

平成25年度 卒業論文

論文題目
発表者を重畳表示するプレゼンテーション支援ソフト

指導教員
舟橋 健司 准教授

名古屋工業大学 工学部 電気情報工学科
平成19年度入学 19213021 番

名前 中江 裕介

目次

第1章	はじめに	1
第2章	発表者の重畳表示	3
2.1	プレゼンテーションにおける問題点	3
2.2	発表者の切り出しと表示	3
2.3	発表者のスクリーン内での移動	4
2.4	発表者によるスクリーンへの書き込み	8
第3章	実験	10
第4章	むすび	18
	謝辞	19
	参考文献	20

第1章 はじめに

近年、マイクロソフトの「Kinect」[1]、ソニー株式会社の「PlayStation Move」[2]、任天堂株式会社の「Wiiリモコン」[3]等の個人向けの安価で直感的に操作できる入力装置の登場によりナチュラルユーザーインターフェース（Natural User Interface: NUI）に対する関心が高まってきている。NUIとは「人間の自然な動作である「触れる」「動く」「話す」（究極的には「考える」などの行為でコンピューターを操作できる）」[4] ユーザーインターフェースを指す言葉で、ユーザーは特別な学習をしなくても直感的に操作でき、操作を覚えるまでの学習コストが低いことが特徴である。

そもそもユーザーインターフェースとは、「コンピュータシステムあるいはコンピュータプログラムと人間（ユーザー）との間で情報をやり取りするための方法、操作、表示といった仕組みの総称」[5]であり、コンピュータ技術の進歩とともに様々な種類のユーザーインターフェースが広く普及している。古くはユーザーから紙テープによる入力に対して結果を出力する非対話型のバッチインターフェースに始まり、ユーザーからキーボードによるコマンドライン入力に対して文字のみの出力をするキャラクターユーザーインターフェース（Character User Interface: CUI）を経て、現在ではユーザーからポインティングデバイス（主にマウス）とキーボードによる様々な入力に対してディスプレイ上にグラフィカルな出力をするグラフィカルユーザーインターフェース（Graphical User Interface: GUI）がユーザーインターフェースの主流になっている。

また、GUIが普及した現在ではバッチインターフェースは廃れてしまったが、CUIは未だに利用されている。一般的にCUIは、コンピューターにとっては理解がしやすく、コマンドの組み合わせによる操作の自動化が容易であり、コンピューターを遠隔操作する際にネットワークに大きな負荷がかからない等のメリットがある反面、基本的なコマンドをある程度覚えるための学習コス

トがかかる, 各プログラムにおけるパラメータを必要に応じて調べる必要がある, 図形的なものの操作には向かないまたは不可能であることが多い等のデメリットもある.

対してGUIは, ユーザーにとっては理解がしやすく, 視覚的に直感的な操作ができる, CUIに比べて基本的な操作を覚えるまでの学習コストが低い等のメリットがある反面, 操作の自動化が容易ではない, コンピューターを遠隔操作するには専用クライアントが必要なことが多くネットワークに大きな負荷をかけることが多い等のデメリットもある.

このようにユーザーインターフェースの種類によってそれぞれメリット, デメリットがあり, それぞれに適した場面で利用することによってユーザーは快適にコンピューターを操作することができる.

そこで本研究ではNUIに適した利用場面の1つとして学会発表等で使用される大型ホールに設置されている大型スクリーンを用いたプレゼンテーションでの利用方法を提案する. 大型スクリーンでプレゼンテーションをするとき, 多くの場合, 発表者は聴衆側を向き, 大型スクリーンに映しだされたプレゼンテーション資料に基づいてプレゼンテーションする. このとき, 発表者がプレゼンテーション資料で聴衆に注目してほしい箇所があった場合, 聴衆はどこに注目して良いかわからなくなることが頻繁に起こる.

