

Title	直感的身体動作を用いたVRアプリケーションに関する研究
Author(s)	藪, 博史
Citation	
Issue Date	2006-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/266">http://hdl.handle.net/10119/266</a>
Rights	
Description	Supervisor:宮田 一乗, 知識科学研究科, 修士

修 士 論 文

直感的身体動作を用いた  
VR アプリケーションに関する研究

指導教官 宮田 一乗 教授

北陸先端科学技術大学院大学  
知識科学研究科知識社会システム学専攻

450074 藪 博史

審査委員： 知識 宮田 教授（主査）  
知識 永井 助教授  
知識 佐藤 助教授  
知識 金井 助教授

2006 年 2 月

# 目次

第1章 序論	1
第2章 変位情報を用いた VR アプリケーションの実装	
-バーチャル紙相撲 “ トントン ” -	4
2.1 はじめに	4
2.2 トントン Ver.1 の背景	4
2.2.1 背景	5
2.2.2 関連研究	5
2.3 トントン Ver.1 のシステム構成と実装方法	6
2.3.1 トントン Ver.1 のシステム構成	6
2.3.2 ジャンピングパッドの実装	7
2.3.3 ジャンピングパッドの揺れの算出	8
2.3.4 紙力士への力の作用	9
2.3.5 紙力士の干渉チェック	10
2.3.6 紙力士の衝突後の挙動	12
2.3.7 勝敗判定	13
2.3.8 映像生成と表示法	14
2.4 トントン Ver.1 の実験結果	15
2.5 トントン Ver.1 の考察と課題	16
2.6 トントン Ver.2 の背景	17
2.6.1 トントン Ver.1 の問題点	17
2.6.2 トントン Ver.2 の考案	18
2.7 トントン Ver.2 のシステム構成と実装方法	18
2.7.1 トントン Ver.2 のシステム構成	18
2.7.2 トントンタンクの実装と測距センサの配置	19

2.7.3 紙力士への力の作用	20
2.7.4 紙力士の干渉と衝突後の挙動	21
2.7.5 勝敗判定	22
2.7.6 映像生成と表示法	22
2.8 トントン Ver.2 の実験結果	23
2.9 トントン Ver.2 の考察と課題	25
2.10 考案	26
<b>第3章 手指認識によるキャラクタアニメーション</b>	
<b>-バーチャル手使い人形“トムテ”-</b>	<b>28</b>
3.1 はじめに	28
3.2 背景	28
3.3 関連研究	30
3.4 システム構成と実装方法	31
3.4.1 システム構成	31
3.4.2 手認識手法	32
3.4.3 指認識手法	34
3.4.4 最適化	36
3.4.5 キャラクタ描画	40
3.4.6 キャラクタアニメーション	42
3.4.7 映像生成と表示法	44
3.5 実験結果	44
3.6 考察	45
<b>第4章 結論</b>	<b>47</b>
<b>付録 本研究に関する研究発表</b>	<b>49</b>
<b>謝辞</b>	<b>50</b>
<b>参考文献</b>	<b>51</b>

# 目 次

1.	トント Ver.1 のシステム構成図	6
2.	ジャンピングパッドの実装	7, 8
3.	センサの出力値と距離の関係	8
4.	ジャンピングパッドの姿勢の算出	9
5.	ジャンピングパッドとプレイヤーの位置関係	10
6.	三角形の干渉チェック	11
7.	作用点の決定	11
8.	仮想球による衝突後の挙動の計算	13
9.	トント Ver.1 の出力映像	14
10.	トント Ver.1 の設置の様子	15
11.	トント Ver.2 のシステム構成図	19
12.	トント Ver.2 の実装	20
13.	WGC からの作用力	21
14.	バンプマッピング	22, 23
15.	トント Ver.2 の出力映像	23
16.	トント Ver.2 の体験の様子	24
17.	SIGGRAPH での体験の様子	25
18.	関連研究	31
19.	トムテのシステム構成	32
20.	肌色領域の抽出	33
21.	画像中にある肌色ヒストグラム	35, 36
22.	ダウン・アップサンプリング処理	37
23.	膨張処理	38
24.	ラベリング処理(1)	39

25 .	ラベリング処理(2).....	40
26 .	ノイズ除去 .....	40
27 .	座標の違い .....	41
28 .	キャラクタアニメーション(1).....	43
29 .	キャラクタアニメーション(2).....	43
30 .	キャラクタアニメーション(3).....	43
31 .	トムテの出力画像 .....	44
32 .	トムテのプログラム .....	45
33 .	トムテの体験の様子 .....	45

# 表 目 次

1 .	トントン Ver.1 に対するアンケート結果 .....	16
2 .	トントン Ver.1 への代表的な意見およびアドバイス .....	16
3 .	トントン Ver.2 に対するアンケート結果 .....	24
4 .	トントン Ver.2 への代表的な意見およびアドバイス .....	24, 25

# 第1章 序論

インターフェースとは、2つのもの間に立って、情報のやり取りを仲介するという意味である。IT 分野におけるインターフェースは、ハードウェアやソフトウェアおよび、ユーザを互いにつなぐものしてとらえることが出来る。1970 年代において計算機は、主として専門のオペレータやプログラマ、技術者・研究者などによって運用・利用されていた。その後、1980 年代から計算機は徐々にオフィス内の一般ユーザにより使われるようになり、作用用途が広がったため、計算機のユーザインターフェースは劇的な進化を遂げた。ユーザインターフェースは、コンピュータがユーザに対して情報を表示する方式や、ユーザが情報を入力するための方式を定めたものであり、当時は CUI (Character User Interface) が主流であった。これは、「copy」や「format」などの命令 (コマンド) を、操作のたびに打ち込む必要があり、各命令に対する専門知識を必要とした。コンピュータが普及し始めた頃から、長い間 CUI が使われてきたが、コンピュータの性能向上・大衆化に伴い、グラフィック表示やマウス操作などを多用した GUI (Graphical User Interface) が使われるようになった。GUI では、コンピュータの画面上に、ウィンドウ、アイコン、ボタンといったグラフィックが表示され、ユーザは、目的の操作をマウスなどのポインティングデバイスを用いて行う方法である。GUI は、操作の習熟に対する労力が CUI と比較して少なく、必要な専門知識も多くない。GUI なくしては、今の情報化社会は築けなかったといっても過言ではない。

近年では、GUI が広く活用され、健常者の誰もがコンピュータを容易に操作できるようになった。しかし、IT が世界中に普及するに伴い、デジタルデバイド (情報格差) という現象が発生した。デジタルデバイドとは、情報機器を使える人と使えない人との格差が拡大することで地域格差や収入格差が拡大し、社会全体の不安定要因ともなりかねない現象である。そこで、現在では、このような格差をなくし、IT 技術を利用しようと思えば、誰でも、どこでも、簡単に利用できる環境が切望されている。すなわち、パソコンを満足に扱えない人々、例えば子供やコンピュータの操作に習熟していない人々による利用、障害者による利用、また移動中・作業中に机に向かって

いない状態での利用などを支援する，新たなユーザインターフェースが求められている．

これに対し，人間の意図や注意，興味をシステムに判断させるユーザインターフェースである，PUI ( Perceptual User Interface ) が開発されている．PUI とは，人間の身振り，声，視線等を入力とするもので，これにより，現在のインターフェースと比較して，より自然な形で機械との協調作業，つまりコンピュータをもっと直感的に操作することが可能になる．PUI を用いたアプリケーションの例として，操作者が半透明のスクリーンに表示された情報を，スクリーン前で身振りにより表示部分の移動，拡大等を行うことができるシステム[25]，熱画像を用いたタブレット型入力装置とそれを用いた描画アプリケーションで，体温を持つ手指・呼気を直接用いて描画することが可能なシステム[26]，利用者が声を発することで，画面に線が描かれ，声の大きさや高さを変化させることで，線の色，太さ，描かれる方向を操作し，一枚の絵を描いていくことができるシステム[27]などがある．CUI でのコマンド入力や，GUI でのポインタ操作と比較して，PUI は操作が簡便であるため，今後は GUI に替わって普及すると考えられる．

一方で，子供たちが遊ぶゲームの形態も大きく変わりつつある．操作キーとボタンのみをインターフェースとしたゲームが長い間楽しまれてきたが，最近では，足のステップや太鼓を叩くという身体動作を入力とした体感型ゲームが多くの人に受け入れられる傾向がある．「ピンボール」[28]を初め，昔よりアーケードゲーム機では，多くの体感型ゲームが存在していた．しかし，家庭用ゲーム機が普及したことにより，アーケードと家庭用ゲームとで操作の共有化が考慮され，アーケードは一時期，十字キーとボタンによる操作が主流となった．しかし，最近では，「Dance Dance Revolution」[2]や「たいこの達人」[3]などの，十字キーやボタンを使わない，いわゆる体感型ゲームが遊ばれる傾向が見え始めている．これは，プレイヤーが，身体的動作による入力方法が直感的であり，操作自体を楽しんでいることが，大きな要因であると考えられる．前述した PUI が求められているのと同じように，ゲーム業界においても，誰でも操作が簡単にできるインターフェースが求められているのだと考えられる．

このように，現在ではコンピュータと人間との距離を縮める方向を目指したインターフェースの高度化が精力的に行われている．ユーザインターフェースの進化として，

メタファを使った間接的な命令手法から，直接入力が可能となる手法へと変貌しつつある．このような手法を実現するには，いかに，システム内での複雑な処理をわかり易く表示するか，またいかに直接的な入力を認識するかが重要となり，これを実現するのは仮想現実（VR）の技術であると考える．

本研究では，VR 技術による直感的なインタラクションの実現を目指した2つのアプリケーション開発を通じて，身体動作による新たな PUI の可能性の解明に取り組んだ．

# 第2章 変位情報を用いた VR アプリケーションの実装 -バーチャル紙相撲 “ トントン ” -

本章では、人が跳ねる、物を押すという直感的身体動作によって、紙相撲を遊ぶことができる VR アプリケーションについて述べる。

## 2.1 はじめに

対戦型のビデオゲームは、依然として高い人気を誇るゲームの1つである。一方で、音楽のリズムに合わせてダンスを踊る、太鼓を叩くなどの体感型ゲームも、爆発的な広がりを見せている。この要因には、プレイヤーの全身を使つての直接的な操作法が、従来のゲームコントローラによる間接的な操作法と比較して、操作が容易であり、かつ、理解しやすいという、インタラクションモデルおよび、ヒューマンインタフェースの観点からの利点が挙げられる。

本章では、始めに、プレイヤーの飛び跳ねる動作を入力とする、変位情報を用いた新たなインタラクション手法を提案し、その応用事例として紙相撲をテーマにした VR アプリケーション (トントン Ver.1 [8]) を紹介する。つづいて、トントン Ver.1 の評価実験の結果を反映させた、プレイヤーが水を揺らす動作を入力とする VR アプリケーション (トントン Ver.2[??]) を紹介する。

## 2.2 トントン Ver.1 の背景

本節では、新たなインタラクションの可能性を探り、制作した VR アプリケーションの研究背景について述べる。

## 2.2.1 背景

紙相撲は、1対の紙力士を土俵上に立て、土俵の傍を指で叩くことで紙力士を操作して勝負をするという簡単な遊びである。紙相撲を遊ぶ上での一番の面白みは、指で叩くというアナログ的な行為によって生じる、予想の困難なカオティックかつ突発的な紙力士の独特な動きにあると考える。また、参加者のスキル差も少なく、老若男女を問わず、誰もが気軽に遊ぶことができるのも魅力の1つである。

本章では、ゲームパッドのような複雑な操作が必要なデバイスではなく、身体動作に直結した、直接的な操作を可能とする新たなインタラクション手法の開発を第一の目的とした。研究の初期段階では、プレイヤーがテーブルを叩く音を、テーブル面に設置された複数個のマイクで拾い、取得された音を解析する方法を考案した。しかし、対戦時に熱中のあまりにテーブルを強打することによるプレイヤーの負傷の可能性が残されることから、この入力法は却下された。そして新たなインタラクション手法として、適度な全身の身体運動を伴い、かつ、負傷の可能性が低く、プレイヤーばかりでなく、観客も楽しめるジャンピングインタラクションを考案した。

## 2.2.2 関連研究

“バーチャファイター” [1] に代表される対戦型のビデオゲームは、既に数多く市場に出ているが、ゲームパッドやジョイスティックによる、キャラクタの複雑な操作を必要とする。一方、叩く行為を入力とするものは、“Dance Dance Revolution” [2] に代表される音楽のリズムに合わせて踊るもの、最近では“太鼓の達人” [3] のように音符に合わせて太鼓を演奏するものがある。また“Jam-O-Drum” [4, 5]では、リズムに合わせてテーブルに設置されたドラムパッドをタッピングし、複数人でドラムのジャムセッションを楽しむ環境を提供する。

これらの研究では、指先、もしくは腕や足の動作を何らかの入力デバイスで取得し、インタラクションする手法を提案しているが、本章で提案しているような飛び跳ねる全身の身体動作を入力とするものは、あまり例がない。

