Vol.2018-CVIM-210 No.29
2018/1/19

頭部搭載型プロジェクタによるARユーザインタフェース

見目 裕介 [*1]　　小室 孝 [*1]

AR User Interface using a Head-Mounted Projector

Yusuke Kemmoku[*1] and Takashi Komuro[*1]

Abstract – 本論文では、ユーザが頭部にプロジェクタとデプスカメラを装着し、平面上に投影された映像に対してタッチによる操作が可能なテーブルトップインタフェースを提案する。ユーザの頭の向きに合わせて投影する映像を変えることで疑似的に広い表示領域を実現することができる。システムの性能を評価するための実験を行い、実空間と投影映像の位置合わせの誤差は水平・垂直方向の平均が約 **10 mm**、システムの遅延は約 **0.2** 秒であることが分かった。

Keywords : augmented reality, peephole interaction, projector-camera system, depth camera

1 はじめに

近年、スマートフォンやタブレットなどのモバイル端末や、PC などの大きなディスプレイやキーボードを用いたシステムが広く普及している。モバイル端末は、持ち運びに適しているが、画面が小さいために表示領域や作業空間が狭いといった問題がある。また、大きなディスプレイやキーボードを用いたシステムは、操作性はよいが、持ち運びが難しいといった問題がある。そのため、持ち運びが容易で、広い表示領域と作業空間を実現するシステムが望まれる。

広い表示領域を実現する方法として、プロジェクタの利用が考えられる。プロジェクタを用いることで一度に広い領域に映像を表示することが可能である。プロジェクタを用いたユーザインタフェース（UI）の研究が存在する [1,2]。平面に投影された映像をタッチすることで、インタラクションを行うことができる。しかし、表示領域を広くするためには、投影平面とプロジェクタの距離を遠くにしなければいけないという、設置場所の制約が存在する。

一方これを解決する研究として、手持ちプロジェクタを用いた UI の研究 [3,4] が存在する。プロジェクタの向きに合わせて表示する映像を変えることにより、表示領域を拡張することができる。しかし、プロジェクタを持つために片手が塞がってしまうという問題や、必ずしも視線の先に映像が投影されないという問題がある。

また、頭部搭載型プロジェクタ（HMP）を用いた UI の研究も存在する [5,6]。これらの研究では、頭部にプロジェクタを装着するため、常に視線の先に映像を表示させることが可能である。しかし、広い領域に映像投影することを目的とした研究は存在しない。

そこで本研究では、ユーザの頭部に装着したプロジェクタでテーブルなどの平面上に映像を投影し、ユーザの頭の向きに合わせて映像を変えることで疑似的に広い表示領域を実現し、その映像に対してタッチ入力が可能なテーブルトップインタフェースを提案する。

なお、常に視線の先に映像を表示する方法として、ヘッドマウントディスプレイ（HMD）を用いるアプローチも考えられる [7,8]。HMD を装着することで、実空間に表示されている仮想物体を空中や平面上でインタラクションを行うことができる。しかし、HMD には装着の手間や見た目の抵抗感に加え、視界が装置で覆われることの不安感、目が疲れやすいといった欠点がある。

プロジェクタを頭部に搭載する場合も、装着の手間や見た目の問題はあるが、視界を覆わなくて済むという利点がある。また、HMD のように眼球の輻輳角と焦点距離の矛盾が生じないため、眼精疲労も抑えられる。しかしながら、映像の没入感や、作業内容を人に見られないという秘匿性については、HMD のほうが優れている。

2 頭部搭載型プロジェクタを用いた AR ユーザインタフェース

本論文では、ユーザが頭部にプロジェクタとデプスカメラを装着し、平面上に投影された映像に対してタッチによる操作が可能なテーブルトップインタフェースを提案する。ユーザは頭部に装置を装着し、椅子などに着座して、テーブルなどの平面に映像を投影する。このシステムには 2 つの特徴がある。第一に、プロジェクタを頭部に装着することにより、視線の先に映像を投影でき、図 1 のようにユーザの中心視野領域の