この問題を解決するために本研究ではバーチャルリアリティ技術の1つであるオーグメンテッドリアリティの考え方に基づいて発表者を重畳表示し, インタラクティブなスクリーンへの書き込みを可能とするプレゼンテーション支援ソフトウェアを提案する. ところで, バーチャルリアリティとは, 「みかけや形は現物そのものではないが, 本質的あるいは効果としては現実であり現物であること」[6]であり, オーグメンテッドリアリティとは, 「現実空間にバーチャルリアリティの情報空間を重畳して, 現実空間を補強すること」[7]である. 聴衆から見て発表者がスクリーンに重畳表示され, 発表者が身振り手振りやインタラクティブな書き込みでプレゼンテーション資料の内容を補完して説明することで, 聴衆の深い理解が期待できる.

第2章では発表者の重畳表示について説明する. 第3章では本研究で作成したソフトウェアの実験と結果について述べる.

第2章 発表者の重畳表示

本章では大型スクリーンを用いたプレゼンテーションにおける問題点と大型スクリーンに映しだされたプレゼンテーション資料上に発表者を切り出して重畳表示する方法について述べる。

2.1 プレゼンテーションにおける問題点

第1章でも述べているがプレゼンテーションに大型スクリーンを用いた場合、発表者がプレゼンテーション資料で聴衆に注目してほしい箇所があったとき、聴衆がその箇所がどこにあるかわからなくことがある。原因としては、プレゼンテーションする際に発表者がプレゼンテーション資料に注目してほしい箇所があったとき、一般的にレーザーポインターやマウスカーソルを用いて注目してほしい箇所を示すが、大型スクリーンの面積に対してレーザーポインターの光点やマウスカーソルはあまりにも小さいため聴衆はそれらを見失うと考えられる。

この問題を解決するために本研究では、発表者を光学カメラと距離カメラで切り出し、プレゼンテーション資料に重畳表示することで、発表者はスクリーン上の好きな位置から自身が示したい箇所を指差し、またインタラクティブに書き込みが可能なソフトウェアの研究、開発をおこなった。

2.2 発表者の切り出しと表示

発表者の切り出しには光学カメラと距離カメラを用いている。研究を進めるうえで本研究で使用するハードウェアは安価で誰でも入手しやすいデバイスが良いと考え筆者はマイクロソフトのKinectを選択した。Microsoft Kinect for Windows SDK ver 1.8 [8]のAPIにより人物の骨格情報をもとにした発表者の切り出しを行った。また、カメラから取得した映像を出力するウィンドウを透過、最前面表示することで発表者がカメラに映る範囲内であればスクリー

ンのどの位置にも自由に移動できるようになっている。移動については次節で詳細を述べる。

2.3 発表者のスクリーン内での移動

本研究では、発表者が聴衆に注目してほしい箇所をより効果的に示すために、発表者自身を切り出し、スクリーン内を自由に移動できるようにしたい。しかし、これを実現する上で以下の問題がある。すなわちスクリーンに映しだされた発表者が自身の意図通りの位置への表示を望むとき、その表示位置を修正するために発表者は、カメラの前を大きく横切ったり、前後に大きく移動しなければならないことがある。そこで本研究では、スクリーンに映しだした発表者が多くの移動を伴わずに、自由にスクリーン内を移動することができるようにするために、発表者の自然な、かつ小さな動作を認識して、それを誇張表現するところで移動を実現する。これにより発表者は多くの移動を伴わずに少ない移動のみで自由にスクリーン内を移動することができるようになる。以下に本研究が提案する発表者の移動の誇張表現に関する動作を示す。ここで Microsoft Kinect for Windows SDK で認識可能な骨格情報を示す(図2.1)。この骨格情報をもとにした発表者の大股移動に関する変数と式を次に示す。

- 発表者の前後移動.
- 発表者の左右移動.
- 発表者の上下移動.

