

修士論文の和文要旨

研究科・専攻	大学院 情報システム学研究科情報メディアシステム学専攻 博士前期課程		
氏名	山野 真吾	学籍番号	1250044
論文題目	水面を利用したディスプレイの情報提示および認識手法		
要旨	<p>スマートフォンやタブレットなどの小型コンピュータは現在多くの人が保持し、常に持ち歩いている状況である。近年では、入浴時でもこれらの電子機器を持ち込み、動画や音楽鑑賞、ウェブページ閲覧、メールチェックなどを行う人が増えている。しかし、電子機器と水場である浴室は機器の故障を誘発させる危険性が高く相性が悪い。そのため、浴室環境でも安心して情報閲覧および操作を行えるシステムが必要であると考えられる。</p> <p>浴室環境を考慮したインタラクティブシステムとしてAquaTop Displayが挙げられる。このシステムは水面を入浴時の身体動作を利用してマルチユーザーかつマルチタッチ可能なウォーターサーフェスシステムである。このシステムで用いられるジェスチャは水特性を上手く利用しているが、ジェスチャ認識精度や情報提示手法に問題がある。そこで、これらの問題を解決し、より安定したジェスチャ認識精度、より入浴時に適した情報提示が可能なシステムを実装し評価することが我々の目的である。</p> <p>我々はまず、これらの問題点の原因を調査した。その後、ジェスチャ認識の解決案として、水面を認識することに特化したビジョンベースのライブラリ群であるAquaTop Core Vision Frameworkを実装し、フレームワークの有効性を検証した。また、浴室環境に適した情報提示の解決案として、浴槽内で動画および音楽鑑賞、ウェブページ閲覧、メール送信など従来のスマートフォンやタブレットで行っていることを水面上で可能にするAquaTop Multi Viewerを提案し実装した。さらに、認識精度を低下させる最大の原因である水面への操作によって発生する波ノイズの対処方法を提案し検証した。今後は提案したシステムを実際の浴室環境で利用し、実装したジェスチャ操作の有効性を検証したい。</p>		

平成25年度修士論文

水面を利用したディスプレイの情報提示および 認識手法

大学院情報システム学研究科 情報メディアシステム学専攻

学 籍 番 号 : 1250044

氏 名 : 山野 真吾

主任指導教員 : 小池 英樹 教授

指 導 教 員 : 野嶋 琢也 准教授

指 導 教 員 : 佐藤 俊治 准教授

提出年月日 : 平成26年2月21日

目次

第 1 章	序論	1
1.1	本研究の背景	1
1.2	本研究の目的	2
1.3	本研究の概要	2
1.4	本論文の構成	4
第 2 章	関連研究	5
2.1	水を利用したディスプレイシステム	5
2.2	浴室環境におけるインタラクティブシステム	13
第 3 章	AquaTop Display	19
3.1	システム概要	19
3.2	問題点と解決案	22
第 4 章	AquaTop Core Vision Framework	28
4.1	本フレームワークの概要	28
4.2	キャリブレーション	33
4.3	手指認識モジュール	34
4.3.1	深度情報の背景差分と 2 値化処理	34
4.3.2	画像フィルタ処理	35
4.3.3	領域抽出アルゴリズム	35
4.3.4	抽出領域の追跡アルゴリズム	36
4.4	ジェスチャの種類とその選別アルゴリズム	37
4.4.1	水面下から指を突き出すジェスチャ	37
4.4.2	水を流すジェスチャ	37

4.4.3	水を掬うジェスチャ	38
4.4.4	水面へのタッチジェスチャ	39
第 5 章	Application Framework	40
5.1	フレームワーク概要	40
5.2	AquaTop User Interface Objects 通信プロトコル	40
5.3	Awesomium HTML UI Engine	42
5.4	流体シミュレーション	43
第 6 章	応用例	47
6.1	AquaTop Multi Viewer	47
6.1.1	システム概要	48
6.1.2	ジェスチャと操作方法におけるインタラクション	52
6.2	AquaTop Shooting Game	64
6.2.1	システム概要	64
6.2.2	ジェスチャと攻撃方法におけるインタラクション	66
6.2.3	スピーカーを利用したフィードバック	67
6.3	大規模なゲームアプリケーションの試作	67
第 7 章	評価	70
7.1	波ノイズが認識システムに与える影響の調査	70
7.1.1	実験目的	70
7.1.2	実験方法	71
7.1.3	実験環境	72
7.1.4	実験結果	72
7.1.5	実験結果に対する考察	78
7.2	掬う動作の認識精度の調査	80
7.2.1	実験目的	80

7.2.2	実験方法	81
7.2.3	実験環境	81
7.2.4	実験結果	81
7.2.5	実験結果に対する考察	82
7.3	デモンストレーション展示から得られた知見	84
7.3.1	デスクトップアプリケーションの展示から得られた知見	84
7.3.2	ゲームアプリケーションの展示から得られた知見	87
7.3.3	得られた知見に対するまとめおよび考察	88
第 8 章	考察	89
8.1	現認識システムの考察と問題点	89
8.2	実際の浴室環境や公共のプールへの実使用に関する考察と今後の展望	92
第 9 章	結論	95
	参考文献	97
	謝辞	101
付録 A	発表文献・受賞業績	A-1
付録 B	展示業績	B-1

表 目 次

5.1 ATUIO の仕様	42
-------------------------	----

図目次

1.1	スマートフォン保持者の増加割合	3
1.2	入浴時におけるスマートフォン使用用途の調査	3
2.1	制御された水滴を利用した多層 2.5 次元ディスプレイ	6
2.2	AquaScript	7
2.3	水膜を利用したウォータースクリーン	8
2.4	TouchPond	9
2.5	Rainterior	10
2.6	FogScreen	11
2.7	多視点観察可能なフォグディスプレイ	12
2.8	Displair	13
2.9	多層構造によるフォグディスプレイ	14
2.10	Thermal Interactive Media	14
2.11	氷で作成されたマルチタッチスクリーン	15
2.12	浴槽内の物品と入浴者の行動を活用したシステム	16
2.13	浴槽内の水圧変化を利用したサウンドシステム	16
2.14	浴槽用の静電容量式タッチセンサ	17
2.15	Bathcratch	18
3.1	システムの構成図および外観	20
3.2	Xbox 360 Kinect	21
3.3	浴槽の代わりとして用いられたコンテナ	22
3.4	本システムで用いる FRP 製の浴槽	25
3.5	白濁剤の有無による深度画像の比較	26

3.6	Community Core Vision	27
4.1	AquaTop Core Vision のユーザーインターフェース	31
4.2	本フレームワークのフローダイアグラム	32
4.3	キャリブレーション方法	33
4.4	射影変換後の認識領域	34
4.5	画像フィルタ処理の過程	35
4.6	手領域の座標補正	37
4.7	掬うジェスチャの認識手法	38
4.8	掬うジェスチャのエラー処理方法	38
4.9	指座標とタッチ検出範囲	39
5.1	各フレームワークの対応関係	41
5.2	Awesomium HTML UI Engine の役割	44
5.3	WebCore と WebView の関係	44
5.4	オーバーレイされたイベント情報の例	45
5.5	流体情報とエフェクト生成方法	46
6.1	水環境特性を利用した操作面の方向	48
6.2	AquaTop Multi Viewer のフローダイアグラム	50
6.3	水平モードの画面表示と機能	51
6.4	自由モードの画面表示と機能	51
6.5	全画面モードの画面表示と機能	51
6.6	流すジェスチャによるコンテンツ切り替えの様子	53
6.7	タッチジェスチャによる水平モード用コンテンツ追加ボタン表示の 様子	53
6.8	タッチジェスチャによる水平モードから全画面モードへの移行の様子	53
6.9	突き出しジェスチャによるコンテンツ移動の様子	54

6.10	突き出しジェスチャによるコンテンツ回転および拡縮の様子	55
6.11	突き出しジェスチャによるコンテンツ消去の様子	55
6.12	タッチジェスチャによる自由モード用コンテンツ追加ボタン表示の 様子	56
6.13	流すジェスチャと流体シミュレーションによるコンテンツの移動の 様子	58
6.14	掬うジェスチャによるコンテンツ移動の様子	59
6.15	掬うジェスチャによるコンテンツ検索の様子	60
6.16	流すジェスチャによるページ送り操作の手法	63
6.17	流すジェスチャによるスクロール操作の手法	63
6.18	シューティングゲーム	64
6.19	シューティングゲームのフローダイアグラム	65
6.20	突き出しジェスチャによる魔法陣エフェクト	66
6.21	手を合わせるジェスチャによる火の玉エフェクト	67
6.22	フィードバック用スピーカーユニット	68
6.23	ビニールプールを用いた大規模ゲームの例	69
6.24	提案する解決案を用いた大規模ゲームの例	69
7.1	設定する閾値とジェスチャ認識範囲	70
7.2	色相値と深度情報の差分値の関係	72
7.3	突き出し動作直後5秒間の水面の状態	73
7.4	突き出し動作における経過時間と基準面からの距離の関係	73
7.5	右から左へ流す動作直後5秒間の水面の状態	74
7.6	右から左へ流す動作における経過時間と基準面からの距離の関係	74
7.7	左から右へ流す動作直後5秒間の水面の状態	75
7.8	左から右へ流す動作における経過時間と基準面からの距離の関係	75
7.9	掬う動作直後5秒間の水面の状態	76

.....

7.10	掬う動作における経過時間と基準面からの距離の関係	76
7.11	激しく水面を叩く動作直後5秒間の水面の状態	77
7.12	激しく水面を叩く動作における経過時間と基準面からの距離の関係	77
7.13	掬うジェスチャによる水面変化の様子	79
7.14	実験風景	82
7.15	画像の大きさにおける掬うジェスチャの成功率	83
7.16	デスクトップアプリケーションの展示の様子	84
7.17	移動, 回転, 拡縮ジェスチャの改良案	85
7.18	掬うジェスチャにおける手の形の例	86
7.19	ゲームアプリケーションの展示の様子	87
8.1	複数のPCを用いたシステムの構成	90

第 1 章

序論

本章では、本研究の社会的背景と目的について述べる。

1.1 本研究の背景

現在、コンピュータは小型かつ高性能化を続けている。その中でも、スマートフォンやタブレットは多くの人が保持する状況になっている。株式会社博報堂 DY ホールディングスのスマートフォン保持率調査（日本）によると 2013 年 11 月における所持率は 55.2%（内、男性：50.1%，女性：58.9%）と報告されている [1]。また、同調査結果より、図 1.1 に 2011 年 2 月から 2013 年 11 月までのスマートフォン保持者の増加割合を示す。図より、2013 年初旬には保持率は 45% 以上になり、2 人に 1 人はスマートフォンを保持していることが分かる。また、スマートフォンやタブレットのような電子機器は基本的に水没させると故障する。そのため、水場での利用は注意を払う必要がある。しかし、近年では入浴時にスマートフォンを持ち込む人が多くなっている。パナソニック株式会社とエコソリューション社の入浴目的・スタイルなどに関する実態調査によると、防水加工の有無に関わらずスマートフォンを入浴時に持ち込む人が多くなっていると報告されている [2]。持ち込み方法は、防水加工されていないスマートフォンの保持者はチャック式保存袋やビニール袋に入れて持ち込む。また、図 1.2 に入浴時におけるスマートフォンの使用用途を調査した結果を示す。調査結果によると、入浴時にスマートフォンを利用

して Web ページ閲覧，動画および音楽鑑賞，メールを行っているもしくは行いたいことが分かる。しかし，防水加工されていないスマートフォンを入浴時に使用する不安として，先述したとおり水没や濡れることによる機器の故障が挙げれる。したがって，浴室環境での情報閲覧および操作に適したインターフェースやシステムを構築することで，入浴時でも安心してスマートフォンと同じような利用方法が可能になる。

1.2 本研究の目的

本研究の目的は，浴室環境での情報閲覧および操作を入浴時における身体動作を利用して可能にするシステムを構築し検証することである。さらに，1.1 章で述べた調査結果を基に，提示する情報（コンテンツ）とその情報を操作する適切な操作方法の実現も目指す。また，提案する操作方法についても，操作性や精度を検証し，入浴時における提案手法の有効性を調査する。

1.3 本研究の概要

本研究では，浴槽に白濁剤を融解させた水を張り，浴槽を上部に取り付けたカメラにより水面付近の手指の状態と連動した映像を投影するインタラクティブなディスプレイシステムである AquaTop Display システムの問題点を示し，解決案として新しいシステムを一から構築した。前システムの認識部分は深度カメラを用いた一般的な手法で手指認識を行っていたが，本システムでは水面環境に適した認識手法を提案し構築を行った。さらに，浴室環境でのコンピュータ操作と情報閲覧をより水特有な環境に適応するように投影する映像に対する画像効果を提案し実装を行った。また，前システムのジェスチャ誤認識の原因を視覚的に表示し，ジェスチャ認識精度を評価した。さらに，学会やテレビ番組での展示を通して得られた知見から，提案したシステムの有効性を考察した。

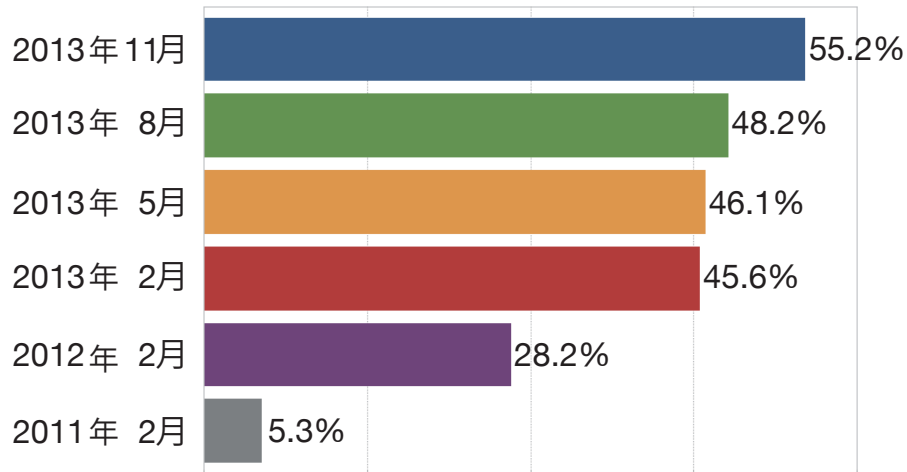


図 1.1: スマートフォン保持者の増加割合

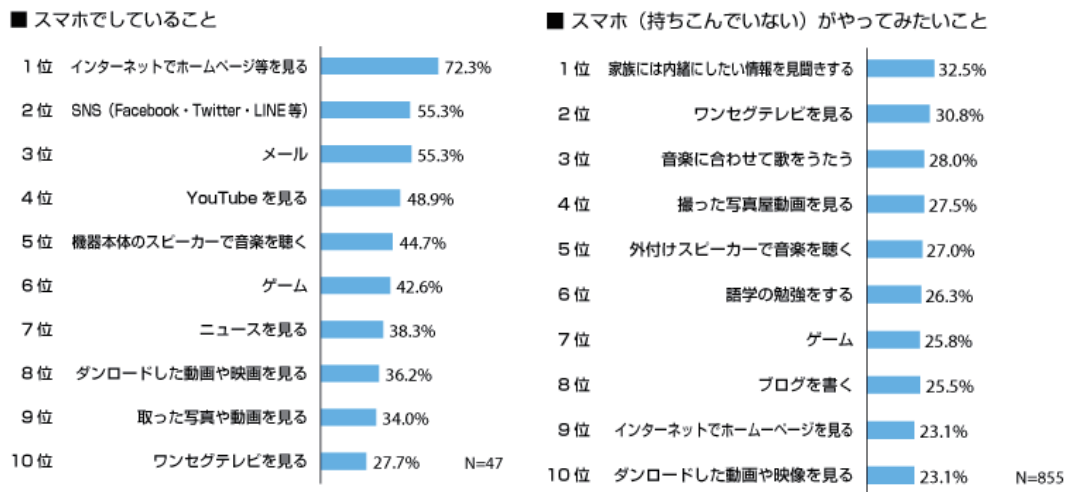


図 1.2: 入浴時におけるスマートフォン使用用途の調査

1.4 本論文の構成

本章では、研究背景として入浴時における電子機器の利用目的や利用方法について言及し、本研究の概要と目的について述べた。

2章では、水を利用したディスプレイ、浴槽環境を利用したインタラクティブシステムについて関連研究を述べる。

3章では、本研究の前システムである AquaTop Display の概要と問題点を説明し、それに対する本研究の解決案と立ち位置を述べる。

4章では、本研究で提案する認識システムの構成や実装手法について述べる。

5章では、認識システムで使用する独自通信プロトコルや、水面ディスプレイをより効果的にするために利用したフレームワークについて述べる。

6章では、本認識システムを利用した応用例について述べる。

7章では、本認識システムの評価とその結果や、実際の展示によって得られた知見について述べる。

8章では、3章で示す AquaTop Display の問題点に対する解決案の達成度と本システムを実環境で利用するための考察を述べる。

9章では、本研究全体のまとめを述べる。

第 2 章

関連研究

本章では本研究と関連のあるシステムとして、水を利用したディスプレイシステム、浴室環境におけるインタラクティブシステムについて述べる。

2.1 水を利用したディスプレイシステム

水をディスプレイシステムに利用する研究が多く行われている。水をそのままスクリーンとして利用するウォータースクリーンや、霧状に加工して利用するフォグディスプレイなどである。

ウォータースクリーン

Barnum らはソレノイドバルブにより精密制御された多支管バルブ機構を複数並べ層状にし、この機構により生成された水滴をカメラでリアルタイムにトラッキングし色を投影する A Multi-Layered Display with Water Drops を開発した (図 2.1) [3]。この多支管バルブ機構は直線は 50 個の噴出孔があり、各層の水滴はカメラで見た時に重ならないように制御されている。つまり、カメラからは各層の全ての水滴の位置が取得可能であり、それらに対して適切にプロジェクションを行うことを実現している。しかし、このシステムは水滴を制御する機構が高度な点や、水滴間の隙間が要因でディスプレイとしての解像度が低いという問題点がある。

Popp はソフトウェアでバルブを制御し、滝のように落ちる水滴で文字を空中に

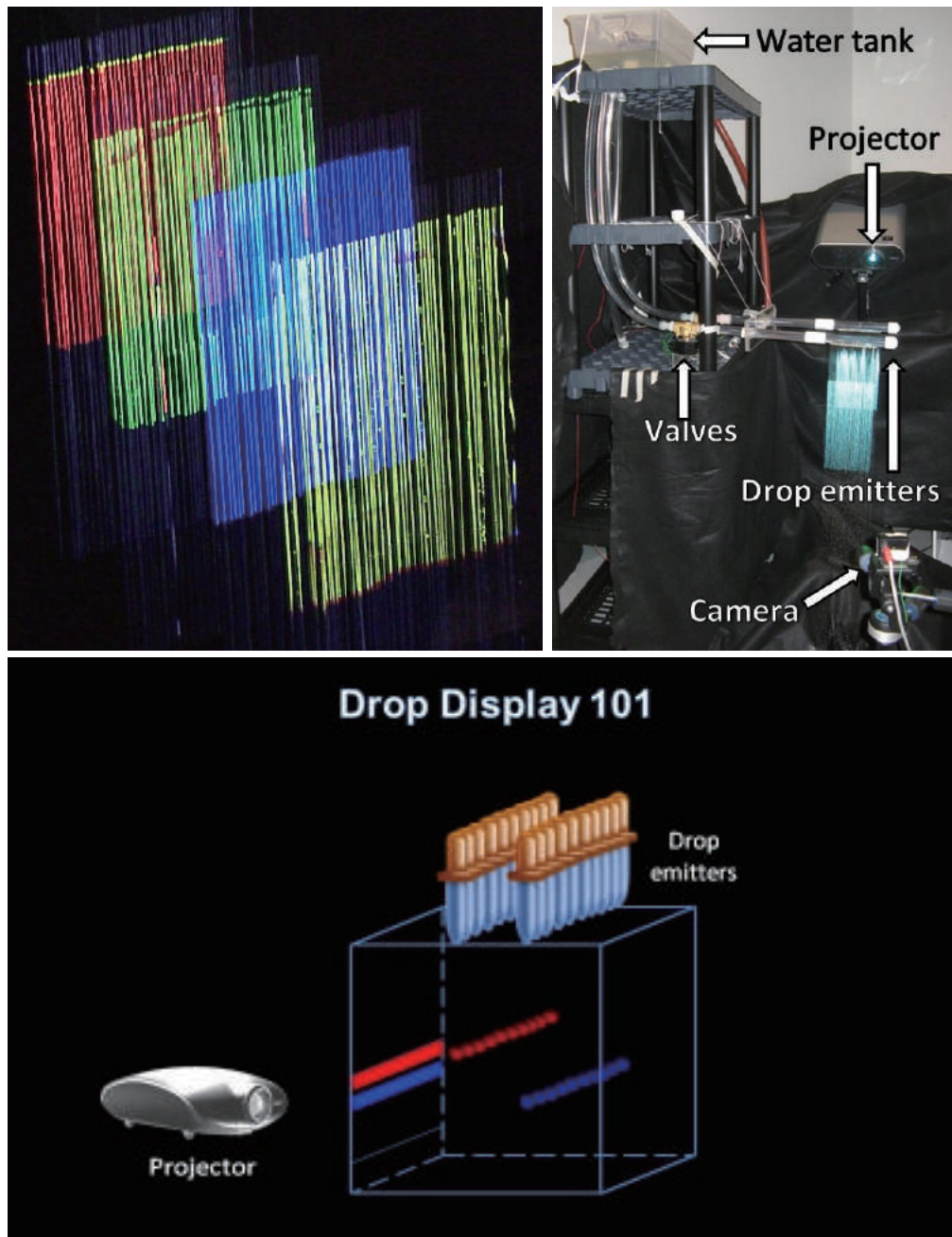


図 2.1: 制御された水滴を利用した多層 2.5 次元ディスプレイ



図 2.2: AquaScript

表示する AquaScript を開発した (図 2.2) [4]. このシステムはあらかじめ設定した文字列の形になるように水滴 1 滴 1 滴の動きを制御している. さらに, 横サイズの延長が容易で最長で 50 メートルの装置が制作可能である. しかし, このシステムはあらかじめ設定した文字列しか表示できないためインタラクティブ性に欠ける問題点がある.

杉原らは流水を円形平板に向けて放水することでドーム状の水膜を形成し, その水膜をスクリーンとして利用するシステムを開発した (図 2.3) [5]. これはアートおよび学術的視点から, 水膜で構成したドーム空間を新しい没入環境として提案するシステムである. 水円形平板に対して毎分 18L の流水が放水することでドーム状に形成された水膜を構成している. この水膜は内部に中空の空間を有しており, その中に頭部を入れ, そこから水膜に投影される映像を閲覧することで普段得られない没入感や臨場感を体験者に与えることが可能になっている.

