平成 29 年度 修士論文

液晶ペンタブレットを用いた
仮想コレクションケースの改良

電気通信大学　情報理工学研究科
情報ネットワーク工学専攻　コンピュータサイエンスコース

学籍番号　1631060
氏名　佐野明日香
指導教員　成見 哲　教授
副指導教員　寺田 実　准教授

概 要

本研究では, ディスプレイの中に展示物が飾られているかのように見せる仮想コレクションケースの改良を行った.

日本の住宅面積は先進諸外国と比べて極端に小さいと言われている. このような環境下では部屋の中に多くの物を置くことができない. そこでフィギュアのようなコレクションをバーチャル空間に落とし込む仮想コレクションケースというアイデアがある. リアルな空間の中にディスプレイを溶け込ませ, 3D スキャンした展示物を鑑賞者の視点に合わせて表示することで, リアルな存在感を引き出すことが可能である. これまでの仮想コレクションケースでは立体表示など鑑賞する際のリアリティや楽しさを重視していたが, 実際のコレクションでは展示物を配置する楽しみもある. 特に展示物が 3D スキャンされて仮想空間にあることから, 現実のフィギュアとは違ってポーズ変更が容易であるなどの自由度が増している.

そこで本研究では, コレクションの展示の際に Kinect や液晶ペンタブレットを用いて操作する方法を提案・実装している. 液晶ペンタブレットを用いて表示される背景画像の上でペンを操作することで, ペンの座標や傾き・回転角に合わせて直感的に展示物を移動・傾斜・回転出来る. また, もともと視点位置を検出するために使われていた Kinect から関節情報を取得することで, 鑑賞者のポーズに合わせて展示物のポーズをリアルタイムに変更することも出来る.

液晶ペンタブレットを用いた本システムとマルチタッチ操作が行えるタッチ液晶を用いて, 展示物の操作に関する比較実験を行った. その結果, 操作速度に関しては本システムのほうが早かったが, 操作精度ではマルチタッチ操作の方が誤差が少なかった. 3 軸同時に操作する本システムの方が一度で設定できる反面微調整が効きにくいためと思われる.

目 次

1 はじめに **4**
1.1 研究背景 4
1.2 目的 8
1.3 本論文の構成 8

2 仮想コレクションケース **9**
2.1 裸眼 3D ディスプレイを使った仮想コレクションケース [5] 9
2.2 仮想コレクションケースの立体感と UI の改良 [6] 10
2.3 「仮想窓」用高精度カメラシステムの開発 [7] 10

3 既存技術研究 **12**
3.1 フォースフィードバックインターフェイス 12
3.2 テーブルトップインターフェース 14
3.3 DollhouseVR 14
3.4 zSpace 15

4 Kinect と液晶ペンタブレット **17**
4.1 Kinect 17
4.2 液晶ペンタブレット 21

5 システムの概要 **22**
5.1 目標とする機能 22
5.2 実装した機能 23
5.3 システムの概要 24
5.4 システムのモード 26

6 システムの実装 **27**
6.1 システムの開発環境 27
6.1.1 ソフトウェア開発環境 27
6.1.2 使用したデバイス 27
6.2 展示物の 3D モデルの実装 27
6.3 ケースの実装 28
6.4 展示物の移動機能の実装 30

6.5 展示物の回転機能の実装 33
6.6 視点移動機能の実装 35
6.7 ポーズ変更機能の実装 36

7 システムの評価 39
7.1 展示物の操作に対する評価 39
7.1.1 実験内容 39
7.1.2 実験用に開発したアプリ 40
7.1.3 結果 44
7.1.4 考察 49
7.2 外部発表時のデモンストレーションからのフィードバック 49
7.2.1 デモンストレーションの内容 49
7.2.2 評価結果 50
7.2.3 EC2017 後の研究 50

8 まとめと今後の課題 51
8.1 まとめ 51
8.2 ケース機能 51
8.3 展示物の移動と回転機能 51
8.4 展示物のポージング 52
8.5 視点移動機能 52

1　はじめに

1.1　研究背景

日本の住宅面積は先進諸外国と比べて極端に小さいと言われている．図 1 を参照すると, 全体の住宅面積は 5 か国の中で一番狭い [1]．また, 他の国々と比較して借家の住宅面積はとても狭くなっている．原因としては人口に対する国土面積が狭いことが考えられ、特に都市圏の人口密度は非常に高くなっている．このような環境下では部屋の中に多くの物を置くことはできない．

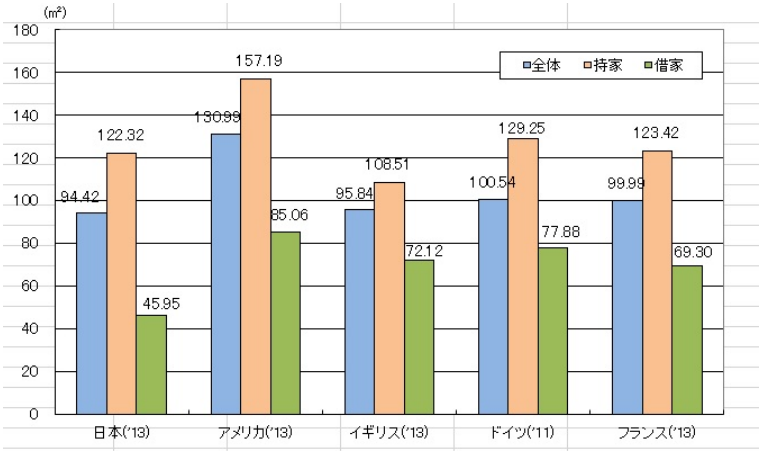

図 1: 一人当たりの住宅面積の国際比較 (参考文献 [1] の図 1)

図 2: バーチャル熱帯魚 (参考文献 [2] より引用)

さらに近年では, ディスプレイをパソコンのモニターとしてではなく, 鑑賞目的の映像を流すために用いる例が見られる. その一つとして, バーチャル熱帯魚 [2] が挙げられる (図 2). これは, 最背面のモニターに魚の泳ぐ映像を映し, 前面の水泡を表示させることで実際に水槽に魚が泳いでるかのように見せている. バーチャルであるが故に, 映像を切り替えることによって水槽内のレイアウトを気軽に変更できるという利点がある.

このようにバーチャル空間に鑑賞物を落とし込むことでレイアウトなどを気軽に変更できるという楽しみ方がある. タブレット端末用アプリであるフィギュコレ [3] では, フィギュアを回転させるなどして鑑賞できるだけでなく, アプリ内のコレクション棚に飾って鑑賞することが可能である (図 3) .

また, リアルな世界に仮想空間を投影する実世界インターフェイスの一種として、Peek-A-Drawer[4] が挙げられる (図 4). これは, 引き出しの中身を撮影し物理的に離れたもう一方の引き出しに設置されたディスプレイに表示するシステムである. 仮想空間をリアルな物体に映すことで実際に物自体を転送したような感覚を得ることができる. このように単にバーチャル空間の映像をディスプレイに表示するのではなく, リアルな空間の中にディスプレイを溶け込ませることで, リアルな存在感を引き出すことができる.

図 3: フィギュコレ (参考文献 [3] より引用)

図 4: Peek-A-Drawer(参考文献 [4] の図 1 より引用)

1.2 目的

本研究ではコレクションケースを仮想世界に取り入れる「仮想コレクションケース」の改良を目的としている. 仮想空間ゆえの楽しみ方も取り入れるが, その際に現実空間においてのコレクションケースへ展示物を飾る楽しみ方も取り入れることとする.

具体例としては, 展示物のレイアウトやポージング, 照明の当て方が該当する. また、現実に展示物を持って配置する感覚に近いユーザーインターフェイスを目指す.

1.3 本論文の構成

本論文の構成と, その内容は以下の通りである.

第一章　はじめに
　住宅面積の狭さと実体の物体を仮想化している例について述べ, 本研究の目的について述べる.

第二章 仮想コレクションケース
　過去に本研究室で行われてきた仮想コレクションケース関連の研究について述べる.

第三章 既存技術研究
　本研究に関連する既存技術及び研究について説明し, 本研究との差異を述べる.

第四章 Kinect と液晶ペンタブレット
　本研究で用いたデバイスである Kinect と液晶ペンタブレットについて述べる.

第五章 システムの概要
　本研究が目標とするシステムの構想について述べる. また構想を基に実装したシステムの概要について述べる.

第六章 システムの実装
　本研究で実際に改良したシステムの実装について述べる.

第七章 システムの評価
　作成したシステムに対する評価実験の手法と結果, デモンストレーションの結果について述べる.

第八章 まとめと今後の課題
　本研究の成果と問題点, 今後の課題について述べる.

2　仮想コレクションケース

本章では, 仮想コレクションケースと関連する研究について記述する. 仮想コレクションケースとは, モニター内に展示物が飾られているかのように見せるシステムである.

2.1　裸眼 3D ディスプレイを使った仮想コレクションケース [5]

神澤によって行われた研究は, ビデオカメラを展示物に対して平行に動かしながら撮影した映像を用意し, Kinect によって検出した観察者の視点位置から最適な映像をディスプレイに表示するシステムである (図 5). このときに裸眼立体視に対応したディスプレイに表示することによって, 立体感のある表示が可能である.

後述する仮想窓システムにより, 観察者の視点位置を検出して最適な画像をリアルタイムで表示する処理は実装されていた. この研究では, 主に展示物の自動撮影システムや立体表示機能を実現しており, 展示物をレイアウトすることに関しては触れられていない.

図 5: 仮想コレクションケース (参考文献 [5] の図より引用)

図 6: クロマキー撮影 (参考文献 [6] の図 1)

図 7: 背景合成後 (参考文献 [6] の図 3)

2.2 仮想コレクションケースの立体感と UI の改良 [6]

新井の研究では, 背景合成と照明によって仮想コレクションケースの立体感の向上と UI の改良が行われた.

背景合成では, クロマキーを用いた撮影方法によって背景に任意の画像を合成し (図 6 , 7), 展示物への照明を正面, 上方, 斜め前から当てることで立体感に変化があるかを比較した. その結果, 背景の合成を行い, 照明の当て方によって影を付けることによって, より立体感の強調された見え方を実現した.

UI の改良では, 展示物の回転方向の追加を行っている. また, ズームや斜め上からの撮影を行うことで新たな視点を追加し, ジェスチャーによってそれら視点を切り替えている.

2.3 「仮想窓」用高精度カメラシステムの開発 [7]

「仮想窓」とは, ディスプレイがあたかも窓のように見えるように画像を表示するシステムである. 多視点で撮影した背景画像から観察者の視点にあわせ適切な画像を抽出し, 画面上に描画することで実現している.

適切な画像を抽出するために観察者の視点位置を検出する必要性があるが, その手法について 2 つの手法を比較している, 1 つは Kinect によって取得した画像から OpenCV によって目の座標を取得する方法 (図 8) であり, もう一つは骨格追従から頭部骨格の中心座標を取得する方法 (図 9) である.

