジェクト選択手法	5
2.1	実世界オブジェクトの選択	6
2.1.1	実世界オブジェクトからのサービス利用	6
2.1.2	実世界オブジェクト選択の定義	6
2.1.3	実世界オブジェクト選択における問題	7
2.2	既存の選択手法	8
2.3	利用環境の分類	13
2.3.1	プライベート空間	14
2.3.2	パブリック空間	14
2.4	機能要件	15
2.5	既存手法の比較	16
2.6	本章のまとめ	18
第 3 章	振る動作を用いた選択手法	19
3.1	振る動作を用いた選択手法の概要	20
3.2	振る動作の設計	21
3.2.1	SR コードの機能要件	21
3.2.2	SR コードの基本設計	23
3.2.3	操作ジェスチャの要件	25
3.2.4	操作ジェスチャの設計	26
3.3	フィードバックの設計	27
3.3.1	フィードバックの要件	27
3.3.2	音によるフィードバック	27
3.3.3	GUI によるフィードバック	28

3.4	振る動作の解析	28
3.4.1	加速度センサを用いた解析	28
3.4.2	4 方向に振る動作の推測	29
3.4.3	操作ジェスチャの認識	31
3.5	SR コードの検証	32
3.5.1	振る動作推測アルゴリズムの検証	32
3.5.2	8 方向 SR コードの検証	33
3.6	考慮すべき問題	34
3.7	本章のまとめ	35
第 4 章	SR コードの使いやすさの分析	36
4.1	SR コードの使いやすさ	37
4.1.1	振る動作を用いた選択行為の行動モデル	37
4.1.2	振りやすさの指標	37
4.2	パターンによる振りやすさの検証	38
4.3	2 回振るコードの振りやすさの検証	40
4.4	各 SR コードの振りやすさ	42
4.5	振り始めの位置による影響の検証	45
4.6	振りやすさの順位付け	45
4.6.1	5 回振る SR コードの振りやすさ	47
4.7	本章のまとめ	48
第 5 章	InfoRod の設計と実装	49
5.1	InfoRod システムの概要	50
5.2	InfoRod システムの実現方法	50
5.3	ユーザ用アプリケーションの設計	51
5.3.1	ソフトウェア構成	51
5.3.2	ハードウェア構成	52
5.4	管理者用アプリケーションの設計	53
5.4.1	ソフトウェア構成	53
5.5	データベースの設計	54
5.6	ユーザ用アプリケーションの実装	56
5.6.1	ハードウェア	56
5.6.2	ソフトウェアの実装	56
5.6.3	利用方法	57
5.7	管理者用アプリケーションの実装	57
5.7.1	ソフトウェアの実装	58

5.8	本章のまとめ	59
第 6 章	評価実験	60
6.1	実験概要	61
6.1.1	実験環境	61
6.1.2	被験者	63
6.1.3	実験手順	63
6.2	実験結果	63
6.3	考察	64
6.4	本章のまとめ	65
第 7 章	結論	67
7.1	本論文のまとめ	68
7.2	今後の展望	68
	参考文献	71

目次

2.1	実世界オブジェクトの選択における問題	8
2.2	Fractal Codes	9
2.3	u-Photo によるライトの操作	9
2.4	Gaze-Link	10
2.5	CiteULike	10
2.6	LookPoint	10
2.7	objSampler	11
2.8	tcts	11
2.9	マルチリモコン クロッサム	12
2.10	Select-and-Point による指先の認識	12
2.11	Select-and-Point におけるジェスチャ操作	12
2.12	Pick-and-Beam	12
2.13	Toss-It	14
2.14	イベント会場の様子	16
3.1	振る動作を用いた選択手法の概念図	20
3.2	SR コードの例	23
3.3	上下に折り返す SR コードの例	23
3.4	入力のキャンセル	27
3.5	振る動作の決定	27
3.6	周辺オブジェクトの検索	27
3.7	GUI によるフィードバックの例	28
3.8	iPhone	29
3.9	iPhone における加速度の座標系	29
3.10	上方向に振った時の加速度の変化	30
3.11	下方向に振った時の加速度の変化	30
3.12	右方向に振った時の加速度の変化	30
3.13	左方向に振った時の加速度の変化	30
3.14	ニューラルネットワークの概念モデル	31

3.15	8 方向の SR コードの例	34
3.16	縦に伸びた 8 方向の SR コードの例	34
4.1	パターンによる検証用 6 種類の SR コード	39
4.2	2 つの振りを組み合わせた SR コード	41
4.3	手首を使ったスイング動作	41
4.4	腕を使ったスイング動作	42
4.5	2 回振るコードの振りやすさの評価結果	43
4.6	振る動作の位置を表す座標系	43
4.7	座標系における振り始め位置	44
4.8	SR コードの分割	44
4.9	上と下で振り始めたときの評価結果	46
4.10	右と左で振り始めたときの評価結果	46
4.11	斜めで振り始めたときの評価結果	46
4.12	振りやすい 5 回振る SR コード	47
4.13	振りにくい 5 回振る SR コード	47
5.1	ユーザ用アプリケーションのソフトウェア構成図	53
5.2	管理者用アプリケーションのソフトウェア構成図	54
5.3	ユーザ用アプリケーションの画面遷移	58
5.4	管理者用アプリケーションの画面	59
6.1	実験環境	61
6.2	評価実験で用いた SR コード群	62

表目次

2.1	既存の実世界オブジェクト選択手法の比較	18
3.1	SR コードの生成可能数	24
3.2	加速度の変化の特徴	30
3.3	振る動作の認識率	33
3.4	パターン生成可能数の比較	34
4.1	パターンによる検証の実験結果	40
4.2	2 回振る動作の振り始め位置	44
5.1	iPhone のスペック	56
6.1	評価実験の結果	63
6.1	評価実験の結果	64
6.2	順位の比較	65

第 1 章

序論

本章では，はじめに本研究の背景，目的について説明し，その後，本論文の構成について述べる．

1.1 本論文の背景

近年、実世界上に存在するオブジェクトからサービスを利用する際のオブジェクトを選択する行為に対する重要性が高まっている。

携帯電話や PDA などのモバイル端末が急速に普及し、1 人 1 台モバイル端末を持つ状況になった。さらに、多くのオブジェクトがネットワーク接続性を持つようになり、ユーザは自身のモバイル端末から情報家電を操作したり、情報媒体に関連付けられたネットワーク情報にアクセスする事が可能になった。このような状況では、以前までユーザが意識せずに行っていた対象を選択する行為を意識的に行う必要がある。

情報家電を操作することを例に挙げて説明する。主にネットワークに接続されていない家電を操作する場合、直接家電のボタンを押したりリモコンを使うため、ユーザが操作する対象を意識的に選択する必要がなかった。ネットワーク接続された情報家電は、ユーザのモバイル端末からネットワーク経由で操作可能になることで、わざわざ近づく必要がなく、また全ての情報家電を 1 つのモバイル端末で操作可能になる。しかし、これにより今までユーザが意識することのなかった対象を選択する行為をユーザはモバイル端末内で行う必要が出てきてしまう。特に外見の似たオブジェクトが複数ある状況では、画面の中にあるネットワーク上の情報と実世界に存在するオブジェクトを頭の中で関連付けることは困難である。

この問題を解決するために、実世界のオブジェクトに直接インタラクトすることで直感的に選択を行える手法が盛んに研究されている。しかし、実世界でインタラクションを行うために利用しやすい状況利用しにくい状況が存在する。例として QR コードを撮影する手法を挙げる。QR コードをオブジェクトに記載することで、ユーザはカメラで QR コードを撮影することでオブジェクトを選択したり、オブジェクトに関連付けられた Web ページにアクセスすることが可能になる。この手法は、選択操作にカメラで撮影するというインタラクションを用いるため、カメラが QR コードを認識出来る距離まで近づく必要があり、また空間の照明がある程度明るい必要があるなどの制約がある。

利用状況によりオブジェクトを選択する手法に求められる要件が大きく異なってくる。利用状況をプライベート空間とパブリック空間の 2 つに分類して比較してみる。例えば、自宅などのプライベート空間での選択は、既に何度も利用したことのあるオブジェクトが多いのに対してパブリック空間では、初めて選択するオブジェクトであることが多い。また、イベント会場のポスターや、駅構内の広告など同じ空間内に存在するオブジェクトはパブリック空間の方が多くなる。さらに、オブジェクトの前に人だかりが出来るなど、パブリック空間ではユーザがオブジェクトに近づけない状況も発生する。最後に、プライベート空間に存在するオブジェクトは管理者がユーザとなるのに対し、パブリック空間に存在するオブジェクトは管理者がユーザとは別に存在し、ユーザはオブジェクトの利用だけを行う。そもそも、パブリック空間に存在するオブジェクトはユーザに利用されることを前提としているため、管理者は多くのユーザに利用されたいと考えている。そこで本論

文では、パブリック空間における選択手法の3つの要件を設定した。以下に3つの要件を挙げる。

- 1) 環境側にデバイスを設置しない事
- 2) 距離的な制約を解消する事
- 3) インタクションを誘発する事

多くのオブジェクトが存在する場合、オブジェクト選択手法のためにオブジェクトの傍にデバイスを設置するのはあまり現実的ではない。また、パブリック空間では移動が困難な状況が考えられるため、オブジェクト選択のためにわざわざユーザが移動せず、ユーザがオブジェクトを発見したその場で選択できるべきである。最後に、オブジェクトを選択する行為が周囲のユーザにも視覚的に見えるべきである。なぜなら、選択行為が見えることでオブジェクトの存在に気づいたり、興味を持つきっかけになるからである。

1.2 本研究の目的

本論文では、パブリック空間において使いやすい実世界オブジェクト選択手法として、モバイル端末を振るという動作を用いた選択手法を提案する。本手法では、モバイル端末をどのように振るか明記したコードを生成し、実世界オブジェクトにそれぞれ貼付する。ユーザは実世界オブジェクトに貼られたコードを目視し、どのようにモバイル端末を振ればいいのか認識する。そして、コードの通りに端末を振ることで実世界オブジェクトに選択し、オブジェクトからサービスを利用することが可能になる。

本論文では、オブジェクトを選択する際のモバイル端末を振るという動作を上下左右4方向の振りを複数組み合わせた動作として設計する。さらに、この選択のための振る動作を表現するコードとしてSRコードを設計した。SRコードは、上下左右の振りを棒で表し、振る順番に棒の端を繋ぎ合わせコードである。

パブリック空間内のオブジェクトの中には、多くのユーザに利用されるオブジェクトとあまり利用されないことのないオブジェクトが存在する。そのため、なるべく多くの人に利用されるオブジェクトには振りやすいSRコードを割り当てる必要がある。そこで本論文では、上下左右に振るという動作及びSRコードを再現するという動作について詳しく分析し、どのようなコードがユーザにとって振りやすいのか整理をすることをひとつの目的とする。

1.3 本論文の構成

本論文は、本章を含め全7章から構成される。第2章では、実世界オブジェクト選択手法について述べる。まず、本論文における実世界オブジェクト選択について整理し、パブリック空間における実世界オブジェクト選択手法の機能要件を導き出す。第3章では、導き出した機能要件を満たすモバイル端末を振るを用いた選択手法の提案を行う。ここではSRコードの基本設計や解析手法の

説明も行う。第4章では、振る動作についての詳しい分析を行うため、検証実験を行いながら振る動作のどこに振りやすさや振りにくさの原因があるのかを明らかにしていく。第5章では、振る動作を用いた選択手法を実現する InfoRod システムの設計と実装について説明する。特に SR コードの管理方法やコードの自動生成アルゴリズムについて言及する。第6章では、5章で設計した自動生成アルゴリズムの評価実験を行う。第7章では、本論文の結論と今後の展望を示す。

第 2 章

実世界オブジェクト選択手法

本章では，はじめに本論文における実世界オブジェクトの選択についての整理を行う．ついで，既存の実世界オブジェクト選択手法を紹介する．その後，選択手法の利用状況を整理し，本論文が対象とするパブリック空間における選択手法の機能要件を導き出す．

2.1 実世界オブジェクトの選択

本節では実世界オブジェクトの選択を整理する為に、まず本論文における実世界オブジェクトについて定義を行う。その後、実世界オブジェクト選択を説明する。

2.1.1 実世界オブジェクトからのサービス利用

近年、多くの情報家電がネットワークに接続し、ネットワーク経由で情報家電の操作を行う環境が整ってきた。情報家電がネットワークに繋がる事で、離れた場所から操作したり、1つのリモコン端末で複数の情報家電を操作する事が可能になった。1つのリモコン端末で複数の情報家電をネットワーク経由で操作する際、どの情報家電を操作するのかといった「対象を選択をする」必要が出てきた。実世界にあるモノを操作したりサービスを受ける場合、ユーザの操作をモデル化すると以下の3つのフェーズに分ける事が出来る。

1)Discovery (対象を発見する)

2)Select (対象を選択する)

3)Use (対象を利用する)

従来の操作手法では対象を選択するという行為はあまり意識せずに行われてきた。テレビのチャンネルを変えろと言う操作を例として挙げる。テレビのボタンを直接押してチャンネルを変える場合、対象の選択はテレビに近づきボタンを押す事で行われる。リモコンでチャンネルを変える場合は、専用のリモコンを持つという行為が対象の選択にあたる。このように、直接ボタンを押したり、専用のリモコンを用いる場合、対象の選択と対象の操作は、同時に行われる事が多くユーザは対象を選択するという事に関して特別意識はしない。しかし、1つのリモコン端末で複数の情報家電を操作する場合、リモコン端末内で対象を選択する必要がある為、対象の選択に意識を向けなければならない。

2.1.2 実世界オブジェクト選択の定義

実世界オブジェクトという言葉はとても広い意味を持ち、そのままの意味では実世界に存在する全てのモノが実世界オブジェクトと言う事が出来てしまう。そのため、実世界オブジェクト選択について述べる前に、本論文で対象とする実世界オブジェクトの定義をする必要がある。第2.1.1項で説明したように、多くのオブジェクトがネットワークに繋がったことにより、一つのモバイル端末でさまざまなサービスを利用すること可能になったが、オブジェクトを選択する行為の必要性が出てきた。本論文では実世界オブジェクトを**ネットワークに関連付けられた実世界に存在するモノ**とする。

さらに実世界オブジェクトはネットワークとの関連付けられ方で2種類に分類出来る。1つ目は、

直接ネットワークに繋がっているオブジェクト。2つ目は、ネットワークデータのアドレスを示すオブジェクトである。この実世界オブジェクトは、ネットワーク上において一意となる ID を保持している。本論文では、前者を**エンティティ型実世界オブジェクト**、後者を**ポインタ型実世界オブジェクト**と呼ぶ。また以下で、それぞれの実世界オブジェクトについて述べる。

エンティティ型実世界オブジェクト

エンティティ型実世界オブジェクトは、自身がネットワークに繋がっている実世界オブジェクトの事である。前述したような情報家電やパソコン、センサなどが例に挙げられる。エンティティ型実世界オブジェクトは、自身がネットワーク上で一意となる IP アドレスなどの ID を保持している。

ポインタ型実世界オブジェクト

ポインタ型実世界オブジェクトは、それ自体はネットワークに繋がっていないが、ネットワーク上のオブジェクトのアドレスを示す実世界オブジェクトである。基本的にポインタ型実世界オブジェクトは情報媒体であり、保持するネットワーク ID は自身が表示する情報に関連したネットワーク上のオブジェクトである。例として Web ページの URL が記載されたポスター、ディスプレイ、広告などが挙げられる。ポスターにはその内容に関連した Web ページの URL が記載されている。ポスターに URL を記載する事で、ネットワーク上のオブジェクトである Web ページとポスターを関連付けられていると言える。

ユーザは実世界オブジェクトに関連付けられたネットワークオブジェクトにアクセスし、サービスを利用するために実世界オブジェクトの選択を行う。ネットワークオブジェクトにアクセスするためには、オブジェクトのネットワーク ID を特定する必要がある。そのため、本研究における実世界オブジェクトの選択とは、**ネットワーク ID を取得する行為**であると言える。

また、2種類の実世界オブジェクトに関連付けられたサービスには特徴がある。エンティティ型実世界オブジェクトから利用可能なサービスは主にファンクションの実行である。ユーザはネットワーク越しにモバイル端末と実世界オブジェクトを連携させ、操作することが可能になる。ポインタ型実世界オブジェクトから利用可能なサービスは主にリソースの取得である。ユーザはポインタ型実世界オブジェクトの示すネットワーク上のリソースをモバイル端末にダウンロードすることになる。リソースの種類は Web サイトの URL から画像ファイル、音楽ファイル、動画ファイルなどさまざまである。

2.1.3 実世界オブジェクト選択における問題

実世界オブジェクト選択をする際、ユーザが直感的にオブジェクトを選べないという問題がある。選択対象を発見する行為が実世界上で行われているのに対し、ネットワーク ID を取得する行為がネットワーク上で行われているのが原因である。発見と選択という2つの行為は異なる世界で行う

必要があり，実世界オブジェクトとネットワーク ID をユーザの頭の中で結びつける事は困難である．例えば，図 2.1 に示すように，複数の同じ種類のデバイスが存在した時，その中の一つをネットワーク経由で操作したいと言う要求があっても容易に行う事は出来ない．なぜなら実世界にあるデバイスと，ネットワーク上のデバイスリストや IP アドレスを関連付ける必要があるからである．

このように，実世界とネットワーク世界の結びつきの悪さが，実世界オブジェクトを選択する際の直感性のなさの原因となっている．そこで「Back to the Real World(現実世界に戻ろう)」という考え方である実世界指向が提唱された [1]．実世界指向には様々な考え方があり，多くの研究が現在行われている．次節でこの問題を解決する手法を紹介する．

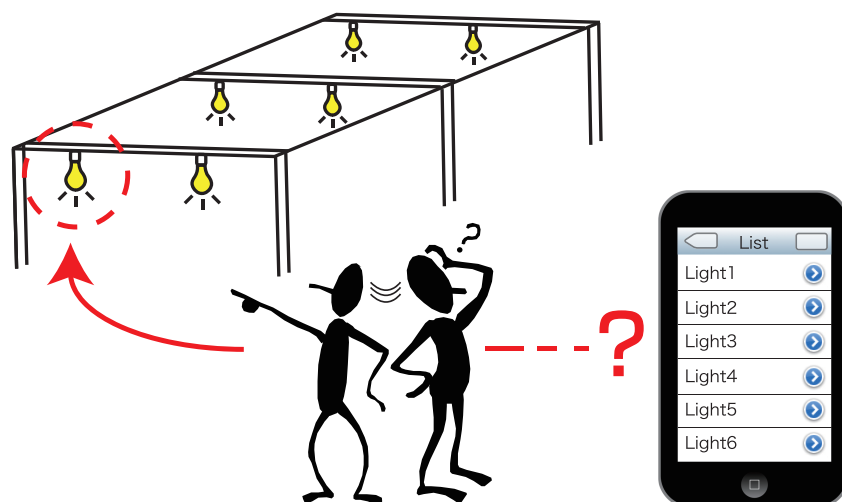


図 2.1 実世界オブジェクトの選択における問題

2.2 既存の選択手法

前述した実空間オブジェクトを選択する際の問題を解決するために，様々な選択手法が研究されている．本節では，まず選択操作時のインタラクションごとに分類し，各インタラクションごとに既存の選択手法を紹介する．

撮るインタラクション

カメラを用いた選択手法は，大きく 2 つに分類出来る．

1 つ目は，オブジェクトに 2 次元コードを貼付し，コードをカメラで認識する手法である．実際に QR コードや PDF417, Maxi Code など多くの 2 次元コードが実用化されている．また，2 次元コードに関する研究は現在も盛んに行われている．例えば，2 次元コードを認識可能な範囲が狭いという問題を解消した Fractal Codes [2] がある．Fractal Codes の例を図 2.2 に示す．Fractal Codes は，コード情報と同時にコードの座標や向きを認識可能な