Dietz らは水面へのタッチセンシングを可能し, 浅い水槽に投影される映像とタッチ操作が同期する TouchPond を開発した (図 2.4) [6]. 映像投影は水槽底に半透

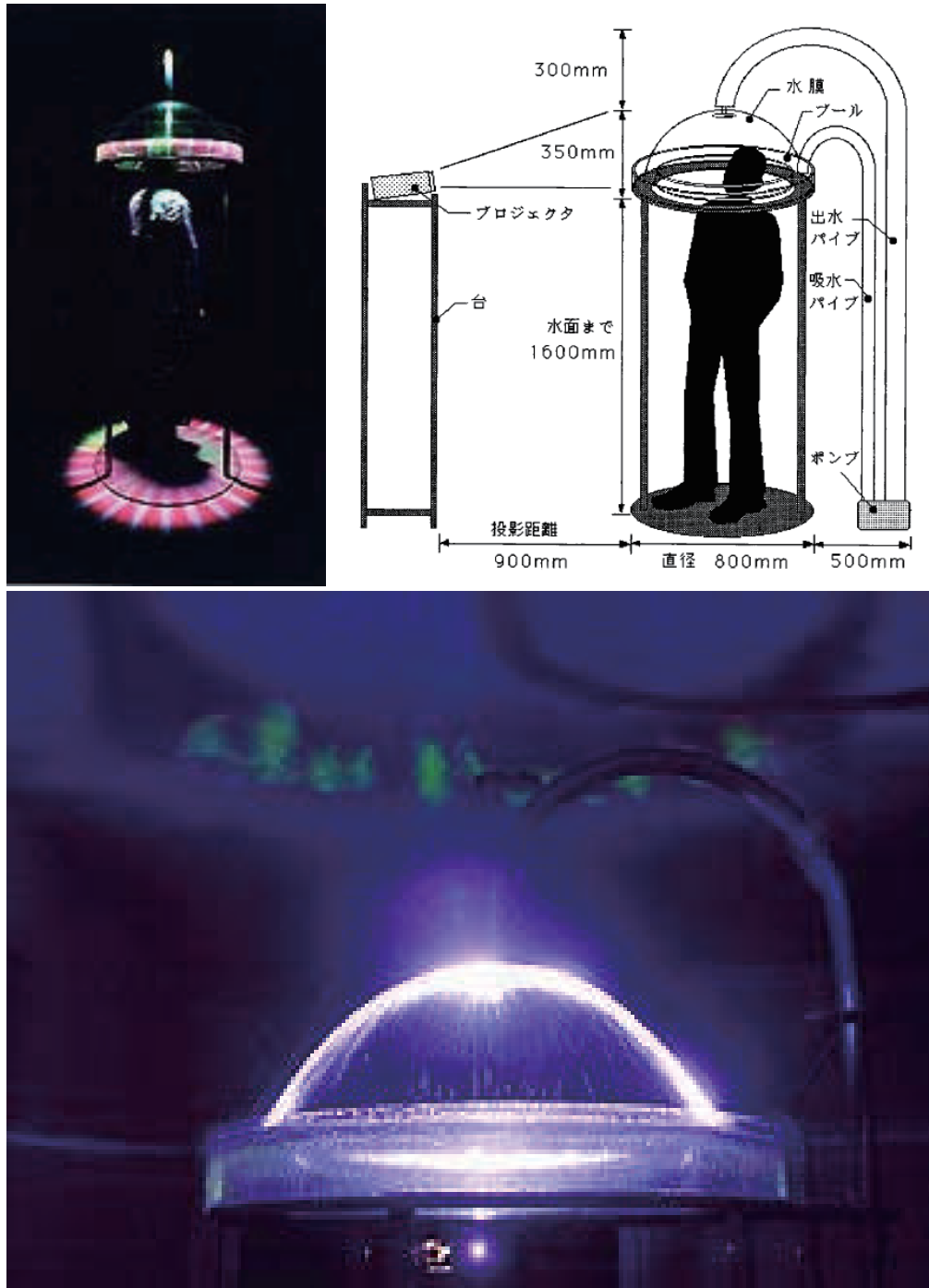


図 2.3: 水膜を利用したウォータースクリーン



図 2.4: TouchPond

明のシートを引き、それに向けてバックプロジェクションを行っている。水面へのセンシングは赤外線 LED 光源と赤外線カメラを用いている。水槽の横側から赤外線照射すると、その赤外線は外に漏れ出さず水槽底と水面の間で全反射する。この状態で水面にタッチすることで、接触面の反射率が変化し、全反射していた状態が崩れ接触面と反対方向に赤外線が漏れる。この漏れた赤外線の位置を水槽の下に設置した赤外線カメラで取得することで水面へのタッチを可能にしている。このセンシング方法は Han が開発した Frustrated Total Internal Reflection (FTIR) の技術を応用している [7]。このシステムの問題点は、水面へのインタラクションはタッチのみであり、水の特性を上手く利用した操作になっていないことである。

奥出らは水面に落ちる雨粒の位置を FTIR 技術によってセンシングし、水槽底に映像を投影する Rainterior を開発した (図 2.5) [8]。システム構成は先述した TouchPond と同様であるが、このシステムの目的は、水の持つ癒やし効果を雨粒とそれに同期した映像によって拡張し雨の日の不快感を軽減することである。

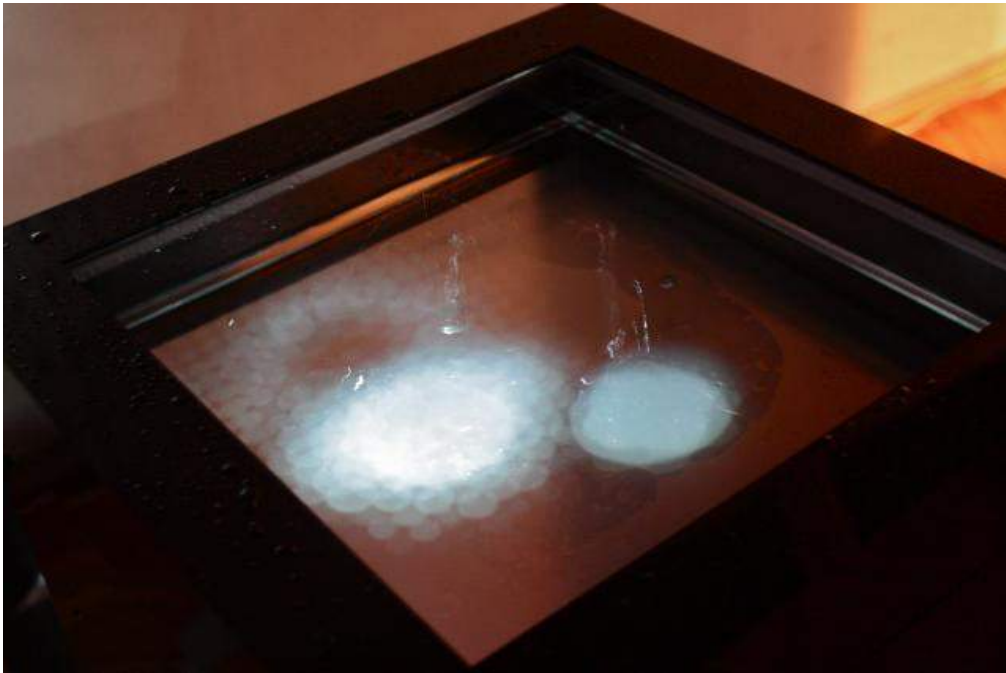


図 2.5: Rainterior

フォグディスプレイ

Rakkolainenらは霧状にした水に映像を投影することで、投影した映像が空中に浮いてる様に見える FogScreen を開発した (図 2.6) [9]. このシステムの最大の特徴は霧状にしたディスプレイ内を歩くことができる点であり、偏光レンズ、シャッターグラスを用いることで立体視も可能になっている。しかし、問題点としてディスプレイへの操作は専用のポインタを用いなければならず、空間をディスプレイにした利点を上手く利用できていない。

八木らは霧による光の散乱が指向性を持つことを利用し、円筒状に放出した霧に対して多方向から異なる映像を複数のプロジェクタにより投影することにより、視点位置に応じた映像を提示可能な多視点観察可能なフォグディスプレイを開発した (図 2.7) [10].

Displair社は紙をかざしても濡れない程の水粒子を噴射して形成した霧状のカーテンに映像を投影するシステムを開発した (図 2.8) [11]. このシステムの利点は



図 2.6: FogScreen

霧を生成機器が小型で扱いが容易なことや、霧状のカーテンに対して特別な機器を使用せず、マルチユーザーかつマルチタッチでの操作が可能なことである。

遠藤らは、多層の霧をフォグディスプレイとして利用することで、自身の存在する空間自体をディスプレイとする無境界ディスプレイの実現を目指し、フォグディスプレイで体験者の周囲を囲む構造の3次元的ディスプレイを開発した(図 2.9) [12]。フォグディスプレイは一般的なディスプレイで体験者を囲むのに比べて、空中に投影している映像が浮かんでいるように見えるなどの高臨場感をより得られるようになっている。

Mineらは水温変化に応じて投影している映像が変化するシステムである Thermal Interactive Mediaを開発した(図 2.10) [13]。これは図中央にある水槽に水が張っており、水槽下部からリアプロジェクション方式で映像が投影されている。さらに、水槽上部には遠赤外線カメラが設置されており水槽の水温分布が取得可能になっている。また、水槽を挟むように水を汲むことが可能な桶があり、水槽端で高温、低温それぞれの水を汲むことが可能になっている。その他の機能としては、



図 2.7: 多視点観察可能なフォグディスプレイ

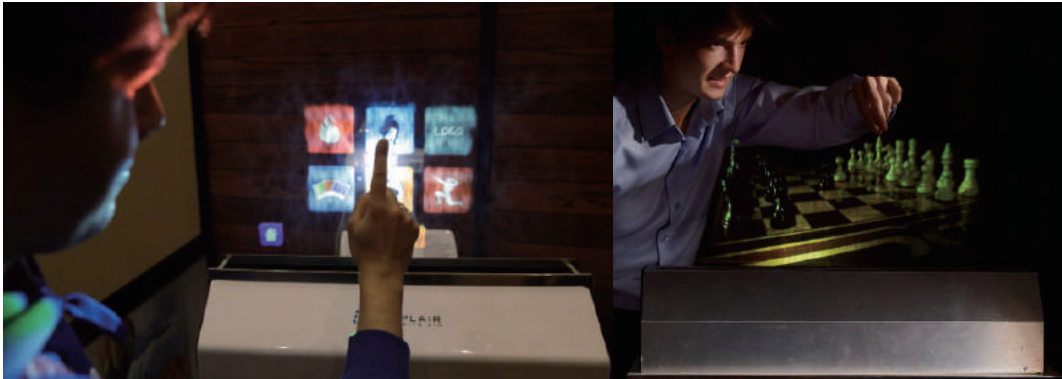


図 2.8: Displair

高温の水を柄杓で注ぐと注いだ領域が赤く表示されるなどの注ぎ動作から生じる水温変化を表示可能にしている。

アイススクリーン

Virolainenらは寸法が50x50x25cmの正方形の水をセンサーで薄く切り、横幅2m、高さ1.5mのアイススクリーンを作成し、それをマルチタッチを可能にする研究を行っている(図2.11) [14]。このシステムではRDIによってマルチタッチを実現しており、タッチした位置に応じてプロジェクションを行っている。

2.2 浴室環境におけるインタラクティブシステム

電子機器は水に弱い点から浴室環境での使用には向かず、入浴時に利用可能なデジタルコンテンツは少ない。この問題を解決するために、浴室環境下に適した入力インターフェースや入浴行為にエンタテインメント性を付加し入浴の楽しさを拡張する研究が行われている。

平井らは浴室内の様々な物品にRFIDタグを付与することでそれらの位置を計測し、利用された物品と入浴者の行動、状態を測定するシステムを開発した [15]。彼らのプロトタイプアプリケーションでは入浴者が任意の物品を利用するたびに



図 2.9: 多層構造によるフォグディスプレイ



図 2.10: Thermal Interactive Media



図 2.11: 氷で作成されたマルチタッチスクリーン

サウンドがインタラクティブに変化し、従来の入浴行動より楽しむことを可能にしている (図 2.12)。さらに、浴槽内の湯水を介して入浴者の状態をセンシングし、その状態を効果音や音楽で表現も行っている。このシステムでは、入浴者が湯水をかき回すなどの動作を行った時には浴槽内の水位や水面の揺れの変化を浴槽と接続された給湯器に内蔵された水圧センサーにより浴槽内の水位や湯水の水面変化を計測し、その結果に基づいてサウンドを生成し入浴自体の楽しさを向上させている (図 2.13)。なお、入浴者が安静状態の場合は、湯水を介した心拍、呼吸の計測結果を利用している。

榊原らは浴槽の裏側に静電容量式のタッチセンサを埋め込み、浴槽縁を操作可能にするインターフェースである TubTouch を開発した (図 2.14) [16]。このインターフェースは手指による近接操作、タッチ操作、スライド操作が可能で浴槽縁に表示するメニュー画面は天井に設置したプロジェクタから投影している。入浴時に手を置くことの多い浴槽縁を入力機器にすることにより、入浴時の自然な身体動作で操作することが可能になっている。

平井らは浴槽の縁を擦ることでDJのスクラッチ演奏が楽しめるエンタテインメントシステムである Bathcratch を開発した (図 2.15) [17]。これは、浴槽裏側に設置

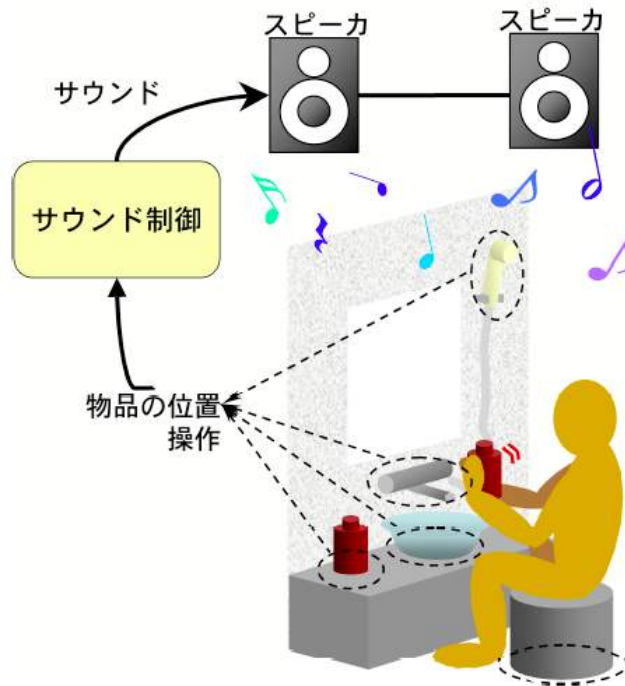


図 2.12: 浴槽内の物品と入浴者の行動を活用したシステム

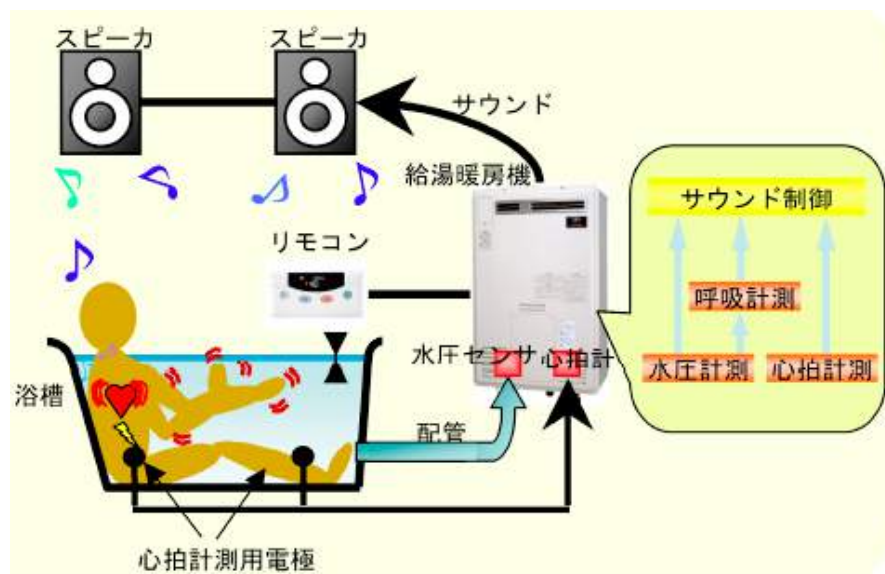


図 2.13: 浴槽内の水圧変化を利用したサウンドシステム

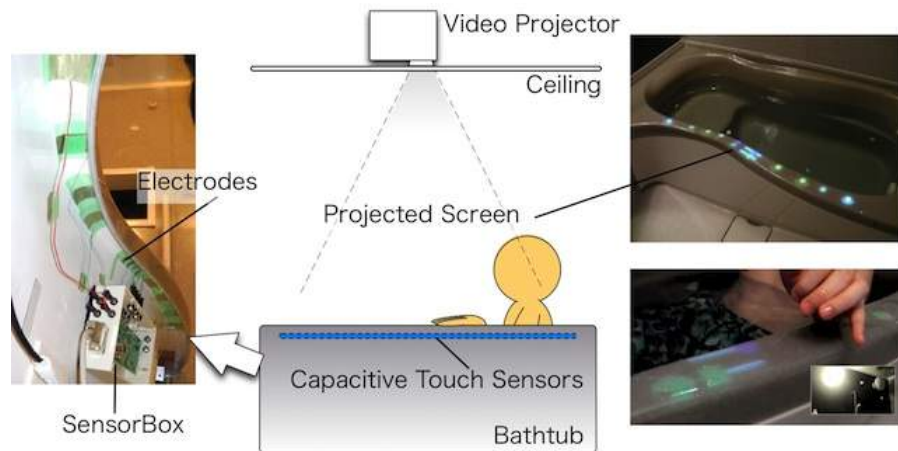


図 2.14: 浴槽用の静電容量式タッチセンサ

した piezo センサで擦った音を音響信号として取り込み、音の特徴を解析することで擦ったタイミングを検出し、それに合わせてスクラッチ音が鳴り、反対側の浴槽縁に設置された静電容量型タッチセンサを操作してスクラッチ音の種類やリズムトラックなどの切り替える。これにより、入浴時に音楽演奏が可能になり、入浴の楽しさを拡張している。なお、タッチセンサは先述した TubTouch の技術を利用している。

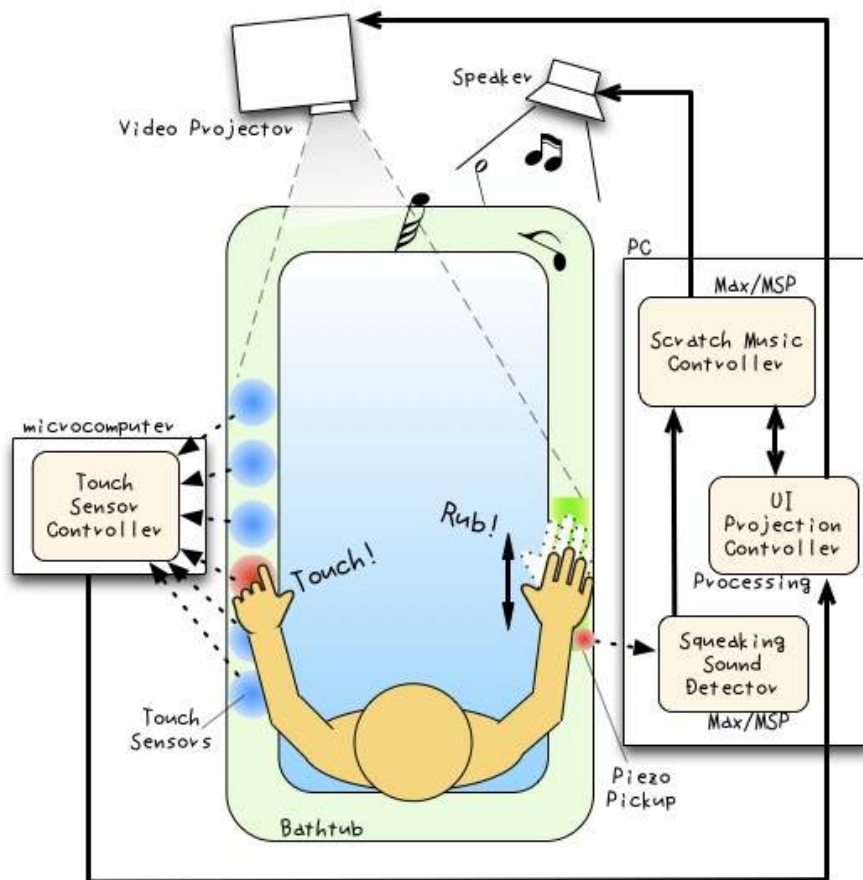


図 2.15: Bathcratch

第 3 章

AquaTop Display

本章では、水面をインタラクティブサーフェスとして利用可能にする AquaTop Display について述べる。

3.1 システム概要

AquaTop Display は高橋らによって研究開発された水面をマルチユーザーかつマルチタッチ可能なインタラクティブサーフェスとして用いることができる技術である [18][19][20][21]。AquaTop Display で用いられるハードウェアは下記のとおりであり、ハードウェア構成とシステム外観を図 3.1 に示す。

演算処理用 PC

Apple 社製の MacBookPro 15-inch, Late 2011 (CPU : Intel Core i7 2.4GHz, RAM : 8GB, GPU : AMD Radeon HD 6770M, OS : Windows 7) が用いられている。

深度カメラ

Microsoft 社製の Xbox 360 Kinect (以下, Kinect) を用いられている (図 3.2)。Kinect は同社のゲーム機である Xbox 360 向けに開発されたゲームデバイスである。このデバイスは、RGB カメラ、深度センサー (赤外線ランダムドットパターンプロジェクタと赤外線カメラ) が搭載されている。さらに、マルチレイマイクロフォンや専用ソフトウェアと動作させるプロセッサを内蔵し

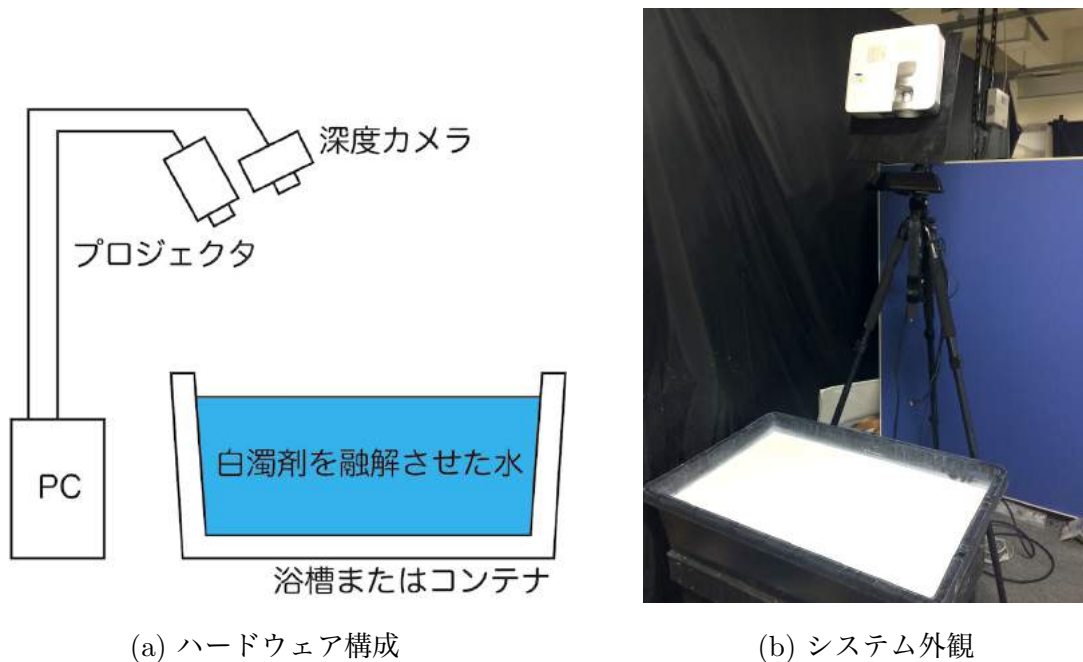


図 3.1: システムの構成図および外観

たセンサーを有しており、プレイヤーの位置、動き、声、顔を認識することが可能である。本システムではRGBカメラと深度センサーのみを用いる。深度センサーは赤外線ランダムドットパターンプロジェクタで照射されたドットパターンを赤外線カメラで撮影することで、対象の形状変化をそのパターン変化から取得し、この変化を演算することにより深度情報を取得している。なお、Kinectで取得できるRGB画像および深度画像の解像度は640x480pxでありフレームレートは30fpsである。

プロジェクタ

EPSON社製のEB-1775W（投影方式：3LCD、解像度：1280x768px、輝度：3000lm）が用いられている。

コンテナ

寸法が縦550mm、横850mm、深さ200mmのプラスチック製コンテナが用いられている（図3.3）。



図 3.2: Xbox 360 Kinect

白濁剤

大日精化工業株式会社の WMP478 ホワイトという水溶性高分子と界面活性剤で入浴剤用に被覆処理を施したルチル型酸化チタンの水性粉末着色剤が白濁剤として用いられている [22].

手指およびジェスチャ認識ソフトウェアとアプリケーション（動画像閲覧およびシューティングゲーム）は、openFrameworks が用いられ C++ で開発されている [23]. openFrameworks はインタラクティブなコンテンツやメディアアートを作成するために特化された C++ のフレームワークであり、Mac OS X, Windows, Linux, iOS, Android, ARM で動作させることが可能である。このフレームワークの特徴は多くのオープンソースライブラリを整理された形で提供している。例えば、グラフィック描写には OpenGL, オーディオ入出力には RtAudio, 画像入出力には FreeImage などである。そのため、複数のライブラリ機能を繋ぎ合わせる事が容易で少ないコード数で高性能なソフトウェアを作成することができる。さらに、公開されている様々なアドオンを使用することで openFrameworks に本来無い機能も随時追加が可能になっている。

深度画像処理では、深度カメラから深度画像取得のために PrimeSense 社が開発している OpenNI が用いられている [24]. また、深度画像から手指およびジェス



図 3.3: 浴槽の代わりとして用いられたコンテナ

チャ認識を行うために OpenCV を利用している [25]. OpenCV は Intel 社および Willow Garage 社によって開発されているクロスプラットフォームの画像処理ライブラリである.

3.2 問題点と解決案

高橋らの手指およびジェスチャ認識システムおよびアプリケーションの問題点は4点ある.

システム起動後のパラメータ調整

認識システムのパラメータ制御がアプリケーション毎に異なり, それを制御するユーザーインターフェースが存在しない点と認識システムがアプリケーションに内蔵される形をとっている点である. 必要最低限のパラメータはキーボード操作によって調整できるが, 実用性は低くシステム起動中のパラメータの再調整は困難になっている. さらに, キャリブレーションを行う時には別のアプリケーションを立ち上げ, 必要な設定ファイルを生成しなければならない. 例えば, カメラや浴槽およびコンテナに人がぶつかるなどの事故で設置位置が変更された場合, システムをシャットダウン, 新しい設定ファ

イルを生成，システムの再起動という手順を踏まなくてはならず操作性に欠ける．解決案として認識システムとアプリケーションを分離し，認識状況を常に確認できる状況を構築し，システム起動後であっても深度カメラで取得する範囲や取得した深度画像に適応する画像フィルタのパラメータをスライダーやボタンを用いて調整可能にする．さらに，キャリブレーション機能を認識システムに内蔵することで不測の事故でカメラや浴槽およびコンテナの設置位置が変更されてもシステムをシャットダウンすることなく再設定を行えるようにする．

システム動作速度

認識システムとアプリケーションを同時に動作させ，これらの処理を同一フレーム間で行っているため，システム全体の平均フレームレートが20fps前後と低い点である．低フレームレートはジェスチャによる操作の使用感に影響を及ぼす可能性があるため，最低でもシステム全体のフレームレートは30fps以上が望ましいと考えられる．30fps以上が望ましい理由は，使用している深度カメラの最大フレームレートが30fpsだからである．解決案として，トラッキングアルゴリズムやプログラム全体の質を改善する．さらに，認識システムとアプリケーションを別々にPCで動作可能し，演算処理自体の負荷を減らすことでシステム動作を向上させる．またこれにより，アプリケーション毎に認識システムを開発する必要性がなくなり，認識システムの汎用性も向上させることができる．

ジェスチャ制御

タッチおよび掬うジェスチャの認識精度が良くない点である．タッチジェスチャに関しては，指認識自体は良好であるが，ジェスチャのイベント処理が甘く，ユーザーがタッチを行いたい位置に指を移動する前にタッチイベントが発生してしまう問題点がある．掬うジェスチャに関しては，ジェスチャ認識

自体の精度やジェスチャ状態を維持する制御が甘く、掬うことに成功したとしても掬った状態で移動を行うとすぐに掬ったコンテンツが落下してしまう問題点がある。解決案として、タッチや掬うジェスチャのイベントトリガーやイベント発生中の処理を見直す。また、ジェスチャ直後の画像に適応する画像エフェクトも最適なものを検証し開発する。

表示コンテンツ不足

開発されたアプリケーションの1つであるウォータートップは水の特性を上手く利用しているが、表示可能なコンテンツは動画像のみと実際の浴槽での使用には乏しい点である。また、システム起動後は新しいコンテンツを追加できないなどの問題点もある。解決案として、コンテンツの種類を増加し、一部コンテンツを Web 上から取得することでコンテンツ不足を解消する。さらに、従来システムではあらかじめロードしなければならなかった動画などもストリーミング可能にすることで、ロード時に発生するシステムのラグを可能な限り低減させる。

なお、提案する解決案の詳細は4章にて述べる。

本研究ではこれらの問題を技術的に解決することを目的とする。提案するシステムで使用するハードウェアとソフトウェアを次のとおりである。演算処理用 PC、深度カメラ、白濁剤は高橋らが使用したものと同様のものを使用する。ただし、演算処理用 PC の OS は Windows7 は使用せず OS X 10.8.4 を用いる。プロジェクタは BenQ 社製の W1070 (投影方式: DLP, 解像度: 1920x1080px, 輝度: 2000lm) を用いている。浴槽は高橋らが用いた図 3.3 で示したコンテナと図 3.4 で示す寸法が縦 60mm から 74mm (浴槽縁が円曲する地点の最小および最大距離), 横 1500mm, 深さ 400mm の FRP 製の浴槽を用いる。また、高橋らの論文では言及されなかった水に白濁剤を融解させることで、水面が深度カメラで取得できる理由を次に示す。図 3.5 は半透明なコンテナに水を張り、白濁剤の有無で深度カメラから取得で



