

平成23年度

筑波大学情報学群情報科学類

卒業研究論文

題目 拡張現実感を用いた
3D表示コミュニケーションシステム

主専攻 情報システム主専攻

著者 城崎 亮

指導教員 田中二郎 志築文太郎 高橋伸 三末和男

要 旨

遠隔コミュニケーションを支援するツールは、音声だけでなく映像を伴った通話をサポートしており、現在の自分の姿を相手へ伝えることができる。現在は平面の映像が主流であるが、3次元表示される方が、ジェスチャなどの会話を補助する情報が伝わりやすくなり、会話を円滑に行うことができると考えられる。

本研究では、遠隔コミュニケーションを支援するため、通話相手の姿を3次元表示するシステムを提案、実装した。本システムでは、通話相手の姿を3次元で見ることができるため、ジェスチャなどの音声以外の情報が相手に伝わりやすくなり、コミュニケーションを円滑に行うことができる。3次元モデルは正面だけでなく、斜めから見た部分なども再現しており、自分の視点の位置に対応した表示を行い、現実世界に近いインタラクションにしている。

目次

第1章	はじめに	1
1.1	背景	1
1.1.1	遠隔コミュニケーション	1
1.1.2	映像付き遠隔コミュニケーション	1
1.2	映像付き遠隔コミュニケーションの問題点	2
1.2.1	遠隔コミュニケーションの利用場面	2
1.2.2	問題点	2
1.3	本研究の目的	4
1.4	本研究のアプローチ	4
1.4.1	想定する未来環境	5
1.4.2	カメラ付きヘッドマウントディスプレイ	5
1.4.3	拡張現実感の利用	5
1.4.4	2台のカメラによる3次元モデル	5
1.5	本論文の構成	6
第2章	システム概要	7
2.1	特徴	7
2.2	拡張現実感を用いた表示	8
2.3	利用例	9
第3章	システム実装	10
3.1	開発環境とシステム構成	10
3.2	映像取得	10
3.2.1	depth camera について	11
3.2.2	OpenNI を利用した映像取得	14
3.3	端末間通信	14
3.3.1	TCP と UDP	14
3.3.2	送信側の処理	15
3.3.3	受信側の処理	15
3.4	3次元モデルの表示	16
3.4.1	3次元モデルの作成	16
3.4.2	Kinect を2台利用した3次元モデルの作成	18

3.4.3	拡張現実感を用いた表示	18
3.4.4	OpenGL による描画	20
3.5	利用方法	20
第 4 章	関連研究	21
4.1	コミュニケーション支援	21
4.2	3 次元モデルの作成	21
第 5 章	議論	22
第 6 章	結論	23
	謝辞	24
	参考文献	25

図目次

1.1	ミーティングのイメージ図	3
1.2	正面から見た画像	4
1.3	斜めから見た画像	4
2.1	右に覗き込んだときに見える映像	7
2.2	左に覗き込んだときに見える映像	7
2.3	現実世界に重畳表示	8
2.4	現実世界	8
3.1	システム概要図	11
3.2	手をかざした3次元モデル	12
3.3	元画像	12
3.4	2台目により補った画像	12
3.5	通常の画像	13
3.6	背景を切り取った画像	13
3.7	Kinectの配置	13
3.8	Kinectの座標値	15
3.9	ARマーカを基準とした座標系	16
3.10	2次元映像をマーカ上に表示	17
3.11	3次元モデルをマーカ上に表示	17
3.12	各座標の対応	19

第1章 はじめに

1.1 背景

遠隔地の相手と実際に会うことなくコミュニケーションをする，遠隔コミュニケーションを行うことが増えている．例えば，電話はその一つである．遠隔コミュニケーションの利用が増えている理由として，携帯電話やパーソナルコンピュータを利用し，様々な場所においてリアルタイムに遠隔コミュニケーションを行うことができるため，といったことが考えられる．

遠隔コミュニケーションを行う場面は様々であり，友人や家族とのコミュニケーションのように，プライベートな会話を行う場面や，あるグループ内でのミーティングのように，ビジネスとして行う場面があり，遠隔コミュニケーションは，我々の生活に欠かせないものとなっている．

1.1.1 遠隔コミュニケーション

遠隔コミュニケーションは相手と自分の距離に関係なく行うことができる．日本にいる人と海外にいる人の間で遠隔コミュニケーションを行うこともあれば，両者が同じ建物の異なる階にいる場合に行うこともある．

また，多人数が同時に遠隔コミュニケーションを行うことができる．この機能は，会議やミーティングといった場合に利用される．

1.1.2 映像付き遠隔コミュニケーション

カメラ付きの携帯電話や，Skype¹のように遠隔コミュニケーションを支援するツールには，リアルタイムに音声と映像を会話相手に伝えることができるビデオ通話がある．映像を会話相手に伝えることにより，話している内容が相手に伝わりやすくなる．たとえば，会話を行う場合，声だけでなくジェスチャなどにより補助情報を付加することがあり，会話相手の理解を正確に，かつ素早くしている．

ジェスチャ以外に，相手の表情なども大切であると考えられる．たとえば，話者の方言や声質により，会話相手は怒っているように聞こえてしまう場合がある．このような場合，同じ言葉だとしても，会話相手の受ける印象が変わってしまうことがある．このように少しの

¹Skype(<http://www.skype.com/>)

誤解が、後に大きな誤解へと変わってしまう。このような場合だとしても、話者の表情を見ることが出来れば、会話相手は誤解しにくくなる。

映像付きの遠隔コミュニケーションについての研究は盛んに行われている [1][2][3][4][5]。

1.2 映像付き遠隔コミュニケーションの問題点

映像付き遠隔コミュニケーションの問題点を上げるため、利用例として、筆者らの場合について述べる。

1.2.1 遠隔コミュニケーションの利用場面

筆者らは1週間毎にミーティングルームに集まり2時間程度のミーティングを行う。ミーティングルームは図 1.1 のような部屋である。

基本的には一箇所に集まるが、参加者の中には別の場所からビデオ通話を利用する人もいる。ビデオ通話を利用している人は、図中にあるように、ノート PC に映される。図中の上方にある大画面ディスプレイには資料などが映っている。ミーティングでは主に議論を行うため、椅子に座ったまま発言する形式により行なっている。プレゼンテーションなどと異なり、参加者全員が発言者となるためである。