Cyber Code [3] を拡張したものである。体系づけられた ID を持つ複数の二次元コードを自己相似性のあるレイアウトで配置することによって認識可能な範囲を広げている。また、オブジェクトの認識や選択に 2 次元コードを撮影するインタラクションを用いた研究は数多く存在する。慶應義塾大学 鈴木らの u-Photo [4] はオブジェクトに貼付した 2 次元コードを写真として保存することで、ビューワー上で写真の中のオブジェクトを選択し機器の操作などのサービスが利用出来る。図 2.3 に u-Photo によるライトの操作の例を示す。

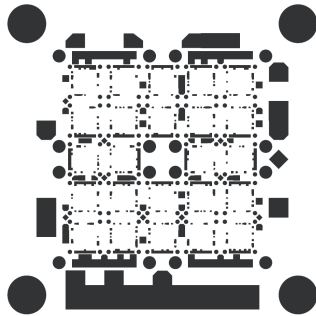


図 2.2 Fractal Codes



図 2.3 u-Photo によるライトの操作

2 つ目は、オブジェクト自体を画像として登録しておき、オブジェクト自体をカメラで認識する手法である。NTT ドコモ 山口らは看板などのオブジェクトの画像をデータベースに登録しておき、カメラで撮影するだけでオブジェクトを特定出来る手法を提案している [5]。また、事前に付加情報としてポスターや看板の関連サイトなどの URL を付加すれば、カメラで撮影しただけで情報を取得することが可能となる。さらに、2 次元コードなどを貼る必要がないため、デザイン上の制約もない。

見るインタラクション

視線を用いたインタラクションは、ユーザがオブジェクトに視線を向けることによってオブジェクトの選択を可能とする手法である。Sony CSL の綾塚らが提案する Gaze-Link [6] は、見るというメタファを用いてユーザがオブジェクトを見ている間接続を可能にする手法である。図 2.4 に Gaze-Link を用いて見ているディスプレイでキー入力を行う例を示す。オブジェクトには Cyber Code が貼られ、ユーザはカメラ越しにオブジェクトを見ることで、見ているオブジェクトへの接続が可能になる。また、綾塚らは Gaze-Link の応用例として万能リモコンを提案している。

Gaze-Link は、カメラをオブジェクトに向けることを”見る”というメタファとしているが、実際にユーザの視線を検出し、オブジェクトの選択を行っている研究事例もある。Queen's University Dickie らが提案する LookPoint [7] は、オブジェクト側にカメラを設置し視線を検出することで、ユーザが視線を向けたオブジェクトを特定している。図 2.6 に LookPoint の実装環境を示す。3 つのディスプレイの真ん中にカメラを設置しユーザの視線を検出する。それぞれのディスプレイには IR Tag が設置されていて、IR Tag から出る赤

外線の光がユーザの目に映ることによって、キャリブレーションせず利用可能になる。

Loughborough University Shi らは、ユーザ側にカメラを配置しユーザの見ているオブジェクトを特定する CiteULike を提案している [8]。図 2.5 に示すように、ユーザの頭部にユーザの視線を検出するカメラと、オブジェクトを映すカメラの 2 種類が搭載されている。これにより、ユーザの目線の先の映像の中から、ユーザが実際に見ているオブジェクトを特定することが出来る。

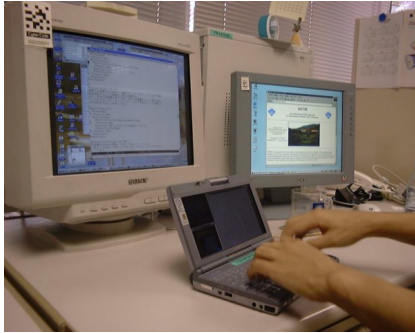


図 2.4 Gaze-Link

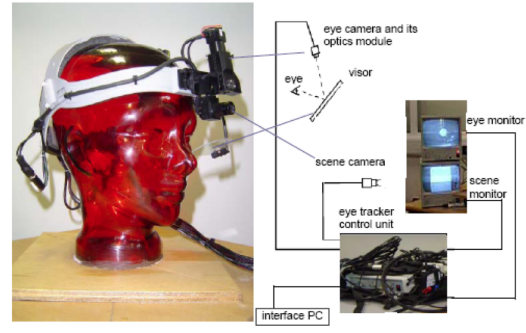


図 2.5 CiteULike



図 2.6 LookPoint

タッチインタラクション

オブジェクトに直接触る、タッチをするというインタラクションを使ってオブジェクトを特定する手法の研究も盛んである。タッチインタラクションで最も主流のアプローチは、RFID を用いたものである。ユーザ側デバイスに RFID リーダを内蔵し、各オブジェクトには RFID タグを貼ることで、ユーザ側デバイスをオブジェクトにタッチするだけでオブジェクトを選択することが可能になる。

慶應義塾大学 由良らによる objSampler [9] は、RFID リーダを内蔵した情報取得端末である。図 2.7 上部にある円柱状のデバイスを objPipette、下部にある台座を objDish と呼ぶ。パブリック空間に存在するオブジェクトの情報を objPipette をタッチするというイン

タラクションで取得する。そして、ユーザの家に置かれた objDish に図 2.7 のように収めることで、専用ソフトウェアに objPipette による取得データが送信され閲覧が可能になる。

Lancaster University Seewoonauth らの提案する Touch&Connect and Touch&Select [10] は、タッチインタラクションを使ってディスプレイ内の細かなオブジェクトの選択を可能にする。この手法もユーザ側には RFID を内蔵したモバイル端末を持たせ、オブジェクト側には RFID タグを貼付ける。図 2.8 に示すとおり、ディスプレイの裏に等間隔に RFID タグを貼付けることでユーザのタッチした場所を特定する。



図 2.7 objSampler

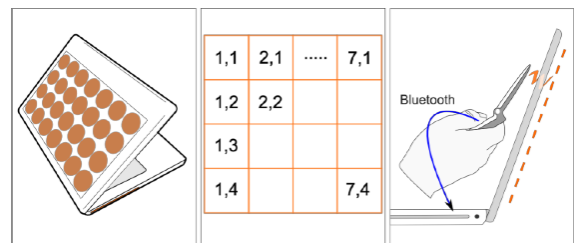


図 2.8 tcts

指し示すインタラクション

ユーザがオブジェクトに対して指し示すことで、オブジェクトを特定する手法を本論文では指し示すインタラクションと分類する。

指し示すインタラクションは、家電を操作するリモコンとして普段から使われているものである。操作する側に赤外線受信部を取り付け、リモコンに赤外線発信部を取り付けることで、リモコンをデバイスに向けてボタンを押すことで操作可能になる。専用リモコンだと 1 つのデバイスしか操作出来ないが、複数のデバイスの信号を記憶することの出来る Crossam などのマルチリモコンなどが実用化されている。図 2.9 にマルチリモコンのクロッサムを示す。しかし、このマルチリモコンはデバイス进行操作する際、図 2.9 から分かるように多くのボタン群から対象デバイスを操作するボタンを探す必要がある。そのため、指し示すというインタラクションの他に、操作するデバイスのボタンを探すという直感的ではない選択操作を行う必要がある。

KIST Yeom らの提案する Select-and-Point [11] は、ヘテロジニアスなマルチデバイス環境におけるデバイス接続をサポートするシステムである。Select-and-Point は、グループワーク環境を想定しており、素手でのラップトップ同士でのファイル交換、壁掛けディスプレイでのプレゼンテーションなどを可能にする。図 2.10 は、Select-and-Point でユーザの指先を認識している様子を示したものである。また、図 2.11 の上部では Select ジェスチャーを、下部では Point ジェスチャーを行っている様子を示す。

Pick-and-Beam は Sony CSL の暦本らが開発を行ったレーザポインタによる離れた位置にある情報を Hyper-dragging によって転送する技術である。Pick-and-Beam は、暦本

らが提唱する実世界に拡張された直接操作環境である InfoRoom [12] で用いる技術の 1 つとして提案された。Pick-and-Beam では、実世界の事物とコンピュータ間の操作である Hyper-dragging である Pick-and-Drop [13] を拡張し、レーザポインタを用いることによって離れた位置からでも Hyper-dragging を実現している。また、レーザポインタは環境側に設置されたカメラによって認識している。図 2.12 に Pick-and-Beam によるオブジェクト選択の様子を示す。



図 2.9 マルチリモコン クロッサム

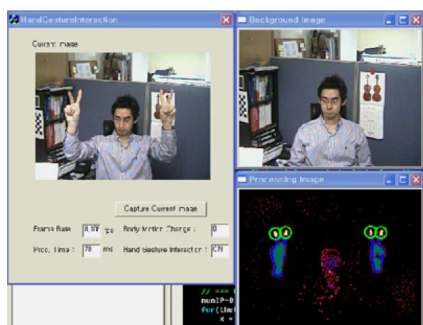


図 2.10 Select-and-Point による指先の認識

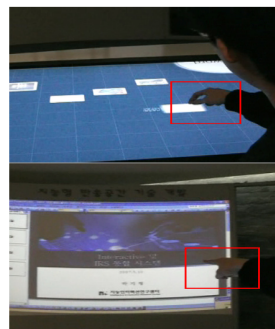


図 2.11 Select-and-Point におけるジェスチャ操作



図 2.12 Pick-and-Beam

近づくインタラクション

近づくインタラクションは、オブジェクトにユーザやモバイル端末を近づけるというインタラクションを用いた選択手法である。

NTT ドコモ 中島らの音のバーコード [14] は、音に情報を埋め込むことを可能にした技術である。オブジェクトに情報を埋め込んだ音を流すことの出来るスピーカを設置し、ユーザはマイクの内蔵されたモバイル端末を持つ。ユーザはオブジェクトに近づき、オブジェクトから流れる音をマイクで読み取るだけでオブジェクトからの情報が取得出来る。

Sony CSL の暦本らは Proximal interactions [15] という、近づくインタラクションを提案している。暦本らは、オブジェクトに近づくという状態を取得するためにワイヤレス通信における受信パケット信号強度を計測することで、近傍にある機器からの発信を特別扱いし、ネットワークの設定に利用する方法を採用している。

言葉を用いたインタラクション

声を使ったオブジェクト選択手法として、法政大学 内田らの情報家電操作のための対話インタフェース [16] が挙げられる。マイクに向かってオブジェクトの名前と操作内容を言うだけでシステムが意味を解釈し操作を実行する。しかし、この手法には欠点がある。例えば、同じ種類のオブジェクトが複数並んだ場合、言葉によって1つのオブジェクトを選択することが非常に困難になってしまう。つまり、本論文の目的である直感的な選択操作が出来なくなってしまう状況が出てくる。

身振りをを用いたインタラクション

ユーザの身振りをを用いたインタラクションの例として、東京大学 矢谷らの Toss-It [17] が挙げられる。Toss-It は、モバイル端末をトスしたり振ったりする動作を用いて、データを送信するオブジェクトを選択している。図 2.13 に Toss-It を用いた情報の受け渡しの例を示す。左の図は、トスの動作によって情報を受け渡す際、トスをする方向に対象となるオブジェクトが複数あってもトスの強弱によって選択出来ることを示している。右の図は、スイングの動作によってスイングした範囲のオブジェクトを全て選択して、データを受け渡している様子を示している。この手法では、モバイル端末の動作は内蔵された加速度センサの値を解析して推測し、赤外線 LED を環境側に配置しカメラで撮影することによって空間内の位置を認識している。

2.3 利用環境の分類

オブジェクトの選択を行う環境によって、実世界オブジェクト選択手法に求められる要件が大きく異なる。本論文では、利用環境をプライベート空間とパブリック空間の2つに大きく分け、各利用環境ごとに必要な実世界オブジェクト選択手法の要件を整理する。

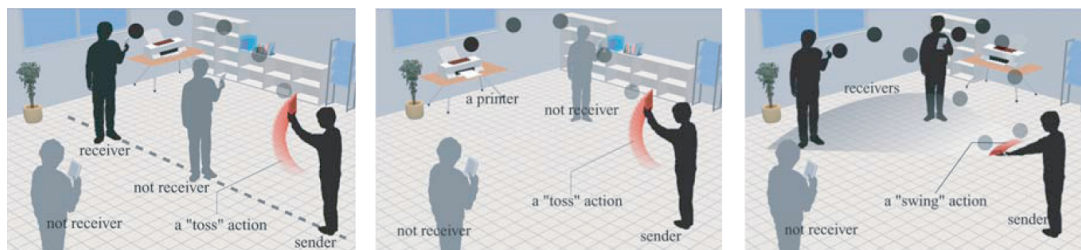


図 2.13 Toss-It

2.3.1 プライベート空間

プライベート空間は、特定の人だけがアクセス出来る空間の事である。特に本論文では、プライベート空間をユーザの管理下にある空間と定義する。例えば、自宅の部屋や研究室などがプライベート空間にあたる。プライベート空間の特徴として、**1) オブジェクトを把握している、2) 同じオブジェクトを何度も選択する事が挙げられる**。基本的にユーザの管理下にあるオブジェクトのみが存在するので、ユーザは空間内にあるオブジェクトの事は把握している為、第 2.1.3 項で挙げた問題はあまり発生しないと考えられる。また、ユーザの部屋にあるテレビなどの情報家電は、一度きりでなく何度も選択される。

2.3.2 パブリック空間

パブリック空間とは、誰でも自由にアクセス出来る空間の事である。特に本論文では、パブリック空間をユーザの管理下でない空間と定義する。例えば、デパート、イベント会場、駅構内、街中などがパブリック空間にあたる。パブリック空間の特徴として、**1) 自由に行動出来ない、2) 未知のオブジェクトがある事が挙げられる**。

パブリック空間の特徴として最も選択手法に影響を与えるのが、自由に行動出来ない点である。例えば、立ち入り禁止の場所や車道を挟んだ反対側の歩道などでは、オブジェクト選択の為にユーザが近づけない事がある。さらに混雑した場所では人の流れがあり自由に立ち止まったりする事も困難である可能性がある。

また、プライベート空間と異なりパブリック空間はユーザにとって未知の場所であることが多い。Lancaster University の Gellersen らは、このような未知の環境におけるオブジェクトへの利用を”Spontaneous Interaction”と呼んでいる [18]。パブリック空間のオブジェクトは、プライベート空間と違いユーザの管理下でないためユーザがオブジェクトの事を把握していない場合がある。パブリック空間における未知のオブジェクト選択は**サービス要求発火型選択**と**オブジェクト発見発火型選択**の 2 種類に分類出来る。以下で 2 つのオブジェクト選択について述べる。

サービス要求発火型選択

サービス要求発火型選択は、ユーザが何らかのサービスを利用したいという要求から、実世界上でサービスを利用可能なオブジェクトを探すオブジェクト選択である。未知の空間で”モバイル端末内のワードファイルを印刷したい”という要求から、ユーザの周辺にあるプリンターを探すといった例が挙げられる。サービス要求発火型選択は、オブジェクト選択手法にユーザの周辺にあるオブジェクトを検索するなどのオブジェクト発見の機能が必要となる。

オブジェクト発見発火型選択

オブジェクト発見発火型選択は、ユーザが実世界上でオブジェクトを発見することに起因するオブジェクト選択である。ユーザが実世界上の広告を見たことから、広告の詳細情報を知ることの出来る Web サイトに QR コードを使ってアクセスするといった例が挙げられる。本論文では、オブジェクト発見発火型選択に焦点を当ててオブジェクト選択手法の議論を進める。

2.4 機能要件

本節では、まず本論文が対象とする利用状況として1つのシナリオを提示する。そして、そのシナリオから対象とする利用状況におけるオブジェクト選択手法の機能要件を導き出す。

シナリオ：イベント会場

大学に勤務している A 君は、大学の研究発表イベントの企画を行っている。このイベントは大学の各研究室が複数のブースを作り、その年の研究成果を発表を行う。会場には **100 近くのブースが設置される予定**であり、各ブースには研究紹介用のポスターが設置される。A 君は、イベントの目玉として各研究ブースの紹介ページへのアクセスと気に入ったブースへの投票システムを任された。このイベントはとても好評で多くの人が訪れるため会場は混雑が予想される。去年はポスターに QR コードを貼ったがブースの前に人だかりが出来てしまい、**わざわざブースに近づくのは面倒**だと不評だった。また、特設ホームページから投票可能にする案もあったが、投票している事が周囲の参加者に視覚的に分かる事で**投票の活性化を計りたい**との要望があった。

環境側へのデバイス非設置

パブリック空間におけるオブジェクト選択手法は、環境側にデバイスを設置する必要のない事が望ましい。シナリオの例では会場にたくさんのブースが存在する為、各ブースごとにデバイスを配置する事は困難である。また、高価なデバイスを設置する事で盗難の危険性も発生する。シナリオの例以外でも、駅構内や町中などパブリック空間では多くのオブジェクトが存在する可能性がある。そのため、このような環境で利用するオブジェクト選択手法は環境側へのデバイス設置を必要としないべきである。



図 2.14 イベント会場の様子

距離的制約の解消

パブリック空間におけるオブジェクト選択手法は、ユーザとオブジェクト間に距離的制約を設けない必要がある。なぜなら、パブリック空間ではユーザがオブジェクトに近づけない状況が多々発生するからである。シナリオの例ではブースに人だかりが出来てしまい近づく事が困難になっている。また、簡単に近づける状況であっても近づく事がユーザにとって労力になってしまう。その為、ユーザがオブジェクトを発見したら、わざわざ移動する事なくその場で選択操作が出来る事が望ましい。

インタラクションの誘発

パブリック空間で利用するオブジェクト選択手法の要件の1つとして、周囲のユーザへの選択操作の誘発が挙げられる。パブリック空間におけるオブジェクト管理者はそもそもユーザに利用してもらうために設置している。シナリオの例でもイベントの主催者は多くの人に利用してもらいたいと考えている。ユーザがオブジェクトを携帯端末の画面内で完結してしまうと、周囲のユーザに選択している事が分からずイベントが盛り上がらない。選択するオブジェクトの種類や状況によっては、周囲のユーザに選択操作が分からない事が望ましい事もあるが、シナリオで挙げたイベント会場などでは選択操作を行っている事が周囲の人間に視覚的に分かる事が望ましい。

2.5 既存手法の比較

本節では、第 2.4 節で挙げた機能要件を第 2.2 節で紹介した既存の選択手法が満たしているのか分析する。

まず、2 次元コードを撮影することによってオブジェクトの選択を行う QR コード、Fractal Codes、u-Photo はオブジェクトとユーザの間に距離的に制約が発生してしまっている。Fractal

Code は、階層構造を用いることによってコードを認識可能な範囲を広げてはいるが、距離的制約の根本的な解決には至っていない。

携帯カメラ検索は、オブジェクト自体を画像としてデータベースに登録することで、オブジェクト全体をカメラで撮影して認識する。これにより、ある程度離れた場所からでも選択が可能になったが、逆にオブジェクトの近くにいと全体を撮影出来ないため離れる必要が出てくる。また、オブジェクトの正面以外では認識し辛いこと、オブジェクトの前に人がいたら利用出来なくなるなどの問題点がある。

Gaze-Link も 2 次元コードを利用しているため、距離的制約の機能要件を満たさない。Look-Point は、オブジェクト側に設置したカメラを用いて、ユーザの視線を検出しているため環境側へデバイスを設置する必要がある、オブジェクトから離れた場所で利用するは出来ない。CiteULike は、ユーザ側に 2 台のカメラを設置して、1 台でユーザの視線を検出し、もう 1 台でユーザの視線の先のオブジェクトを映す。映されたオブジェクトの中からユーザの視線の先にあるオブジェクトを計算し特定をする。そのため、環境側にデバイスを設置する必要はなく、またオブジェクト全体を認識しているため、距離的制約もある程度軽減する。また見るというインタラクションは周囲の人に対して視覚的に認識し辛いこと、全体的にインタラクションの誘発効果は少ない。

objSampler, Touch&Connect and Touch&Select はオブジェクトに直接接触するというインタラクションを用いているため距離的には大きな制約が掛かってしまう。また、オブジェクト側には RFID タグを貼る必要があるが、RFID タグは安価で大量に生産出来るため、大量のオブジェクトが存在しても簡単に対応出来る。

Select-and-Point, Pick-and-Beam は指し示すというインタラクションを用いた選択手法である。そのため、距離的にはあまり制約を受けないが、両手法とも環境側にカメラを設置する必要がある。