図 3.4: 本システムで用いる FRP 製の浴槽

きる深度情報の比較を視覚的に表したものである。深度画像のカラースケーリングは、HSV モデルの色相値を深度データ値で正規化をしている。具体的には、コンテナ周辺の床の深度値を色相値の 120、コンテナの縁の深度値を色相値の 0 に割り当てている。図からわかるように、白濁剤無しの図 3.5a はコンテナ内外で深度値が近くなっており、透明な水では深度カメラの赤外線パターンがコンテナの底まで貫通することがわかる。一方、白濁剤有りの図 3.5b はコンテナ内部の色がコンテナの縁の色に近いことから、深度カメラの赤外線パターンが水面で反射していることがわかる。したがって、白濁水を用いることで水面状態を深度カメラで取得できることがわかる。

提案するシステムは NUI Group Community によって開発された Community Core Vision (CCV) を基に水面を利用したディスプレイ向けに改良および最適化を行った [26]、CCV は FTIR, DI, DSI 向けに開発されたクロスプラットフォームのマルチタッチ認識アプリケーションである。図 3.6 に CCV のユーザーインターフェースを示す。特徴としては、図からわかるように入力画像に対して平滑化、ハイパスフィルタ等の画像フィルタをスライドバーとボタンで容易に調整可能な点である。本認識システムでも CCV のように画像フィルタやトラッキングパラメータを容易に調整可能にしている。

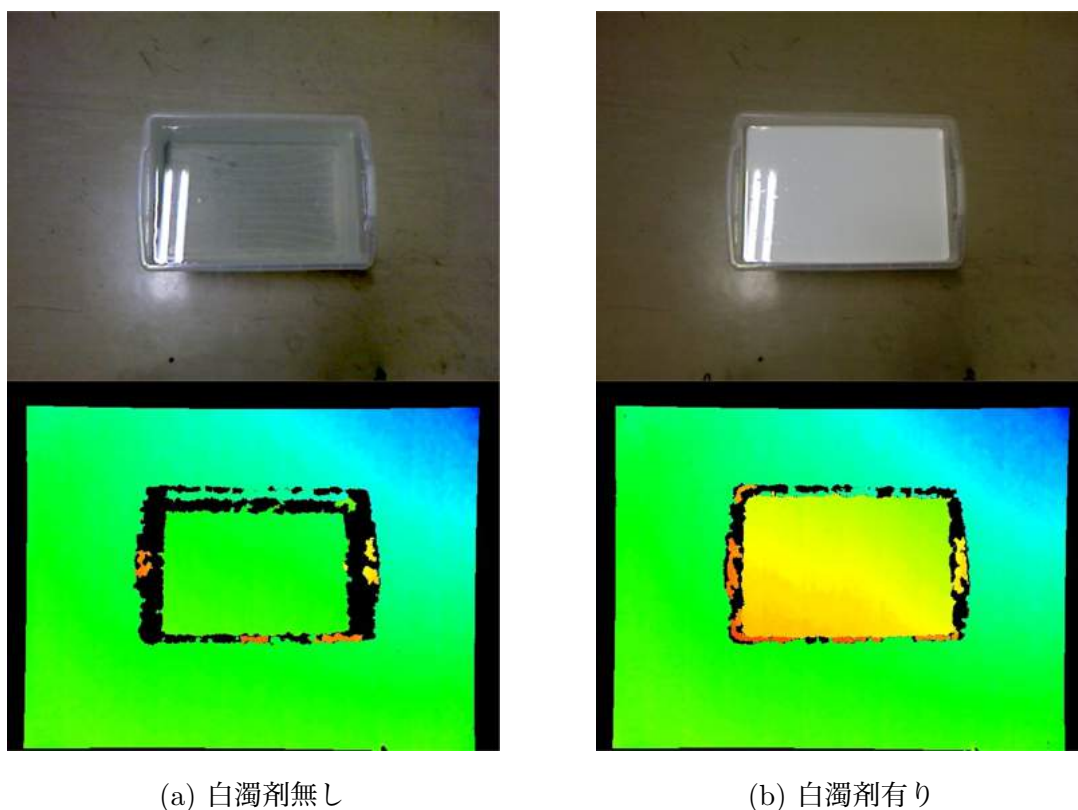


図 3.5: 白濁剤の有無による深度画像の比較

システム実装は高橋らのシステムと同様に openFrameworks を用いて C++ で開発を行った。

深度画像処理では、深度カメラから深度画像と RGB 画像の取得を行うため openFrameworks のアドオンである ofxKinect を用いた。ofxKinect は openFrameworks のオープンソースコミュニティである OpenKinect が開発をしている libfreenect ライブラリを組み込み済みプロジェクトテンプレートにしたものである [27]。なお、OpenNI は使用していない。また、取得した深度画像に適応する画像フィルタや輪郭抽出のために OpenCV を用いている。

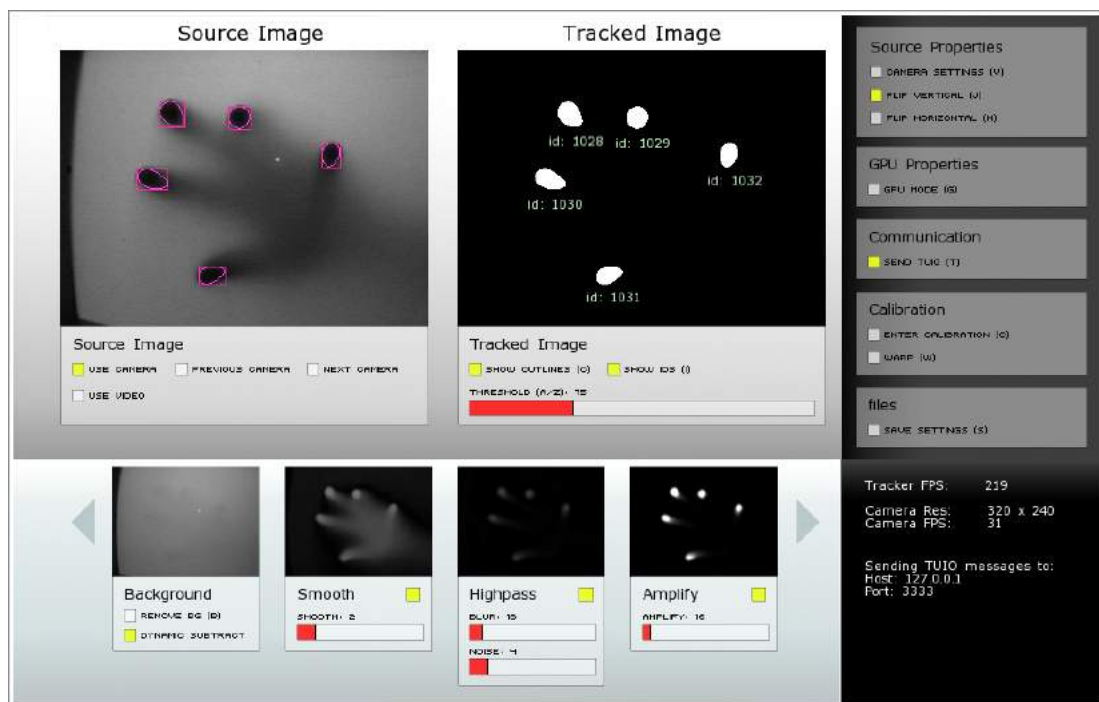


図 3.6: Community Core Vision

第 4 章

AquaTop Core Vision Framework

本章では、本研究で提案する手指およびジェスチャ認識フレームワークである AquaTop Core Vision Framework の実装について述べる。

4.1 本フレームワークの概要

本フレームワークは openFrameworks 用アドオンとして開発した。ユーザーインターフェースを図 4.1 に示す。本認識フレームワークのユーザーインターフェースは現在の認識状況、取得する深度カメラの設定、トラッキングデータに適応する画像フィルタの設定を詳細に行うためにボタンやスライダーを複数搭載している。つまり、システム起動後でも認識パラメータを調整可能である。搭載している機能は次のとおりである。

① Source Image

深度カメラから取得した深度画像と RGB 画像を表示する。RGB 画像は⑥で述べるボタンによって表示有無を変更可能で、表示する場合は RGB 画像に深度画像をオーバーレイした映像が描写される。原則、この画像内にプロジェクタにより投影されている映像を収めなければならない。

② Homography Area

手指トラッキングの有効範囲を設定する。有効範囲は四隅の小矩形で囲まれた領域内側である。なお、各小矩形の位置はドラッグアンドドロップで移動

可能である。また、小矩形の4点の座標を基にカメラ座標からプロジェクション座標への座標変換が行われる。

③ Camera Settings

取得したカメラ画像の水平および垂直反転の切り替えを行うボタン。このボタンにより深度カメラを設置する位置や方向に囚われない手指トラッキングが可能になっている。また、ボタンによりカメラ画像を切り替えた場合は、手指トラッキングの有効範囲を再度設定する必要がある。

④ Tracking

トラッキング対象を設定するトグルボタンで、Blobsが水面下から突き出した指と手、Fingersが水面上の指、Objectsがオブジェクト認識に対応している。なお、FingersトラッキングはBlobsのトグルボタンが有効になっている場合のみ行われる。

⑤ Other Properties

5つのボタンがあり、Show RGBImageはSource Imageの位置へのカラー画像をオーバーレイ表示有無、Show IDはセッションIDの表示有無、Draw Outlineは認識領域に対する最小の矩形描写の有無の切り替えを行い、Remove BGは背景差分情報の更新、Save Settingsは現在の設定をXMLファイルに書き出す。また、書き出した設定ファイルを認識システムへドラッグアンドドロップすることにより、その設定を適応することが可能である。

⑥ Debug Information

現在のフレームレートや各種トラッキング数などを表示する。

⑦ Tracking Below Image

水面下から突き出した指のトラッキング画像を表示する。なお、認識位置は赤点として描写される。

⑧ Tracking Above Image

手と水面上の指のトラッキング画像を表示する。認識した領域を前者は緑枠、後者は緑点として描写する。また、4.4章で後述するジェスチャ別に認識した領域を囲む枠色や点色が変化する。タッチジェスチャは赤点、掬うジェスチャは青枠に変化する。

⑨ Below Image Slider

水面下から突き出した指のトラッキング画像に適応する画像フィルタのパラメータを調整する。調整可能なパラメータは上から認識可能な深度と面積の範囲設定、ノイズ除去の強度、認識した領域の平滑化度合いである。

⑩ Above Image Slider

手と水面上の指のトラッキング画像に適応する画像フィルタのパラメータを調整する。調整可能なパラメータの内、上から4つは Below Image Slider と同様であり、最下部のスライダーは指をタッチジェスチャとして認識する深度範囲である。

また、その他の設定パラメータは別途用意されている XML の設定ファイルに直接記述することで行う。この設定ファイルの読み込むはシステム起動を行った時のみ実行される。

図 4.2 に本フレームワークのフローダイアグラムを示す。本フレームワークでは、手指領域のトラッキング部とジェスチャ認識部に分かれている。前者は深度カメラから取得から取得画像の領域分割までを行う。後者は分割された領域群とジェスチャの紐付け、ジェスチャ別にノイズ処理、認識情報の送信を行う。

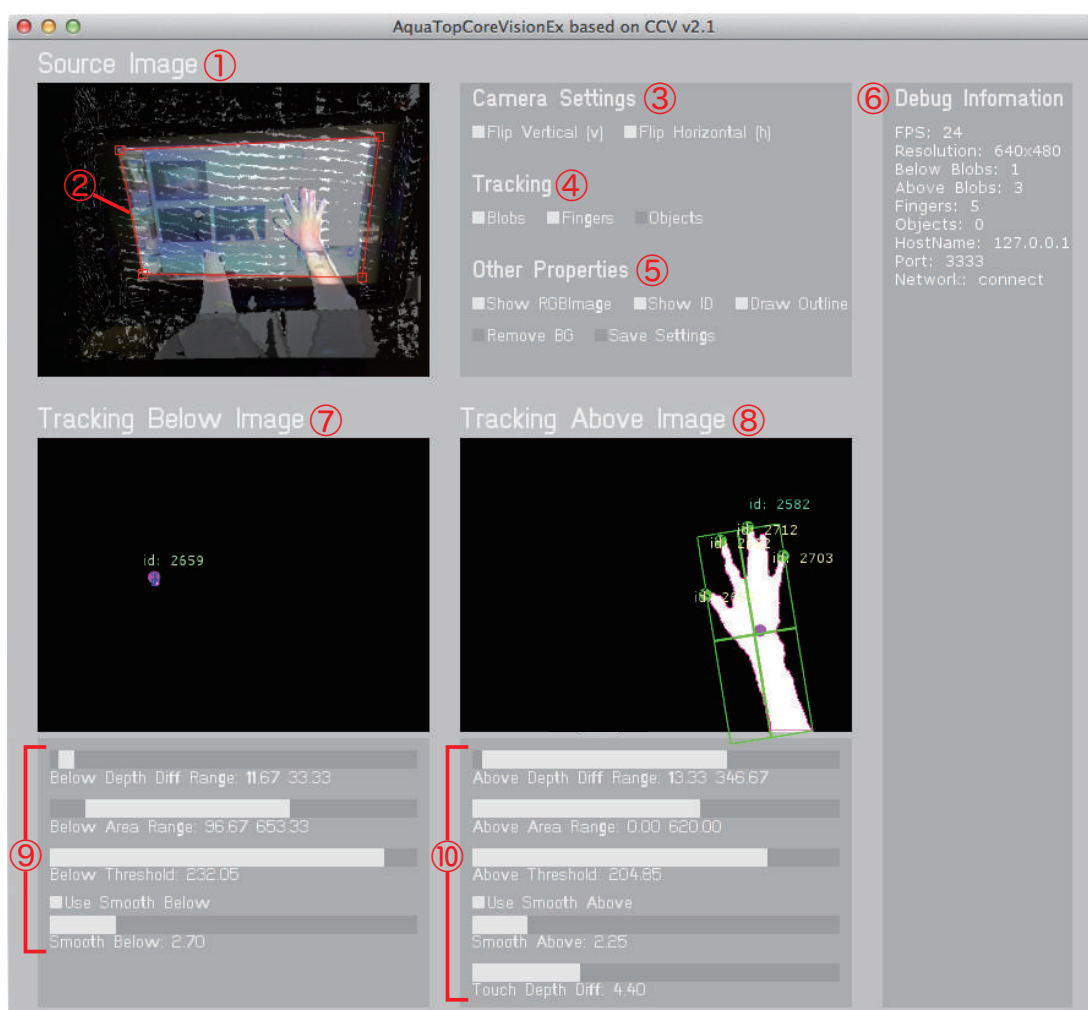


図 4.1: AquaTop Core Vision のユーザーインターフェース

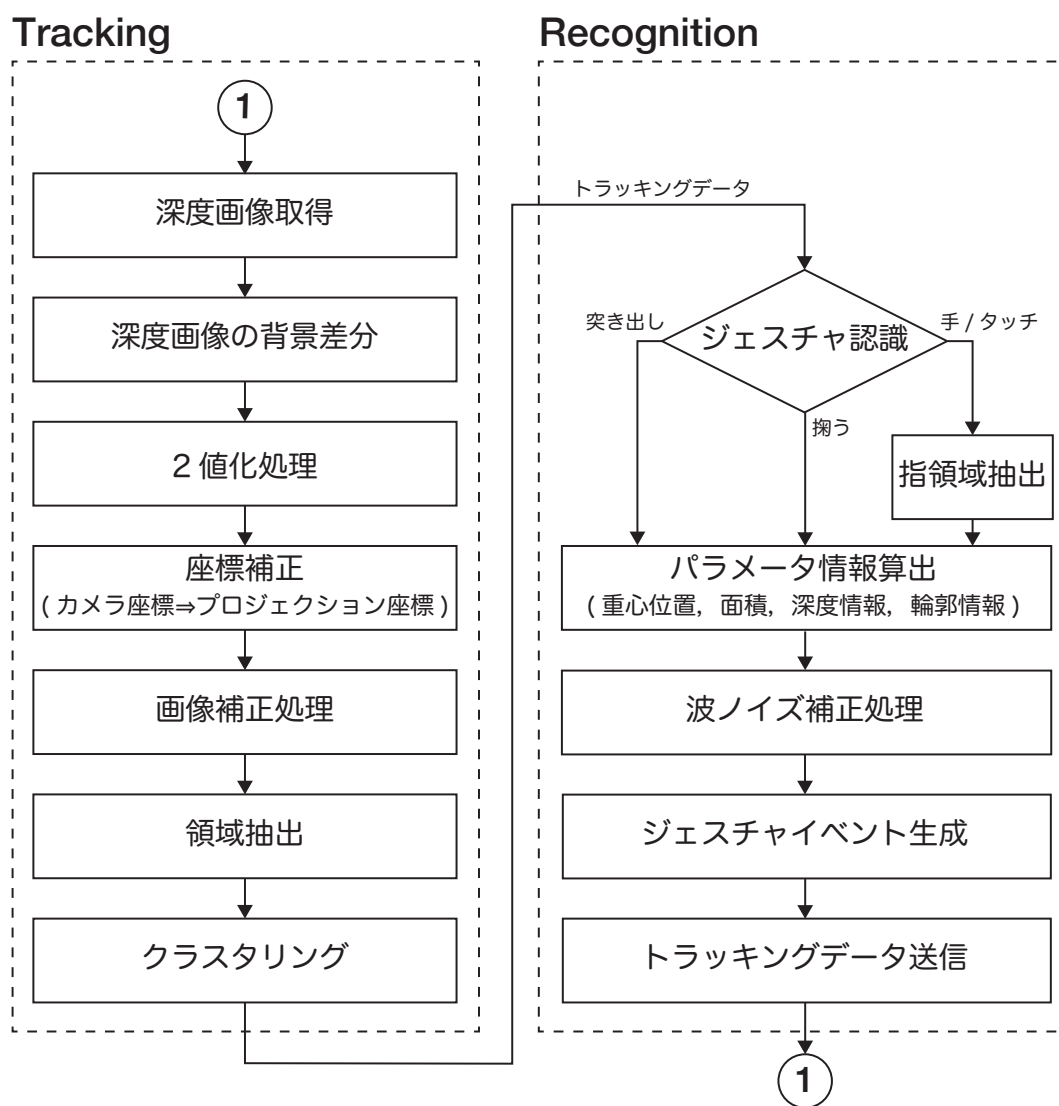


図 4.2: 本フレームワークのフローダイアグラム

4.2 キャリブレーション

キャリブレーション方法を図 4.3 に示す。黒領域が深度カメラから取得した画像全体、黒領域内部の画像はカメラから見える水面にプロジェクション領域を表している、赤枠は先述したとおり手指認識を行う領域を設定するものである。キャリブレーションは赤枠の四隅をプロジェクション領域の四隅に一致させることで行う。

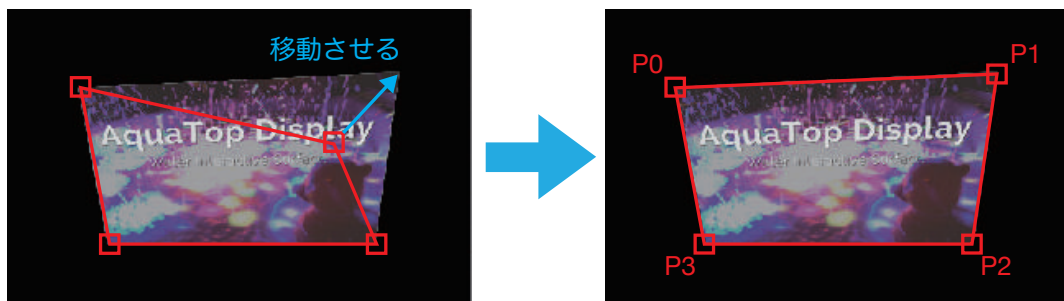


図 4.3: キャリブレーション方法

さらに、この赤枠の四隅の座標を基にカメラ座標からプロジェクション座標への座標変換を行う。座標変換を行うことにより、カメラで取得した手指の重心座標と水面ディスプレイの座標が一致する。座標変換は図 4.4 に示すように、赤枠の四隅の座標 (P0~P3) を深度画像の解像度である 640x480 に射影変換することで行う。射影変換で用いる座標変換式は次のとおりである。

$$u = \frac{ax + by + c}{gx + hy + 1}$$

$$v = \frac{ad + by + f}{gx + hy + 1}$$

u, v は変化後の座標, x, y は変換前の座標, $a \sim h$ は変換係数である。この変換式を 1 次多項式に展開すると,

$$u = ax + by + c - gux - huy$$

$$v = dx + ey + f - gvz - hvy$$

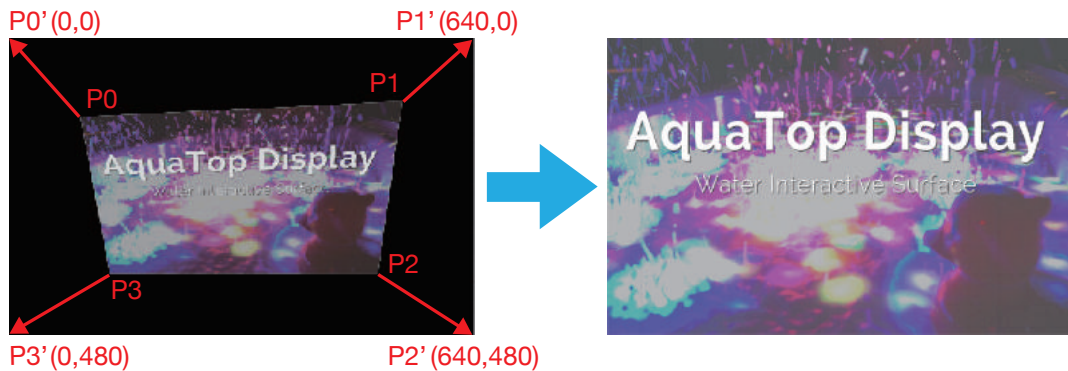


図 4.4: 射影変換後の認識領域

となる。さらに、行列として表現すると、

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

3x3 の正方行列となる。つまり、射影変換は変換前の座標 $P0 \sim P3$ と変換後の座標 $P0' \sim P3'$ から変換式を用いて 8 次連立 1 次方程式を立て、変換係数 ($a \sim h$) の 8 個の未知数を解くことで行う。

4.3 手指認識モジュール

本フレームワークの手指認識モジュールの詳細を述べる。このモジュールでは深度カメラからの深度画像取得からジェスチャのクラスタリングまでを行う。なお、このモジュールは図 4.2 の Tracking 部分のことである。

4.3.1 深度情報の背景差分と 2 値化処理

背景差分を行うために手が認識画面内に入っていない深度画像を取得し、背景画像とする。背景画像と現在の深度画像の差分を取り、差分値が図 4.1 の◎および◎の Depth Diff Range の範囲内であれば白、それ以外は黒とする 2 値化を行う。ま



図 4.5: 画像フィルタ処理の過程

た、Depth Diff Range は突き出し用と手およびタッチジェスチャ用の 2 種類あるため、認識するジェスチャに最適な認識範囲を設定可能である。

4.3.2 画像フィルタ処理

画像フィルタは平滑化処理とノイズ除去処理の 2 種類がある。平滑化後、ノイズ除去を行う。具体的には、OpenCV の関数である cvSmooth 関数を用いて 2 値画像を閾値に従い単純平滑化を行う。閾値は平滑化を行うカーネルの 1 辺の大きさで、単位はピクセルである。また、単純平滑化とはピクセルの閾値 * 閾値 隣接の総和を計算後、 $1/(\text{閾値} * \text{閾値})$ によってスケーリングすることである。なお、2 値画像に対しこの処理を行うと 255 階調のグレースケール画像に変換される。さらに平滑化後の画像に対して、threshold 関数を閾値に従い適応し 2 値画像に戻す。これらの処理により、検出した領域の輪郭補正と波や深度カメラのセンサー値抜けに起因するノイズなどを除去が可能になる (図 4.5)。また、cvSmooth 関数と threshold 関数の閾値は図 4.1 の⑨および⑩の Smooth および Threshold スライダーの値であり、値の範囲はそれぞれ 0~15、0~255 である。

4.3.3 領域抽出アルゴリズム

領域抽出はまず、2 値画像の白部分の輪郭情報を OpenCV の輪郭抽出関数である cvFindContours 関数を用いて取得する。次に、輪郭情報から面積を算出し、取得した面積の値が図 4.1 の⑨および⑩の Area Range スライダーの範囲内であれば、抽

出領域の面積以外のパラメータを算出し、それ以外は削除する。算出するパラメータは、抽出領域の最小矩形情報、抽出領域を包含する真っ直ぐな矩形、重心、重心位置における深度情報である。さらに、突き出しと水面上の手指の選別は4.3.1章~4.3.3章の処理結果によって選別し、突き出しは図4.1の Tracking Below Image、水面上の手指は Tracking Above Image の抽出情報とする。つまり、選別の精度はユーザーの図4.1の⑨および⑩のスライダーによる閾値設定に依存する。また、指認識は Tracking Above Image の抽出した領域の輪郭情報から凸包を検出し、凸包の座標から指座標を算出することで行う。その際、指座標における深度値も合わせて算出する。なお、凸包の検出には OpenCV の関数である ConvexHull 関数を用いる。

4.3.4 抽出領域の追跡アルゴリズム

抽出領域の追跡は抽出領域のパラメータである重心位置の変化によって行う。先ず、前フレームの全抽出領域と現フレームの抽出領域の対応関係を検出するために、両フレーム間の抽出領域の重心間における平方距離を計算する。計算結果が0ならばその前フレームの抽出領域のセッションIDを現フレームの抽出領域のセッションIDとする。計算結果があらかじめ設定した閾値以下ならば、対応する前フレームの抽出領域の重心情報を C++ コンテナの1つである map コンテナに格納する。その後、格納した重心情報から現フレームの抽出領域の重心に一番近いものを検索し、その抽出領域のセッションIDを現フレームの抽出領域のセッションIDとする。それ以外は現フレームの抽出領域のセッションIDを-1とする。なお、この処理は現在のフレームで抽出されている領域数分行う。次に、セッションIDに応じて追跡した抽出領域の情報を更新する。セッションIDが-1の場合は抽出領域の情報を削除し、それ以外の場合は抽出領域のパラメータを全て更新する。

4.4 ジェスチャの種類とその選別アルゴリズム

本フレームワークのジェスチャ認識モジュールの詳細を述べる。このモジュールは図 4.2 中の Recognition 部分のことである。

4.4.1 水面下から指を突き出すジェスチャ

水面をディスプレイにする問題点として、水面下からの突き出しや水を流すなどの操作を行うと水面が揺れ波が立ってしまい、認識精度に影響するノイズを発生させること挙げられる。特に、水面下からの突き出し検出はこの影響を最も受ける。解決案として、突き出し領域の検出後、その領域が0.4秒以上存在したら突き出しジェスチャとして認識するようにした。