話者は話しているときは、相手の顔を見て、相手が自分の話を聞いてくれているか、話が正しく伝わっているか、を確認しながら話すことが多い。相手があまり理解していないように見えれば、補足の説明を加えることがある。話者は一通り話すと、ビデオ通話を利用し参加しているユーザに、「ここまでの話は理解できましたか」とノート PC の画面を見ながら確認を行う。図中の下方や右の列に座っている人は、ノート PC の画面をのぞき込み、相手の様子を確認する。また、話の中では、「テーブルの上にカメラを置いて、プロジェクターはどの辺に置きましょうか？」など機材の配置などの話などを行うことがあり、そのようなときには手を使い、機材の位置関係を表すことがある。

1.2.2 問題点

このシナリオの中には次のような問題点がある。

1. 話者がビデオ通話を利用し参加しているユーザの顔を見ながら話すことができない。
2. 3 次元的な動作を伴うジェスチャが相手に伝わりにくい。

1 について、先ほどのシナリオの中では、一通り話し終えた後にしか、ビデオ通話を利用し参加している相手に確認しないため、話のどこがわかりにくかったのか、ということが話者には伝わりにくい。会話相手は、話を遮ることを避けるために、質問は話の後に行う人もいる。このとき、どこに誤解が生じているのかを探すため、話を戻し、少しずつ確認しながら、もう一度繰り返すことになる。このように、議論を円滑に進められない場合がある。

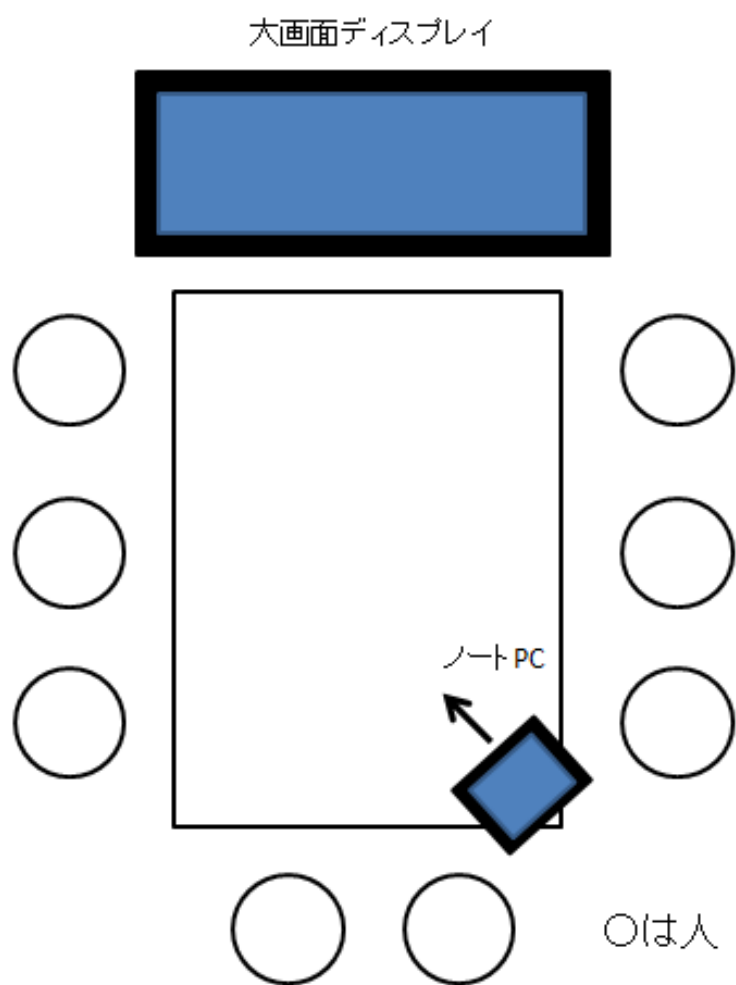


図 1.1: ミーティングのイメージ図

2 について，シナリオに出てくるような配置に関する話の場合，上下に設置するような機材もあるため，2 次元上に表すことはできない．遠近法のように大きさにより距離を表すこともできるが，それだけでは正確な情報を知ることは難しく，3 次元情報を相手に伝えることは困難である．

1.3 本研究の目的

本研究では，既存の映像付き遠隔コミュニケーションの問題点を解決し，遠隔コミュニケーションをより円滑に行うことを可能に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，映像付き遠隔コミュニケーションの問題点を解決するため，映像付き遠隔コミュニケーションを行う際に，会話を行なっているユーザへ通話相手の 3 次元モデルを提示することを提案する．

3 次元モデルを表示することにより，2 次元では表現できない，3 次元の動きを表現することができる．見ているユーザが少し右に移動したとすれば，相手の左側面が見えるようになる (図 1.2,1.3)．これは現実世界においては当然のことであるが，現在主流のビデオ通話では，



図 1.2: 正面から見た画像



図 1.3: 斜めから見た画像

ユーザが右に移動したとしてもディスプレイを斜めから見るだけであり，表示される映像に変化はなく，斜めから見るため見づらくなってしまふ．これを現実世界と同様に，コンピュータのディスプレイ上に表現することは難しい．その理由として，ユーザの顔の方向へディスプレイを動かすことが難しいためである．

本研究では，カメラ付きのヘッドマウントディスプレイを利用することにより，この問題を解決した．3 次元モデルの表示には拡張現実感を利用した．

1.4.1 想定する未来環境

本研究は、多くの人が日常的にヘッドマウントディスプレイを着用するような、ウェアラブルコンピューティングが一般化した近未来環境を想定している。既存のヘッドマウントディスプレイは、大きい、重い、高価である、といった問題点がある。しかし、近年、ヘッドマウントディスプレイの技術が進歩しており、小型化や低コスト化が進んでいる。筆者は小型化や低コスト化がさらに進み、デザインが向上し一般に普及すると考える。