“太鼓の達人” が好評を博している要因のひとつは、プレイヤーのバチさばきの動作

を観客が見て楽しみ、自分も体験してみたいという感情を沸き立たせるところにあると考える。本章でのジャンピングインタラクションは、より大きな身体動作を伴い、かつ、単純な直接入力法であるため、完成度の高い“太鼓の達人”とは単純比較はできないが、「誰も」が「気軽」に「観客と共に」楽しめる方法であると考える。

## 2.3 トント Ver.1 のシステム構成と実装方法

本節では、トント Ver.1 のシステム構成とその具体的な実装方法について述べる。

### 2.3.1 トント Ver.1 のシステム構成

本システムは、図1に示すように、プレイヤーの動作入力デバイスであるジャンピングパッド2基、A/D変換ボードと2系統出力機能を持つグラフィックカードを搭載したPC1台、および映像出力系であるプロジェクタ、反射鏡、映像投影スクリーンを2組から構成されている。なお、ジャンピングパッドの四隅には、床面からの距離を測定するセンサが設置されており、A/D変換ボードを介してリアルタイムで測定するものとする。

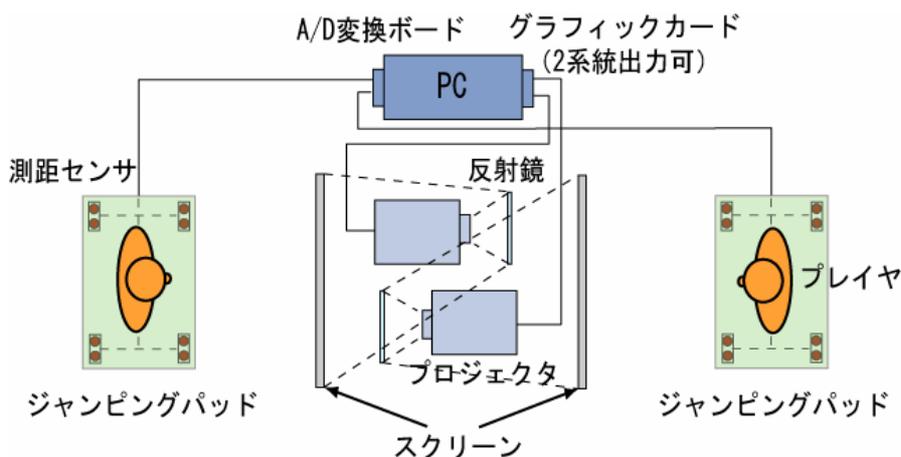


図1: トント Ver.1 のシステム構成図

システム内での処理の流れを以下に示す。

- (P1) ジャンピングパッドに設置された測距センサにより，床面とジャンピングパッド間の距離を測定する．
- (P2) 測定結果からジャンピングパッドの揺れを算出する．
- (P3) 揺れの情報を基に，紙力士への影響をシミュレートする．
- (P4) 紙力士同士の干渉，および応力を計算する．同時に，勝敗の判定を行う．
- (P5) 各プレイヤーの視点に応じて紙力士の動作映像を生成し，プロジェクタを介してスクリーンに投影する．

### 2.3.2 ジャンピングパッドの実装

ジャンピングパッドは，図 2 に示すような，125mm×92mm×15mm の木板の底面に，タイヤチューブ(膨らました状態で半径約 250mm， 径約 90mm)を並列して 2 個，測距センサ (SHARP 製，パーツ番号：GP2D120) を 4 隅に設置したものである．プレイヤーがジャンピングパッド上で体を動かすことでパッドが揺れ，その揺れに応じて，測距センサが取得する距離データが変化する．

システムで使用した測距センサは，赤外光を発受光して反射物までの距離を電圧として出力するものであり，測定範囲は 5～30cm である．測距センサからの出力電圧は，A/D 変換ボード(インターフェース社，PCI-3133，12 ビット 16 チャンネル入力)により，デジタルデータへと変換される[9]．



(a) ジャンピングパッドの裏面

(4 隅に設けられているのは，センサ保護用のゴムパッド．測距センサはマジック

クテープでそれらの脇に固定される.)



(b) 測距センサ (c) プレイヤが乗った場面

図2: ジャンピングパッドの実装

### 2.3.3 ジャンピングパッドの揺れの算出

ジャンピングパッドの揺れの算出にあたり, まず, 測距センサから取得されたデータを, 距離のデータに変換する必要がある.

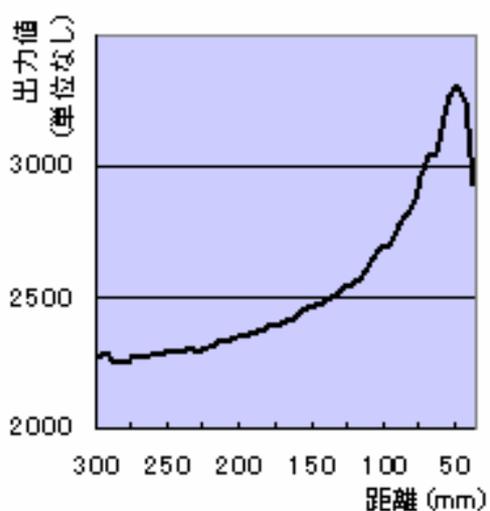


図3: センサの出力値と距離の関係

(センサと反射物との距離が近づくにつれて, 出力値は増加する)

図3に、センサから取得されたデータと距離の関係を示す。測定される距離に応じて、センサからは $\pm 5V$ の範囲の電圧が発せられる。この出力電圧をA/Dコンバータを介して12bitで変換した値が図3の出力値である。これらの実測値を用いて、センサからの出力情報を距離の情報へと変換する。そして、ジャンピングパッドの4隅に設置された測距センサからリアルタイムで距離のデータをサンプリングし、サンプリングした時点でのジャンピングパッドの姿勢を算出する。

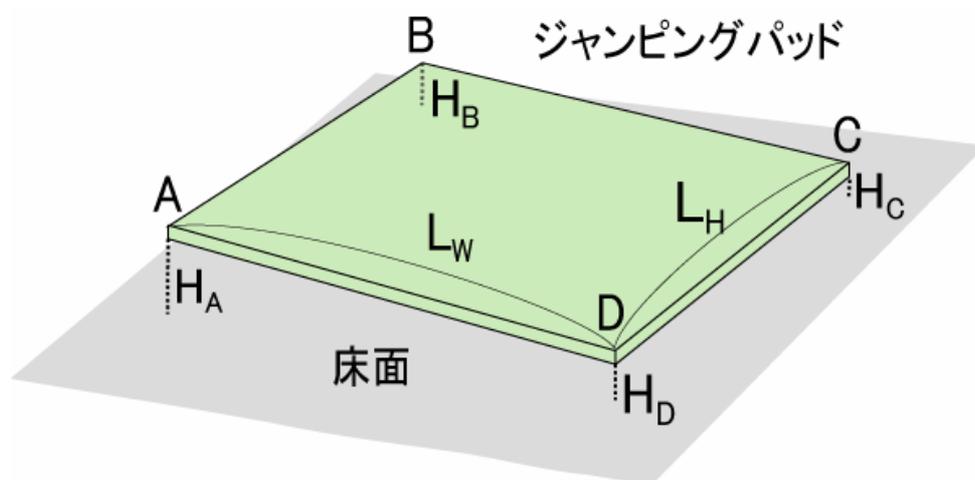


図4： ジャンピングパッドの姿勢の算出

ジャンピングパッドの姿勢の算出は、あらかじめ測定したジャンピングパッドの縦横の長さ ( $L_W$ ,  $L_H$ ) と、4隅の床面からの距離情報 ( $H_A$ ,  $H_B$ ,  $H_C$ ,  $H_D$ ) を用いておこなう。すなわち、図4における ABD および、BCD の3次元空間内での頂点座標値から、それぞれの三角形の法線ベクトルを求め、その平均をジャンピングパッドの法線ベクトルとする。また、法線ベクトルの変化量をサンプリングの時間間隔  $t$  で割ることで法線ベクトルの速度  $V_n$  を、さらに、 $V_n$  を  $t$  で割ることで、法線ベクトルの加速度  $A_n$  を求める。

### 2.3.4 紙力士への力の作用

紙力士への力の作用は、2.3.3節で求められた  $V_n$  および  $A_n$  を用いて行う。

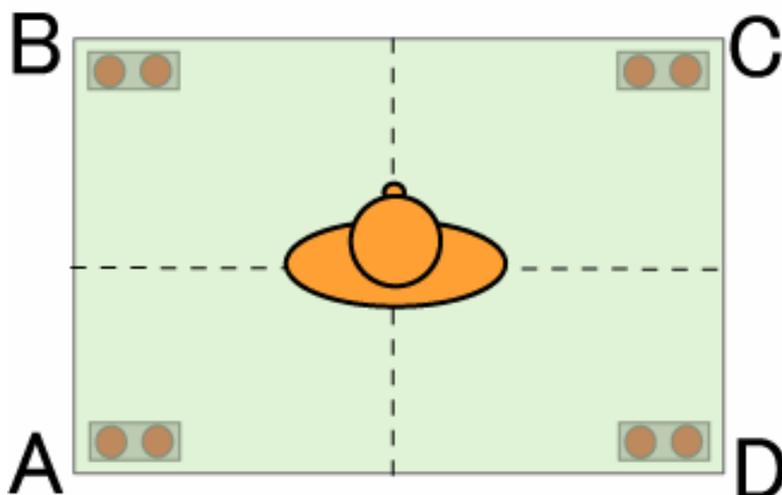


図 5： ジャンピングパッドとプレイヤーの位置関係

プレイヤーは、図 5 に示すように、ジャンピングパッドの中心位置に辺 AD を背に、A→B 方向を向くものとする。

ある時間における紙力士の位置を  $P_1$  とした場合、 $t$  時間後の位置  $P_2$  は、式(1)で与えるものとする。

$$P_2 = P_1 + V_n \cdot t \quad (1)$$

但し、ジャンピングパッドの反動で意図とは異なる方向に力士が移動する可能性があるため、 $|A_n|$  がある閾値より大きい場合のみ力士を移動させる。ここで、閾値は経験的に設定した。

### 2.3.5 紙力士の干渉チェック

紙力士の干渉チェックは、以下の 2 段階の精度で行う。

まず、精度の粗い判定法として、境界球を用いた干渉チェックを行う。すなわち、境界球をそれぞれの紙力士に割り当て、それぞれの球が重なり合った時点で干渉の可能性があると判断し、制度を上げた 2 段階目の干渉チェックを行うものとする [5, 6, 10]。

2 段階目の干渉チェックでは、紙力士を構成する各ポリゴン（実際には三角形メッシュ）レベルでの干渉チェックを行う。すなわち、それぞれの力士を構成する三次元

三角形同士の干渉チェックを，総当りで行うものとする．

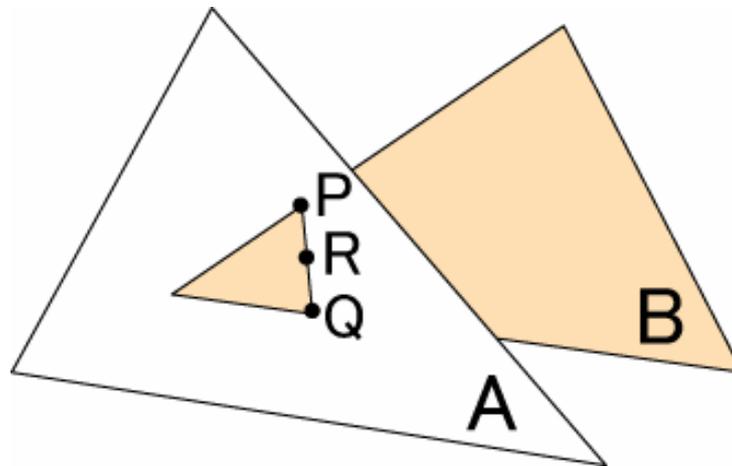
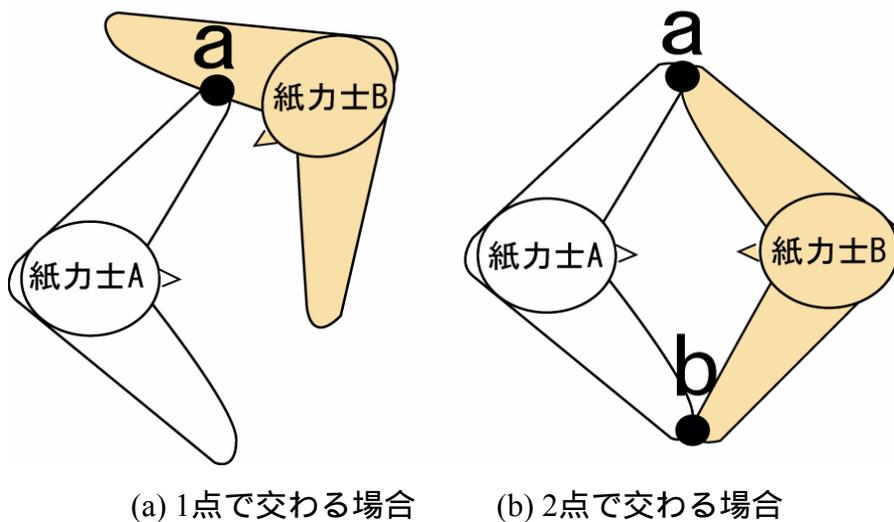