Proximal interactions は、近づくというインタラクションを用いている。近づいている状態を取得するために、ネットワークの電波強度を利用しているためオブジェクト側にデバイスが必要である。音のバーコードには、オブジェクト側にスピーカを設置する必要がある。両手法ともオブジェクトに近づく必要があり、近づくだけで選択が出来てしまうため周囲へのインタラクションの誘発効果は少ない。

Toss-It は身体性を用いたインタラクションである。環境側にカメラを設置する必要があるが、トスの強弱で対象となるオブジェクトの距離を調節出来る。また、身体性を用いるため周囲へのインタラクション誘発効果は高い。

以上のように、本論文で提示したパブリック空間において使いやすい機能要件を全て満たす既存の手法はない。実世界のインタラクションを使っている研究が多いので、インタラクションの誘発に関しては多くの研究が要件を満たしていると言える。しかし、選択操作時にカメラや RFID タグなどを用いるため、使用デバイスによる距離的な制約が発生してしまい距離的制約の解消という機能要件を満たしている研究は少なかった。

表 2.1 既存の実世界オブジェクト選択手法の比較

既存の選択手法	デバイスの非設置	距離的制約	インタラクション
QR コード	○	×	○
Fractal Codes	○	×	○
u-Photo	○	×	○
携帯カメラ検索	○	△	○
Gaze-Link	×	×	△
LoolPoint	×	×	△
CiteULike	○	△	△
objsamplers	○	×	○
Touch&Connect and Touch&Select	○	×	○
Select-and-Point	×	○	○
Pick-and-Beam	×	○	○
Proximal interactions	×	×	○
音のバーコード	△	×	×
Toss-It	×	○	○

2.6 本章のまとめ

本章ではまず、実世界オブジェクトの定義をした後、実世界オブジェクト選択について整理した。次いで、実世界オブジェクト選択における問題意識を示し、その問題を解決する既存の実世界オブジェクト選択手法を紹介した。さらに、実世界オブジェクト選択の機能要件をシナリオから導き出し、既存の手法を比較することで本研究の問題意識及び、研究意義を示した。

第 3 章

振る動作を用いた選択手法

本章では，実空間オブジェクト選択手法として振る動作を用いた手法を提案する．はじめに，提案する振る動作による選択手法の概要を説明する．ついで，振る動作によるオブジェクトの選択に用いる SR コードの設計をする．さらに，振る動作の解析手法について述べる．最後に設計した SR コードの検証を行うことで本論文における問題領域を導き出す．

3.1 振る動作を用いた選択手法の概要

本論文では、実空間オブジェクト選択手法として振る動作を用いた手法を提案する。図 3.1 に振る動作を用いた選択手法の概念図を示す。本手法では、ユーザがモバイル端末を持っている事を前提とし、各実空間オブジェクトにモバイル端末をどのように振るか示したコードを貼付する。また、各実空間オブジェクトとコードの対応はサーバ上で管理する。

オブジェクトを選択する場合、まずユーザはオブジェクトに貼られたコードを見る必要がある。そしてどのように振ればいいのか認識した後、自身のモバイル端末でコードを再現するように振る。すると、振られた動作に対応したオブジェクトがサーバ上でマッチングされ、オブジェクトのネットワーク ID がユーザのモバイル端末に送信される。

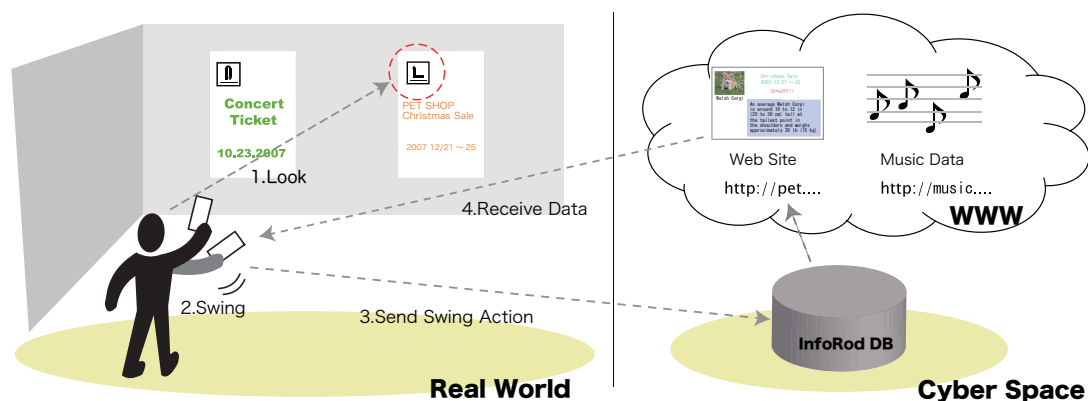


図 3.1 振る動作を用いた選択手法の概念図

前章では、混雑した駅構内やイベント会場などのパブリック空間で利用しやすい実世界オブジェクト選択手法がないと述べた。以下で、振る動作を用いた選択手法が公共空間における実空間オブジェクト選択手法の機能要件を満たしている事を説明する。

環境側へのデバイス非設置

モバイル端末を振るという動作は、環境側にデバイスを設置しなくても推定する事が出来る。例えば、モバイル端末に内蔵された加速度センサのデータや、カメラの画像を解析する事によって端末の動きを求める事が出来る。モバイル端末に内蔵されたデバイスのデータを解析に用いる事で、振る動作の推測が端末内で完結するため環境側に新たなデバイスを設置する必要がなくなる。

距離的制約の解消

振る動作を用いた選択手法では、ユーザとオブジェクト間の距離がユーザの視力に依存している。しかし、実世界のオブジェクトを認識し、選択を行おうとする行為が既に利用者の視力に依存している事になる。本手法ではなるべく遠くからでも認識しやすいコードを設計し、コードの視覚的な認識に関しての検証を行う事でオブジェクトは確認出来ているが、コードが見えないという状況をなくす。

インタラクションの誘発

モバイル端末を振るという動作を選択手法として用いる事で、現在行っている選択操作を周囲のユーザに視覚的に伝達する事が可能である。他人のオブジェクト選択を見ることがオブジェクトの存在に気づいたり、多くの人が選択しているオブジェクトに対して何かの関心を抱く可能性が出てくる。それらが周囲の人の選択操作に繋がり空間内の選択が活性化する事が考えられる。逆にモバイル端末のボタンや画面だけで選択操作が完結してしまうと、周囲に選択をしている事が分からず選択操作の誘発は起こらない。

3.2 振る動作の設計

本節では、実世界オブジェクトを選択する際に用いる振る動作を設計する。本手法では、用途の異なる2種類の振る動作を設計する必要がある。1つ目は、実世界オブジェクトを特定する為の振る動作である。この振る動作は、実世界オブジェクトに貼付するコードに示す。このコードを本論文ではSwing Reference Code(以下、SRコード)と呼ぶ。2つ目は、オブジェクト選択時の操作を定義した振る動作である。この振る動作は、どのオブジェクトを選択した時でも利用可能であらかじめ定義されているものである。本論文では、この振る動作を操作ジェスチャと呼ぶ。本節では、SRコードと操作ジェスチャの機能要件の整理、設計を行う。

3.2.1 SRコードの機能要件

SRコードの要件として、実際に端末を振るユーザ側の視点からの要件と、SRコードを生成、管理する管理者側の視点からの要件がある。まず、ユーザ側の要件として1) **記載した動作の認識が容易であること**、2) **振る動作での再現が容易であること**が挙げられる。次に、管理者側の要件として、1) **多くのパターンを生成可能であること**、2) **コードの管理が容易であること**以下で、SRコードの設計におけるユーザ側の要件と管理者側の要件をそれぞれ述べる。

認識の容易性

SRコードはオブジェクトに貼付した際、容易に認識出来るものでなければならない。具体的には、1) **遠距離からでも認識可能であること**、2) **瞬時に理解可能であること**、3) **見間違いが少ないこと**である必要がある。遠距離からでも認識可能にすることで、第3.1項で述

べたようにオブジェクトは見えるがコードが見えないという状況をなくす。また SR コードは振る動作の内容を示すコードであるため、ユーザが深く考え込むことなく SR コードに示された振る動作を理解出来なければならない。瞬時に理解出来ることで、オブジェクトの前に人混みが出来ていたとしても一瞬だけコードが確認出来れば選択することが出来る。最後に利用者にとって見間違えることが少ないコードを設計することで、確実に利用者の意図したオブジェクトの選択を可能にする。

再現の容易性

本手法で用いる振る動作は、利用者にとって容易に再現可能である必要がある。振る動作の再現が簡単であるほどユーザは気軽にオブジェクトの選択を行い、困難であるほどユーザはオブジェクトの選択を躊躇してしまうからである。また、多くのパターンが生成可能でも振る動作に多大な時間がかかるなど、利用者に負担のかかる振る動作も望ましくない。例えば端末を上下に何回かシェイクするという動作を用いると対象となるオブジェクトの数が増えるほど利用者への負担が増えるので現実的ではない。

多様なパターンの生成

本手法で用いる振る動作は、多くのパターンのコードを生成可能な必要がある。なぜなら対象となるオブジェクトと関連付けるため、生成可能なコードのパターン数が本手法で利用出来るオブジェクトの数に相当するからである。本手法で用いる SR コードは覚えるものではなく、オブジェクトに貼られた SR コードをユーザが見て、その場で再現することを想定している。そのため、生成可能な SR コードのパターンは多いほど良い。また、SR コードとして三角や四角などの図形を採用すると、数十パターンを図形を生成するのは困難であり、また生成数に比例してコードが複雑になってしまう。そのため、SR コードは単純に多くのパターンが生成出来るだけでなく、簡単にコードを生成可能である事と、コードの複雑さが統一的な必要がある。

コード管理の容易性

SR コードの要件のひとつにコードの管理が容易である事が挙げられる。SR コードのデータはサーバ上で管理を行うため、簡単にデータベースで保存出来るものが望ましい。さらに、**コードの生成可能数が計算出来る事やユーザへの負担が計算可能な事**も重要である。例えば、ある空間のオブジェクトにコードを貼付ける場合、どの程度のコードの複雑さなら、いくつのコードが生成可能なのか計算出来る必要がある。また、よく選択されるオブジェクトには振りやすいコードを割り当てるなどの要望に答えるために、コードの振りやすさを計算する事も必要である。

3.2.2 SR コードの基本設計

本論文では、第 3.2.1 項で述べた機能要件を満たすコードとして上下左右 4 方向の線を複数組み合わせたコードを提案する。図 3.2 に本論文で設計した SR コードの例を示す。

SR コードは、コードが示す振る動作に対応した縦と横の線を繋ぎ合わせる事で形成される。振る動作が上から右や左から下など縦と横の線を繋げる場合は直角になる様に繋ぎ合わせる。SR コードは連続して同じ方向を繋げないように設計している。なぜなら、同じ方向を連続で振るという動作はユーザが 2 つの振る動作の区別をつけるのが困難であり、コードに表記する上でも認識し辛くなってしまうからである。

図 3.2 の SR コードは上、右、下、右の順に振るという動作を表す。空間内に存在するオブジェクトの数に応じて、組み合わせる数を変化させる事が可能である。図 3.2 の SR コードは 4 つの線の組み合わせだが、空間内に存在するオブジェクトが少ない場合はより簡単なコードで利用可能である。

SR コードの表記に関しては、上から下や右から左など同じ方向の線を繋げる場合は、図 3.3 の様にカーブを描くようにして 2 つの線を繋げる。最後に振る動作に対応した線を矢印として描くことで、最初に振る動作は SR コードにある線の中の矢印のついていない端だと簡単に理解出来る。SR コードをより強調するし、周囲との区別させるために太い枠で矢印を囲う。また、天井に吊るされたプロジェクターに SR コードを貼付けた場合など天地の認識が出来なくなる可能性があるため、SR コードの下にあたる部分に横線を一本描くこととする。

また、以下で設計した SR コードが第 3.2.1 項で述べた機能要件を満たすことを示す。

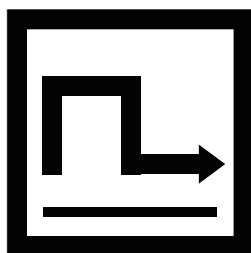


図 3.2 SR コードの例

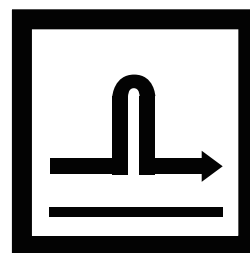


図 3.3 上下に折り返す SR コードの例

認識の容易性

SR コードは振る動作に対応した線を繋げて形成される。振る動作を 1 つずつ分けて表すのではなく、全ての振る動作を繋げて表すことで振る動作を形として認識することが出来る。そして、一筆書きの要領でコードを表記するため、簡単に振り始めを認識することが可能である。また、SR コードによっても組み合わせる数や順番によってさまざまなパターンが存在する。どのようなパターンだと、どのくらいの距離・時間で視認可能なのか調査する

ことが重要である。

再現の容易性

ユーザは SR コードをなぞるように端末を振ることで簡単に再現可能である。特に連続で同じ方向の振る動作が続くことがないので一連の振る動作を流れるように行える。また、あらかじめ設定した振る動作の組み合わせ数によって、利用者への負担がある程度保証される。振る動作を 4 つ組み合わせた動作を用いるなら、利用者は 4 回振るだけで必要な操作が完了する。しかし振る回数の違いだけでなく、パターンの違いによって振りやすさは異なるのか、異なるならどんなパターンのコードが振りやすいのか知ることが重要である。

多様なパターンの生成

上下左右 4 方向の振る動作を組み合わせた SR コードは、簡単な振る動作で多くのパターン数を生成出来る。表 3.1 に SR コードの生成可能なパターン数を示す。例えば、上下左右の振りを 4 つ組み合わせたコードは 108 通り ($4 * 3 * 3 * 3 = 108$) 生成可能である。この計算式で 2 つ目以降に 3 を掛けているのは、同じ方向の動作を連続で組み合わせない設計にしているためである。そのため、1~3 つの組み合わせも含めると 160 通りのパターンが出来ることになる。パターン数が不足したら振る動作を増やす事で対応出来るが振る回数を増やしてしまうとそれだけユーザに負担が掛かってしまう。そのため、広いイベント会場などで多くのオブジェクトを選択対象とする場合、位置情報と振る動作を組み合わせる事で多くのオブジェクトに対応させる。

表 3.1 SR コードの生成可能数

組み合わせ数	同じ回数内のパターン数	全ての回数を合わせたパターン数
1	4	4
2	12	16
3	36	52
4	108	160
5	324	484

コード管理の容易性

SR コードは文字列で表現することが出来るため簡単にデータベースに保存可能である。3.2 の SR コードの場合だと”Up, Right, Down, Right”といった文字列で表現出来る。例えば、記号の組み合わせで数多くのパターンを生成させる場合、各記号ごとの振る動作を表したパターンファイルが必要となる。しかし、SR コードの場合は多くのパターンを生成してもそれぞれの振る動作は完全に独立している。そのため、4 方向を識別するパターンファイ

ルを用意するかユーザ端末内に 4 方向の振る動作を推測するソフトウェアを導入させるだけで全てのパターンに対応出来る。

また、前述したようにコードの生成可能数も簡単に計算出来る。そして、どんなパターンがユーザにとって振りやすいのか解明することが出来れば、コードごとのユーザへの負担を計算することが出来るため、多くの回数選択されるオブジェクトにより振りやすい SR コードを割り当てることも可能になる。

3.2.3 操作ジェスチャの要件

操作ジェスチャは、本手法を用いて実世界オブジェクトを選択する際に、振る動作のみで統一的に操作を可能にするために設計する。まず、本手法を用いた実世界オブジェクトの選択行為において、どのような操作を行う必要があるの述べ、次に操作ジェスチャはどんな設計であるべきなのか整理する。以下に振る動作の選択において必要な操作を挙げる。

入力のキャンセル

入力のキャンセルは SR コードを振る動作で再現する際に、利用者の意図するものと異なる方向がソフトウェアに認識された時や選択途中で異なるオブジェクトを選択しようとする時などに、現在認識されている振る動作の記録を消去する操作である。キャンセルには 2 種類あり、全ての動作記録を消す操作と 1 つだけ動作記録を消す操作がある。1 つだけ動作記録を消す操作は、ソフトウェアがユーザの意図と異なる方向を認識してしまった場合に用いる。

振る動作の決定

振る動作の決定は、ユーザが SR コードを再現し終わり、ソフトウェアに認識された方向が選択対象の SR コードと同じ事を確認した後にオブジェクトの選択を実行する操作である。

周辺オブジェクトの検索

周辺オブジェクトの検索は、2 章で述べたサービスの利用に起因するオブジェクト選択を実現したり、未知のオブジェクトを発見するための操作である。この操作を行うことで、ユーザの周辺に存在する選択可能なオブジェクトのリストが表示される。

オブジェクト選択後の操作

オブジェクト選択後の操作は、選択したオブジェクトごとに異なる。選択したオブジェクトがライトならオンとオフの操作、ミュージックプレイヤーなら再生、停止、早送り、巻き戻しなどの操作が必要になってくる。複雑な操作が必要な場合は、振る動作だけでなく

GUI による操作で対応することも考えられる。

操作ジェスチャは SR コードと異なり，ユーザが振る動作を記憶する必要がある．よってユーザの記憶を補助するためにも，操作ジェスチャが直感的に 1 つ 1 つの動作と結びつくようなものである必要がある．また，上下左右の方向に振る動作の最中に行う操作は，明確に区別がつくように上下左右に振る動作と全く異なるよう設計する．

3.2.4 操作ジェスチャの設計

本項ではキャンセル，振る動作の決定，周辺オブジェクトの検索，オブジェクト選択後の操作という 4 つの操作ジェスチャの設計について説明する．

入力のキャンセル

図 3.4 にキャンセル操作に対する操作ジェスチャの動作を示す．キャンセル操作には，左右に細かくシェイクするという動作を割り当てる．シェイクするという動作は，端末に認識された振る方向の情報を振り払うというイメージを連想させる．また，1 回の振りのキャンセルと，全ての振りのキャンセルはシェイク動作を続ける時間によって区別する．シェイクの動作を行うとまず 1 回振る動作がキャンセルされる．さらにシェイクし続けると全ての動作がキャンセルされる．

振る動作の決定

図 3.5 に振る動作を決定するという操作に対する操作ジェスチャの動作を示す．振る動作を決定した際に，手前に端末を引く，つり上げるというジェスチャを用いる．つり上げるというジェスチャは，SR コードからオブジェクトの情報を取得するというイメージを連想させる．SR コードを再現する際の上に振る動作に比較的似ている動作であるが，単純に上の方向に振る動作と違いユーザの顔の方に端末を傾けながら引くという動作であるため 2 つの動作を区別することは可能である．

周辺オブジェクトの検索

図 3.6 に周辺のオブジェクトを検索するという操作に対する操作ジェスチャの動作を示す．周辺オブジェクトを検索する操作には，円を描くというジェスチャーを割り当てる．空間全体やこの辺りという意味のジェスチャで，円を描く動作を行うことがある．既に定着している円を描くというジェスチャを周辺のオブジェクトを検索するという操作ジェスチャに割り当てることでユーザも記憶しやすく，操作を行うときも自然に動作が出てくる．

オブジェクト選択後の操作

オブジェクト選択後の操作に関しては，オブジェクトごとに必要な操作が異なるという理

由から現段階での設計は行わない。機器を操作するジェスチャの設計は今まで数多くされており、本論文では議論を避ける。しかし、再生する、停止するといった多くの機器で共通の操作はあらかじめ操作ジェスチャを設計しておく必要がある。また、早稲田大学の木村らはユーザはより簡単なジェスチャを求めるという傾向があり、基本的なジェスチャをあらかじめ提供しておき、ユーザの好みによって再設定可能なシステムが好ましいと述べている [19]。



図 3.4 入力のカンセル



図 3.5 振る動作の決定



図 3.6 周辺オブジェクトの検索

3.3 フィードバックの設計

本節では、振る動作を用いた実世界オブジェクト選択手法におけるフィードバックの設計を行う。まず、本手法におけるフィードバックの要件を整理し、その後フィードバックの設計を行う。