また、最近では、テレビ、映画、ゲームなど、3次元映像を表示するものが増えている。映画やゲームはとても人気があり、今後ますます増えると思われる。3次元映像対応テレビに関しては、まだ一般家庭に普及しているとはいえずらい。原因としては高価であることや、専用の眼鏡を着用しなければならない、といったことが考えられる。しかし、これらの原因は製品の生産性の向上や、着用しても違和感がない眼鏡が登場することにより解決し、3次元映像が主流となると考える。

1.4.2 カメラ付きヘッドマウントディスプレイ

前節に述べたように、本システムではユーザはカメラ付きのヘッドマウントディスプレイを着用していることを前提としている。このヘッドマウントディスプレイには目線に近くなるようにカメラを取り付けるため、ヘッドマウントディスプレイを着用したとしても、現実世界を普段の目線に近い映像を見ることができる。本研究では、現実世界の映像に相手の3次元モデルを重畳表示させる。ヘッドマウントディスプレイを着用するため、ユーザはディスプレイを覗く必要はない。また、複数人が同時に利用する場合にも、各ユーザ毎にそれぞれ適した表示を行うことができる。

1.4.3 拡張現実感の利用

拡張現実感とは現実空間に情報を付加する技術である [6][7]。付加された情報は現実世界では見えないため、携帯端末のカメラを利用することにより見るといったものが多い。付加される情報はテキストや仮想の3次元オブジェクトなど様々である。本研究ではカメラ付きのヘッドマウントディスプレイを利用することにより、付加情報を見ることができるようにしている。

通話相手の3次元モデルを常に表示してしまうと、視界を遮り邪魔になることがある。そのため、拡張現実感を利用し、現実世界の一部と関連付けを行うことにより、表示する条件を設定する。現実世界の特定の位置に重畳表示することにより、その方向を向いている時のみ表示する。

1.4.4 2台のカメラによる3次元モデル

本研究では、通話相手の3次元モデルを現実世界に重畳表示するが、3次元モデルを作成するため、カメラを2台利用する。カメラに対して奥に移動するような動作だとしても、もう

1 台のカメラから見れば容易にわかる。

この映像をそのまま並べた場合、ユーザは2つの映像を見る必要があり、負担が大きくなってしまう。そのため、3次元モデルによる表示にし、見るユーザが自由に視点を移動できるようにする。一度に見える映像は一方向のみだが、視点を移動する前の映像と視点を移動したあとの映像の関係により、ユーザは3次元的な情報を理解しやすくなる。

1.5 本論文の構成

本章では、遠隔コミュニケーションについての有用性と問題点、研究の目的とアプローチについて述べた。第2章では今回作成したコミュニケーションシステムの概要について述べる。第3章ではシステムの実装について述べる。第4章では関連研究について述べる、第5章で試用から得られた知見や問題点について述べ、議論を行う。さらに、今後の課題と本システムの発展について述べる。最後に、第6章で結論を述べる。

第2章 システム概要

遠隔コミュニケーションを支援するため、通話中の相手の姿を3次元表示するシステムを作成した。本章では本システムの概要について述べる。

2.1 特徴

本システムでは遠隔地にいる相手の3次元モデルを見ることができる。この3次元モデルは相手の姿の前半分が再現される。そのため、正面だけでなく、斜めから見たとしても、相手の姿を見ることができる(図 2.1, 2.2)。また、このモデルは相手の動きをリアルタイムに再現するため、実際に会って会話を行なっているように、臨場感を感じることができる。3次元表示されるため、2次元の映像よりも多くの情報を得ることができる。



図 2.1: 右に覗き込んだときに見える映像



図 2.2: 左に覗き込んだときに見える映像

2.2 拡張現実感を用いた表示

本システムでは3次元表示する方法として、現実世界にいるかのように見せるため(図 2.3, 2.4), 拡張現実感 (AR) を採用した。



図 2.3: 現実世界に重畳表示



図 2.4: 現実世界

ユーザは、カメラ付きのヘッドマウントディスプレイを着用することにより、現実世界を背景とし、通話相手の3次元モデルを見ることができる。現実世界の一部に重畳表示することにより、実際に会って会話をしているように感じさせる。3次元表示されるため身振りや手振りが相手に伝わりやすくなる。3次元表示する方法として拡張現実感以外にも、仮想現実が考えられる。

仮想現実

仮想現実を利用する場合、人工的に作成した空間をユーザへ提示する。通話を行うユーザは全員仮想空間に没入し、現実とは異なる空間中において会話を行うことができる [3]。ヘッドマウントディスプレイを利用し、ユーザに仮想空間のみを見せれば、別の場所にいることに似た感覚を与えることができる。また、仮想空間であるため、必要なオブジェクトがあったとしても簡単に用意できる。しかし、仮想空間へ没入するため、現実空間と仮想空間の間にギャップが生じてしまう。例えば、仮想空間中を移動するためには何らかの入力装置が必要である。また、会話を紙にメモするためには現実世界を見る必要がある。

拡張現実感

拡張現実感を利用した場合、現実世界の一部に情報を付加することができ、付加情報を現実世界に重畳表示できるため、普段と同様の行動を行うことができる。

本研究では、コミュニケーションを自然に行うことができるようにするため、拡張現実感を採用した。

2.3 利用例

とある大学のとある研究室に所属する A 君は卒業条件を満たし、卒業を待つだけの日々を過ごしていた。そんなとき、研究室のメンバーによるお疲れ様会を行うことになり、A 君は一発芸をやらなければならなくなった。しかし、A 君にはそんなことをやった経験はなく、何をやっていいのかわからなかった。そこで、普段から面白い話をしている B 君に相談することにした。B 君は「それなら任せろ」と、相手に面白く見せるための、話し方や動きのコツを実演して教えてくれた。A 君が見よう見まねでやってみたところ、お世辞にも面白いとは言いがたいものだった。その後、A 君は練習を重ね、前日に B 君に確認してもらうことにした。しかし、B 君は別の用事のため、その日は研究室には行けなかった。そこで、本システムを利用し、B 君に見てもらうことにした。A 君の練習の成果もあり、前回よりとても良くなっていた。B 君は、「そこで腕をもっと開いて」「そこはもっと思い切りやって迫力を出す」などのさらに細かい部分のアドバイスをを行った。実際に会い話しているのではないが、本システムを利用することにより、動作を容易に伝えることができた。そのため B 君のアドバイスはより細かい部分にまで届き、A 君の芸は素晴らしいものとなった。結果、お疲れ様会では大成功をし、思い残すことなく卒業し、大学院へ入学したのだった。2 年後、A 君は修論という壁にぶつかるが、それはまた別の話である。