1 フレーム当たりの処理に時間がかかることと偏向式 3D 映像システムによるメガネ着用時の目の認識率の低さから後者の方法を採用している.

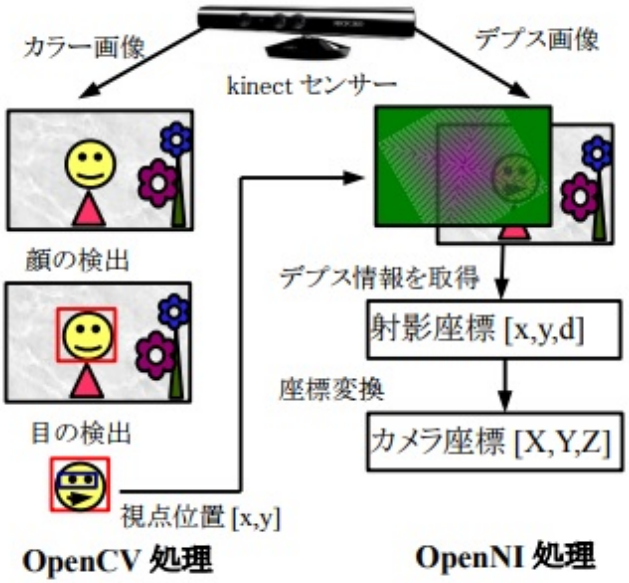

図 8: 画像から観察者の視点位置検出 (参考文献 [7] の図より引用)

図 9: 骨格追従による観察者の視点位置検出 (参考文献 [7] の図より引用)

3　既存技術研究

本章では, 本研究に関連する既存技術及び研究について説明し, 本研究との違いについて述べる.

3.1　フォースフィードバックインターフェイス

仮想空間内への入力に対するユーザインタフェースの一例として, Geomagic Phantom[8] という既存製品が挙げられる.

ユーザーが PC 上の仮想オブジェクトに触れて操作できる高精度の力覚インタラクションを実現するデバイスである. 主に外科手術のリハーサルやリハビリのためのトレーニングシステム, 3D モデリングなどに利用されている. 仮想オブジェクトを通じて物体に触れたときに伝わる反力を再現し, 3 次元的な操作性を向上させている.

ユーザーへフォースフィードバックを行うことで現実での感覚へと近づけているが, デバイスの値段は 45 万円ととても高価である.

図 10: Geomagic Phantom(参考文献 [8] より引用)

図 11: 活発な駆動ペンタブレット LCD パネルを備えた新しい対面型触覚デバイス (参考文献 [9] より引用

また, 他の例では, 活発な駆動ペンタブレット LCD パネルを備えた新しい対面型触覚デバイス [9] という研究が挙げられる. この研究では, よりリアルな操作入力のために液晶ペンタブレットへフォースフィードバック機能を取り入れている. 液晶ペンタブレットの背面にアームを取り付け, ペンによる圧力を感知するとペンタブレットが沈むように動く (図 11). また, カメラによってユーザーの位置を検出しており, 装置に近づくと液晶部をユーザー側に向ける機能がある. 液晶ペンタブレットの角度によって, 液晶部に表示されているオブジェクトの見え方が更新される.

本研究との差異

フォースフィードバックを行うために専用のデバイスを用いるため, 高価格になりやすく入手性も悪い. 本研究では, 展示物を置いた感覚のフィードバックだけに絞ることで手に入りやすいデバイスを用いた.

図 12: テーブルトップインターフェース (参考文献 [10] の図 7 より引用)

3.2 テーブルトップインターフェース

仮想空間への直感的な入力方法として, テーブルトップインターフェース [10] という研究が挙げられる (図 12). テーブルトップインターフェイスは複数人で直感的に手で操作するインターフェイスとして利用されている. この研究では, 赤外線や映像処理によって, 手や物体を認識することで操作を入力し, 仮想空間内でオブジェクトを操作する.

複数人で操作可能であり, 大型であるが比較的低価格な入力方法である.

本研究との差異

複数人で操作することを想定しているため, テーブル上を全て撮影できるカメラや赤外線カメラを使用している. 本研究では, 複数人の操作は出来ないが, 大規模な設備は不要である.

3.3 DollhouseVR

DollhouseVR[11] は, 仮想空間上の部屋をレイアウトするシステムである (図 13). 部屋の俯瞰画像が表示されたタブレットをタッチやスワイプなどの操作をすることで, 家具の配置を変更する. VR ゴーグルを装着することでレイアウトした仮想空間上の部屋に入ることも可能である. 俯瞰画像を表示することでレイアウトの配置をわかりやすくし, 仮想空間へのインタラクションをタブレットを触れる操作によって行っている.

図 13: DollhouseVR(参考文献 [11] の図 1 より引用)

本研究との差異

本研究では, 移動や回転に加えて傾斜といった操作を行うため, マルチタッチ操作では対応が難しい. また, 操作対象の形 (ポーズ) を変更する操作も必要なため, マルチタッチ操作は適していない.

3.4 zSpace

zSpace[12] は専用のスタイラスペンとディスプレイを使用した、仮想現実プラットフォームである. ユーザーの 3D メガネをリアルタイムで追跡し, スクリーンに表示した仮想オブジェクトがまるで浮き出ているかのように立体的に表示することができる (図 14). また, スタイラスペンによって, 浮き出ている仮想オブジェクトを引っ張るようにして, 動かすことが可能である.

本研究との差異

スタイラスペンは 3 次元空間の位置を検出できるため, 本研究で使用している液晶ペンタブレットよりも自由度が高い. ただし, zSpace のアプリケーションでは, 仮想オブジェ

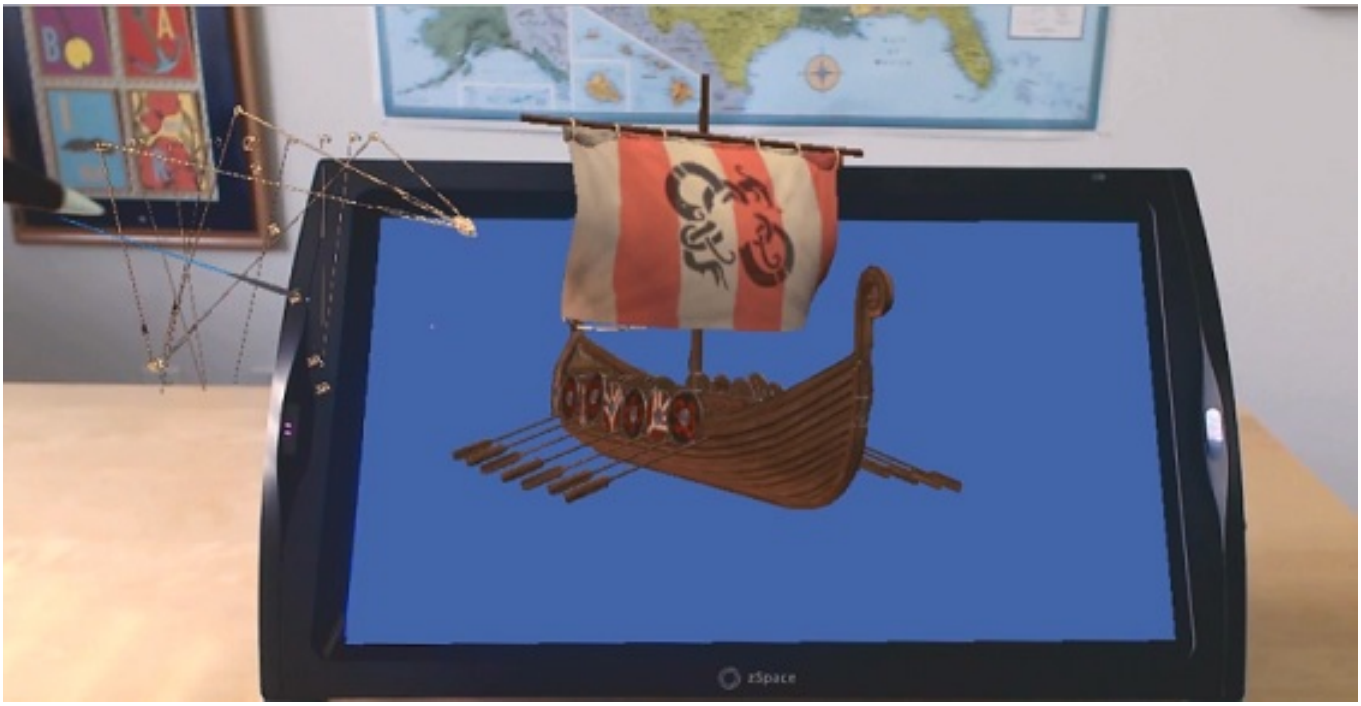

図 14: zSpace(参考文献 [12] より引用

クト自体とスタイラスペンの位置を一致させて操作させておらず, 展示物を置く感覚とは違うことと考えられる.

4　Kinect と液晶ペンタブレット

本章では, 本研究で使用する 2 つのデバイスについて説明する.

4.1　Kinect

Microsoft 社から発売されている Kinect v2(以下 Kinect)[14] を本研究では採用した (図 15). Kinect には, カメラ以外にも Depth センサと音センサが内蔵されている. 詳しい仕様は表 1, 2 に記載する. 人体の関節情報を取得する際には, 取得した Color 情報 (図 16) と Depth 情報 (図 17) を用いて各骨格部位ごとに Kinect からの距離を算出し, 識別器によって骨格部位が体のどの部位に当たるかを割り出している (図 18). 音センサではマイク入力も可能である.

本来はゲーム用デバイスであるが, PC 接続用のケーブル [15] と公式から提供されている SDK[16] を用いることでパソコンからの制御が可能である. なお, ケーブルは 2018 年 1 月 15 日現在, 生産は中止されている.