まず、この内、発表者の前後移動、左右移動に誇張表現を適用する条件は、大股移動を採用する。広い場所では、少ない歩数で移動するときは一般的に大股移動するからである。

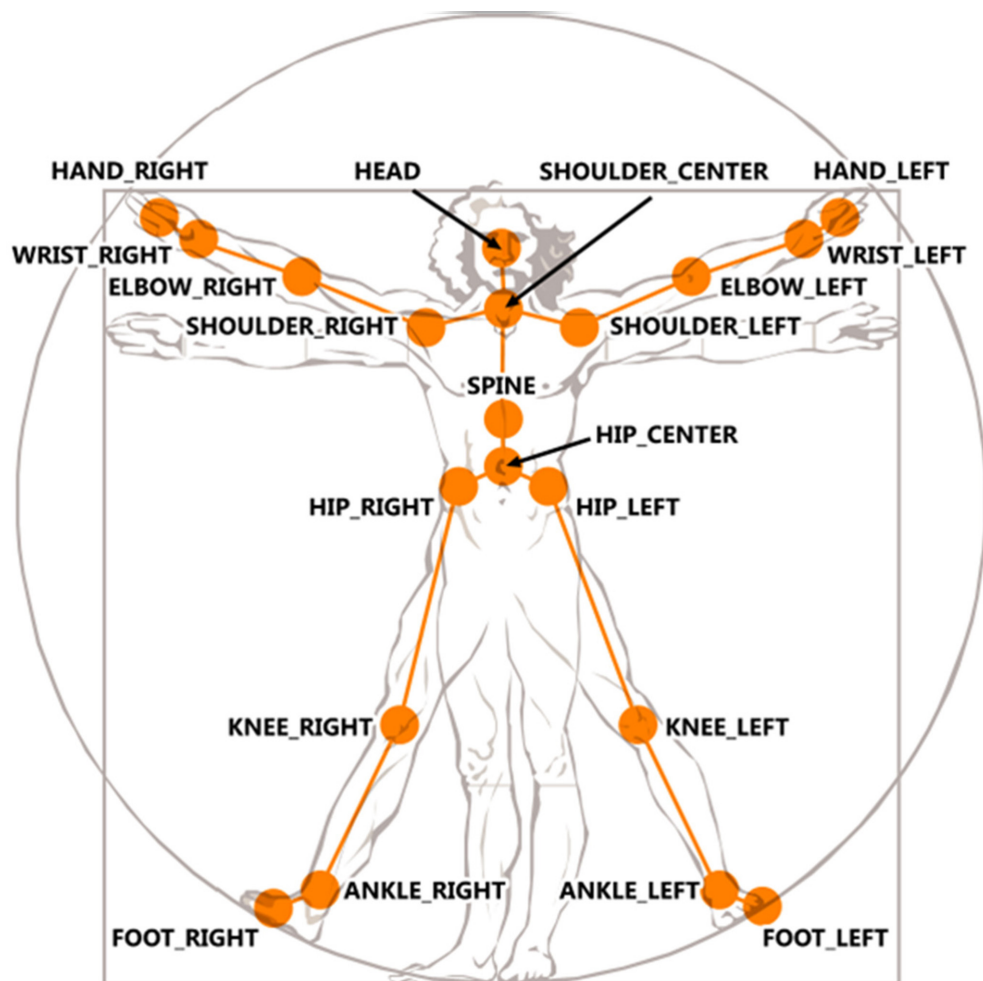


図 2.1: 認識可能な骨格情報 [9]

- 足首座標 (ankle): $\vec{a} = (a_x, a_y, a_z)$
- 膝座標 (knee): $\vec{k} = (k_x, k_y, k_z)$
- 股関節座標 (hip): $\vec{h} = (h_x, h_y, h_z)$
- 膝関節角度 : θ