第3章 システム実装

3.1 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 2010 を使用した。OS は Windows7, CPU は Intel Core i3 CPU 540 3.07GHz 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, このヘッドマウントディスプレイには USB カメラを取り付けた。ヘッドマウントディスプレイに取り付ける USB カメラには Logicoool 社²の Logicoool HD Webcam C615 を利用した。

ユーザを撮影するための depth camera として Microsoft 社³の Kinect を 2 台用いた。

実装には 3 つのライブラリを使用した。kinect からの入力を受け取るために OpenNI⁴を用いた。実世界との対応付けを行うために ARToolKit⁵を用いた。USB カメラから得られた画像に 3 次元モデルを重畳表示するために OpneGL⁶を用いた。

図 3.1 に本システムの構成図を示す。この図は一方向しか表していないが, 実際には Kinect は 4 台になり, ヘッドマウントディスプレイは 2 台になり, 双方向に行う。

本システムは以下の 3 つの処理部からなる。

- 映像取得部: kinect カメラから映像の取得
- 端末間通信部: 2 台の端末間の通信
- 表示部: 拡張現実感を用いた 3 次元モデルの表示

3.2 映像取得

本システムではユーザを写すカメラとして, depth camera を利用する。より正確な 3 次元モデルの復元を行うため, depth camera を 2 台利用する。depth camera より取得した映像を相手の端末に送るために整理する。

¹eMagin 社 (<http://www.emagin.com/>)

²Logicoool 社 (<http://www.emagin.com/>)

³Microsoft 社 (<http://www.microsoft.com/>)

⁴OpenNI(<http://www.openni.org/>)

⁵AR ToolKit(<http://www.hitl.washington.edu/artoolkit/>)

⁶OpenGL(<http://www.opengl.org/>)

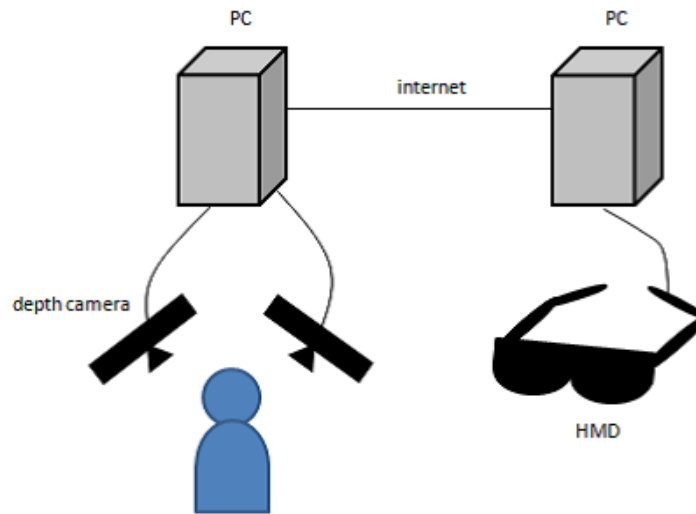


図 3.1: システム概要図

3.2.1 depth camera について

depth camera とは、通常のカメラが取得できる色情報に加え、各画素におけるカメラからの距離情報も取得できるカメラである [8]。

通常のカメラを使用した場合、カメラからの距離を測定するために、2 台以上のカメラが必要となる [9][10]。映像を 3 次元に復元するためには画素ごとの距離が必要であるため、2 台以上のカメラが必要となる。また、2 台のカメラを利用したとしても、両方のカメラに写っている部分の距離しか測定することはできない。例えば、片方のカメラには写っているが、もう片方には写っていない部分の距離は計算できない。

depth camera を利用する場合、1 台のみで距離の取得を行うことができるため、1 方向から見える部分の復元を行うことができる。しかし、1 台の場合、手の影になっている部分などの復元を行うことができない。この問題を解決するため 2 台利用する。2 台利用することにより、1 台では見えない部分の 3 次元復元を行うことができる。

3 次元表示する場合、見えない部分があるとどのような姿勢なのか、理解することは難しい (図 3.2, 図 3.3)。

この問題を解決するため、depth camera を 2 台利用し、1 台だけでは復元を行えない部分を補った。

また、カメラからの距離情報を用いることにより、カメラから遠い部分は背景として扱うことができ、被写体の特定を簡単に行うことができる、という利点もある (図 3.5, 図 3.6)。

本システムでは、kinect から約 1mm 離れている位置で利用すると仮定しているため、カメラからの距離が 2mm 以上離れている場合、相手には表示されないようにしている。

depth camera には入手が容易である Microsoft 社の Kinect を 2 台使用した。2 台の配置は 3 次元モデルの復元を簡単化するため、図 3.1 のようにした。