図 6： 三角形の干渉チェック

図6に示すように，三角形 A，B の干渉チェックでは，まず三角形 A と，三角形 B を構成する3つの辺との交差判定を行う．もし，3辺とも交差が無い場合は，この逆の判定，すなわち，三角形 B と，三角形 A を構成する3つの辺との交差判定を行う．このようにして求めた交点 P，Q の中点 R を，力の作用点とする．



(a) 1点で交わる場合 (b) 2点で交わる場合

図 7： 作用点の決定

ここで、図7に示すように、複数の作用点が存在する場合がある。図7は試合中の様子を上方から見たものである。図7(a)に示すように、紙力士同士が、1点のみで交わる場合は、上記の方法で作用点を求めるが、図7(b)で示すように、同時に2点以上で干渉する場合は、総当りでの衝突判定時に最初に求められた交点(a, bのどちらかになる)のみを作用点とした。そのため、実際の紙力士の挙動とは異なる場合も生じる。

実装に用いた紙力士のポリゴン数は、平均して約900枚であり、実時間での衝突判定が可能であった。なお、力士形状の簡素化は、衝突判定に大きな影響のない程度に留め、かつ、衝突判定の効率化が図れるものになっている。

### 2.3.6 紙力士の衝突後の挙動

紙力士の衝突後の挙動は、紙力士を仮想的に球体とみなして計算する。また、紙力士は衝突判定時に土俵に設置しているものと仮定し、以降の計算は、図8に示すような土俵上に球を投影した2次元での計算とする。ここで、2.3.5節で求めた作用点を球の衝突点とする。

衝突応答は、両力士の衝突時の速度ベクトルから、運動量保存則を用いて行う。両力士の正確な衝突後の動きを計算する為、図8に示すように、衝突の際の速度ベクトルをX'Y'直交座標系に座標変換して計算を行う。座標変換はXY座標軸を原点を中心に $-\theta$ 回転させるものとする。ここで、X'軸は2つの球の中心を通る直線、Y'軸はこれに直交するものとし、 $\theta$ はX軸とX'軸のなす角とする。

$V_1, V_2$ を座標変換した後の新たな速度ベクトル $V_1', V_2'$ に対して、以下のようにして、衝突後の速度ベクトルを求める。ここで、2体の紙力士の質量を、それぞれ $m_1, m_2$ とする。実装時の力士の質量は、重い力士、軽い力士などの紙力士の特徴を考慮して、その値を経験的に設定している。

まず、衝突前後の速度ベクトルの関係式は、式(2)で表される。ここで、 $e$ は跳ね返り係数で、パラメータとして与え、システムの施行実験の過程で、経験的に設定している。

$$e \cdot (V_1' \cdot x - V_2' \cdot x) = -(V_1 \cdot x - V_2 \cdot x) \quad (2)$$

次に、式(3)に示す運動量保存則により、X'Y'座標軸上での衝突後の速度ベクトル

$V_1''x$  ,  $V_2''x$  は、それぞれ式(4) , (5)で与えられる .

$$m_1 \cdot V_1''x + m_2 \cdot V_2''x = m_1 \cdot V_1'x + m_2 \cdot V_2'x \quad (3)$$

$$V_1''x = \frac{(m_1 - m_2 \cdot e) \cdot V_1'x + (1 + e) \cdot V_2'x}{m_1 + m_2} \quad (4)$$

$$V_2''x = \frac{(1 + e) \cdot V_1'x + (m_2 - m_1 \cdot e) \cdot V_2'x}{m_1 + m_2} \quad (5)$$

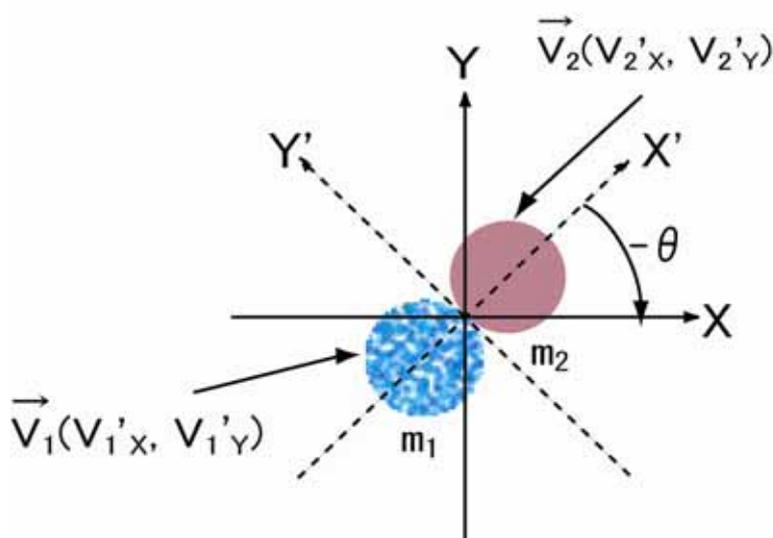


図 8 : 仮想球による衝突後の挙動の計算

以上の各力士の速度ベクトルの算出後、座標軸を  $\theta$  だけ回転して元に戻すことで、 $XY$  座標軸における各力士の衝突後の速度ベクトルが求められる .

挙動計算においては、仮想的に球として衝突後の挙動計算をしている . すなわち、力士の回転運動などは考慮されておらず、球の単純な衝突の挙動のみ扱うものとしている . したがって、実際の紙力士とは違う動作をする場合もある .

### 2.3.7 勝敗判定

勝敗の判定条件には以下の 3 つの条件を設定し、いずれかを満たしたときに勝敗を決めるものとする .

(J1) 土俵を割る : プレイヤが操作する紙力士の中心が設定された土俵枠の外側に位

置した場合、そのプレイヤーの負けとする。

(J2) 体力がなくなる：プレイヤーが操作する紙力士が衝突のたびに、体力パラメータを後述する手法で減少させ、先に0になったほうを負けとする。

(J3) 時間切れ：勝敗が決せず、一定時間（ここでは30秒に設定）経った場合、引き分けとする。

判定条件 J1 に関しては、紙力士の中心座標値と土俵枠との、内外判定を行えばよい。すなわち、丸い土俵枠であれば、土俵の中心からの距離と土俵の半径との関係で容易に判定可能である。判定条件 J2 に関しては、両者のジャンピングパッドから得られる法線ベクトルの加速度  $A_n$  を評価して行い、加速度の絶対値が小さいプレイヤーの力士に対し、体力パラメータを一定量減らすこととし、その残量で判定する。判定条件 J3 に関しては、試合経過時間を計算すればよい。

### 2.3.8 映像生成と表示法

紙力士の映像は、図9に示すように、DirectX Graphics を用いてリアルタイムで描画し、各プレイヤーの正面に設置されたスクリーンに表示させる。ジャンピングパッドとスクリーンとの物理的な位置関係と制約条件から、床面に設置したプロジェクタから投影された映像は、鏡で反射されてスクリーンに背面投射させることとした。

投影されるCG映像は、1枚のグラフィックスカードで、2名のプレイヤーに対して設定されたカメラパラメータ（それぞれに対してビューアングルなどを変更）を用いて、2画面分がリアルタイムで同時に生成される。



図9： トントン Ver.1 の出力映像

## 2.4 トントン Ver.1 の実験結果

体験システムは 4m×2m の設置スペースに，図 10 に示すように設置される．ここで，プレイヤーの向き合う方向が，設置スペースの長手方向となる．

投影スクリーンには障子紙を用い，塩化ビニール製のパイプを加工して作成したフレームに固定した．なお，垂直に設置された塩化ビニール製のポールは，セメントを流し込んだ金属製の缶で固定されており，スクリーン枠を滑り止めのゴムワッシャを介してボルトで支持している．実験に用いた PC は，DELL 製 DimensionXPS (Pentium4 3GHz，1G メモリ)で，グラフィックスカードとして，GeFORCE 6800 Ultra (256MB) を用いた．

実験の結果，2 台のジャンピングパッドからの入力に合わせて，30fps の映像出力速度を保って体験することができた．



図 10： トントン Ver.1 の設置の様子

2004 年 8 月 31 日に東京科学未来館にて開催された IVRC (国際バーチャルリアリティコンテスト) 東京予選大会において，本手法の評価実験を行った．表 1 にアンケ

ートの集計結果（総数 82 件）を示し，表 2 に代表的な意見やコメントを示す．

表 1： トント Ver.1 に対するアンケート結果

Ver.1 の評価項目	満足 (%)	普通 (%)	悪い (%)
台の踏み心地	31.7	54.9	13.4
楽しさ	78.0	20.7	1.2
力士の数	20.0	53.8	26.3
力士のデザイン	36.6	34.1	29.3
力士の操作	45.1	30.5	24.4
映像の質	28.0	62.2	9.8

表 2： トント Ver.1 への代表的な意見およびアドバイス

代表的な意見	代表的なアドバイス
<ul style="list-style-type: none"> <li>・ 簡単で面白い</li> <li>・ 体力が必要で よい汗をかいた</li> <li>・ 飛べて楽しい</li> <li>・ 相手が子供だっ たのに負けた</li> </ul>	<ul style="list-style-type: none"> <li>・ 力士の操作性がほしい</li> <li>・ 必殺技を増やしてほしい</li> <li>・ キャラクタに特性をつけてほし い</li> <li>・ 力のフィードバックがほしい (台が震えるなど)</li> </ul>

評価実験の結果，面白さの点で高評価を受けた．その理由としては，力士の操作が単純かつ明確であること，ジャンプする行為の面白さなどが挙げられる．しかし，映像の質や表現法に関しては，ある程度の評価を受けたものの，力士の技が少ないため，そのほとんどが押し相撲となり，紙相撲の試合が単調化するなどの問題点が指摘された．

## 2.5 トント Ver.1 の考察と課題

トント Ver.1 では，プレイヤーの飛び跳ねる行為が，CGの紙力士にリアルタイムで反映され，今までにない新たなインタラクションを実現することができた．

課題としては，まず紙力士の技の種類を増やすこと，力士に特性をつけることなど

が挙げられる。また、摩擦力を考慮した厳密な剛体の物理シミュレーションの適用も必要であろう。入力デバイスであるジャンピングパッドに関しては、前後方向のみの操作でも勝敗が決するため、その操作法に改善の余地があると考えられる。

一方、今回の評価実験において、改善のための意見として、フィードバックの問題が挙げられた。本システムは、構造上、相手のジャンプ行為に伴う振動、すなわち力の伝達をお互いに感じるができない。したがって、力のフィードバックが実現できれば、より楽しいインタラクションを実現できるのではないかと考え、以降述べるトント Ver.2 では、この部分にも注力した。

## 2.6 トント Ver.2 の背景

本節では、トント Ver.1 での問題点などを考察し、その考察結果を反映して制作した VR アプリケーションの研究背景について述べる。

### 2.6.1 トント Ver.2 の問題点

トント Ver.1 では、プレイヤーが飛び跳ねるという行為に CG の紙力士の動きがダイレクトに反映され、紙力士を前や横などに移動させて対戦するというインタラクションを楽しむことができた。これにより、自分があたかも紙力士自身となって跳ねている、という感覚を得ることができる。しかし、これは紙力士をプレイヤーの思い通りに前後左右に操作することができるということであり、予測不可能なカオティックな動きをするという紙相撲ならではの楽しさとはまったく異なるものである。したがって、いたずらに力士の技を増やしたりするだけの改良では、トント Ver.1 の「ゲーム性」を高めるだけにとどまってしまうのではないかと考えた。また、力のフィードバックを実現する手段は大掛かりになると判断し、トント Ver.1 の延長線上での改良はやめることにした。

そして、ゲーム性の高いトント Ver.1 とは異なる紙相撲の実現システムトント Ver.2 を新たに考案した。すなわち、元来の紙相撲の面白みである、自分の力士を直接操作したくても操作できない、予測不可能な動きをする紙力士を再現し、かつ、力の伝わりをより明確に感じるができるインタラクションモデルの考案および、

VR アプリケーションの開発を目指した。

## 2.6.2 トント Ver.2 の考案

トント Ver.2 では、トント Ver.1 における直感的な身体的インタラクシインターフェイス性を追求し、新たなインターフェースを考案した。すなわち、トント Ver.1 の実験および評価の結果から得られた、力のフィードバックなどの問題点を踏まえ、対戦相手から力の伝達を得ることが出来るような新たなシステムを考案した。

トント Ver.2 では、水を媒体とすることで対戦相手からの力の伝播と、予測の困難なカオティックかつ突発的な紙力士の独特の動きの再現を試みた。また舞台設定を水中での紙相撲とすることで、波紋を用いて力の伝わりを視覚化することも試みた。

## 2.7 トント Ver.2 のシステム構成と実装方法

本節では、トント Ver.2 のシステム構成とその具体的な実装方法について述べる。

### 2.7.1 トント Ver.2 のシステム構成

本システムは、図 11 に示すような 4 つのモジュールから構成される。各構成要素は以下のとおりである。

- (1) 水面上に浮かぶスクリーンに映像を投影するプロジェクタ。
- (2) プレイヤが押下し波を起こすためのウェーブジェネレータキューブ（発泡スチロール製、以下、WGC と略す。）を、3 個 2 対配置したトントタンク（アクリル製の水槽）。
- (3) スクリーン 4 隅および、6 個の WGC の下部に配置された、距離を測定するセンサ群。
- (4) A/D 変換ボードを介して(3)からのデータを取得し、システム全体を制御するための PC 1 台。