[*1]埼玉大学
[*1]Saitama University

© 2018 Information Processing Society of Japan

日本バーチャルリアリティ学会 複合現実感研究会 Vol.21, No.1, 2018

大部分にその映像が表示されるという点である。第二に、 頭の向きによって投影される映像を変えることで、平面上に映像を固定させることができるという点である。これらにより疑似的に大画面を実現する。また平面に重畳表示された映像とのインタラクションとして、デプスカメラを用いて平面へのタッチ検出を行うことで、平面全体をタッチパネル化する。

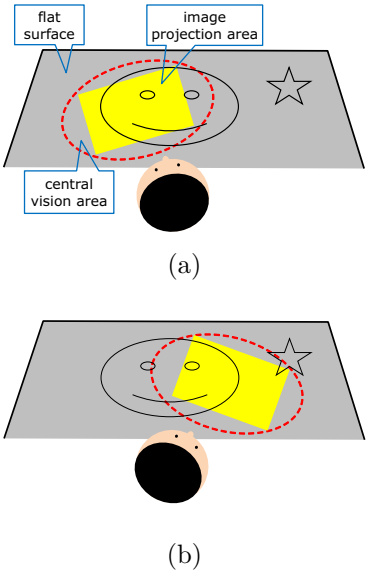


図 1　提案システムの動作イメージ

2.1　実験システム

モバイルプロジェクタとデプスカメラを用いて、図 2 に示すような実験システムを試作した。モバイルプロジェクタは Koolertron 社の Mini Micro 640×480 HD DLP Home Theater Projector を用いた。デプスカメラは、SoftKinetic 社の DepthSense 325 を用いた。これは、60 fps で 320×240 のデプスマップと赤外画像を取得でき、15 cm–100 cm の近距離の範囲で用いることができる。またこの試作システムでは、デプスカメラとプロジェクタはデスクトップ PC(CPU: Intel Core i7-4790 3.6 GHz, RAM: 8 GB) に接続しており、PC は演算に用いる。

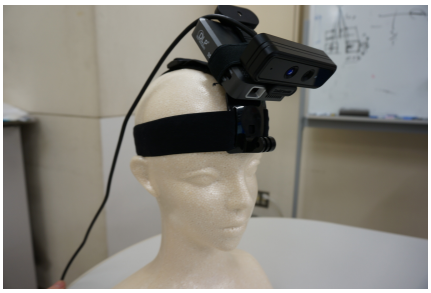


図 2　実験システム

画像処理には OpenCV、仮想物体の描画には OpenGL、3D 点群の処理には Point Cloud Library、デプスカメラから得たデプス画像の取得や、デプス画像を 3 次元点群に変換するために DepthSense SDK を利用した。

2.2　重畳表示手法

プロジェクタで投影した映像が正しい大きさや形で平面に表示されているように見えるようにするために、映像の重畳表示を行う。まず、デプスカメラを用いて実空間の 3 次元シーンを取得する（図 3 (a)）。次に、取得した 3 次元シーンから平面を推定し、その平面を仮想世界にも配置する（図 3 (b)）。そして、その平面上に重畳表示する映像を貼り付け、透視投影変換を行うことで、3 次元シーンを 2 次元画像に変換する（図 3 (c)）。このとき、透視投影変換を実行するときの視野角のパラメータは、プロジェクタの視野角と一致させる。そしてその 2 次元画像をプロジェクタで実空間の平面に投影する（図 3 (d)）。以上によって、平面に映像を重畳表示させることができる。

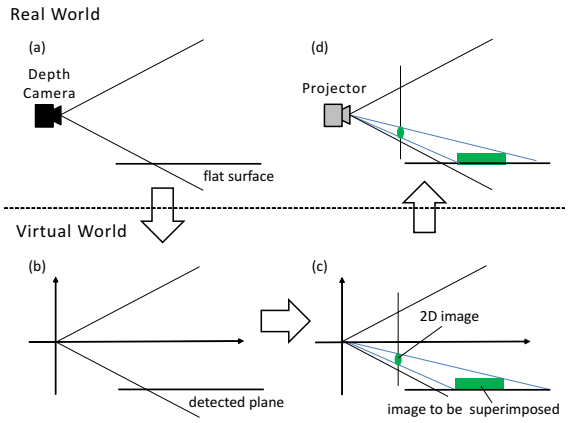
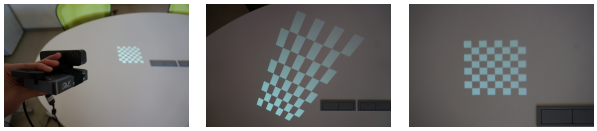