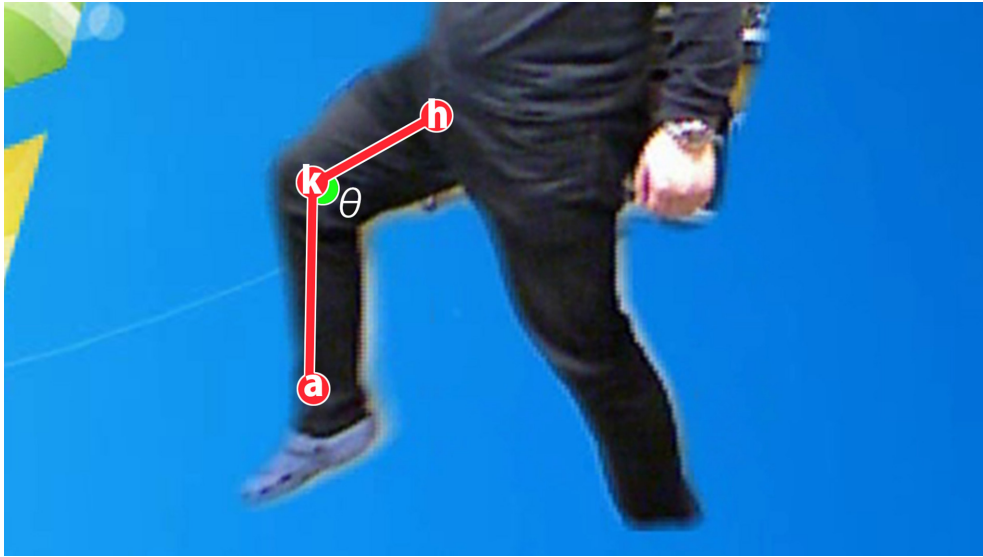


図 2.2: 前後左右移動の判断基準

このとき、それぞれの変数は図 2.2 の X, Y, Z 座標に対応しており、Kinect で取得した膝と股関節、足首の座標をもとに、膝から足首、膝から股関節に向かうベクトルを求め、その内積のアーコサインを計算することで膝関節角度 θ を次の式 2.1 で求めている。

$$\theta = \cos^{-1} \left(\frac{\vec{ka} \cdot \vec{kh}}{|\vec{ka}| |\vec{kh}|} \right) \quad (2.1)$$

$$\vec{ka} \cdot \vec{kh} = (a_x - k_x)(h_x - k_x) + (a_y - k_y)(h_y - k_y) \quad (2.2)$$

$$+ (a_z - k_z)(h_z - k_z) \quad (2.3)$$

$$|\vec{ka}| = \sqrt{(h_x - k_x)^2 + (h_y - k_y)^2 + (h_z - k_z)^2} \quad (2.4)$$

$$|\vec{kh}| = \sqrt{(a_x - k_x)^2 + (a_y - k_y)^2 + (a_z - k_z)^2} \quad (2.5)$$

この膝関節角度 θ が閾値より小さな値になったら大股移動を開始している状態として、その瞬間から発表者の頭部座標の移動距離を計測し、前後移動の場合は映像出力の拡大縮小に関するパラメータを増減させ、左右移動の場合は映像出力のディスプレイ上の配置座標に関するパラメータを横方向増減させている。大股移動を開始している状態から膝関節角度 θ が閾値より大きな値になったら大股移動が終了した状態として、発表者の頭部座標の移動距離の計測を終了する。また、前後移動を誇張表現した際に、発表者が後ろに下がってスクリーン上での表示が小さくなると聴衆から見えなくなってしまう

う. そこで閾値を設けることで移動を制限する.



図 2.3: 上下移動の判断基準

次に, 発表者の上下移動に誇張表現を適用する条件は, 片手を挙げた背伸びや両膝の屈伸を採用する. これは板書や一般的な大きさのスクリーンで発表する際に発表者が上や下の届きにくい箇所を指し示す際の動作を参考にしている. 両膝の屈伸は左右それぞれの膝関節角度が共に閾値より小さな値になったときを屈伸している状態とし, この状態の間は映像出力のディスプレイ上の配置座標に関するパラメータを縦方向に増加させる. また, 片手を挙げた背伸びは肘関節と肩関節, 肩中心座標の3点の座標をもとに式2.1を適用することにより肩関節角度 ϕ を求め, ϕ が閾値より小さく, かつ肘関節座標が肩関節座標より上にあるときを片手を挙げた背伸び状態とし, この状態の間は映像出力のディスプレイ上の配置座標に関するパラメータを縦方向に減少させる.