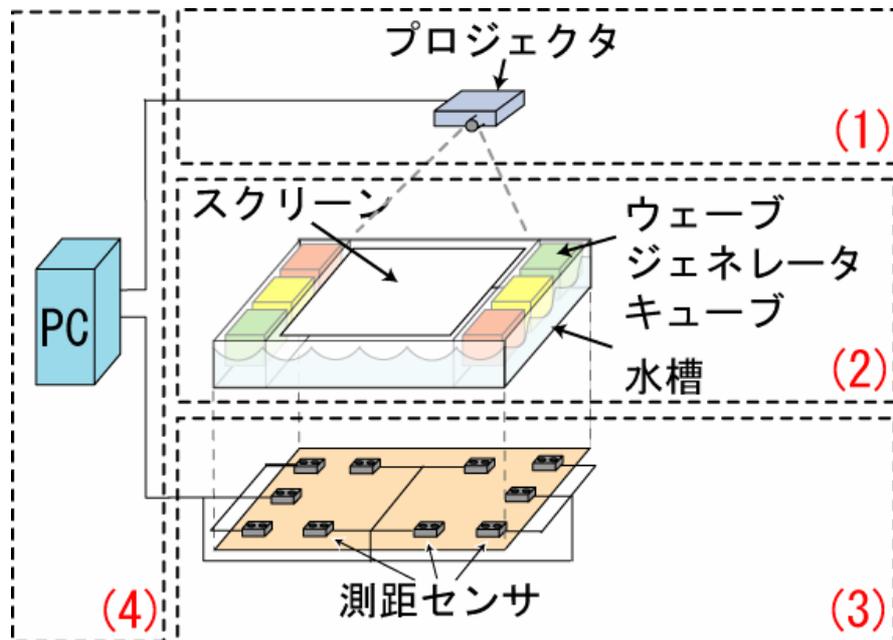


図 11： トントン Ver.2 のシステム構成図

システム内での処理の流れを以下に示す。

- (P1) 測距センサにより、6 個の WGC および、スクリーンの 4 隅の水槽底面からの距離を測定する。
- (P2) 測定結果から、WGC の押下の速度とスクリーンの傾きを算出する。
- (P3) P2 の情報を基に、紙力士への影響を算出する。
- (P4) 紙力士同士の干渉、および応力を算出する。同時に、勝敗の判定を行う。
- (P5) 紙力士の動作映像を生成し、プロジェクタを介してスクリーンに投影する。

## 2.7.2 トントンタンクの実装と測距センサの配置

トントンタンクには、図 12 にあるように、1060mm×640mm×160mm のアクリル製の水槽を用いた。また、耐久性を考慮し、アクリルの板厚を 10mm とした。この水槽に水を深さ 50mm 程に入れ、水槽の両端に 185mm×190mm×100mm の WGC を 6 つ浮かべ、水槽の中央に 600mm×600mm×5mm の白のスチレンボードを浮かべて配置する。また、水槽の下には、図 11(3)に示すように、スクリーンの下 4 隅および、各 WGC のほぼ中央にそれぞれ 1 個ずつ測距センサを配置した。さらに、プレイヤーへの防水対策

として，WGC 上部をビニールで覆った．

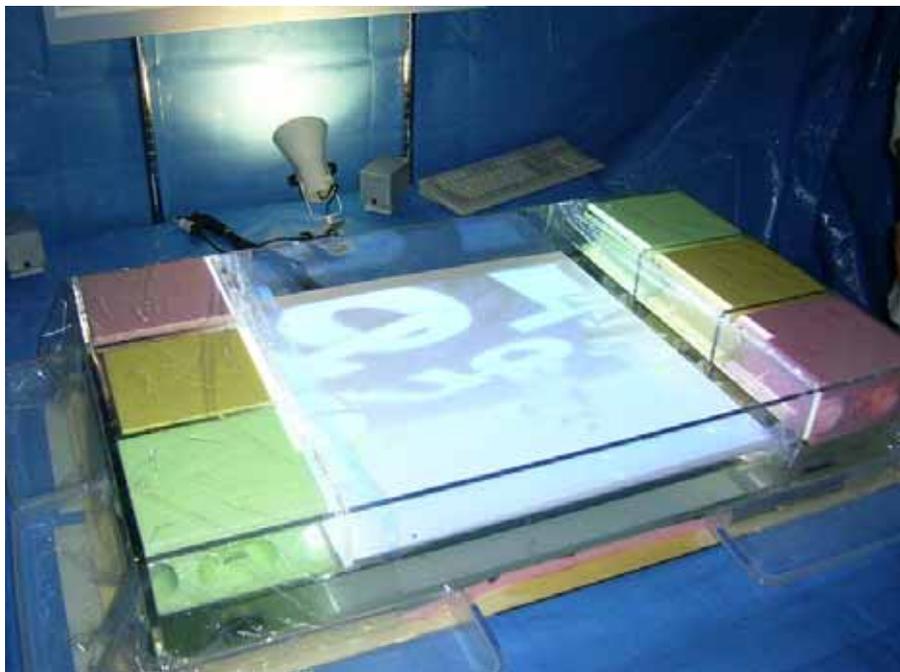


図 12： トントン Ver.2 の実装

変位情報の取得には，2.3.2 節と同じ測距センサを用いた．測距センサは，水槽底面を隔てて上向きに設置され，WGC やスクリーンまでの距離を測定する．2.3.3 節で述べた手法で，リアルタイムで変位情報を取得し，WGC に対しては押下の速度を，スクリーンに対しては傾きを算出する．スクリーンの傾きの算出は，2.3.3 節のジャンピングパッドの姿勢の算出と同じであり，スクリーンの法線ベクトルの速度を用いて求める．

### 2.7.3 紙力士への力の作用

紙力士へ及ぼす力は，以下の 2 力を与えるものとする．

- (1) WGC からの作用力: WGC は紙相撲のフィールドと同じ座標系にあるものとして，以下の計算を行う．各プレイヤーに対して 3 つの WGC で最も移動速度が高い WGC を求める．ここで，求めた WGC を  $W_1, W_2$ ，その移動速度を  $V_1, V_2$  とし，図

13 に示すように、2 体の紙力士に向かって、 $W_1, W_2$  の中心から  $V_1, V_2$  の大きさに応じた力を作用させる。図 13 の点線の矢印は  $W_1$  からの、実線の矢印は  $W_2$  からの、それぞれ力の大きさと作用方向を示している。ここで、 $W_1, W_2$  の中心位置をそれぞれ  $C_1, C_2$  とすると、 $P_1$  に位置する紙力士の移動後の位置  $P_2$  は、式 (6) で与えられる。

$$P_2 = P_1 + |V_1|(P_1 - C_1) + |V_2|(P_1 - C_2) \quad (6)$$

- (2) 土俵からの作用力：土俵からの 2 体の紙力士に対する作用力の計算には、スクリーンの法線ベクトルの速度を用い、各力士の新たな位置を式(1)で求める。ただし、2.3.4 で述べたような法線ベクトルの加速度による閾値処理は行わない。以上で求められた移動量を合算して、最終的な各力士の位置を決める。

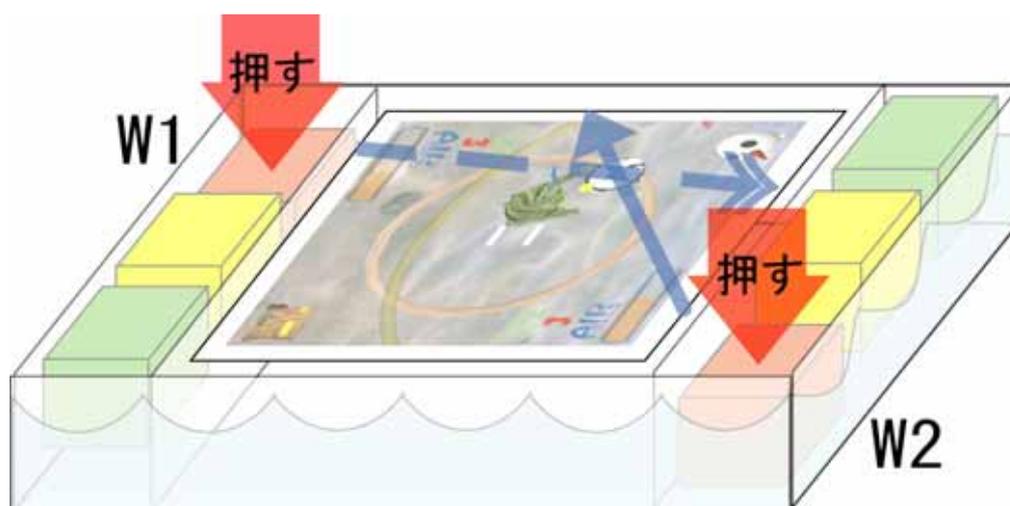


図 13： WGC からの作用力

## 2.7.4 紙力士の干渉と衝突後の挙動

紙力士の干渉判定及び衝突後の挙動は、2.3.5、2.3.6 節と同じアルゴリズムを用いているが、水中での挙動を表現するため、水抵抗を考慮した挙動計算を行っている。

具体的には、算出した衝突後の速度に対して、1.0 未満の値を乗算して実際よりも遅い速度にし、水の抵抗がある様な紙力士の動きを表現している。

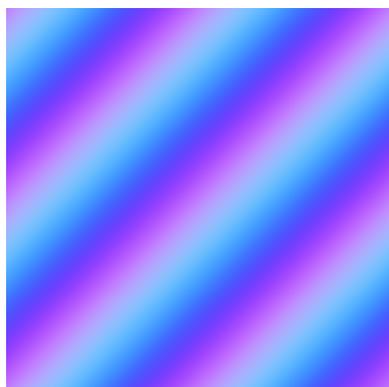
## 2.7.5 勝敗判定

勝敗の判定条件は以下の2つの条件のいずれかを満たした時に勝敗を決めるものとする。

- (J1) 土俵を割る：水中での紙相撲の土俵として，円筒形の空間フィールドを定義し，プレイヤーの紙力士が円筒の外側に位置した場合，そのプレイヤーを負けとする．ここで，高さには制限はないものとする．
- (J2) 時間切れ：勝敗が決せず，一定時間（ここでは，45 秒に設定）経った場合，引き分けとする．

## 2.7.6 映像生成と表示方法

図 15 で示すように，紙相撲の映像は，DirectX Graphics を用いてリアルタイムで描画し，水槽の斜め上からスクリーンに投影させた．特殊効果として，水面の揺れの特殊効果を画像を用いて演出し，WGC からの作用力を波紋で表現した．図 14 (a) はテクスチャを貼り付ける位置 (UV 値) のずらし量を埋め込んだ画像であり．これを用いて，図 14 (b) のような対象画像に対して，図 14 (a) のずらし量を加えることで出力画像が歪む．これを連続的に行うことで，水面の揺らめきを表現している．波紋は，プレイヤーが押した WGC を検知し，その WGC の座標を中心とした，ドーナツ状の幾何形状を拡大させながら描画して表現した．



(a) 揺れを表す画像



(b) 対象画像



(c) 揺れを表現された画像

図 14：揺れの特特殊効果

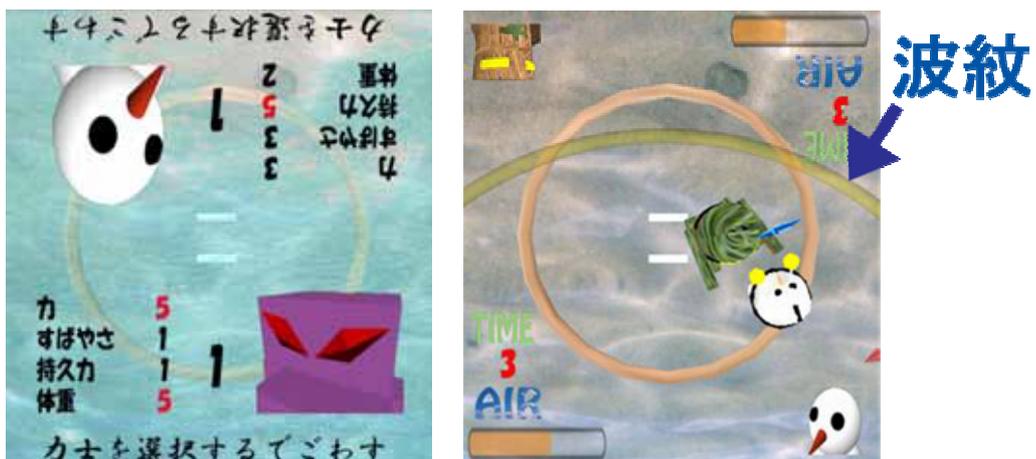


図 15： トントン Ver.2 の出力映像

## 2.8 トントン Ver.2 の実験結果

2004年10月29,30日に開催されたIVRCの岐阜本選において100人以上に体験してもらい,総合評価で3位という高評価を得た.大会での体験の様子を図16に示す.