図 3　映像の重畳表示手法

この手法を用いて平面に映像の重畳表示を行った結果を図 4 に示す。図 4 (a) はプロジェクタと平面の位置関係を表した図である。図 4 (b) は重畳表示の手法を用いずに投影した結果である。図 4 (c) は重畳表示の手法を用いて投影した結果である。これらの図より、装置を傾けても同じ形状が保たれ、平面に重畳表示されていることが分かる。



(a)　(b)　(c)

図 4　平面に斜めに投影した映像

© 2018 Information Processing Society of Japan

2.3 位置合わせ手法

平面に映像を固定させるためには、実空間と映像の位置合わせが必要となる。位置合わせを行う手法として、三次元点群同士の位置合わせを行う方法や複数フレームで特徴点の対応付けを行う方法が存在する。しかし提案システムの使用場面として、テクスチャや凹凸の少ない平面を想定しているため、これらの方法を用いることはできない。

また、慣性センサを用いて位置合わせを行う方法もあるが、慣性センサとデプスカメラ間のキャリブレーションをする必要がある。デプスカメラから得られる実シーンのみを用いることで、装置間のキャリブレーションが不要となる。

そこで本研究では、三次元シーンから推定した平面の情報のみを用いて、頭部の姿勢推定を行い、実空間と映像の位置合わせを実現する。

本研究では頭部の位置、姿勢において次のような仮定をおく。提案システムの使用場面として、椅子などに着座した状態で机などの平面に映像投影する場合を想定している。そのため、頭部の動きをパンとチルトの 2 自由度のみと仮定する。そこでその 2 自由度を、慣性センサを用いずにデプスカメラから得られる実シーンのみを用いて求める。

頭部の姿勢推定を行うために、まず、デプスカメラから取得したデプス画像を 3 次元点群に変換し、RANSAC [9] を用いて平面推定を行うことで、平面の法線ベクトル (a, b, c) を得る。次に、推定した平面に重畳するための仮想物体をフィッティングさせる。法線ベクトル $(0, 1, 0)$ の仮想物体の平面を、推定した平面の法線ベクトル (a, b, c) と一致するような、x 軸周りの回転量 α と y 軸周りの回転量 β を求める。2 つの法線ベクトルを一致させるようにすると、

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = R_y(\beta)R_x(\alpha)\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \sin\alpha\sin\beta \\ \cos\alpha \\ \sin\alpha\cos\beta \end{pmatrix} \quad (1)$$

が成り立つ。これより α と β は、

$$\alpha = \arctan\left(\frac{\sqrt{a^2+b^2}}{b}\right) \quad (2)$$

$$\beta = \arctan\left(\frac{a}{c}\right) \quad (3)$$

と求まる。

頭部の姿勢推定を行うために、初めて平面が検出がされたときの頭部の姿勢を基準として、ある時刻 k における相対的な頭部の姿勢を求める。基準の姿勢からある時刻 k のときの頭部の回転行列 R_k は平面推定のときに求めた回転行列を用いて 、

$$R_k = (R_y(\beta)R_x(\alpha))^{-1} \quad (4)$$

と書ける。

また、頭部の回転中心と装置の位置が一致していないため、装置は回転による並進が起こる。この並進量を求める。頭部と装置の位置関係を図 5 に、時刻 0 とときの装置の姿勢から時刻 k のときの装置の姿勢の変化を図 6 に示す。

