

筑波大学大学院博士課程

システム情報工学研究科修士論文

身体に貼るというメタファに基づく  
個人コンテンツ管理インタフェース

大谷裕昭

コンピュータサイエンス専攻

指導教員 田中二郎

2008年3月

## 要 旨

現在、インターネットなどの爆発的な普及により、ユビキタスコンピューティング環境がより身近になってきている。ユビキタスコンピューティングにおいては、利用者はコンピュータの技術を意識しないと言われている。そのため、コンピュータにあまりなじみのない利用者でもコンピュータを簡単に利用する事が出来る。しかし、ユビキタスコンピュータ環境におけるコンピュータ上のデータ管理手法についてはあまり議論はされておらず、未だにフォルダを用いる手法が主流である。

コンピュータの小型化に伴い我々はいつでも色々なコンテンツを楽しむ事が出来る。携帯電話で映画を見たり、デジタルカメラで写真を撮影したり、MP3 プレーヤーで音楽を再生するのはその一例である。手軽に楽しめるようになる反面我々が保有するコンテンツは膨大な量になっている。目的となるコンテンツへのアクセスを容易に出来るように、コンテンツを管理する必要がある。また、小型化されたコンピュータを扱う人はコンピュータに詳しい人やそうでない人など様々である。そのため、コンテンツ管理は誰でも行えるようにする必要がある。

そこで本研究では、コンピュータ上のコンテンツ管理を身体を用いて行うためのインタフェースを実装した。本インタフェースではカメラに映し出された利用者の身体にデータを貼り付けていくことでデータ管理を行う。身体に貼りついているデータは手を用いて操作する事が可能である。このように、自然なインタフェースを利用する事でコンピュータになじみのない利用者でも効率的にデータ管理が行う事が出来ると考えられる。

# 目次

第1章	はじめに	1
1.1	背景	1
1.2	研究の目的と論文構成	3
第2章	柔らかいストレージ	4
2.1	柔らかいストレージとは	4
2.2	柔らかいストレージの利用例	4
2.3	ストレージ技術の隠匿	6
2.3.1	ストレージと個人の対応付け	6
2.3.2	ストレージの存在の錯覚	6
2.4	NSミラーの実装	7
2.4.1	NSミラー概観	7
2.4.2	写真データ表示方法	7
2.4.3	写真の交換	7
第3章	身体に貼り付けるインタフェース	10
3.1	従来のコンテンツ管理手法の問題点	10
3.2	コンテンツ管理の関連研究	11
3.3	インタフェースの提案	12
3.3.1	柔らかいストレージとの相違点	13
3.3.2	実装方針	14
	任意の位置にコンテンツを貼り付ける	14
	指先を用いてのデータ操作	14
	画像処理による身体認識	15
3.4	プロトタイプ利用イメージ	15
3.5	データの操作	15
3.5.1	データの閲覧	17
3.5.2	データの選択	17
3.5.3	データの移動	19
3.5.4	データの貼り付け	19
3.5.5	データの削除	20

<b>第4章</b>	<b>システム利用例</b>	<b>21</b>
4.1	写真管理システム . . . . .	21
4.2	会場入場システム . . . . .	21
4.3	音楽ファイルと歌詞管理 . . . . .	21
<b>第5章</b>	<b>実装</b>	<b>23</b>
5.1	環境・言語 . . . . .	23
5.2	トラッキング . . . . .	23
5.2.1	指先のトラッキング . . . . .	23
5.2.2	身体のトラッキング . . . . .	27
5.3	身体的位置の決定 . . . . .	30
5.3.1	制約 . . . . .	30
5.3.2	頭頂座標の検出 . . . . .	30
5.4	データと身体座標の対応付け . . . . .	32
5.5	データ操作 . . . . .	32
5.5.1	データ表示 . . . . .	32
5.5.2	データ選択 . . . . .	33
5.5.3	データ移動と貼り付け . . . . .	33
5.5.4	データ削除 . . . . .	34
5.6	実験 . . . . .	34
5.6.1	実験1 . . . . .	34
	実験結果 . . . . .	35
	考察 . . . . .	35
5.6.2	実験2 . . . . .	36
	実験結果 . . . . .	36
	考察 . . . . .	36
<b>第6章</b>	<b>まとめ</b>	<b>38</b>
	謝辞	39
	参考文献	40

# 目次

1.1	フォルダでの管理 . . . . .	2
1.2	デスクトップでの管理 . . . . .	2
2.1	持ち運ぶイメージ . . . . .	5
2.2	カメラ型入力アプライアンスとプリンタ型出力アプライアンス . . . . .	5
2.3	ダウンロード . . . . .	6
2.4	NSミラー概観 . . . . .	8
2.5	マーカー . . . . .	8
3.1	ツリー構造 . . . . .	10
3.2	ツリーでの構造表示 . . . . .	11
3.3	フォルダ構造 . . . . .	11
3.4	身体にデータを貼り付ける . . . . .	12
3.5	利用イメージ . . . . .	13
3.6	指先 . . . . .	15
3.7	プロトタイプ利用イメージ . . . . .	16
3.8	データの初期表示 . . . . .	17
3.9	指先近くのデータ表示 . . . . .	17
3.10	データを指差しての選択 . . . . .	18
3.11	データ移動前の選択 . . . . .	18
3.12	データ移動中 . . . . .	18
3.13	データ貼り付け中 . . . . .	19
3.14	データ貼り付け完了 . . . . .	19
3.15	データ削除 . . . . .	20
4.1	入場システム . . . . .	22
4.2	音楽ファイルと歌詞の管理 . . . . .	22
5.1	キャプチャ画像 . . . . .	24
5.2	ガウシアンフィルタ適用 . . . . .	24
5.3	肌色マスク . . . . .	25
5.4	肌色検出画像フィルタ適用 . . . . .	25
5.5	肌色検出画像フィルタ未適用 . . . . .	25

5.6	背景差分失敗	26
5.7	背景差分成功	26
5.8	ガウシアンフィルタ適用	27
5.9	手輪郭抽出	27
5.10	輪郭の重心と指先の検出	28
5.11	キャプチャ画像	28
5.12	背景と身体の分離	29
5.13	身体重心と外接円	29
5.14	入力画像	31
5.15	処理後画像	31
5.16	指先の滞留	33
5.17	データの選択状態	33
5.18	実験初期状態	35
5.19	カメラに近づく	36
5.20	カメラから遠ざかる	36

# 第1章 はじめに

## 1.1 背景

今日、ユビキタスコンピューティング [1] という概念が普及してきている。ユビキタスコンピューティングとは Mark Weiser が提唱したコンピュータの利用概念で、生活や社会にコンピュータが偏在し、いつでも何処でもコンピュータの支援を得られるような概念のことである。インターネットなど、通信技術の発展や携帯電話の普及で、我々はいつでもネットワークへ接続可能でコンピュータの支援を得られる環境がある。

ユビキタスコンピューティング環境においてはパソコンや携帯電話に限らず、車や冷蔵庫など生活に深く溶け込んでいるものもネットワークに接続される端末として利用される。ユビキタスコンピューティング技術を利用し、車がネットワークに接続している例として、VICS があげられる。VICS とは VICS センターが収集・処理・編集した道路交通情報を通信・放送メディアによって送信し、カーナビゲーションなどの車載装置に文字や図形として表示させるシステムである。<sup>1</sup>

また Mark Weiser は場所的な制約だけでなく、使いにくさの解消についての重要性も述べている。我々がパソコンに触れる時、従来ではマウスやキーボードなどといったユーザインタフェースが用いられる。しかし、そういったものになり近年では、手書き入力・音声認識・指差しといったような本来人の情報処理能力に近い技術が提案されている。

また、D.A.Norman は、コンピュータ利用者にその技術を意識させないことを「invisible」と述べ、その重要性を Invisible computing として提唱している [2]。Invisible Computing では、コンピュータ技術を道具へ埋め込みコンピュータ技術を意識させない道具を「情報アプライアンス」と呼び、Invisible Computing の究極の形であると述べている。情報アプライアンスは1つのタスクに対して特化されている。情報アプライアンスの利用者は、1つのコンピュータを利用するのではなく、複数のコンピュータを利用することになる。これは、Weiser の唱えるユビキタスコンピューティングに一致する。

ユビキタスコンピューティング環境において、RFID を用いての物品管理も主流になってきている。例えば工場で生産したものに RFID タグを埋め込み配送ルートを追跡を行ったり、本に RFID タグを埋め込み貸し出し・返却の自動化を行ったりしている。RFID に記録されている情報を専用のリーダーで読み取り、情報をコンピュータ上に表示している。しかし、コンピュータ上に表示されたデータを操作するためにはマウスやキーボードなどが用いられる。また、一般にコンピュータ上に記録されているデータはフォルダを用いて管理したり (図 1.1)、

---

<sup>1</sup> wikipedia:<http://ja.wikipedia.org/wiki/VICS>

コンピュータのデスクトップ上に並べておく(図 1.2) などで管理されている。つまり、コンピュータ上でデータの管理を行うためにはマウスやキーボードの操作・フォルダ構造の概念を理解する必要があり、利用者にコンピュータ技術を習得する事を強要している。

コンピュータに記録されているデータで我々がしばしば管理しているものは、画像データや音楽データや文書データなどが上げられる。これらはデジタルカメラで撮影された写真をダウンロードしたり CD からダウンロードしたものや WEB からダウンロードしたものである。これらのデータは個人個人が保有するものであり、個人コンテンツであると言える。