- 肘関節 (elbow): $\vec{e} = (e_x, e_y, e_z)$
- 肩関節 (shoulder): $\vec{s} = (s_x, s_y, s_z)$
- 肩中心座標 (center): $\vec{c} = (c_x, c_y, c_z)$
- 肩関節角度: ϕ

$$\phi = \cos^{-1} \left(\frac{\vec{s}\vec{e} \cdot \vec{s}\vec{c}}{|\vec{s}\vec{e}| |\vec{s}\vec{c}|} \right) \quad (2.6)$$

$$\vec{s}\vec{e} \cdot \vec{s}\vec{c} = (e_x - s_x)(c_x - s_x) + (e_y - s_y)(c_y - s_y) \quad (2.7)$$

$$+ (e_z - s_z)(c_z - s_z) \quad (2.8)$$

$$|\vec{s}\vec{e}| = \sqrt{(c_x - s_x)^2 + (c_y - s_y)^2 + (c_z - s_z)^2} \quad (2.9)$$

$$|\vec{s}\vec{c}| = \sqrt{(e_x - s_x)^2 + (e_y - s_y)^2 + (e_z - s_z)^2} \quad (2.10)$$

これらの方法により、発表者の自然な、かつ小さな動作を認識して、それを誇張表現することで発表者のスクリーン内での移動を実現する。

2.4 発表者によるスクリーンへの書き込み

現在、一般的なプレゼンテーションソフトウェアに多く実装されている書き込み機能は、聴衆に注目してほしい箇所を示す際に効果的ではあるが、発表者がコンピュータの前でマウスを操作するためにコンピュータの前まで移動しなくてはならない。本研究ではスクリーン内に重畳表示された発表者自身の手で描画を行えるようにしたい。しかし、発表者の手が常にペンの状態であると、発表者の意図に反して線が描かれてしまう。そこで、板書や一般的な大きさのスクリーンで発表する際に指差す動作を参考にして、発表者が仮想のスクリーンに触れているときだけ、書き込みを行えるようにする。具体的には、手(a)が、腰(w)より上にあり、頭(h)よりも後ろに下がった状態のときだけ、手をペンの状態であるとするすることで、自然な動作を認識して書き込みを行うこと(図2.4)が可能である。



図 2.4: 書き込み時の判断基準

第3章 実験

以上の提案をもとにWindows PC上で実験システムを作成した。本研究ではプレゼンテーションにおいて、発表者が聴衆に示したい箇所を効果的に示せるようにすることを目的としている。本実験システムでは入力デバイスにマイクロソフトから発売されているKinectを用いた。Kinectは光学カメラと距離カメラを併せ持ったデバイスである。これによって発表者のキャプチャーをしている。本実験システムでは、発表者は、前後移動（拡大縮小）、左右移動、背伸び屈伸による上下移動と、スクリーン内への書き込みが可能である。

本実験システムを使用して擬似的なプレゼンテーションを行い、システムの評価を行った。図3.1, 図3.2, 図3.3, 図3.4, 図3.5, 図3.6, 図3.7に実験の様子を示す。なお、システムの利用者が自身でPCのモニタを確認して使用する時に、鏡のように左右反転して表示することで移動を容易にすることが可能である。もちろん、スクリーンを直接見て確認する場合には、左右反転しない表示も可能である。また、評価基準は以下のとおりである。



