

拡張現実感を用いた仮想立体の仮配置システム

小林 智 輝^{†1} 田 中 二 郎^{†2}

我々が普段生活しているとき、違う場所にある物の大きさを比較したい場合がある。例えば、家の空いているスペースに合う家具を買いたい場合や、家の車庫に自動車が入るかどうかを知りたい場合などである。大きさを比較したいとき、実際に並べて置いてみると大きさの違いがわかりやすい。しかし同じ場所にない物を並べて大きさを比較することはできない。そこで、現実には存在しない仮想の立体を実物の代わりに仮配置し、試し置きすることによって実際に大きさを比較することができると筆者は考える。

本研究では仮想の立体を目の前の実世界に重畳表示し、試し置きを行うシステムを提案する。システムのユーザは仮想立体を作成し、試し置きを行う。

A Virtual Rectangular Solid Fitting System by Augmented Reality

TOMOKI KOBAYASHI^{†1} and JIRO TANAKA^{†2}

In regular life, there are situations when people want to compare sizes of distant objects. For example, people may want to buy furniture and they need to know whether it fits the available space in their house. They may also want to know whether a certain car fits into their garage. If objects are placed in front of them, people can easily understand the difference between the sizes of objects. But this cannot be done with distant objects. Therefore the authors believe that people can compare sizes of distant objects if they place a virtual object in the space needed instead of a real object.

In this paper, we propose a system that places a virtual object in front of a user. The user creates a virtual object and then tries to place it in a given space.

^{†1} 筑波大学システム情報工学研究科コンピュータサイエンス専攻
Department of Computer Science, University of Tsukuba

^{†2} 筑波大学システム情報系
Faculty of Engineering, Information and Syatems, University of Tsukuba

1. はじめに

1.1 背景

我々が普段生活しているとき、物の大きさを比較したい場合がある。例えば、家の空いているスペースに合う冷蔵庫を買いたい場合や、家の車庫に自動車が入るかどうかを知りたい場合などである。物の大きさを比較しようとする場合、比較したい物を仮配置し、試し置きして並べてみると大きさの違いがわかりやすい。試し置きには以下のような特徴がある。尚、試し置きとは目の前に物を試しに置き、周りの物やスペースと大きさを比較することを指す。

大きさの違いが目に見える

試し置きを行った後に、様々な角度から見ることによってどの方向にどの程度の大きさの違いがあるかを確認できる。

他の情報を得ることができる

実際に物を動かすことで、大きさの比較をするだけでなく長さを測っただけではわかり辛い他の情報も得ることができる。例えば、冷蔵庫を玄関からスペースまで持っていく際にぶつかって邪魔になる場所があるか等である。

1.2 試し置きの問題点

実物体を試し置きすることで大きさを比較する場合、以下のような問題が起こる場合がある。

比較したい物が手元にない

例えば家の車庫に自動車が入るかどうかを知りたい場合、家にいる場合は自動車が、自動車販売店にいる場合は車庫がそれぞれ手元にない。このように違う場所の物の大きさを比較したい場合、比較したい物の少なくとも1つは手元にない。そのため、このような場合実物体を試し置きして大きさを比較することはできない。

比較したい物を動かすことができない

例えば模様替えてタンスを空いているスペースに移すことができるかどうかを知りたい場合、タンスは重く、動かし辛い物である。このような重い物は、動かすために人手が必要であったり、道具が必要であったりと、動かすことが容易でない。また、空きスペースのような実体がない物は動かすことができない。このような物に対して実物体を試し置きして大きさを比較するという方法をとるのは困難である。

1.3 本研究の目的

本研究では、実物体を試し置きして大きさを比較する際発生する、比較したい物が手元がない・動かすことができないという問題点を解決する試し置きシステムを提案し、実現することを目的とする。

1.4 本研究のアプローチ

本研究では、仮想立体を用いて試し置きを行うシステム¹⁾²⁾を提案する。仮想立体は現実には存在しない物体なので、重さはない。よって、冷蔵庫のように本来は重く動かすことが困難な物であっても、仮想立体を代用すれば容易に試し置きができると考えられる。仮想立体を用いて試し置きを行うため、仮想立体を実世界に表示させる必要がある。そこで、本研究では拡張現実感技術を用いて仮想立体を実世界に重畳表示させる。

拡張現実感とは、現実の環境から知覚に与えられる情報に、コンピュータが作り出した情報を重畳表示させる技術である。近年、拡張現実感を用いた様々な研究がされている³⁾⁴⁾。本研究では現実の環境の情報である実世界の画像に、コンピュータが作り出す仮想立体を重畳表示させるために用いる。

1.5 本論文の構成

本章では実物体を並べて大きさを比較する際に起こりうる問題について挙げ、それを解決できるような試し置きシステムを作成するアプローチについて述べた。第2章では、試し置きシステムの要件を挙げ、その要件を満たすアプローチについて述べる。その後、システムの利用シナリオについて述べる。第3章では作成したプロトタイプシステムの使用方法について述べ、第4章では実装について詳細に述べる。第5章では関連研究について述べ、第6章ではシステムの試用から得られた知見や、そこから考えられる改善案、発展について議論する。最後に第7章で結論を述べる。

2. 仮想立体の試し置きシステム

2.1 システムの要件

本研究では仮想立体を用いて試し置きを行うシステムを提案する。まず、このシステムを作成する際の要件を述べる。

任意の大きさ・場所で仮想立体を作成できる

試し置きを行いたい物はテレビや冷蔵庫、自動車などが考えられる。これらの物の大きさは様々である。よって試し置きを行いたい場合、ユーザが任意の大きさの代わりとなる仮想立体を作成できる必要がある。