実験に用いたPCは,DELL製DimensionXPS(Pentium4 3GHz,1Gメモリ)で,グラフィックスカードとして,GeFORCE 6800 Ultra(256MB)を用いた.



図 16： トントン Ver.2 の体験の様子

評価実験において、体験者からは「波で操作できるのが良い」や「水中で紙相撲の発想が面白い」などのコメントが多数寄せられ、新たなインタラクション手法による紙相撲を楽しんでもらうことができた。また本システムは、岐阜県で行われた IVRC 本選大会において、2 日間で計 10 時間の体験に耐え、その耐久性も確認できた。

評価実験の結果を、表 3 にアンケート集計結果（総数 93 件）を示し、表 4 に代表的な意見やコメントを示す。

表 3： トントン Ver.2 に対するアンケート結果

Ver.2 の評価項目	満足 (%)	普通 (%)	悪い (%)
楽しさ	86.0	14.0	0.0
パネルの押し心地	35.5	51.6	12.9
波紋の表現	33.3	59.1	7.5
映像の質	52.7	47.3	0.0
水の使用	74.2	21.5	4.3

表 4： トントン Ver.2 への代表的な意見およびアドバイス

代表的な意見	代表的なアドバイス
<ul style="list-style-type: none"> <li>・ 水中の雰囲気が良い</li> <li>・ 発想が面白い</li> </ul>	<ul style="list-style-type: none"> <li>・ 応答性が悪い</li> <li>・ 操作が難しい</li> </ul>

<ul style="list-style-type: none"> <li>・ アトラクション性</li> <li>・ うまく操作できないもどかしさ</li> </ul>	<ul style="list-style-type: none"> <li>・ 水に少し濡れる</li> <li>・ 波紋の表現の改良</li> </ul>
--	---

評価実験の結果，楽しさという点で高評価を受けた．これは，トント Ver.1 と同じく，単純な操作性によるものだと考えられる．また，媒体に用いた水に関しては，良い印象を得ることができた．これは，波紋による力の伝わりが視覚化されていることや，自分が押した，もしくは相手が押した水の感触を感じることができることなどが評価されたと考えられる．

また，図 17 に示すように，2005 年 7 月 31 日から 8 月 4 日にアメリカのロサンゼルスで開催された SIGGRAPH2005 Emerging Technology においても展示発表を行った．全世界から集まったおよそ 1300 人の人々に対し，水中での紙相撲を楽しんでもらうと共に，日本の伝統的遊びを体験してもらった．また，実装が困難とされている水を使用したアプリケーションを実現することで，赤外線を用いたセンサの応用性・堅牢性が評価された．SIGGRAPH では，5 日間で計 40 時間の体験に耐え，その堅牢性を改めて確認できた．



図 17：SIGGRAPH での体験の様子

## 2.9 トント Ver.2 の考察と課題

以上，実世界では実現困難な水中での紙相撲が楽しめる VR アプリケーションを紹介

介した。このアプリケーションでは、自分や相手の力の伝わりを水の波紋から見ることができ、さらに水を押すという感触を楽しむこともできる。既存の対戦型ゲームにおいてキーパッドを連打する行為と、トント Ver.2 でのインタラクションの差異の一つとして、柔らかな力のフィードバックが挙げられる。すなわち、パネルを用いて水を押し出すという動作とともに、相手の力をパネルからも同時に感じるができる。また、トント Ver.1 では2画面出力で表示していた体験空間を、トント Ver.2 では1つに統合し、仮想の世界と現実の世界とを融合したことで、一体感を演出したと考える。

課題としては、WGC の入力に対するシステムの反応が鈍いことが指摘されており、データの取得および解析法の改良が必要である。また、水の揺れや波紋の表現に関しても改善の余地がある。キャラクタの水中での動きやアニメーションに関しても、細かな動きを与え、質の向上を目指したい。一方、水が予想以上に飛び跳ねることがあったので、一層の防水処理が望まれる。

## 2.10 考察

以上、変位情報を用いた新たなインタラクション手法を考案し、VR アプリケーションへの応用事例を2件紹介した。

トント Ver.1 では、プレイヤーが飛び跳ねる行為を直接紙力士に反映させるというインタラクション法を実現し、単純でかつ直感的な動きを楽しむ環境が提供できた。トント Ver.2 では、力の伝わりを視覚化し、水を媒体とした力の感触のフィードバックなどのインタラクションを実現できた。

自分があたかも紙力士になったつもりで直接操作ができるトント Ver.1 と、紙相撲本来のインタラクションに近い操作法を再現したトント Ver.2 とでは方向性が大きく異なる。すなわち、前者がキャラクタへの直接関与というインタラクションに対し、後者はフィールドへの関与を経由したキャラクタへの間接関与というインタラクションをとっている。

これらのアプリケーションでは同等のセンシングデバイスを用い、変位情報のみを用いてインタラクションを実装している。変位情報という非常に単純なデータのみを用い、その測定法も容易で堅牢な手法であるため、システム全体の信頼度は非常に高

いものであった。また、乱暴な操作に対する耐久性も極めて高いものになった。

両アプリケーションでは、ゲームセンターでの対戦ゲームのように相手が見えないというような対戦形態ではなく、どちらも相手の顔や体の動きを見ながら対戦する形態を実現することができた。これにより、Ver.1 では、親と子供が顔合わせで対戦をし、子供が親に勝って喜び、負けた親がそれを驚き、お互いに楽しむという場面が見受けられた。Ver.2 でも、相手が発生させた波と一緒に、どのように相手が波を起こしたかなどの様子を見ることで楽しむという場面もいくつか見受けられた。これらにより、対面型の対戦の楽しさも改めて証明することができた。

本章で述べた VR アプリケーションでは、多くの人々に楽しんでもらうことができたが、今後は、このインタラクション手法を違う遊びの表現へと適用したい。

本手法の応用例として、物体の変形操作やジェスチャ認識による応用システムも考えられる。例えば、多数の小型測距センサを内蔵したオブジェクトを用意し、これを手で変形した場合の変位量を測定して対象物の幾何形状の変形操作を行う、モデリングシステムへの適用が考えられる。また、手の位置の移動量を同様のデバイスと手法を用いることで、直感的なインタラクションを伴う新たなアプリケーションの実装が可能であると考えられる。

# 第3章 手指認識によるキャラクタアニメーション –バーチャル手使い人形 “ トムテ ” –

本章では、人の手や指の動きによる直感的身体動作によって、リアルタイムでキャラクタを操作して遊ぶことができる VR アプリケーションについて述べる。

## 3.1 はじめに

キャラクタアニメーションにおいて、キャラクタのモーション生成を支援するツールの開発が多数行われている。しかし、これらはノンリニア編集を主たる対象とし、高品質なアニメーションを制作できるが、高い技術スキルや多くの時間を必要とし、気軽に利用できるとは言えない。

本章では、アニメーション制作の支援手法として、手使い人形をテーマとした、直感的なキャラクタアニメーションの制作手法を提案する。提案手法では、操作者の手指の場所や動きを画像認識し、その結果を用いて、任意の CG キャラクタを自由かつ直感的に画面内で操作することで、リアルタイムでのアニメーション生成を可能とした。

## 3.2 背景

コンピュータアニメーションは、映画やビデオゲームなどの分野において欠かせない技術である。その中でもキャラクタアニメーションはコンテンツ制作において重要な要素のひとつであり、様々な技術開発が現在でも進められている。例えば、キャラ

クタアニメーションの研究では、モーションキャプチャで取得した人の動きのデータを基に、データベース化したデータから、新たなモーションを出力する手法[11, 12, 16]や、動きを解析しそれを定義付けすることで新たなスタイルのモーションを生成する手法[13, 14, 15, 17, 18]などがある。

アニメーションの制作技法としては、モーションキャプチャによるモーションデータの編集および、再利用技術や、アニメータによるアニメーション生成などが挙げられる[23]。一般に、これらのアニメーション制作手法には莫大なコストがかかり、かつ、高度な経験やノウハウなどが必要とされるため、アニメーション制作者の負担は大きい。一方で、最近ではアニメーション制作の支援手法の提案が多くなされている。例えば、アニメーションのモーションデータベース化[12]、また3DCG作成ソフトにアニメーションテンプレート機能でアニメーション生成を補助する機能も存在する[24]。ただし、これらのアニメーション制作支援技術は、主としてノンリニア編集を扱ったものであり、人の直感的な動きを表現できないことや、多大な時間と煩雑な作業が依然として必要とされているのが現状である。

キャラクタアニメーションに関連性の高いものとして、古来より伝わる人形劇がある。人形劇とは、人形遣いが人形を操って演じる劇のことであり、指で操るギニョール、糸で操るマリオネット、文楽のように一体の人形を何人かの手で操るものなどが世界中に数多く存在する。昔より人々は手指を使うことで、自分のキャラクタである人形を自由に操作し、多くの物語を演じさせてきた。

本研究では、手使い人形をテーマとしたアニメーション制作の支援手法として、USBカメラに映る手指の場所や動きを認識し、それを入力としてキャラクタに様々なアニメーションをリアルタイムで生成する手法を提案する。これにより、ユーザは人形劇における演技者のように、任意のタイミングで意図したアニメーションをキャラクタに付加し、物語を表現することが可能になる。なお、本研究で開発したアプリケーションをトムテ<sup>1</sup>と名づける。

---

<sup>1</sup> “トムテ”という名前は、北欧に住む三角帽子をかぶった妖精から名前を取った。トムテは、主にヨーロッパ、シベリア、北アメリカに住んでいるが、密度の差はあれ世界中に住んでいる。手紙やカードの裏に「トムテ、トムテ、トムテ、…」と書いて出せば、表に書いた思いが通じるという言い伝えがある。

### 3.3 関連研究

Pinocchio, a virtual character[19]（以降，*Pinocchio*）は，図 18（a）に示すような，糸あやつり人形をテーマとした VR アプリケーションである。*Pinocchio* は，スクリーンに「不思議の国のアリス」の少女アリスを表示し，上向きに設置されている CCD カメラの上で，十字型コントローラ（クロス）を動かすと，画面の中のアリスがクロスに連動して動き，操り人形のピノキオを操作する。アリスは鏡の前でピノキオを操作しているため，アリスとピノキオの 2 体の動きは鏡にも写り込んでいる。ところがクロスを左右に回転させると，部屋の表示もあわせて回転し，2 体が鏡の世界へと吸い込まれ，反対にピノキオがアリスを操ることになる。すなわち，クロスに付けられたマーカの動きを読み取ることでクロスの回転を検知し，アリスのいる世界と鏡に映る人形の世界を行き来することができる。*Pinocchio* は，キャラクタを操作するという意味では本研究と扱うテーマは同じであるが，*Pinocchio* ではクロスという操作デバイスを必要とする一方で，本研究では手だけで操作するというインターフェースの面で異なる。

the ATI interactive circus[20]（以降，*Circus*）は，図 18（b）に示すように，サーカス団の団員を操作して楽しむことができる VR アプリケーションである。*Circus* で操作できるのは，魔法使いの猿や，ロボット，道化師，炎使い，軽業師など 5 つのキャラクタである。操作者は 1 体のキャラクタを選択し，「Dance Dance Revolution」[2] で用いられているマッドコントローラや，左右の手に装着するゲームパッド，膝や二の腕に装着するスイッチ，大砲型のカメラやグローブデバイスなどの多様なデバイスで操作する。操作に用いるデバイスや可能な演技はキャラクタに依存している。例えば，軽業師ならキャラクタが空中にジャンプしている間に，体験者が全身のスイッチを操作して演技する，猿の魔法使いでは，右手に装着したバーチャル・グローブの動きに連動して，猿が空中に呪文を描く，などである。*Circus* では，身体動作による入力という点で本研究と関連しているが，入力デバイスを操作者の身体に装着するという点が，本研究と異なる。

一方，手指認識の研究例として，ステレオカメラによる指先と指の関節を検知する手法[21] や，円を画像内で走査することにより指先を検知する手法[22] などがある。これらの手法は，実時間で解析が可能であるという点で本研究と関連しているが，使

用しているデバイスがステレオカメラであることや，画像処理の手法などの面において，本研究の手法とは実装方法が異なる．



( a ) Pinocchio, a virtual character



( b ) the ATI interactive circus

図 18：関連研究の例

## 3.4 システム構成と実装方法

本節は，トムテのシステム構成とその具体的な実装方法について述べる．

### 3.4.1 システム構成

本システムは，図 19 に示すように，USB カメラとコンピュータのみで構成されている．USB カメラから取得した画像を入力とし，取得した画像から指の位置や動きを自動認識し，認識結果を用いてキャラクタをリアルタイムで描画する．