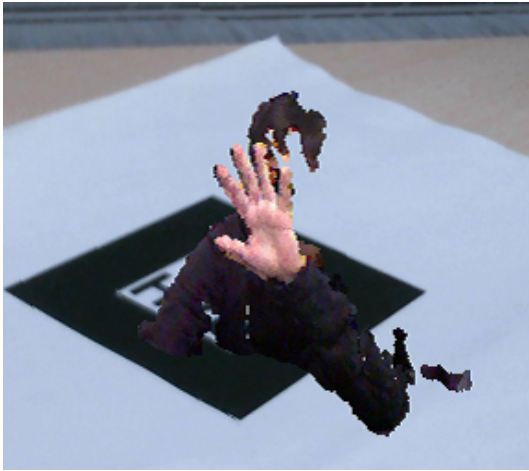


図 3.2: 手をかざした 3 次元モデル



図 3.3: 元画像



図 3.4: 2 台目により補った画像



図 3.5: 通常の画像

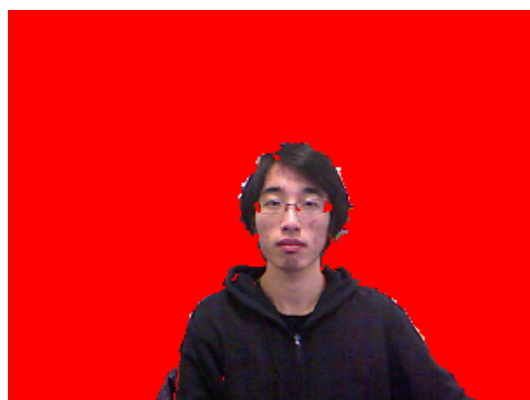


図 3.6: 背景を切り取った画像

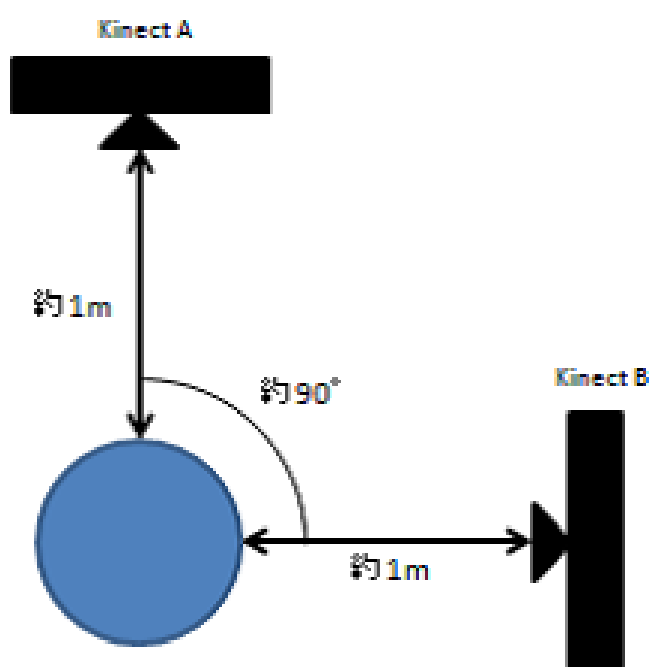


図 3.7: Kinect の配置

Kinect について

Kinect では 640×480 の解像度の映像を取得することができ、それぞれの画素において、色情報である RGB 値と距離を取得することができ、距離は 1mm 単位の精度で取得できる。

Kinect を利用するためには Microsoft 社の Kinect SDK を利用する方法と、OpenNI を利用する方法がある。2 つには異なる部分がいくつかあるため、どちらを利用するか用途により選択する必要がある。

Kinect SDK を利用した場合、1 台のカメラでしか距離を取得できず、2 台のカメラは色情報しか取得できない、という問題があった。OpenNI を利用した場合、両方のカメラから距離を取得できるため、本研究では OpenNI を利用した。

3.2.2 OpenNI を利用した映像取得

Kinect を扱うため OpenNI を利用した。実際に Kinect を利用するために行ったことを示す。まず、初期化を行う。OpenNI には、色情報を扱う ImageGenerator クラスと、距離情報を扱う DepthGenerator クラスが存在する。これらのクラスの GetMetaData メソッドを利用することにより、色と距離情報を取得することができる。Kinect には色情報を取得するカメラと距離情報を取得するカメラの 2 つが並んでいる。これらの 2 つが少しずれているため、色情報と距離情報も少しずれてしまう。DepthGenerator クラスの GetAlternativeViewPointCap クラスの SetViewPoint メソッドを利用することにより、2 つのズレを修正することができる。

これらのデータは 1 次元配列に格納されている。色情報は各色符号なし 8bit、距離情報は符号なし 16bit の値で取得することができる。本システムでは 2 台の Kinect を利用するため、初期化も 2 台分行う必要があった。

Kinect から映像情報を取得する際に、x 座標が右から左へと増加することに注意する (図 3.8)。多くの USB カメラは、x 座標が左から右へと増加するため、同じように扱うと逆になってしまう。

3.3 端末間通信

端末間通信は Windows 上においてソケット通信を行うための API である Winsock を使用した。送信する端末と受信する端末において実装したことを述べる。まず、通信プロトコルとして TCP と UDP がある。

3.3.1 TCP と UDP

TCP は通信を行うときにデータが届いたか確認が行われるため、データが相手に届くことが保証される。端末 A からデータを送ると、端末 B はデータを受信したことを示す情報を端末 A に送り返す。UDP はデータが届いたか確認などは行われないため、データが相手に届くことが保証されない。しかし、TCP よりも通信速度が早いというメリットがある。本システム

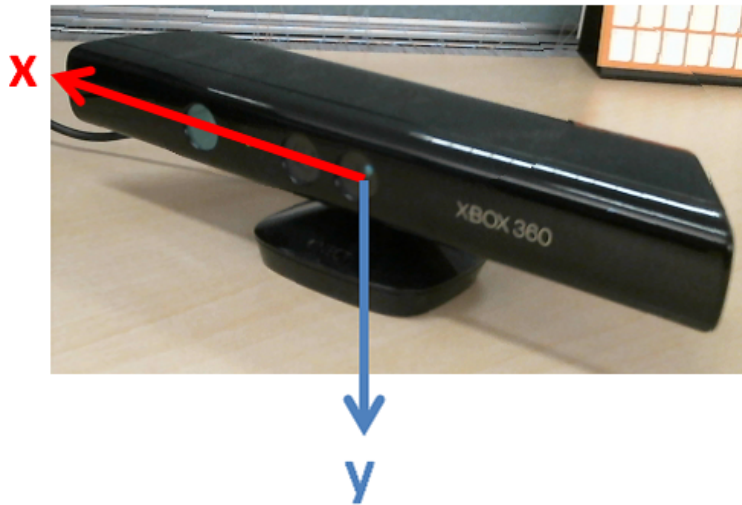


図 3.8: Kinect の座標値

では映像を扱うため、通信の信頼性よりも通信速度が重要であると考えられる。よって、UDP を利用した。

3.3.2 送信側の処理

送信側の端末は、Kinect から取得した映像情報を 1 行 (640pixel) ずつ読み込み、送信する。送信する 1 行のデータの内容は RGB 値 [3byte×640pixel]、距離情報 [2byte×640]、行番号 [2byte]、Kinect 番号 [1byte]、である。行番号は何行目を送っているかを表す。UDP を利用するため、抜けるデータが発生したり、送信した順番と受信した順番が異なる可能性がある。そのような場合においても、何行目かがわかり 3 次元モデルを綺麗に復元することができる。