また、コンピュータの小型化に伴い、我々は様々なコンテンツをより手軽に楽しむ事が出来るようになった。例えば、携帯電話を用いて映画を鑑賞したり、デジタルカメラを用いて写真を撮影したり、MP 3 プレーヤーを用いて音楽鑑賞が出来る。手軽に楽しめるようになる反面、コンピュータに記録されているデータ量は膨大な量になっていると考えられる。目的となるコンテンツへのアクセスを容易にするためにも膨大な量の個人コンテンツを管理する必要がある。また、小型化されたコンピュータを用いる人は技術に詳しい人、そうでない人と様々であり、コンテンツ管理は誰もが行きやすくすべきである。

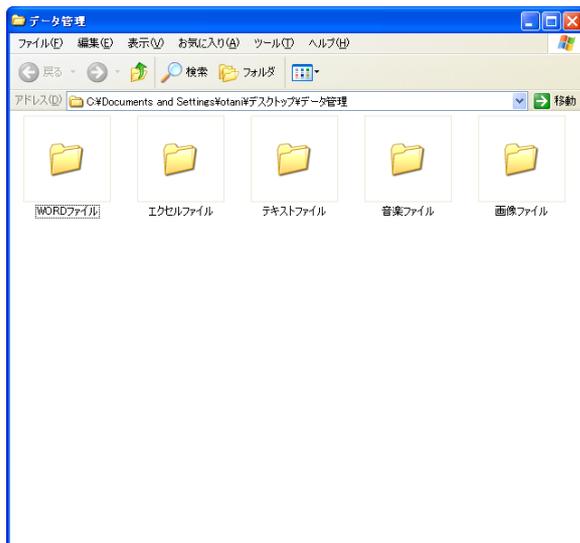


図 1.1: フォルダでの管理

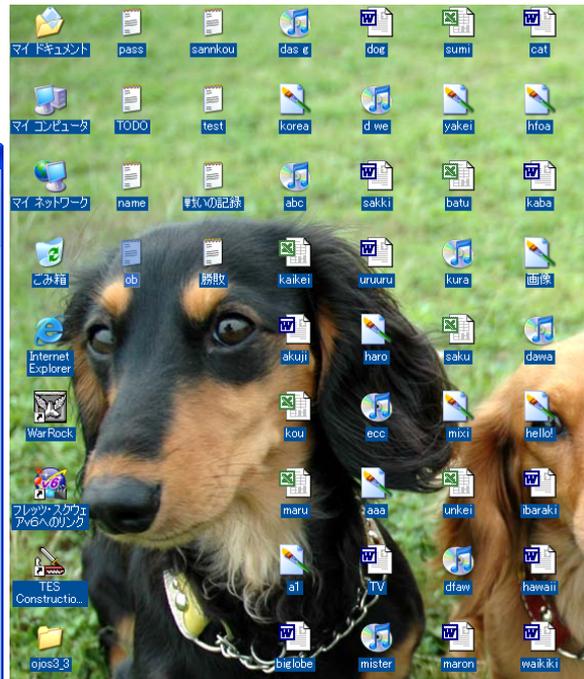


図 1.2: デスクトップでの管理

## 1.2 研究の目的と論文構成

本研究の目的はコンピュータ上にある個人コンテンツの管理をコンピュータ利用者が技術を意識せずに行えるようにすることである。技術を意識させないことで、利用者に操作の習得を強要しないため、コンピュータになじみのない人でも使えるようにする事が出来る。コンピュータの技術を意識させないために、マウスやキーボードを用いない自然なインタフェースを持つ管理手法を提案・実装した。

本研究ではコンピュータ上でコンテンツの管理を行うのではなく、利用者の身体を用いてコンテンツの管理を行う手法を採用した。インタフェースはディスプレイとカメラから構成され、ディスプレイの前に立つことでコンテンツの管理を行う事が出来る。コンテンツの管理はディスプレイに映し出された利用者の身体にデータを貼り付ける事で行われる。コンテンツの管理を行うためには、貼り付けたデータの移動や削除が必要がある。そこで、本研究では、手を用いて身体に貼りついたデータの操作を行えるようにした。

本論分の構成は2章で先行研究として、柔らかいストレージについて述べる。3章で本研究で提案するコンテンツ管理インタフェースを述べ、4章では、提案したインタフェースを用いた利用例を挙げ、5章ではインタフェースの実装と検証実験について述べる。最後に6章でまとめとする。

## 第2章 柔らかいストレージ

### 2.1 柔らかいストレージとは

ユビキタス環境でもデータを記憶するためのストレージは非常に重要である。ユビキタス環境において情報アプライアンスはネットワーク上で相互接続されるため、ネットワークストレージが広く利用されると思われる。しかし、ネットワークストレージでは、GUIの操作が必要であり、またIDやパスワードを入力するためキーボードが必要になってくる。これらの問題はユビキタスコンピューティングの概念に反するものである。そこで、岩淵らはNormanの提唱するInvisible Computingのようにネットワークストレージの技術を隠匿するため、自分の身体をストレージとして用いる、柔らかいストレージの提案を行った[3][4]。体が持つストレージを柔らかいストレージと呼んでおり、柔らかいストレージでは、情報アプライアンスの利用者の体そのものをネットワークストレージのキーとして利用し、さらにアクセスのためのインタフェースにも体を用いる手法を提案している。利用者は体内に専用のストレージを持ち、データを自分の体に記録しているという錯覚を持つ。

柔らかいストレージでは体をデジタルデータの保管場所として利用し、日常生活の中で情報アプライアンスと共に利用される。利用者は図2.1のように、データを携帯するイメージを持つ事が出来る。

情報アプライアンスの利用者は特別な知識を必要とせず柔らかいストレージを利用することが可能である。柔らかいストレージを利用することで、従来の物理的なメディアと比べ以下の利点がある。

- データの持ち運びの負担の解消
- 紛失の恐れ of 解消
- 物理的なデータ破損の恐れの解消

上記の事は小型のデバイスについても同様の事が言える。自分の身体以外のものを使う事を強制してしまうと負担が生じ、小型化しても紛失するリスクが増えてしまう。

### 2.2 柔らかいストレージの利用例

- カメラ型アプライアンス  
現在、デジタルカメラで撮影した写真を見るためにはデジタルカメラにメモリカード

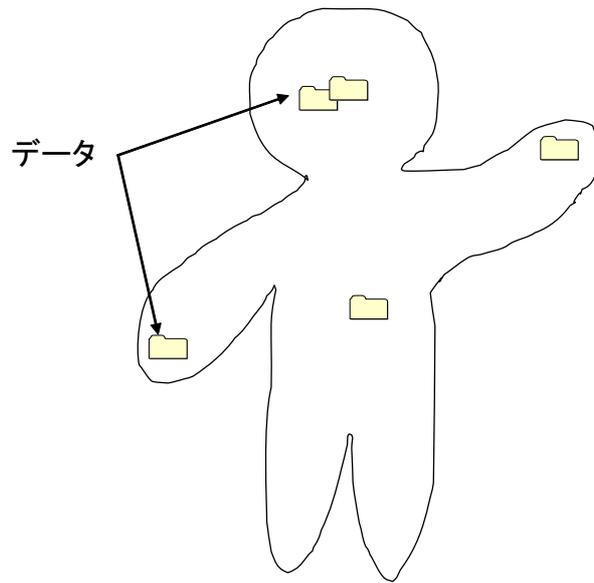


図 2.1: 持ち運びイメージ

を挿入し、撮影したデータを記憶した後、パソコンで再生するか、プリントアウトして見る。柔軟いストレージを利用すると、デジタルカメラで撮影された写真は体内へ格納され、プリンタなど出力に関するアプライアンスを利用する時は体内から直接データを出力する。(図 2.2)

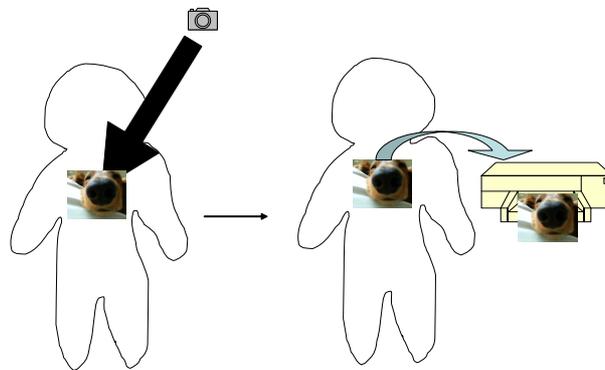


図 2.2: カメラ型入力アプライアンスとプリンタ型出力アプライアンス

- 商品販売端末

デジタルデータの販売に柔軟いストレージを利用する。最近では音楽や映画のコンテンツなどはダウンロード販売が広がっており、物理的なメディアを必要としない。現在では、ダウンロードされたデータはCDに書き込むなどして再生をする。柔軟いス

ストレージを利用することで、店先でコンテンツを買うと同時に体内へデータがダウンロードされ(図 2.3)、記録されたデータを体内から出力し、再生する。



図 2.3: ダウンロード

## 2.3 ストレージ技術の隠匿

### 2.3.1 ストレージと個人の対応付け

情報アライアンスは利用者の指紋などの生体情報から個人の ID を得る。得られた ID からその ID に対応づけられたネットワークストレージにアクセスする。個人 ID は身体そのものなので、利用者の身体はリムーバルストレージと同様の働きをし、どこでもネットワークストレージを利用する事が出来るといえる。