4.4.2 水を流すジェスチャ

水を流すジェスチャは水面付近の身体動作として非常に自然な動作である。ジェスチャの認識方法は、まず、手として認識された領域の重心座標を手の平中央付近へ来るように座標補正を行う。手の平中央付近とは図 4.6 に示す位置のことである。次に、補正済み座標の前フレームと現フレームの座標変化から変化方向と加



図 4.6: 手領域の座標補正

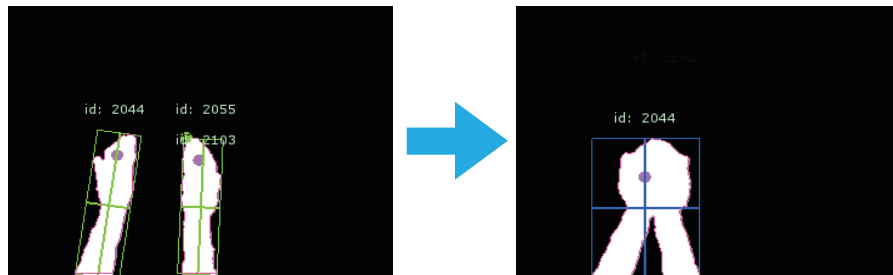


図 4.7: 掬うジェスチャの認識手法

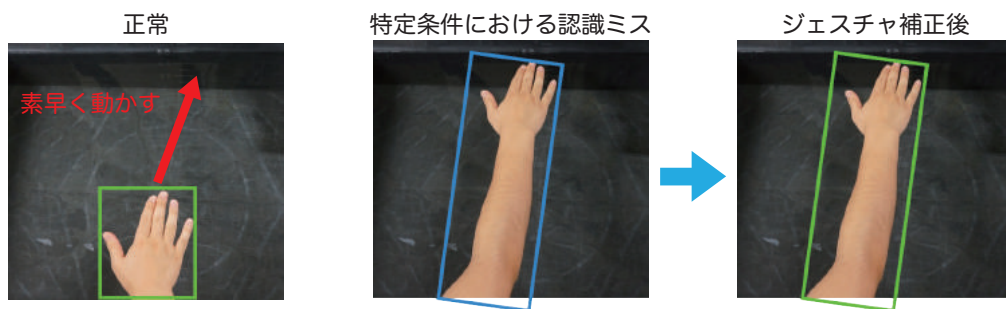


図 4.8: 掬うジェスチャのエラー処理方法

速度を算出し、これらの計算結果を流す対象に適用することで流すジェスチャとしている。このジェスチャは6章で後述するスクロール、全体的な移動、ページ送りなどに利用する。

4.4.3 水を掬うジェスチャ

掬うジェスチャは図4.7で示すように、手として認識された2つの領域を合流させることで行う。合流後のセッションIDは手として認識された2つの領域のセッションIDの若い方が適用される。掬うジェスチャはセッションIDが若い方の領域の面積が前フレームのその領域の面積と比較して1.5倍増かつ最小矩形の縦横比の差があらかじめ設定した閾値より小さい場合に認識される。本認識システムでは閾値を2.5としている。また、最小矩形の縦横比を用いてジェスチャを補正する理由を述べる。手として認識される領域の最小矩形は緑枠で表示されるが、認識された手を深度カメラのフレームレートよりも早く動かし、面積が前フレームの面

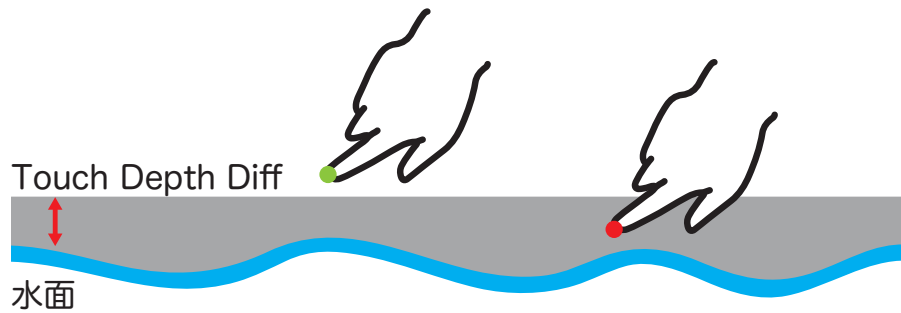


図 4.9: 指座標とタッチ検出範囲

積よりも 1.5 倍になった場合、掬うジェスチャとして誤認識される問題点がある。これを解決するために、最小矩形の縦横比から掬うジェスチャを手の認識に戻している (図 4.8)。

4.4.4 水面へのタッチジェスチャ

タッチジェスチャは図 4.9 に示すように、指座標における深度値が図 4.1 の⑩の Touch Depth Diff のスライダー値以下の範囲の時に検出する。また、タッチジェスチャが連続して認識しないように、ジェスチャ認識後 1000 ミリ秒はジェスチャの認識を行わない。

第 5 章

Application Framework

本章では本システムで用いる Application Framework について述べる。

5.1 フレームワーク概要

本フレームワークと 4 章で述べた認識システムのフレームワークの対応関係を図 5.1 に示す。ATUIO トラッカーアプリケーションは提案する認識システム、ATUIO クライアントアプリケーションは 6 章で述べる応用例のアプリケーションに対応する。トラッカーアプリケーションによって取得されたトラッキングデータは ATUIO プロトコルを通してクライアントアプリケーションに送信する。また、クライアントアプリケーション内ではトラッキングデータを必要に応じて流体シミュレーションや Awesomium HTML UI Engine に適応させる。

5.2 AquaTop User Interface Objects 通信プロトコル

AquaTop User Interface Objects (ATUIO) は NUI Group Community によって開発されたマルチタッチやタンジブルインターフェースの情報の送受信に特化した通信プロトコルである Tangible User Interface Objects (TUIO) を独自拡張した通信プロトコルである。TUIO はカルフォルニア大学バークレー校の The Center for New Music and Audio Technologies (CNMAT) によって開発された OpenSound Control (OSC) のプロトコルを基に開発されており、タッチ認識および物体認識

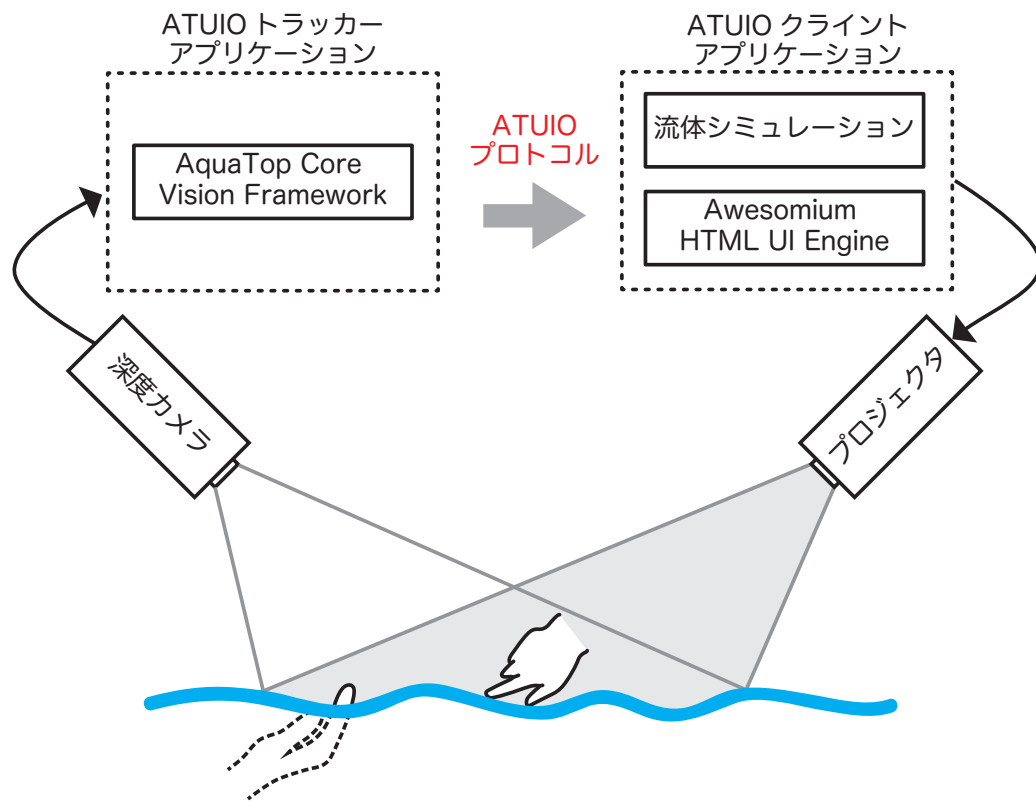


図 5.1: 各フレームワークの対応関係

の ID, 座標情報, 面積, 角度および回転速度 (物体認識のみ), イベント情報 (新規追加, 更新, 削除) を送受信が可能である [28][29]. しかし, 本システムでは複数のジェスチャを選別の必要性があるため TUIO の送受信情報では不十分である. そのため, ATUIO では TUIO の送受信情報に加え, 面積, 経過時間, 座標情報における深度情報, ジェスチャ情報を送受信可能にしている. 表 5.1 に ATUIO の通信情報の仕様を示す. なお, 認識システムには ATUIO プロトコルを送信する機能, アプリケーション側には ATUIO プロトコルを受信する機能を割り当てている. 受信側には ATUIO のイベントリスナーがあり, トラッキングデータの追加, 更新, 削除を逐次的に自動更新することが可能になっている.

表 5.1: ATUIO の仕様

パラメータ名	備考
sid	セッション ID
xpos	X 座標 (0.0-1.0)
ypos	Y 座標 (0.0-1.0)
area	面積 (0-307200px)
age	経過時間 (ms)
depth	トラッキングデータの XY 座標における深度値 (mm)
kind	ジェスチャの種類 (0:オブジェクト, 1:手, 2:タッチ, 3:掬う, 4:突き出し)

5.3 Awesomium HTML UI Engine

予め用意したコンテンツ以外に Web 上のコンテンツをネイティブアプリケーションに取り込むために Simmons らによって開発された Awesomium HTML UI Engine (以下, Awesomium) を用いる [30]. Awesomium はオープンソースの Web ブラウザである Chromium を基に組み込み用 Web ブラウザとして開発されたもので, C++ と .NET で記述されたネイティブアプリケーションに組み込むことが可能である [31]. Awesomium の使用に関しては, アプリケーション毎に 2,900 ドルの使用料が発生するが, 売上が 10 万ドル以下または, 学術利用の場合は無料である. 図 5.2 に Awesomium の概要を示す. 橙線はネイティブアプリケーションから UI コンテンツ, 紫線は UI コンテンツからネイティブアプリケーションへの命令の流れを表している. 主に前者は, クリック, スクロール, ページを戻すおよび進むなどの Web ブラウザに関連する操作を行う. 一方後者は, 更新した Web ページ情報をネイティブアプリケーションで扱える形に変換を行う. つまり, Awesomium

はネイティブアプリケーションと Web ページのブリッジの役割を果たしている。

Awesomium は WebCore と WebView と呼ばれる機能を有している。基本的に、1つの WebCore に対して取得している Web ページと同等数の WebView が存在する (図 5.3)。WebCore は UI コンテンツ取得した情報の更新, WebView は WebCore によって更新された情報をネイティブアプリケーションで扱える形に変換, ジェスチャイベントを Web ブラウザのイベントに変換などを行っている。ネイティブアプリケーションにおける Web ページ情報は, Awesomium 内部で Web ページを画像情報に変換し, その画像情報をテクスチャ (unsigned char 型) として扱う。画像情報への変換は逐次的に行われ, その都度テクスチャを更新する。つまり, パラパラ漫画のように画像情報を扱うことでネイティブアプリケーション内での Web ブラウジングを実現している。また, テクスチャとして扱うことで本来の Web ページ情報の上にネイティブ側から追加のイベント情報をオーバーレイ可能である。図 5.4 にイベント情報のオーバーレイの例を示す。図 (左) が取得した本来の Web ページ情報, 図 (右) がページを戻るおよび進むボタンをネイティブ側から追加したものである。初期設定として, ネイティブ側でページを戻るまたは進むボタンのイベントを Awesomium のページを戻る (GoBack 関数) または進む (GoForward 関数) イベントを紐付けることにより, ネイティブ ⇒ Awesomium ⇒ Web ページの順に設定したイベントが伝播する。これにより, ネイティブ側のイベントで Web ページのイベントを処理することが可能になる。また, 特定のジェスチャに Awesomium のイベントを紐付けることでも同様な処理を行うことが可能になる。

5.4 流体シミュレーション

流体シミュレーションは Stam の Navier-Stokes 方程式を利用したゲーム向けリアルタイム流体力学アルゴリズムを基に開発された MSAFluid を用いた [32]。MSAFluid は流体情報をベクトル場として扱っている。図 5.5 に MSAFluid を用い

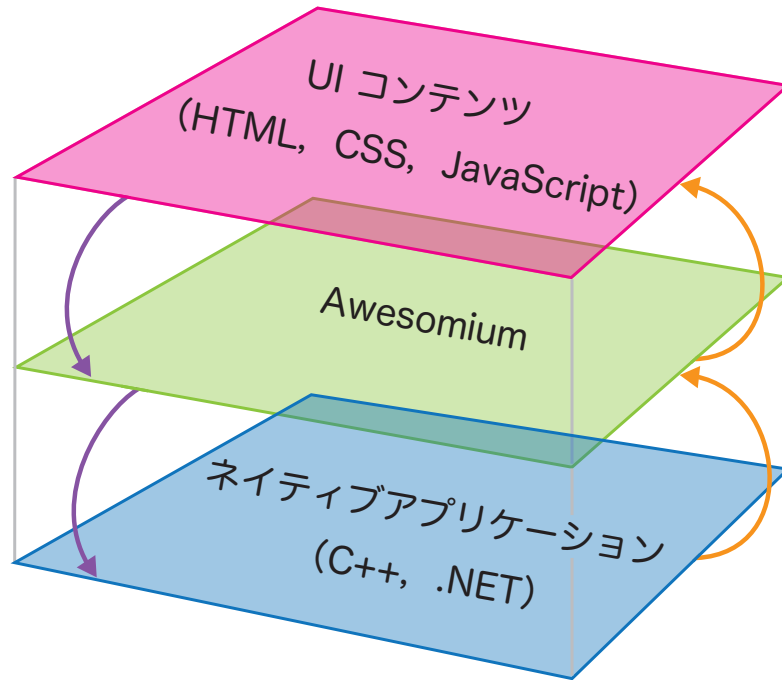


図 5.2: Awesomium HTML UI Engine の役割

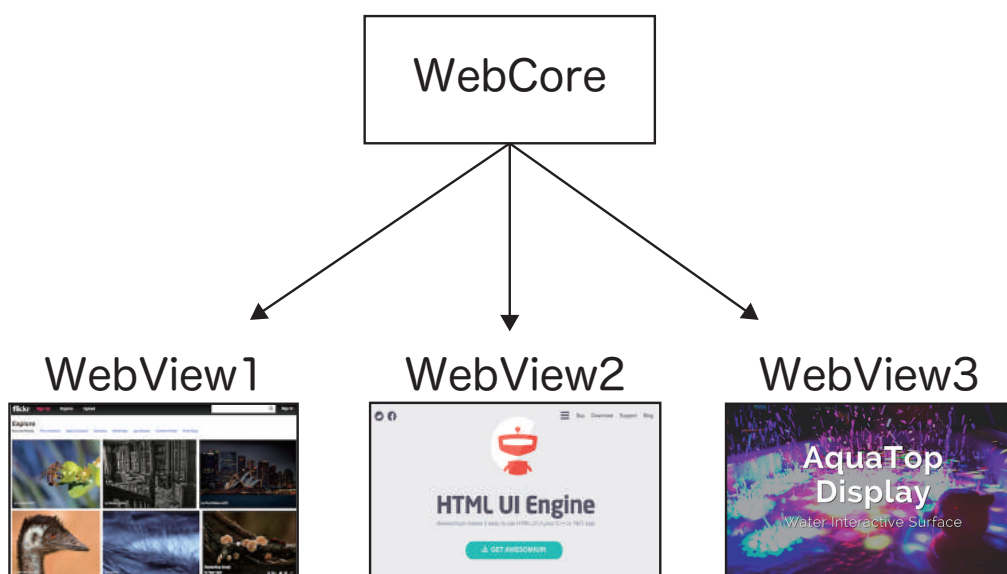


図 5.3: WebCore と WebView の関係

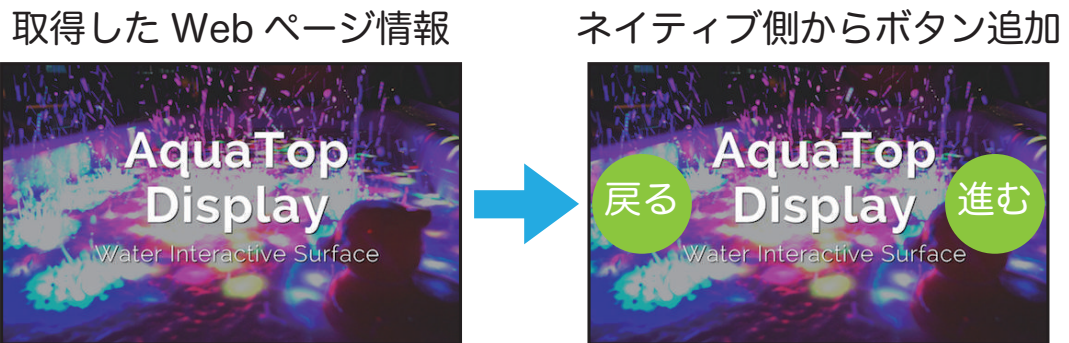


図 5.4: オーバーレイされたイベント情報の例

た流体情報とエフェクト生成方法を示す。図（上）はベクトル場に掛かる力の様子を表している。赤線はベクトル場に対して力を加えている位置である。図（中）はベクトル場全体の様子を表している。図（下）はベクトル場の力の大きさと向きに対して色をマッピングしたものである。6章にて述べる応用例のアプリケーションでは、認識した手の1フレーム前と現在のフレーム位置情報からベクトル場に加える力の大きさと向きを算出し適応することで、流体情報を生成している。また、表示しているコンテンツと流体情報の組み合わせについては、コンテンツの中心座標における加速度を流体情報から取得し、その加速度をコンテンツの位置情報に加算することで行う。これにより、コンテンツが水面を流れるながら浮遊しているように見せることが可能になる。

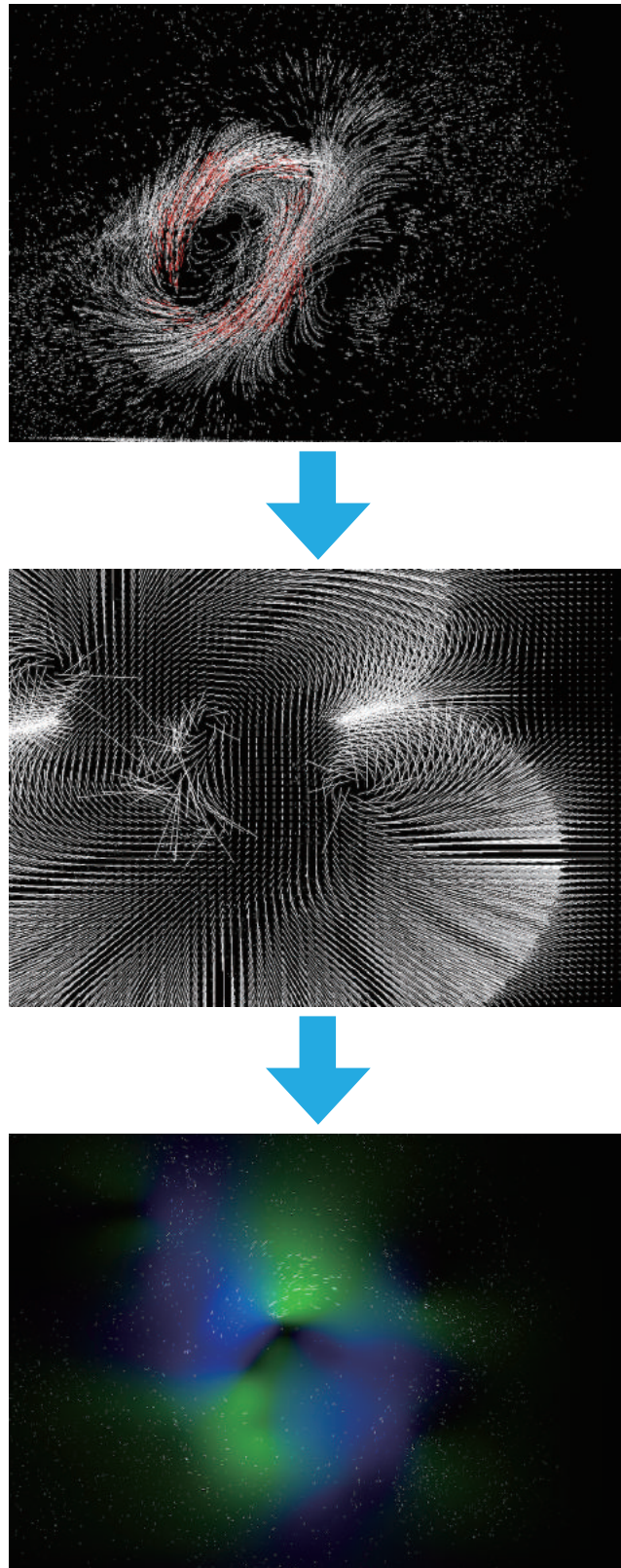


図 5.5: 流体情報とエフェクト生成方法

第 6 章

応用例

本章では提案する認識システムを用いた 3 つの応用例について述べる。

6.1 AquaTop Multi Viewer

本システムは水環境特性を利用し、動画像、音楽、Web ページ、ブックビューワー、メール送信機能の操作および閲覧を可能にするものである。利用する水環境特性は次の 3 点である。

1. 従来のマルチタッチサーフェスではユーザーが表示されている情報に対して操作を行わない限り情報はその場に表示され続けるが、本システムでは流体シミュレーションによって計算された加速度を基に水の流れに応じて情報が水面を流れていくように表示される点。
2. 従来のマルチタッチサーフェスでの操作はスクリーン上で行うが、本システムでは図 6.1 で示すようにスクリーン（水面）上下から行うことが可能で、それぞれ違う機能を割り当てることができる点。
3. 従来のマルチタッチサーフェスのスクリーン素材は硬質のため変化することが無く常に一定であるが、本システムはスクリーン素材に水を利用するためスクリーンの一部を掬い取ることで切り離すことができる点。

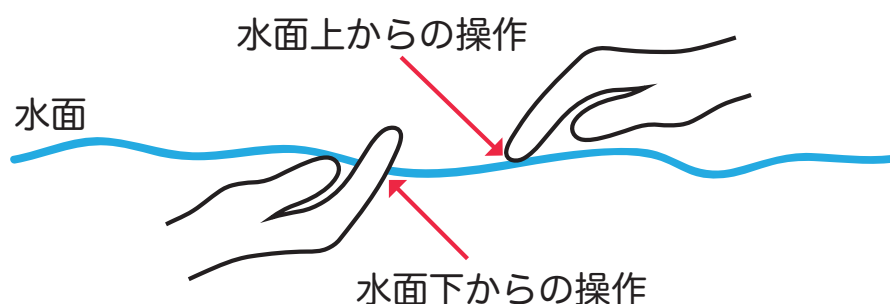


図 6.1: 水環境特性を利用した操作面の方向

また、本システムで使用可能なジェスチャは、突き出し、流す、水面上へのタッチ、掬うの4種類である。なお、Web ページ操作のジェスチャは複数のジェスチャを複合することによって実現している。

6.1.1 システム概要

本システムにおけるフローダイアグラムを図 6.2 に示す。本システムではまず、認識システムから送信されたトラッキングデータをジェスチャイベントリスナーによって取得し、取得データが新規追加、更新、削除の内どのイベントなのかを検証する。その後、取得データをジェスチャ毎に現在のディスプレイモードに適したイベント処理を行う。ディスプレイモードとは、図中のディスプレイモードセレクトタからわかるように、水平、自由、全画面の3種類のモードのことであり、各ディスプレイモードの機能の概要は次のとおりである。

水平モード

水平モードの画面表示と機能を図 6.3 に示す。このモードでは画面中心のコンテンツに対して操作が可能になっている。画面左下のボタンをタッチすることで、メニューが開き、任意のコンテンツ追加ボタンをタッチすることでコンテンツを追加することができる。コンテンツの切り替えは流すジェスチャで行う。また、ディスプレイモード変更については、右下のディスプレイボタンをタッチすることで自由モード、画面中心のコンテンツの右上の拡大ボ

タンをタッチすることで全画面モードへ変更できる。

自由モード

自由モードの画面表示と機能を図 6.4 に示す。このモードでは全てのコンテンツを全て同時に操作が可能になっている。画面左上のボタンをタッチすることで、水平モードと同様にメニューが開き、任意のコンテンツを追加することができる。また、メニュー最下位のボタンをタッチすることでディスプレイモードを水平モードへ変更することができる。全画面モードへの変更は水平モードと同様に任意のコンテンツ右上の拡大ボタンをタッチすることで行う。コンテンツの移動、回転、拡大縮小、消去は水面の下からのジェスチャで行う。さらに、流すジェスチャを行うことで表示している全コンテンツに対して流体シミュレーションの計算結果を適応し、流れるように移動させることや、掬うジェスチャを行うことで掬ったコンテンツを別の場所に移動させることもできる。

全画面モード

全画面モードの画面表示と機能を図 6.5 に示す。このモードでは拡大ボタンをタッチした任意のコンテンツを全画面表示するものである。全画面表示後の操作はコンテンツに応じて異なる。例えば、Web ページならばページ操作、動画ならば再生、停止、早送り、巻き戻しである。また、コンテンツ右上の縮小ボタンをタッチすることで全画面モードを解除することができる。解除後は全画面モードが適応されるひとつ前のディスプレイモードに変更する。