また、試し置きを行いたい場所は、家や家具販売店など様々である。よって、仮想立体の作成には持ち運べないような装置を使用せず、任意の場所で仮想立体を作成できる必要がある。

ユーザが自由に仮想立体を動かすことができる

試し置きを行う場合、大きさを比較するために仮想立体と実物体の辺を合わせることが考えられる。その際ユーザは何回か仮想立体の位置を微調整する必要がある。よって、ユーザが気軽に仮想立体を動かすことができるように、操作を容易にする必要がある。

また、試し置きの特徴として、実際に動かすことにより冷蔵庫を玄関からスペースまで持っていく際に邪魔になる場所などの、長さを測っただけではわかり辛い情報も得ることができる、というものがある。これを実現するために試し置きを行う仮想立体はユーザが自由に動かすことができる必要がある。

目の前の実世界を見ながら利用できる

試し置きを行う場所は実世界であり、仮想立体の比較対象である実物体やスペースは実世界に存在する物である。よって、システムを利用する際に実世界を見ることができなければならない。

2.2 仮想立体の作成・操作手段の検討

仮想物体を操作する手法については多くの研究がなされている。具体的には専用のデバイスを用いる方法や素手を用いる方法があり、これら2つの方法のうちどちらを用いるかを要件を踏まえて検討する。

専用のデバイスを用いる方法

専用のデバイスを用いることによって、視覚フィードバック以外にもデバイスに固有のフィードバックを実現できるという利点がある。一方、デバイスの形状によりユーザは制約を受ける。また、操作を行うために専用のデバイスを持ち歩く必要がある。具体例としてパドル状のデバイスを用いて仮想物体の操作を行う方法がある⁵⁾。

素手を用いる方法

デバイスを手に持ったり、手に装着する必要があるため、ユーザは操作の際に制約を受けない。また、操作を行うために専用のデバイスを持ち歩く必要がない。具体例として素手を用いてマウスにおけるクリックや選択操作を行う方法がある⁶⁾。

要件より、仮想立体は自由に動かせる必要があるが、素手を用いた方がデバイスを用いるよりも自由度が高い。また、試し置きは一般的に手で物体を置くことが多い。よって、素手

を用いた操作の方がユーザにとって自然である。以上より、本システムの仮想物体の作成・操作手段には素手による方法を採用する。

2.3 システム概要

システムの要件を満たすような仮想立体の試し置きシステムの概要は以下のようになる。

システムのユーザは仮想立体を作成し、実世界に重畳表示させる。その仮想立体を操作することで試し置きを行う。この仮想立体は自由に保存・出現が可能である。仮想立体の作成・操作は検討より、素手を用いて行う。ユーザには目の前の実世界とそこに重畳表示された仮想立体を提示する。この時画像を表示するデバイスはユーザの動きに追従する物を使用する。これにより、目の前の実世界を見ながらシステムを利用できるという要件を満たす。

2.4 想定する未来環境

ユーザの動きに追従する画像提示を行うデバイスの1つに、ヘッドマウントディスプレイがある。現在、ヘッドマウントディスプレイは小型・軽量化・低価格化・デザインなどが不十分であり、一般にはあまり普及していない。しかし、本研究ではヘッドマウントディスプレイにカメラやコンピュータが内蔵され、さらに広く普及し、使用することが一般的となっているような近未来環境を想定し、システムに使用する。

2.5 システムの特徴・利点

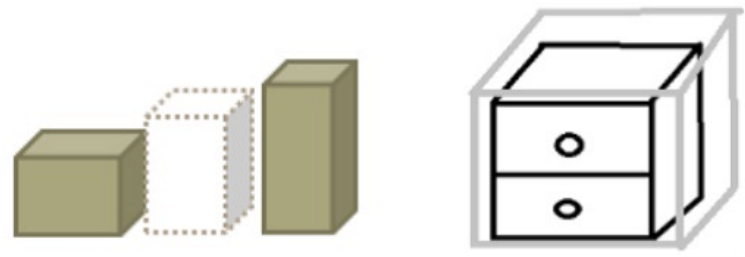
システムのユーザは任意の場所に任意の大きさの仮想立体を作成し、試し置きを行うことができる。これにより、手元にない物を仮想立体で代用することによって、同じ場所にない物の大きさを比較することができる。また、スペースのような実体がない物も仮想立体として視覚化し、試し置きの対象とすることができる。重さのない仮想立体を用いるため、大きさを比較したい物が重く動かすことが困難な物であっても問題なく試し置きを行うことができる。また、現実には存在しない仮想立体を用いることを生かし、実物体同士の場合は不可能な、重ねることによる大きさの比較も可能になる。

2.6 利用シナリオ

本システムで可能になる試し置きは大きく分けて2種類ある。

- 実世界の空いているスペースへの仮想立体の試し置き (図1(a))
- 実物体に仮想立体を重ねるように試し置き (図1(b))

実世界の空いているスペースへの仮想立体の試し置きは、主に手元にない物体と目の前のスペースとの大きさの比較に用いる。実物体に仮想立体を重ねるように行う試し置きは、主に目の前の物体と離れた場所の物体やスペースとの大きさの比較に用いる。これら2種類の試し置きそれぞれについて考えられる利用シナリオを示す。