### 2.3.2 ストレージの存在の錯覚

利用者の身体の中にストレージがあると錯覚させるため、利用者と情報アライアンス間のインタラクションとして、アドレッシングとフィードバックを実装する。利用者が身体の中にデータを保存しているように見せかけるためには、アドレッシングによる保存先とフィードバックから想像される保存先を一致させる必要がある。

アドレッシングには指紋認識・顔認識などの生体認証を用いるが、どの認証を用いるかはアライアンスによる。たとえばデジタルカメラなど手に持つようなアライアンスであれば、指紋認証が効果的であると考えられる。

フィードバックについては、ストレージにアクセスしていることを身体と密接に関係して

いる表現手法を利用者に提示する必要がある。たとえばアクセスしている時に音や振動などを利用者にフィードバック出来れば、フィードバックから想像される保存先は確かなものになると考えられる。

## 2.4 NS ミラーの実装

柔らかいストレージを用いるアプライアンスとして、Natural Storage Mirror(以下 NS ミラー)と Natural Storage Camera(以下、NS カメラ)が実装されている。NS ミラーは利用者の体内に保存されている画像を閲覧するための鏡で、NS カメラは撮影した写真を体内へ保存するカメラである。NS カメラで写真を撮影した人が NS ミラーの前に立つと体内に記録されている写真画像が鏡に表示されるようになっている。

### 2.4.1 NS ミラー概観

NS ミラーは体内に保存した写真データを映し出す鏡である。自分の身体の周りに柔らかいストレージに保存されているデータが表示される。NS ミラーは図 2.4 のようなイメージで使われる。鏡に映った時に画像データは身体から飛び出すように表示され、その後身体の周りを回転しながら表示される。また、NS ミラー上で自分の手を用いてデータを他人に渡す事も可能である。

NS ミラーのプロトタイプはディスプレイとカメラから構成される。利用者は図 2.5 のような四角形のマーカーを取り付ける事で、鏡に映りこんだ人の位置と ID を検出出来る。マーカーを検出するために ARToolKIT[6] を用いて実装を行っている。

### 2.4.2 写真データ表示方法

NS ミラーの前に立ち閲覧できるデータは、身体の中央から飛び出してくるように見え、自分の身体の周囲をゆっくりと回転する。写真を大きくしたい場合は鏡に近づき、小さくしたい場合は鏡から遠ざかる。複数人が同時に映りこんだ場合、各々がマーカーを装着することで、それぞれの身体の周りに写真が表示される。

### 2.4.3 写真の交換

写真を交換するために、NS ミラーでは図 2.5 のようなマーカーを用いる。このときマーカーは個人の身体の位置と ID を検出するために用いたものと別のものを用いる。このマーカーを NS ミラーに映すと、マーカー上に白い四角形のカーソルが仮想的に表示される。このカーソルはカーソルの最も近くに表示されている写真を吸い付ける効果がある。

写真を渡すためにはまず NS ミラーにマーカーの認識をさせることから始める。マーカー



図 2.4: NS ミラー概観



図 2.5: マーカー

を手に装着し、マーカーの認識が終わったら、カーソルを表示されている写真データの近くに持っていく、写真をカーソルへ吸い付かせる。吸い付かせたまま、写真を渡したい人の身体の近くへ移動させ、マーカーを非表示にさせることで、データの譲渡が完了する。

## 第3章 身体に貼り付けるインタフェース

### 3.1 従来のコンテンツ管理手法の問題点

コンピュータのディスプレイは現実世界でコンピュータ利用者の机の上を表現している。コンピュータでファイルを管理する事は現実世界での机にある物を整理する事と同義であると言える。我々は机の物を机の引き出しやフォルダなどに格納する事によって整理している。コンピュータにもフォルダメタファと呼ばれるものが存在し、それを用いて我々はコンピュータ上のファイルを管理する。コンピュータのフォルダはコンピュータの記憶メディアに保存されているデータを整理・管理するためのツリー構造(図 3.1)を持つ。

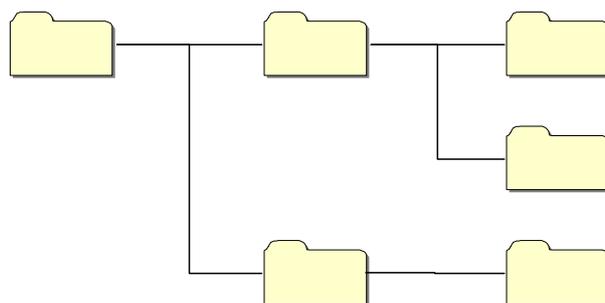


図 3.1: ツリー構造

次に、ファイル管理の手法について具体的に述べる。

机の物をフォルダを用いて整理した場合、フォルダがどんなものをまとめたものであるかが分かるよう、背中に名前を振っておく。また、それらのものを本棚などに整理しておく事により机にある物の管理を行っている。コンピュータでも同様の操作が可能であり、コンピュータに記憶されているファイルを用意されたフォルダへと格納する。このときにどんなファイルを格納したかが分かるようにフォルダに名前を付けておく。次に現実における本棚に相当するフォルダを用意し、そのフォルダへファイルを格納したフォルダを移動させる事でコンピュータ上に記憶されているファイルを管理する事が出来る。この構造は、ツリー構造で図 3.2 のように示され、コンピュータ上では実際に図 3.3 のように表示される。

しかし、利用者がコンピュータに関する知識がなかった場合、フォルダとファイルを混同してしまう事がある。現実のフォルダはファイルと呼ばれることもあり、コンピュータ上のファイルを管理するために存在するフォルダメタファの使い方にギャップがあるからだと考えられる。

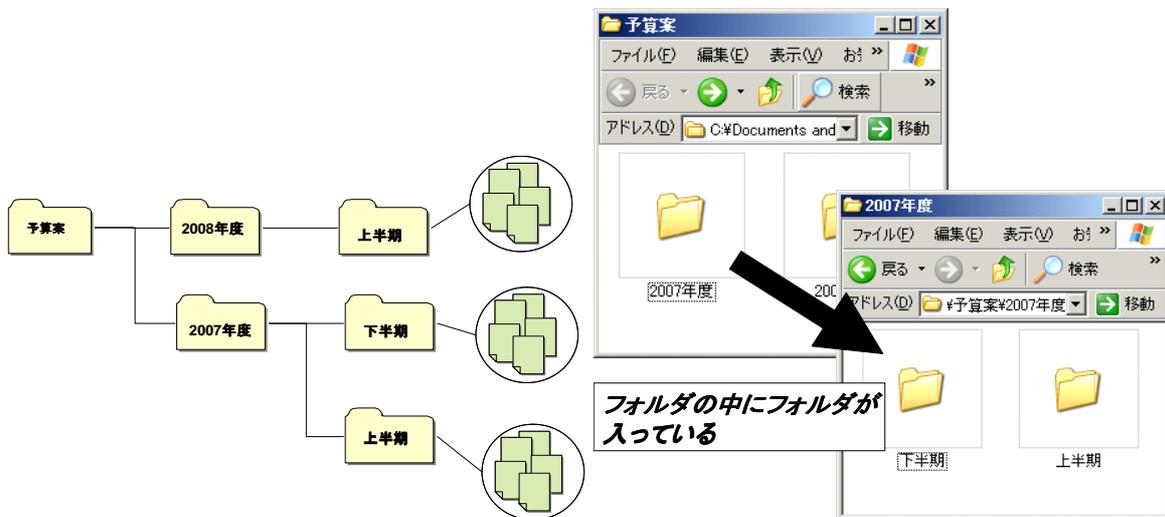


図 3.2: ツリーでの構造表示

図 3.3: フォルダ構造

また、ファイル名やフォルダ名をつけるためにはキーボードを用いる。ファイルやフォルダを移動するためにはマウスを用いる。これらの操作は慣れていない人にとっては大きな負担になってしまう。

つまり、コンピュータにあまり触れる事のない人はコンピュータ上でのファイル管理を効率的に行う事が出来ない。

### 3.2 コンテンツ管理の関連研究

#### Data mountain

コンピュータ上のデータを管理する研究として Data mountain と呼ばれる研究がある [5]。Data mountain は 3D のインタフェースを持つ。Data mountain のインタフェースでは 3D で山の斜面のようなグラフィックが描かれており、そこにデータを貼り付ける事でデータを効率的に管理出来るシステムである。プロトタイプではブックマークされた WEB ページを用いてデータ管理を行っている。システムを起動するとブックマークされた WEB ページが山の斜面にレイアウトされた状態で表示される。この時 WEB ページはサムネイルで表示されている。

サムネイルをクリックする事でサムネイルを掴むことが可能で、その状態でドラッグすると貼りつける位置を任意に変更する事が出来る。サムネイルをダブルクリックする事でサムネイルが拡大表示され、概観が見やすくなる。また、サムネイル上でマウスカーソルを滞留させておくと、WEB ページのタイトルがポップアップ表示される。

実験は IE4 と Data mountain を用いて行われている。結果は IE4 によるブックマーク管理手法より Data mountain によるブックマーク管理手法のほうが使いやすいという結

果が出ている。しかし、実験の参加者はウェブブラウザとインターネットについての選択問題に完璧に答えられた人であり、コンピュータの知識は十分に持っている。Data mountain ではマウスを用いる必要があるため、コンピュータの知識のない人にはWEBブラウザによる管理とData mountain による管理のしやすさにあまり変わりがないと考えられる。

### 3.3 インタフェースの提案

本研究では、利用者がコンピュータ技術を意識しない個人コンテンツ管理手法を実現するために、システム利用者自身の身体を用いてコンテンツを管理する手法を提案する。利用者は自身の身体へとデータを貼り付ける事でコンピュータ上のデータが管理できるようにする。イメージ図を図 3.4 へと示す。