また、水平モードではタッチと流すジェスチャ、自由モードでは全てのジェスチャ、全画面モードでは表示コンテンツ別に用意されたジェスチャが使用可能である。

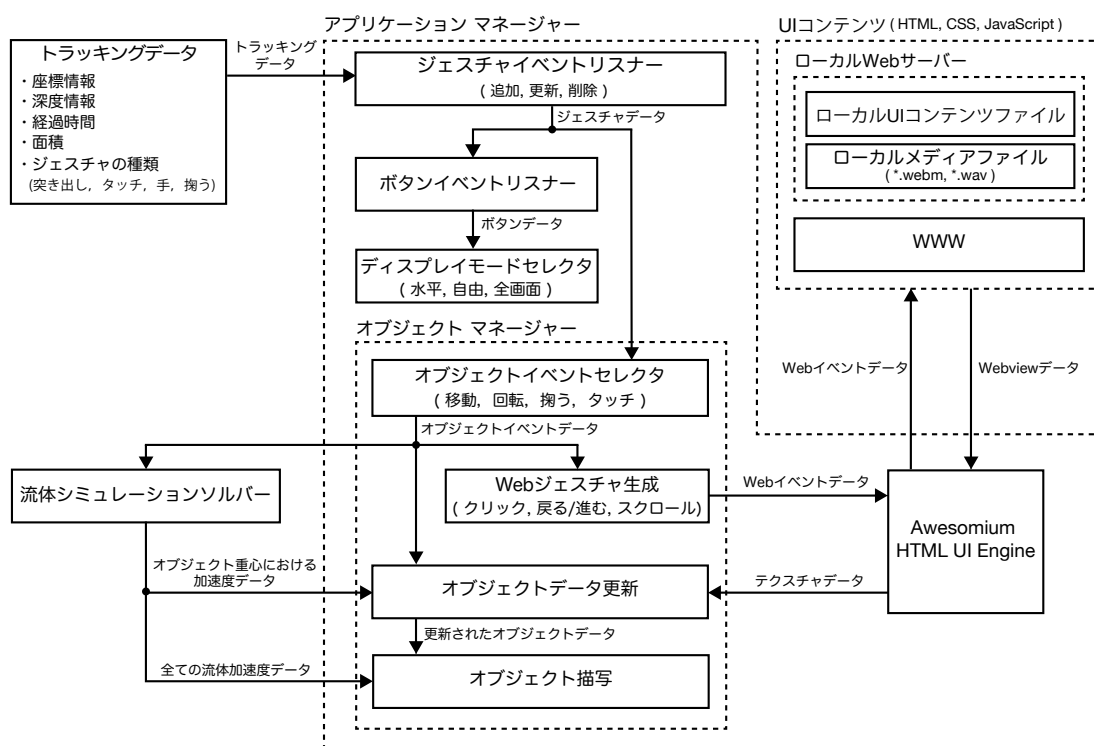


図 6.2: AquaTop Multi Viewer のフローダイアグラム



図 6.3: 水平モードの画面表示と機能



図 6.4: 自由モードの画面表示と機能

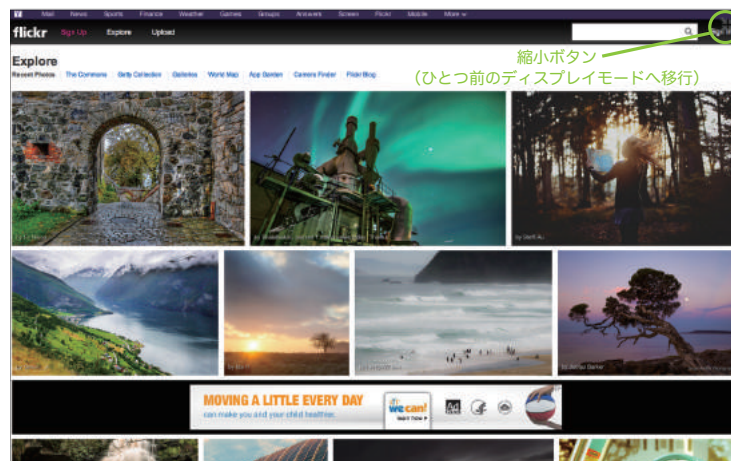


図 6.5: 全画面モードの画面表示と機能

6.1.2 ジェスチャと操作方法におけるインタラクション

3種類のディスプレイモード別に各ジェスチャと操作方法のインタラクションの詳細を述べる。

水平モードにおけるジェスチャと操作方法

流すジェスチャはカバーフロー表示されているコンテンツの切り替えに用いている。図6.6に示すように切り替えたい方向に流すジェスチャを行うとコンテンツを切り替えることができる。

タッチジェスチャはメニューボタンの操作や画面中心のコンテンツ操作に用いている。前者は、メニューボタンをタッチすることでコンテンツ追加ボタンが表示される（図6.7）。表示されたコンテンツ追加ボタンは一定時間が過ぎると自動的に沈む画像エフェクトと共に消える。後者は共通操作である全画面モードへの移行（図6.8）と特定のコンテンツの操作を行う。特定のコンテンツとは動画、音楽、ブックビューワーのことで、それぞれ動画の再生および停止、音楽の再生および停止、ページ送りを行うことができる。

自由モードにおけるジェスチャと操作方法

突き出しジェスチャはコンテンツの移動、回転、拡縮、消去の操作に用いている。4種類の操作はコンテンツ内に突き出している指の本数と指同士の動きで操作を切り替えている。各操作の認識手法は次のとおりである。

移動

図6.9に示すようにコンテンツ内に指を1本突き出すと、移動可能状態を示す赤枠がコンテンツの周囲に表示される。赤枠表示後、移動させたい任意の場所に指を突き出したまま移動させることでコンテンツの移動を行う。なお、移動操作はマルチユーザー対応である。また、水面のノイズ除去やトラッキ

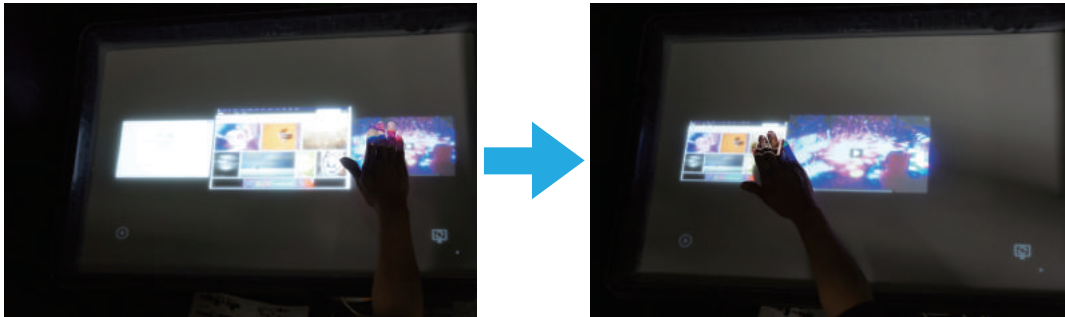


図 6.6: 流すジェスチャによるコンテンツ切り替えの様子

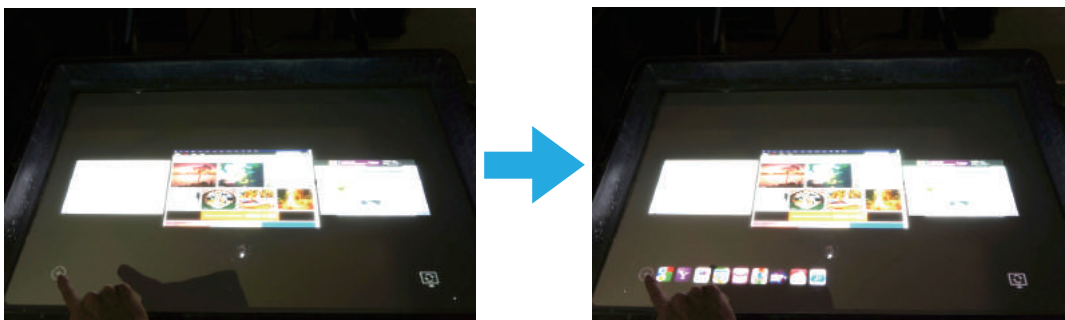


図 6.7: タッチジェスチャによる水平モード用コンテンツ追加ボタン表示の様子

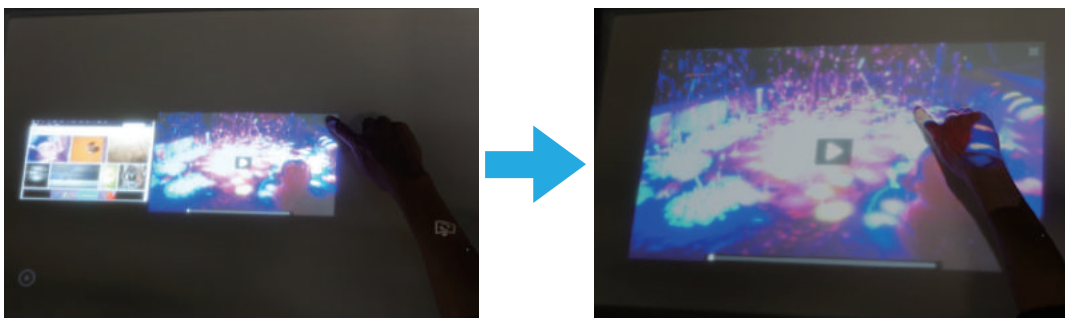


図 6.8: タッチジェスチャによる水平モードから全画面モードへの移行の様子



図 6.9: 突き出しジェスチャによるコンテンツ移動の様子

ングデータ通信によって最大で 500 ミリ秒の遅延が存在する。

回転・拡縮

図 6.10 に示すようにコンテンツ内に指を 2 本突き出すと、回転および拡縮可能状態を示す緑枠がコンテンツの周囲に表示される。緑枠表示後、2 本の指間の角度から回転率、距離から拡縮率を算出し、算出した数値を基にコンテンツの回転および拡縮操作を行う。回転および拡縮率は 1 フレーム前と現在のフレームの位置関係から算出している。なお、回転および拡縮操作は同時に行うことが可能であり、マルチユーザー対応である。

消去

図 6.11 に示すようにコンテンツ内に指を 3 本以上突き出すと、消去可能状態を示す青枠がコンテンツの周囲に表示される。青枠表示後、突き出した指を全て水中に沈めるとコンテンツの消去を行うことができる。消去は 2 段階になっており、1 段階目は操作を行ったコンテンツにブラーが掛かり水中に沈んだ状態になる。2 段階目は沈んだ状態のコンテンツに対して、もう一度消去操作を行うことで完全にコンテンツを消去することができる。

タッチジェスチャは水平モードと同様にメニューボタンの操作や任意のコンテンツ操作に用いている。前者は、メニューボタンをタッチすることでメニューの開閉が行える (図 6.12)。なお、メニューが開いた状態ではコンテンツ追加ボタンと

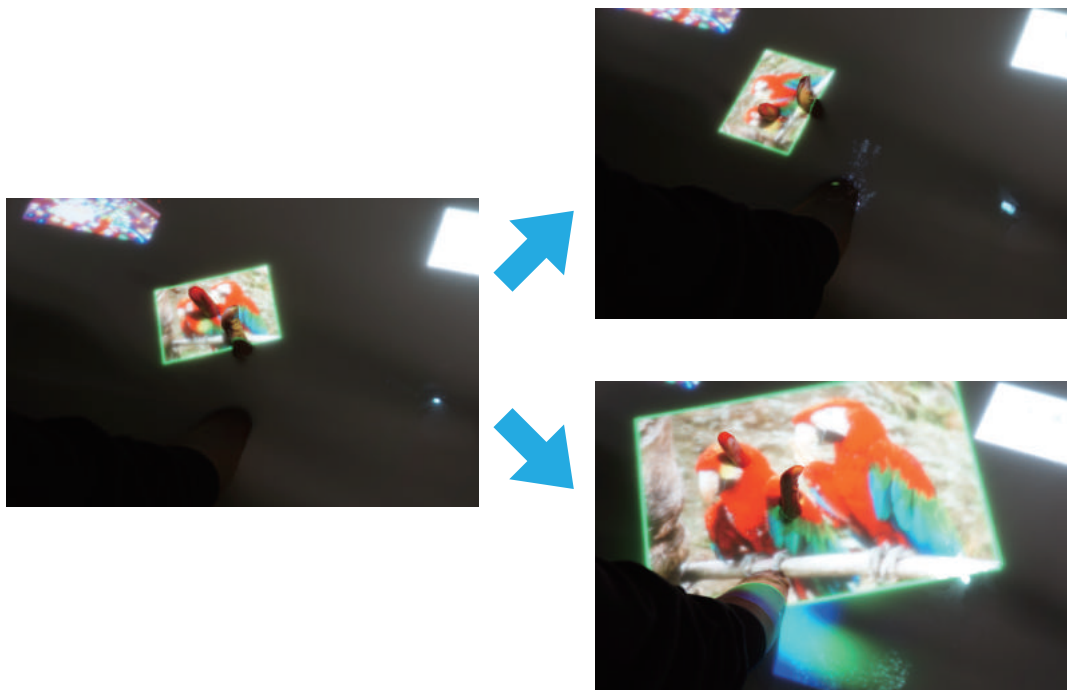


図 6.10: 突き出しジェスチャによるコンテンツ回転および拡大の様子

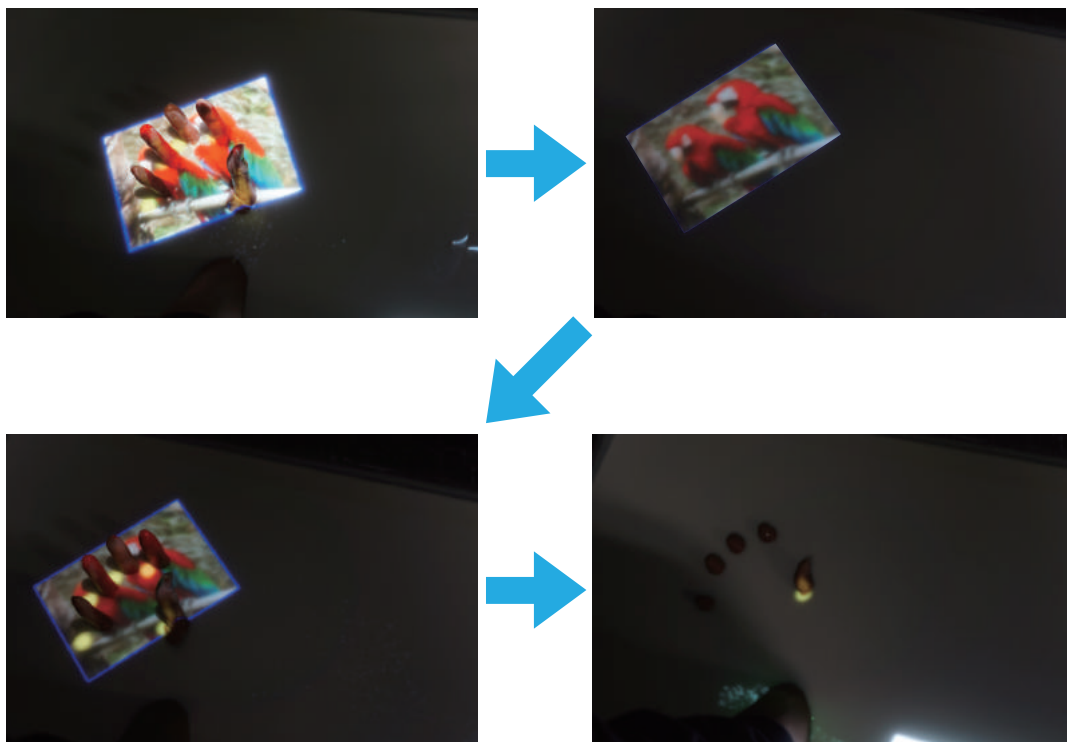


図 6.11: 突き出しジェスチャによるコンテンツ消去の様子

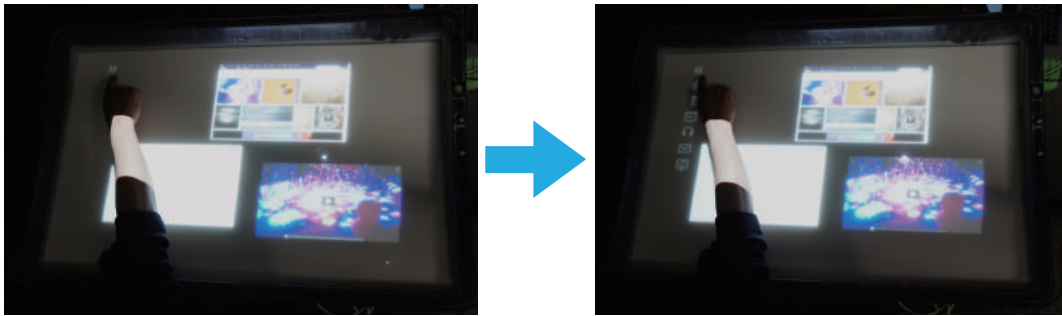


図 6.12: タッチジェスチャによる自由モード用コンテンツ追加ボタン表示の様子

水平モード移行ボタンが表示される。コンテンツ追加ボタンをタッチすると、指定したコンテンツが水底から浮き上がるように表示される。ただし、特定の検索用 Web ページ、メールソフトに関しては画面上にそのコンテンツが存在しない時のみ追加される。既存する場合は、コンテンツの周りに赤枠が表示され点滅しユーザーの注意を引くようになっている。コンテンツ操作は水平モードと同様である。

流すジェスチャは全表示コンテンツを一度に移動させるために用いている。具体的には、ジェスチャの軌道と移動距離のデータから流体シミュレータを用いて流体情報を計算し、計算結果をコンテンツ全体に適応することで表示コンテンツが水面を浮遊しているような表現を可能にしている。例えば、図 6.13 に示すように右から左へ流すことで、左側に表示コンテンツを一度に移動させることが可能である。なお、実際の水面の波データと流体シミュレータで得られたデータは一致していない。

掬うジェスチャはコンテンツを掬い上げ、任意の場所へ移動させるために用いる。図 6.14 に示すように、掬い上げたコンテンツは掬い上げると手の平に収まる大きさに縮小され、掬い上げた状態を保ちながら移動することでコンテンツの移動を行う。掬うジェスチャの解除方法は両手を開くまたは、水中に沈めることで行う。ジェスチャ解除すると、掬い上げていたコンテンツは水面に落下し、掬い上げる前の大きさに戻る。また、図 6.15 に示すように、検索用コンテンツの上に

Web ページとメールソフト以外のコンテンツを落下させると、そのコンテンツに関連するコンテンツ群を自動的に検索する。検索が始まると自動的に全画面モードに移行し、検索結果を表示する。検索結果は最大で動画、画像、音楽、Web ページの4種類のカテゴリ別に行われ、検索元のコンテンツを中心に枝に繋がれた状態で表示される。枝に繋がれたカテゴリをタッチすることで、そのカテゴリの検索結果を表示することができる。選択したカテゴリの検索結果をタッチすることで、選択したコンテンツを追加することができる。なお、コンテンツ追加と同時に検索用コンテンツは消去される。

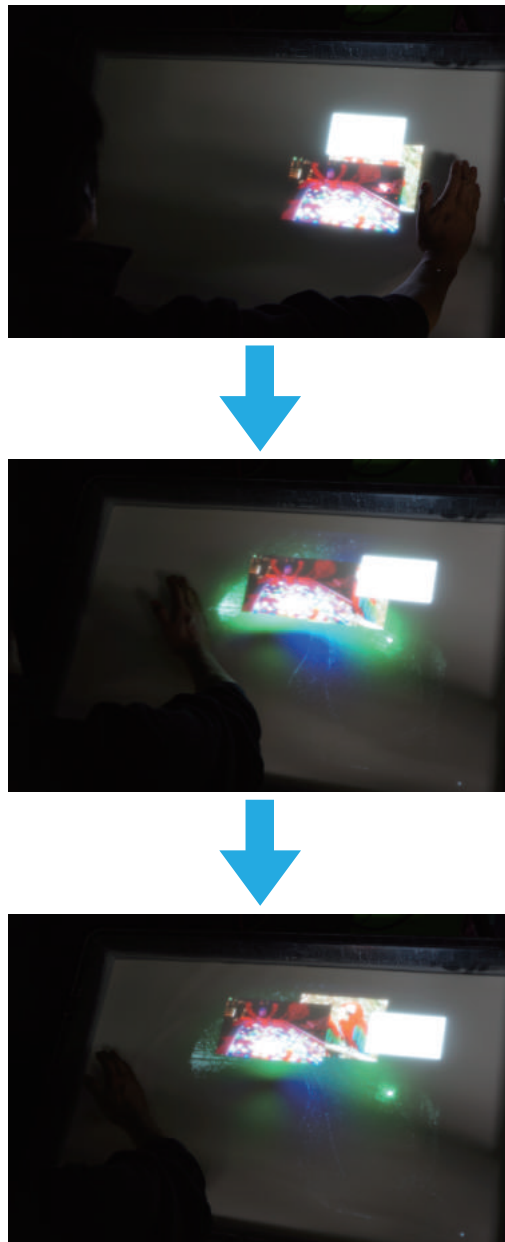


図 6.13: 流すジェスチャと流体シミュレーションによるコンテンツの移動の様子

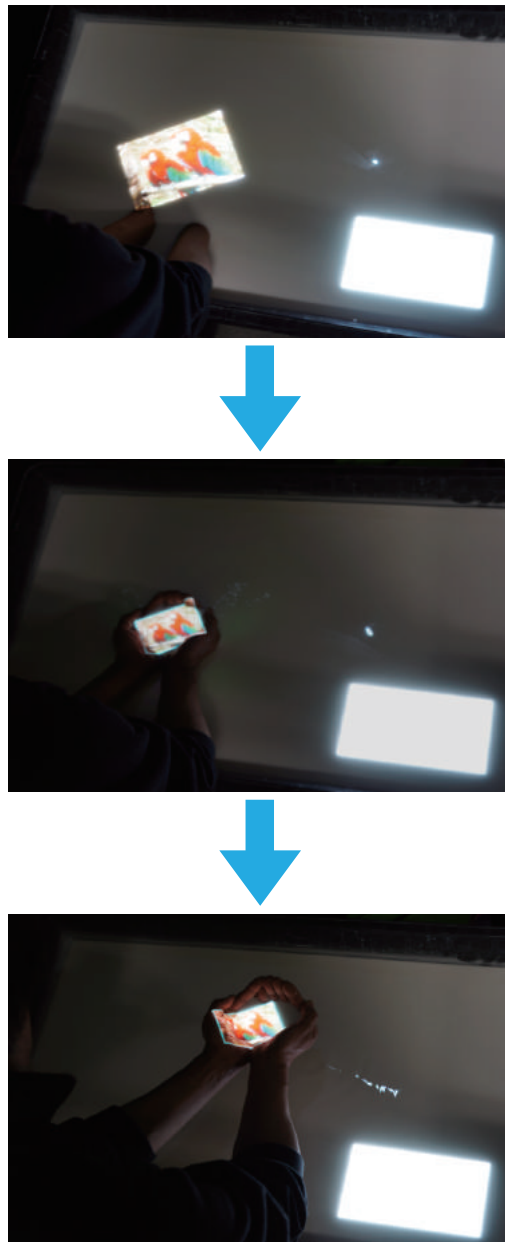


図 6.14: 掬うジェスチャによるコンテンツ移動の様子

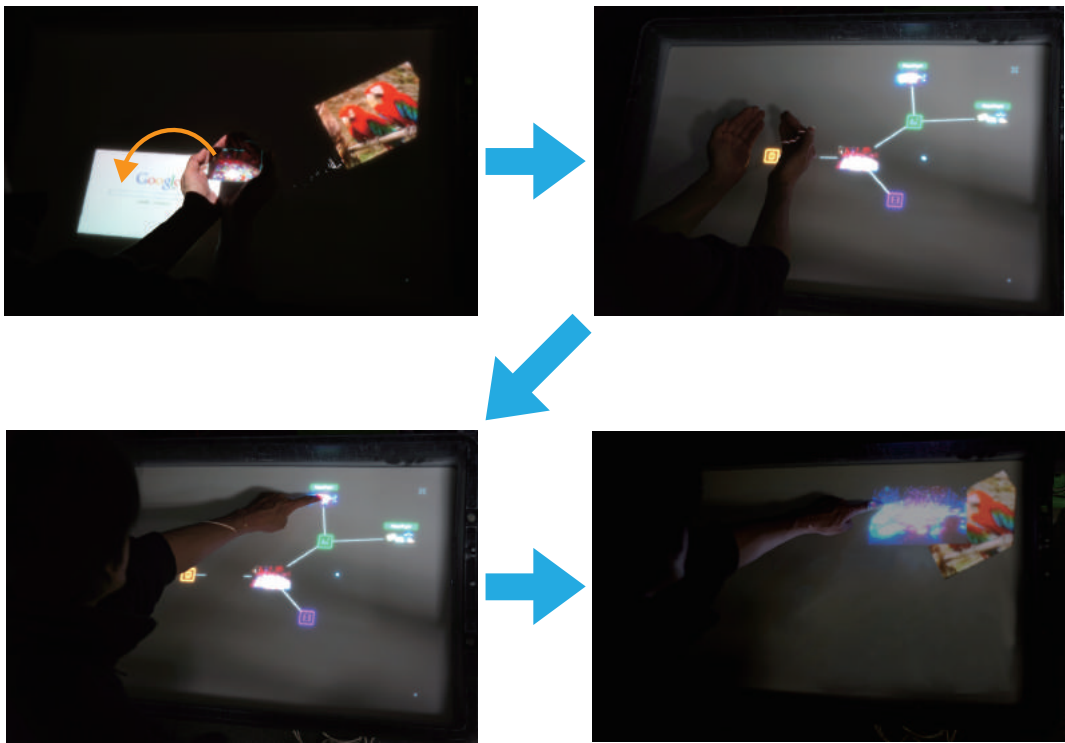


図 6.15: 掬うジェスチャによるコンテンツ検索の様子

全画面モードにおけるジェスチャと操作方法

全画面モードは先述したとおり、コンテンツによりジェスチャに対する操作が異なるため、コンテンツ別にジェスチャと操作の対応関係を述べる。なお、タッチジェスチャによるコンテンツ右上に縮小ボタンの操作は全てのコンテンツが使用可能である。

画像

全画面表示のみであり、特別な操作は行わない。

地図

突き出しジェスチャにより地図内の移動、拡縮操作が行える。

動画

タッチジェスチャにより動画の再生および停止、流す動作により動画の巻き戻しおよび先送りが行える。

音楽

タッチジェスチャにより音楽の再生および停止が行える。

ブックビューワー

流すジェスチャによりページ送りが行える。

ペイントソフト

突き出しジェスチャによりペンおよびブラシによる線の描写、タッチジェスチャにより色やツールの変更が行える。

メールソフト

タッチジェスチャによりソフトウェアキーボードを用いた文字入力とメール送信が行える。

Web ページ

タッチジェスチャによりリンクのクリック、流すジェスチャによりページを戻るおよび進む操作とスクロールが行える。

全画面モードにおける流すジェスチャによる2種類のイベント処理手法の詳細は次のとおりである。ページ送り操作のイベント処理手法を図6.16に示す。図中の赤色および青色の矩形はイベントトリガーを表している。ページを戻る操作は、赤色の矩形より右側の範囲から左端の青色の矩形に向かって流すジェスチャを行うことで実現している。ただし、イベントが発生する条件として、一連の流れを赤色の矩形を通過後、1000ミリ秒以内に完了しなければページ送りのイベントは発生しない。なお、この操作はページを戻る操作のものであり、ページを進む操作はページを戻る操作を左右反転させたものである。スクロール操作のイベント処理手法を図6.17に示す。図中の赤色および青色の矩形の役割とイベントの発生条件はページ送りと同様である。スクロール操作は赤色の矩形から右上端の青色の矩形に向かって流すジェスチャを行うと下にスクロール、この操作を上下反転させた動作を行うと上にスクロールをする。



図 6.16: 流すジェスチャによるページ送り操作の手法

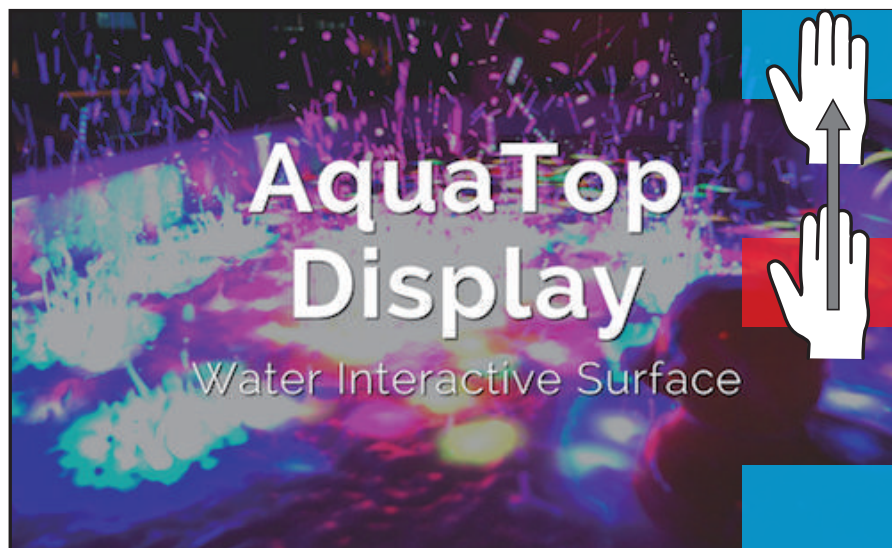


図 6.17: 流すジェスチャによるスクロール操作の手法

6.2 AquaTop Shooting Game

本システムは従来のコントローラを利用したゲームとは異なり、操作面と表示が一致させた攻撃やスピーカユニットによる物理的な爆発エフェクトが可能な新しいシューティングゲームである (図 6.18).