Kinect 番号は 2 台ある Kinect のうちどちらから取得した情報かを表す。受信側の端末では、送られてきたデータが、2 台あるうちのどちらの Kinect から取得したデータなのかわからないため、識別するために付いている番号。どちらの Kinect から取得したデータかにより、表示するときの座標が異なるため必要である。

3.3.3 受信側の処理

受信側の端末は 1 行を受信し、行番号と Kinect 番号を基に、配列に保存する。これらのデータを基に定期的に描画を行い、表示を更新する。

このように 1 行ずつ送信する理由は、 640×480 のデータを全部まとめて送った場合、受信側の端末の描画処理が間に合わず、意図したとおりに表示できないためである。1 行ずつ処理を行うことにより、描画処理が間に合わなかったとしても、少し映像が乱れるだけに留めることができる。

3.4 3次元モデルの表示

本システムでは，ARToolKit[11]を使用し，3次元モデルを現実世界に重畳表示する．ARToolKitは，黒枠に囲まれたマーカーを利用することにより，容易に拡張現実感を利用することができるライブラリである．ARToolKitにより重畳表示を行う物体は，OpenGLを利用することにより自由に描画できる．

3.4.1 3次元モデルの作成

相手の端末から送られてきたデータを基に，3次元モデルの作成を行う．

受信側の端末が保持しているデータは Kinect から取得したものであり，これをマーカーを基準とした座標系 (図 3.9) に合わせ，3次元モデルを作成する．

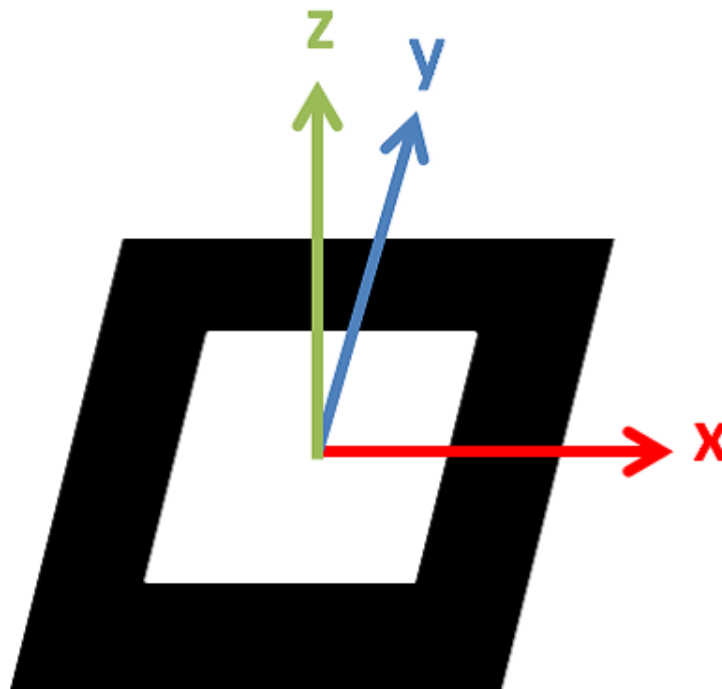


図 3.9: AR マーカを基準とした座標系

具体的に，保持しているデータの x 座標と y 座標をマーカーを起点とした座標系の x 座標と z 座標に割り当てる．Kinect から取得した映像の y 座標は上から下に増加するが，マーカーを起点とした座標に合わせるため反転させる．ここで行うと，マーカー上に平面の映像を表示することができる (図 3.10)．次に，Kinect からの距離情報をマーカーを起点とした z 座標に対応付けることにより，図 3.10 を 3次元にすることができる．距離情報は 1 mm 単位であるため，1画素あたりの大きさに揃える必要がある．実測により調査したところ，Kinect から 1 mm 離れた位置では，1画素あたり約 2 mm であった．



図 3.10: 2 次元映像をマーカ上に表示



図 3.11: 3 次元モデルをマーカ上に表示

3次元モデルを作成する際に、カメラにより取得できる映像は、カメラからの距離が近いものほど大きく映り、遠いものは小さく映る、ことに注意しなければならない。Kinectにおいても同じ事が起こるため、距離により大きさの調整を行った。実測により調査したところ、Kinectから1 m離れた位置を基準とすると、Kinectから0.8 m離れた位置では約1.22倍の大きさであった。Kinectから1.2 m離れた位置では約0.82倍の大きさであった。結果を見ると0.8 mから1.2 mの間では、0.2 m近づくに連れ、約1.22倍の大きさになっている。このデータを基に、1 m離れた位置を基準とし、近づく毎に小さくし、離れる毎に大きくした。

3.4.2 Kinectを2台利用した3次元モデルの作成

ここまで、Kinectを1台利用した場合について述べた。次に、Kinectを2台利用した場合について述べる。

基本は1台の時と同じである。異なる点は、2代目のKinectの座標のARマーカの座標への対応付けである。2台目のKinectでは、Kinectのx座標をARマーカを起点としたy座標に、Kinectからの距離をARマーカを起点としたx座標に対応させる。x座標は反転させる必要がある。Kinectのy座標をARマーカを起点としたy座標に対応させることは1台目と同じである。今回はKinectの配置を図3.7のように固定しているため、1台目のKinectのx座標と2台目のKinectからの距離情報が同じ座標に対応する(図3.12)。1台目のKinectからの距離情報と2台目のKinectのx座標も同様である。