(a) 空いているスペースに試し置き (b) 実物体に重ねるように試し置き

図1 試し置きの種類

2.6.1 実世界の空いているスペースへの仮想立体の試し置き

Aさんはテレビを持っていなかったので、テレビを買いたいと思った。そこでAさんはショッピングサイトを見てテレビを探した。しかし、家の空いているスペースにどのテレビが入るかわからなかった。そこで、試し置きシステムを用いてテレビと同じ大きさの仮想立体を試し置きして、家の空いているスペースにテレビが入るかどうかを調べた。このようにして、Aさんは家の空いているスペースに合ったテレビを選び、買うことができた。

このシナリオのように大きさを比較したい物が手元になくても、試し置きをすることで並べて大きさを比較することができる。

2.6.2 実物体に仮想立体を重ねるように試し置き

Bさんは最近持っている服が増えてきたので、部屋の空きスペースに合うようなタンスを買いたいと思った。そこで試し置きシステムを用いて部屋の空きスペースと同じ大きさの仮想立体を作成・保存し、家具店へ向かった。そして家具店において生成した立体を出現させてタンスに重ねるように置き、部屋の空きスペースに入るかどうかを調べた。部屋の空きスペースに合うタンスが見つかったので、それを購入した。

このシナリオのように、部屋の空きスペースのような実体のない物と、タンスのような実物体の大きさを、重ねることにより比較することができる。

3. 試し置きシステムのプロトタイプ

仮想立体の試し置きシステムのプロトタイプを作成した。ユーザはヘッドマウントディスプレイを装着する。ユーザに目の前の実世界を提示できるように、ヘッドマウントディスプレイにカメラを取り付け、カメラの映像をヘッドマウントディスプレイに表示する。今回、ユーザが作成し試し置きを行う立体としては直方体を採用した。なぜなら、試し置きを行いたい物として考えられる実物体(家具や自動車など)やスペースは外接する直方体によって表すことができる場合が多いからである。

3.1 仮想立体とのインタラクション

3.1.1 仮想立体の作成

空間の任意の場所に仮想立体を作成するために、仮想立体を作成する位置を指定する必要がある。本研究では一定時間空間を指さすことによってユーザは仮想立体を作成する位置を指定する。この時、指定した位置がユーザにわかるようにその位置に仮想の球を作成する。空間を指さすことによって位置を指定する際、位置の決定までに一定時間の猶予を設けることで、ユーザが間違った位置に仮想立体を作成する可能性を低くする。具体的な仮想立体の作成の手順は以下に示す。

ユーザはまず、一定時間空間を指さすことによってその場所に小さな仮想の球を作成する。この時、最初は指先の動きに従って緑色の球が表示される。その後一定時間空間を指さすことにより緑色の球が赤色の球になり、位置が固定される(図2)。作成した球は手を広げた状態でかざすことにより、手の位置に移動させることができる(図3)。赤色の球を2つ作成した時に、図(4)のように2つの球を1番遠い頂点とする直方体を作成する(図5)。

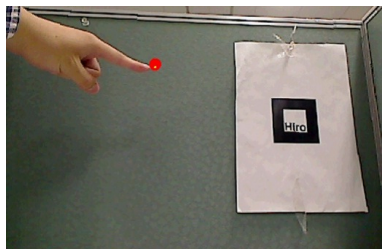


図2 仮想の球の作成

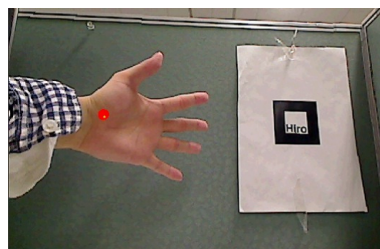


図3 作成した球の移動

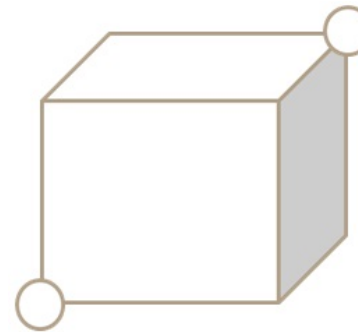


図4 2つの球を1番遠い頂点とする直方体

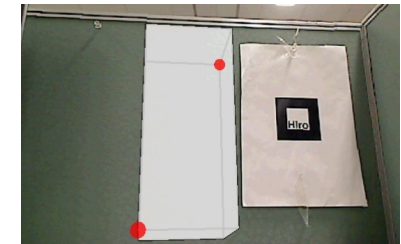


図5 作成された直方体

3.1.2 仮想立体の操作

作成した仮想の直方体に対しては以下の4つの操作を行うことができる。

- 移動操作
- リサイズ操作
- 保存操作
- 出現操作

以下にそれぞれの操作の方法を述べる。

移動操作

手を広げてかざし動かすと、仮想立体が手の動きに合わせて動く(図6, 図7)。目的の位置まで移動させた後、手を握ることによって移動が終了する。移動操作時、ユーザに立体を操作していることを示すために仮想立体の枠の色を変えている。