図 15: Kinect v2

表 1: Kinect v2 の動作環境 (参考文献 [16][17] より引用)

項目	要件
対応 OS	Windows8 以降
消費電力	36W
CPU	2.66 GHz 以上のデュアルコア 64 ビット (x64) プロセッサ
接続ポート	USB3.0, PCI-Express 経由の USB 接続の場合は PCI-EX2 Gen2 (5Mbps) 以上
メモリ	2GB RAM
GPU	DirectX 11 対応グラフィックスカード

表 2: Kinect v2 の仕様 (参考文献 [18] より引用)

項目	仕様
Color(色) 情報	解像度: 1920 × 1080 , fps: 30fps
Depth(深度) 情報	取得範囲: 0.5m から 8.0m
人物領域	6 人
人物姿勢	6 人
関節	25 関節/人 (図 18 を参照)
手の開閉	あり
角度	水平: 70 度, 垂直: 60 度

図 16: Kinect の Color 画像

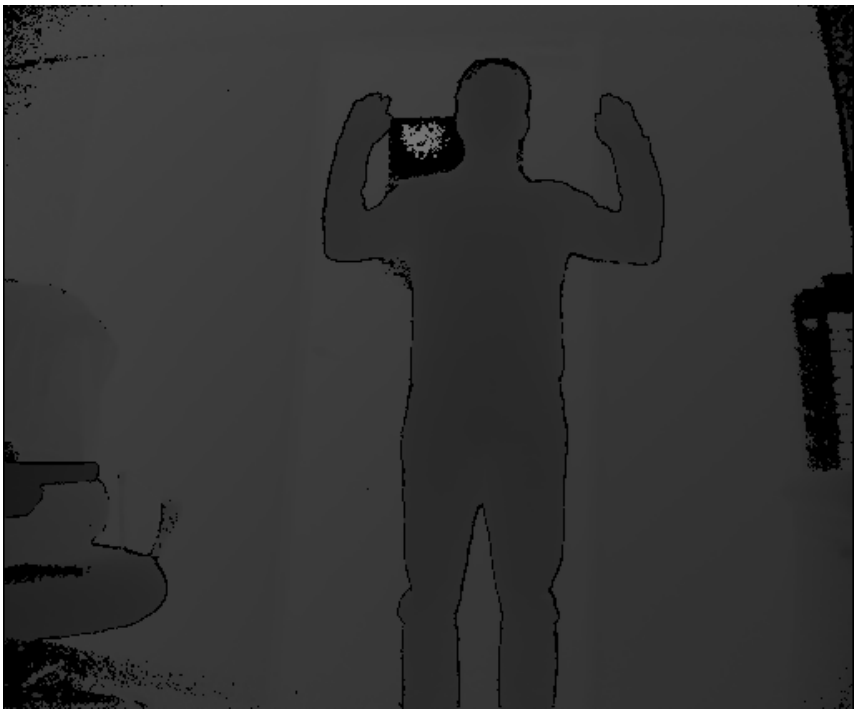

図 17: Kinect の Depth 画像

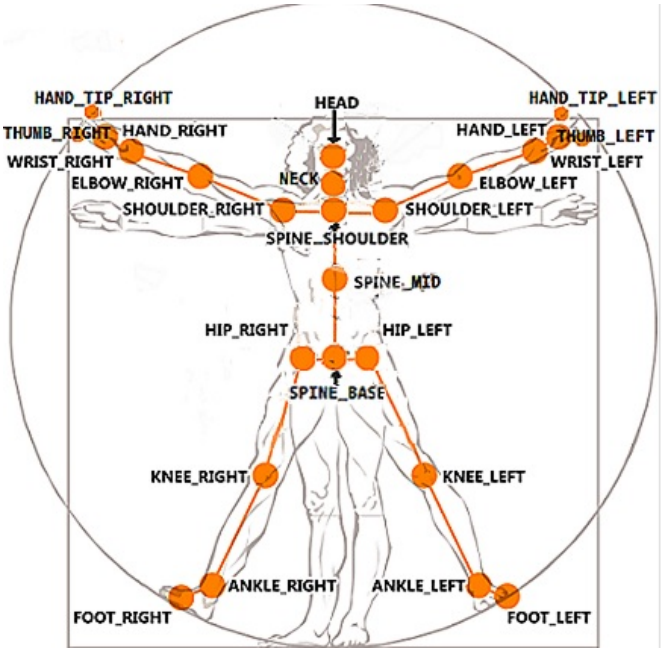

図 18: Kinect の検出骨格 (参考文献 [18] より引用)

4.2　液晶ペンタブレット

Wacom 社から発売されている Cintiq 13HD[19] を本研究では採用した (図 19). 液晶ペンタブレットはタブレットとペンで構成されており, ペンの位置, 筆圧が取得できるデバイスである. タブレット部分が液晶になっており, 外部からの映像入力ができるものを液晶ペンタブレットと呼ぶ.

主に, CAD やイラスト描画, 電子サインなどに使用されている. 液晶ペンタブレットは, 単純な選択操作において正確で最も速いという利点がある [13]. そのため今回は, 操作用デバイスとして採用している.

ペンには付属のプラペンではなく, 別売りのアートペン [20] を使用した. アートペンは, 筆圧やペンの傾斜角度だけではなく, ペン軸の回転角度も取得が可能である. 今回は操作の種類が多いので, 取得できる情報が多いアートペンを採用した.

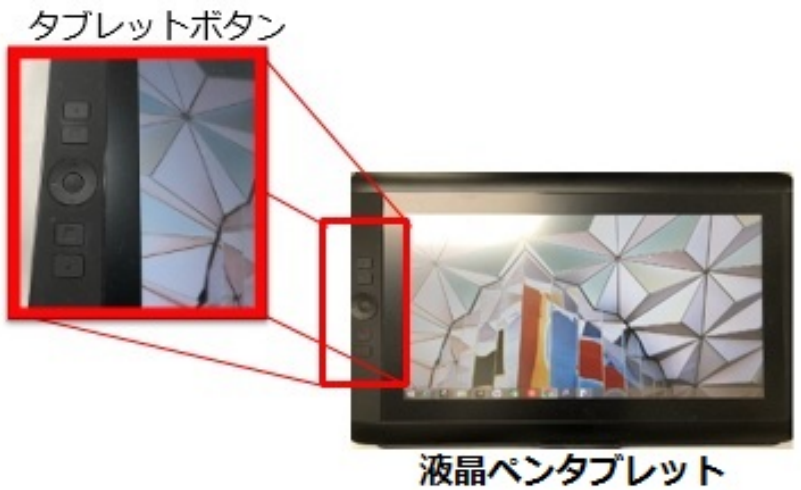

図 19: Cintiq13HD

5　システムの概要

本章では, まず本研究が目標とするシステムの構想について述べる. 次に実際に開発したシステムの概要について説明する.

5.1　目標とする機能

仮想コレクションケースは, 展示物を飾る基本的な機能を備えているだけでなく, 展示物を飾るためのセットアップ作業を楽しんだり, 展示物と鑑賞するユーザーの間のインタラクションがあるなど, 仮想世界ならではの機能が求められる.

そこで, 目標とするシステムの構想について述べる. これらは構想であり, 今回実装していない機能も多い.

ケース機能
: 展示物を保管するための箱状の形をした空間 (ケース) を備える. ケースは仮想空間ではなく実空間で箱状の形をしたディスプレイで構成される. 仮想空間内にもそれと同一のケースを備える.

展示物の 3D モデル表示機能
: 現実で所持している展示物を 3D スキャンするなどして 3D モデルデータを作成し、ディスプレイに表示できる. 3D モデルにアニメーションの動きを設定すると手を振るなどの動きを表現できる.

展示物の移動機能
: 展示物を移動できる. 少ない動作や直感的な方法で行えるのが望ましい.

展示物の回転機能
: 展示物を任意の角度に回転できる. 少ない動作や直感的な方法で行えるのが望ましい.

展示物のポーズ変更
: 展示物が人のような物体の場合, 任意のポーズへ変更することができる.

背景変更機能
: ケースの背景を変更できる. 展示物の種類によって自動的に変わる.

レイアウト保存読み込み機能
: 作成した展示物のレイアウトを保存できる. 保存したレイアウトを読み込むことができる.

視点移動機能

ユーザーが動くとその位置から見たときのコレクションケース内を角度に応じてディスプレイに表示できる.

ステレオ表示機能

ディスプレイに立体表示 (ステレオ表示) でき, 奥行を感じることができる機能.

展示物とのインタラクション機能

ユーザーが声で呼びかけると, 展示物の動きや音で反応を返す機能.

5.2 実装した機能

目標とする機能のうち, 本研究では, 以下の機能の実装行った.

ケース機能

物理的にはディスプレイ一枚で箱状にはなっていないが, 仮想空間では箱状のケースを Unity を用いて作成した.

展示物の 3D モデル表示機能

今回使用した 3D スキャナでは, 50cm 以下の小さいものは正確にスキャンできなかったため, 1m 以上となる人間をスキャンして 3D モデルデータを作成した. また, そもそも 3D モデルの存在するフィギュアも表示できる.

展示物の移動機能

液晶ペンタブレットのペンを操作することで展示物を移動できる.

展示物の回転機能

液晶ペンタブレットのペンを操作することで展示物を回転できる.

展示物のポーズ変更

Kinect でユーザーのポーズを検出し, 展示物も同じポーズに変更できる.

背景変更機能

今回は実装していない.

レイアウト保存読み込み機能

今回は実装していない.

視点移動機能

Kinect によってユーザーの目の位置を検出し, その位置から見える角度で展示物を表示できる.

ステレオ表示機能

ステレオ表示には対応していない.

展示物とのインタラクション機能

今回は実装していない.

5.3 システムの概要

システムの全体を図 20 に示す. 液晶ペンタブレットからはペン位置を, Kinect からは音声と人物の位置, 関節情報を受け取り, PC で処理を行う. 処理結果を液晶ペンタブレットの液晶とディスプレイにそれぞれ出力を行う. 図 21 は全体の写真である.

液晶ペンタブレットからの入力によって, コレクションケース内部の展示物の位置を変更する. その際に, 回転操作も行う. また, Kinect からユーザーの位置情報を検出し, その位置から見えるであろうコレクションケースの見え方で表示する. ユーザーが操作しなくても, 様々な方向からレイアウトの確認および鑑賞が可能である. Kinect からの骨格情報を用いて展示物のポーズを変更できる.

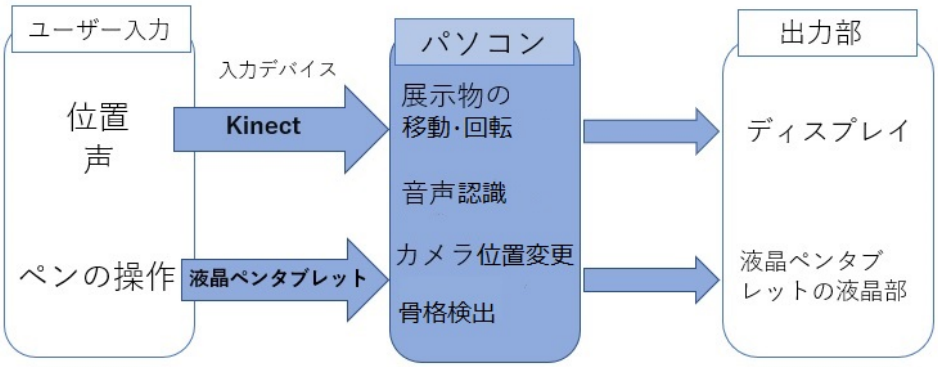

図 20: システムの全体ブロック図

図 21: システムの全体図

5.4 システムのモード

本システムには, 2 つのモードが存在する. それぞれのモードへの切り替えはスペースキーを押すことによって行う.

操作モード

展示物のレイアウトやポーズを変更するモードである. 液晶ペンタブレットと Kinect を用いて, コレクションケースの内部をレイアウトする. 主に展示物の移動やポージング, 操作に特化した視点の動きを行う.