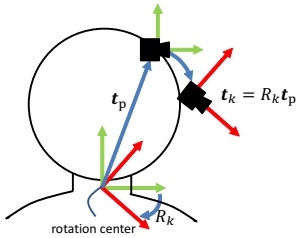


図 5 頭部回転による装置の姿勢

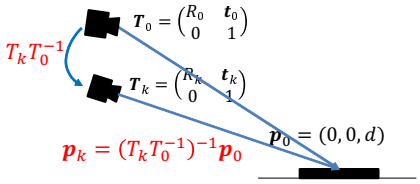


図 6 頭部回転による仮想物体の位置変化

まず、頭部の回転中心から装着時の装置の位置までの並進を $\boldsymbol{t}_p$ とする。そして、この位置から頭部の回転 R_k が起こったときのプロジェクタの位置 $\boldsymbol{t}_k$ は、

$$\boldsymbol{t}_k = R_k\boldsymbol{t}_p \quad (5)$$

となる。式 (4) と式 (5) より、回転中心からある時刻 k のときの変換行列 T_k は、

$$T_k = \begin{pmatrix} R_k & \boldsymbol{t}_k \\ 0 & 1 \end{pmatrix} \quad (6)$$

となる。

T_k と、初めて平面検出したときの姿勢 T_0 を用いて、仮想物体の位置を決定することができる。デプスカメラから平面までの距離を d とする。仮想物体の中心位置を $\boldsymbol{p}_0 = (0, 0, d)$ とすると、仮想物体の位置 $\boldsymbol{p}_k$ は、

$$\boldsymbol{p}_k = (T_kT_0)^{-1}\boldsymbol{p}_0 \quad (7)$$

となる。

重畳表示の手法と位置合わせの手法を用いて、ユーザが頭を動かした時にシステムが机に映像を投影した映像を図 7 に示す。

© 2018 Information Processing Society of Japan

日本バーチャルリアリティ学会 複合現実感研究会　Vol.21，No.1, 2018

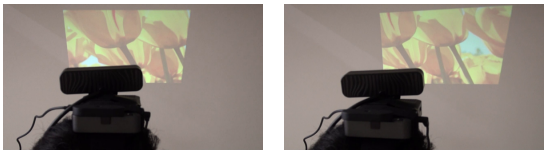


図 7　頭を動かしているときの位置合わせの結果

2.4　タッチ検出手法

平面に重畳表示された映像に対するインタラクションを実現するために、指による平面へのタッチの検出を行う。本研究では、デプスカメラを用いることで、映像投影する平面をタッチパネル化する。

これまで、デプスカメラを用いた平面のタッチ検出手法が存在している。Wilson らは、デプスカメラから得られるデプス画像を用いたタッチ入力の検出手法を提案した [10]。この手法では、デプス画像の背景差分を用いている。しかし、タッチ時の指のデプスと平面のデプスの値が近いために、指先検出することが難しいという問題点が存在する。また、カメラを固定させて、デプス画像の背景差分をすることでタッチ検出を行っているため、本研究のようなデプスカメラを固定しない場合には背景が毎フレーム変わってしまうため、適用することができない。そこで、既存のタッチ検出手法を拡張して、デプスカメラが固定されていなくてもタッチ入力を検出する手法を述べる。

背景差分の代わりに、指先内部のデプスの平均値と、外部のデプスの平均値を比較を行う。第一に、デプスカメラから取得した赤外画像 (図 8 (a)) からエッジ検出 (図 8 (b)) を行い、エッジ画像にクロージングを適用させることで、指先の内部を埋める (図 8 (c))。そして、クロージング画像を左上から走査して初めて見つけた白画素を指先位置とする。最後に、指先位置を中心としたある一定の領域で、指先内部のデプス値の平均と、指先外部のデプス値の平均の差が、しきい値以下ならタッチ入力と認識する (図 8 (d))。