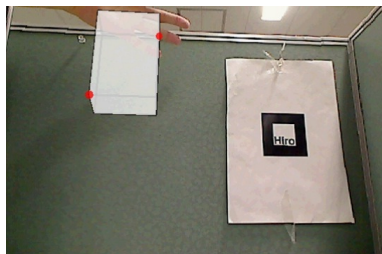


図 6 移動操作前

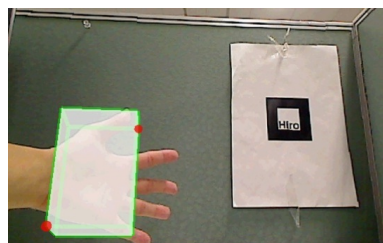


図 7 移動操作後

リサイズ操作

指を 2 本出した状態で手をかざすと、仮想の球が手の動きに合わせて動き (図 8, 図 9), それに伴い直方体の頂点が移動し、直方体がリサイズされる。

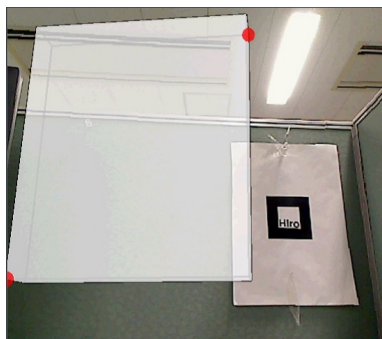


図 8 リサイズ操作前

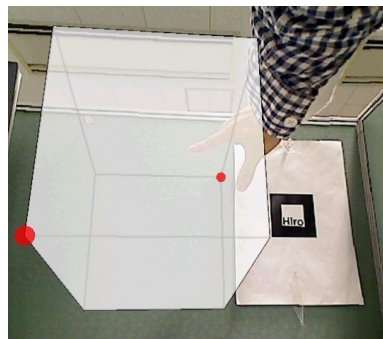


図 9 リサイズ操作後

保存操作

仮想立体を一定時間指さすことによって仮想立体を目の前から消し、保存を行う (図 10, 図 11)。仮想立体を指さしてから実際に保存操作が行われるまで、保存操作が行われることを示すために移動操作の時と同様に仮想立体の枠の色を変える (図 10)。これは仮想の球を作成する時と同様に、保存完了まで一定時間の猶予をもたせている。

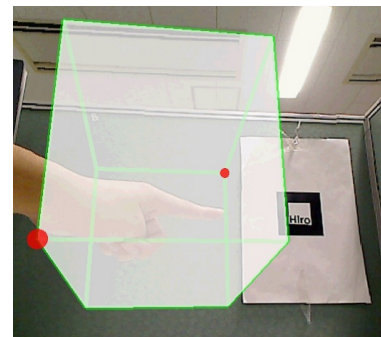


図 10 保存操作

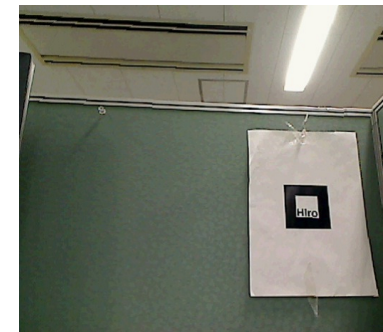


図 11 保存操作後

出現操作

保存を行っている状態において空間に手を広げてかざすことによって保存した立体の出現を行う。出現させた立体は保存前と同様に操作を行うことができる。

4. システムの実装

4.1 開発環境とシステム構成

開発言語は C++, 開発環境には Visual Studio 2008 を使用した。OS は Windows Vista, CPU は Intel Core 2 Duo 3.00GHz である。ヘッドマウントディスプレイには eMagin 社^{*1} の Z800 3DVISOR を使用した。また、ヘッドマウントディスプレイには USB カメラを 2 つ取り付け (図 12)。ユーザはこれを装着する。USB カメラには Logicoool 社^{*2} の 2-MP Portable Webcam C905m を用いた。さらにハードウェアとして計算機を用いた。

実装には 3 つのライブラリを使用した。画像解析には OpenCV^{*3}を用いた。実世界との対応付けのために AR ToolKit⁷⁾を用いた。USB カメラから得られた画像に仮想立体を重畳表示するために OpenGL^{*4}を用いた。

カメラから入力された画像は計算機に送られ、以下の処理を行う。

(1) 指先・手の重心の位置検出

*1 eMagin 社 (<http://www.emagin.com>)

*2 Logicoool 社 (<http://www.logicoool.co.jp/ja-jp>)

*3 OpenCV (<http://opencv.jp>)

*4 OpenGL (<http://www.opengl.org>)



図 12 USB カメラを取り付けたヘッドマウントディスプレイ

- (2) 手の動作識別
- (3) 指先、もしくは手の重心の 3 位置次元位置の計算
- (4) 3 次元位置情報の実世界への対応付け
- (5) 仮想立体の生成・移動・重量表示

カメラから入力された画像と、重畳表示させた仮想立体はヘッドマウントディスプレイに提示する。

4.2 指先・手の重心の位置検出

ヘッドマウントディスプレイの左右に取り付けたカメラの画像から指先の認識及び画像上の位置の検出を以下のように行う。

- (1) 入力画像からの手の領域・輪郭の検出
- (2) 指先の認識・座標の記録

4.2.1 入力画像からの手の領域・輪郭の検出

まず、カメラからの入力画像の肌色領域を認識し、肌色領域を表す 2 値化画像を作成する。本研究ではカメラから得られた RGB 表色系の画像を YUV 表色系に変換し、閾値を定めることにより肌色領域を検出した。尚、本研究で用いた YUV 表色系は淵ら⁸⁾ が用いた擬似的な YUV 表色系であり、Y に 16, U と V にそれぞれ 128 を加えることで一般的な YUV 表色系となる (擬似的な YUV 表色系であるため、以下の式では Y' , U' , V' と表す)。RGB 表色系から擬似的な YUV 表色系に変換する式を式 (1)、式 (2)、式 (3) に、肌色と認識する色の閾値を式 (4) に示す。