3.3.1 フィードバックの要件

端末を振る動作を行うことにより、端末の角度が斜めになったり目線から外れて、画面の中が見えにくくなることがある。よって本手法におけるフィードバックの要件として画面を見なくても振った方向を認識可能な事が挙げられる。また、次の振る動作にすぐ移るために素早く早く振った方向を認識出来る事や、1回の振りだけでなく振った方向全体の確認を行う必要がある。

3.3.2 音によるフィードバック

画面が見えない状況でのフィードバックを可能にするため、本手法ではフィードバックに音を用いる。音によるフィードバックでは、ユーザに振った方向を伝えるため方向によって異なる音を鳴らすというアプローチを採用する。ユーザに各方向の音を識別させるためには、直感的に判断出来る音を各方向に対応付ける必要がある。そのため、本論文では上方向に振られた時には高い音を下方向に振られた時には低い音を鳴らす。左右の方向に関しては、直感的に判断出来る例がないため出来るだけ異なる音を流す。上下方向は直感的に判断出来るため、ユーザは実質左右 2 種類の音を識別出来るようになるだけで音によるフィードバックを得ることが可能になる。

しかし、音によるフィードバックは万能でない。本論文で想定するパブリック空間では周囲の環

境音が大きい事も想定されるため、音によるフィードバックを聞き逃してしまう可能性がある。また、SR コードを全て再現し終わった後に 1 振りごとのフィードバックを新たに得る事は難しい。これらの問題点を補うために、次項では GUI によるフィードバックを説明する。音が聞こえない状況においては GUI によるフィードバックで振った方向の確認を行う。

3.3.3 GUI によるフィードバック

音によるフィードバックを聞き逃した時や、振る動作全体の確認を行う時のために GUI による視覚的なフィードバックを行う。視覚的なフィードバックでは、ユーザの振った動作が SR コードに沿っているか即座に理解出来る必要がある。そのため、振る動作を 1 つずつ表示するのではなく、SR コードのように各振る動作を繋げて表示を行う。図 3.7 に GUI によるフィードバックの例を示す。図 3.7 は、3.2 の SR コードを再現した際の GUI のフィードバックである。最後の振る動作が終わったら GUI のフィードバックとして表示されたコードと実際の SR コードが同じものかどうか確認するだけで正確に振る動作が行えたかすぐ理解出来る。

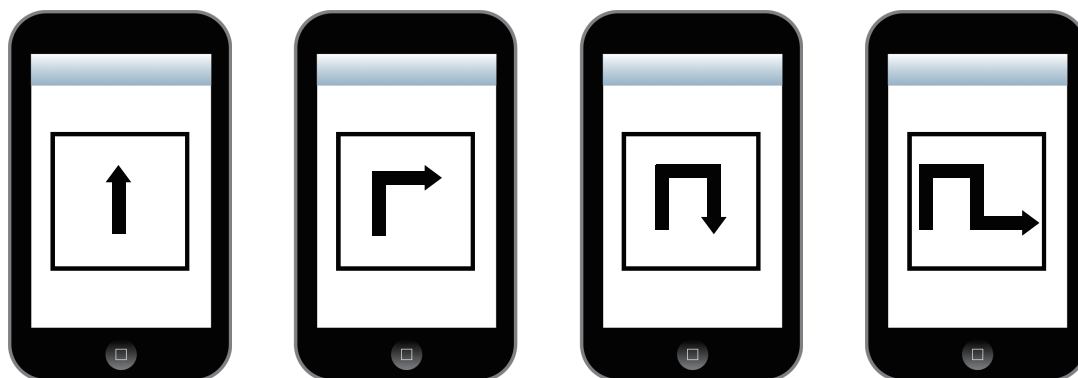


図 3.7 GUI によるフィードバックの例

3.4 振る動作の解析

本節では、本論文における端末を振る動作の解析について述べる。はじめに振る動作の解析に加速度センサを用いた理由を説明する。ついで SR コードを再現した時の振る動作の解析、操作ジェスチャを行った時の振る動作の解析について説明する。

3.4.1 加速度センサを用いた解析

本論文では、振る動作の解析に、3 軸方向の加速度を取得可能な加速度センサを利用する。加速度センサを用いた手法以外には、端末に内蔵されたカメラの映像を解析することで端末の動きを求める手法がある [20] [21] [22]。しかし、映像の背景を静止させる必要があるため、混雑した人混

みの中での利用や歩きながらの利用には不向きである。よって本手法では、環境側にデバイスを設置する必要がなく、利用者が振る動作を行う時に周囲の環境を全く気にすることのない加速度センサのデータを解析する手法を用いる。

本論文では、加速度センサの内蔵されたモバイル端末として apple 社の iPhone [23] を用いる。図 3.8 に本論文で用いる iPhone を示す。加速度センサから加速度の変化を調べることで、端末の動きの変化を知ることが出来る。また、地球の重力 (地面方向への $1g = \text{約 } 9.8m/s^2$) が端末にかかる向きを調べることで、地面に対する端末の傾きを知ることが出来る。図 3.9 に、iPhone に対する加速度 X, Y, Z, 3 方向の直交座標系を示す。図 3.9 を見れば分かるように、X, Y, Z の 3 軸にはそれぞれプラス方向と、マイナス方向がある。また、加速度センサにおける X, Y, Z の 3 軸は端末を基準としたものに常に固定されているため、端末を Y 軸方向に反転させて持つと、利用者から見た X 軸のプラス、マイナス方向が逆転する。



図 3.8 iPhone

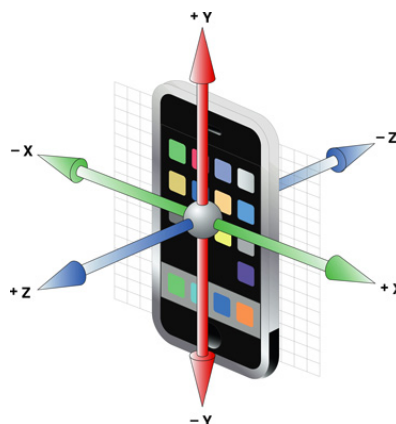


図 3.9 iPhone における加速度の座標系

3.4.2 4 方向に振る動作の推測

SR コードを再現するためにユーザは上下左右の 1 方向に振るという動作を複数回行う。そのため、モバイル端末側でユーザが上下左右どの方向を振ったのか推測する必要がある。以下では上下左右に振った時の加速度データの特徴を説明し、どのような方法で方向を推測するのか述べる。

図 3.10~3.13 は、それぞれ上下左右の 1 方向に端末を振ったときの加速度 x, y, z の変化を示したものである。4 つのグラフを見ると全ての振りで加速度 y の数値が一度増加するという特徴が見られるが、全ての振りで共通しているので方向の識別には使えない。上下の振りでは加速度 z 、左右の振りでは加速度 x において各振り特有の特徴が表れている。

上方向の振りでは、加速度 z の値が一度減少してから一気に増加し、また元の値に戻る。下方向の振りでは、加速度 z の値が一度増加してから一気にマイナス値まで減少し、また元の値に戻る。

上下方向における加速度 z の変化が、左右方向では加速度 x で起きる。右方向の振りでは、加速度 x の値が一度減少してから一気に増加し、また元の値に戻る。左方向の振りでは、加速度 x の値が一度増加してから一気にマイナス値まで減少し、また元の値に戻る。各方向における加速度の変化の特徴を表 3.2 にまとめる。

本論文ではユーザの振った方向を推測するために加速度の変化の凹凸を抽出する。4 方向に一度振ると加速度 x, z のどちらかが凹凸に変化を起こす。抽出した凹凸と各方向の変化をマッチングさせることでユーザの振った方向が推測出来る。

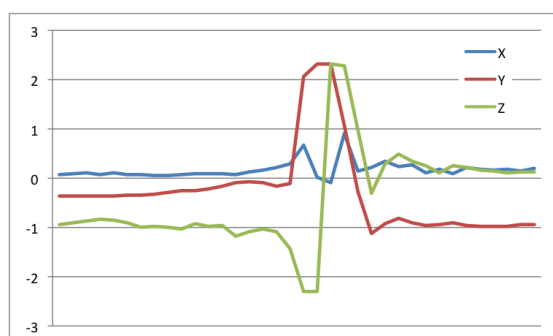


図 3.10 上方向に振った時の加速度の変化

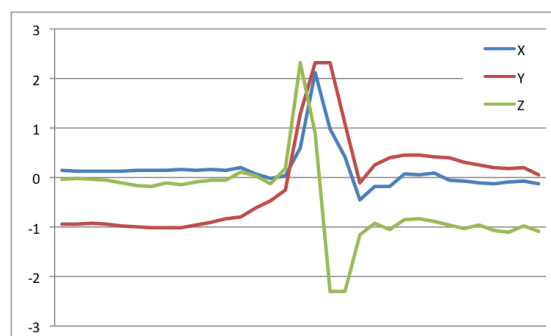


図 3.11 下方向に振った時の加速度の変化

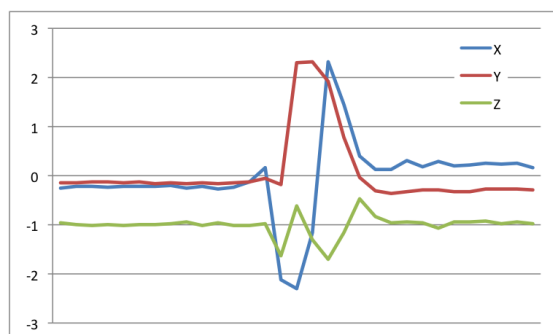


図 3.12 右方向に振った時の加速度の変化

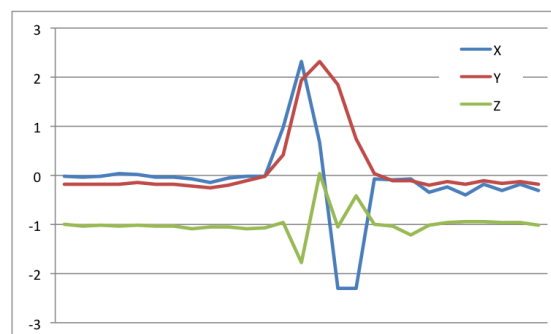


図 3.13 左方向に振った時の加速度の変化

表 3.2 加速度の変化の特徴

振る方向	加速度 X	加速度 Y	加速度 Z
上方向	-	-	マイナス⇒プラス
下方向	-	-	プラス⇒マイナス
右方向	マイナス⇒プラス	-	-
左方向	プラス⇒マイナス	-	-

3.4.3 操作ジェスチャの認識

上下左右 4 方向に振る動作は，第 3.4.2 節で述べたように，ルールベースで定義が可能である．しかし，操作ジェスチャはルールベースで定義するのが困難である．そのため，本論文では操作ジェスチャをニューラルネットワークアルゴリズムを用いて推測する．

図 3.14 にニューラルネットワークの概念モデルを示す．ニューラルネットワークでは複数のニューロンが入力層，中間層，出力層の 3 層にわたって構成しており，それぞれの層を構成するニューロン同士が結合している．各層のニューロンの数は，入力層は加速度データ配列の数，出力層は推測したいジェスチャの数，中間層は加速度データ配列数の 1.5 倍とした．

各ニューロンは，発火のしやすさとニューロン同士の結合の強さを示す 2 つの値を持っている．入力された値に対して発火が起きると，結合の強さと入力値を掛け合わせた値を次のニューロンに渡す．ジェスチャを推測する場合は，まず入力層の各ニューロンに加速度データを配置する．そして，中間層，出力層と値を伝播て行くことで出力層の各ニューロンの値を求められる．出力層の中で値が最大となったニューロンの発火状態があらかじめ決めた閾値以上の場合，ユーザに振られたジェスチャであると推測することが出来る．

また，ニューロンの発火のしやすさとニューロン同士の結合の強さを示す値は，学習フェーズの初期状態ではランダムで決定され，エラー率が一定以下になるまで最適化を繰り返す．その手順を以下に示す．

1. 各ジェスチャの教師データを入力層に配置する
2. 出力層まで伝播させる
3. エラー率 (出力値と理想値との差分) を計算する
4. エラー率が一定以上ならエラー率に応じたニューロンの値を調整
5. エラー率が一定以下になるまで 2～4 を繰り返し替える
6. 1 に戻る

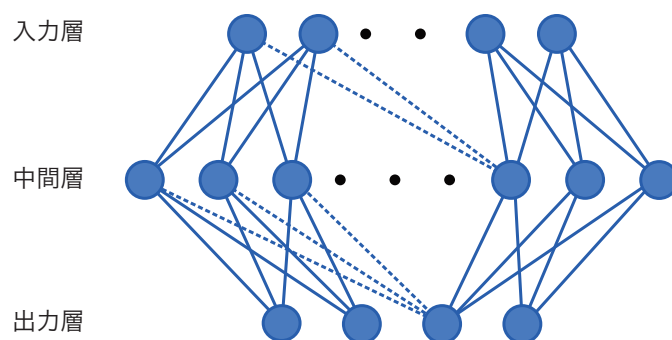


図 3.14 ニューラルネットワークの概念モデル

また、上記のアルゴリズムで推測する場合、操作ジェスチャが振られている最中の加速度データを取得し、入力層のニューロンに配置する必要がある。ユーザが操作ジェスチャが振られている最中はボタンを押し続けるという操作をすることで、最適な加速度データが取得可能になるが、ユーザへの負担も大きい。そのため、本論文では加速度データの分散値を算出し、操作ジェスチャの振り始めと振り終わりを検知することで推測に最適な加速度データ配列を取得する。

3.5 SR コードの検証

本節では、まず振る動作の解析アルゴリズムの認識率を検証を行い、ついで、8方向のSRコードの実現可能性について検証する。

3.5.1 振る動作推測アルゴリズムの検証

第3.4.2項では、本論文における4方向に振る動作の推測手法について述べた。本論文で提案した振る動作を用いた選択手法は、モバイル端末を振るという動作によってオブジェクトを選択している。ユーザの負担を極力減らすためにも、高い確率でユーザの意図した通りの方向を推測する必要がある。そのため本項では、第3.4.2項で述べた振る動作推測アルゴリズムの認識率を検証する。

実験概要

本検証実験の被験者は20代の男性7人である。被験者は全員右利きであった。apple社のiPhoneに第3.4.2項で述べた動作推測手法を実装したソフトウェアをインストールした。各被験者の前で上下左右それぞれの振る動作を2回ずつ行った。さらに3分間被験者に振る動作の練習を行わせて、どのように振ると正しく推測するのか学習させた。その後、上下左右の振る動作をランダムに行わせた。各方向が50回ずつになるまで繰り返した。最後に、本検証実験に対してのインタビューを行った。

実験結果

実験結果を表3.3に示す。誤認識率は被験者が意図した振る動作とは異なる動作を認識してしまった確率を表す。未検知率は被験者が振る動作を行ったにも関わらず振る動作推測ソフトウェアが振る動作と認識しなかった確率である。認識率は誤認識と未検知がなく正常に振る動作の推測が行われた確率を示す。

被験者へのインタビューでは、上方向に振る際iPhoneの重みのせいで他の方向と比べて振りにくいという意見が多かった。また逆に下方向への振りは、重力に従ってiPhoneを振り落とすだけで良いため他の動作より楽だという意見が見られた。

考察

実験結果で示した通り、振る動作の推測ソフトウェアが正しく振る動作を認識した確率

表 3.3 振る動作の認識率

方向	誤認識率	未検知率	認識率
上方向	1.50%	9.75%	88.75%
下方向	2.00%	6.00%	92.00%
右方向	0.25%	9.00%	90.75%
左方向	1.25%	8.00%	90.75%
平均	1.25%	8.19%	90.56%

は 90.56% となった。特に注目すべきことは、間違っ振る方向を認識してしまう確率が 1.25% と低かった事である。間違っ方向を認識してしまうと、間違っ認識された入力を消去する必要がある。間違っ推測を消去する操作の上に、もう一度正しい振る動作を行う必要があるため、ユーザへの負担は非常に大きい。逆に振った方向がソフトウェアに検出されないことは、間違っ推測されてしまった場合よりも負担は小さい。なぜなら同じ動作をもう一度繰り返すだけで良いからである。

また、各方向によって認識率が若干だが異なっている。上方向の認識率が 88.75% と他の方向よりも低くなっている。逆に下方向の認識率は 92% と他の 3 方向よりも高い。さらにユーザへのインタビューから認識率として出た数値以上に、上方向へ振る動作は振りにくく、下方向への振りは振りやすいことが分かった。左右の方向に関しては、ユーザへのインタビューでは好みが分かれたが認識率としては近い値となった。

3.5.2 8 方向 SR コードの検証

第 3.2.2 節で上下左右 4 方向の SR コードにおけるパターン生成可能数について述べた。本手法では、SR コードで生成出来るパターンの数が扱えるオブジェクトの数に相当するため、定義した SR コードによって生成可能なパターン数は多いほど良い。表 3.4 に 4 方向と 8 方向の SR コードでのパターン生成可能数の比較を示す。この表から分かる通り、8 方向の SR コードは 4 方向に比べ圧倒的に生成出来るパターン数が多い。そこで、上下左右 4 方向に斜め 4 方向を合わせた 8 方向の SR コードの実現可能性について述べる。

コードの視認性

8 方向の SR コードにおいて最も懸念すべき事項として、コードの視認が難しくなってしまうことが挙げられる。8 方向の SR コードの視認において考えられる問題は 2 つある。

1 つ目は、斜めのコードの認識のしにくさである。上下左右 4 方向の場合は、線と線の繋がりは 90 度か折り返しのみであったが、8 方向になると、45 度、90 度、135 度と折り返しになってしまう。さらに、45 度で曲がったところからさらに 45 度、90 度、135 度、折り返す可能性があるため、1 つの振る動作をどの方向に振ればいいのかユーザが考える負担が増

表 3.4 パターン生成可能数の比較

組み合わせ数	4 方向	8 方向
1	4	8
2	12	56
3	36	392
4	108	2744
5	324	19208

えてしまう．さらに，4 方向の SR コードよりも遠くの場所から視認出来にくくなることが考えられる．図 3.15 に 8 方向の SR コードの例を示す．

2 つ目は，SR コードとしての表記の問題がある．斜めの線は縦横一方向に長く伸びるため，4 つの振る動作を組み合わせると縦長，横長になってしまう．縦長，横長なコードになってしまうと，正方形の枠の中にコード全体を入れる必要があるため 4 方向の SR コードに比べると，非常に小さくなってしまう．図 3.16 に縦に伸びた 8 方向の SR コードを示す．

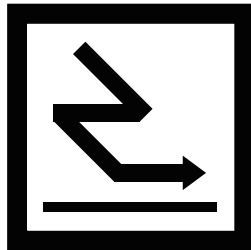


図 3.15 8 方向の SR コードの例

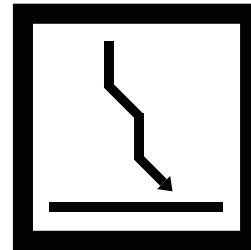


図 3.16 縦に伸びた 8 方向の SR コードの例

結論

前述した理由から 8 方向の SR コードはユーザが視認しにくい．そのため，本論文では 8 方向の SR コードは利用せずに，上下左右 4 方向の SR コードを用いる．本手法を利用する空間においてオブジェクトの数が多すぎる場合は位置情報と組み合わせることでオブジェクトを一意に識別する．

3.6 考慮すべき問題

第 3.5.1 項で行った検証実験の結果，振る動作の方向によって振りやすさが異なることが分かった．SR コードは 4 方向の振る動作を複数組み合わせるコードであるため，この影響が大きくなり振る回数が同じコードでも振りやすさが大きく異なることが考えられる．

パブリック空間には多く利用されるオブジェクトとあまり利用されないオブジェクトが存在す

る。多く利用されるオブジェクトにはより使いやすい SR コードを割り当てるべきである。よって、本論文で提案する振る動作を用いたオブジェクト選択手法を実現するためには、1 つ 1 つの SR コードの振りやすさを正確に把握する必要がある。