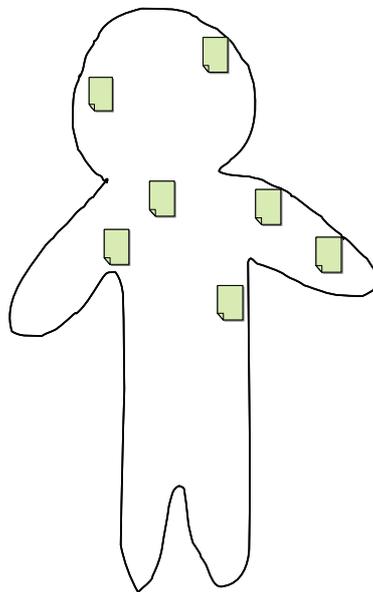


図 3.4: 身体にデータを貼り付ける

身体へ貼り付けたコンテンツは閲覧する事が可能で、利用者は直感的に必要なコンテンツへとアクセスが可能になる。身体へ貼り付けてコンテンツを管理する事で、フォルダ構造を撤廃し、コンピュータに触れる機会のない利用者でもデータ管理が行えると考えられる。データを身体へ貼り付けるためには、何処からかデータを取得する必要がある。本研究では情報アプライアンスを用いる事でデータの取得する手法を想定する。利用者が例えばMP3プレーヤーを手に持ち、カメラが設置されているディスプレイの前に立つことで、MP3プレーヤーの中に保存されている音楽データがディスプレイに表示される。利用者はディスプレイに表示された音楽データを自分の身体に貼り付ける事で自分のコンテンツを管理する事が出来る。(図 3.5)

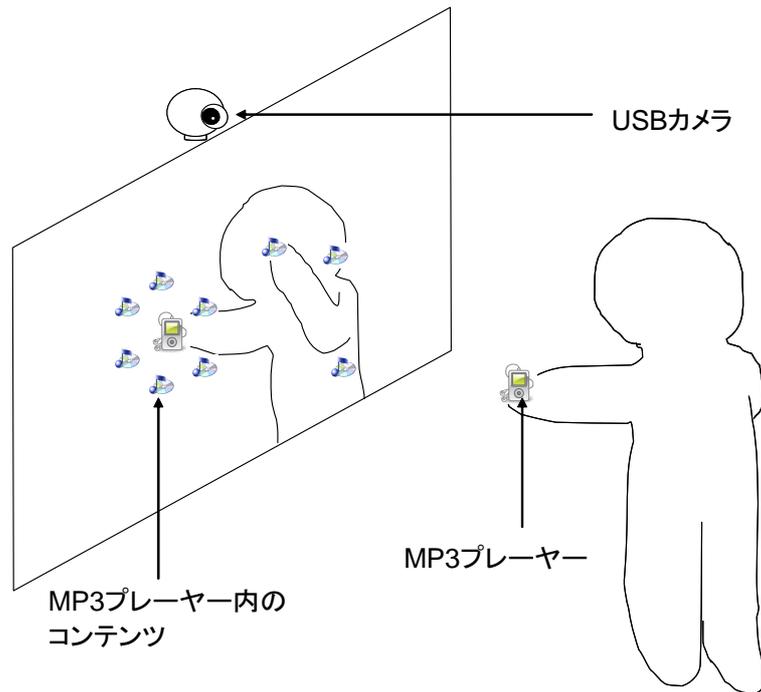


図 3.5: 利用イメージ

### 3.3.1 柔らかいストレージとの相違点

柔らかいストレージ(以下、NS)のコンセプトはコンピュータの技術を意識させない事である。このことは、利用者がネットワークストレージの技術を知らなくてもネットワークストレージを利用出来るということで、つまりは、普段コンピュータになじみのない人でも簡単にネットワークストレージを利用出来るという利点がある。NSではネットワークストレージのキーとして利用者の身体を利用している。身体を利用することで、ストレージにアクセスするために機械を使わずにすみ、利用者は自然にネットワークストレージを利用する事が出来る。

本研究ではこの点に着目し、利用者がコンピュータになじみのない人でも利用出来るインタフェースを開発するため身体を利用する。NSはネットワークストレージに記録されているデータを身体に記録されているように見せかけているが、データを身体に置くという事しか考えられていない。NSでは記録されているデータの操作や身体の何処に保存されているかが未検討である。従って、本研究とNSとの相違点は、コンピュータに記録されているデータを身体の任意の部位に保存しているように見せかけ、さらにデータの操作を行えるようにするインタフェースを開発するところである。

### 3.3.2 実装方針

ただ貼り付けるだけでは、身体を現実の机の上として見たてると、机の上に物が乱雑に置いてあるのと同様で、管理しているとは言い難い状態である。そのため、利用者が管理しやすく、且つコンテンツへのアクセスを容易に出来るようなインタフェースでなくてはならない。提案するインタフェースは以下の条件に基づいて実装を行う。

1. 身体の任意の位置にコンテンツを貼り付けられる
2. 指先を用いてデータの操作を行う
3. 画像処理による身体認識

#### 任意の位置にコンテンツを貼り付ける

利用者が好きな位置にコンテンツを貼り付ける事が出来るようにする事でコンテンツの管理がしやすくなると考えられる。利用者が身体に貼るデータの種類と、身体の位置に何かしらの法則を持たせる事が出来るためである。目の付近に画像データや動画データを貼り付ける。画像や動画は見るものなので目の付近に画像データや動画データは存在すると利用者は理解出来る。また、耳の付近に音楽データを貼り付ける。音楽は聴くものなので耳の付近に音楽データが存在すると利用者は理解出来る。これは、現実世界の机に例えれば、一番上の引き出しには月曜日に使う書類を、次の引き出しには火曜日に使う書類を置いておく…といった事と同義である。管理法則は利用者の感覚で決める事が可能であり、上記の事は単なる一例で、音楽データを目に貼り付けたり、口に貼り付けたりする利用者もいると思われる。つまり、利用者はデータを貼り付ける時に何かしらの法則を定め、その法則を覚えている事で、データの管理・データへのアクセスを容易にする事が出来る。

#### 指先を用いてのデータ操作

従来のコンテンツ管理方法では、入力インタフェースとしてマウスやペンタブレットを用いて行っている。コンピュータの操作に慣れていない人にとって、コンテンツを管理する際にマウスやペンタブレットを用いるのは非常に負担になってしまう。従って、これらのデバイスに代わる入力インタフェースが必要となる。マウスやペンタブレットに変わる入力インタフェースとしてレーザーポインタや手が考えられる [7][8]。コンピュータを意識しないインタフェースを目指すため、レーザーポインタのような特別なデバイスを用いるのは意に反する。また、現実世界でも物の移動などは手を用いて行っている。よって本研究では、手を用いて指先を認識することで、データの操作を行う。ここで言う指先とは図3.6のように、指を1本だけ突き出した状態の事を指す。



図 3.6: 指先

### 画像処理による身体認識

NS ミラーでは人の位置の認識と、データ操作をするための手の認識をマーカーを用いて行っていた。マーカーを用いる事は利用者に特別なデバイスを装着する事を強要してしまい、研究の意に反してしまう。また、NS ミラーを利用するために逐一マーカーを付けているのは利用者にとっても負担になってしまう。従って、マーカーを利用せずに身体の認識を行う事が望ましい。本研究ではカメラで撮影された画像を処理し、人の位置と手の位置を認識出来るようにする。また、そうすることで、NS ミラーではマーカー周辺に表示されていたデータを身体の任意の位置へ表示させる事が可能になる。

## 3.4 プロトタイプ利用イメージ

本インタフェースの利用イメージを図 3.7 に示す。図では利用者の身体にデータが貼りついており、利用者がそのデータを操作している。

本インタフェースは USB カメラとディスプレイから構成される。ディスプレイの前に立つと利用者に貼りついたデータが表示される。USB カメラで人の位置を検出し、利用者にデータが貼りついているように見せている。また、データの管理や操作も、ディスプレイの前に立つ事で可能になる。

USB カメラで指先の位置を検出し、身体に貼りついているデータへ触れる事が出来る。今回実装したプロトタイプでは画像データ(以下、データ)を扱う事にした。

## 3.5 データの操作

本インタフェースでは「閲覧」「選択」「移動」「貼り付け」「削除」のデータ操作を行う事が可能である。これらの操作は実世界で物を管理する際の動作に対応付けられる。(表 3.1) それぞれについては次節で詳述する。



図 3.7: プロトタイプ利用イメージ

インタフェースでのデータ管理	現実世界での物管理
データの閲覧	物の閲覧
データの選択	物を指差す
データの移動	物を移動
データの貼り付け	物の格納
データの削除	物の廃棄

表 3.1: 動作の対応関係

### 3.5.1 データの閲覧

貼り付けたデータを管理するためにはデータが閲覧出来る必要である。USB カメラを用いてデータの閲覧が出来るようにした。USB カメラで利用者の身体を撮影する事でディスプレイにはデータが貼りついた利用者の身体が表示される。

貼り付けたデータが多くなった時、データは混雑した表示になってしまうと考えられる。それを避けるために、何もしていない状態でディスプレイの前に立つと、表示されるデータは、存在が分かる程度の大きさに縮小して表示されるようにする(図 3.8)。また、何もしていない状態の時にデータを表示させない方法も考えられる。しかし、利用者は身体しかディスプレイに映らないため、「貼りついている」というフィードバックを得る事が出来ない。そのため、利用者がデータを貼り付けた位置を覚えていたとしても、データが何処に貼りついているかなどを直感的に理解する事が難しいと考えられる。