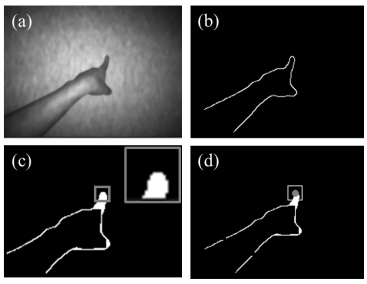


図 8　タッチ検出手法の処理過程

図 9 は、システムがタッチ入力を認識している様子を示している。ユーザはプロジェクタで投影された仮想ボタンをタッチしている。

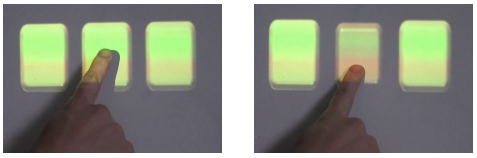


図 9　仮想ボタンをタッチしている様子

2.5　デモンストレーション

作成した実験装置を用いて、このシステムの有用性を示すようなアプリケーション例を実装した。

図 10 (a) はユーザがシステムを頭部に装着して使用している様子を示している。

図 10 (b) は写真ビューアを操作している様子を示している。ユーザは頭部を動かしてスクリーンを見回すことで、整列表示された写真の全体を見ることができる。またそれぞれの写真は大きいサイズで配置されているため、疑似的に拡大されたスクリーンによって、ユーザは各写真を見やすいサイズで見ることができる。

図 10 (c) はマップビューアを操作している様子を示している。ユーザは広域表示されたマップのあるエリアをタッチすることで、その周辺を拡大して表示し、建物の詳細な情報を見ることができる。

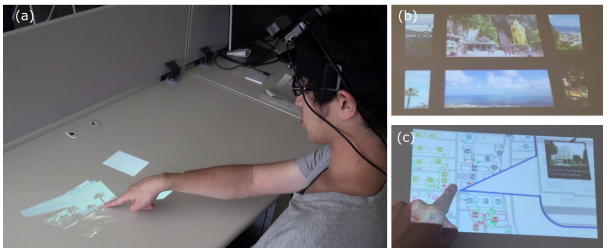


図 10　(a) 使用場面、(b) 写真ビューア、(c) 地図ビューア

3　評価実験

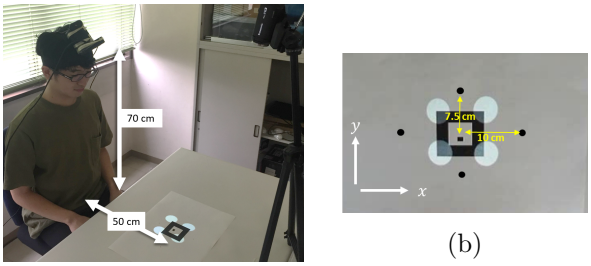
3.1　位置合わせの精度評価

実験システムと位置合わせの手法を用いて、頭を動かしたときにどのくらい映像が固定されるのかを測定した。AR マーカの 4 頂点にプロジェクタで円を投影し、位置合わせを適用させたときの位置と回転の誤差を求めた。

実験条件： 図 11 (a) は実験のセットアップを示す。ユーザと AR マーカまでの距離は約 50 cm、机から装置までの高さは約 70 cm で行った。実験中は机の上部のビデオカメラから 30 fps で映像を撮影する。この映像を解析することで、各マーカの姿勢推定を行う。姿勢推定で取得した変換行列から、並進 3 自由度と回転 3 自由度の計 6 自由度を算出した。マーカの姿勢推定には ARToolKit [11] を用いた。

© 2018 Information Processing Society of Japan

この実験では、回転中心から装置までの並進量 $\boldsymbol{t}_{\mathrm{p}}$ は手動で設定した。投影映像を安定させるために、式 (2)、式 (3) で得た回転角 α と β に 1€ filter [12] を適用した。1€ filter とは、速度に応じてフィルタの強さが変わる一次ローパスフィルタである。このシステムは約 30 fps で動作した。また、プロジェクタ・カメラ間のキャリブレーションは事前に行った。



(a) (b)

図 11　実験環境

タスク： 実験装置を装着した被験者は椅子に着座し、前方の机上にある AR マーカの中心を見る。プロジェクタは AR マーカの 4 頂点に、円を投影する。その後、ユーザは指定した位置を見るために頭を動かす。投影したマーカに位置合わせを適用することで、AR マーカの 4 頂点に重なるように投影される。視線の移動方法を次の 4 ステップを 1 セットとして行う：1) 中央から指定した位置へ移動、2) 指定した位置で静止、3) 指定した位置から中央へ移動、4) 中央で静止。この 1 セットを各方向に連続して行う。各ステップは 1 秒で行い 1 セットあたり 4 秒かかる。時間はメトロノームを用いて実験を行った。