鑑賞モード

レイアウトしたコレクションケースを鑑賞するモードである. Kinect を用いて, 鑑賞者の視点にあったコレクションケースの見え方で表示する. 主に鑑賞に特化した視点の動きを行うため, 展示物の操作はできない.

6　システムの実装

本章では, 本研究での開発環境やシステムの全体, 実装方法について説明する.

6.1　システムの開発環境

6.1.1　ソフトウェア開発環境

3D モデルを扱うために Unity を用い, 開発当時の最新バージョンである 5.6.1f を使用した. また, 言語は C#を用いた.

6.1.2　使用したデバイス

本研究では, 前述した Kinect v2 をユーザー検出用デバイスとして, Cintiq 13HD を操作用デバイスとして使用した.

今回は Unity で 3D モデルといった素材やエディタ拡張用のスクリプトなどを使用するためにパッケージングされたアセットを使用した. それぞれの操作用デバイスを使用するために使ったアセットは下記の通りである.

WinTablet[22]
: 液晶ペンタブレットの入力情報を Unity で使用するためのスクリプトが入ったアセット. 本研究では, 液晶ペンタブレットの入力によって展示物を操作するために使用した.

Kinect v2 with MS-SDK[23]
: Kinect で取得した情報を Unity で使用するためのスクリプトを含んだアセット. 骨格認識や音声認識を行うために使用した. 音声認識には, 実行環境用の Speech Platform-Runtime[24], 音声認識用 SDK である Speech PlatformSDK[25] を使用し, 音響モデルは Kinect のマイクに最適化された LanguagePack[26] の日本語版を使用した.

6.2　展示物の 3D モデルの実装

展示物を仮想化するにあたり, 展示物を 3D モデルとして取り込む必要がある. そこで, 赤外線センサーを内蔵したカメラで撮影したオブジェクトを 3D モデル化するアプリ、iTSeeze[27](図 22) によって展示物を 3D スキャンした. 展示物を 3D モデル化した後, 3DCG 作成ソフトである Blender[28] によって, 3D モデルを動かすための仕組み (以下, Rig と表記する) を入れた.

図 22: iTSeez3D

後述するコレクションケースへとインポートされた 3D モデルは, 3D モデルの名称と情報を格納するためのリスト (DisplayList) に格納されて管理を行う.

6.3 ケースの実装

展示物を展示するコレクションケースは, Unity 上で 5 辺を囲った箱を作成し, 図 23 のようにカメラオブジェクトを 2 つとカーソルオブジェクトを設置した. それぞれのカメラの表示先と撮影方法は表 3 の通りである.

カメラ 1 では, 液晶ペンタブレットで展示物を操作するためにコレクションケース上部から撮影した (図 24). 撮影の仕方は遠近感を生じさせないために平行投影を用いた. カメラ 2 では, 主に観賞用として用いるためコレクションケースを横から撮影した (図 25). 撮影の仕方は遠近感を出すために透視投影を用いた.

カーソルオブジェクトは, コレクションケース上方の 1 面を移動するように設定した.

表 3: カメラの出力先と撮影方法

名称	出力先	撮影方法
カメラ 1	液晶タブレット	平行投影
カメラ 2	ディスプレイ	透視投影

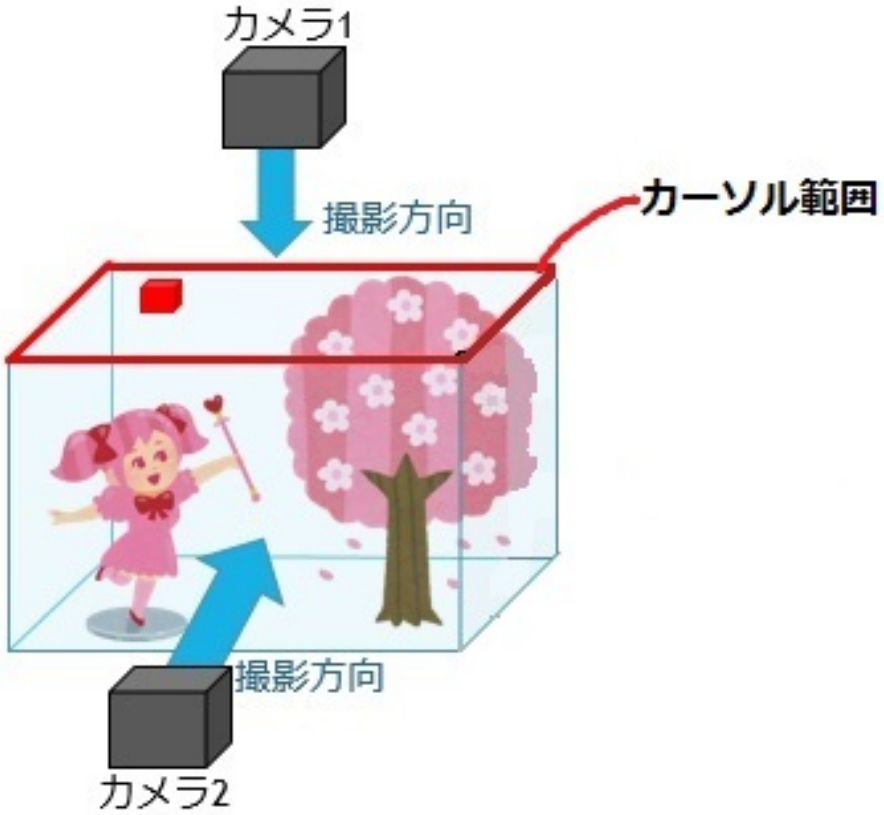

図 23: コレクションケースの配置

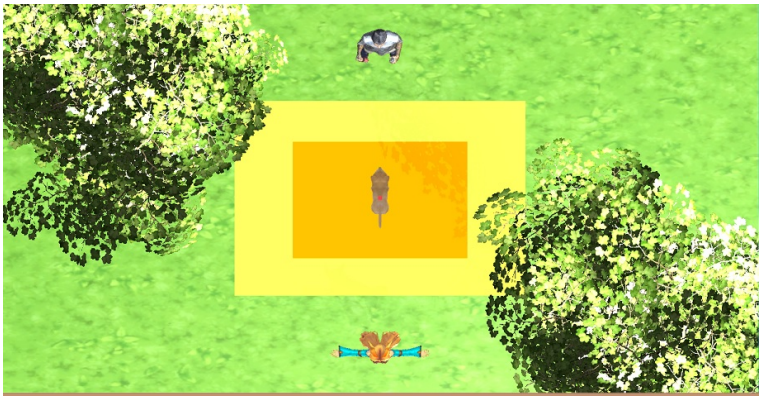

図 24: カメラ 1

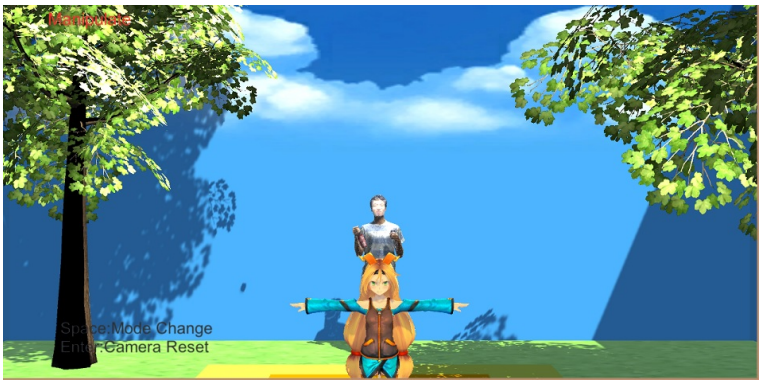

図 25: カメラ 2

6.4　展示物の移動機能の実装

展示物の移動では, 仮想コレクションケースの俯瞰画像を液晶ペンタブレットの液晶部に表示し, ペンで展示物を操作する.

カーソルオブジェクト作成のためには, 最初に Wintablet アセットのスクリプトから、液晶タブレットに対するペンの座標を取得した. 次に, コレクションケースの上部の面を移動するカーソル作成のために, 取得したペンの座標を俯瞰画像を撮影するカメラのスクリーン座標へと変換した. その後, スクリーン座標からワールド座標に変換した数値を, カーソルオブジェクトの位置座標に代入することでカーソルオブジェクトを作成している. 展示物の移動を行う処理については図 26 に示す.

カーソル位置の更新では, 液晶ペンタブレットのペンの位置に合わせるため, カーソル位置の座標を毎フレーム更新する. 次にタブレットがペンからの圧力を感知しているかを確認する. 感知されていない場合には, 処理は終了し, Start に戻る. 感知されていた場合, カーソルの座標に展示物があるかどうかを DisplayList を参照し, 確認する. 展示物の存在が確認されない場合, 処理は終了し, Start に戻る. 展示物の存在が確認された場合, 展示物の中心座標をカーソル位置の座標まで移動し, 液晶タブレット側に付属しているボタンを押しながら行った場合は次節で述べる展示物の回転操作を行う.

また, これらの方法では移動や回転操作を行う場合に常にペンをタブレットへと押し付けて圧力を感知する必要性があったため, 操作が行いづらいという観点から図 27 の処理に変更した.

展示物を選択する際に操作している対象を変数によって保持することで, ペンからの圧力を感知しない状態でも移動や回転の操作を行うことができるよう変更した.

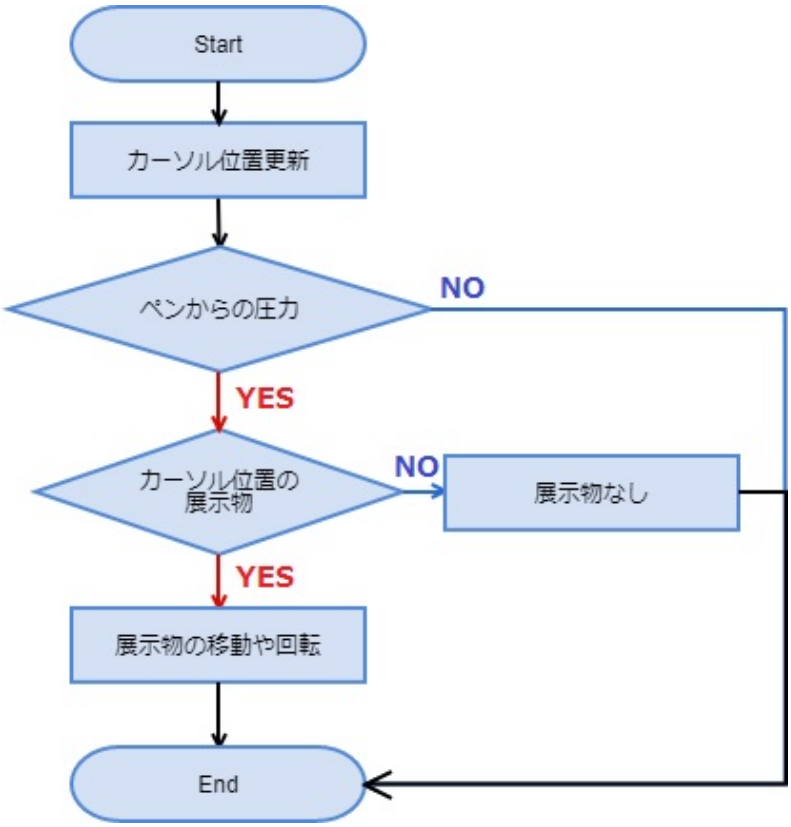
Start
カーソル位置更新
ペンからの圧力
NO
YES
カーソル位置の
展示物
NO
展示物なし
YES
展示物の移動や回転
End