また、本インタフェースでは指先を身体上で指し示す事で、データを閲覧できるようにする。身体上で指先を這わせると、指先付近の画像が拡大され、何が貼り付けられているかが見やすく表示される(図 3.9)。逆に指先が離れると画像は図 3.8 のような状態に戻る。上記の閲覧方法を取る事で、利用者は必要となるデータの選択を容易に行う事が出来る。



図 3.8: データの初期表示



図 3.9: 指先近くのデータ表示

### 3.5.2 データの選択

身体に張り付いているデータから目的となるデータを見つけられたとしても、データを使う事が出来なければ意味がない。利用者が使いたいデータに対し、様々な操作が出来るようにデータの選択が出来るようにする必要がある。

従来のデータ管理手法では、マウスクリックやタップなどにより利用したいファイルを選択している。今回は、マウスやペンタブレットを指先で代替するため、クリックやタップをする操作が困難である。マウスやペンタブレットに代わるポインティングデバイスでのポイン

ティングは囲う動作 [9] などで行う事も可能である。しかし、我々が現実世界で物を選択する時は指差しを行う。利用者にとっては指差しする行為が物を選択する自然な動作であると考えられる。そこで、本インタフェースではクリックやタップの代わりに、データ上を指差し、指先を滞留させる事によりデータを選択する手法をとる(図 3.10)。データを選択する事により、後述するデータの移動、削除、貼り付けなどの操作を行う事が出来る。上記の選択方法により、利用者は表示されているデータを普段行っている自然な動作で選択する事が出来る。また、データの選択中、指先を用いてのデータ閲覧はする事が出来ないようにしている。



図 3.10: データを指差しての選択



図 3.11: データ移動前の選択



図 3.12: データ移動中

### 3.5.3 データの移動

我々は物を管理する時、以前とは違った場所に物を置いて管理する事も少なくない。これは気分的な理由で移動させる事もあるし、また、置いておいた場所がわかりにくいという理由で移動させる事もある。そのため、本インタフェースにもデータを貼り付ける位置を変更出来るような機能が備わっているべきである。

本インタフェースではデータ選択状態(図 3.11)で指先を動かすと選択されているデータを移動させる事が出来る(図 3.12)。移動させる手順について述べる。まずはじめに、移動させたいデータを指先で選択する。選択されたデータは指先を動かしても追従する。目的となる場所に指先を移動させ、データを貼り付ける操作を実行する事でデータの移動が完了する。移動させたデータがあった位置に指先を近づけても移動させたデータは表示されない。移動させた身体的位置へ指先を近づける事によって移動させたデータが閲覧出来る。



図 3.13: データ貼り付け中



図 3.14: データ貼り付け完了

### 3.5.4 データの貼り付け

データの貼り付けは利用者の身体上へと行われる。また、利用者が指定した位置に貼り付けられる事が必須である。本インタフェースでは、貼り付けたいデータを選択状態にし、身体上で指先を滞留させる事によって(図 3.13)、データの貼り付けが完了する(図 3.14)。データ選択中は身体がデータに覆われる事がないので、利用者が指定する位置に貼り付ける事が出来る。また、データ選択中は身体の輪郭が見えるため、スムーズに身体の範囲内へ貼り付ける操作が可能になると考えられる。データを貼り付けた後、指先を用いてのデータの閲覧や、別のデータを選択出来る状態になる。貼り付けたデータは他のデータを選択していなければ、指先を貼り付けた場所へ近づければ拡大表示される。

### 3.5.5 データの削除

我々は物を管理する時、廃棄をする事で管理を行う事が多々ある。管理するものを増やし続けるだけでは、いずれ管理能力は追いつかなくなってしまう。そのため、本インタフェースにも貼りついているデータを削除出来る機能が備わっているべきである。

本インタフェースでは、貼り付ける操作とは逆に、身体の外側の領域を用いる事にした。つまり、削除したいデータを選択し、身体の外側で指先を滞留させる事で、そのデータを削除する事が出来る。現実世界で物を廃棄するという事は、物を管理する領域から移動させる事と同義であると考えられる。例えば、冷蔵庫を廃棄するという事は、「家(部屋)」という、物を管理すべき領域から冷蔵庫を移動させる事である。従って、身体というデータを管理する領域から身体の外側へデータを移動させる事は、利用者にとって、データを削除するための自然な動作ではないかと考えられる。



図 3.15: データ削除

## 第4章 システム利用例

### 4.1 写真管理システム

普段余りコンピュータを使わないAさんが、友達との旅行中デジタルカメラで撮影した写真をコンピュータ上で管理しようと試みた。デジタルカメラからコンピュータ上へ写真の転送は出来たが、マウスやキーボードの操作に不慣れなAさんにはデータの移動や削除などが難しく、写真を管理するには難しい状況である。そこで、本システムをAさんが利用し、普段物の管理を行っているようにデータの操作を行えば、簡単に写真の管理をする事が出来る。また、友達と一緒に写真を閲覧したい時も、Aさんはディスプレイの前に立って手で操作するだけで閲覧する事が出来る。

### 4.2 会場入場システム

本システムを柔らかいストレージと組み合わせる事を考える。システムのイメージを図4.1に示す。利用者Aさんはインターネット販売で、あるアーティストのライブチケットをダウンロード購入し、柔らかいストレージへと記録する。記録されたデータを本システムを用いて手へと移動させる。当日ライブ会場へ訪れたAさんは入場ゲートにある読み取り機に手をかざす。読み取り機は生体認証を行い、Aさんを認識する。Aさんを認識したら対応するネットワークストレージ中のデータで手に保存されているデータを探す。読み取り機がチケットを認識したらチケットデータベースの個人情報と、チケットの個人情報とを照会し、それが一致したらAさんの入場が完了する。

### 4.3 音楽ファイルと歌詞管理

利用イメージを図4.2に示す。Aさんは気に入ったCDがあるとレンタルして家族にコンピュータへDLしてもらおう。また、CDはいつか返さなくてはいけないため、歌詞を参照出来る期間が限られている。WEBページで歌詞を参照出来るサイトはあるが、コンピュータ操作の苦手なAさんにとって音楽を聴く時に毎回サイトにアクセスするのは非常に辛い作業であるため、テキストファイルで歌詞を管理する方法を取った。しかし、コンピュータ上での管理はAさんには難しい。そこで、Aさんは自分の身体を利用してDLした音楽ファイルと歌詞を管理する。Aさんは好きな音楽を頭方向にあまり聴かない音楽を足方向に貼り付ける事で管理する事が出来る。対応した歌詞を音楽ファイルのすぐ横に貼り付けておく事で、簡



図 4.1: 入場システム

単に参照する事も出来た。本インターフェースを使う事で、コンピュータの苦手な A さんでも好きな時に音楽を聴くことが出来、また歌詞も参照することが出来た。



図 4.2: 音楽ファイルと歌詞の管理

## 第5章 実装

### 5.1 環境・言語

開発言語はC++、開発環境としてVisualStudio.NET 2003を使用。OSはWindowsXP、Pentium4・2.80GHzである。画像処理にはOpenCV[10]を用いている。OpenCVとはIntelが無償で公開しているライブラリである。USBカメラにはLogicoolのQcamPro4000を利用した。

### 5.2 トラッキング

#### 5.2.1 指先のトラッキング

指先のトラッキングの精度を上げるためにいくつかの前処理を行っている。指先トラッキングまでのプロセスは以下の通りである。

肌色検出 手領域検出 手輪郭検出 指先検出

- 肌色検出

指先を抽出するためにはまず、背景から肌色領域を分離する必要がある。本研究ではキャプチャされたRGB空間画像をYUV空間へ変換して肌色を検出する。YUV空間とはY:輝度、U:輝度信号と赤色成分の差、V:輝度信号と青色成分の差である。RGB空間からYUV空間への変換は次の式で行われる[11]。

$$Y = (299 * R + 587 * G + 114 * B) / 1000$$

$$U = (-147 * R - 289 * G + 436 * B) / 1000$$

$$V = (615 * R - 515 * G - 100 * B) / 1000$$

検出実験は室内(蛍光灯有り)で行う。肌色検出技術としてしばしば用いられる[12]RGB空間[13]・HSV空間[14]・YUV空間[15]を用いて行った。それぞれの色相空間に対し肌色らしい閾値を与え検出を行い、一番精度のよかったYUV空間による肌色検出を本研究では採用した。

キャプチャされた画像(図5.1)から肌色領域を取りやすくするために、キャプチャ画像にガウシアンフィルタを複数回かけたものを入力画像(図5.2)とした。キャプチャ画像をそのまま用いると肌色に影がかかってしまっている場合もあり、肌色の検出制度が

悪くなってしまう。ガウシアンフィルタとはあるピクセルに注目した時、そのピクセルの周囲に存在するピクセルの画素の平均値が注目されているピクセルに入力される。つまり、画像が平滑化され、影がかかっている肌色部分も肌色と認識されやすくなる。

肌色の検出は入力画像を1ピクセルごとにその画素を調べる。入力画像はRGB空間の3チャンネルで構成されており、ピクセルごとにRGB値を上記の式で、YUV値へ変換する。変換されたYUV値に対し、肌色らしいかそうでないかを調べていく。肌色らしい色の範囲内であればそのピクセルには白色を入力し、逆に肌色らしい範囲内になければ、そのピクセルには黒色を入力する。この処理を各ピクセルごとに行う事で、肌色だけを抽出した白色と黒色で構成されたマスク画像を生成する事が出来る(図5.3)。