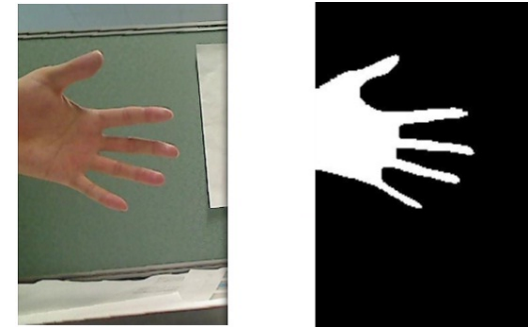
$$Y' = \frac{256 \times R + 504 \times G + 98 \times B}{1000} \quad (1)$$

$$U' = \frac{-148 \times R - 291 \times G + 439 \times B}{1000} \quad (2)$$

$$V' = \frac{439 \times R - 368 \times G - 71 \times B}{1000} \quad (3)$$

$$48 < Y' < 224 \quad -34 < U' < -3 \quad 12 < V' < 127 \quad (4)$$

作成した 2 値化画像はこのままでは手の領域以外の微小な肌色領域も検出している。そこで、ラベリングを行うことによってこの微小な領域を除去し、手の領域だけを検出した。ラベリングには井村が提供するラベリングクラス^{*1}を用いた。これを用いると、画像の連続している領域に面積が大きい順に領域番号を割り当てることができる。また、設定した大きさより小さい領域は取り除くことができる。このラベリングクラスを用いて微小な領域を除去し、面積が最大の肌色領域を識別する。肌色領域の中で、手の領域が最大の面積を持つと仮定し、肌色領域の中で面積が最大のものを手の領域として認識した。これにより手の領域のみを抽出した 2 値化画像ができ、この画像から手の輪郭の検出を行う。輪郭の検出には OpenCV を用いた。



(a) カメラからの入力画像 (b) 手の領域を検出した 2 値化画像

図 13 手の領域の検出

*1 ラベリングクラス (<http://oshiro.bpe.es.osaka-u.ac.jp/people/staff/imura/products/labeling>)

4.2.2 指先の認識・座標の記録

検出された手の輪郭から画像内における指先の位置を求める。手の輪郭の指先の位置を求める方法は文献⁹⁾を参考にした。まず、輪郭に沿って一定の間隔を開けて3点ずつ点を取り、座標を得る。この時、走査を開始する点を P_1 、 P_1 から一定の間隔はなれた点を P_2 、 P_2 からさらに一定の間隔はなれた点を P_3 とし、 P_1, P_2, P_3 のなす角を θ とする。この3点を用いたベクトルの内積・外積を用いて3点のなす角度 θ を計算する。式(6)を満たすとき θ が30度以下の凸点となる。

$$\sin\theta = \frac{P_2P_1 \times P_2P_3}{|P_2P_1||P_2P_3|} \quad \cos\theta = \frac{P_2P_1 \cdot P_2P_3}{|P_2P_1||P_2P_3|} \quad (5)$$

$$0 \leq \sin\theta \leq \frac{1}{2} \quad \frac{\sqrt{3}}{2} \leq \cos\theta \leq 1 \quad (6)$$

式(6)を満たした時、点 P_2 を指先と認識し、点 P_2 を2値化画像として記録する。この点を端点と呼ぶ。輪郭に沿って順番に角度を計算していくと連続的に条件を満たす端点が記録されていき、端点の集合ができる(図14(b))。その集合に対して先述したラベリングクラスを用いて重心を計算し、その重心の画像内における座標をリストに記録する。以上の処理を左右のカメラからの入力画像のそれぞれに行い、指の端点の画像内での座標をリストにすべて記録しておく。また、手の輪郭から、手の重心の画像内での座標を左右のカメラからの入力画像のそれぞれにおいて求め、記録しておく。手の重心の座標の検出にはOpenCVを用いた。



(a) 手の輪郭



(b) 端点の集合

図14 指の端点の集合

4.3 手の動作識別

本システムでは仮想立体の作成・操作に3種類の手の動作を用いる。まず空間を1本の指で指さす動作、次に指を2本出した状態で手をかざす動作、最後に手を広げてかざす動作である。そこでユーザがどの動作を行っているか識別する必要がある。本研究では左カメラ画像と右カメラ画像のそれぞれから作成した指先の座標を記録したリストのサイズを用いて手の動作を識別した。1本の指で指さす動作を行っている時、ユーザから見える指先は1つである。よって、2つのリストのサイズが1の時は1本の指で指さす動作をしていると認識する。同様に2つのリストのサイズが2の時は指を2本出した状態で手をかざす動作をしていると認識し、2つのリストのサイズが5以上の時は手を広げてかざす動作をしていると認識する。

4.4 指先、もしくは手の重心の3次元位置の計算

指先、手の重心の座標は既に求めたが、この座標は入力画像上の座標で2次元である。そこで、2つのカメラから求めた2つの座標情報から3次元位置を計算する。本研究ではステレオ法を用いて3次元位置を計算した。ステレオ法とは、三角測量の原理を応用し、測定物体を異なる位置から撮影して得た複数の画像から、測定物体の3次元位置を求める方法である。この方法を用いて2つのカメラの midpoint を原点とした座標系での3次元座標を求めた。ステレオ法を用いて3次元座標を求める方法は文献¹⁰⁾を参考にした。左右のカメラからの入力画像上での座標を (x_l, y_l) 、 (x_r, y_r) とし、カメラの焦点距離を f 、左右2つのカメラ間の距離を D とする。これらを用いると2つのカメラの midpoint を原点とした座標系における3次元座標 (X, Y, Z) は式(7)で求めることができる。

$$Z = \frac{B \cdot f}{x_l - x_r} \quad X = Z \cdot \frac{x_l + x_r}{2} \quad Y = Z \cdot \frac{y_l + y_r}{2} \quad (7)$$