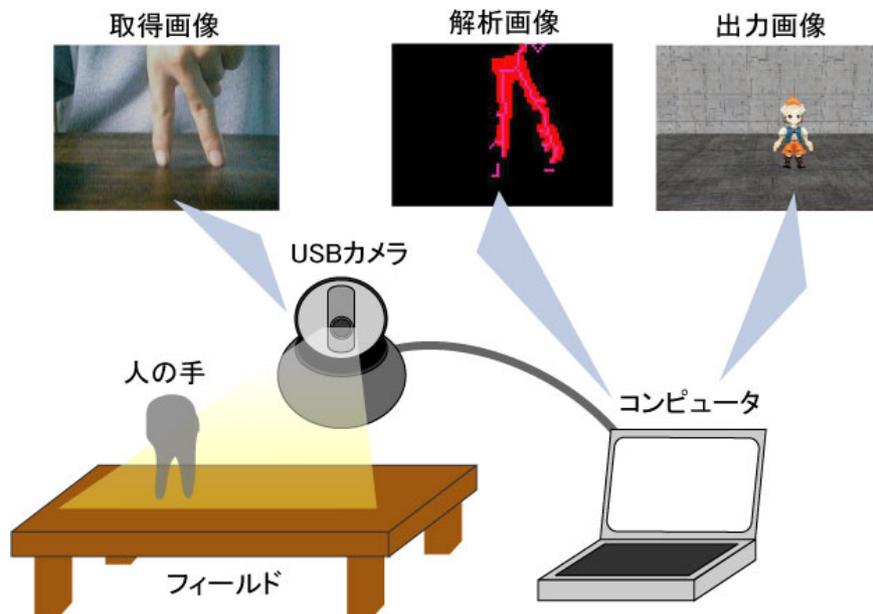


図 19：トムテのシステム構成

システム内での処理の流れを以下に示す。

- (P1) USB カメラから画像を取得する。
- (P2) 取得した画像から肌色領域を抽出する。
- (P3) (P2)の抽出画像からノイズを除去する。
- (P4) (P3)の処理画像に、ダウン・アップサンプリング，膨張処理，ラベリング処理を施して，計算の最適化を行う。
- (P5) (P4)の処理画像から指の特徴を抽出する。
- (P6) (P5)の結果から指の特徴量を計算する。
- (P7) (P6)の結果を用いて出力画像を生成する。

### 3.4.2 手認識手法

操作者の手の認識には、画像処理による色解析手法の一つである肌色解析[9]を用いた。肌色解析には、入力された画像のデータを RGB 表色系から CIE  $L^*a^*b$  表色系へ変換し、画像内の肌色と判定された画素を抽出する手法を用いる。CIE  $L^*a^*b$  表色系は、類似した色の抽出がしやすいという特徴があるため、肌色認識の表色系として扱いやすい。しかし、色空間の変換アルゴリズムが複雑であるため、CPU による実時

間処理は困難であると考えられる．実時間処理を考慮し，負荷の軽い HSV 表色系を用いることも考えられたが，室内環境での蛍光灯の明かりも肌色と判断してしまう恐れがあった．そこで，ViewFrame[9]で提案されている，GPU を用いた表色系の変換処理を用い，実時間における肌色解析を実現した．具体的な手法としては，ピクセルシェーダによる色解析によって，RGB 表色系から CIE  $L^*a^*b$  表色系に変換した後，式(7)の範囲内の値を肌色と判定する．そして，肌色と判定された画素を白画素((R,G,B)=(255,255,255))とし，それ以外を全て黒画素((R,G,B)=(0,0,0))とした．

$$\begin{aligned} 10 < L < 90 \\ 0 < a < 25 \\ 0 < b \\ -5 < a - b < 25 \end{aligned} \quad (7)$$

ただし，L：明度指数，a, b：知覚色度である．

GPU による肌色解析処理の様子を図 18 に示す．図 20 (a) は USB カメラでキャプチャした画像であり，図 18 (b) は，肌色と判定された画素を白，それ以外は黒として 2 値化した画像（肌色画像）である．



(a) 入力画像



(b) 肌色画像

図 20：肌色領域の抽出例

### 3.4.3 指認識手法

3.4.2 で述べた肌色画像のヒストグラムの特徴を解析することで、指の特徴量を抽出する。

図 21 (a) に示すように、3.4.2 で取得した肌色画像から画像の横軸 (x 座標軸) の画素 1 列ごとの肌色の数をヒストグラムとして生成する。ここで、図 21(a) の例は、図 20 (b) に対するものである。

まず、生成したヒストグラムから、式 (8) で表されるように、肌色が確認された列の数  $n$  を求める。つづいて、式 (9) に示すように、画像内の肌色画素  $w(x)$  の総数を  $n$  で割ることで、x 座標値における肌色の平均  $m$  が計算される。例えば、図 21 (a) の場合では、ヒストグラムにある肌色の総数が 619 画素であり、肌色が確認された x 座標値が 22 個あったので、各 x 座標値における肌色の平均が 28 画素となった。図 21 (b) は 28 画素以上の肌色があった x 座標値と肌色の数を全て列挙したものである。

$$n = \sum_x u(x) \quad u(x) = \begin{cases} 1: w(x) > 0 \\ 0: otherwise \end{cases} \quad (8)$$

$$m = \frac{\sum_x w(x)}{n} \quad (9)$$

ただし、 $w$  : 画像内の肌色画素、 $x$  : x 座標値である。

図 21 (b) に示すように、指の特徴抽出では、式 (9) で求めた各 x 座標値における肌色の平均  $m$  (図 21 (b) 青色ライン) 以上ある x 座標値に注目する。図 21 (b) を見て分かるように、x 座標の値が 2 個以上連続しているものがある。この開始および終了の x 座標値をそれぞれ  $S$ ,  $E$  とする。すなわち、 $S$  は  $w(x) > m$ ,  $w(x-1) < m$  を満たす x 座標値、 $E$  は  $w(x) > m$ ,  $w(x+1) < m$  を満たす x 座標値である。式 (10) に示すように、連続する座標値が一定の数以上ある部分を 1 本の指  $I$  と仮定する。指の中心軸  $C$  は、式 (11) で表すように、連続している x 座標値の内の重心の x 座標値とする。例えば、図 21 (b) の場合に、1 つ目の指は 39 から 44 まで x 座標値が続いていることから中心値は 41 (もしくは 42)、2 つ目の指は 52 から 56 まで x 座標値が連続していることから中心値は 54 となる (いずれも図 21 (b) の赤円に示す)。ここで、肌色解

析時に取得する画像は 720x480 であり，図 21 の x 座標値が取得画像の解像度に比べて極端に低いが，これは 3.4.4 で後述するダウンサンプリング処理のためである．

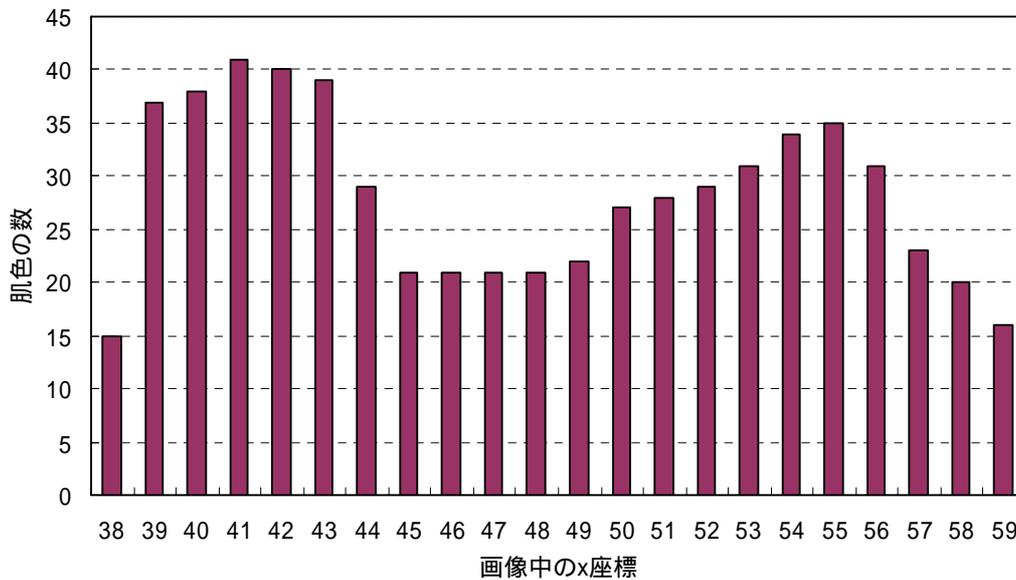
次に，指が鉛直方向に向かって立てられることを前提とし，指の先端の座標を求める．式 (12) に示すように，各指の中心値  $C_i$  の画像内の縦軸 (y 座標) 上で最も下に位置する肌色画素  $w(C_i, y_{max})$  を探し，その xy 座標を指の先端  $Fi(x, y)$  とした．

$$i = \sum I \quad I = \begin{cases} 1: S - E > H \\ 0: otherwise \end{cases} \quad (10)$$

$$C_i = \frac{S_i + E_i}{2} \quad (11)$$

$$F_i(x, y) = F_i(C_i, w(C_i, y_{max})) \quad (12)$$

ただし，H：定数である．



(a) 肌色ヒストグラム (抜粋)



(b) 平均値以上の肌色ヒストグラム (抜粋)

図 21 : 図 20 の画像に対する肌色ヒストグラム

### 3.4.4 最適化

3.4.2 や 3.4.3 で述べた処理は計算量が多いため、スムーズな動画生成が困難である。また肌色解析において、ノイズが混じることがあり、正確な解析をできない場合がある。したがって、処理の軽減やノイズ除去を行うために、以下に述べるような最適化処理を施した。

#### ( ) ダウン・アップサンプリング

解析対象の画像をダウンサンプリングし、指の特徴抽出処理後の画像をアップサンプリングすることで、処理対象の画素数を削減して CPU への負荷軽減を図った。

ダウンサンプリングでは、図 22 (a)(c) に示すように、肌色解析した画像において、各注目画素と注目画素の右・下・右下の 3 近傍の画素に注目し、肌色を 1、そうでないものを 0 と 2 値化した場合に、4 画素とも肌色の場合は肌色 (つまり 1)、それ以外の場合は全て肌色ではない (つまり 0) として、全体画像を 1/4 に縮小した。アップサンプリングは、図 22 (b)(d) に示すように、ダウンサンプリングの定義を反転し、注目画素が肌色 (つまり 1) の場合は、拡大後の 4 画素すべてを 1 とし、それ

以外の場合は全て 0 とした .

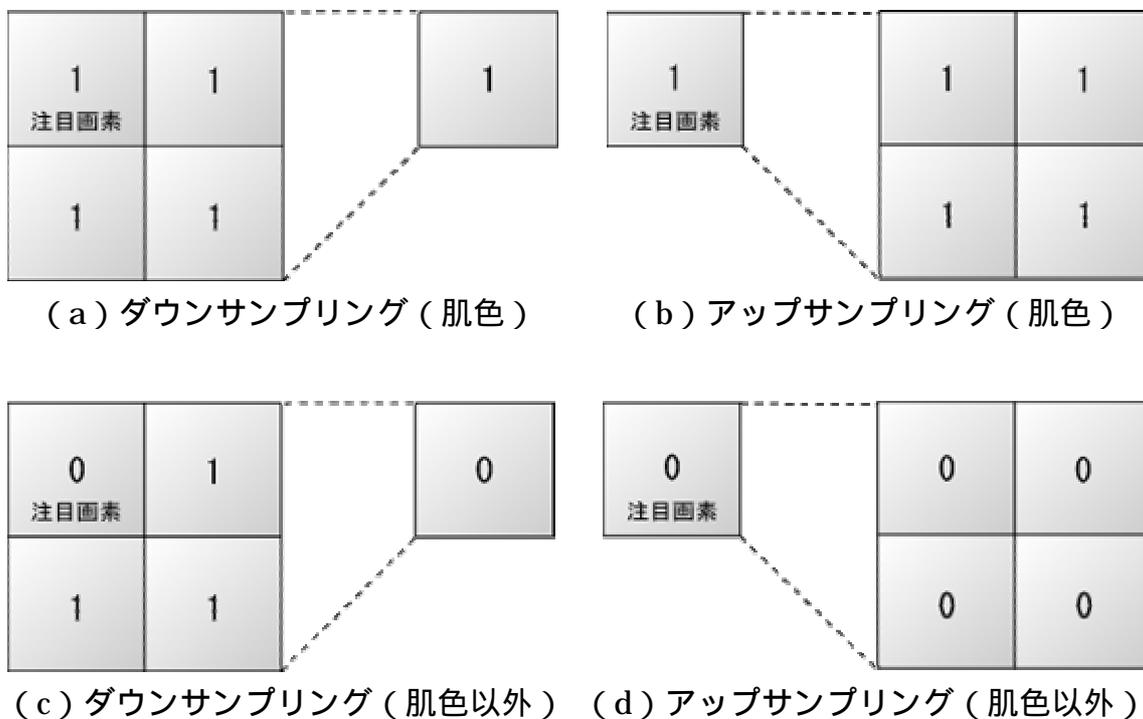
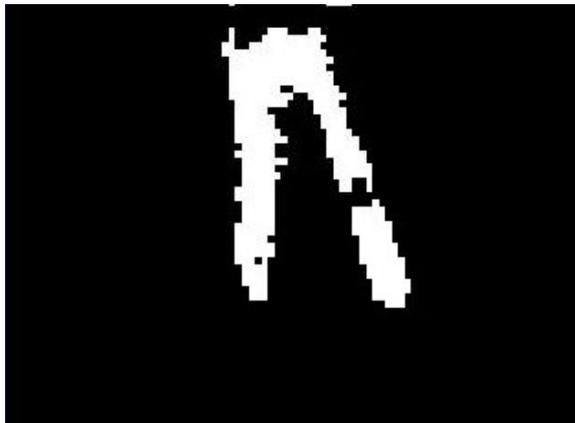


図 22 : ダウン・アップサンプリング法

### ( ) 収縮・膨張処理

図 23 (a) に示すように , 解析後の画像は指の形が細くまた途中で切れてしまう場合があるため , 指の特徴を正確に抽出できない可能性がある . そこで , 図 23 (b) に示すように , 取得した画像を膨張処理し , 指の切れをなくすことで特徴を抽出しやすくした . 具体的には , 画像の左上から走査を行い , 肌色画素を注目画素とした場合に , 注目画素の左上・左・上の画素を肌色画素とするという処理を行う .