生成された肌色マスクと図5.1とのANDを取得することで、肌色だけが検出された画像を得る事が出来る(図5.4)。図5.5はガウシアンフィルタ未適用の場合の肌色検出である。図5.4と比べると明らかに取得している肌色領域の精度が違うのが理解出来る。



図 5.1: キャプチャ画像



図 5.2: ガウシアンフィルタ適用

- 手領域検出

抽出された肌色領域を入力画像とし、手の領域を取得する。手画像から指先の検出を行っている研究は多数あるが[17][18][19]、キャプチャされている画像の中の肌色らしい領域は、手のみである。しかし本研究では、手に加え顔が肌色らしい領域としてキャプチャ画像内に存在し、顔と指先が重なってしまう状況もあるため、違ったアイデアが必要になる。そこで手をポインティングデバイスの様に使う事に着目した。身体に貼りついているデータを操作するためには手を動かす必要がある。よって、動いている肌色領域が手であるという考えの下、背景差分法による実装を行った。

背景差分法とは、背景となるフレームと別のフレームを比べ、背景となるフレームと異なる画素があった場合、その部分を抽出する技法である。本研究の背景差分法では、あるフレームとその1フレーム前の画像を比べて画素がある程度違っている部分があっ



図 5.3: 肌色マスク



図 5.4: 肌色検出画像フィルタ適用



図 5.5: 肌色検出画像フィルタ未適用

たら、そこは動いた部分であると認識される。ここで気をつける事として、単に画素が違う部分をとってくるのでは、図 5.6 のようになってしまう。移動前に肌色だったところが黒色になってしまうため、画素が違くと判断され、手の領域であるように誤認識されてしまう。誤認識された画像と現在入力されている画像とを見比べ、肌色と認識されている領域で、現在の入力画像で同じ領域が黒色であれば、そこは手の領域ではないと判断すれば、正しい手の領域を抽出する事が出来ると考えられる。

肌色検出と同様に 1 ピクセルずつ画素を調べていき、画素が違う場所があれば白色を、同じであったら黒色をそのピクセルに入力していく。また、誤認識されたピクセルにも黒色を入力していく。

複雑背景下においてこの条件を与えるのは難しいため、手領域検出用の画像を図 5.4 のような背景が黒色のものを使用している。以上の処理により、動いている部分のみを抽出した白色と黒色で構成されたマスクが作成される。このマスクと入力画像の AND を取る事で手領域が検出された画像を取得する事が出来る (図 5.7)。



図 5.6: 背景差分失敗



図 5.7: 背景差分成功

- 手輪郭検出

手を検出しただけでは、「そこに肌色の領域がある」という事しか認識しておらず、手を構成する要素の座標などの情報を得る事が出来ない。手の輪郭の抽出を行い、座標などの指先検出に必要な情報を取得する。図 5.7 にガウシアンフィルタを適用した画像を入力画像 (図 5.8) とし、手の輪郭を取得する。輪郭をとるにあたり、図 5.8 を 2 値化する。2 値化を行う事で、輪郭となるエッジの検出の精度を上げることが可能になる。OpenCV では輪郭を取得する関数があるのでそれを用いて輪郭情報を得る事が出来る。得られた輪郭情報から輪郭を構成する領域の面積を調べ、ある程度の大きさ (本研究では 500 ピクセル以上) があつた場合それを手の輪郭であると判断する。入力画像は動いている肌色領域を抽出しており、顔が動いてしまったらその領域も手の領域であると認識されてしまうため、領域の面積を元にノイズの除去を行っている。ノイズの除去を行

う事で、手領域のみの輪郭を抽出する事が出来る (図 5.9)。



図 5.8: ガウシアンフィルタ適用

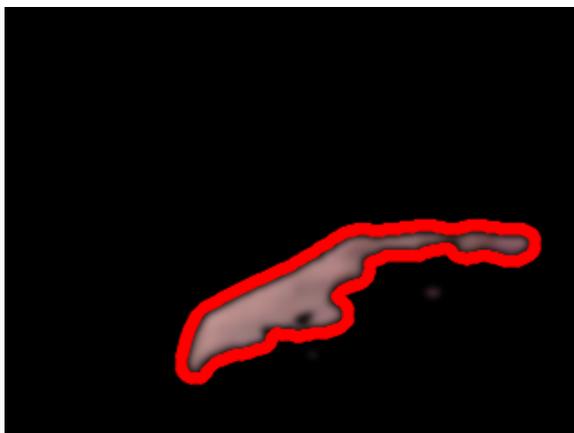


図 5.9: 手輪郭抽出

- 指先検出

図 5.9 のように、抽出された輪郭画像を入力画像とし、指先の検出を行う。手の輪郭から指先の曲率を用いて指先をトラッキングする手法 [18][19][20] がある。しかし、抽出された輪郭は図 5.9 のように純粋な手の形をしておらず、指先と同程度の曲率を持った部分が多数見られるため、曲率を用いた指先のトラッキングを利用する事は出来ない。そこで本研究では、手の輪郭情報から、輪郭の重心を計算し、重心を元に指先の検出を試みた。

手の領域を抽出した場合、その重心は手首側のほうに偏るため、指先は重心から最も遠い場所にあると言える。実際に指先を取得した画像を図 5.10 に示す。また、図 5.11 はその時にキャプチャされた画像である。図 5.10 のうち白く型取られているのが手の輪郭である。輪郭内部にある円印は輪郭の重心で、輪郭上にある円印は指先を検出している事を示している。

図 5.11 と図 5.10 を見比べると、顔と手両方の肌色領域が重なっていても、指先を検出している事がわかる。

## 5.2.2 身体のトラッキング

身体のトラッキングは背景の初期化から行う。背景の初期化はディスプレイの前に誰も立たない状態で 100 フレーム分画像をキャプチャする。この時カメラを動かしてしまうと初期化に失敗してしまうため、しっかりと固定しておく。キャプチャが終わったら 100 フレーム分の輝度の平均を求める。光の加減で、背景の輝度が変わってしまうこともあるので、100 フ

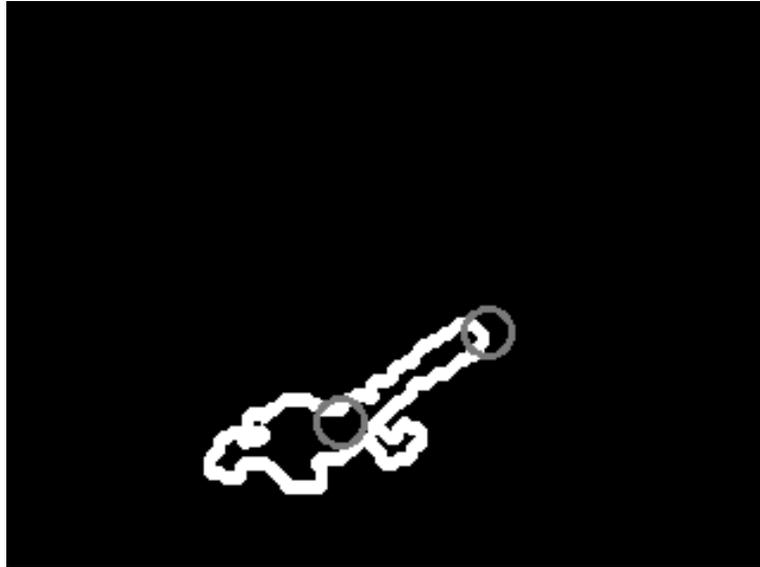


図 5.10: 輪郭の重心と指先の検出



図 5.11: キャプチャ画像

フレーム分の輝度振幅もこの時計算しておく。

初期化が終わりディスプレイの前に立つと、人の身体は最初に求めた輝度と輝度振幅の範囲内にないと認識されるため、背景と人の身体を分離する事が出来る(図 5.12)。身体と背景の分離が終了したら、その画像を入力画像とし、身体の輪郭を抽出する。背景の中でも時折輝度振幅外の領域が検出されてしまうため、手の輪郭を抽出する時と同様に、ある程度の大きさ以上のものを身体の輪郭として抽出する(図 5.13)。また、身体の輪郭情報から身体の重心と輪郭の外接円を計算し、その半径と中心座標を求める。



図 5.12: 背景と身体の分離



図 5.13: 身体の重心と外接円

## 5.3 身体の位置の決定

身体の任意の位置にデータを貼り付けているように見せかけるには、コンピュータ上のデータが記憶されている場所と、身体の何処にデータを貼り付けたかの対応が必要になる。そのため、キャプチャされた画像から身体範囲内の任意の位置座標を求める必要がある。本研究では、頭の頂点を中心とした座標系を用いて身体範囲内の位置座標を取得する。以後、頭の頂点の座標を頭頂座標と記述する。

### 5.3.1 制約

本研究では、頭頂座標からの身体的位置座標取得簡略化のため以下の制約を定めた。

- 頭より上に手を上げない
- 頭を傾けない
- カメラに近づきすぎたり遠ざかりすぎたりしない

### 5.3.2 頭頂座標の検出

利用者は頭頂座標初期化のため、最初は直立不動の状態ディスプレイの前に立つようにする。頭頂座標を取得するための入力画像は図 5.14 のような背景と身体が分離されている画像で、背景は黒色で身体は輪郭内部を塗りつぶしている。しかし、図 5.13 のように輪郭と認識されなくても背景にノイズが混ざってしまう可能性もあり得るため、このノイズを除去する必要がある。