図 26: 展示物の移動

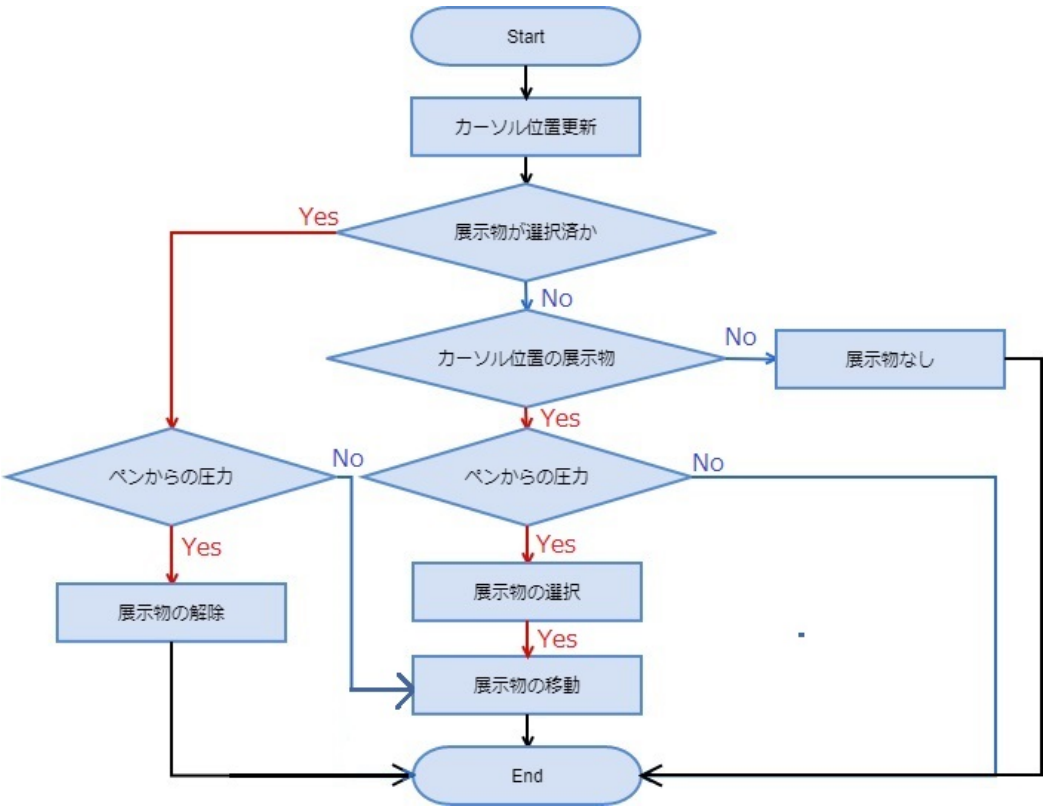

図 27: 実装した展示物の移動

6.5 展示物の回転機能の実装

展示物の回転においては, ペンの傾斜角度とペン軸の回転から操作を行う. アートペンからとれる情報を表 4 および図 29 に示す.

3 つの数値から, 展示物の角度を割り出して回転させる必要がある. そのために, 傾斜角度, 傾斜方位の 2 種から展示物を傾斜させる角度を算出し, FromToRotation 関数を用いることで指定した傾斜角度まで World 座標を基準とした軸で展示物を回転させる (図 28 の左側). その後, 展示物基準とした y 軸を中心にアートペンから得られた軸回転の数値で回転を行うことで, アートペンの動きと同期している (図 28 の右側). アートペンの傾斜, 回転と展示物の回転との対応を図 29 を示す.

表 4: アートペンからの回転に関する取得情報

種類	数値
傾斜角度	0(タブレットに対してペンが垂直) ～ 1.0(タブレットに対してペンが平行)
軸回転	(タブレットの上方向を北として時計回りに)0 ～ 1.0
傾斜方位	(タブレットの上方向を北として時計回りに)0 ～ 1.0

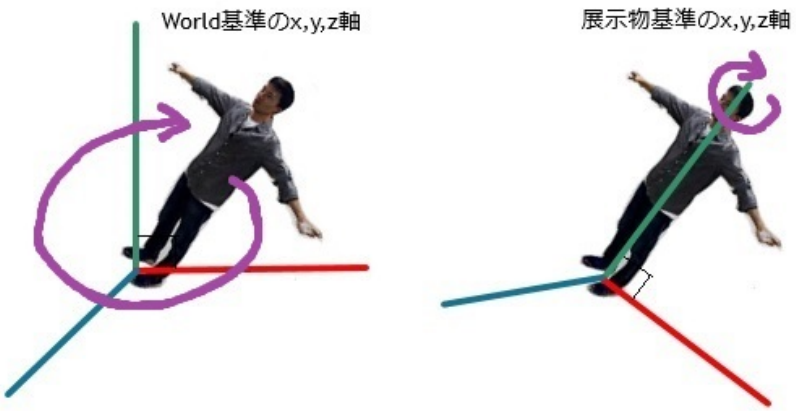

図 28: 展示物の回転

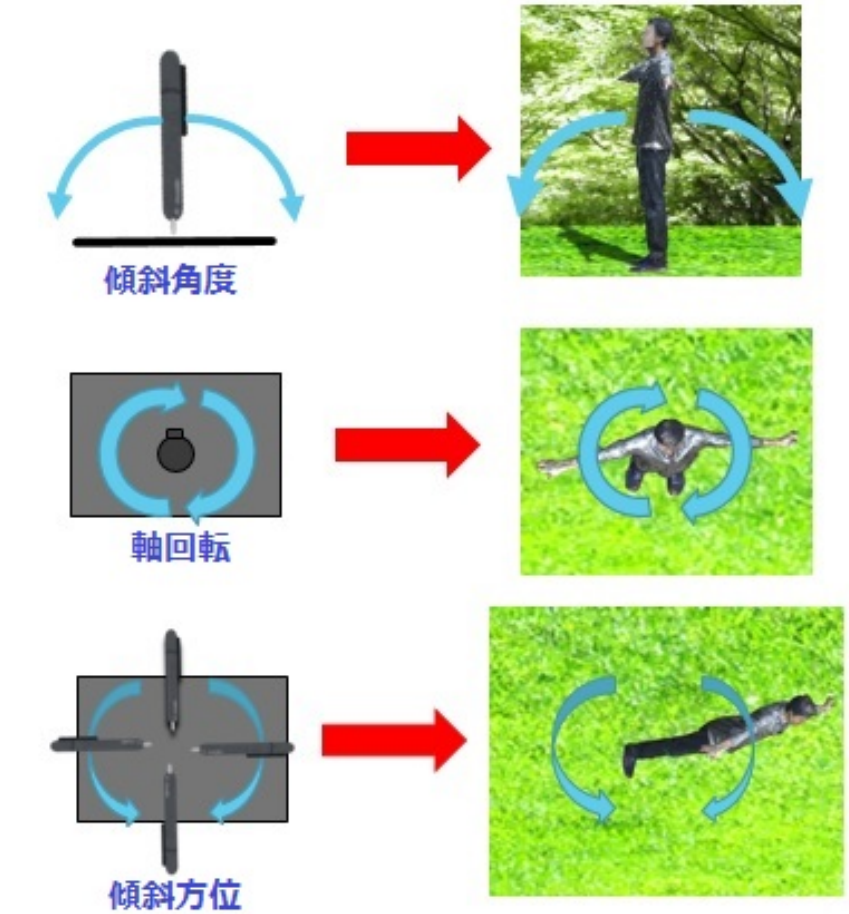

図 29: アートペンからの取得情報と展示物の操作

6.6 視点移動機能の実装

Kinect を用いてユーザー位置を追跡し、その座標からディスプレイに映すカメラの位置を調整した. Kinect に対するユーザー位置の座標を取得し, 仮想空間内のカメラをコレクションケースの中心回りに回転する仕組みを採用している.

図 30 の左は現実の Kinect とユーザーの位置, 図 30 の右は仮想空間内のコレクションケースとカメラの位置を上から眺めた図である. Kinect を原点としてユーザーの位置は x, y, z 座標で取得できる. Kinect の z 座標から (1) 式を用いて円の半径 r を取得し, x 座標から (2) 式によって回転角 θ_U を取得した. その後, 式 (3), (4) によってそれぞれ x_U, z_U 座標を求めた.

操作用モードでは, 定数は $a = 3$ である. 基本的に振り幅を大きくし, 少ない動きでコレクションケースの後ろまでカメラを回り込ませる必要があるからである観賞用モードでは, 定数は $a = 1$ である. あまり振り幅を大きくせずに, 現実に近い移動感覚でカメラを動かすためである.

なお, 実物のコレクションケースを眺めるときはケースが回転することはないが, 本研究では展示物の後ろを簡単に確認できるようにするために, ユーザーが右に移動するだけでケース内の展示物を時計回りに回転させている.

$$r = z, \tag{1}$$

$$\theta_U = a\frac{180}{x}, \tag{2}$$

$$x_U = rsin\ \theta_U, \tag{3}$$

$$z_U = rcos\ \theta_U, \tag{4}$$

ここでの a は定数であり, モードによって値が異なる.

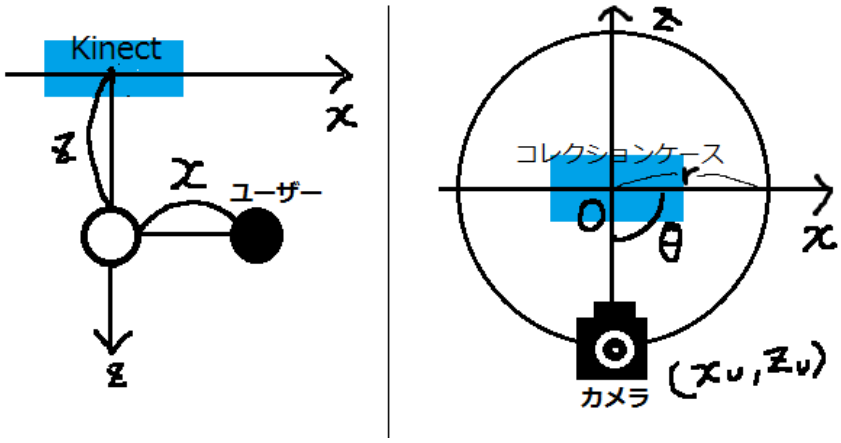

図 30: カメラの位置

6.7 ポーズ変更機能の実装

展示物を選択した際に, ペンに付属したボタン (図 31) を押すことによって展示物のポージングモードが開始される. Kinect の正面で, 展示物にさせたいポーズをユーザーが行うことでポージング変更を行う. 検出したユーザーの関節の角度をそれぞれ算出し, 展示物のモデルの関節にその角度を反映させることによって同じポーズを再現している. 今回は最初から関節位置の設定されたモデルを実験用に用いた.