図 11 (b) は指定された位置を示している。指定した位置は上下左右の 4 箇所とした。指定された位置はプロジェクタの投影サイズを考慮して、AR マーカの中心から上下に 7.5 cm 離れた位置、左右に 10 cm 離れた位置とした。指定した位置を見る順番は右、左、上、下とした。

3.2　結果

図 12 はタスクを行ったときの並進と回転の誤差を示している。各ステップは 30 フレームで行われ、120 フレームで 1 セットを表す。また、頭が静止するとき（ステップ 2、4）の並進誤差を表 1 に、回転誤差を表 2 に示す。

表 1　頭部が静止しているときの並進誤差

	x [mm]	y [mm]	z [mm]
MAE	10.2	7.20	14.3
RMSE	14.6	9.75	17.7
SD	13.0	9.45	15.5

また、頭部が動いているときの誤差の最大値は、x は 37.6 mm、y は 27.1 mm、z は 37.3 mm、ϕ（x 軸回りの回転量）は 9.77 deg、θ（y 軸回りの回転量）は 21.4 deg、ψ（z 軸回りの回転量）は 4.77 deg であった。

表 2　頭部が静止しているときの回転誤差

	ϕ [deg]	θ [deg]	ψ [deg]
MAE	4.02	4.69	1.01
RMSE	5.42	6.74	1.39
SD	5.09	6.72	1.39

3.3　システムの遅延評価

システムの遅延を測定するために、先の実験と同様のタスクを行い、その様子を被験者の頭部と投影映像が映るようにビデオカメラで撮影し、解析を行った。被験者の頭部が静止したとき (ステップ 2) から投影されたマーカが AR マーカの頂点に戻るまでの時間を遅延時間とした。その結果、システムの遅延は約 0.2 秒であることが分かった。

3.4　考察と限界

図 12(a) より、頭を左右に動かしたときは x 方向の誤差が、頭を上下に動かしたときは y 方向の誤差が大きくなることが分かった。これはシステムの遅延による影響だと考えられる。また、表 2 より x および y 方向の誤差は約 10 mm 程度生じるため、アプリケーションを作製する際はこの誤差を考慮して表示する映像の大きめに設定する必要がある。

今回用いた位置合わせの手法は平面の法線に依存するため、平面検出の誤差が投影誤差に関与すると考えられる。平面検出の精度がより高くなれば、並進・回転の誤差が小さくなると思われる。

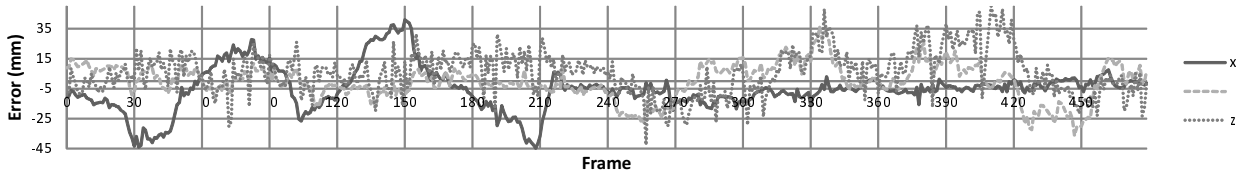
本手法は平面の法線を用いているため、x、y 方向の並進 2 自由度と回転 1 自由度を取得することができない。そこで、慣性センサなど他のセンサを用いることで自由度を増やすことができると考えられる。

4　おわりに

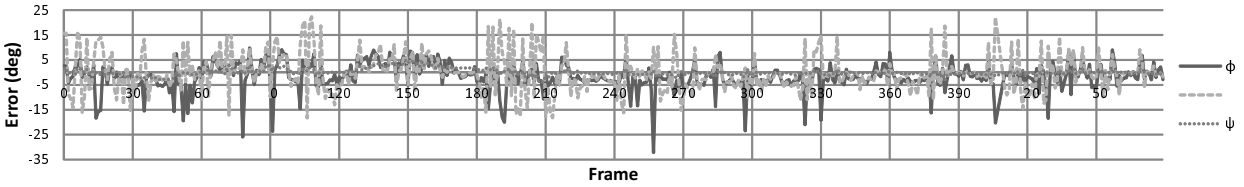
本論文では、ユーザの頭部に装着したプロジェクタでテーブルなどの平面上に映像を投影し、ユーザの頭の向きに合わせて映像を変えることで擬似的に広い表示領域を実現し、その映像に対してタッチ入力が可能なテーブルトップインタフェースを提案した。