この式を用いて2次元座標から3次元座標を計算する。手の動作識別において1本の指で指さす動作をしていると認識されている場合は指先の、手を広げてかざす動作をしていると認識されている場合は手の重心の3次元座標をそれぞれ計算する。

4.5 3次元位置情報の実世界への対応付け

前述したように求めた3次元座標はカメラの midpoint を原点とした座標系である。カメラはユーザの動きに追従するので、この3次元座標を用いて仮想立体を作成すると仮想立体がユーザに追従して動いてしまう。そこで、本研究ではARToolKitを用いて実世界への対応付けを行うことによって仮想立体が実世界の定点に表示されるようにした。

AR ToolKit は黒枠で囲まれた矩形(以下、AR マーカと呼ぶ。)を認識し、カメラを原

点とする座標系 (以下、カメラ座標系と呼ぶ) における AR マーカの位置・姿勢といった情報を得ることができる。あらかじめ設定を行ってあれば AR マーカの矩形の中に書く物には制限はない。カメラ座標系における座標を、AR マーカを原点とした座標系 (以下、マーカ座標系と呼ぶ) における座標に変換することで 3 次元座標の実世界への対応付けを行う。ARToolKit を用いることによって、マーカの中心座標 (X_m, Y_m, Z_m) をカメラ座標系における座標 (X_c, Y_c, Z_c) に変換する座標変換行列を得ることができる。式 (8) の座標変換行列における、 $r_1 \sim r_9$ は回転成分を表しており、 t_x, t_y, t_z はそれぞれ並進成分を表している。式 (8) を式 (9) のように簡略化すると、カメラ座標系における座標をマーカ座標系に変換する数式は、式 (10) のようになる。すでに求めてある 3 次元座標を、この方法でマーカ座標系に変換し、変換先の座標に仮想立体を作成する。

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (8)$$

$$A = T \cdot B \quad (9)$$

$$T^{-1} \cdot A = B \quad (10)$$

4.6 仮想立体の生成・移動・重畳表示

4.6.1 仮想の球の作成・移動

一定時間空間を指さすことで仮想の球を作成する機能を実現するために、1 本の指で指さす動作を行った回数を記録するカウンタを用意した。1 本の指で指さす動作を行っていると認識された場合には指先の座標に緑の球を出現させ、カウンタの数字を増やす。1 本の指で指さす動作を行っていないと認識された場合はカウンタの数字を初期の値に戻す。カウンタの値が一定値を超えたときにその時の指先の座標に赤い球を出現させ、座標を記録する。その後、手を広げてかざす動作を認識した場合、作成した球の座標を手の重心の座標に変更する。以上のようにして仮想の球の作成・移動機能を実現している。

4.6.2 仮想の直方体の作成

仮想の球を 2 つ作成した時、球の座標を基に直方体を作成する。1 回目に作成した球の座標を (a_x, a_y, a_z) 、2 回目に作成した球の座標を (b_x, b_y, b_z) とすると直方体の各頂点の座標は図 15 のようになる。仮想の直方体の頂点の座標はすべて記録しておき、仮想の直方体は操作が行われるまで記録された座標に表示させる。以上のようにして 2 回空間を指さすこと

で仮想の直方体を作成する機能を実現している。

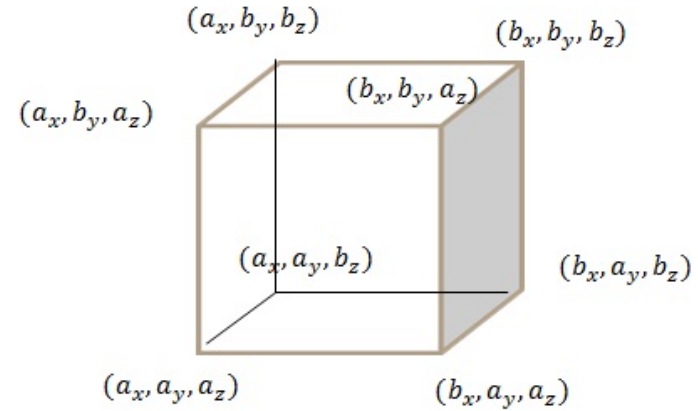


図 15 仮想の直方体の各頂点の座標

4.6.3 仮想の直方体の操作

移動操作

仮想の直方体が表示されている時に手を広げてかざす動作を認識した場合、図 16 のように手の重心の位置にユーザ側の面の中心がくるように仮想の直方体の移動を行う。

このように移動させるために、以下のように頂点の座標を変更する。表示されている直方体の縦、横、奥行のそれぞれの長さを求める (それぞれの長さを l_y, l_x, l_z とする)。そして手の重心の座標を (g_x, g_y, g_z) とおくと移動後の直方体の 1 番遠い 2 つの頂点 $A(a_x, a_y, a_z)$ 、 $B(b_x, b_y, b_z)$ の座標は式 (11)、式 (12) のように計算できる。

$$a_x = g_x - \frac{l_x}{2} \quad a_y = g_y - \frac{l_y}{2} \quad a_z = g_z \quad (11)$$

$$b_x = g_x + \frac{l_x}{2} \quad b_y = g_y + \frac{l_y}{2} \quad b_z = g_z - l_z \quad (12)$$