また, ポージングの際には音声で操作を行う. Kinect のマイクを通じて音声を取得し, SDK で音声認識し文字列に変換する. 操作内容は表 5 に記載する.

展示物のポージング停止では, 展示物の 3D モデルの関節に角度を反映させるのを停止することによって, 音声コマンドを入力した瞬間のポーズを維持する. また, ポージングの開始では, 展示物の関節に角度を反映させることを再開する. ポージングモードの終了では, ポージングを終了して, 展示物の選択へと再び戻る.

ポージングモードに入った場合にはその展示物が選択されているので, 別の展示物を選択したい場合には一度ポージングモードを終了させる必要がある.

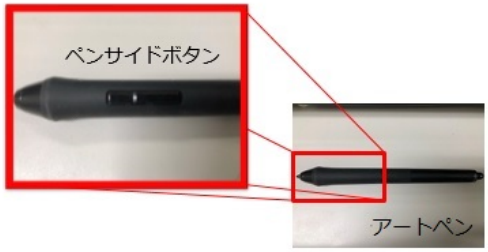

図 31: アートペン

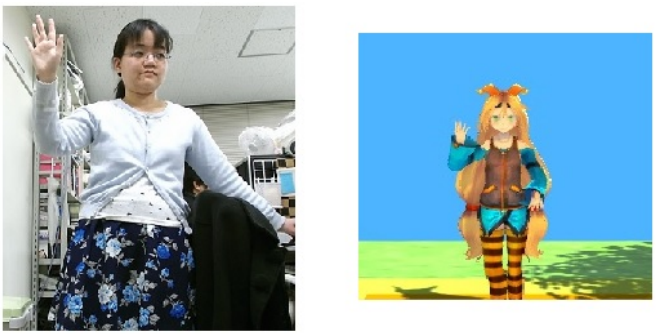

図 32: ポーズ変更の例

表 5: 音声コマンド一覧

音声コマンド	操作
とまって ポーズ	展示物のポージング停止
うごいて	展示物のポージング開始
終了	ポージングモードの終了

7　システムの評価

この章では, 実装したシステムに対する被験者実験とその結果，対外発表で行ったデモンストレーションからのフィードバックについて記述する.

7.1　展示物の操作に対する評価

今回実装した ” 展示物を移動や回転する操作 ” に対する評価実験を行った.

7.1.1　実験内容

液晶ペンタブレットを用いた本研究での操作方法以外に, 比較のためタッチ機能付ディスプレイによるマルチタッチ機能を用いた操作方法に関して, 評価実験を行った.

一つのシステムに予め 3 個の展示物が配置された見本のコレクションケースを用意し，もう一方のシステムでは 2 種類の操作でそれぞれ見本の通りに展示物を設置する. その際に完成までにかかった時間と展示物の位置によって，操作速度と精度を評価する.

実験手順は以下の通りである.

1. 2 つの操作方法について説明し, それぞれ操作を 3 分間行い, 慣れる.
2. 見本のシステムの NEXT ボタンをクリックし, 見本システムへ展示物をセット，タイマーにて計測を開始する
3. 比較用のマルチタッチ操作によって, 見本のシステムの通り展示物を設置する
4. 設置が終了した時点の時間と展示物の座標を記録し, NEXT ボタンをクリックして見本システムの展示物を再設置する
5. 本研究の液晶ペンタブレットを使って, 見本システムの通り展示物を設置する
6. 設置が終了した時点の時間と展示物の座標を記録する
7. 2〜5 を 2 回繰り返し，かかった時間と展示物の座標誤差, 回転誤差を平均する

操作用システムにおいては, 2〜5 の手順が終わるごとに初期の展示物の位置はリセットされる. なお、見本システムと比較用のマルチタッチ操作アプリについては次項において記述する. 初期配置の展示物の座標や角度は表 6 に, 見本の展示物の座標や角度は表 7 に示す.

座標誤差に関しては, 見本システムでの展示物の座標及び回転角度と実験で置いた展示物の座標及び回転角度の差を取得し, 差を加算後, 人数で割って平均誤差を算出する.

表 6: 操作用展示物の初期座標と初期角度

	座標 x	座標 z	回転角度 x(度)	回転角度 y(度)	回転角度 z(度)
展示物 1	0	-4.0	0	180	0
展示物 2	0	4.0	0	180	0
展示物 3	0	0	0	0	0

表 7: 見本の展示物の座標と角度

	座標 x	座標 z	回転角度 x(度)	回転角度 y(度)	回転角度 z(度)
展示物 1(1 回目)	-8.0	1.5	30	30	0
展示物 2(1 回目)	8.0	-4.0	0	-60	0
展示物 3(1 回目)	-4.0	-2.5	-20	30	0
展示物 1(2 回目)	-3.0	1.5	0	180	60
展示物 2(2 回目)	5	-4.0	0	180	0
展示物 3(2 回目)	-5.5	3.0	-30	120	0

7.1.2 実験用に開発したアプリ

今回の実験では見本用のコレクションケースを表示するプログラムと操作の比較用のマルチタッチ入力の展示物操作システムの 2 種類を開発した. カメラ操作などは, 本研究のシステムと同じものを使用した.

見本システム

展示物の飾られたコレクションケースを表示するアプリである. 画面に表示された展示物とコレクションケース内の他のオブジェクトを見ることで, 展示物の位置を把握する.

図 33 の黄枠で囲われた画面右上の Next ボタンを押すことによって, あらかじめ登録されている数種の中からレイアウトパターンを切り替えて表示する. 画面左上赤枠範囲に表示されるテキストは操作方法の指定であり, Pen(液晶ペンタブレット操作) と Touch(マルチタッチ操作) の 2 種類のテキストが表示される.

視点の移動については本研究システムのカメラの動きと同じ設定を利用している.

操作比較用マルチタッチアプリ

展示物の移動操作と回転操作を本研究のシステムと比較するためのアプリである.

マルチタッチ操作アプリでは, 移動したい展示物をタップしたまま, 指を移動させることで展示物の位置を移動させることができる (図 34). また, 図 35 のように, 展示物をタッチしている指を軸にもう 1 本別の指で赤矢印のように動かすことによって, 展示物を軸中心で回転することが可能である. さらに, 展示物をロングタップすることによって傾斜操作を行える状態 (図 36) になり, この状態ならばスワイプによって傾斜操作を行うことが可能である.

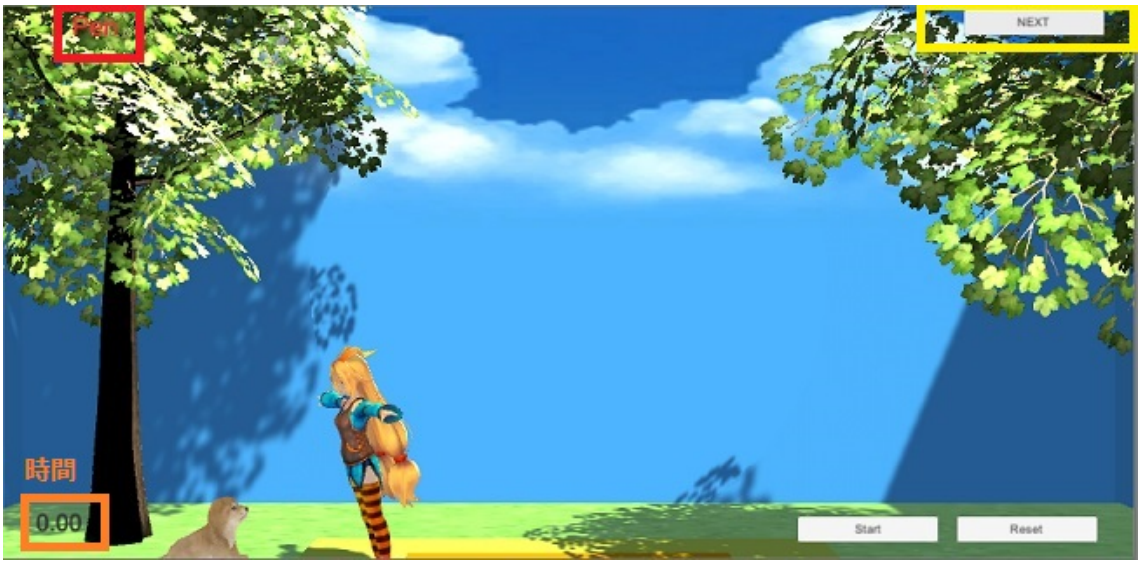

図 33: 見本用アプリ

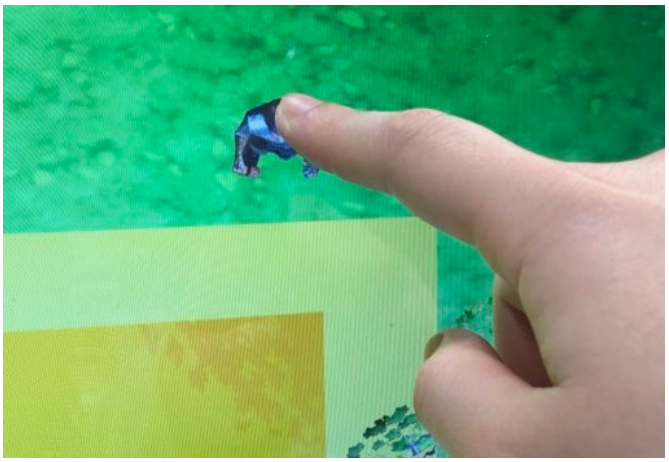

図 34: 比較用マルチタッチ操作での展示物の移動

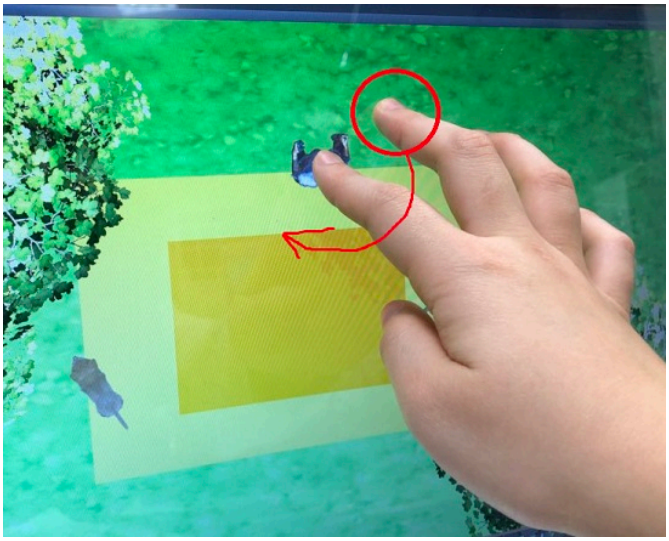

図 35: 比較用マルチタッチ操作での展示物の回転

図 36: 比較用マルチタッチ操作での展示物の傾斜

7.1.3 結果

2 つの操作によって, 完成までにかかった時間を表 8 に, 平均と標準偏差を表 9 に示す. また, 図 37 では, それぞれの操作における実験 2 回分の時間平均を表している. これらは被験者 5 名の時間の平均を出している.