2台のKinectから復元した3次元モデルを組み合わせることにより、1台の時より多くの部分を復元することができる。ARマーカの原点を中心として、2つの3次元モデルを組み合わせる。2つの3次元モデルを作成する際に、ARマーカの原点を中心とすることにより、1つの3次元モデルになるように重ねる。

3.4.3 拡張現実感を用いた表示

3次元モデルの表示にはARToolKitを使用した。ARToolKitは黒枠に囲まれたマーカを利用する。

ARToolKitは、拡張現実感を利用し、マーカに情報を付加し、容易に現実世界に付加情報の重畳表示を行うことができる。ARToolKitはマーカとカメラの相対位置を計算し、マーカを起点とした座標系上に付加情報の表示を行う。これにより、視点を変更すると3次元モデルの見え方も変わる。

ARToolKitは、マーカがカメラに写っていないならば利用できないが、拡張現実感を容易に利用できるという利点がある。

表示する3次元モデルはOpenGLにより描画を行う。

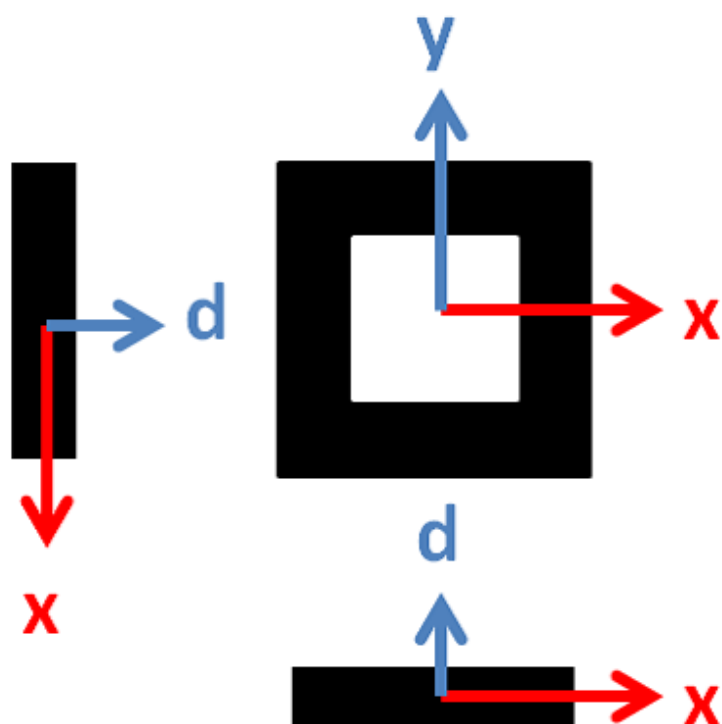


図 3.12: 各座標の対応

3.4.4 OpenGL による描画

3次元モデルはOpenGLを利用し描画を行う。OpenGLはグラフィック用のライブラリであり、直線や平面の描画などを容易に行うことができる。

OpenGLを利用し描画を行う際には、図3.9の座標系を使用する。3次元モデルを描画するため、各画素に小さな立方体を描画した。

図3.7のように配置を行ったため、マーカを基点としたx座標にはKinect Aのx座標とKinect Bの距離を、y座標にはKinect Aの距離とKinect Bのx座標を、z座標にはKinect Aのy座標とKinect Bのy座標を、それぞれ利用することができる。

3.5 利用方法

通話を行いたいユーザはdepth cameraを2台を設置し、カメラ付きのヘッドマウントディスプレイを装着する。このカメラの映像がヘッドマウントディスプレイに表示される。今回作成したシステムを起動すると、映像を表示するには受信用のアプリケーションを起動し、ヘッドマウントディスプレイに付いているカメラに、ARマーカを写すことにより、マーカの上に相手の姿が3次元表示される。通話を行うユーザが2人とも、送信用と受信用のアプリケーションを起動することにより、双方向に映像を送ることができる。移動して自分の視点を変更する、もしくは、マーカを移動させると、相手の姿の見え方が変わる。たとえば、右に覗き込むように見た場合、相手の左側が見える(図2.1)。逆に、左に覗き込むように見れば、相手の右側が見える(図2.2)。

カメラにARマーカが写っていない場合は相手の姿は表示されない。

第4章 関連研究

4.1 コミュニケーション支援

岡田は、協調作業支援における、コミュニケーションの重要性を述べており [2]、アバタを用いたコミュニケーション支援システムを作成した。岡田は、コミュニケーション中の、人の視線や口の動きなどの細かい動作が、コミュニケーションにおいて重要であると述べている。岡田のシステムでは、人の視線や口の動きなどの細かい動作に注意し、アバタを用いることでそれを表現している。

葛岡ら [4] や西岡 [5] らは、コミュニケーションにおいて、周囲の環境を共有することが重要であると述べており、遠隔地にいる2人のユーザが視界を共有する、コミュニケーション支援システムを作成した。

本研究では、ユーザのジェスチャのように、3次元的な動きを伝えることを目的としている点が異なる。

4.2 3次元モデルの作成

延原らは弾性メッシュを用いることにより、多視点画像から高精度の3次元形状復元を行っている [10]。通常のカメラを利用し、正確な3次元モデルを復元する場合、延原らが行っているような難しいものとなるが、本研究では depth camera を利用し、簡単化を行った。

國田らは多眼カメラを用いることにより、任意視点人物像を生成している [12]。3次元モデルを作成しているわけではないが、任意視点から見るができるようにするという点で共通している。