計算した 2 つの点の座標を基に、仮想の直方体の作成の時と同じ方法で各頂点の座標を計算し、求めた座標に仮想の直方体を表示させる。

リサイズ操作

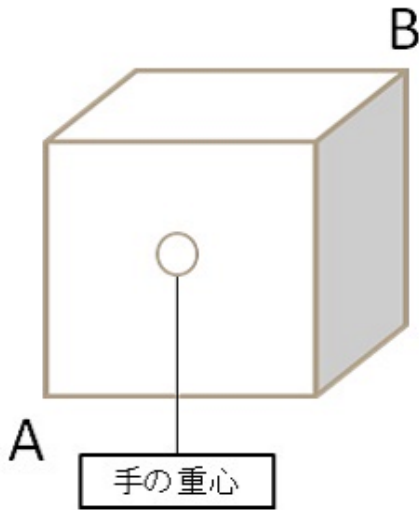


図 16 移動操作後の仮想の直方体の位置

仮想の直方体が表示されている時に指を 2 本出した状態で手をかざす動作を認識した場合、手の重心と 2 つの仮想の球との距離をそれぞれ計算する。その後、手の重心からの距離が短い方の仮想の球が表示されている直方体の頂点の座標を手の重心の座標へ変更する。

保存操作

仮想の直方体の保存操作時の処理は以下のように行う。仮想の直方体が表示されている状態で以下の 2 つの条件を満たしたとき、仮想の直方体を指さしていると認識する。

- 1 本の指で指さす動作を認識している
- 指先の座標が仮想の直方体の内部にある

1 本の指で指さす動作の認識はすでに行っているため、指先の座標が仮想の直方体の内部にあるかどうかの判定を行う。式 (13)、式 (14)、式 (15) を全て満たす時、指先の座標 (f_x, f_y, f_z) は直方体の内部にあると認識する (直方体の 1 番遠い 2 つの頂点の座標を (a_x, a_y, a_z) , (b_x, b_y, b_z) とし、縦、横、奥行のそれぞれの長さを l_y, l_x, l_z とする)。

$$|a_x - f_x| < l_x \quad |b_x - f_x| < l_x \quad (13)$$

$$|a_y - f_y| < l_y \quad |b_y - f_y| < l_y \quad (14)$$

$$|a_z - f_z| < l_z \quad |b_z - f_z| < l_z \quad (15)$$

仮想の球の作成時と同様にカウンタを用意し、仮想の直方体を指さしていると認識したらカウンタの値を増やす。仮想の直方体を指さしていないと認識したらカウンタの値を初期値に戻す。そして、カウンタの値が一定値を超えたときに、座標を記録したまま仮想の直方体の描画処理のみをやめることで仮想の直方体の保存を行う。

出現操作

保存が行われている時に手を広げてかざす動作を行ったりと認識された場合、記録されている各頂点の座標を基に再び仮想の直方体の描画処理を行うことによって保存した直方体の出現操作を実現している。

以上のように仮想の直方体への各操作を実現している。

5. 関連研究

5.1 仮想物体の操作

舟橋らや友添らは仮想空間において仮想立体を仮想の手によって操作する研究を行った¹¹⁾¹²⁾。この仮想の手は実際の手の動きに従って動く。仮想立体を操作するという点で本研究と関連がある。本研究は仮想の手でなく実際の手を用いて直接仮想の立体を操作する点、仮想空間ではなく実空間に重畳表示された仮想立体を操作の対象としている点でこれらの研究とは異なる。

笠原は実世界と仮想物体を重ねた仮想空間を構築する研究を行った¹³⁾。実世界画像に仮想物体を重畳表示することが可能な点で本研究と関連がある。仮想物体の操作はマウスを用いて 2 次元画面上で行う。本研究は特別なデバイスを使わずに実世界で仮想物体に直接触れるように操作する点においてこの研究とは異なる。

5.2 仮想物体の作成

Lau らは写真に直接スケッチすることにより立体を描きモデリングを行う研究を行った¹⁴⁾。実世界の映像を基に仮想の立体を作成する点で本研究と関連がある。この研究では仮想の立体の作成に専用の GUI を用いる。本研究では仮想の立体の作成にハンドジェスチャを用いる点で異なる。

椎尾らは専用のデバイスを用いて空間上に仮想のメモ書きを行うシステム「空気ペン」を開発した¹⁵⁾。空間の任意の場所に仮想物体を作成できる点で本研究と関連がある。本研究では、作成する対象が立体である点、また特別なデバイスを用いずに素手を用いて仮想物体を作成している点でこの研究とは異なる。

6. 予備的な評価

今回作成したプロトタイプシステムの試用を行った。以下に試用から得られた知見や考えられるシステムの発展、課題点について述べる。

6.1 研究の目的に対する知見

この試用により、今回作成したプロトタイプシステムを用いることにより試し置きを行う上での問題を解決できると筆者は考えた。以下に理由を示す。

まず、同じ場所がない物を並べて大きさを比較する際に、比較したい物の少なくとも1つは手元がないために実際に並べることができないという問題があった。この問題についてはシステムを用いて手元がない物を仮想立体で代用することによって、実際に並べることができ、解決できる。