キャプチャされた画像内には三原色、つまり RGB 値が (255, 0, 0)、(0, 255, 0)、(0, 0, 255) の色は存在しないという考えの下、輪郭内部は (255, 0, 0)、(0, 255, 0)、(0, 0, 255) のどれかの色で塗りつぶす。1 ピクセルごとに画像を調べ、輪郭内部を塗りつぶした色が入力されていないピクセルには黒色を入力、塗りつぶした色が入力されているピクセルには白色を入力する (図 5.15)。そうすることで背景にノイズの混じらない、純粋な身体の輪郭と背景を分離した画像が取得出来る。

頭頂座標は身体の正中線で最も高い位置にある。正中線とは生物体の前面・背面の中央を頭頂から縦にまっすぐ通る線の事である。図 5.15 のような画像の中で、正中線の X 座標上で一番上の Y 座標が頭頂座標であると言える。本研究では正中線の X 座標の認識に輪郭の外接円の中心座標を用いた。利用者は直立不動でディスプレイの前に立つため、外接円の中心座標の X 座標は正中線の X 座標と一致する。本研究では検出された外接円の中心座標の X 座標上で一番最初に白色が検出されたピクセルが頭頂座標として検出される。

しかし、重心の座標は腕を上げるなど直立状態でない場合は体の中心からずれてしまうため、重心の X 座標を用いて頭頂座標を検出することは出来ない。また、身体も動いてしま

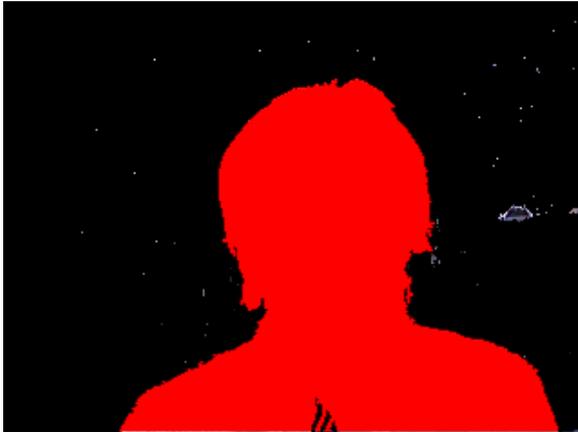


図 5.14: 入力画像



図 5.15: 処理後画像

うため、それに合わせて頭頂座標を修正していかなければならない。そこで、利用者のとっている動作を認識し、頭頂座標の修正を行っていく。姿勢の認識は初期状態からの重心の移動と、外接円の半径長の増減から判断する。

- 重心が上(下)に移動した場合  
この場合、カメラに近づく(遠ざかる)か、両腕を初期重心より上げる(下げる)かしている状態である。左右に重心は移動していないので、検出された重心の X 座標を用いて頭頂座標を検出する。
- 重心が左(右)に移動した場合  
この場合、左(右)に平行移動したか、左(右)腕を初期重心と同じ高さで伸ばしている状態である。平行移動した場合は外接円の半径長に変化はなく、腕を伸ばしている場合は半径長は長くなっているため、このことから動作を判断する。平行移動した場合は検出された重心の X 座標を用いて頭頂座標を検出する。腕を伸ばしている場合は、初期重心の X 座標を用いて頭頂座標を検出する。
- 重心が左(右)上に移動した場合  
この場合、左(右)に平行移動した後、カメラに近づいているか、左(右)腕を初期重心より高く上げている状態である。平行移動しカメラに近づいた場合は、半径長に変化はなく腕を上げている場合は半径長は長くなっているため、このことから動作を判断する。平行移動しカメラに近づいた場合は、検出された重心の X 座標を用いて頭頂座標を検出する。腕を上げている場合は初期重心の X 座標を用いて頭頂座標を検出する。
- 重心が左(右)下に移動した場合  
この場合、左(右)に平行移動した後、カメラから遠ざかるか、左(右)腕を初期重心

より低く下げている状態である。平行移動しカメラから遠ざかる場合は、半径長に変化はなく腕を下げている場合は半径長は長くなっているため、このことから動作を判断する。平行移動しカメラから遠ざかる場合は、検出された重心の X 座標を用いて頭頂点座標を検出する。腕を下げている場合は初期重心の X 座標を用いて頭頂座標を検出する。

以上のように、利用者の動作を認識し、それに対応した頭頂座標が検出される。しかし、このままでは複数を組み合わせた動作、たとえば「平行移動して腕を上げる」といったような場合に正しい頭頂座標を検出することが出来ない。本研究では、一つの動作が終了したら初期重心を更新していく手法を取ることで、複数の動作に対応した頭頂座標も検出している。

頭頂座標の検出が完了することで、頭頂座標を中心としたピクセル単位での座標系が構成される。肩部分から頭頂部にかけて身体領域範囲外の部分が存在するが、この部分は輪郭範囲外とみなすことで座標系から除外する。以上より、カメラによりキャプチャされた画像内における利用者の身体範囲内における任意の座標を取得する事が出来た。

## 5.4 データと身体座標の対応付け

本研究では、身体の座標とコンピュータに記録されている位置を一致させるために Dictionary 型を用いている。頭頂座標を中心とした座標系においてのデータ座標位置 (以下、データ座標位置) をキーとして、コンピュータに実際に記憶されているデータにアクセスしている。この時のデータ座標位置は、画像の中心座標が格納される。尚、キャプチャされた画像の座標系 (画像左上を原点とする) を用いた場合、身体が動いてしまうため、貼り付けた位置と表示される位置にずれが生じてしまう。イメージを表 5.1 に示す。

座標	パス
(35,100)	C:/Program Files/106ab0e9.jpg
(56,12)	C:/Documents and Settings/otani/Snow - GalleryPlayer.jpg

表 5.1: データ座標位置とパス

## 5.5 データ操作

### 5.5.1 データ表示

利用者がディスプレイの前に立つと、映し出された利用者の身体の何処にデータが張り付いているかが分かる程度の大きさでデータが表示される。画像データは頭頂座標の初期化終了時と同時に表示する。この時の画像データは最大 25\*25 の領域内で表示されるようにする。これより小さい場合、利用者にデータの貼りついている位置が視認しづらい状況になってしまう。また、利用者の指先がデータが貼りついている座標へ近づいた場合データは大きく表

示される。この時の画像データは最大 150\*150 の領域内で表示されるようにする。画像が正方形で無かった場合は辺の長さの比を求め、表示領域内に収まるように調整してから表示する。指先の座標を監視するクラスからデータを表示するクラスへと指先の座標データを渡し、データ座標位置を照らし合わせ座標間の距離を計算する。座標間の距離が近かった場合、そのデータの表示を大きくする。指先の座標とデータ座標位置が近いかどうかの判断はデータの元の大きさに依存する。大きく表示される時のデータ表示領域内に指先が入ると近いと判断される。

### 5.5.2 データ選択

データを選択したい時には指先をデータの上に滞留させておく必要がある。しかし、指先は動いている状態でしか検出されない。そこで、指先の座標を監視し、指先の座標が検出されなかった場合 1 つ前に検出された指先の座標を用いる。そうすることで、指先の座標が検出されない状態でも指先が滞留しているように見せかける事が出来る。

滞留が終わってデータが選択状態になると、画像データのサイズから画像データの中心位置を計算する。利用者が画像データのどの位置で指先を滞留させても (図 5.16)、画像データの中心が利用者の指先に貼りつく (図 5.17)。データが選択されると、指先の周囲のデータを拡



図 5.16: 指先の滞留



図 5.17: データの選択状態

大表示しない状態に遷移する。

### 5.5.3 データ移動と貼り付け

データを選択した状態で指先を移動させると、指先の座標を監視するクラスからデータを表示するクラスへと指先の座標データが渡される。それを元に、指先に画像データの中心が来るように再計算し、データが指先に追従するようにしている。

貼り付ける時も選択する時と同様に指先の座標が検出されない場合は 1 つ前に検出された指先の座標を用いて滞留しているかどうかを得るようになる。滞留が終わってデータが身体に

貼りつくとき、その画像データに対応する位置座標キーが更新され、データの移動・貼り付けが行われる。貼り付けが終了すると、指先の状態が身体に貼り付いているデータを拡大表示する状態に遷移する。

#### 5.5.4 データ削除

データ削除も選択する時と同様に指先の座標が検出されない場合は1つ目に検出された指先の座標を用いて滞留しているかどうかを得るようにする。滞留が終わると、Dictionary内のデータを削除し、再描画が実行されデータが削除されるようになる。削除が終了すると、指先の状態が身体に貼り付いているデータを拡大表示する状態に遷移する。この時削除されるのはDictionary内のキーとパスだけであり、コンピュータ上に記録されているデータは消えることはない。こうすることで、利用者の意図しないデータ削除への対策を行っている。

### 5.6 実験

本研究の提案による座標取得手法を用いて実験を行った。実験は、図 5.18 のように身体に4枚の画像データを貼り付けた。それぞれの画像データのデータ座標位置とデータ名は以下のように定めた。

- 右上：データ座標位置 (20, 50)、データ名 (dst1.jpg)
- 右上：データ座標位置 (30, 100)、データ名 (dst2.jpg)
- 右上：データ座標位置 (-20, 30)、データ名 (dst3.jpg)
- 右上：データ座標位置 (-30, 150)、データ名 (dst4.jpg)

今回の実験環境ではY軸は下方向が正の方向になっており、制約により頭から上に手は上がらないため、 $y < 0$  な座標は存在しない。

#### 5.6.1 実験 1

指が画像に触れた時に、指差された画像を正しく認識しているかどうかの実験を行った。実験では、4つの画像を右上 右下 左下 左上という順に1秒間指差しを行うか画像が認識されるまで指差しを行い、対応した画像のデータ名をコンソール上に表示させた。1つの画像につき25回分、計100回の検証を行った。