そのため、各 SR コードの振りやすさを明確に知ることがとても重要になる。また、振る動作を用いた選択手法にとっての SR コードの使いやすさとは、振りやすさだけではなく認識のしやすさも重要な要素である。振りやすさだけでなく、認識のしやすさも含めた総合的な SR コードの使いやすさを把握する必要がある。

3.7 本章のまとめ

本章では、本論文で提案するモバイル端末を振る動作を用いたオブジェクト選択手法の概要について述べた。次いで、本手法で利用する振る動作をオブジェクトの選択と選択時の操作の 2 種類に分類し、それぞれ機能要件を整理した後に設計を行った。さらに、それぞれの振る動作に対する解析手法について説明を行った。最後に、オブジェクト選択に用いる SR コードの有用性についての検証を行い、本論文で考えるべき問題を導き出した。

第 4 章

SR コードの使いやすさの分析

本章では，オブジェクト選択をする際に用いる SR コードの使いやすさについて分析を行う．SR コードを再現する際の振りやすさに着目し，評価実験を行う．そして，SR コードの振りやすさを数値化し，比較可能にする．

4.1 SR コードの使いやすさ

本節では、SR コードの使いやすさについて整理する。まず、振る動作を用いた選択行為をモデル化し、選択行為全体における各フェーズの影響力を説明する。その後、SR コードの振りやすさについて述べる。

4.1.1 振る動作を用いた選択行為の行動モデル

SR コードの使いやすさを分析するためには、まず振る動作を用いたオブジェクト選択行為をモデル化し、それぞれのフェーズごとに使いやすさを考察する必要がある。振る動作を用いたオブジェクト選択行為をモデル化すると視認フェーズ、スイングフェーズ、確認フェーズに分けることが出来る。視認フェーズでは、SR コードを目視しどの方向を振るべきか認識する。スイングフェーズでは、視認フェーズで認識した方向にモバイル端末を振る。確認フェーズでは、振る動作に対するフィードバックから振った方向が正しくシステムに認識されたか確認する。また、確認フェーズで正しく認識されていないことが分かったらキャンセル操作を行い、全ての振る動作を認識させたら決定操作でオブジェクトの選択を行う。

まず、視認フェーズについて考察する。SR コードが複雑なパターンになると、視認に関して SR コードの使いやすさが低下する。特に、折り返す組み合わせが含まれる SR コードは線同士が狭くなり視認性が低下する。確認フェーズにおいても、GUI による SR コード全体の確認に関しては、視認フェーズと同様のコードで確認しにくくなる。また、音によるフィードバックに関しては、第 3.3.2 項で述べたように直感的ではない右左方向の振る動作を多く含んだ SR コードが使いにくいと考えられる。

最後にスイングフェーズについて考察する。各方向に振る動作の振りやすさに違いがあることは第 3.5.1 項の実験で明らかになったが、SR コードのように連続でモバイル端末を振る場合、どのようなコードが使いやすいのか分かっていない。各 SR コードの振りやすさを求めるためには、振る動作の振りやすさに影響を及ぼす要因なども明らかにする必要がある。よって、本論文では SR コードの振りやすさについて分析を行う。

4.1.2 振りやすさの指標

SR コードの振りやすさを求めるにあたり、振りやすさの指標を定める必要がある。振りやすい SR コードを決める指標として、**振る回数**と**振る動作の認識率**、**ユーザにとって振りやすさ**の 3 種類が存在する。以下でそれぞれの指標について説明する。

振る回数

SR コードは、一方向に振る動作を複数組み合わせたコードである。そのため、組み合わ

せた数によって、ユーザが振るべき回数も異なってくる。当然、振る回数の少なければ早くコードを再現可能であるため、ユーザへの負担が少なく振りやすいコードと言える。振る回数は非常に分かりやすい振りやすさの指標である。

振る動作の認識率

第 3.5.1 項で述べたように、本論文で用いた振る動作推測プログラムは振る方向によって認識精度が異なる。これにより、振る動作を複数組み合わせた SR コードは、それぞれ認識率が異なってくると考えられる。認識率の低いコードは、同じ方向の振りをやり直す可能性が高いため、ユーザにとって振りにくいコードと言える。

認識率は振る動作を推測するアルゴリズムに大きく依存しているため、各アルゴリズムごとに使いやすい SR コードの基準は異なる。さらに究極的にはどんな SR コードでも完璧に認識可能なアルゴリズムが考え出される可能性もある。

ユーザにとっての振りやすさ

ユーザにとっての振りやすさとは、コードに対するユーザの主観的な振りやすさの評価である。第 3.5.1 項で行った実験でも、上方向に振る動作に対して多くのユーザは重力に逆らった動作であるため振りにくいと感じたという結果が出た。ユーザにとっての振りやすさの指標は、振る動作を推測するプログラムのアルゴリズムに依存することがないため、本論文で述べたアルゴリズム以外を本手法に用いた場合でも、SR コードの振りやすさを知る上で有効になる。そこで、本論文ではまずユーザにとっての振りやすさについて分析することで、振りやすい SR コードを導き出す。

4.2 パターンによる振りやすさの検証

第 3.5.1 項では、4 方向のうちの 1 方向を振る動作において振りやすさの違いがある事が分かった。本節では、SR コードが振る動作の構成要素のみによって振りやすさが決まるのか、もしくは組み合わせの順番によっても振りやすさが変化するか実証を行う。

実験概要

振る方向の構成要素は同じだが組み合わせの順番が異なる 6 つの SR コードを用意して、被験者に振る動作で再現させた。構成要素となる振る方向は、右方向が 2 つ、上下が 1 つずつである。図 4.1 に実験に用いられた SR コードを示す。各 SR コードの下にはコード番号が表記されている。ディスプレイに各 SR コードを表示し、被験者をディスプレイの前に直立させた。被験者を直立させたのはパブリック空間では直立した状況でのオブジェクト選択が多いと考えられるためである。被験者に iPhone を持たせ、コードを再現するように振らせた。SR コードを振らせた後、被験者には SR コードの振りやすさを主観的に 4 段階で評

価させた。評価 1 は振りにくい、評価 2 はやや振りにくい、評価 3 はやや振りやすい、評価 4 は振りやすいである。

被験者は男性 5 人、女性 1 人の合計 6 人である。また、被験者は全員右利きだった。

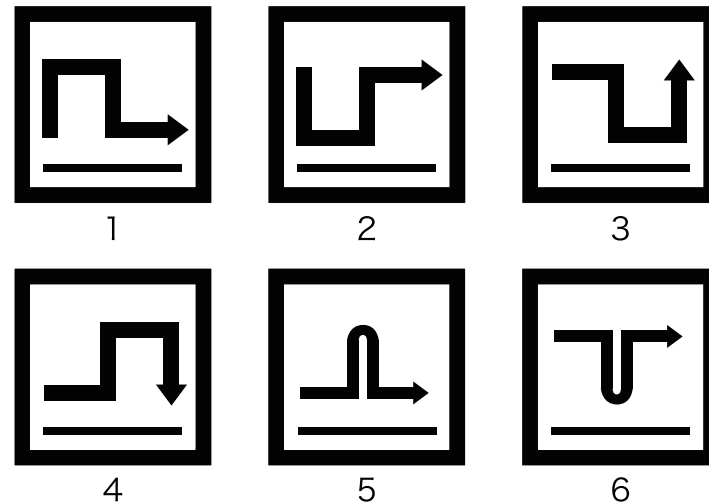


図 4.1 パターンによる検証用 6 種類の SR コード

結果と考察

本検証実験の結果を表 4.1 に示す。表 4.1 からパターンによって SR コードの使いやすさに違いがあることが分かる。振る順番を変えることによって、2 回目以降の振る動作を行う位置が変わった事が原因だと考えられる。例えば、1 番のコードでは最初に上方向に振るため、2 回目に振る動作は最初に振った位置よりも上の場所から振らなければならない。

また、番号 5 と 6 のコードは、他 4 つのコードと比べて評価結果が高かった。番号 5 と 6 のコードは、それぞれ上から下、下から上という戻る組み合わせが含まれている。被験者のインタビューからも、上から右や下から右のような直角に曲がる組み合わせよりも上から下、下から上のような戻る組み合わせのほうが振りやすいことが分かった。戻る振る組み合わせの方が振りやすいと感じた原因としては、端末の位置がもとに戻ることが考えられる。

さらに、番号 1 と 4 のコードが、他のコードに比べ極端に評価結果が低かった。被験者のインタビューから、上方向に振った後に右を振るのが難しい、上を振るなら最後の方が良いといった意見を得た。最初に下方向に振る番号 2 と 3 のコードよりも、最初に上方向に振る 1 と 4 のコードの方が評価結果が低いことから、最初の振り始めから端末が離れた方向によっても振りやすさが異なることが考えられる。

本論文では、SR コードの再現を始める最初の位置をホームポジションと呼ぶ。本実験で用いた全ての SR コードが最終的には最初に振り始めた地点から 2 回右に振った位置で振り終わることになる。そのため、被験者から端末が右側に行き過ぎて振りにくいという意見を得た。

実験結果やインタビューから以下の事を仮定した。

- SR コードの構成要素だけでなく，組み合わせる順番によっても振りやすさは変わる
- ホームポジションから離れた振る動作ほど振りにくいと感じる
- 離れた方向によっても振りやすさが変わる

表 4.1 パターンによる検証の実験結果

コード番号	評価結果
1	2.17
2	2.83
3	2.83
4	2.33
5	3.67
6	3.50

4.3 2 回振るコードの振りやすさの検証

第 4.2 節の実験から直角に曲がる組み合わせよりも戻る組み合わせの方が振りやすいということが分かった。つまり，SR コードを再現する動作を 1 つ 1 つの振りに分けて考えてみると，1 つの振る動作の振りやすさは前の振りに強く影響を受けていることになる。そのため，SR コードの使いやすさを知るためには 2 回の振る動作を合わせて考えてみる必要がある。そこで，本節では上下左右 4 方向を 2 回振る SR コードの全てのパターンの使いやすさを検証する。

実験概要

2 つの振りを組み合わせた SR コードを 12 種類用意し，被験者に振る動作で再現させた。図 4.2 に用意した 12 種類のコードを示す。第 4.2 節の実験と同様に，ディスプレイに SR コードを表示し，被験者にモバイル端末で再現させた。SR コードを振らせた後，被験者には SR コードの振りやすさを主観的に 4 段階で評価させた。最後に，実験を行って感じたこと等をインタビューした。

また，被験者は 20 代の男女 18 名で全員が右利きだった。

実験結果と考察

図 4.5 に評価結果を各コードにマッピングさせた図を示す。評価結果が 3 以上のコードは特に振りやすいコードとし，評価結果を赤く表記する。また，評価結果が 2.0 未満のコードは特に振りにくいコードとし，評価結果を青く表記する。

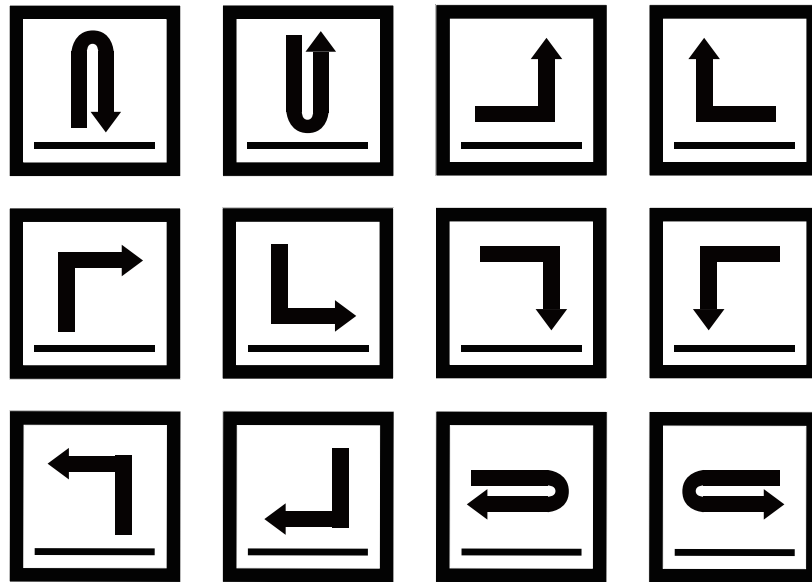


図 4.2 2つの振りを組み合わせた SR コード

被験者の振り方を観察すると、手首のスナップで振る人と腕全体を動かして振る人の2種類に分類されることが分かった。また被験者全体の中で手首のスナップで振る人は3人、腕全体で振る人は15人であった。右手で振る場合、手首の可動範囲は右方向の方が狭いため、手首のスナップで振る人は、右方向への振りが振りにくく左方向への振りが振りやすいと答えた。腕全体で振る人は、左方向へ振ると腕が身体に対してクロスしてしまい窮屈になることで振りにくいと答える人が多かった。逆に多くの人が右方向へは振りやすいと答えた。

図 4.3 に手首を使ったスイング動作の様子を示す。(a) は左に振ったときの様子、(b) は振る前の様子。(c) は右に振ったときの様子を示す。図 4.4 に腕を使ったスイング動作の様子を示す。(a) は右に振ったときの様子、(b) は左に振ったときの様子を示す。

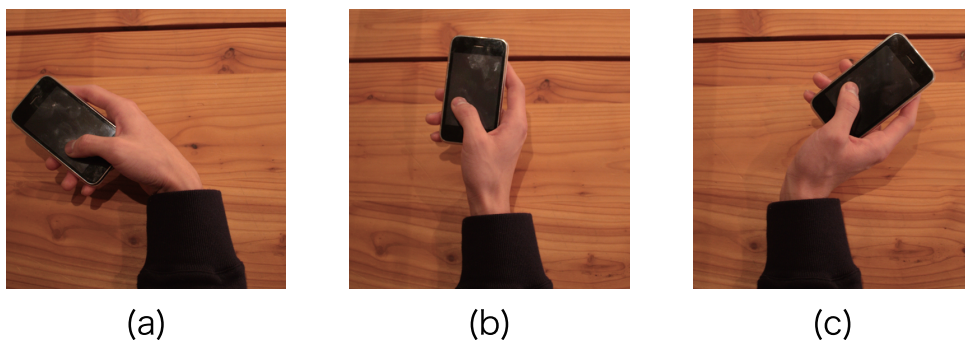


図 4.3 手首を使ったスイング動作

第 3.5.1 節の上下左右に 1 回振る実験では、左方向に振る動作が振りにくいという結果は



図 4.4 腕を使ったスイング動作

得られなかった。1 回振るだけでは腕が身体にクロスしても窮屈だと感じないが、左に 1 回振った位置から次の動作に移る時に窮屈だと感じる傾向にあると考えられる。このことから、1 回の振る動作の振りやすさは振り始める位置の影響を強く受けることが分かる。2 回目の振る動作の開始位置別にコードを分類すると、右側から開始するコード (右→上, 右→下, 右→左) の平均値は 3.00, 左側から開始するコード (左→上, 左→下, 左→右) の平均値は 2.50, 上側から開始するコード (上→下, 上→右, 上→左) の平均値は 2.91, 下側から開始するコード (下→上, 下→右, 下→左) の平均値は 2.79, となり右側, 上側, 下側, 左側の順番で振りやすいことが分かる。

評価結果を見ると上→下や右→左などの折り返す組み合わせのコードの評価の高さが目立っている。4 段階評価の平均値が 3 以上のコードは、全て折り返す組み合わせである。2 回目に振る方向がどの方向でも折り返す組み合わせの評価が高いことから、1 回の振る動作の振りやすさは前の振る動作の方向に大きく影響すると言える。

また、全体的に左方向に振るコードの評価が低い結果になった。これは、腕全体で振る人が被験者の多くを占めていたためである。

4.4 各 SR コードの振りやすさ

本節では、前述した検証実験の結果をもとに、各 SR コードの振りやすさを求める方法を考察する。

第 4.3 節の検証実験で 1 回振る動作の振りやすさには前に振った動作の方向と、振り始める位置の影響を強く受けることが分かった。複数の振る動作から構成される SR コードの振りやすさ求めるために、SR コードを 1 つ 1 つの振りに分けて振りやすさを考えると、1 回の振る動作の振りやすさに影響を与える要素は**振る方向**、**前回の振る方向**、**振り始めの位置**の 3 つであると言える。

各振る動作の開始位置を整理するために、図 4.6 に SR コードを再現するときの各振る動作の開始位置を表現する座標系を示す。ホームポジションを原点とし、左右の振りを x 軸、上下の振りを y 軸とする。x の値は右方向の振り 1 回ごとに 1 増加し、左方向の振り 1 回ごとに 1 減少する。

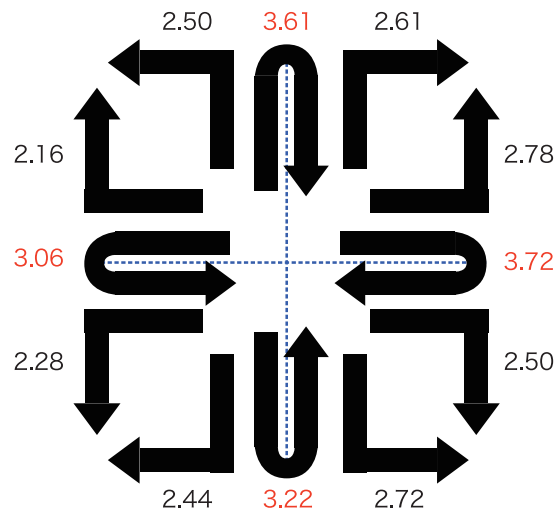


図 4.5 2 回振るコードの振りやすさの評価結果

y の値は上方向の振り 1 回ごとに 1 増加し，下方向の振り 1 回ごとに 1 減少する．つまり，ホームポジションから右方向に 1 回振った後の振る動作の開始位置は (1, 0) と表現出来る．さらに，下方向に振った後の振る動作の開始位置は (1, -1) となる．

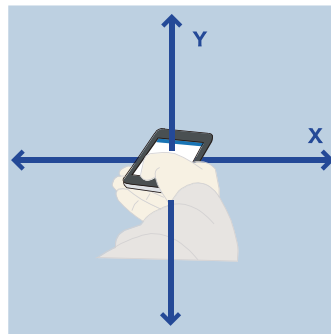


図 4.6 振る動作の位置を表す座標系

さらに，4 回振る SR コードを再現する際に考えられる 2 回振る動作の開始位置を表 4.2 に整理した．表 4.2 をみると，1 回目の振りは (0,0) のホームポジションから必ず振られる．2 回目の振りは，上下左右に 1 回振られた位置から始まることになる．3 回目の振りは，ホームポジションか，ホームポジションから上下と左右の一回ずつ移動した位置から始めることになる．

表 4.2 を視覚的に分かりやすくするために座標系にマッピングしたものを図 4.7 に示す．図 4.7 の (a) から順番に 1 回目，2 回目，3 回目における振り始めの位置となる．

1 つの振る動作の振りやすさは前の振る動作に大きく影響されるため，本論文では，SR コードの振りやすさを求めるために，SR コードを 2 つの振りのセットで分割して考える．図 4.8 に SR コードを 2 つの振りのセットで分割した様子を示す．(a) に右，上，下，右の SR コードを示す．

表 4.2 2 回振る動作の振り始め位置

振る順番	振り始めの座標
1 回目	(0,0)
2 回目	(1,0), (-1,0), (0,1), (0,-1)
3 回目	(0,0), (1,1), (1,-1), (-1,1), (-1,-1)

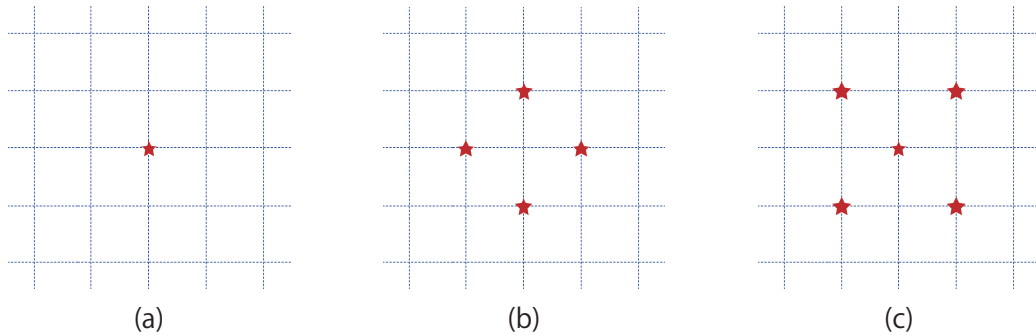


図 4.7 座標系における振り始め位置

(b) に (a) の SR コードを分割したとそれぞれの振る動作の開始座標を示す。(a) の SR コードなら, ”上, 右” と ”右, 下” と ”下, 右” の 3 つに分割出来る。また, 分割した 3 つの振る動作は開始地点がそれぞれ異なり, 1 つ目の動作は (0, 0), 2 つ目の動作は (1, 0), 3 つ目の動作は (1, 1) からとなっている。本論文では, 分割した 3 つの振る動作の使いやすさを求めることで, 全体の SR コードの振りやすさが計算出来ると考えた。