(a) 元画像



(b) 膨張処理後の画像

図 23：膨張処理の例

## ( )ラベリング処理

3.4.2 の肌色解析の結果から，手指の肌色の他に，肌色の誤認識などによるノイズが混在するため，ラベリング処理によるノイズ除去を行った．ラベリング処理とは，画像内の同じ連結成分を構成する画素に同じ番号（ラベル）を付け，異なる連結成分を構成する画素に異なる番号（ラベル）を付ける処理のことである．本研究では，画像の左上から走査（ラスタスキャン）を行い，肌色画素を注目画素とした時に以下の処理を行う．

ラベルの付いてない画素を注目画素とする．

図 24 (a) に示すように，注目画素の上の画素が，ラベルを持つ時（図では 3 番ラベルとする），上の画素のラベルを注目画素に付ける．

図 24 (b) に示すように，注目画素の上の画素が肌色画素でなく（肌色画素ではないので 0 とする），左の画素がラベルを持つ時，そのラベルを注目画素に付ける．

図 24 (c) に示すように，注目画素の上も左も肌色画素でない時，新しいラベルを注目画素に付ける．例えば，前回までに付加したラベルが 3 番だった場合，4 番ラベルを付加する．

ラベルのついていない肌色画素がなくなるまで， から までの処理を繰り返す．

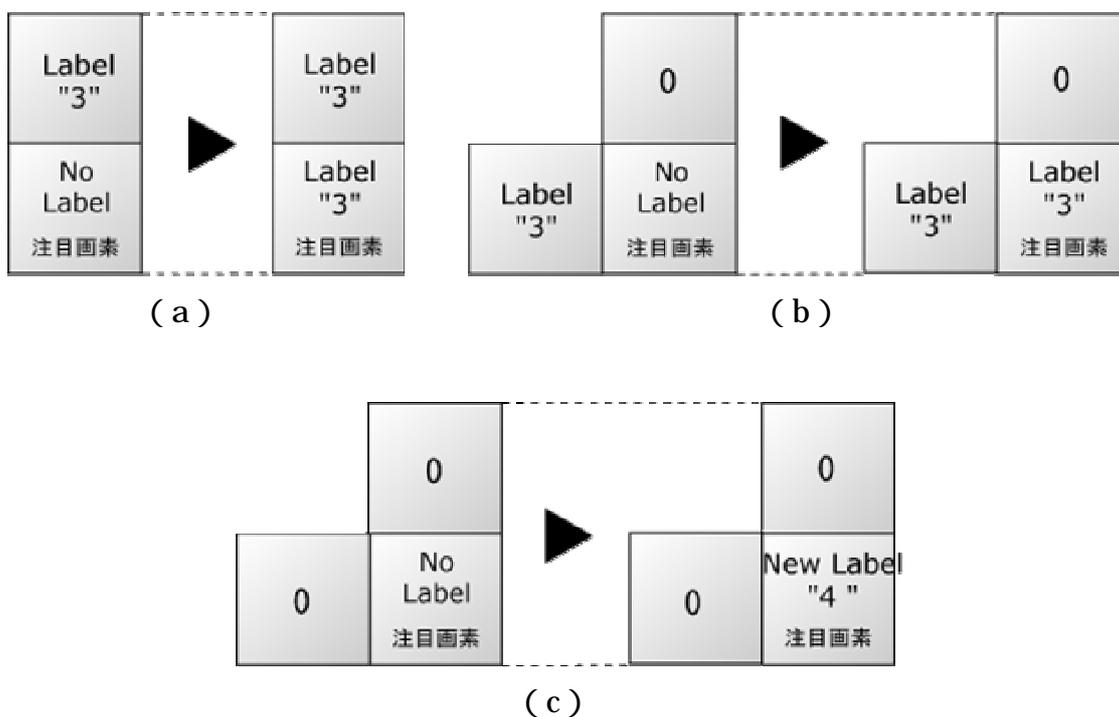


図 24 : ラベリング処理 ( 1 )

しかし、上記の処理だけでは、隣接しながらもラベル ( 番号 ) が違っている場合があり、正確なラベリングが完成しないので、以下の処理も加えることでラベリングを完成させた。

ラスタスキャンを行い、図 25 に示すように、注目画素の上と左の画素のラベルが違ふところを探す ( 図では 3 番ラベル、もう一方を 5 番ラベルとする )。ここで、小さいラベルの方を「小ラベル」( 図では 3 番ラベル )、大きい方のラベルを「大ラベル」( 図では 5 番ラベル ) と名づけ、大ラベルを小ラベルに置き換える。

注目画素も、小ラベルに置き換える。

全画素に対して、大ラベルと同じラベルを見つけた時、それを小ラベルに全て置き換える。この処理は、 の処理が行われるごとに実行する。

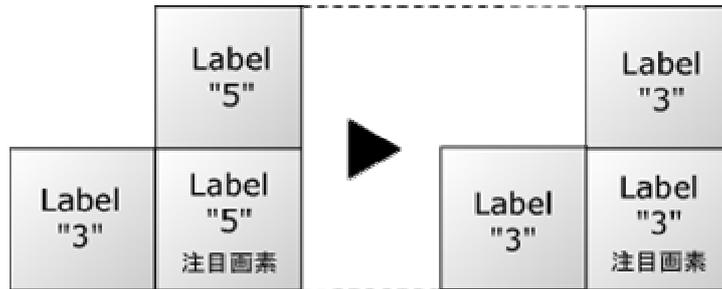


図 25 : ラベリング処理 ( 2 )

以上の処理を終えた後に , ラベリングされた画素でラベリング数が少ないものをノイズとして除去した . 図 26 に示した例では , 図 26 ( a ) の下方にある分離した肌色領域が図 26 ( b ) ではノイズとして除去され , 手の肌色領域だけが認識結果として残されていることが分かる .



( a ) ノイズ除去前



( b ) ノイズ除去後

図 26 : ノイズ除去の例

### 3.4.5 キャラクタ描画

本研究では , キャラクタを手指で操作ができるということから , 操作者の手の指の先端にキャラクタの足元が描画されるようにした . 現実世界と仮想世界が整合するように , キャラクタの描画位置を , 認識した指の位置を用いて以下のように計算する .

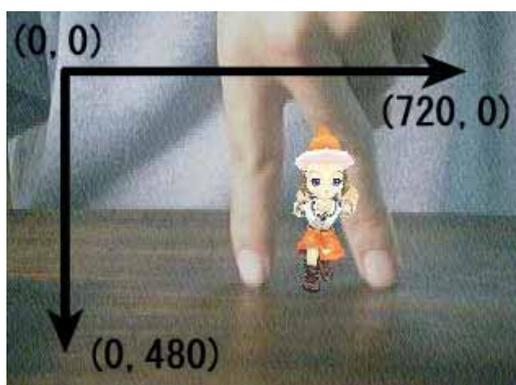
図 27 で示すように , 特徴抽出された操作者の 2 本の指先の座標の中間の座標値  $Ch$

を式 (13) で求め,  $Ch$  の座標位置にキャラクタの足が描画されるように, キャリブレーションを行った後に, 実際のキャラクタ描画を行う. 特徴抽出の結果, 3本の指が検知された場合, 特徴量が多い2つ, すなわちヒストグラムの高さが高い2つを2本の指として,  $Ch$  の座標値位置を求める.

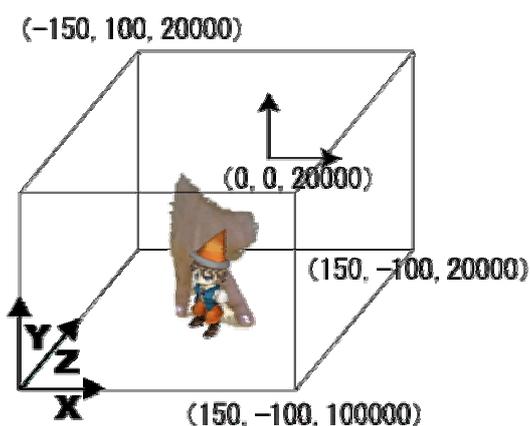
$$Ch(x, y) = \frac{F_1(x, y) + F_2(x, y)}{2} \quad (13)$$

ここで, キャラクタ描画には DirectX Graphics を用いているが, DirectX の座標系と画像処理で用いた座標系が異なるため, 式 (14) に示すような座標変換を行う. 3次元変換を行う時のキャラクタの奥行き, つまり  $z$  軸の値は, 認識した指の肌色の量に応じて変化させ, 肌色が多ければキャラクタを手前に, 少なければキャラクタを奥に描画するように計算した.

$$P \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 0.416 & 0 & 0 & -149 \\ 0 & -0.414 & 0 & 98 \\ 0 & 0 & -0.0035 & -51.76 \end{pmatrix} * \begin{pmatrix} Ch(x) \\ Ch(y) \\ \sum_x w(x) \\ 1 \end{pmatrix} \quad (14)$$



(a) 画像処理の座標系



(b) 描画処理の座標系

図 27: 処理の座標系の違い

## 3.4.6 キャラクタアニメーション

認識した指の動きから，手指の様々なジェスチャを認識することで，手使い人形のキャラクタアニメーションを表現した．

### ( ) 立つアニメーション

図 28 (a) に示すように，握った状態の手を映すと，立っているキャラクタが映された手と同じ位置に表示される．これは，色がある一定の量以上認識され，かつ指の数  $i$  を 2 本認識することができなかった場合に，肌色の中心に立っているキャラクタを描画させるようにしているとする．

### ( ) 歩きアニメーション

図 28 (b) に示すように，指を 2 つ立てている状態では，立っていたキャラクタが歩き始める．これは，色がある一定以上認識され，また指を 2 つ認識することができた場合に，2 つの指の間の座標に歩いているキャラクタのアニメーションを描画する．

### ( ) 回転アニメーション

図 29 (a)(b) に示すように，指を 2 つ立てている状態で片方の指を折り曲げる動作を映すと，折り曲げた指の方向に向かって少しずつキャラクタが回転する．これは，2 つ立てている指の内，片方の指の長さが他方の半分以下の場合に，キャラクタが短いほうの指の方向にキャラクタを回転させる．

### ( ) ジャンプアニメーション

図 30 に示すように，指を 2 つ立てている状態で両方の指を折り曲げる動作を映すと，キャラクタがしばらく飛び跳ねる．これは，2 つ立てている両方の指の先端の位置が，前フレームおよび次フレームと比較して，一定の速さで上に上がった場合に，ジャンプをするキャラクタを描画させる．



(a)“ 停止 ”



(b)“ 歩き ”

図 28 : キャラクタアニメーション (1)

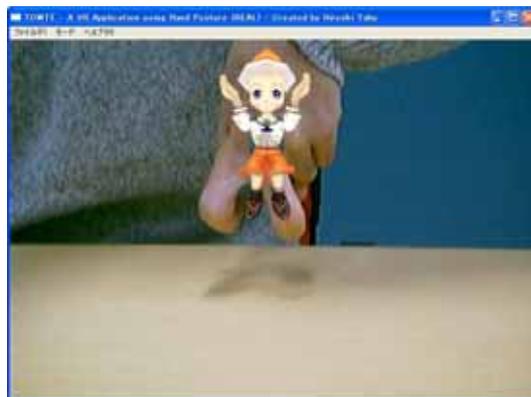


(a)“ 右回転 ”



(b)“ 左回転 ”

図 29 : キャラクタアニメーション (2)

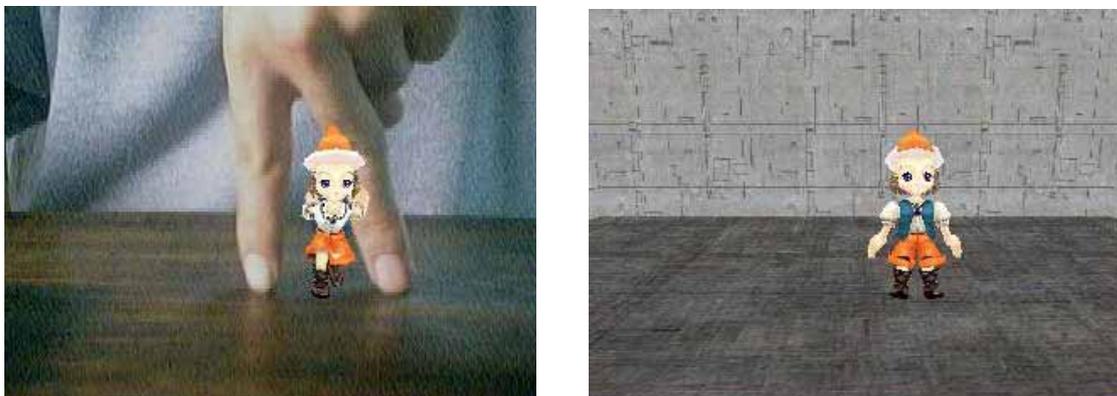


“ ジャンプ ”