図 3.1: システム外観

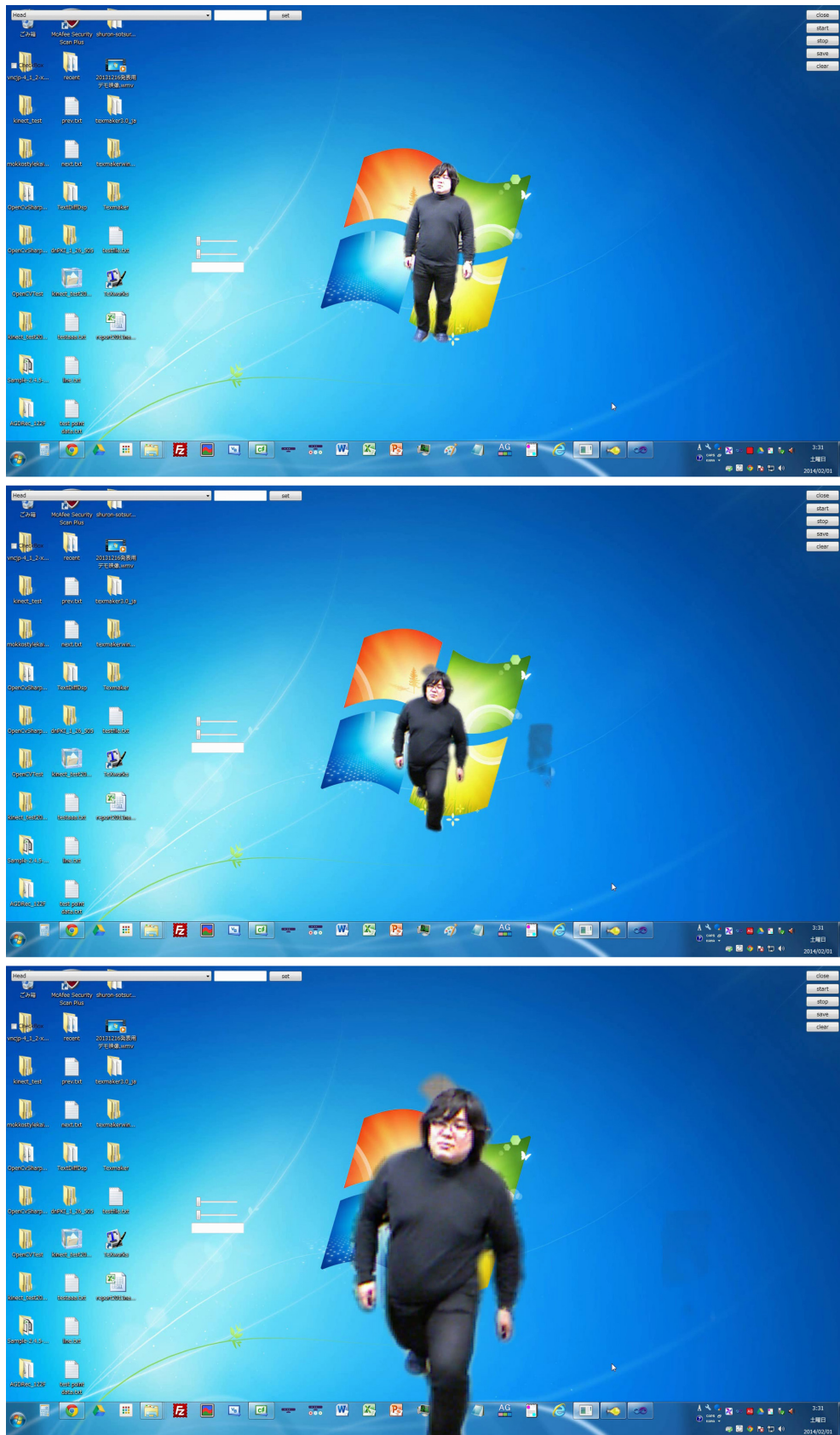


図 3.2: 前進



図 3.3: 後退



図 3.4: 上移動



図 3.5: 下移動



図 3.6: 左右移動

問題解決

親ウィンドウ

ParantHeight

ParantWidth

Kinect出力

KinectHeight

KinectWidth

(Kinectの座標)

$$= \left(\frac{ParantWidth - KinectWidth}{2}, \frac{ParantHeight - KinectHeight}{2} \right)$$