6.2.1 システム概要

本システムは高橋らが作成したシューティングゲームを改良したものである。主な改良点は、物理的な爆発エフェクト用のスピーカユニット増設、触覚フィードバック用スピーカユニット追加と一部画像エフェクトを OpenGL Shading Language (GLSL) で記述したところである。GLSL は C 言語をベースとした高レベルシェーディング言語であり、従来の OpenGL のハードウェア固定機能に囚われない柔軟かつ高品質なリアルタイム 3DCG シーンを構築可能にするものである。本システムにおけるフローダイアグラムを図 6.19 に示す。



図 6.18: シューティングゲーム

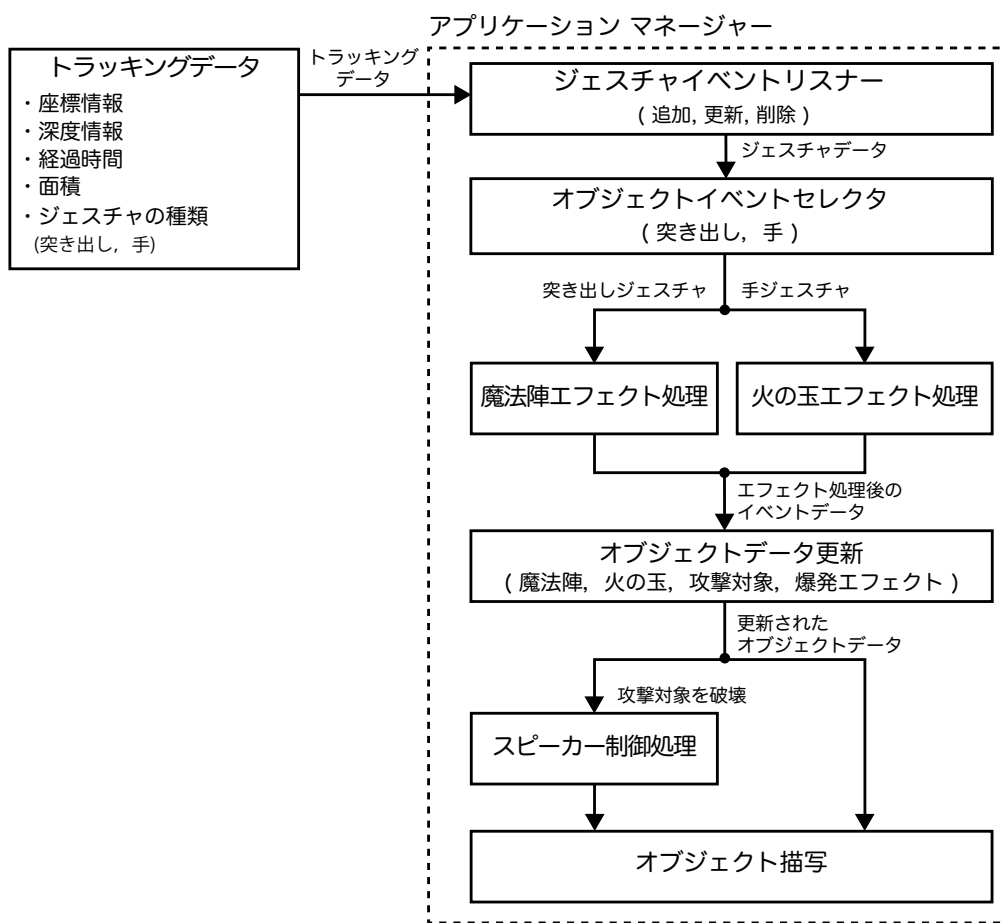


図 6.19: シューティングゲームのフローダイアグラム

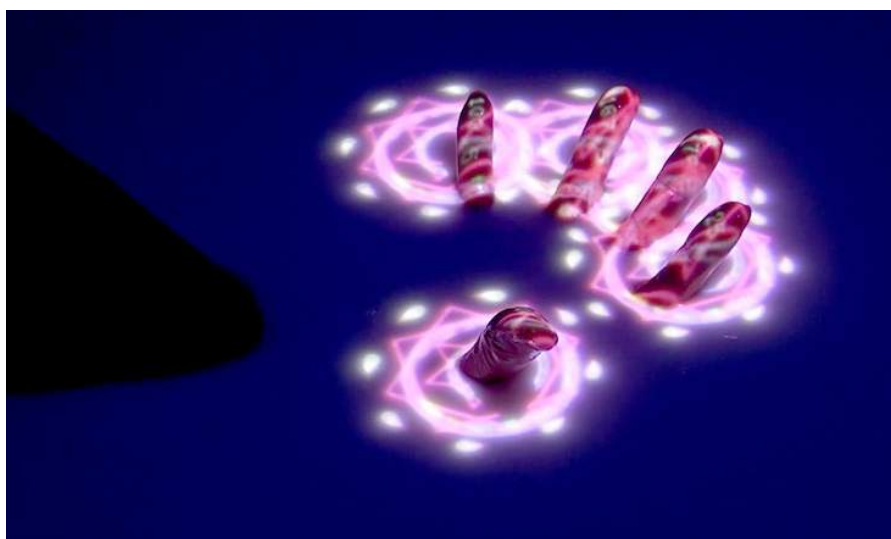


図 6.20: 突き出しジェスチャによる魔法陣エフェクト

6.2.2 ジェスチャと攻撃方法におけるインタラクション

攻撃方法は魔法陣攻撃，火の玉攻撃の2種類がある．魔法陣攻撃は突き出しジェスチャによって行う．このジェスチャを行うと魔法陣が指の周りに出現する（図 6.20）．その後，指を魔法陣を飛ばしたい方向動かし水面に沈めると魔法陣を発射することができる．指を突き出した位置と魔法陣が表示される位置が一致することで，ゲーム内に自身の身体が入り込んでいる感覚が得られ没入感の高い攻撃操作が可能になっている．火の玉攻撃は図 6.21 に示すように両手を合わせるジェスチャを行うと合わせた手の中心付近に火の玉が出現する．その後，火の玉を飛ばしたい方向に合わせた手を押し出すことで火の玉を発射することができる．火の玉の出現条件は現在の手として認識されている領域の面積が1フレーム前の領域の1.5倍以上になることである．つまり，2つの手として認識されている領域を手を合わせることで合流させ，1つの認識領域にすることである．

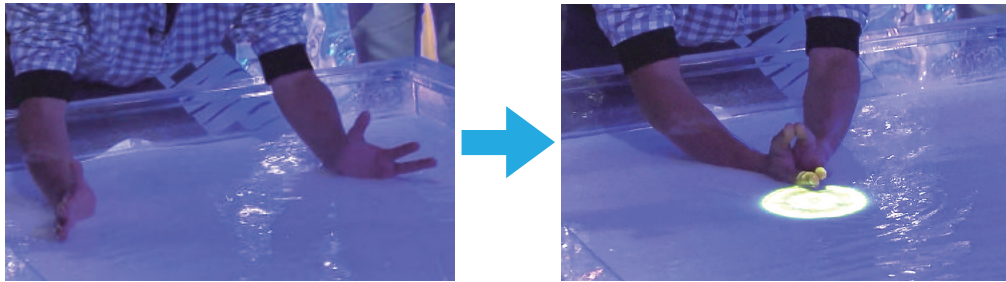


図 6.21: 手を合わせるジェスチャによる火の玉エフェクト

6.2.3 スピーカーを利用したフィードバック

図 6.22 に用いるスピーカーユニットを示す。スピーカーユニットは防水加工を施し水中に沈めて使用する。スピーカーの制御は Arduino Mega 2560 で行う。スピーカーを利用したフィードバックには 2 種類あり、赤枠で囲まれたスピーカーユニットは触覚フィードバック用、青枠で囲まれたスピーカーユニットは爆発エフェクト用に使用する。前者は、ゲーム中の特定の敵からの攻撃を受けた際に駆動させ、水中に沈めているユーザーの手に振動を伝え実際に攻撃を受けている感覚を与える。後者は、ユーザーの攻撃によって敵が倒されると駆動させ、プロジェクションによるエフェクトと共に水しぶきを上げる。これにより、敵を倒すと水面が爆発したように見え臨場感の高い演出が可能になる。

6.3 大規模なゲームアプリケーションの試作

ゲームアプリケーションの大規模化の試作を図 6.23 に示す。使用したビニールプールは縦 1720mm、横 2620mm、深さ 500mm の大きさである。試作ゲームを実際に稼働させた結果、認識精度に致命的な問題があることがわかった。試作ゲームでの深度カメラの位置は水面全体を撮影するため水面から最大で 3.3m 離れた位置に設置したため、手の認識は可能であるが、突き出しジェスチャの認識は深度カメラのノイズや認識に必要な領域を取得できないなどの理由からほぼ不可能であった。



図 6.22: フィードバック用スピーカーユニット

解決案としては、2台の深度カメラを用いた場合の例を次に示す。まず、水面へのプロジェクションを行っている領域を縦半分に分け、左右の領域をそれぞれ別の深度カメラで撮影する。その後、2台分のトラッキングデータをゲームアプリケーションに送信し、座標情報の補正を行う。これにより、深度カメラ1台あたりの設置位置が低く設定可能になり、ノイズや認識に必要な領域を取得できないなどの問題点を解決できると考えられる。提案する解決案の試作を図6.24に示す。このシステムは深度カメラを1台しか用いていないが、提案手法により大規模な浴槽でも突き出しジェスチャの認識が可能であることを示している。また、深度カメラの数をさらに増やすことにより、1台の深度カメラでは認識できないような広い領域でも正確にトラッキング可能になると考えられる。



図 6.23: ビニールプールを用いた大規模ゲームの例

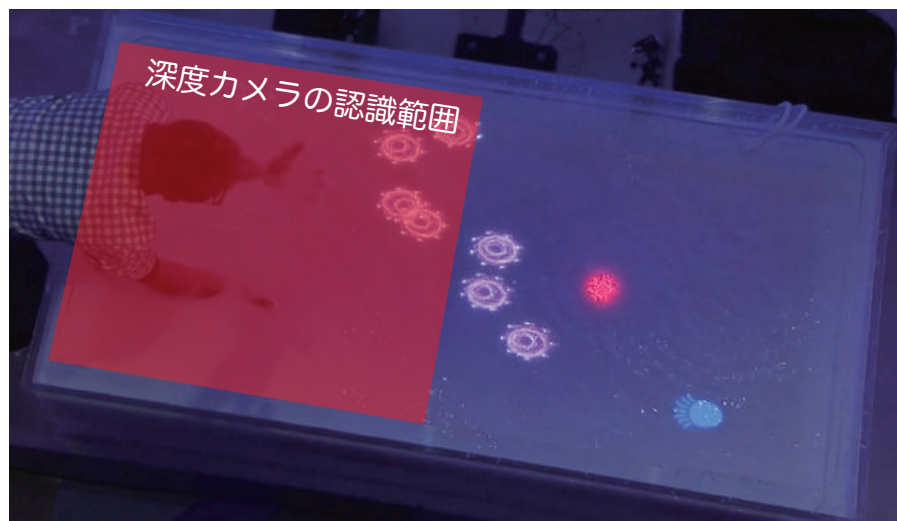


図 6.24: 提案する解決案を用いた大規模ゲームの例

第 7 章

評価

7.1 波ノイズが認識システムに与える影響の調査

本評価実験は波ノイズが認識システムの認識率に及ぼす影響を調査する。

7.1.1 実験目的

本評価実験の目的はジェスチャを行うことによって引き起こされる水面の波の影響を可視化することで、最適な閾値を調査することである。図 7.1 に閾値の詳細を示す。この閾値は図中のジェスチャ認識範囲と安全マージンの間の値のことであり、水面に波が立っていない状態（基準状態）からの指の付き出し具合と指認識を開始する深度情報に関係している。この値を適切に設定することで波ノイズの影響を軽減しながら指認識を行うことが可能になる。

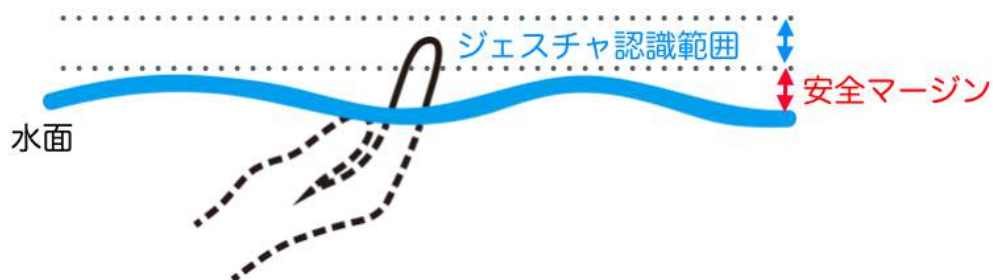


図 7.1: 設定する閾値とジェスチャ認識範囲

7.1.2 実験方法

水面に波の立っていない状態（基準状態）から突き出し、右から左へ流す、左から右へ流す、掬う、激しく水面を叩く5つの動作を行った直後の水面の深度情報を0.2秒毎に5秒間取得する。次に、あらかじめ指定した水面の端4点のXおよびY座標から射影変換を行うことで取得した深度情報を水面のみの深度情報に補正する。この端4点は実験開始時に設定し、各ジェスチャの評価実験が終了するまで変更しない。最後に、補正した深度情報に対してHSVモデルの色相値（Hue）を深度情報で正規化し画像化する。カラーマッピングは基準状態の深度情報を0mmとし、この位置からの深度情報の差分値（Depth Diff Value）に対して行う。深度情報の差分値である入力値は-15を最小値、15を最大値とする-15から15の範囲、色相値である出力値は240を最小値、0を最大値とする240から0の範囲とする。入力値が指定の範囲外の値を取る場合は、次の式によって補正される。

$$InputValue = \begin{cases} InputMax, & \text{if } DepthDiffValue > InputMax \\ InputMin, & \text{if } DepthDiffValue < InputMin \\ DepthDiffValue & \end{cases}$$

色相値と深度データ値の関係は線形であり、

$$Hue = \frac{InputValue - InputMin}{InputMax - InputMin} * (OutputMax - OutputMin) + outputMin$$

を用いて求められる。図7.2に使用するカラーバーとその対応関係を示す。また、実験アプリケーションのフレームレートは50fpsに設定しているため、0.2秒は10フレームに相当する。なお、深度情報の取得タイミングはアプリケーション内部の経過時間ではなく、経過フレーム数で行う。フレームレートの誤差は±0.3fps以内であるため各画像前後の最大誤差は2.4ミリ秒である。