図 30 : キャラクタアニメーション (3)

### 3.4.7 映像生成と表示法

手使い人形の映像は，図 31 に示すように，DirectX Graphics を用いてリアルタイムで描画し，表示させた．図 31 の左の図は，現実世界と手使い人形を合成して表示させ，実際の手の動きと同時にキャラクタアニメーションを確認することができる．図 31 の右の図では，CG の世界の中で手使い人形のキャラクタアニメーションのみを表示させた．



(a) 現実世界でのキャラクタ “トムテ” (b) 仮想世界でのキャラクタ “トムテ”

図 31：トムテの出力画像の例

## 3.5 実験結果

操作画面は，図 32 に示すような 4 画面構成とした．左上には USB カメラがキャプチャした映像を，右上には画像処理後の画像を表示し，左下には仮想世界とキャラクタを描画，右下ではキャプチャした映像とキャラクタを描画している．体験時は，モード切り替えを行い，図 32 の左下の仮想現実画面か右下の現実世界画面をフルスクリーンにし，体験者に手使い人形を楽しんでもらえるようにした．

図 33 に示すように，机の上にディスプレイと USB カメラを置き，USB カメラの前に体験スペースを設置した．実験に用いた PC は DELL 製 INSPIRON8600 (Pentium M 1.3GHz, 1G メモリ) で，グラフィックスカードに GeForce FX Go5650

(128MB)を用いた。

実験の結果，体験者の入力に合わせて，約 10fps の描画速度を保つことができた。



図 32：トムテの操作画面の例



図 33：トムテの体験の様子

### 3.6 考察

トムテでは，操作者の手指の動作に応じて，CG キャラクタの動きがリアルタイムで反映され，現実の手使い人形劇と同じような感覚で遊ぶことができ，新たなインタ

ラクションを実現することができた。

現実の手使い人形では、実際の手の動きがそのまま人形の動きとなるため、人形での複雑な動きは表現するのは大変困難とされていた。トムテでは、そのような複雑な動きをライブラリ化して登録しておき、簡単な手指の動きを認識させることで、いつでもキャラクタのしてほしい動きを引き出すことが可能となった。

これにより、現実の手使い人形と同じように、実際の手を動かしてキャラクタの動きを表現することができた。トムテは実時間で手指の動きを認識し、キャラクタの動きを表現するシステムであり、その直感的な入力方法が、現実の手使い人形と似ていることから、手使い人形を手に装着して動きを表現するのと、ほとんど違いを感じさせることがなくなり、どのようなユーザでもほとんど説明をすることなく楽しんでもらうことが可能となった。

また、操作者は手に何も装着せずに体験できることから、自由に手指を動かしてキャラクタアニメーションを表現することができる。さらに、コンピュータと USB カメラだけのシステム構成であるため、特別な環境やデバイスを必要とすることなく、簡単に使用することができる。USB カメラに映る手を認識するだけということから、消耗品もなく、堅牢性にも長けている VR アプリケーションとなった。

課題としては、まず映像出力速度の向上、そして手指認識の堅牢性を高めることが挙げられる。GPU による肌色認識や画像処理の最適化処理により、CPU 処理への負担が大幅に軽減されているものの、10fps での映像出力速度はまだ十分ではないと感じた。肌色認識処理だけでなく、特徴抽出の一部の処理も GPU に負担させることや、特徴抽出の精度を高める手法としてハフ変換などの画像処理を用いるなど、映像出力速度の改善の余地が残されていると考える。また、提案手法では、複数の手や指先の正確な認識が不可能であることから、複数キャラクタのアニメーションを表現することができないので、指の動きを認識する手法に関してもラベリングなどの手法を向上させることによる、複数でのキャラクタのアニメーション表現の実現を目指したい。

## 第4章 結論

本研究では、直感的な身体動作を入力として、飛び跳ねる行為や手の押下す行為で力の伝達を伝える VR アプリケーションとして“トントン”，手指を動かす行為でキャラクターを操作できる VR アプリケーションとして“トムテ”，という新たなインタラクション手法を用いた作品を2つ提案した。これらの提案はともに、直感的に物を操作することや仮想世界と現実世界との高い親和性をを感ずることができ、また老若男女誰でも簡単に遊ぶことができることを目的とした VR アプリケーションである。

“トントン”では、手の押下という身体動作を入力とした手法を提案し、昔の日本の伝統的遊びである「紙相撲」をテーマに、バーチャル紙相撲トントンを実装した。トントン Ver.1 では、対戦形式を用い、互いにジャンピングパッドの上を飛び跳ねることで仮想世界に力を伝達し、バーチャルで紙相撲を体験できる VR アプリケーションを実現した。また、フィードバックなどの問題を改良するため、トントン Ver.2 では、対戦形式を用い、互いに水槽に浮くパネルを押下することで、水を押下する感触や相手が押下した力を楽しみながら、現実では実現不可能な水中での紙相撲を体験できる VR アプリケーションを実現した。本手法により、両アプリケーションが測距センサーを用いていることから、堅牢性が高く、応用性もあることを証明することができた。

“トムテ”では、操作者の手の位置・形・動きという身体動作を入力とした手法を提案し、キャラクターアニメーションとして古来より親しまれていた「手使い人形」をテーマに、バーチャル手使い人形を実装した。トムテでは、USB カメラとディスプレイだけを用いて操作者の手の様々な動きを認識し、その動きに合わせてキャラクターをリアルタイムで操作することで、手使い人形を体験できる。本手法により、デバイスやマーカをつけることなく、コンピュータの操作が可能となったことから、ジェスチャ認識などでの応用性があることを証明することができた。

トントンでは、今はなくなりつつある日本の伝統的な遊びであった紙相撲の面白さを、現代の子供たちに新たな形で伝えることができないかということをもとにモチベ

ーションとした。そして、紙相撲の“叩く”という身体動作を、“飛び跳ねる”“水を押下”といった身体動作に置き換え、リアルタイムで映像が変化するだけでなく、波の行き来により力の伝達も実現し、紙相撲本来の面白さに加えた新たな面白さを実現した。

トムテでは、身体動作による操作が可能となるインターフェースのものが増え続ける中で、手の動きによってキャラクタの豊かな動きを表現し、物語を伝えるという人形劇の面白さの実現を目指した。トムテは、プレイヤーが人形はもちろんのこと、デバイスやマーカ類を一切付けずに、“手指の動き”を画像認識する。そして、現実でも難しい動きを表現し、望みの動きを直感的に演じることができる面白さを提供した。

このように、直感的身体動作に応じて楽しめるシステムを提供できたことから、次世代ユーザインターフェースである PUI (Perceptual User Interface) としての新たな可能性を示すことができた。しかし、ユーザの入力に対するコンピュータからのフィードバックがまだ十分でないことから、ユーザをさらに楽ませるシステムの開発余地が十分にあると考える。

## 付録 . 本研究に関する研究発表

- [1] 藪 博史 鎌田 洋輔 高橋 誠史 河原塚 有希彦 宮田 一乗, 変位情報を用いた新たなインタラクション手法の提案～VR アプリケーションへの応用例～, インタラクション 2005 論文集, 93-94, 2005/02/28.
- [2] 藪 博史, 鎌田 洋輔, 高橋 誠史, 河原塚 有希彦, 宮田 一乗. 変位情報を用いた VR アプリケーションの実装 -バーチャル紙相撲 “トントン” An Implementation of Virtual Reality Application using Displacement Data -Virtual Paper-Sumo "TonTon"-, 芸術科学会論文誌 Vol4. No.2, 第 20 回 NICOGRAPH 投稿論文特集, pp 36-46.
- [3] 藪 博史, 鎌田 洋輔, 高橋 誠史, 河原塚 有希彦, 宮田 一乗, ジャンピングインタラクションを用いた VR アプリケーション, NICOGRAPH2004 秋季大会, 101-106, 2004/11/05.
- [4] 藪 博史, 鎌田 洋輔, 高橋 誠史, 河原塚 有希彦, 宮田 一乗. トントン～変位情報を用いた VR アプリケーション～ Ton2 -A VR Application With Novel Interaction Method Using Displacement Data-, インタラクティブ東京, 2005/08/25-26.
- [5] Hiroshi Yabu, Yousuke Kamada, Yukihiro Kawarazuka, Kazunori Miyata, Masafumi Takahashi, Ton2 -A VR Application with Novel Interaction Method using Displacement Data-, ACM SIGGRAPH 2005, Emerging technologies.

# 謝辞

本研究を通して、お世話になった多くの方々にこの場を借りて感謝の気持ちを申し上げます。主指導教官である宮田一乗教授には、大学院に入学する前から多くのことについてご助言いただき、また入学してからの2年間の間も、数々のご指導を下さったり、研究環境を整えて下さったりなど、自分のやりたいと思うことを常に最大限サポートしていただきました。おかげで、研究発表などにも多く参加させていただき、業界に関連した自分以外の作品や作品を制作した人々と触れ合うことで多くの刺激を受ける貴重な体験をすることができました。

また、宮田研究室のメンバーには、本研究の作品製作をするにあたり貴重なアドバイスやサポートをしていただき、大変感謝しています。この2年間研究室のメンバーと、共に笑い、遊び、励ましあい、共に作業できたことを幸せに感じます。

支えてくださった皆さま、改めてこの場を借りて感謝を申し上げます。

## 参考文献

- [1] <http://www.sega-am2.co.jp/vf.net/index.html>
- [2] <http://www.konami.co.jp/am/ddr/>
- [3] <http://www.namco.co.jp/donderpage/index.php>
- [4] <http://www.jamodrum.net/>
- [5] T. Blaine , T. Perkis , "Jam-O-Drum , A Study in Interaction Design , " Proceedings of the ACM DIS 2000 Conference
- [6] R.Parent , “Computer Animation: Algorithms and Techniques ,” Morgan Kaufmann Pub , 2001
- [7] E. Lengyel , “Mathematics for 3D Game Programming and Computer Graphics , ” Charles River Media , 2003
- [8] 藪・鎌田・高橋・河原塚・宮田 , “ジャンピングインタラクシオンを用いた VR アプリケーション” , NICOGRAPH2004 秋季大会 , セッション VR & Visualization , pp. 101-106
- [9] 河原塚・高橋・宮田 , “ViewFrame – 画像処理による位置検出法を用いた「借景」” , 情処ヒューマンインタフェース研究会 , 2003-HI-106(7) , pp.45-51
- [10] Mark Deloura , “Game Programming Gems , ” Charles River Media , 2000 , pp.390-402
- [11] 角文雄 , 中嶋正之:“動力学モデルによるキャラクタアニメーション用動作生成システム(MODAN)の構築”,芸術科学論文誌
- [12] 角文雄 , 中嶋正之:“動きの抽出と適用技術を利用した既存アニメーション再利用化のための 二次元動画データベースシステム”
- [13] James W.Davis, Hui Gao:"Gender Recognition from Walking Movements using Adaptive Three-Mode PCA"
- [14] M.Unuma, K.Anjyo and R.Takeuchi:"Fourier Principles for Emotion-based Human Figure Animation"

- [15] Nikolaus F. Troje: "Decomposing biological motion: A framework for analysis and synthesis of human gait patterns"
- [16] Matthew Stone, Doug DeCarlo, Insuk Oh, Christian Rodriguez, Adrian Stere, Alyssa Lees, Chris Bregler: "Speaking with Hands: Creating Animated Conversational Characters from Recordings of Human Performance", SIGGRAPH 2004.
- [17] Eugene Hsu, Kari Pulli, Jovan Povic: "Style Translation for Human Motion". SIGGRAPH2005.
- [18] Nikolaus F. Troje: "Decomposing biological motion: A framework for analysis and synthesis of human gait patterns".
- [19] <http://www.rbbtoday.com/column/gameint/20041101/page3.html>
- [20] <http://www.rbbtoday.com/column/gameint/20051107/page6.html>
- [21] James M. Rehg, Takeo Kanade, "Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking," Third European Conf. on Computer Vision, May 1994, pages 35-46.
- [22] Julien Letessier, "Visual Tracking of Bare Fingers for Interactive Surfaces," UIST '04, October 24-27 2004, Volume 6 Issue 2.
- [23] 栗原恒弥+安生健一, "3DCG アニメーション 基礎から最先端まで".
- [24] <http://www.alias.co.jp/products-services/motionbuilder/new/demos.shtml>
- [25] Andrew D. Wilson, "TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction", SIGGRAPH2005 Emerging Technology.
- [26] 岩井大輔, 金谷一郎, 日浦慎作, 井口征士, 佐藤宏介, "ThermoPainter: 熱画像を用いたタブレット型入力装置とそのインタラクティブ描画システム", インタラクティブ東京.
- [27] 藤井博文, "koekaki (コエカキ)", インタラクション 2005.
- [28] <http://pinballpark.at.infoseek.co.jp/>