仮想空間において、推定した平面上に映像を重畳し、透視投影変換を行うことで得られる二次元画像を、実空間において、プロジェクタで平面に投影することで、平面に映像を重畳表示することができた。また、推定した平面の情報を用いて頭部の姿勢推定を行うことで実空間と映像の位置合わせが可能なことを確認した。さらに、デプスカメラを用いて指先内部のデプスの平均値と、外部のデプスの平均値を比較することでタッチ認識を行うことができた。また、このシステムの有用性を示すようなアプリケーションの例を実装した。

© 2018 Information Processing Society of Japan



(a)



(b)

図 12　頭を各位置に向けたときの並進の誤差 (a) と回転の誤差 (b)

実験装置を用いて、実空間と投影映像のずれを測定することで位置合わせの精度評価実験を行った。その結果、頭が静止している時 x および y 方向で平均が約 10 mm の誤差であることが分かった。またシステムの遅延評価では、約 0.2 秒の遅延が生じることが分かった。

今後の課題として、様々なインタラクションを可能とするためにマルチタッチの認識が挙げられる。また、頭部にプロジェクタとデプスカメラを装着し、平面に広く映像投影することで、疑似的な大画面が実現されるかどうかを被験者を集めて調査することである。将来、装置の小型化によって、イヤホン型やヘッドセット型のようなシステムが可能になると考えられる。それにより、持ち運びがしやすく手軽に使用することができるシステムが実現できると期待される。

参考文献

[1] Helena Roeber, John Bacus, and Carlo Tomasi. Typing in thin air: the canesta projection keyboard-a new method of interaction with electronic devices. In *21nd Human Factors in Computing Systems*, pages 712–713, 2003.

[2] Andrew Wilson, Hrvoje Benko, Shahram Izadi, and Otmar Hilliges. Steerable augmented reality with the beamatron. In *the 25th annual ACM symposium on User interface software and technology*, pages 413–422, 2012.

[3] Xiang Cao and Ravin Balakrishnan. Interacting with dynamically defined information spaces using a handheld projector and a pen. In *the 19th annual ACM symposium on User interface software and technology*, pages 225–234, 2006.

[4] Bonifaz Kaufmann and Martin Hitz. X-large virtual workspaces for projector phones through peephole interaction. In *the 20th ACM international conference on Multimedia*, pages 1279–1280, 2012.

[5] Masahiko Inami, Naoki Kawakami, Dairoku Sekiguchi, Yasuyuki Yanagida, Taro Maeda, and Susumu Tachi. Visuo-haptic display using head-mounted projector. In *the IEEE Virtual Reality 2000 Conference*, pages 233–240, 2000.

[6] Pranav Mistry, Pattie Maes, and Liyan Chang. Wuw-wear ur world: a wearable gestural interface. In *extended abstracts on Human factors in computing systems*, pages 4111–4116, 2009.

[7] Jens Grubert, Matthias Heinisch, Aaron Quigley, and Dieter Schmalstieg. Multifi: Multi fidelity interaction with displays on and around the body. In *the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3933–3942, 2015.

[8] Gun A Lee, Mark Billinghurst, and Gerard Jounghyun Kim. Occlusion based interaction methods for tangible augmented reality environments. In *the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, pages 419–426, 2004.

[9] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[10] Andrew D Wilson. Using a depth camera as a touch sensor. In *ACM international conference on interactive tabletops and surfaces*, pages 69–72, 2010.

[11] ARToolKit. `http://www.hitl.washington.edu/artoolkit/`.

[12] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2527–2530, 2012.

© 2018 Information Processing Society of Japan