次に、重い物の大きさを比較したい場合にその重さにより動かし辛く、実際に並べて大きさを比較することが困難であるという問題があった。この問題についてもシステムを用いて重い物を仮想立体で代用することで解決できる。なぜなら、仮想立体には重さがなく、動かすことが容易であるからである。

6.2 課題と本システムの発展

一方、試用を通して問題を感じる点も何点かあった。以下にそれを示す。

試し置きした際に仮想立体が周りの物に接触しているかどうかわかりづらい場合があった。今回開発したシステムでは試し置きの結果は目視によって得る。システム側で周りの物に接触しているかどうかを判定できると良いと感じた。その方法としては試し置きを行う場所の周りの物もユーザが位置を指定し、仮想立体化するというものが考えられる。また、ある物体が手前にある物に隠されて見えなくなる隠ぺい関係の表現を実現することで、仮想立体が接触しているかどうかによりわかりやすくなると考えられる。

本研究ではARマーカによって仮想立体を実世界に対応付けている。そのためARマーカがヘッドマウントディスプレイに取り付けたカメラに映っていないと仮想立体が正しい場所に表示されない。試用を行った際、仮想立体を作成する時や試し置きを行う時手によってARマーカが隠されてしまい、試し置きがスムーズに行えない場合があった。この問題の対応策としては、ARマーカを使わないマーカレスARの技術を用いることが考えられる。

7. 結 論

本研究では拡張現実感を用いて実世界に重畳表示される仮想立体により試し置きを行う

システムを提案し、プロトタイプの実装を行った。空間を指さし、直方体の頂点を指定することによって直方体を作成し、手をかざして動かすことにより仮想立体を操作する。これにより同じ場所がない物の大きさを比較することが可能になる。仮想立体のため、重さはなく試し置きを容易に行える。また、仮想立体のリサイズ・保存・出現機能も実現した。

本システムでは仮想立体を試し置きの対象として採用している。そのため仮想立体であることを生かし、並べる以外にも重ねることにより大きさを比較できるようになると筆者は考える。

今後は、仮想立体と実物体の接触判定をシステム側において行うことなどを発展として考えている。また、本システムの評価を行い、評価に基づいた改善を行いたい。

参 考 文 献

- 1) 小林智輝. 拡張現実感により提示される仮想立体の試し置きシステム. 学士論文, 筑波大学情報学群情報科学類, 2012.
- 2) 小林智輝, 田中二郎. 拡張現実感を用いた仮想の立体による試し置きシステム. 全国大会講演論文集第74回平成24年(4), pp.313-314, 2012.
- 3) Azuma, R. A Survey of Augmented Reality. Teleoperators and Virtual Environment 6, 4, pp. 355-385, 1997.
- 4) Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., and MacIntyre, B. Recent Advances in Augmented Reality. IEEE Computer Graphics and Applications, Vol.21, pp.34-47, 2001.
- 5) 加藤 博一, Mark, B., Ivan, P., 鉄谷 信二, 橘 啓八郎. 拡張現実感技術を用いたタンジブルインタフェース. 芸術科学会論文誌, Vol. 1, No. 2, pp.97-104, 2002.
- 6) 蔵田武志, 大隈隆史, 興梠正克, 坂上勝彦. ハンドマウス: ビジュアルウェアラブルズが可能にする拡張現実環境に適したインターフェイス. 電子情報通信学会技術研究報告, PRMU Vol.100, No.565, pp.69-76, 2001.
- 7) 加藤博一. 拡張現実感システム構築ツール ARToolKit の開発. 電子情報通信学会技術研究報告 PRMU Vol.101, No.652, pp.79-86, 2002.
- 8) 淵一馬, 高橋伸, 田中二郎. ハンドジェスチャによる範囲選択手法を使った撮影システム. 全国大会講演論文集第70回平成20年(4), pp.221-222, 2008.
- 9) 加茂浩之, 田中二郎. ウェアラブル拡張現実感による情報端末の仮想化. マルチメディア, 分散, 協調とモバイル (DICO2011) シンポジウム, pp.1223-1233, 2011.
- 10) 神原誠之, 大隈隆史, 竹村治雄, 横矢直和. ビデオスルー型拡張現実感のための実時間ステレオ画像合成. 電子情報通信学会論文誌 D-II, Vol.J82-D-II, No.10, pp.1775-1783, 1999.
- 11) 舟橋健司, 安田孝美, 横井茂樹, 鳥脇純一郎. 3次元仮想空間における仮想手による物体操作モデルと一実現法. 電子情報通信学会論文誌 D-II, Vol.J81-D-II, No.5,

pp.822-831, 1998.

- 12) 友添雄亮, 町田貴史, 清川清, 竹村治雄. 遠隔仮想物体操作と仮想空間ナビゲーションのためのジェスチャを用いた統一的操作手法. 電子情報通信学会技術研究報告, IE, Vol.103, No.642, pp.35-40, 2004.
- 13) 笠原元. カラー画像と距離画像の融合によるパノラマ仮想空間の構築. 奈良先端科学技術大学院大学情報科学研究科修士論文, 修第 760 号, 1998.
- 14) Lau, M., 大河原昭, 三谷純, 五十嵐健夫. 写真の上にスケッチと制約を記入していくことによるモデリングシステム. 第 18 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS 2010), pp.47-52, 2010.
- 15) 椎尾一郎, 山本吉伸. コミュニケーションツールのための簡易型 AR システム. インタラクティブシステムとソフトウェア VIII, pp.117-124, 2000.