ホームポジションからの 2 回振るコードの振りやすさは, 第 4.3 節の検証実験で求めた。4 つの振る動作を組み合わせた SR コードの使いやすさを求めるためには, 2 回目の振る動作の開始位置である (1,0), (-1,0), (0,1), (0,-1) と, 3 回目の振る動作の開始位置である (1,1), (1,-1), (-1,1), (-1,-1) からの 2 回振る動作の使いやすさを求める必要がある。

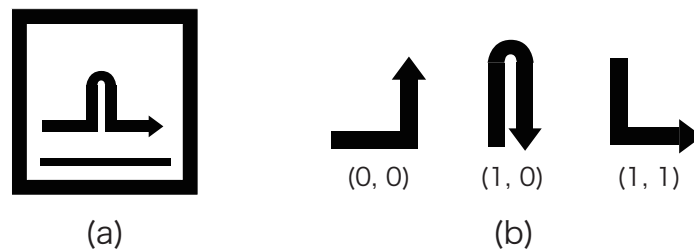


図 4.8 SR コードの分割

4.5 振り始めの位置による影響の検証

第 4.4 節で、SR コードの使いやすさを求めるためには、2 回目の振る動作の開始位置である (1,0), (-1,0), (0,1), (0,-1) と、3 回目の振る動作の開始位置である (1,1), (1,-1), (-1,1), (-1,-1) からの 2 回振る動作の使いやすさを求める必要があると述べた。本節では、上記 8 つの位置から 2 回振るコードを被験者に再現させ、使いやすさを主観的に評価させる。

実験手順

ディスプレイに図 4.2 と同様の 2 回振るコードをそれぞれ表示し、被験者にモバイル端末でコードを再現させた。被験者は直立状態で振る動作を行った。また、振り始めの位置を調整するため 2 回振るコードを再現させる前に調査する振り始めの位置まで端末を振らせた。例えば (1, 0) の開始位置の振りやすさを調査する場合は、右方向に 1 回振らせた。そして、被験者にそれぞれの位置から振ったコードを 4 段階で主観的に評価させた。

結果と考察

本検証実験の結果を、図 4.9, 4.10, 4.11 に示す。図 4.9 は、開始位置を上と下にしたときの 2 回振るコードの振りやすさを示す。図 4.10 は、右と左から振り始めたときの振りやすさ、図 4.11 は、斜めの 4 箇所から振り始めたときの振りやすさを示す。それぞれの図における評価結果が 3 以上のコードは特に振りやすいコードとし、評価結果を赤く表記する。また、評価結果が 2.0 未満のコードは特に振りにくいコードとし、評価結果を青く表記する。

図 4.9, 4.10, 4.11 を見ると、同じ 2 回の振る動作でも開始位置によって全く振りやすさが異なることが分かる。例えば、同じ”右、上”の組み合わせでも、左斜め下 (-1, -1) の開始位置では評価が 3.62 と高いのに対し、右斜め上 (1, 1) の開始位置では 1.77 と非常に低くなっている。この 2 点以外にもホームポジションから近づく振る動作は評価が高く、遠ざかる振る動作は評価が低い傾向になっている。

特に評価が低いのが、斜めの開始位置からさらに斜めに遠ざかる振る動作だった。4 つの位置から考えられる 8 つの組み合わせ全てが評価の平均が 1 点代だった。また、左下に遠ざかる組み合わせは、ほぼ全ての被験者が振りにくいと答えた。

4.6 振りやすさの順位付け

本節では、各 SR コードを振りやすさで順位付けする方法を説明する。

SR コードを第 4.4 節で説明したように、2 回振る組み合わせで分割する。4 つの振る動作を組み合わせた SR コードの場合は、3 つの 2 回振る組み合わせに分割出来る。さらに、2 回振る組み合わせの開始位置を計算し第 4.3 節、第 4.5 節で求めた評価結果から振りやすさの数値を求める。全

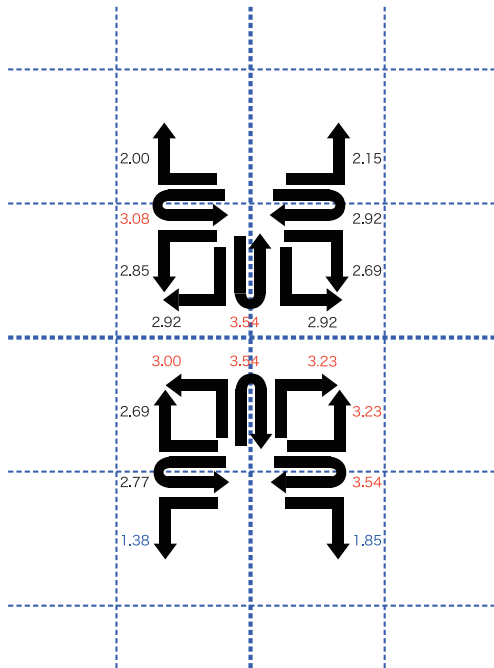


図 4.9 上と下で振り始めたときの評価結果

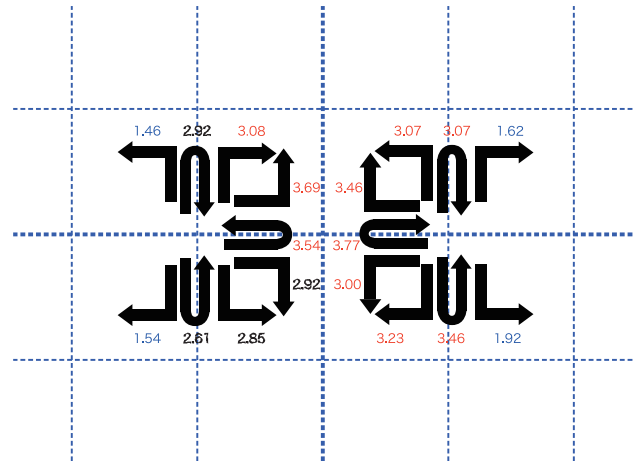


図 4.10 右と左で振り始めたときの評価結果

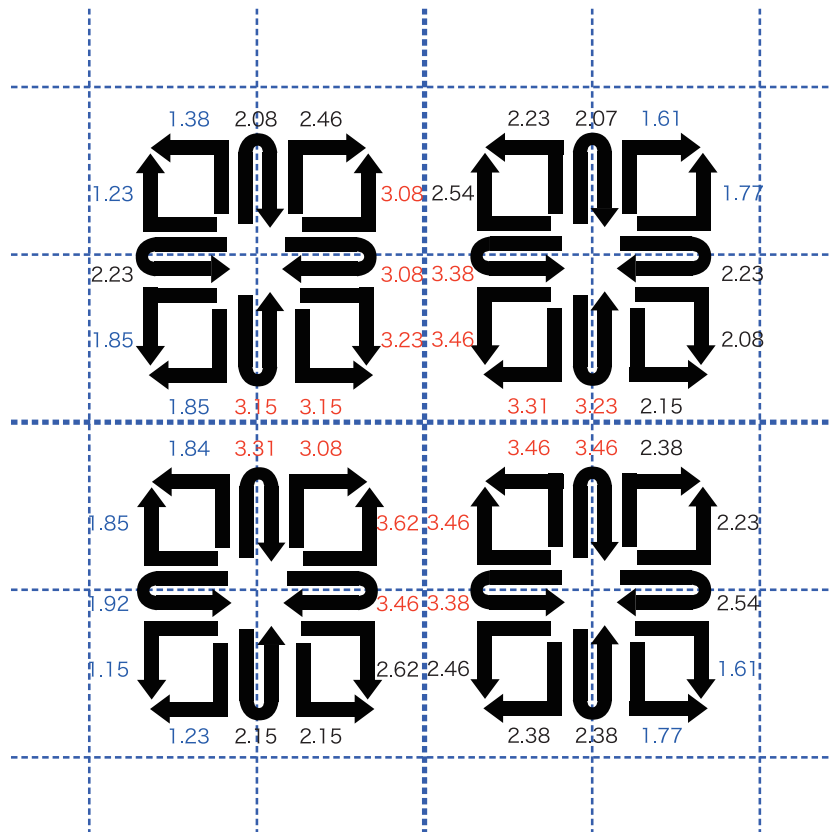


図 4.11 斜めで振り始めたときの評価結果

ての組み合わせの振りやすさの数値を合計することで、その SR コードの振りやすさの数値を求めることが出来る。この数値を比較することで、各 SR コードごとの振りやすさの順番を求めることが出来る。

しかし、3つの振りを組み合わせた SR コードと4つの振りを組み合わせた SR コードは数値を足しただけでは、どちらが振りやすいか比較することが出来ない。しかし、振る回数が異なるため、オブジェクトの選択操作にかかる絶対的な労力が3つの振りを組み合わせた SR コードの方が少ない。そのため、本論文では、振る回数が少ないコードをより振りやすいとし、振る回数が同じコードの同士では振りやすさの値を足して行った値で振りやすさを比較する。

4.6.1 5 回振る SR コードの振りやすさ

前述した実験により、4 回振る SR コードを対象として振りやすさの数値化を行った。しかし、利用空間内のオブジェクトの数が増えた場合、5 回以上振る SR コードをする必要がある。

5 回振る SR コードの振りやすさを数値化する場合は、4 回振る SR コードと同じく、まずは4つの2回振るコードに分解する。1 から3 目までの2回振るコードの振りやすさの数値化は、前述した実験により算出した。4 目目の2回振るコードは、振り始める位置が (1, 2), (1, -2), (-1, 2), (-1, -2), (2, 1), (2, -1), (-2, 1), (-2, -1) になる可能性がある。この振り始める位置での振りやすさはまだ実験を行っていないが、前述した実験で得られた結果を見ると、ホームポジションから離れた振る動作は非常に振りにくいことが分かり、さらにホームポジションから離れたこれらの位置はさらに振りにくいことが考えられる。そのため、4 目目の2回振るコードの振り始める位置が (1, 2), (1, -2), (-1, 2), (-1, -2), (2, 1), (2, -1), (-2, 1), (-2, -1) の SR コードは、使いにくい SR コードとして利用しない。それ以外の SR コードは、5 回振る SR コードでも前述した実験から得られた2回振るコードの振りやすさの値を用いることが出来る。

図 4.12 は、振りやすい5回 SR コードの例である。上下に振る動作を繰り返しているため、ホームポジションから離れていない。図 4.13 は、振りにくい5回振る SR コードの例である。5 回目に振る動作はホームポジションから上に2回、左に3回離れるため非常に振りにくい SR コードである。そのため、本論文では図 4.13 は使わない。

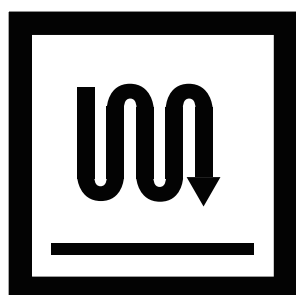


図 4.12 振りやすい5回振る SR コード

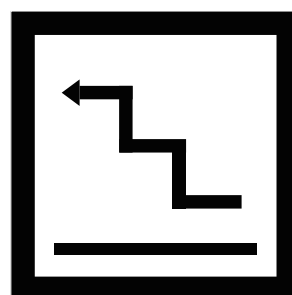


図 4.13 振りにくい5回振る SR コード

4.7 本章のまとめ

本章では、まず SR コードの使いやすさについて整理し、振る動作推測アルゴリズムに依存しないユーザの主観的な振りやすさの指標によって SR コードの振りやすさを求める方針をたてた。そして、検証実験を行い SR コードの振りやすさが振る方向の構成要素だけでなく、パターンによって変化することを明らかにした。さらに、SR コードを 2 回振るコードに分解し、4 回振る SR コードを再現する際に考えられるそれぞれの振り始めの位置から 2 回振るコードの振りやすさを算出する評価実験を行った。これにより、SR コードの振りやすさを数値化し、各 SR コードの振りやすさを比較、順位付けすることを可能にした。

第 5 章

InfoRod の設計と実装

第 3 章で，振る動作を用いた選択手法についての基本設計を行った．続く第 4 章では，本手法に用いる SR コードの振りやすさを比較，順位付けする方法を明らかにした．本章では，本論文で提案した振る動作を用いたオブジェクト選択手法を実現可能にする InfoRod システムの詳細設計及び実装について述べる．

5.1 InfoRod システムの概要

本節では、本論文で提案する振る動作を用いた選択手法を実現する InfoRod システムの概要を説明する。

InfoRod システムはユーザ用ソフトウェアと管理者用ソフトウェアに分かれている。ユーザ用ソフトウェアは、ユーザの持つモバイル端末の中にインストールされていて、振る動作によるオブジェクトの選択を実現する。振る動作による入力を受け付け、振る動作の推測、データベースへの問い合わせ、フィードバックなどを行う。管理者用ソフトウェアは、実世界オブジェクトの SR コード割り当てを管理するソフトウェアである。主に新規オブジェクトの登録、SR コードの割り当てや管理、発行などを行っている。実空間オブジェクトの管理者は、管理者用ソフトウェアを利用して実世界オブジェクトに SR コードを割り当て、発行し、貼付する。

5.2 InfoRod システムの実現方法

振る動作を用いたオブジェクト選択手法の実現を考えた時、InfoRod システムをローカル空間に限定して利用する場合とあらゆる場所で利用する場合の 2 種類の方法が考えられる。これら 2 つの方法をローカル空間での利用とグローバル空間での利用とし以下で詳細を述べる。

ローカル空間での利用

ローカル空間に限定して利用する場合の例としてイベント会場が挙げられる。イベントの主催者が管理者となり、イベント来場者のモバイル端末に InfoRod のユーザ用アプリケーションを配布する。イベント空間上に存在する選択対象オブジェクトの数が多すぎると、エリアを複数に分割し、エリアごとにオブジェクトを管理する必要がある。

グローバル空間での利用

InfoRod システムをあらゆる場所では実現する場合は、InfoRod システムをグローバルで運営する必要がある。パブリック空間に存在するオブジェクトの管理者は、グローバルな InfoRod システムに登録することで InfoRod によるオブジェクト選択が可能となる。オブジェクトを InfoRod システムに登録する際、オブジェクトの位置情報とオブジェクトの視認可能範囲をシステムに入力する必要がある。つまり、動く可能性のあるオブジェクトは基本的に利用出来ない。(特例として全ての場所で利用可能な SR コードを確保するということも可能ではある)

あらゆる場所の選択を実現出来る方が望ましいが、解決すべき課題がいくつか存在する。最も大きい問題はモバイル端末からユーザの正確な位置情報取得可能になることである。特に屋内や地下でも正確に位置情報を取得可能であり、さらに建物にいる場合は駅の何階に居るのか知ること重

要になる。また、オブジェクトの正確な視認可能範囲を調べることは大きな負担となってしまう。建物の窓から見える看板広告など、階層をまたいでの視認などを上手く管理することはとても難しい。これらの理由から本論文では、次節からローカル空間での利用を想定して InfoRod システムの設計と実装を行う。

5.3 ユーザ用アプリケーションの設計

本節では、InfoRod におけるユーザ用アプリケーションの設計について述べる。まず、ソフトウェア構成について説明した後、ハードウェア構成について説明する。

5.3.1 ソフトウェア構成

InfoRod におけるユーザ用アプリケーションは、大きく分けて7つのモジュールと1つのデータファイルから構成される。図 5.1 に InfoRod における利用者側アプリケーションのソフトウェア構成図を示す。また、以下にそれぞれのモジュールとデータファイルについて説明する。

動作データ取得モジュール

動作データ取得モジュールは、動作を推測するために必要なデータを取得するモジュールである。本アプリケーションでは、モバイル端末に内蔵されている加速度センサからデータを取得する。

動作推測モジュール

動作推測モジュールは、モバイル端末の動きを推測するモジュールである。動作データ取得モジュールから加速度データを受け取り解析を行う。本モジュールでは、方向解析部と操作ジェスチャ解析部に分かれていて、上下左右に振られた動作と操作ジェスチャを推測する。本モジュールは推測結果を、選択受付モジュールとサービス実行モジュールに受け渡す。選択受付モジュールには上下左右に振られた方向と操作ジェスチャを返し、サービス実行モジュールには操作ジェスチャのみを返す。

フィードバックモジュール

フィードバックモジュールは、アプリケーションが推測した動作結果をユーザにフィードバックとして返すモジュールである。本モジュールでは、選択受付モジュールから受け渡された上下左右に振られた方向に対して、音と GUI の 2 通りの方法でユーザにフィードバックを返す。

選択受付モジュール

選択受付モジュールは、ユーザからの振る動作での入力を受け付けるモジュールである。

動作推測モジュールから振る動作の推測結果を受け取り、現在振られた方向を配列として蓄積し、フィードバックモジュールに振られた方向を渡す。キャンセル操作が結果として返ってきたら配列の方向を消去し、選択の決定操作が結果として返ってきたら特定情報取得モジュールに方向配列データを渡す。

特定情報取得モジュール

特定情報取得モジュールは、オブジェクトを特定するために必要なデータを集めるモジュールである。選択受付モジュールから方向配列データを受け取ると、現在時刻、現在位置のデータを取得し、オブジェクト情報取得モジュールへ全てのデータを渡す。

オブジェクト情報取得モジュール

オブジェクト情報取得モジュールは、InbfoRod データベース問い合わせでオブジェクトの詳細情報を取得するモジュールである。特定情報取得モジュールから受け取ったデータからクエリを作成し、InfoRod データベースの問い合わせを行う。オブジェクトの詳細情報を受け取ったら、サービス実行モジュールへ渡す。

サービス実行モジュール

サービス実行モジュールは、選択したオブジェクトからサービスを実行するモジュールである。オブジェクト情報取得モジュールから選択したオブジェクトの詳細情報を受け取り、サービスを利用可能な状態にする。さらに、詳細情報から読み取った操作ジェスチャの情報を学習結果データに送る。そして、サービスと操作ジェスチャとマッピングを行い、動作推測データからの推測結果からサービスを実行する。また、振る動作だけでなく GUI からのサービス実行も実現する。

学習結果データ

学習結果データは、操作ジェスチャを第 3.4.3 項で説明したニューラルネットワークアルゴリズムで推測する際に必要なデータファイルである。各ニューロンの発火のしやすさ、次のニューロンへの結合の強さを表す。

5.3.2 ハードウェア構成

InfoRod におけるユーザ用アプリケーションに必要なデバイスは大きく分けてモバイル端末自身とネットワーク上のデータベースの 2 つである。そして、モバイル端末自身に動作解析デバイス、位置情報取得デバイス、フィードバックデバイスを持つ必要がある。また、モバイル端末はネットワークに通信可能である必要がある。動作解析デバイスには、加速度センサや画像解析に用いるカメラなどが利用出来る。位置情報取得デバイスには、GPS や Wifi が受信出来れば電波の強度