第5章 議論

今回提案した遠隔コミュニケーション支援システムを試用した。

まず、表示されたモデルについて述べる。今回作成したシステムでは、2台の Kinect を利用し、ユーザの3次元モデルを表示することができた。しかし、3次元モデルを表示する際に、各 Kinect からのデータを上手く組み合わせることが出来ず、つなぎ目を正確にに表示出来なかった。原因としては、Kinect を図 3.1 のように、理想的な状態にすることが難しいことが原因であると考えられる。また、実装の章で述べた縮尺について、本システムでは線形近似を行ったが、実際には異なるためズレが生じていると考えられる。これらの問題に関して、今回は計算の簡単化のために2台の Kinect を図 3.1 のように行ったが、実際に利用する場合にはこのように配置できるとは限らないため、起動時に2台の Kinect 間の相対位置の計算を行う必要があると考えている。2台の Kinect 間の回転と並進を利用すれば、現在のシステムよりも3次元モデルを正確に表示できると考えられる。

また、本システムでは Kinect を2台しか利用していないが、数を増やし、周囲を全部カバーできるように配置することにより、通話相手の背面なども表示できると考えられる。

次に、動作速度について述べる。Kinect に写っているユーザが動いてから、少しの遅延はあるが、ほぼリアルタイムに動作を相手へ伝えられた。しかし、描画処理に時間がかかってしまうため、連続的に動作の表示が行えておらず、パラパラ漫画のようにコマ送りのようになってしまった。この問題を解決するには、描画の改善が必要であると考えられる。現在は小さな立方体をいくつも描画しており、隣接した立方体の重なっている部分など無駄な描画も発生している。描画をもっと効率的に行う事が出来れば、スムーズに動作が表現されると考えられる。

次に、本システムの発展可能性について述べる。Kinect では写っている人物の骨格を取得することができる。この骨格情報を利用することにより、ハンドジェスチャを利用したインタラクションも考えられる。たとえば、手を振る動作をした場合は、通話を切るといったものである。このようにジェスチャを利用することにより、マウスやキーボードなどの入力装置が必要なくなり、カメラがあるだけで利用できると考えられる。

第6章 結論

本研究では，拡張現実感を用いることにより，通話相手の3次元モデルを表示するコミュニケーションシステムを作成した．

拡張現実感を利用し，3次元モデルを現実世界に重畳表示するため，広い角度から通話相手を見ることができるようになった．また，従来のビデオ通話では伝わりにくかった，奥行きのある動作に関して，通話相手が横を向かずに，見る側のユーザの視点を移動することにより，見ることができるようになった．

ユーザを写すカメラとして，カメラからの距離情報を取得できる depth camera を利用することにより，3次元モデルを容易に作成することができた．また，2台利用することにより，1台の時よりも，より多くの3次元モデルを作成することができた．

今後は，3次元モデルをより正確に表示すると共に，動作速度の向上についても取り組む必要があると考えている．

謝辞

本論文を執筆するにあたり、指導教員である田中二郎先生をはじめ、志築文太郎先生、三末和男先生、高橋伸先生には丁寧なご指導、有益なアドバイスを頂き、心から感謝を申し上げます。また、田中二郎先生にはテーマ選びから研究の進め方、論文執筆に至るまで細かいご指導を頂きました。深くお礼申し上げます。

インタラクティブプログラミング研究室の皆さまにはゼミや日頃の生活の中で、貴重な意見を頂き、大変お世話になりましたことをここに感謝致します。特に NERF チームの皆様には、テーマ選びの相談から本論文の推敲まで、面倒を見て下さったことに深く感謝申し上げます。

プログラムの不具合に悩んでいるときに助けてくださった加茂浩之先輩、大江龍人先輩には深く感謝しております。

最後に、私を支えて下さいました家族、友人、大学生活でお世話になった全ての方々に心より御礼申し上げます。ありがとうございました。

参考文献

- [1] 中西 英之, 吉田 力, 西村 俊和, 石田 亨 FreeWalk:3 次元仮想空間を用いた非形式的なコミュニケーションの支援. 情報処理学会論文誌 39. pp.1356-1364, 1998.
- [2] 岡田謙一. 協調作業におけるコミュニケーション支援. 電子情報通信学会誌. Vol.89 No.3 pp.213-217, 2006.
- [3] 松本 敏宏, 松浦 宣彦, 菅原 昌平, 正木 茂樹. デスクトップ VR 会議におけるアバタ表現方式の評価. 情報処理学会研究報告. [グループウェア] 99(88), pp.13-18, 1999.
- [4] 葛岡 英明, 山下 淳, 小山 慎哉, 山崎 敬一. 実世界を指向した遠隔コミュニケーション空間の開発. 日本バーチャルリアリティ学会サイバースペースと仮想都市研究会第 5 回シンポジウム「生活に浸透するサイバースペース」予稿集, pp.59-64, 2002.
- [5] 西川 敦, 貴島 茂雄, 加藤 朋宏, 丸谷 誠慶, 山口 哲, 宮崎 文夫. 実時間遠隔コミュニケーションを支援する相互視界共有システム. 電子情報通信学会論文誌. D-I, 情報・システム, I-情報処理, pp.292-304, 2005.
- [6] Ronald T.Azuma. A Survey of Augmented Reality. Teleoperators and Virtual Environments 6, pp.355-385, 1997.
- [7] 神原 誠之, 大隈 隆史, 竹村 治雄, 横矢 直和 ビデオシー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.1-9, 1999.
- [8] Andrew D.Wilson, Hrvoje Benko Combinig Multiple Depth Cameras and Projectors for Interactions On Above and Between Surfaces. UIST2010, pp.273-282, 2010.
- [9] 源田 大輔, 向川 康博, 尺長 健. 視点と法線を組み合わせた任意視点映像の生成法. 情報処理学会研究報告. CVIM, pp.53-60, 2003.
- [10] 延原 章平, 和田 俊和, 松山 隆司. 弾性メッシュモデルを用いた多視点画像からの高精度 3 次元形状復元. 情報処理学会論文誌 コンピュータビジョンとイメージメディア 43, pp.53-63, 2002.
- [11] 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, パターン認識・メディア理解, pp.79-86, 2002.

- [12] 國田 豊, 稲見 昌彦, 前田 太郎, 舘 . 多眼カメラを用いた任意視点人物像の実時間生成システム. 電子情報通信学会論文誌 D-II. vol. 84, no.1 pp.129-131, 2001.