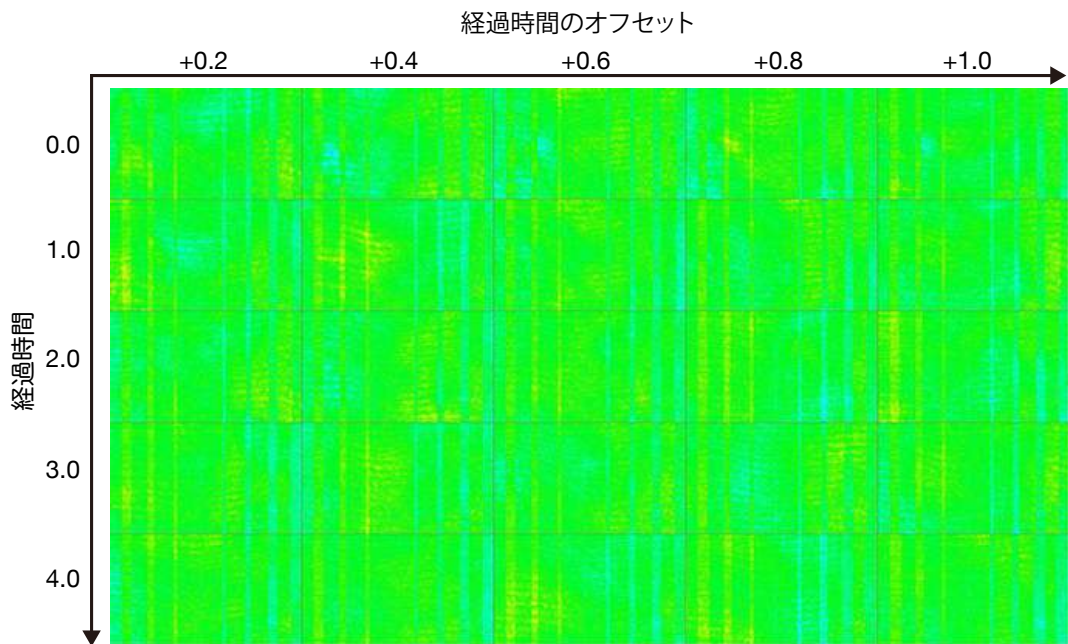


図 7.3: 突き出し動作直後 5 秒間の水面の状態

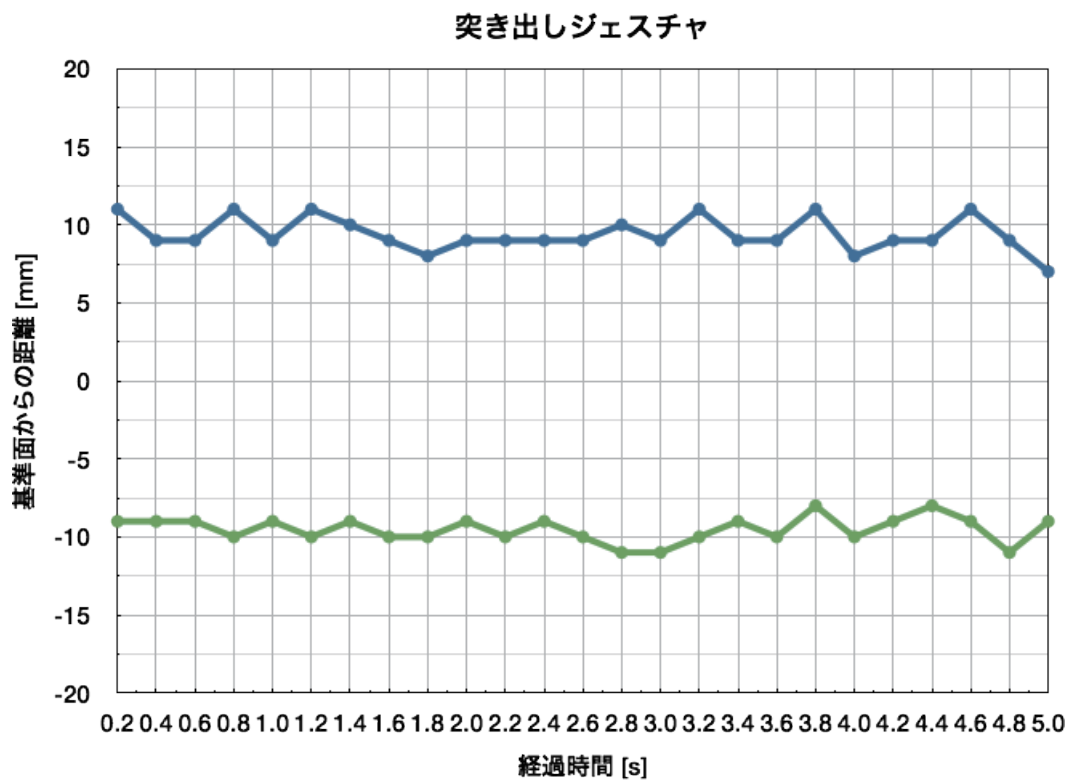


図 7.4: 突き出し動作における経過時間と基準面からの距離の関係

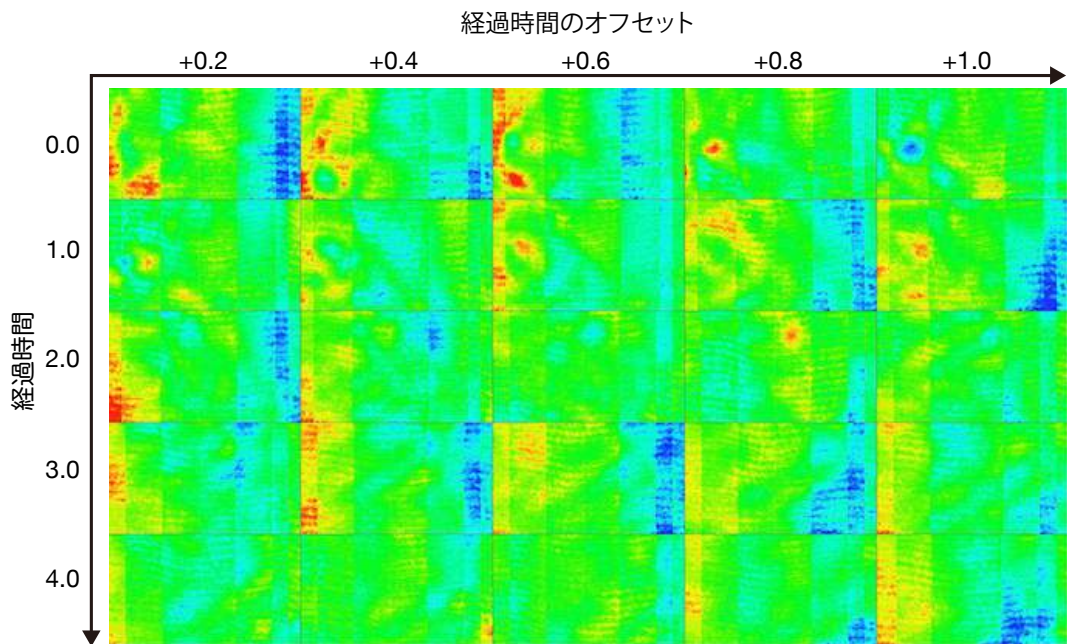


図 7.5: 右から左へ流す動作直後 5 秒間の水面の状態

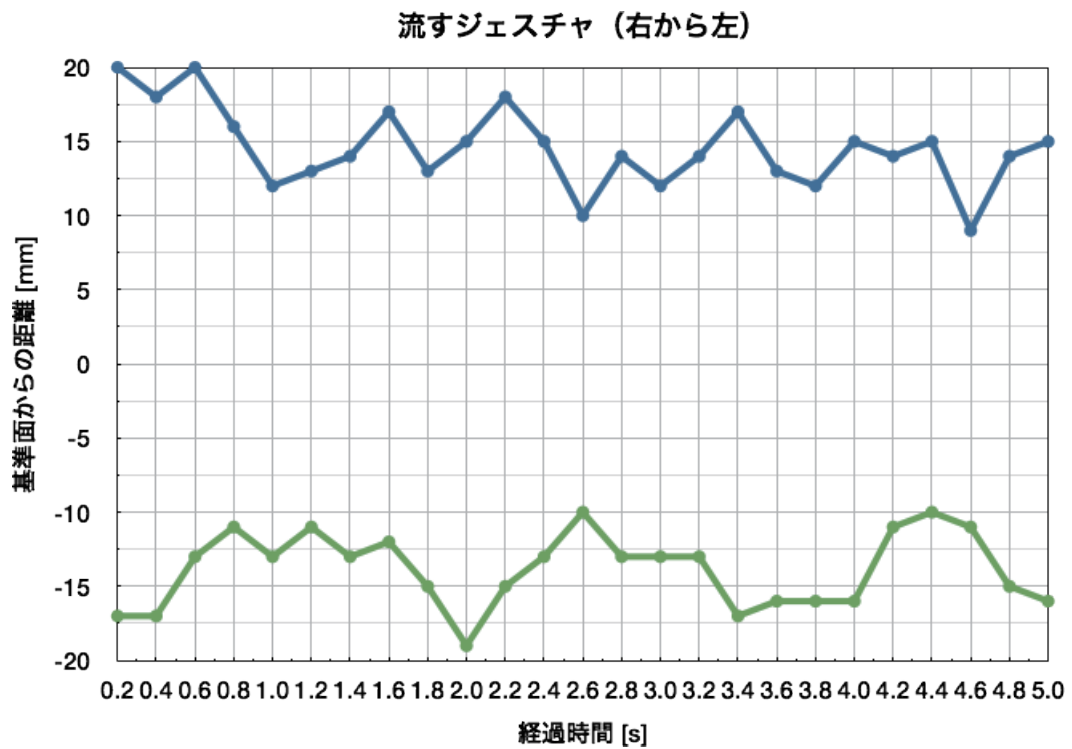


図 7.6: 右から左へ流す動作における経過時間と基準面からの距離の関係

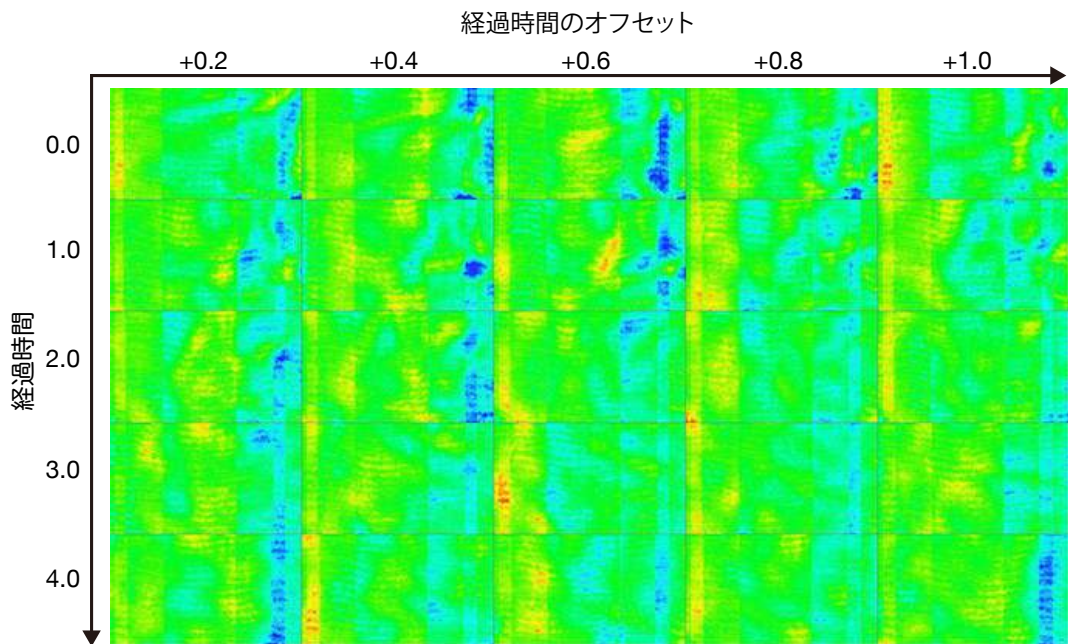


図 7.7: 左から右へ流す動作直後 5 秒間の水面の状態

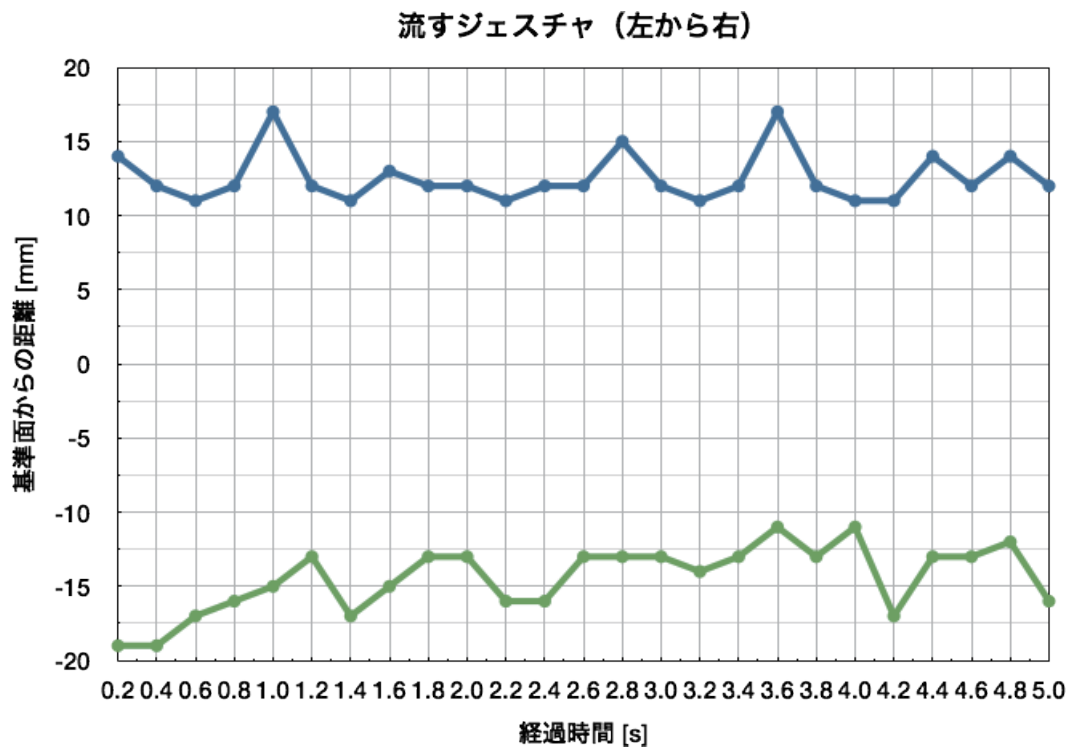


図 7.8: 左から右へ流す動作における経過時間と基準面からの距離の関係

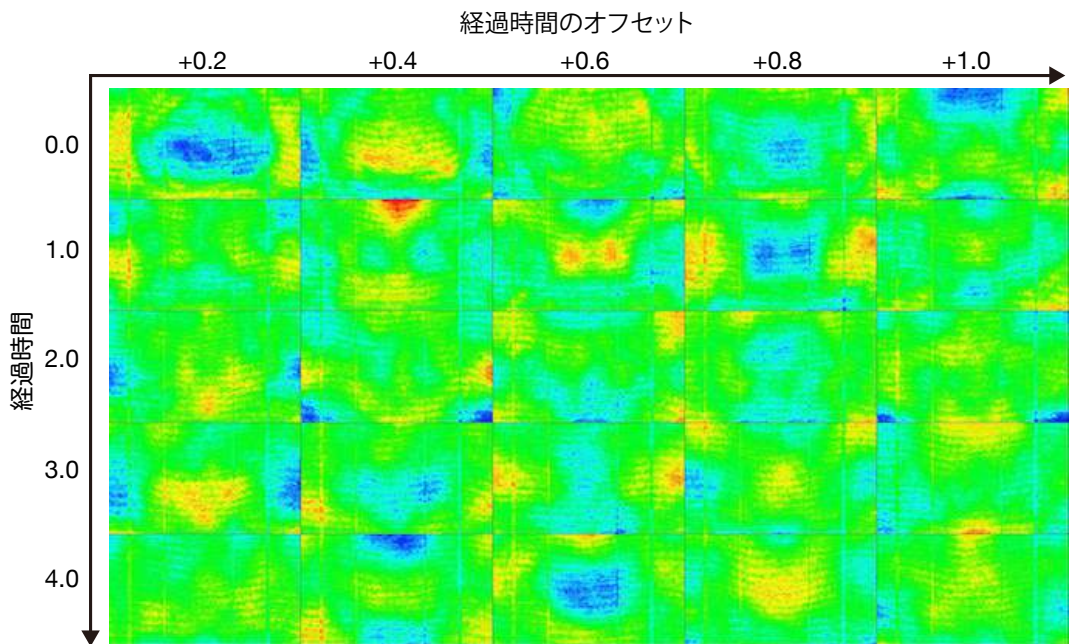


図 7.9: 掬う動作直後5秒間の水面の状態

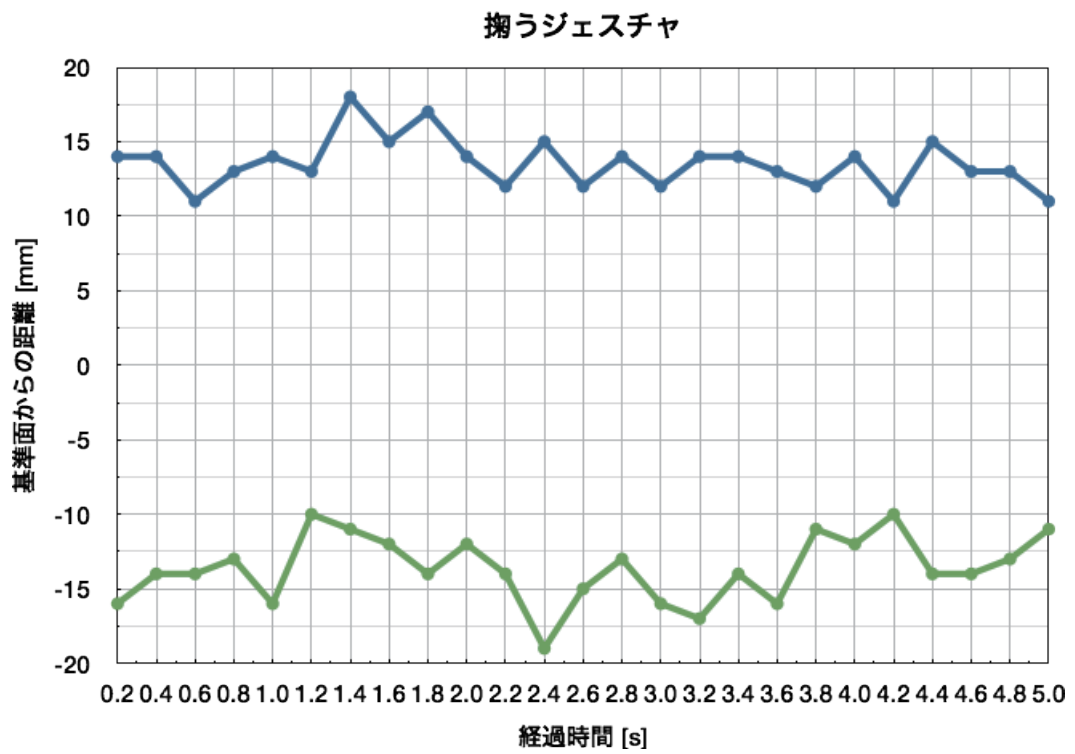


図 7.10: 掬う動作における経過時間と基準面からの距離の関係

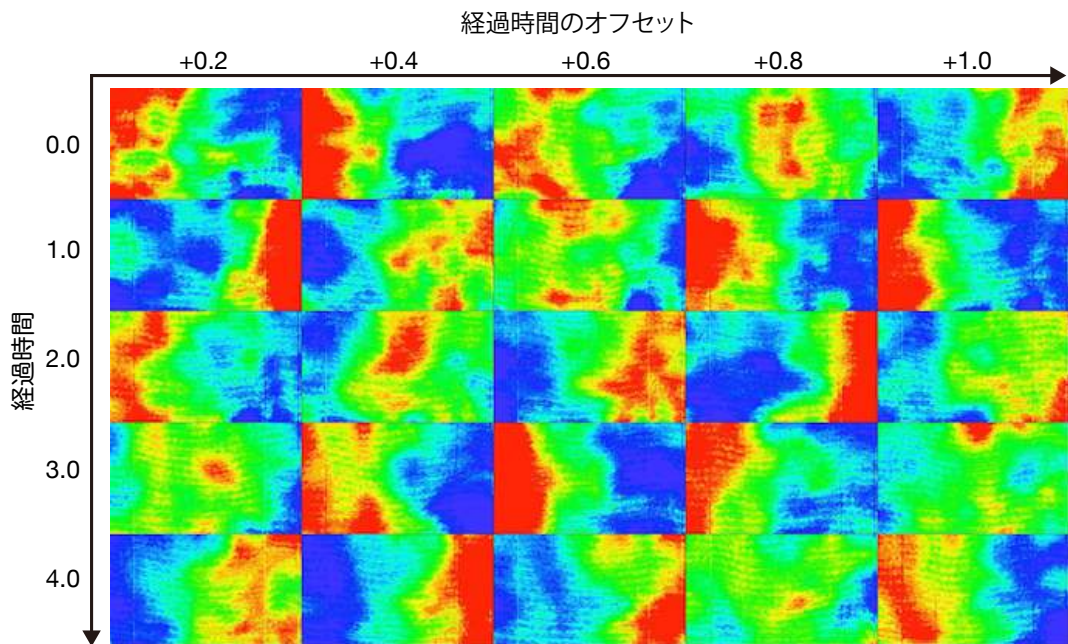


図 7.11: 激しく水面を叩く動作直後5秒間の水面の状態

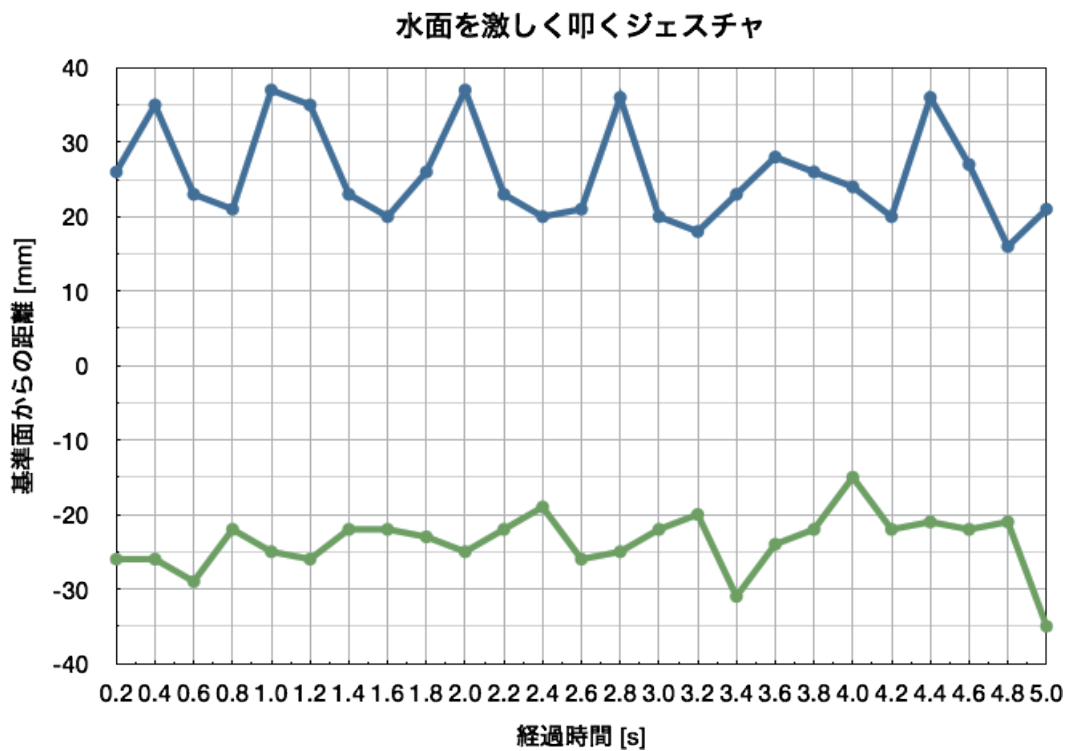


図 7.12: 激しく水面を叩く動作における経過時間と基準面からの距離の関係

7.1.5 実験結果に対する考察

各ジェスチャの実験結果に対する考察は次のとおりである。

突き出し動作

図 7.3 より、突き出し動作は水面の状態をほとんど変化させないことがわかる。図 7.4 のグラフより、基準面から最大 11mm、最小-11mm、平均 9.4mm の変化だとわかる。したがって、突き出し動作によって発生する波ノイズが認識システムの認識率に及ぼす影響は無いと言える。

右から左へ流す動作

図 7.5 より、右から左へ流す動作は水面の状態を浴槽両端で大きく変化させるが、中央付近は変化は小さいことがわかる。浴槽両端の基準面からの距離の変化が大きくなる原因は、波が浴槽両端で跳ね返るためだと考えられる。また、波ノイズは比較的大きいノイズで出現し、0.2 秒後にはノイズの形が大きく変化することがわかる。図 7.6 のグラフより、基準面から最大 20mm、最小-19mm、平均 14.6mm の変化だとわかる。したがって、右から左へ流す動作によって発生する波ノイズは浴槽両端の認識システムの認識率に影響及ぼす可能性があるが、主に操作を行う中央付近の認識率にはほとんど影響を及ぼさないと言える。

左から右へ流す動作

図 7.7 より、左から右へ流す動作も右から左へ流す動作と同様に水面の状態を浴槽両端で変化させるが、中央付近の変化は小さいことがわかる。図 7.8 のグラフより、基準面から最大 15mm、最小-19mm、平均 12.56mm の変化だとわかる。したがって、左から右へ流す動作によって発生する波ノイズが認識システムの認識率に及ぼす影響は右から左へ流す動作と同様なものになると考えられる。

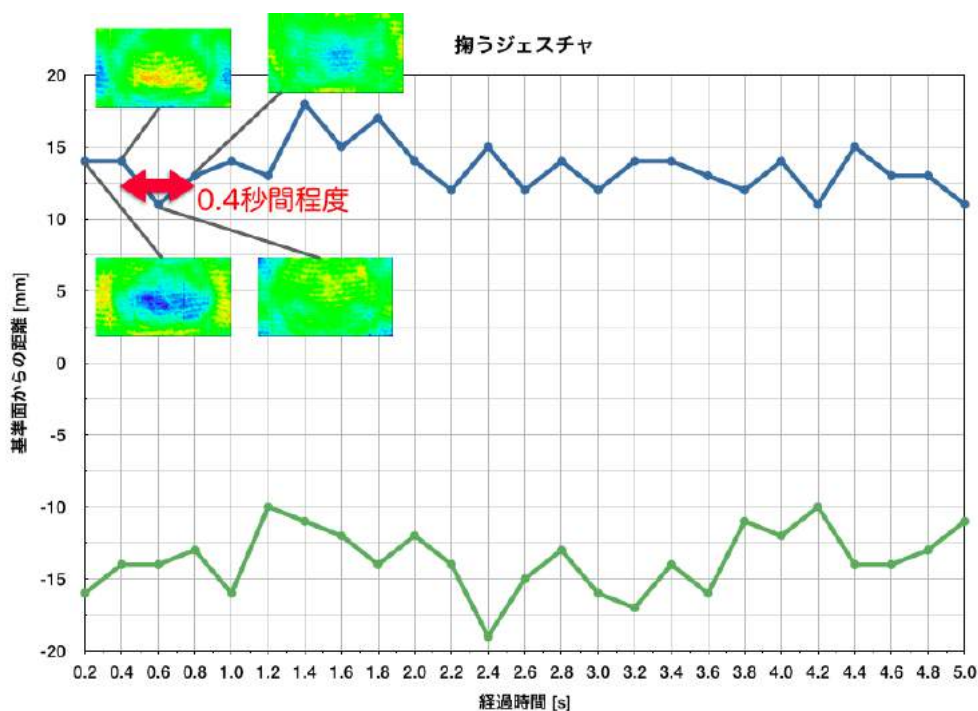


図 7.13: 掬うジェスチャによる水面変化の様子

掬う動作

図 7.9 より、掬う動作は水を掬った地点から円状に水面の状態を変化させることがわかる。基準面からの距離の変化は掬い上げた地点（本実験では浴槽中央）と浴槽両端で大きくなることもわかる、また、0.2 秒から 0.4 秒間で波ノイズは大きく変化している。図 7.13 に示すように 0.2 秒後の水面の状態は、中央付近が基準面より低く、浴槽両端が高くなっている。それに比べて、0.4 秒後は中央付近が基準面より高く、浴槽両端が低くなり、高低差が反転していることから、比較的快速で水面が変化していることがわかる。図 7.10 のグラフより、基準面から最大 17mm、最小-17mm、平均 13.52mm の変化だとわかる。したがって、掬う動作によって発生する波ノイズは局所的に見ると認識システムの認識率に影響を及ぼす可能性あるが、波ノイズの変化速度が早いため、致命的な影響を及ぼす可能性は低いと考えられる。

激しく水面を叩く動作

図 7.11 より、激しく水面を叩く動作は水面の状態は全体的に激しく変化させていることがわかる。他の動作と異なり、波ノイズの面積も大きく変化速度も遅いことがわかる。図 7.12 のグラフより、基準面から最大 37mm、最小-35mm、平均 25.68mm の変化だとわかる。したがって、激しく水面を叩く動作によって発生する波ノイズが認識システムの認識率に及ぼす影響は甚大で波ノイズが収まるまでは正確に認識することは不可能だと考えられる。

全実験結果の考察

激しく水面を叩く動作は以外は水面の変化速度や変化状況から認識システムの認識率に影響を及ぼす影響は低く、深度情報に対して取得する深度情報の範囲や適切な画像フィルタを適用することによって、ある程度の波ノイズを消去できると考えられる。激しく水面を叩く動作の場合は、波ノイズが収まるまで認識システムを停止させ、誤認識を防ぐ必要があると考えられる。突き出し動作は激しく水面を叩く動作以外によって発生する波ノイズの影響も考慮して、水面から 14mm 程度突き上げた地点から認識を開始すると波ノイズの影響を抑えて認識できると考えられる。なお、閾値である 14mm は、流す動作と掬う動作の基準面より上の水面変化の平均値 13.56mm から算出したものである。

7.2 掬う動作の認識精度の調査

本評価実験は掬う対象の大きさにおける掬うジェスチャ精度をを調査する。

7.2.1 実験目的

本評価実験の目的は掬うジェスチャの認識率が掬う対象であるプロジェクトによって投影されている動画像や Web ページ（以下、投影オブジェクト）の大きさ

の関係进行调查することにより、掬うジェスチャが正しく機能する投影オブジェクトの最小サイズを調査することである。ジェスチャが正しく機能するとは、ジェスチャを行った際、手の平に投影オブジェクトが収まっている状態のことである。

7.2.2 実験方法

被験者には実験の意図や掬うジェスチャの使用方法は伝えず、水面にプロジェクションされている画像を掬うことだけを指示した。読み込む画像（基準画像）の大きさは800x600pxであり、プロジェクタから投影される画像の大きさは40x30cmである。実験で使用する画像の大きさは基準画像を含め5種類である。残りの4種類の画像の大きさは基準画像から20%ずつ大きさを縮小したものである。実験の流れについては、被験者に画像の大きさ毎に5回、計25回の掬うジェスチャを行う。実験終了後に実験の意図を伝え、自由記述で掬うジェスチャの使用感や実験の感想を記入してもらう。

7.2.3 実験環境

図7.14に実験風景を示す。被験者は5人であり、そのうち高橋らの前システムを体験したことがある被験者が3人含まれている。プロジェクタのレンズ面と水面との高さは最小1030mm、最大1300mmであり、プロジェクタの解像度は1440x900px、輝度は2000lmである。プロジェクタから投影される領域の大きさは縦幅450mm、横幅700mmである。照明環境は天井の蛍光灯を消灯した薄暗い部屋で行った。

7.2.4 実験結果

実験結果を図7.15に示す。縦軸は成功率、横軸は使用した画像が水面上にプロジェクションされた時の大きさである。画像の大きさが40x30cmから16x12cmまでの範囲では掬うジェスチャは9割程度成功した。しかし、大きさが8x6cmにな



図 7.14: 実験風景

るとジェスチャの成功率は5割程度となり精度が極端に悪くなった。自由記述で回答された意見は次のとおりである。初めて掬うジェスチャを行った被験者は、実際に画像を掬っているように感じた、見た目は小さい画像の方が掬い易いと思ったが実際は大きい方が掬い易く違和感があったという意見があった。高橋らの前システムを体験したことがある被験者は、本システムの方がスムーズに掬え、明らかにジェスチャ認識の精度が向上した。また、片手でも掬えるとより自然に見えるという意見もあった。

7.2.5 実験結果に対する考察

実験結果より特定の大きさまでジェスチャの成功率が良好であったのは、掬うジェスチャの認識範囲より画像の大きさが大きいからだと考えられる。一部の被験者がこの大きさで掬うジェスチャに失敗した原因については掬い上げる動作が早過ぎたからだと考えられる。実験における画像の最小サイズである 8x6cm にお

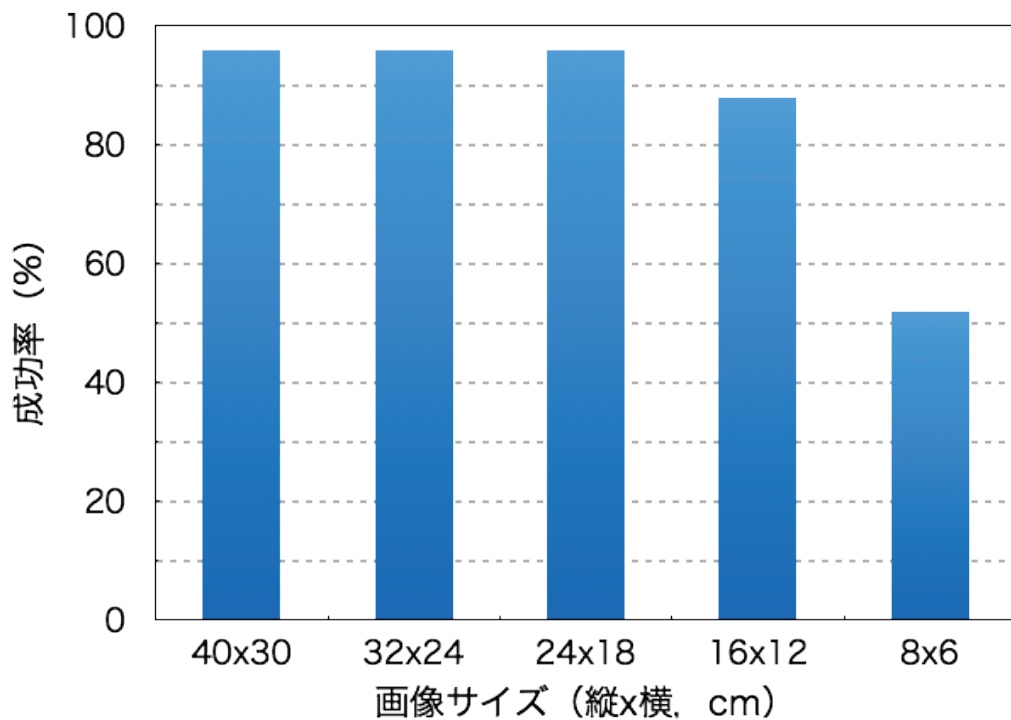


図 7.15: 画像の大きさにおける掬うジェスチャの成功率

いて、掬うジェスチャが正しく機能しなかった原因については、掬うジェスチャのイベントトリガーは両手を掬う形にした時のおおよそ中心に点として存在するため、両手が画像に触れていても掬い上げる時にイベントトリガーが画像内に存在せずジェスチャが正しく機能しなかったからだと考えられる。この問題点を解決するには掬うジェスチャのイベントトリガーを点から面に変更することで解決すると考えられる。なぜなら、面としてイベントトリガーを扱うことで、掬い上げる時に両手の一部が画像に触れるだけで掬うイベントを発生させることが可能になるからである。また、掬うジェスチャが安定に動作する画像の最小サイズは、16x12cm 程度だと言える。この最小サイズをデスクトップアプリケーションで使用する投影オブジェクトの最小サイズにも適用することでアプリケーション全体で安定したジェスチャ操作が可能になると考えられる。