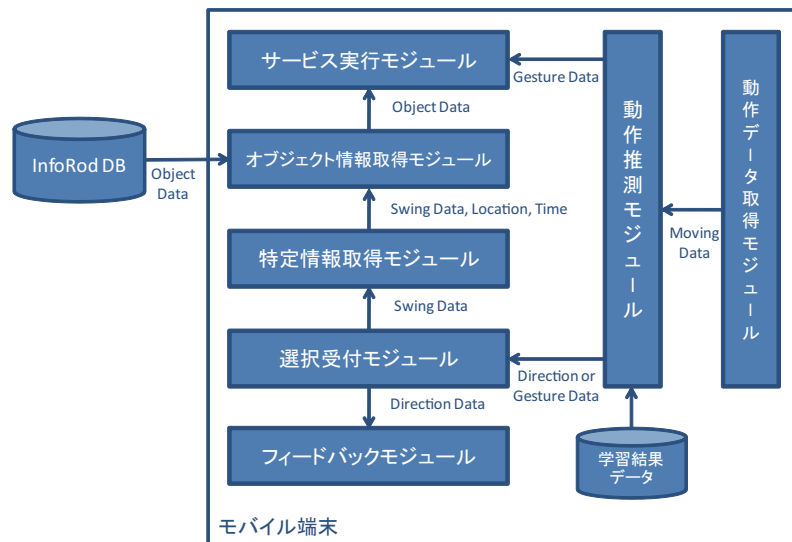


図 5.1 ユーザ用アプリケーションのソフトウェア構成図

から位置情報を取得する PlaceEngine [24] や Locky.jp [25] を利用可能である。GPS は、地下や屋内や高層ビル群などでは位置情報を正確に取得するのが難しい。PlaceEngine や Locky.jp は無線 LAN の電波が届かない場所では位置情報を取得出来ない。GPS と PlaceEngine, Locky.jp を 2 つ合わせて用いることで、より広い範囲で正確に位置情報を取得することが可能である。フィードバックデバイスとしては、スピーカーとディスプレイが必要である。スピーカーでは、音によるフィードバックを返し、ディスプレイでは GUI によるフィードバックを返す。

5.4 管理者用アプリケーションの設計

本節では、InfoRod における管理者用アプリケーションのソフトウェア構成について述べる。ソフトウェア構成では、InfoRod を構成するモジュール群を挙げ、各モジュール群の機能を述べる。管理者用アプリケーションは、複数の管理者が容易に利用出来ることを考慮し、Web アプリケーションとして設計を行った

5.4.1 ソフトウェア構成

管理者用アプリケーションは、オブジェクト管理機構と管理者用 UI モジュールから構成される。以下でそれぞれの機構、モジュールの詳細を述べる。

管理者用 UI モジュール

管理者用 UI モジュールは、本アプリケーションのユーザインタフェースを実現するモジュールである。主に地図の表示、オブジェクトの表示、入力の実現する。InfoRod

データベースから各オブジェクトの詳細情報を取得し、地図上にオブジェクトをマッピングする。また、指定されたオブジェクトの詳細情報などを表示する。オブジェクトの新規追加、変更、削除などの入力受付があった場合は、入力結果をオブジェクト管理機構に渡す。

オブジェクト管理機構

オブジェクト管理機構は、オブジェクトのデータを管理する機構である。本機構は、コード自動割り当てモジュール、整合性チェックモジュール、コード発行モジュールのモジュールを擁している。コード自動割り当てモジュールは、コードの自動割り当て要求があった場合、割り当て可能な SR コードのリストから最も振りやすい SR コードを選びオブジェクトに割り当てる。整合性チェックモジュールは、オブジェクトの新規追加や内容変更があった場合に同一空間内のオブジェクトとの整合性が保たれているか確認をする。コード発行モジュールは、SR コードの画像を生成し、画像データを管理者用 UI モジュールに返す。

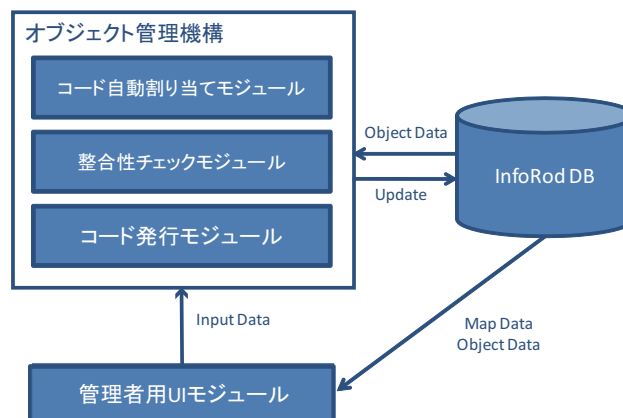


図 5.2 管理者用アプリケーションのソフトウェア構成図

5.5 データベースの設計

本節では、InfoRod システムにおけるデータベース構成について説明する。InfoRod データベースでは7つのテーブルから構成されている。各テーブルの詳細を以下で述べる。

Object テーブル

Object テーブルは InfoRod データベースにおけるメインカラムである。Object テーブルは、Owner テーブル、Swing テーブル、Location テーブルの外部キーとして owner_id、swing_id、location_id を保持している。さらに、オブジェクトのアイコン画像の URL(icon_url)、オブジェクト作成日時(created_date)、InfoRod システムによる選択の開始日時(start_date)、終了日時(finish_date) を保持する。

Swing テーブル

Swing テーブルは、SR コードの振る動作を表現したテーブルである。SR コードを構成する振る動作の方向を direction1～direction4 に保持する。上下左右の振る動作はそれぞれ、"Up", "Down", "Right", "Left" の文字列で表現される。

Location テーブル

Location テーブルは、オブジェクトの位置を表現するテーブルである。主キーとなる id と、緯度 (latitude), 経度 (longitude) を保持する。また、管理用アプリケーションで使用する地図データが単一の画像の場合は、x、y 座標を保持する。

Resource テーブル

Resource テーブルは、オブジェクトの持つリソースを表現するテーブルである。主キーとなる id, このリソースを保持するオブジェクトへの外部キーとして object_id を保持する。また、リソース名 (name), データタイプ (data_type), 転送モード (transfer_mode), このリソースの説明文 (description) を保持する。データタイプは url や image, music, movie が入り、転送モードは、packet, stream, event が入る。

Function テーブル

Function テーブルは、オブジェクトの持つファンクションを表現するテーブルである。主キーとなる id, このファンクションを保持するオブジェクトへの外部キーとして object_id, このファンクションに対応した操作ジェスチャへの外部キーとして gesture_id を保持する。IP アドレスやポート番号, URL などファンクションを実行する際のポインター (pointer) や、ファンクション名 (name), ユーザの入力データタイプ (input data_type), 転送モード (transfer_mode), このファンクションの説明文 (description) を保持する。

Gesture テーブル

Gesture テーブルは、操作ジェスチャを表現するテーブルである。主キーとなる id を保持する。また、操作ジェスチャ名 (name), 学習結果ファイル (learned_data), 操作ジェスチャの説明文 (description), 操作ジェスチャイメージへの URL (image_url) を保持する。

Owner テーブル

Owner テーブルは、オブジェクトの管理者を表現するテーブルである。主キーとなる id, 管理者名 (real_name), ログイン名 (login_name), パスワード (password), 権限 (authority) を保持する。

5.6 ユーザ用アプリケーションの実装

本節では、ユーザ用アプリケーションの実装について説明する。まず、ハードウェア、ソフトウェアの実装について述べ、本アプリケーションの利用方法を説明する。

5.6.1 ハードウェア

本論文では、ユーザ用アプリケーションを搭載するモバイル端末として apple 社の iPhone を利用した。iPhone は、5.3.2 で述べたユーザ用アプリケーションにおけるハードウェア要件を満たしている。また、3.4.1 で述べた理由から本論文では振る動作を加速度センサを用いる。そのため、加速度センサを内蔵している iPhone を実装で利用した。iPhone のスペックを表 5.1 に示す。

表 5.1 iPhone のスペック

名称	スペック
サイズ	高さ 155.5mm 幅 62.1mm 奥行き 12.3mm
重量	135g
液晶サイズ	3.5 インチ
解像度	480 × 320 ピクセル
OS	Mac OS X
CPU	412MHz
メモリ	128MB
HDD	8GB
通信機能	Quad-band(850/900/1800/1900MHz)GSM/EDGE
無線 LAN	IEEE802.11b/g
位置情報	Assisted GPS(A-GPS), デジタルコンパス, Wi-Fi, 携帯電話通信
その他	スピーカ内蔵

5.6.2 ソフトウェアの実装

本論文では、ユーザ用アプリケーションを iPhone 用アプリケーションとして実装した。以下でユーザ用アプリケーションを構成する主要モジュールについての説明を行う。

動作推測モジュール

動作推測モジュールは、方向解析部と操作ジェスチャ解析部から構成されている。動作データ取得モジュールから、50 ミリ秒毎に X,Y,Z 方向の加速度データを受け取り、加速度

配列を作成している。この加速度配列をもとに2種類の解析部によって動作の推測を行っている。各解析部の説明は、第3.4.2項と第3.4.3項で述べている。また、加速度配列は操作ジェスチャ解析部、方向解析部の順番で解析を行う。操作ジェスチャ解析部で、加速度データ配列が操作ジェスチャの動作であると判定されれば、その加速度配列は方向解析部で解析されることなく配列がリセットされる。

サービス実行モジュール

サービス実行モジュールは、まず InfoRod データベースからオブジェクトの保持するリソースとファンクションの情報を受け取る。オブジェクトにリソースの情報しかない場合は、すぐにリソースを再生させる。リソースが url の場合はブラウザを起動し Web ページを表示させ、音楽ファイルや画像ファイル、動画ファイルの場合を再生させる。オブジェクト情報にファンクションまたは、ファンクションとリソースがある場合は、サービス選択を受け付ける画面を表示する。この画面では GUI によるサービスの実行と、操作ジェスチャによるサービスの実行が可能である。操作ジェスチャによるサービスの実行は、各ファンクションに対応する操作ジェスチャと動作推測モジュールの結果をマッピングさせることで実現する。

5.6.3 利用方法

本項では、利用者用アプリケーションの利用方法を説明する。

図 5.3 にユーザ用アプリケーションの画面遷移の様子を示す。まず、アプリケーションを起動するとオブジェクト選択画面が表示される。オブジェクト選択画面では、振る動作によるオブジェクト選択を受け付けている。ユーザの利用したいオブジェクトに”上、下、右、上”の SR コードが貼られていたら、ユーザはこの画面でモバイル端末を上、下、右、上の順にモバイル端末を振る。1回の振る動作ごとに中央の画面に矢印が追加されていく。全ての振る動作を終え、アプリケーションがユーザの振った方向を正しく認識したら、画面の中央にオブジェクトに貼られたコードと同じコードが表示される。ユーザはつり上げるジェスチャを行うことによってオブジェクトの選択を決定することが出来る。オブジェクトの選択完了すると、画面にはオブジェクトのイメージ画像と利用可能なファンクションリスト、リソースリストが表示される。各ファンクション、リソースのボタンを押すとサービスを実行することが出来る。また、選択したオブジェクトが Web ページの URL や、画像ファイル、音楽ファイルなどのリソースしか持たない場合は、この画面に移行することなくリソースが表示される。複数のリソースを持つ場合は、リソースリストが表示される。

5.7 管理者用アプリケーションの実装

本節では、管理者用アプリケーションの実装について説明する。まず、ソフトウェアの実装について述べ、本アプリケーションの利用方法を説明する。

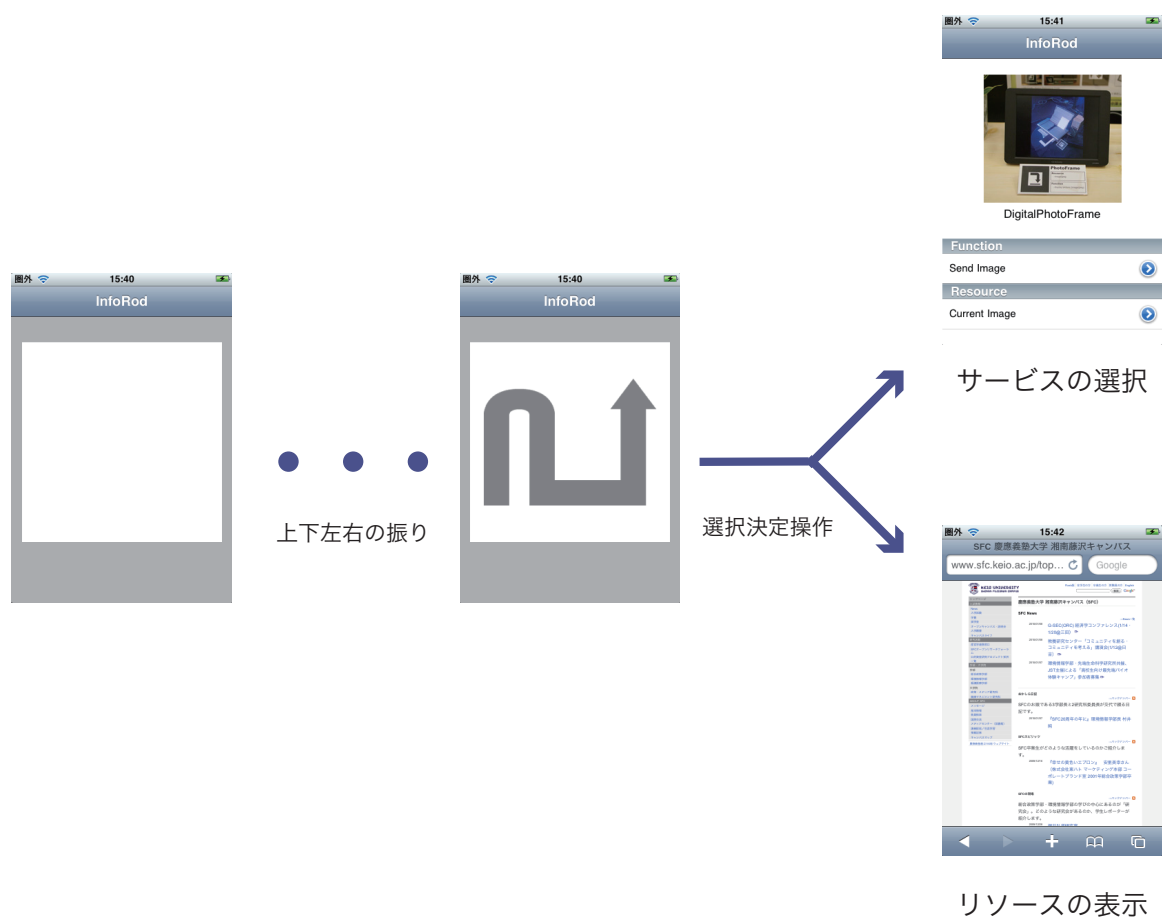


図 5.3 ユーザ用アプリケーションの画面遷移

5.7.1 ソフトウェアの実装

本項では、管理者用アプリケーションを構成する主要モジュールについての説明を行う。以下で、整合性チェックモジュールとコード自動割り当てモジュールについての説明を行う。

管理者用 UI モジュール

情報表示モジュールは、InfoRod に登録されている情報を地図上に表示する地図表示部と、登録されている詳細情報を表示する詳細情報表示部から構成される。地図表示部を実装するため、本論文では GoogleMapsAPI を利用し、GoogleMap 上にマーカーとして登録されている情報を表示した。マーカーをクリックすると、対象の詳細情報が右側のフォームに表示される。図 5.4 に情報表示モジュールのスクリーンショットを示す。情報表示モジュールは、GoogleMapsAPI を利用するため、JavaScript を用いて実装を行った。

コード自動割り当てモジュール

コード自動割り当てモジュールは、指定されたオブジェクトに現在割り当て可能な SR コードの中から最も振りやすいものを割り当てるモジュールである。本モジュールは、第 4.3 節と、第 4.5 節で求めた各開始位置から 2 回振るコードの振りやすさのデータを保持する。さらに、第 4.6 節で説明した SR コードの振りやすさの計算により、現在割り当て可能な SR コードそれぞれの振りやすさを数値化し比較することで、最も振りやすい SR コードを特定する。



図 5.4 管理者用アプリケーションの画面

5.8 本章のまとめ

本章では本論文で提案した振る動作を用いた実世界オブジェクト選択手法を実現する InfoRod システムについて述べた。InfoRod システムはユーザ用アプリケーションと管理者用アプリケーションから構成されるため、それぞれの設計と実装、さらにデータベースの設計について詳細を説明した。

第 6 章

評価実験

本章では，本論文で求めた各 SR コードの振りやすさの正当性を評価する．本実験では，ランダムに選んだ 4 種類の SR コードを被験者にモバイル端末を使って再現させ，4 つの SR コードをそれぞれ振りやすい順番に答えさせる．その後，本実験で被験者の答えた振りやすさと本論文で求めた各 SR コードの振りやすさを比較し，考察する．

6.1 実験概要

振る動作を用いた実世界オブジェクト選択手法では、振る動作を示した SR コードをユーザが再現することで選択を可能にする。しかし、振る動作の特性から SR コードはそれぞれ振りやすさが異なる。パブリック空間に設置されたオブジェクトは利用される頻度や優先度が異なることから、選択操作の使いやすさに大きな影響を及ぼす SR コードの振りやすさを明確にすることはとても重要である。そのため、本論文では 2 回振るコードの振りやすさを振り始める位置ごとに評価することで、SR コードの振りやすさを値として算出した。本実験では、計算した算出した SR コードの振りやすさとユーザが実際に SR コードを振ってみて感じた振りやすさを比較することで、本論文で算出した SR コードの振りやすさの値の正当性を評価する。

本節では、はじめに本実験を実地した実験環境を示し、ついで本実験の被験者を述べる。その後、本実験の実験手順を説明する。

6.1.1 実験環境

本研究では、30 インチのディスプレイに 4 つの SR コードを 1 列に並べて表示する。また、ディスプレイの 1m 手前に被験者を立たせ、振る動作をスムーズに行えるように被験者の半径 1m の障害物になり得るものを取り除いた。また、本実験では振る動作の推測アルゴリズムに左右されない SR コード自体の振りやすさを計測するため、モバイル端末上では振る動作を推測するソフトウェアを起動させなかった。図 6.1 に本実験を実施した実験環境を示す。

本実験で用いた SR コードは全て 4 つの振りを組み合わせたコードで、ランダムに生成した。図 6.2 に本実験で用いた SR コードを示す。SR コードを 4 つごとにグループ分けし点線で囲った、本実験では全部で 10 グループ用意した。図 6.2 に示す通り、グループは上から順番に番号付けし、グループ内のコードを左から (a), (b), (c), (d) としている。