図 5.18: 実験初期状態

### 実験結果

実験結果を 5.2 に示す。表中「認識」は指差した画像とコンソールに出力された名前が一致した場合にカウントする。表中「誤認識」は指差した画像とコンソールに出力された名前一致しなかった場合にカウントする。表中「非認識」は指差しを行ってもコンソール上に名前が出力されなかった場合にカウントされる。

イメージ名	認識	誤認識	非認識
dst1.jpg	24/25	0/25	1/25
dst2.jpg	25/25	0/25	0/25
dst3.jpg	23/25	0/25	2/25
dst4.jpg	25/25	0/25	0/25

表 5.2: 実験 1 : 結果

### 考察

どの画像もほぼ確実に認識する事が出来た。dst1.jpg に関する非認識は、顔の肌色領域を検出してしまい、重心が指先側によってしまい、指先が正しく認識されなかったためだと考えられる。動いている肌色領域を手と認識しているため、顔領域上で手がぶれてしまうと、顔の肌色領域が手と認識されてしまうため起こると思われる。dst2.jpg 付近は顎のラインに影がついているため、そもそも肌色領域ではないとの認識から dst1.jpg の時のような事は起きないと考えられる。以上の事から顔上で手がぶれてしまった時の対応策が必要である。dst3.jpg にだけ注目し追加の検証を行ったところ、手の角度により指先の認識精度が変わってくると

ということがわかった。手首より下に指先を持ってきた場合は高精度の指先認識率であり、手首より上に指先を持ってきた場合はやや精度が落ちてしまう事が分かった。

## 5.6.2 実験2

身体を移動させた時にし移動前と同じ場所に画像データが貼り付いているかどうかの実験を行った。初期化したときの頭頂座標とキャプチャされた画像における頭頂座標の差分を取得する。次に、初期状態でのキャプチャ画像中でのデータの座標と移動後のデータの座標の差分を取得する。指差した画像データに対し差分を照らし合わせ、差分の差が大きかった場合はERRORと、小さかった場合はCORRECTとコンソール上に表示させる実験を行う。身体のぶれなどを考慮し、誤差の差10ピクセル・5ピクセル・0ピクセルの場合についての実験を行った。

### 実験結果

平行移動・垂直移動といった上下左右の動きに関しては差10ピクセル・5ピクセル・0ピクセルともにCORRECTを出力した。また、カメラに近づく遠ざかるといった前後の動きに関してもともにCORRECTを出力した。しかし、前後移動に関しては図5.19 図5.20のように、図5.18と比べると明らかに貼り付いている位置が違ってしまふ。



図 5.19: カメラに近づく



図 5.20: カメラから遠ざかる

### 考察

上下左右に移動するだけであれば、頭頂座標を中心とした座標系を用いる事でかなりの精度でのデータ座標位置が取得可能である。前後の移動に関しては今回は制約を設けているの

で想定の範囲内ではあったが、今後に向けて何かしらの対応が必要である。カメラに近づく事で、キャプチャされた画像中の身体領域の割合は増え、逆に遠ざかる事で身体領域の割合は減る。このことを用いれば上下左右の移動だけでなく、前後の移動にも対応した座標系を得られるのではないかと考えられる。また、画像データを指差しするためには腕を上げる必要がある。腕を上げての指差しで CORRECT が検出されたことから、腕を上げた場合の重心による頭座標の検出も正しく行えている事がわかった。

## 第6章 まとめ

本研究では、利用者の身体を用いてデータを管理するインタフェースの提案とプロトタイプの実装を行った。利用者はディスプレイに映った自分の身体へ、コンピュータ上に記憶されているデータを貼り付ける事で管理を行う。貼り付いたデータは手を用いて操作する事が可能で、Weiserの唱える自然なインタフェースであると言える。

本研究では、頭頂座標から身体の任意の位置を取得し、データを貼り付けている。頭頂座標はキャプチャされた身体画像の輪郭を取得し、輪郭の重心を用いる事での検出を行った。頭頂座標によるデータ位置の取得精度を確認するために2つの実験を行った。1つ目は貼り付いたデータを指差す事により、正しいデータを認識しているかどうかを、2つ目は身体が動いても身体に貼り付いているデータの位置が変化しないかどうかを検証した。

以上の実験により、頭頂座標による身体位置座標を取得する手法は高精度を保つ事が分かった。また、頭頂座標を取得するために重心を用いる手法も高い精度を保つ事が分かった。

本インタフェースでは、柔らかいストレージで未検討であった、データの操作・管理とデータの保存場所について実現した。また、本インタフェースに自然なインタフェースを用いる事で、コンピュータ技術を意識させないようにした。そのため、コンピュータについて知識がない人でもデータの管理をコンピュータ上で行うより簡単に行う事が可能になった。

## 謝辞

本稿作成にあたり、ご指導くださった田中二郎教授、ならびに高橋伸講師、三末和男准教授、志築文太郎講師には心から感謝致します。

高橋伸講師につきましては、ゼミやミーティングを開いて頂きご意見をくださったことや、相談に乗っていただき感謝の言葉も御座いません。また田中研究室のみなさんにはゼミ、ミーティングを通じ貴重なご意見を頂きました。特にユビキタスチームのメンバーの方々には、ご自身がお忙しい中、議論などを交わしていただき色々な意見を頂きました。ここに感謝の意を表します。

最後に、自分が挫けそうになった時に叱咤激励してくださった家族・友人に改めて感謝いたします。

## 参考文献

- [1] M.Weiser:”The computer for the twenty-first century.”, In Scientific american,pp 94-104. Scientific american,September 1991.
- [2] D.A.Norman:”The Invisible Computer”, MIT Press,1998.
- [3] S.Iwabuchi,B.Shizuki,K.Misue and J.Tanaka:”Natural Storage in Human Body” , Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information and engineering Systems (KES2005),pp430-436.
- [4] 岩淵 志学, 志築 文太郎, 田中 二郎:”柔らかいストレージ : 体と融合したストレージの提案と実装”, 日本ソフトウェア科学会第 21 回大会,2004 年 9 月.
- [5] G.Robertson,M.Czerwinski,K.Larson,D.C.Robbins,D.Thiel,M.Dantzich,”Data mountain:using spatial memory for document management” Proceedings of the 11th annual ACM symposium on User interface software and technology(UIST '98),pp.153-162.
- [6] ARToolKIT,<http://www.hitl.washington.edu/artoolkit/>
- [7] Dan R Olsen.J and Nielsen.T:”Laser pointer interaction,in CHI '01”, Proceedings of the SIGCHI conference on Human factors in computing systems,ACM Press(2001),pp.17-22.
- [8] 小渡 悟,星野 聖:”単眼動画像からの手の位置と形状の推定に基づくポインティングデバイスの提案”, 電子情報通信学会技術研究報告 HIP, ヒューマン情報処理 vol102,No735,pp79-84.
- [9] T.Hisamatsu,B Shizuki,S.Takahashi and J.Tanaka:”A novel click-free interaction technique for large-screen interfaces”, Proceedings of the 7th Asia-Pacific conference on Computer-Human Interaction(APCHI2006),10pages(CD-ROM).
- [10] OpenCV,<http://www.intel.com/technology/computing/opencv/index.htm>
- [11] 色変換,<http://www.cvl.iis.u-tokyo.ac.jp/miyazaki/tech/tech01.html>
- [12] V.Vezhnevets and V.Sazonov A.Andreeva,”A survey on pixel-based skin color detection techniques”, Graphicon2003,13th International Conference on the Computer Graphics and Vision,2003 .

- [13] F.Solina,P.Peer,B.Batagelj,S.Juvan,"15 seconds of fame an interactive,computer-vision based art installation", Proceedings of the 7th International Conference on control,Automation, Robotics and Vision (ICARCV2002),pp.198-204.
- [14] 水谷 啓,鈴木 寿,"色情報に注目した顔領域の抽出", 電子情報通信学会技術研究報告 PRMU・パターン認識・メディア理解, vol.102 No.532(20021213),pp.91-96.
- [15] 加藤 丈和,柴田 智行,和田 俊和,"最近傍識別器を用いた背景差分と色検出の統合", 情報処理学会研究報告 CVIM[コンピュータビジョンとイメージメディア],vol.2004,vol.6(20040122) pp.31-36.
- [16] 松井 望,山本 喜一,"バーチャルキーボード:ビデオ画像からの頑健な実時間指先検出の実現", 日本ソフトウェア科学会プログラミングおよび応用のシステムに関するワークショップ論文集 (2000).
- [17] 吉村 康弘,片山 晋,古谷 博史,"指先情報を用いた Vision-based Interface の提案", 火の国情報シンポジウム 2005.
- [18] 佐々木 博史,黒田 和宏,真鍋 佳嗣,千原 国宏,"『てのひらめにゆう』: ウェアラブルコンピュータ用入力インタフェース", 日本バーチャルリアリティ学会論文誌, vol.7 No.3 pp393-401(2002)
- [19] T.Lee T.Hollerer,"Handy AR:Markerless Inspection of Augmented Reality Objects Using Fingertip Tracking", In Proc.IEEE International Symposium on Wearable Computers(ISWC),Boston,MA,Oct.2007.
- [20] 佐々木 博史,"ウェアラブルコンピュータに適した手を用いたデバイスレスインタフェースの研究", 奈良先端技術大学博士論文,2003.