表 8: 操作方法別の完成時間 (単位:秒)

	マルチタッチ 1 回目	本システム 1 回目	マルチタッチ 2 回目	本システム 2 回目
被験者 1	122.24	128.00	188.27	231.59
被験者 2	107.53	42.08	281.28	90.8
被験者 3	87.72	78.37	303.2	80.13
被験者 4	154.89	76.62	204.84	78.00
被験者 5	108.73	41.16	293.46	58.45

表 9: 操作方法別の完成時間比較 (単位:秒)

	マルチタッチ (標準誤差)	本システム (標準誤差)
1 回目	116.22(11.13)	73.24(15.87)
2 回目	254.21(23.94)	107.79(31.39)

また, それぞれの操作による座標精度を表 10, 図 38, 39 に, 回転角度の精度を表 11, 図 40, 41 に示す. 被験者一人一人の結果に関しては巻末の付録 A に添付する. 1 回目の結果 (図 38,40) は, 被験者 4 名の誤差を平均し, 2 回目の結果 (図 40, 41) は被験者 5 名の誤差を平均したものを算出している, 1 回目の結果では, 1 名が誤った測定の仕方を行ってしまい, 値が正確に計測できていないため, 考慮していない.

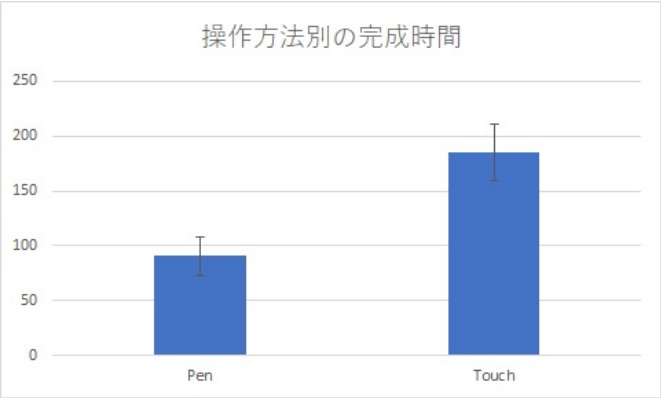

図 37: 操作方法別の完成時間比較 (単位:秒)

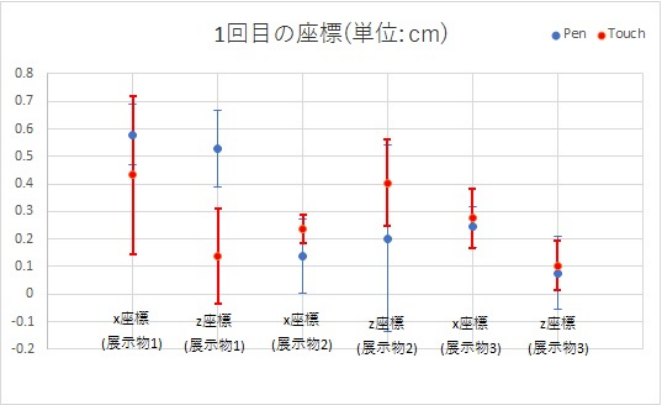

図 38: 1 回目の 3 つの展示物の座標誤差

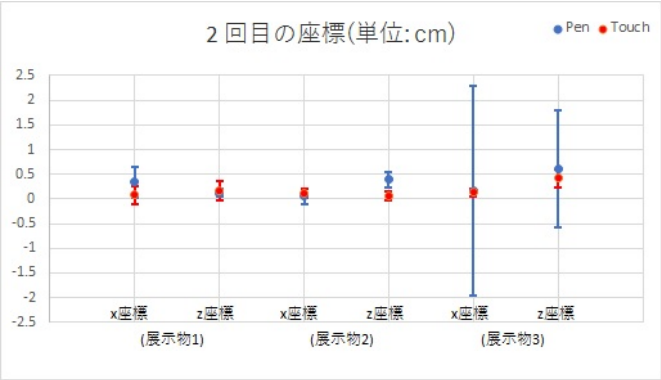

図 39: 2 回目の 3 つの展示物の座標誤差

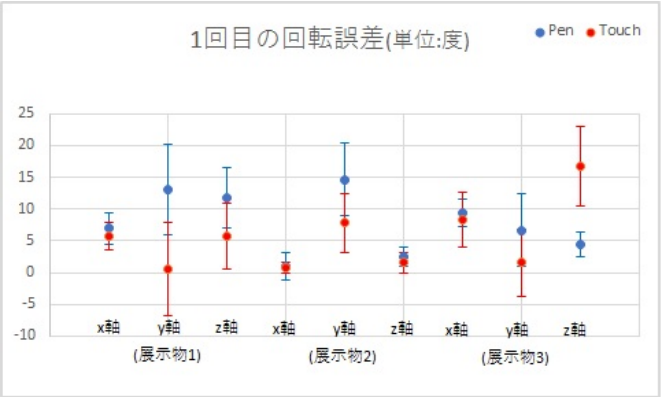

図 40: 1 回目の 3 つの展示物の回転角度誤差

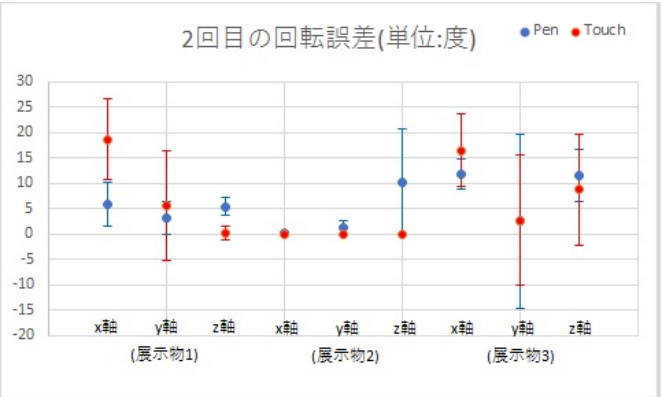

図 41: 2 回目の 3 つの展示物の回転角度誤差

表 10: 操作方法別の座標誤差 (単位: cm)

	展示物 1	展示物 2	展示物 3
	x 座標 (標準誤差) z 座標 (標準誤差)	x 座標 (標準誤差) z 座標 (標準誤差)	x 座標 (標準誤差) z 座標 (標準誤差)
1 回目マルチタッチ (被験者 5 名)	3.39(2.96) 0.79(0.67)	3.07(2.83) 1.32(1.73)	1.89(1.62) 1.08 (1.18)
1 回目マルチタッチ (被験者 4 名)	0.43(0.29) 0.14(0.17)	0.24(0.05) 0.40(0.16)	0.28(0.10) 0.10 (0.08)
1 回目本システム (被験者 5 名)	3.20(2.63) 1.098(0.58)	3.54(3.41) 1.26(1.48)	1.81(1.56) 0.99 (1.07)
1 回目本システム (被験者 4 名)	0.58(0.11) 0.53(0.14)	0.14(0.13) 0.14(0.34)	0.24(0.07) 0.08(0.13)
2 回目マルチタッチ (被験者 5 名)	0.08(0.18) 0.16(0.19)	0.11(0.10) 0.07(0.08)	0.14(0.08) 0.42(0.18)
2 回目本システム (被験者 5 名)	0.34(0.31) 0.12(0.08)	0.05(0.15) 0.39(0.16)	2.37(2.12) 1.85(1.19)

表 11: 操作方法別の回転誤差 (標準誤差)

	展示物 1	展示物 2	展示物 3
	x 軸 (度) y 軸 (度) z 軸 (度)	x 軸 (度) y 軸 (度) z 軸 (度)	x 軸 (度) y 軸 (度) z 軸 (度)
1 回目マルチタッチ (被験者 5 名)	3.29(2.99) 31.75(31.78) 5.96(3.95)	0.69(0.69) 31.81(39.82) 1.22(1.22)	13.85(6.51) 29.44 (31.29) 9.37 (8.85)
1 回目マルチタッチ (被験者 4 名)	5.73(2.22) 0.48(7.35) 5.72(5.08)	0.86(0.86) 7.85(4.66) 1.53(1.53)	8.25(4.28) 1.58 (5.32) 16.77 (6.28)
1 回目本システム (被験者 5 名)	6.46(1.97) 22.79(36.21) 8.18(5.11)	0.61(1.75) 12.06(27.04) 0.77(2.03)	8.70(1.77) 36.78 (30.43) 0.99 (3.77)
1 回目本システム (被験者 4 名)	6.90(2.47) 13.00(7.11) 11.75(4.73)	0.97(2.21) 14.62(5.73) 2.45(1.47)	9.40(2.09) 6.68(5.81) 4.46 (1.92)
2 回目マルチタッチ (被験者 5 名)	11.26(7.99) 5.64(10.80) 0.22(1.45)	0(0) 0(0) 0(0)	16.59(7.26) 2.76(12.77) 8.84 (10.92)
2 回目本システム (被験者 5 名)	24.02(4.36) 3.22(3.29) 5.49(1.70)	0.25(0.25) 1.27(1.27) 10.36(10.36)	11.90(2.93) 2.56(17.23) 11.58(5.14)

7.1.4 考察

操作方法別の完成時間比較 (表 9, 図 37) より, 本システムのほうが操作に必要な時間は短かった. この理由として, 本システムのほうが 3 軸同時に操作が可能である点が挙げられる. マルチタッチ操作では, 同時に操作できるのは 2 軸までであり, 3 軸の操作が必要とされる場合にはおいてはマルチタッチ操作の方が早いと考えられる.

また, マルチタッチ操作のほうが短い時間で行える被験者もいた. その場合の被験者は, 実験手順 1 においてもマルチタッチ操作のほうが慣れるのが早かった.

座標誤差においては, 平均誤差では本システムの方が誤差が大きい結果となったが, 標準誤差ではあまり有意性は見られなかった. また, 展示物を移動する際に, ユーザーからの感想から, ペンタブレットの端に行くほどカーソルが本来の位置よりずれてしまい, 操作がしにくいといったことが挙げられたことより, 平均誤差が生じたと考えられる.