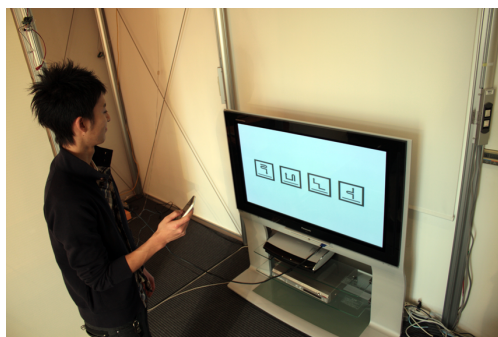


図 6.1 実験環境

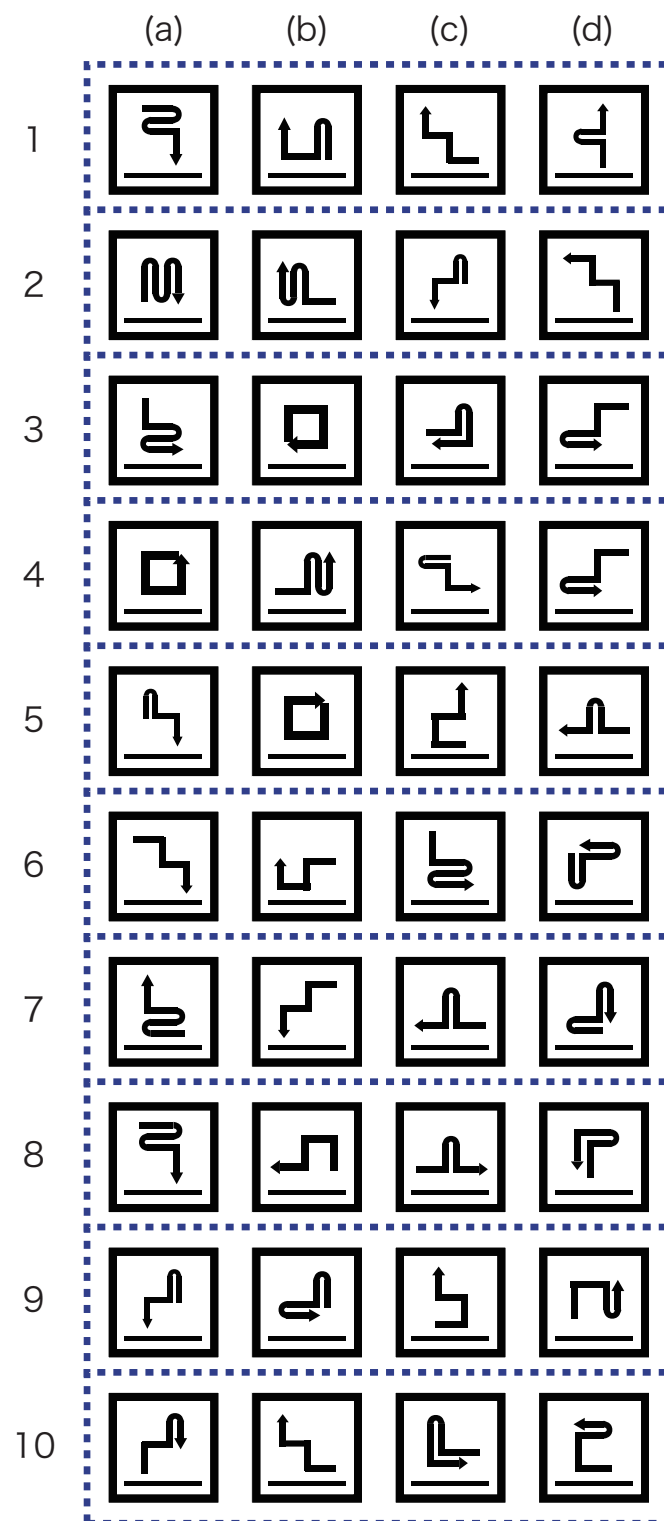


図 6.2 評価実験で用いた SR コード群

6.1.2 被験者

本実験の被験者は男性 11 人，女性 2 人の合計 13 人である．最低年齢は 19 歳，最高年齢は 56 歳である．また，被験者は全員右利きであった．

6.1.3 実験手順

被験者の目の前にあるディスプレイ上に図 6.2 に示したグループごとに SR コード並べて表示した．そして，被験者にそれぞれの 4 つの SR コードをモバイル端末を振ることで再現させた．その後，4 つの SR コードを振りやすいと感じた順に答えさせた．また，すぐに振りがやすさが判別出来なかった場合は複数回 SR コードを再現させた．

6.2 実験結果

本節では，評価実験の結果を述べる．

表 6.1 に評価実験の結果を示す．表 6.1 では，グループごとに評価結果の平均値と算出した振りがやすさの値を示している．評価結果の平均値は，被験者が 4 つのコードを順位付けした際に，1 位に 4 点，2 位に 3 点，3 位に 2 点，4 位に 1 点を割り当て，全ての被験者の順位付けの結果の平均値である．また，小数点以下第二位を四捨五入している．算出した振りがやすさの値は，第 4.6 節で述べている計算方法で SR コードの振りがやすさを計算した値である．

表 6.1: 評価実験の結果

グループ	SR コード	評価結果の平均	算出した振りがやすさの値
1	右, 左, 右, 下	3.23	9.99
	上, 下, 左, 上	2.62	8.69
	左, 上, 左, 上	1.77	4.85
	上, 左, 右, 上	2.38	8.66
2	上, 下, 上, 下	3.38	10.76
	左, 上, 下, 上	2.23	8.23
	上, 下, 左, 下	2.54	8.81
	上, 左, 上, 左	1.85	5.88
3	下, 右, 左, 右	2.58	9.64
	上, 右, 下, 左	2.75	8.61
	右, 上, 下, 左	2.67	9.16
	左, 下, 左, 右	2.00	5.47
4	左, 下, 右, 上	2.69	8.75

表 6.1: 評価実験の結果

グループ	SR コード	評価結果の平均	算出した振りやすさの値
	右, 上, 下, 上	3.31	9.08
	左, 右, 下, 右	2.15	8.70
	左, 下, 左, 右	1.85	5.47
5	上, 下, 右, 下	2.69	9.03
	下, 左, 上, 右	2.62	8.21
	左, 上, 右, 上	2.46	8.32
	左, 上, 下, 左	2.23	6.93
6	右, 下, 右, 下	1.62	6.03
	左, 下, 左, 上	2.23	5.67
	下, 右, 左, 右	2.92	9.64
	下, 上, 右, 左	3.31	10.17
7	左, 右, 左, 上	2.77	8.76
	左, 下, 左, 下	1.31	4.97
	左, 上, 下, 左	2.85	6.93
	左, 右, 上, 下	3.08	10.36
8	右, 左, 右, 下	3.00	9.99
	上, 左, 下, 左	1.92	7.20
	右, 上, 下, 右	2.54	8.00
	上, 右, 左, 下	2.54	8.99
9	上, 下, 左, 下	2.46	8.81
	上, 下, 左, 右	3.77	9.59
	右, 上, 左, 上	1.69	8.39
	上, 右, 下, 上	2.08	8.53
10	上, 右, 上, 下	2.38	6.83
	左, 上, 左, 上	1.92	4.85
	左, 上, 下, 右	2.92	8.23
	左, 上, 右, 左	2.77	8.32

6.3 考察

本節では、本実験の評価結果を考察する。

表 6.1 で示した評価結果の平均と算出した振りやすさの値をそれぞれグループないで順位付けする。表 6.2 で 2 つの順位を比較する。本実験の順番、算出した順番では、グループ内の (a), (b),

(c), (d) のコードの順位を左から示す。全順位の一致では、本実験の順番と算出した順位の順番が完全に一致している場合は○とし、それ以外は×とする。1位の一致では、順位1位のコードがお互いに一致する場合は○とする。4位の一致では、順位4位のコードがお互いに一致する場合は○とする。

本実験によって求められた順位が本当のSRコードの振りやすさの順番とすると、本論文によって4つのSRコードの振りやすさの順番を完全に推測出来たのは10グループ中5つであった。グループ8では、(c)と(d)のコードの評価の平均が同じ値だったため、全順番の一致とした。

オブジェクトを管理する際に、割り当て可能なコードの中から1番振りやすいコードを割り当てたいという要求がある。本実験では、10グループ中8グループの1位のSRコードを計算によって求めることが出来た。また、1番使いにくいコードの場合は10グループ中9グループで推測することが出来た。さらに、順番が完全に一致しなかったグループに着目してみると、5グループ中4グループは、2つのコードの順番が前後しているだけだった。つまり、振りやすさ1位のコードを4位に推測してしまうような大きな間違いは少なかった。

また、本実験での順番と算出した順番における全てのコードの順位の誤差を足して行くと12になる。1グループにおける最大の誤差は、1位と4位、2位と3位、3位と2位、4位と1位のような組み合わせになり6になる。全10グループの最大の誤差は60になるため本論文で導いた手法による推測は2割の誤差率となる。

表 6.2 順位の比較

グループ	本実験での順番	算出した順番	全順番の一致	1位の一致	4位の一致
1	1, 2, 4, 3	1, 2, 4, 3	○	○	○
2	1, 3, 2, 4	1, 3, 2, 4	○	○	○
3	3, 1, 2, 4	1, 3, 2, 4	×	×	○
4	2, 1, 3, 4	2, 1, 3, 4	○	○	○
5	1, 2, 3, 4	1, 3, 2, 4	×	○	○
6	4, 3, 2, 1	3, 4, 2, 1	×	○	×
7	3, 4, 2, 1	2, 4, 3, 1	×	○	○
8	1, 4, 2, 2	1, 4, 3, 2	○	○	○
9	2, 1, 4, 3	2, 1, 4, 3	○	○	○
10	3, 4, 1, 2	3, 4, 2, 1	×	×	○

6.4 本章のまとめ

本章では、評価実験を行うことで本論文で導いたSRコードの振りやすさを順位付けする手法の正当性を評価した。評価実験では、4つのコードの中からユーザが主観的に振りやすいと思った順に順位付けをさせる。その結果と、本論文で導いた手法により計算された順位付けと比較すること

で，本論文で導いた手法によりどの程度 SR コードの振りやすさを推測可能か示した．

第 7 章

結論

本章では，本論文のまとめと今後の展望を述べる．

7.1 本論文のまとめ

本論文では、モバイル端末を振るという動作を用いた実世界オブジェクト選択手法を提案した。本手法では、各実世界オブジェクトに振る動作を示したコードを貼付する。ユーザは、コードを振る動作で再現することによってオブジェクトの選択をする。

本手法では、オブジェクト選択に上下左右の振る動作を複数組み合わせで作った SR コードを用いる。上下左右の認識率の実験により、上下左右各方向の振りは振りやすさが異なることが分かり、SR コードごとに振りやすさに大きな差が出ることが分かった。そのため、本論文では数回にわたる検証実験により、振る動作の振りやすさに影響する要因について整理した。

まず、モバイル端末を振るとき、ユーザは手首で振る人と腕で振る人の 2 種類に分かれる。このモバイル端末の振り方によって、ユーザの振りやすいと思う方向が異なる。モバイル端末を右手で持ち手首で振る場合は、手首の可動領域が右方向に狭いため、右方向への振りを 2 回以上行くと振りにくいと感じ、逆に左方向への振りは、比較的振りやすいと感じる。モバイル端末を右手で持ち腕で振る場合は、左方向への振りを 2 回以上行くと、腕が身体にクロスしてしまうため振りにくいと感じ、右方向へは自由に振ることが出来るため、振りやすいと感じる。本論文の評価実験における被験者を見てみると、手首で振る被験者は少数だった。

また、上下左右の振る動作を複数回組み合わせる場合、振り始める位置と前の振りが大きく振りやすさに影響してくる。振り始める位置がホームポジション (最初に振った位置) から離れている場合、ホームポジションに近づく振りは振りやすいと感じ、ホームポジションから遠ざかる振りは極端に難しいと感じる。

さらに各開始地点から全ての 2 回振る動作の振りやすさの主観的な値を検証実験によって求めた。これにより、上下左右を組み合わせたコードの振りやすさを求めることが可能になる。この振りやすさの値は、振る動作の推測プログラムに依存するものではないため、多くの振る動作を用いたソフトウェアや、研究に用いることが出来る。

7.2 今後の展望

本節では、本論文における今後の展望を述べる。以下に、本論文における今後の展望を整理する。

グローバル空間での実現

本論文では提案した手法を実現する InfoRod システムをローカル空間に限定して設計、実装した。グローバルな空間で本手法を実現しようとした場合、SR コードの視認可能な範囲をオブジェクト管理者がシステムに登録する必要がある。屋外にあるオブジェクトなどは、近くの建物の窓からなど階層を超えた視認が可能となってしまうため、視認可能な範囲や空間をしっかりと定義し、グローバルな空間でも利用可能にすることが必要である。

振る動作学習メソッドの確立

再現する SR コードによって振り始める位置を調整することで、振りやすさは向上する。例えば、右方向に大きく移動する SR コードを再現する場合、あらかじめ最初に振る位置を左にすると比較的スムーズに振る動作を行うことが出来る。しかし、検証・評価実験では、再現する SR コードによって最初に振る位置を調整する被験者はほとんどいなかった。これは振る動作に対して被験者が慣れていないことが原因だと推測出来る。また、モバイル端末を振る動作は腕で振る場合と手首で振る場合に分かれる。腕で振る場合は、1 回に振る移動距離が長いと、手首で振る場合は 1 回に振る手首の角度の大きさが大きいと次に振る動作が振りにくくなる。これらの事から、ユーザが簡単に振る動作を行えるような学習メソッドを用意することで、振る動作を用いたソフトウェアが広く普及することが可能になると考えている。

振る回数の異なるコードにおける振りやすさ

本論文では、振る回数の少ないコードほどユーザの負担が少ないと仮定して振りやすさを数値化し、同じ振る回数の SR コードに限定した振りやすさの順位付けを行った。つまり、3 回振るコードの中で最も振りにくいパターンの SR コードと 4 回振るコードの中で最も振しやすいパターンの SR コードの振りやすさを本論文で実装した自動コード割り当てモジュールが比較すると 3 回振るコードの方が振りやすいコードとして認識される。今後は、振る回数の少ないコードほど振りやすいという仮定を検証するとともに、異なる振る回数のコード同士でも使いやすさの比較を可能にする方法を確立する必要がある。

さらなる振る動作の分析

本論文では、右利きの人でしか検証実験を行わなかった。また、被験者の多くが 20 代の男性と非常に偏ってしまったこと、被験者の人数が少ないことも問題点として挙げられる。そのため、今後は幅広い年齢層・性別・利き腕の被験者で多くの実験を行い、さらに正確な振りやすさを分析したい。

謝辞

本研究を進めるにあたり，御指導を頂きました慶應義塾大学大学院政策・メディア研究科 徳田英幸博士に深く感謝致します。また，同大学環境情報学部准教授 高汐一紀博士，同大学環境情報学部専任講師 中澤仁博士，同大学政策・メディア研究科特別研究助教 由良淳一博士には，本論文の執筆にあたってご助言を賜りましたことを感謝致します。

慶應義塾大学徳田研究室の諸先輩方には，折に触れ貴重な示唆や御指導を頂きました。特に，同大学政策・メディア研究科特別研究助教 伊藤昌毅博士には，本研究の草稿時から多くの助言と励ましを頂きました。

そして，研究の日々を共に過ごした伊藤友隆氏，生天目直哉氏，橋爪克弥氏に深く感謝します。また，論文執筆時に多大な協力をしてくださった徳田義幸氏，山本純平氏，米澤祐紀氏，西條晃平氏をはじめとした i 208 の皆様に深く感謝します。

最後に，長い研究生活を常に支え続けてくれた家族と，山内美樹氏に深く感謝します。

2010 年 2 月 12 日

河田 恭兵

参考文献

- [1] Pierre Wellner, Wendy Mackay, and Rich Gold. Back to the real world. *Commun. ACM*, Vol. 36, No. 7, pp. 24–26, 1993.
- [2] Yuji Ayatsuka. Fractal codes: Layered 2d codes with a self-similar layout. *Pervasive 2007 Advances in Pervasive Computing*, pp. 83–86, 5 2007.
- [3] Yuji Ayatsuka and Jun Rekimoto. Active cybercode: a directly controllable 2d code. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pp. 490–495, New York, NY, USA, 2006. ACM.
- [4] 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 Gellersen, et al. [26], pp. 190–207.
- [5] Takayasu Yamaguchi, Hiroshi Aono, and Sadayuki Hongo. Feature of signboard pictures using a mobile camera. In *IEICE Tech. Rep.*, PRMU2004-105, pp. 1–6, Fukui, Nov. 2004. Thu, Nov 18, 2004 - Fri, Nov 19 : (PRMU).
- [6] Yuji Ayatsuka, Nobuyuki Matsushita, and Jun Rekimoto. Gaze-link: a new metaphor of real-world oriented user interfaces. *Transactions of Information Processing Society of Japan*, Vol. 42, No. 6, pp. 1330–1337, 20010615.
- [7] Connor Dickie, Jamie Hart, Roel Vertegaal, and Alex Eiser. Lookpoint: an evaluation of eye input for hands-free switching of input devices between multiple computers. In *OZCHI '06: Proceedings of the 18th Australia conference on Computer-Human Interaction*, pp. 119–126, New York, NY, USA, 2006. ACM.
- [8] Fangmin Shi, Alastair Gale, and Kevin Purdy. Helping people with ict device control by eye gaze. pp. 480–487. 2006.
- [9] Jun'ichi Yura, Jin Nakazawa, and Hideyuki Tokuda. objsampler: A ubiquitous logging tool for recording encounters with real world objects. In *Proceedings of IEEE The 12th IEEE International Conference on Embedded and Real-Time Computing System and Applications*, pp. ppNA–ppNA, 8 2006.
- [10] Khoovirajsinh Seewoonauth, Enrico Rukzio, Robert Hardy, and Paul Holleis. Touch & connect and touch & select: interacting with a computer by touching it with a mobile

- phone. In *MobileHCI '09: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 1–9, New York, NY, USA, 2009. ACM.
- [11] Hyunglae Lee, Heeseok Jeong, JoongHo Lee, Ki-Won Yeom, Hyun-Jin Shin, and Ji-Hyung Park. Select-and-point: a novel interface for multi-device connection and control based on simple hand gestures. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pp. 3357–3362, New York, NY, USA, 2008. ACM.
 - [12] 暦本純一. Inforoom:実世界に拡張された直線操作環境. インタラクシオン 2000 論文集, pp. 9–16, 2000.
 - [13] Jun Rekimoto. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *10th annual ACM symposium on user interface software and technology (UIST)*, pp. 31–39, Banff, Canada, 1997.
 - [14] Hosei Matsuoka, Yusuke Nakashima, and Takeshi Yoshimura. Aerial acoustic communications in audible band — acoustic ofdm —. In *IEICE Tech. Rep.*, Vol. 106 of *EA2006-24*, pp. 25–29, Ibaraki, June 2006. Fri, Jun 23, 2006 : National Institute of Advanced Industrial Science and Technology (EA).
 - [15] Jun Rekimoto, Yuji Ayatsuka, Michimune Kohno, and Hauru Oba. Proximal interactions: A direct manipulation technique for wireless networking. In *In Proceedings of INTERACT2003, Sep.-Oct*, pp. 511–518. Press, 2003.
 - [16] Naokazu UCHIDA, Hiroki TOKIWA, Mai NISHI, Akira TAKAGI, Hideki ASOH, Masatomo HASHIMOTO, Akira MORI, Hideyuki NAKASHIMA, Yukihiro ITOH, Ichiro KOBAYASHI, and Kazuo YANA. Development of a speech dialogue interface system for home automation. *IEICE technical report. Natural language understanding and models of communication*, Vol. 104, No. 540, pp. 19–24, 20041215.
 - [17] Koji YATANI, Toshiya KISHIMURA, Koiti TAMURA, Masanori SUGIMOTO, and Hiromichi HASHIZUME. Toss-it : Intuitive information transfer techniques for mobile devices. *Technical report of IEICE. HIP*, Vol. 104, No. 169, pp. 19–24, 20040702.
 - [18] Peter Fröhlich, Rainer Simon, Lynne Baillie, Joi Roberts, and Roderick Murray-Smith. Mobile spatial interaction. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, pp. 2841–2844, New York, NY, USA, 2007. ACM.
 - [19] Hiroaki Kimura, Sota Matsuzawa, and Tatsuo Nakajima. Toward defining primitive accelerometer-based gesture interaction for casual applications. In *UCS*, pp. 51–57, 2009.
 - [20] Haruhisa Kato, Akio Yoneyama, and Yasuhiro Takishima. An intuitive interface based on camera parameters for portable devices. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Research posters*, p. 142, New York, NY, USA, 2006. ACM.
 - [21] M. Hachet, J. Pouderoux, and P. Guitton. A camera-based interface for interaction with mobile handheld computers, 2005.

- [22] Jingtao Wang, Shumin Zhai, and John Canny. Camera phone based motion sensing: interaction techniques, applications and performance study. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pp. 101–110, New York, NY, USA, 2006. ACM.
- [23] Apple Inc. *iPhone*. <http://www.apple.com/iphone/>.
- [24] 暦本純一, 塩野崎敦, 末吉隆彦, 味八木崇. PlaceEngine: 実世界集合知に基づく WiFi 位置情報基盤. インターネットコンファレンス 2006, pp. 95–104, 2006.
- [25] 伊藤誠悟, 吉田廣志, 河口信夫. locky.jp: 無線 lan を用いた位置情報・測位ポータル (テーマ: モバイルコンピューティング・一般). 情報処理学会研究報告. MBL, [モバイルコンピューティングとユビキタス通信研究会研究報告], Vol. 2005, No. 90, pp. 25–31, 20050915.
- [26] Hans-Werner Gellersen, Roy Want, and Albrecht Schmidt, editors. *Pervasive Computing, Third International Conference, PERVASIVE 2005, Munich, Germany, May 8-13, 2005, Proceedings*, Vol. 3468 of *Lecture Notes in Computer Science*. Springer, 2005.