図 7.16: デスクトップアプリケーションの展示の様子

7.3 デモンストレーション展示から得られた知見

アメリカ・ロサンゼルスで開催された国際学会である ACM SIGGRAPH 2013 Emerging Technologies (SIGGRAPH2013), 日本科学未来館で開催された経済産業省および一般財団法人デジタルコンテンツ協会主催の Digital Content Expo 2013 (DCEXPO2013) やテレビ番組で本システムのデモンストレーションを行った際に得られた知見について述べる。

7.3.1 デスクトップアプリケーションの展示から得られた知見

図 7.16 にデスクトップアプリケーションの展示の様子を示す。投影オブジェクトの移動, 回転, 拡張を行うためのジェスチャである突き出しに関しては, 投影オブジェクトを水面の下から突き刺して操作することはコンピュータの操作知識が

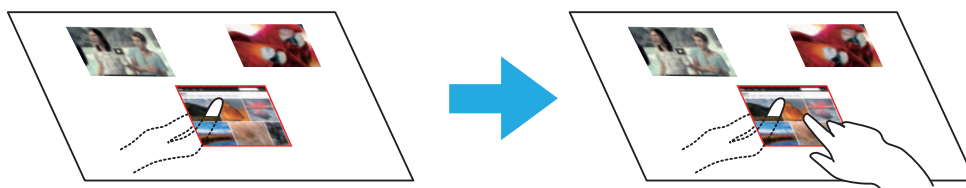


図 7.17: 移動, 回転, 拡縮ジェスチャの改良案

少ない人にとっても自然な動作であるという意見を貰えた。一方、スマートフォンやタブレット機器などを普段から使用している操作知識の多い人は、突き出しジェスチャは新鮮であるが、図 7.17 に示すように水面の下からの突き出しジェスチャによる操作に加えて、水面の上からのタッチジェスチャでも移動, 回転, 拡縮を行えた方が良いという意見が多かった。実際の展示でも、操作説明を受けていない人は先ず一般的なタッチパネルのように水面の上から操作を積極的に行っていた。また、突き出しジェスチャの認識率は指の細い体験者や指を突き出す動作自体に慣れていない体験者は上手く認識できない問題点があった。特に、水面の下からの指の認識に必要な高さまで指を突き出すことができない、上手く指に角度を付けることができない体験者が突き出しジェスチャを失敗する傾向が見られた。突き出しジェスチャの認識率は、深度カメラを真上付近ではなく、斜め方向に撮影するように設置することで向上すると考えられる。なぜなら、斜め方向から撮影することにより突き出した指の側面を撮影することができ、深度カメラで認識する指の領域を大きくできるからである。これにより、指の細い体験者でも安定して突き出しジェスチャを認識できると考えられる。回転や拡縮操作の認識精度はおおむね良好であったが、操作対象の投影オブジェクトの大きさが小さすぎると突き出す行為自体が不可能になったり、動作が安定しないことがあった。これは、投影オブジェクトの最小サイズを予め指定することで解決できると考えられる。

流体シミュレーションと流すジェスチャについては、水を流す際に発生する触覚的なフィードバックが作用して投影オブジェクトを操作している感覚が得られるため水を使ったディスプレイと親和性が高いという意見が多かった。また、流

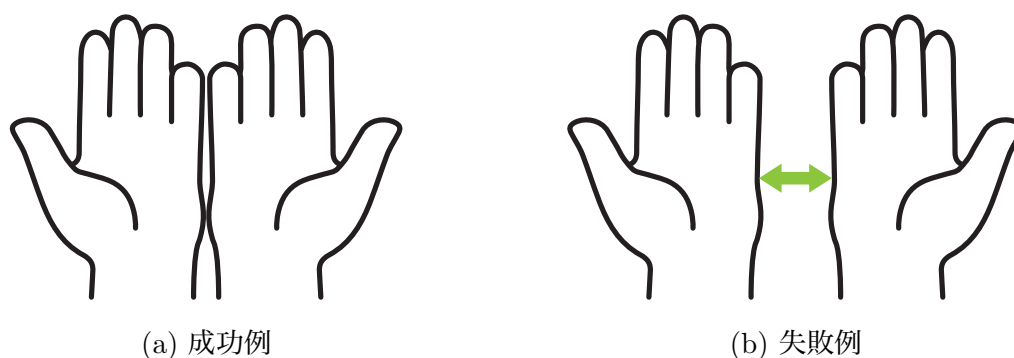


図 7.18: 拵うジェスチャにおける手の形の例

すジェスチャ応用した Web ページ用のジェスチャは従来のタッチパネルの Web ブラウザ操作と類似していることから詳しい説明を行わなくても操作ができる体験者が多かった。

拵うジェスチャについては、投影オブジェクトを拵い上げ別の場所に移動させる動作が水の特性を上手く利用できているという意見がもられた。特に、拵った際に投影オブジェクトが手の平サイズに縮小される点や手の平の中で動画が再生される点が好評であった。改良意見としては、拵い上げている途中でも、手の平の水が無くなった際は投影オブジェクトも保持している水と共に水面に落ちる画像効果がある方が良いという意見があった。また、拵うジェスチャの認識率はおおむね良好であったが、手のサイズが小さい体験者、ジェスチャの形が上手くできない体験者や投影オブジェクトが手の平より小さい場合は上手く動作しないことが多かった。図 7.18 に拵うジェスチャの手の形における成功例と失敗例を示す。上手く拵うジェスチャができない体験者は図 7.18b のように左右の手の平の間隔が広い傾向があった。そのほかの失敗例としては、体験者が拵った投影オブジェクトを覗き込むことによって、体験者の頭部が深度カメラに映り込み、手の平自体を認識ができなくなることで起こった。オクルージョンによる拵うジェスチャの認識率の低下は深度カメラの位置を体験者が拵った投影オブジェクトを覗き込む動作を行っても、体験者の頭部が深度カメラに映らない位置に設置できれば解決すると



図 7.19: ゲームアプリケーションの展示の様子

考えられる。

7.3.2 ゲームアプリケーションの展示から得られた知見

図 7.19 にゲームアプリケーションの展示の様子を示す。指を突き出すことにより指の周りに魔法陣エフェクトが表示され、指を飛ばしたい方向へ倒すことにより魔法陣が発射させることについては、単純な操作であるが実際に魔法を使っている気分になった、入力と水の特徴を上手く利用しているという意見が得られ、深い没入感を体験者に提供することができたと言える。

水中に沈めたスピーカーシステムについては、水が飛び跳ね実際に爆発を起こしているように感じる、水しぶきが幻想的という意見が得られた。一方で、水しぶきが激しいため水槽の周りが水浸しになり体験者が足を滑らす危険性があるという意見もあった。これは、スピーカーの周波数や水しぶきを出す方向を調整することで解決できると考えられる。

7.3.3 得られた知見に対するまとめおよび考察

デモンストレーションの際に体験者から得られたコメントから、提案したアプリケーションは操作が直感的で水の特徴を上手く利用していると考えられる。また、実際の浴槽、温泉やプールなどの公共の場所で使用したいという意見が多く得られた。温泉やプールなど大規模な場所に本システムを利用するには1台の深度カメラでは画角や解像度や照明の関係上、実現不可能である。そのため、複数台の深度カメラを利用する、深度カメラを利用できる環境を構築するなど課題が多いと考えられる。また、テレビ局の展示ではスタジオ照明の種類により深度カメラから取得する深度情報に甚大な影響を及ぼすものがあつた。リハーサル時の照明では正しく動作していたシステムでも、本番用の照明では深度情報が全く取得できない問題があつた。これは本番で使用されている照明の光度が高いまたは、水銀照明を使用していたため深度カメラから投影される赤外線パターンに何らかの悪影響を及ぼしたのだと考えられる。これらの照明に関する問題は実際に展示を行う場所へ行かなければ分からないが、プールなどの屋外で本システムを使用するためには照明問題も解決しなければならない。

第 8 章

考察

本章では本システムの考察と問題点を述べる。その後、本認識システムの実環境への利用に関する考察および今後の展望について述べる。

8.1 現認識システムの考察と問題点

本論文では 3 章で水面を利用したディスプレイにおける手指およびジェスチャ認識の要求事項を以下のように述べた。

- 認識システムとアプリケーションを分離すること
- システム動作速度を向上させること
- 認識システムの処理状況を常に確認できること
- システム起動後でも細かい認識パラメータ調整ができること
- 各種ジェスチャの認識精度を向上させること

この各項目について認識システム設計、評価実験、デモンストレーションで得られた知見の結果から述べる。

認識システムの動作速度

図 8.1 に示すように、本認識システムは単独で動作するように設計することで、認識システムとアプリケーションを別々の PC で動作させることが可能

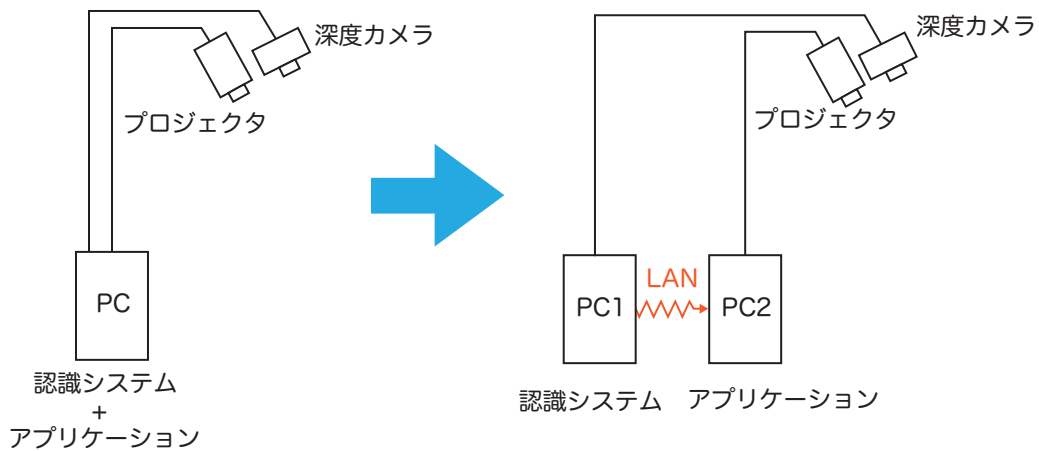


図 8.1: 複数の PC を用いたシステムの構成

になりシステムに掛かる負荷を小さくすることができた。これにより、システム動作を安定した状態で向上させることが可能になった。また、単独で起動できるようにしたことで、アプリケーションに依存しない認識システムが構築できたとと言える。

認識システムの認識精度と使いやすさ

本研究の目的の1つである認識精度の向上は、高橋らの前システムに比べて飛躍的に良好になったと考える。特に、掬うジェスチャは前システムでは単純な閾値処理であったため、掬う対象が小さいと上手く掬えなかった、しかし、本システムでは面積の増加量、認識領域の輪郭情報からジェスチャ補正をすることで掬う対象がある程度小さくても90%前後の精度で掬うことができるようになった。タッチジェスチャに関しては、高橋らの前システムではタッチを行いたい位置まで行く前にタッチイベントが発生してしまいユーザーの意図した位置でタッチが困難であった。しかし、イベントトリガーなどを見直し開発したタッチジェスチャは水面でキーボード操作やWebページを操作できる程度の精度まで向上した。また、その他のジェスチャ認識もジェスチャ別に認識パラメータをスライダーで細かく設定できるため認識精度が向上したと言える。

本認識システムのユーザーインターフェースは、キャリブレーション領域設定が容易な点や認識パラメータを各種スライドバーおよびボタンを調整しその結果を視覚的に表示する設計にした。これにより、システム起動後でも細かいパラメータ調整がプログラムの知識が無い人でも行えた点から前システムに比べて、認識システムの使いやすさは向上したと考えられる。

水面に特化したジェスチャ

本システムでは水面に特化したジェスチャの制御と認識方法を提示した。中でも、掬うジェスチャと流すジェスチャは水を掬う、流すという浴室環境では自然な動作とコンピュータの操作を上手く紐付けられたと考える。これは、学会やテレビ番組でデモンストレーションを行った時の反響の大きさからも見て取れる。また今後は、浴槽内から浴槽外へのジェスチャによるインタラクションも開発する必要があると考えられる。例えば、浴槽内で掬うジェスチャを行い投影オブジェクトを掬い、その掬った投影オブジェクトを浴槽外へ水と一緒に投げ捨てることで投影オブジェクトの削除を行うなどである。このように、浴室環境全体を使ったジェスチャを取り込むことでより自然な情報操作が可能になると考えられる。

波ノイズに起因する誤認識

突き出しジェスチャは波のノイズの影響を特に受けることが分かっている。現認識システムでは、突き出し領域を検出後、0.4秒以上その領域が存在する場合のみ突き出しジェスチャとして認識するようにしているが、これは認識速度が0.4秒遅延すると同意である。実際に、ファイル操作などを行う時、この遅延は操作性観点から致命的な問題点だと考えられる。これを解決するためには、波のノイズ除去の手法を変更し、意図的な遅延がないジェスチャ認識を行う必要があると考えられる。また、水面に激しく波を立てた際は認識不能になることが多い問題点については、発生する波ノイズを予想し認識

不能状態を回避し、それが困難な場合は警告を出しユーザーに現在の水面の状況を提示する必要があると考えられる。なぜなら、突然認識システムが機能停止するのはユーザービリティの観点から好ましく無いからである。

8.2 実際の浴室環境や公共のプールへの実使用に関する考察と今後の展望

本節では実環境で本システムを用いる際の問題点とその考察について述べる。

表示品質と白濁剤濃度

本研究を通して、白濁剤を融解させた水への映像投影は可能であると分かった。しかし、現状では文字を視認するには白濁剤を通常の数十倍の濃度で融解させなければならない問題点がある。実際の浴室環境で本システムを適用するには、通常白濁剤（入浴剤）の濃度もしくは白濁剤を未使用で文字を視認できる程度の表示品質を保つ必要があると考えられる。

実際の浴室での利用

本研究ではプロトタイプシステムとして、実際の浴槽を利用して操作精度や使用感を検証した。しかし、浴槽外からの操作しか行っていない。実利用を検証するには、実際に浴槽に入り、操作精度や使用感を検証する必要があると考えられる。特に、浴槽外内では視線位置や手の可動範囲異なるため、提案するジェスチャが浴槽に入った状態で適切かどうかを再考する必要がある。また、浴槽内から浴槽外へのジェスチャや風呂桶などの実物体の利用も効果的操作になりうると考えられる。例えば、現在の投影コンテンツの削除方法は3本以上の指を突き出して、一気に指を沈めるジェスチャであるが、投影オブジェクトを掬うジェスチャや風呂桶で掬い、浴槽外へ投げ捨てるようなジェスチャに置き換える。水を掬い捨てる動作は浴室環境では自然な行為か

つ直感的な操作であると共に、没入感向上も期待できると考えられる。

浴室環境でのカメラ使用

浴室環境でのカメラ使用は倫理的に問題が、現認識システムではキャリブレーションを行う時に RGB 画像を取得している。実際の浴室環境、特に銭湯やプールなどの公共の場所で利用する場合は深度カメラのセンサー情報のみでキャリブレーションを行う必要があると考えられる。さらに、浴室環境は密閉された空間であり、水では無くお湯を使用するため浴室内の室温や湿度が非常に高くなっている。そのため、深度カメラのレンズ部分の曇り、結露が発生する可能性がある。したがって、本システムを実際の浴室環境に適応する前に、先ず深度カメラにとって悪条件の環境で使用に耐えうるのかを調査する必要がある。

銭湯やプールへ利用

本研究では、大規模化に向けてゲームアプリケーションを試作した。その結果を踏まえて考察すると、銭湯やプールのような広い場所で本システムを利用するには、ハードウェア構成を根本的に変更する必要があると考えられる。なぜなら、25m プールなどで本システムを利用すると演算処理用 PC と深度カメラの数が膨大になり、認識システムの制御が困難になるからである。解決案としては、先ず、広角かつ超解像度のハイスピードカメラを1台もしくは数台用いてプール全体を取得する。その後、認識対象付近を切り出して個別に画像処理するなどの手法が考えられる。また、同時使用人数の問題も考えられる。デモンストレーションは、基本的に1人、多くても同時に3人程度で本システムを体験してもらった。その時、体験者は説明員の指示に従い、操作を行ったため複数人の使用でも予想外の行動をする人が少なかった。しかし、公共の場所での実利用となると、利用者が予想外の行動をする場合や同時使用人数の増加が発生すると予想される。これらを解決するためには、

ジェスチャ認識のエラー処理だけでなく、利用者間の関係も認識する必要があると考える。利用者間の関係とは、例えば、手として認識された領域が誰の手領域かを検出および認識することである。

第 9 章

結論

本論文では、提案する水面ディスプレイを利用した時における入浴時の情報閲覧および操作に最適な操作方法や情報提示手法を示し、実験や、提案する認識手法を用いた応用例を学会やテレビ番組に展示し得られた知見から評価した。

1 章では、本研究の社会的背景と本論文の目的と概要を述べた。

2 章では、本論文の関連研究を述べた。関連研究では水を利用したディスプレイや浴槽環境におけるインタラクティブシステムを挙げ、その問題点を示した。

3 章では、前 AquaTop System の概要を説明し、その認識システムと情報提示手法の問題点を示し、本研究で提案した手法がその解決案になることを述べた。

4 章では、本研究で提案する認識システムのフレームワークである AquaTop Core Vision の詳細について述べた。

5 章では、認識システムのトラッキングデータ送信に利用する通信プロトタイプや水環境特性を上手く利用するために用いた Application Framework について述べた。

6 章では、本研究で提案する認識システムを用いた応用例を述べた。さらに、大規模化に向けた認識システムの問題点を挙げ、その解決案を示した。

7 章では、誤認識の原因の 1 つである波ノイズを視覚化し、その影響を少なくする解決案と掬うジェスチャにおける認識精度について述べた。

8 章では、3 章で示した AquaTop Display の問題点に対する解決案の達成度と本システムを実環境で利用するための考察を述べた。

本論文では，前 AquaTop Display の認識手法の問題点に対する解決案や浴室環境での情報提示手法を述べたが，実験や展示によって得られた意見から，提案した認識手法は前認識システムより精度が向上していると言える．実際の浴室環境向けに利用するには現在の認識システムではいくつか問題点が存在するが，新しい入浴時のコンテンツとしては期待できる．

参考文献

- [1] 株式会社博報堂 D Y ホールディングス. 第 8 回 全国スマートフォンユーザー 1000 人定期調査. <http://www.hakuhodo.co.jp/archives/newsrelease/15354>.
- [2] パナソニック株式会社, エコソリューション社. 入浴の目的・スタイルなどに関する実態調査. http://sumai.panasonic.jp/sumai_create/enq/002/.
- [3] Peter C. Barnum, Srinivasa G. Narasimhan, and Takeo Kanade. A Multi-layered Display with Water Drops. *ACM Trans. Graph.*, Vol. 29, No. 4, pp. 76:1–76:7, 2010.
- [4] Julius Popp. bit.fall. <http://juliuspopp.de/html/bitfall.php>.
- [5] 杉原有紀, 舘暲. かぶり型水ディスプレイの開発. 日本バーチャルリアリティ学会論文誌, Vol. 6, No. 2, pp. 145–152, 2001.
- [6] Paul H. Dietz, Jonathan Westhues, John Barnwell, Jefferson Y. Han, and William Yerazunis. Submerging Technologies. In *ACM SIGGRAPH 2006 Emerging Technologies*, 2006.
- [7] Jefferson Y. Han. Low-cost Multi-touch Sensing Through Frustrated Total Internal Reflection. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, pp. 115–118, 2005.
- [8] Erika Okude and Yasuaki Kakehi. Rainterior: An Interactive Water Display with Illuminating Raindrops. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pp. 270–271, 2011.
- [9] Ismo Rakkolainen, Stephen DiVerdi, Alex Olwal, Nicola Candussi, Tobias

- Hüllerer, Markku Laitinen, Mika Piirto, and Karri Palovuori. The Interactive FogScreen. In *ACM SIGGRAPH 2005 Emerging Technologies*, 2005.
- [10] 八木明日華, 井村誠孝, 黒田嘉宏, 大城理. 多視点観察可能なフォグディスプレイ. 情報処理学会シンポジウム論文集, インタラクション 2011, No. 3, pp. 315 – 318, 2011.
- [11] Displair 社. Displair. <http://displair.com/>.
- [12] 遠藤祐二, 稲沢綾二, 前田広一朗, 板井志郎, 三輪敬之. 霧スクリーンの多層構造化による 3 次元的ディスプレイ装置の開発. ヒューマンインタフェースシンポジウム 2010DVD-ROM 論文, pp. 431 – 434, 2010.
- [13] Mark R. Mine, Dustin Barnard, Bei Yang, and Daniel Baker. Thermal Interactive Media. In *ACM SIGGRAPH 2011 Emerging Technologies*, pp. 18:1–18:1, 2011.
- [14] Antti Virolainen, Arto Puikkonen, Tuula Kärkkäinen, and Jonna Häkkinen. Cool Interaction with Calm Technologies: Experimenting with Ice As a Multitouch Surface. In *ACM International Conference on Interactive Tabletops and Surfaces*, pp. 15–18, 2010.
- [15] 平井重行, 櫻文喜, 藤井元. 日常生活の入浴行動をより楽しめる浴室を目指して. 情報処理学会研究報告, pp. 1 – 8, 2006.
- [16] 榊原吉伸, 林宏憲, 平井重行. TubTouch: 湯水の影響や自由形状への適用を考慮した浴槽タッチ UI 環境. 情報処理学会論文集, Vol. 54, No. 4, pp. 1538 – 1550, 2013.
- [17] Shigeyuki Hirai, Yoshinobu Sakakibara, and Seiho Hayakawa. Bathcratch: Touch and Sound-based DJ Controller Implemented on a Bathtub. In *Pro-*

-
- ceedings of the 9th International Conference on Advances in Computer Entertainment*, pp. 44–56, 2012.
- [18] Yoichi Takahashi, Yasushi Matoba, and Hideki Koike. Fluid Surface: Interactive Water Surface Display for Viewing Information in a Bathroom. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, pp. 311–314, 2012.
- [19] 的場やすし, 高橋陽一, 徳井太郎, 小池英樹. お風呂ディスプレイ: 浴槽型没入映像体験システム. In *Workshop on Interactive Systems and Software*, pp. 239 – 240, 2012.
- [20] Yasushi Matoba, Yoichi Takahashi, Taro Tokui, Shin Phuong, and Hideki Koike. Aquatop display. In *Proceedings of the Virtual Reality International Conference: Laval Virtual*, pp. 24:1–24:4, 2013.
- [21] Hideki Koike, Yasushi Matoba, and Yoichi Takahashi. AquaTop Display: Interactive Water Surface for Viewing and Manipulating Information in a Bathroom. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, pp. 155–164, 2013.
- [22] 大日精化工業株式会社. <http://www.daicolor.co.jp/>.
- [23] openFrameworks. <http://openframeworks.cc/>.
- [24] PrimeSense 社. OpenNI. <http://www.openni.org/>.
- [25] Intel 社, WillowGarage 社. OpenCV. <http://opencv.org/>.
- [26] NUI Group Community. Community Core Vision. <http://ccv.nuigroup.com/>.
- [27] OpenKinect. libfreenect. <http://openkinect.org/>.

-
- [28] NUI Group Community. TUIO. <http://www.tuio.org/>.
- [29] CNMAT. OSC. <http://archive.cnmat.berkeley.edu/OpenSoundControl/>.
- [30] Awesomium HTML UI Engine. <http://www.awesomium.com/>.
- [31] Chromium. <http://www.chromium.org/>.
- [32] J. Stam. Real-Time Fluid Dynamics for Games. In *Proceedings of the Game Developer Conference*, 2003.

謝辞

本研究を遂行するにあたり、日頃から熱心なご指導、そしてご鞭撻を賜りました小池英樹先生に厚く御礼申し上げます。様々な面で貴重なご指導、ご意見、ご助言を頂きました野嶋琢也先生、佐藤俊樹先生に心より感謝いたします。

研究を遂行するにあたりハードウェア面のシステムを構築して頂いた的場やすしさん、St. Andrews 大学に留学した時にご指導、ご助言を頂きました Aaron Quigley 先生、Miguel Nacenta 先生、Uta Hinrichs さん、留学中、安心して研究が遂行できるように生活面のお世話をして頂いた Michel Brock さんに心より感謝いたします。

最後に、多大なるご協力を頂き、共に研究生活を送ってきた小池研究室、野嶋研究室の皆様心より感謝いたします。

付録 A

発表文献・受賞業績

- [1] Yasushi Matoba, Yoichi Takahashi, Taro Tokui, Shin Phuong, Shingo Yamano and Hideki Koike. AquaTop Display: A True ” Immersive” Water Display System. In *ACM SIGGRAPH 2013 Emerging Technologies*, pp.4:1– 4:1, 2013.
- [2] 小池研究室. アクアトップディスプレイ. DIGITAL CONENT EXPO 2013. Innovative Technologies2013 および Culture 特別賞受賞.
- [3] 的場やすし, 高橋陽一, 山野真吾, 徳井太郎, Shin Phuong, 小池英樹. AquaTop Display. 2013 アジアデジタルアート大賞展 インタラクティブアート部門, 優秀賞受賞.
- [4] Hideki Koike, Shingo Yamano and Yasushi Matoba. An Interactive Surface on the Water. *IEEE Computer Graphics and Applications*, 2014. (投稿中)

付録B

展示業績

- [1] SMAPxSMAP オドロキ隊 (フジテレビ, 2日間)
- [2] ワールドビジネスサテライト トレたま (テレビ東京, 1日間 x2回)
- [3] めざましテレビ (フジテレビ, 1日間)
- [4] Nスタ (TBS テレビ, 1日間)
- [5] Fantastico (TV GLOBO, 1日間)
- [6] DIGITAL CONTENT EXPO (日本科学未来館, 3日間)
- [7] ACM Interactive Tabletops and Surface 2013 (スコットランド・セントアンドリュース, 1日間)
- [8] ACM SIGGRAPH 2013 Emerging Technologies (アメリカ・アナハイム, 5日間)
- [9] Bath-time goes interactive in Japan (Reuters, 1日間)
- [10] Daily Planet (Discovery Canada, 1日間)