回転角度の誤差においては, 標準誤差ではあまり有意性は見られなかった. マルチタッチ操作においては y 軸の回転と xz 軸の回転で操作が異なるため, 1 回目の展示物 2 のような y 軸回転だけの場合は本システムよりも誤差が少ないのではと考えていたが, あまり差は見られなかった. また, 2 回目の展示物 1 のような, タッチペンをタブレットに並行に寝かせる操作の場合, タッチペンの先がタブレット端末より離れてしまうためマルチタッチに比べ座標や回転誤差が生じやすいのではないかと考えていたが, あまり差は見られなかった.

展示物 3 では, 人型モデルではなく, 犬型のモデルを用いたためタッチ操作での回転角度誤差が大きくなっている. この理由としては, 頭を回転軸の中心と設定していた人型とは異なり, 胴を回転軸の中心とおいてしまったためである. そのため苦戦する被験者が多く, 誤差も大きくなっていると考えられる.

7.2 外部発表時のデモンストレーションからのフィードバック

2017 年 9 月 15 日から 18 日にかけて開催されたエンターテイメントコンピューティング 2017(EC2017) へ参加し, 15 日にデモンストレーション発表を行った. その際のフィードバックを元に操作方法を修正した.

7.2.1 デモンストレーションの内容

液晶ペンタブレットによってコレクションケース内の展示物を操作するデモを行った. 操作モードに関しては, 4.5.3 の項目に記述した液晶ペンタブレットに圧力が感知されたときのみ展示物を移動する仕様 (図 26) であった. 鑑賞モードでは, Kinect から検出した人物の座標位置を仮想空間内でそのまま用いることによって, コレクションケースを眺め

る仕様であった.

7.2.2 評価結果

20 名程度のユーザーに体験を行ってもらった. 展示物の移動に関しては, 全員すぐに慣れて使用できていた. 展示物の回転操作に関しては, ペン先をタブレットに押し付けたまま, タブレットのボタンを押したまま回転させるのは難しく, ユーザーが操作に慣れるまでに時間がかかるだけではなく, 操作できないユーザーもいた.

7.2.3 EC2017 後の研究

液晶タブレットのボタンは使用せずに, ペンだけの操作に変更した. また, ペン先をタブレットに押し付けたままの操作は難しいため, 展示物を選択式にし, 選択後は押し付けずとも移動や回転を行えるように変更した.

8　まとめと今後の課題

本章では, 本研究の成果, 問題点, 今後の課題について述べる.

8.1　まとめ

本研究では, 個人で入手可能な液晶ペンタブレットと Kinect を用いて, 展示物のレイアウトが可能な仮想コレクションケースシステムを構築した. 具体的な項目ごとに達成した点と今後の課題について述べる.

8.2　ケース機能

達成した点

展示物を保管するための箱状の空間を仮想空間内で作り, 実空間でディスプレイに映すためのカメラを配置した.

今後の課題

視点の移動を行った際に, 手前の壁は非表示, 奥側の壁は表示するといったケース自体の表示切り替え機能が必要である.

8.3　展示物の移動と回転機能

達成した点

液晶ペンタブレットを用いて, 展示物の移動や回転が可能なシステムを構築した. ペンの動きと同期した展示物の移動回転方法を実装した.

今後の課題

本研究では, 高さ一定の面内での移動は可能なシステムとなっているが, 展示物の高さ自体は操作できない. この問題に対しては, 高さも取得可能なセンサーを使用する必要がある. 解決方法には 2 つの種類が挙げられる. 1 つ目は, LeapMotion などの手の位置を詳細に検出できるセンサーデバイスを用いて, デバイスを所持する手の位置を感知する方法である. 2 つ目は, 操作用デバイスに直接センサーを付け, デバイスの上下位置を感知する方法である.

8.4 展示物のポージング

達成した点

Kinect を用いて, ユーザーの骨格情報を取得することで, 展示物のポージング変更を可能にした. また, 音声認識でポージングの開始終了を行えるようにした.

今後の課題

Kinect によって, 検出できていない骨格部位がある際に, 誤認識してしまい, ポーズが想定したものと異なってしまうことがある. また, ポージングの際に Kinect の検出範囲まで下がらなければならない. また, Kinect の認識できる関節数だとより自然な人体ポージングには限界がある.

関節が曲がる方向や角度などの基準を決めておくことで, 想定外の方向へ曲がってしまうことを避けたり, 骨格部位を検出する際に前後数フレームの値を平均化するなどで自然なポーズになりやすくなると考えられる.

また, 第 6.2 節で作成したモデルのポージングも課題として挙げられる. Kinect から検出し, モデルへと反映する関節数を調整することで実装可能だと考えられる.

8.5 視点移動機能

達成した点

Kinect を用いて, ユーザーを認識し, ユーザーの位置に応じた表示を実装した. また, 各モードに合わせた視点移動を実装した.

今後の課題

本研究では, 展示物を操作する際に便利な視点の移動を行ったが, 実際のコレクションケースを飾っているようなリアルな視点の移動機能は実装していない. 今後は操作モードと鑑賞モードでカメラの移動方法を大きく変える必要がある.

参考文献

[1] 国土交通省,“平成 28 年度　住宅経済関連データ - 国土交通省”(オンライン),
http://www.mlit.go.jp/statistics/details/t-jutaku-2_tk_000002.html
(最終アクセス　2018 年 1 月 15 日).

[2] 株式会社マイクロリサーチ,“REAL AQUARIUM(バーチャル熱帯魚)｜株式会社マイクロリサーチ”(オンライン),
http://www.mrl.co.jp/product/realaqua/index.html
(最終アクセス　2017 年 7 月 26 日).

[3] UX-Design Tokyo,“フィギュコレ”(オンライン),
http://www.uxdesigntokyo.jp/apps/figcol/figcol.html
(最終アクセス　2017 年 7 月 26 日).

[4] Itiro Siio, Jim Rowan, Elizabeth Mynatt:“’Peek-A-Drawer: Communication by Furniture”, Computing Systems, pp. 582-583, Minneapolis, 2002/4/20

[5] 神澤俊:“裸眼 3 D ディスプレイを使った仮想コレクションケース”, 電気通信大学, 情報工学科成見研究室卒業研究 (2012).

[6] 新井百越:“仮想コレクションケースの立体感と UI の改良”, 電気通信大学, 情報・通信工学科成見研究室卒業研究 (2014).

[7] Senarath C.M.:“「仮想窓」用高精度カメラシステムの開発”, 電気通信大学, 情報・通信工学科成見研究室卒業研究 (2011).

[8] 株式会社スリーディー,“3 次元力覚入出力デバイス Geomagic Phantom ＜ジオマジック ファントム＞ — 株式会社スリーディー”(オンライン),
http://www.ddd.co.jp/phantom/
(最終アクセス　2017 年 7 月 26 日).

[9] Motoki Takagi, Jumpei Arata, Akihito Sano and Hideo Fujimoto, “A new encounter type haptic device with an actively driven pen-tablet LCD panel”, International Conference on Robotics and Biomimetics, 2011/12/7-11.

[10] 小池英樹: “実世界インタフェースの新たな展開: 5. テーブルトップインタフェース.” 情報処理 51.7, 789-794, 2010.

[11] 杉浦裕太、尉林暉、チョントビー他:“Dollhouse VR: 複数人が空間を多角的に見ながら協調してレイアウトを検討できるシステム”, 情報処理学会インタラクション 2016,2016/3/4.

[12] zSpace 社,“zSpace 株式会社 -Official Home Page”(オンライン),
http://jp.zspace.com/
(最終アクセス　2018 年 1 月 15 日).

[13] 益岡了、尾崎洋、牧山臨　他:“ポインティングディバイスの特性比較”, 日本デザイン学会　第 58 回研究発表大会,2011/6/5.

[14] Microsoft, “Kinect - Windows アプリの開発”(オンライン),
https://developer.microsoft.com/ja-jp/windows/kinect
(最終アクセス　2018 年 1 月 2 日).

[15] Microsoft,“Xbox One S Kinect アダプター — Xbox”(オンライン),
https://www.xbox.com/ja-JP/xbox-one/accessories/kinect/kinect-adapter
(最終アクセス　2018 年 1 月 15 日).

[16] Microsoft, “The Kinect for Windows v2 sensor and free SDK 2.0 public preview are here Kinect for Windows Product Blog”(オンライン),
https://blogs.msdn.microsoft.com/kinectforwindows/2014/07/15/the-kinect-for-windows-v2-sensor-and-free-sdk-2-0-public-preview-are-here/
(最終アクセス　2018 年 1 月 13 日).

[17] Microsoft, “Kinect for Windows に関してよく寄せられる質問 - Windows アプリ開発”(オンライン),
https://developer.microsoft.com/ja-jp/windows/kinect/faq
(最終アクセス　2018 年 1 月 15 日).

[18] “Kinect v1 と Kinect v2 の仕様比較”(オンライン),
http://www.buildinsider.net/small/kinectv2cpp/01
(最終アクセス　2018 年 1 月 3 日).

[19] Wacom,“Cintiq 13HD —ワコムタブレットサイト— Wacom”(オンライン),
https://tablet.wacom.co.jp/article/cintiq-13hd
(最終アクセス　2017 年 7 月 7 日).

[20] Wacom,“Wacom｜ペンタブレット製品オプション”(オンライン),
https://tablet.wacom.co.jp/products/products_option20.html
(最終アクセス　2018 年 1 月 2 日).

[21] Unity, “Unity”(オンライン),
https://unity3d.com/jp
(最終アクセス　2018 年 1 月 15 日).

[22] 凹,“凹み Tips”(オンライン),
http://tips.hecomi.com/?page=1501426018
(最終アクセス　2018 年 1 月 2 日).

[23] Unity,“Kinect v2 Examples with MS-SDK - アセットストア”(オンライン),
https://www.assetstore.unity3d.com/jp/#!/content/18708
(最終アクセス　2018 年 1 月 2 日).

[24] Microsoft,“Download Microsoft Speech Platform - Runtime (Version 11) from Official Microsoft Download Center”(オンライン),
https://www.microsoft.com/en-us/download/details.aspx?id=27225
(最終アクセス　2018 年 1 月 15 日).

[25] Microsoft,“Download Microsoft Speech Platform - Software Development Kit (SDK) (Version 11) from Official Microsoft Download Center”(オンライン),
https://www.microsoft.com/en-us/download/details.aspx?id=27226
(最終アクセス　2018 年 1 月 2 日).

[26] Microsoft,“Download Kinect for Windows SDK 2.0 Language Packs from Official Microsoft Download Center”(オンライン),
https://www.microsoft.com/en-us/download/details.aspx?id=43662
(最終アクセス　2018 年 1 月 2 日).

[27] iTSeez3D,“#1 Mobile 3D Scanner App for iPad — itSeez3D”(オンライン),
https://itseez3d.com/
(最終アクセス　2018 年 1 月 3 日).

[28] “Blender.jp”(オンライン),
https://blender.jp/
(最終アクセス　2018 年 1 月 4 日).

謝辞

本研究を進めるにあたり, ご指導いただいた主指導教員の成見哲教授, および副指導教員の寺田実准教授に感謝いたします.