問題解決

新ウィンドウ

ParantHeight

ParantWidth

Kinect出力

KinectHeight

KinectWidth

(Kinectの座標)

$$= \left(\frac{ParantWidth - KinectWidth}{2}, \frac{ParantHeight - KinectHeight}{2} \right)$$

問題解決

親ウィンドウ

ParantHeight

ParantWidth

Kinect出力

KinectHeight

KinectWidth

(Kinectの座標)

$$= \left(\frac{ParantWidth - KinectWidth}{2}, \frac{ParantHeight - KinectHeight}{2} \right)$$

図 3.7: 書き込み

- 発表者は自由にスクリーン内を移動できるか.
- 発表者は自由にスクリーン内に線を描けるか.
- 誇張表現は適切なタイミングで認識されるか.

被験者からの感想と、実験中の様子の観察より以下の通り評価した.

- 前後移動による拡大縮小は概ね良好である.
- 左右移動は実験した環境が狭かったため十分な結果が得られなかったが、狭い場所でも大股移動の認識ができるように改良の余地がある.
- 上下移動は移動速度の調整ができないため大きく移動するときはストレスになる可能性がある.
- 線を描く機能は誇張表現適用前と適用後でズレが発生するため改良の余地がある.
- 誇張表現のタイミングに関しては全体的にぎこちなさがあるが、前後移動による拡大縮小は非常に安定している.

以上の通り、いくつかの問題点は残されているが、比較的良好な結果が得られた。本システムによりプレゼンテーションの支援が可能であると判断できる。

第4章 むすび

本研究では, 大型スクリーンでプレゼンテーションする際に発表者が示したい箇所を聴衆が見失うという事態を解決するために光学カメラと距離カメラを併せ持つ安価で簡単に入手できる Kinect を用いた発表者の切り出しと重畳表示, スクリーン内での移動, 注目箇所への描画機能を実現した.

実用性を考慮した今後の課題は, 直感的な動作でプレゼンテーション資料のページ送り, 戻し機能の実装, 同じく直感的な動作でスクリーン内に描き込んだ線を消す機能の実装である.

謝辞

本研究を進めるにあたって，日頃から多大な御尽力を頂き，ご指導を賜りました名古屋工業大学，舟橋健司 准教授，山本大介 准教授，伊藤宏隆 助教，に心から感謝致します．また，本研究に対して御討論頂きました本学 中村研究室の皆様ならびに中部大学 岩堀研究室の皆様に深く感謝致します．最後に，本研究に多大な御協力頂きました舟橋研究室諸氏に心から感謝致します．

参考文献

- [1] マイクロソフト Kinect,
<http://www.xbox.com/ja-JP/kinect>
- [2] ソニー株式会社 PlayStation Move,
<http://www.jp.playstation.com/ps3/move/>
- [3] 任天堂株式会社 Wii リモコン,
http://www.nintendo.co.jp/wii/features/wii_remote.html
- [4] 中村薫, 田中和希, 宮城英人:
KINECT for Windows SDK プログラミング C#編, 秀和システム, 2012.
- [5] IT用語辞典 BINARY,
<http://www.sophia-it.com/content/ユーザーインターフェース>
- [6] 1999 第13会「大学と科学」公開シンポジウム組織委員会編:
バーチャルリアリティ - 人工現実感と人間のかかわりを考える,
クバプロ, 1999.
- [7] 舘暲: バーチャルリアリティの基礎 1 人工現実感の基礎
- 臨場感・現実感・存在感の本質を探る, 培風館, 2000.
- [8] マイクロソフト msdn KINECT for Windows BLOG,
<http://blogs.msdn.com/b/kinectforwindows/archive/2013/09/16/updated-sdk-with-html5-kinect-fusion-improvements-and-more.aspx>
- [9] マイクロソフト Developer Network, Managed Code Reference, Microsoft.Kinect, JointType Enumeration,
<http://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx>