

K-001

視点追跡による立体視を用いたプロジェクション型 AR の提案

Proposing AR System Using Viewpoint Tracking Stereopsis with A Projector

道後千尋[†]
Chihiro Dogo

小林達也[‡]
Tatsuya Kobayashi

柳原広昌[‡]
Hiromasa Yanagihara

半谷精一郎[†]
Seiichiro Hangai

1. はじめに

AR (Augmented Reality) とは現実世界と仮想世界の融合を目的とする技術の総称である。このようなシステムの多くは大掛かりな機材の導入や空間的制約、身につける煩わしさという課題がある。

そこで一般に浸透した機器を使い、低コストで AR 体験が出来るシステムを提案した先行研究に、鷗沼らのシースルーモバイル AR システム [1] がある。このシステムではタブレット PC とデブスカメラで構成され、PC の裏側へ手を伸ばすことで、画面に表示された仮想 3D 物体を操作可能である。しかしタブレット PC を片手で支える点、可視領域の狭い点で課題がある。また低コストの 3D 表現としては視点追跡を用いた先行研究がある。この手法は Wii リモコン [2] やタブレット内蔵カメラ [3] などを使い視点の三次元座標をとり、位置に応じた画像を見せ立体感をだす。どちらの先行研究も画面内で完結しており AR 機能はない。

本研究では上記 2 点の先行研究を踏まえ、低コストで 3D 表現が可能な AR システムを提案する。提案するプロジェクション型 AR システムはプロジェクタの投影画像をユーザの視点位置に応じて調整することで、運動視差による裸眼立体視を実現する。この分野には木島らのバーチャル解剖模型 [4] があるが、本システムは立体物に映像を投影して内部が見える類の立体感を目指すものではなく、平面に映像を投影して立体物があるように見せるという立体感を目指している。そのため投影スクリーンとなる立体物を用意する必要はなく、より個人での使用を意識したシステムとなる。

本システムでは、ユーザの視点位置はカメラの画像解析によってリアルタイムに追跡する。また、カメラとプロジェクタを任意の場所に設置・移動できるようにするため、2 つの AR マーカを用いたカメラ-スクリーン間のキャリブレーション方法を提案する。これによりシステムの利便性が高まり、家庭内で簡単に裸眼立体視が体験可能となる。評価実験では投影精度の定量評価、及び立体感のユーザ評価を行い、提案システムの有効性を示す。

2. 提案する視点追跡による立体視 AR システム

投影する画像をユーザの視点に応じて調整することで、プロジェクタによる裸眼立体視を実現する。本研究で使用するシステムの様子を図 1 に示す。この装置は以下のハードウェアからなり、カメラによってユーザの視点を追跡し、それに応じてリアルタイムで画像を生成し投影している。

- ・プロジェクタ (vivitek 社) Qumi Q5-WT
- ・カメラ (logicool 社) HD Pro Webcam C920t
- ・PC (Intel Core(TM)2 Quad CPU)

カメラ画像は解像度 640 × 480、フレームレートは 30FPS、投影の解像度は 1024 × 768 に設定し、CG と現実の重ね合わせに必要なカメラの焦点距離、スクリーンの投影領域の大きさなどの各種パラメータは予め測定した。

ユーザの視点検出のための視点追跡は asmLibrary [5]、投影画像の生成には OpenGL を用いる。



図 1: 提案システムの様子



図 2: キャリブレーションに使用した AR マーカ

2.1. キャリブレーション

視点位置に対応する映像を投影するためには、目とスクリーンの相対位置姿勢 X_{SE} を出す必要がある。カメラ画像解析による視点追跡で算出されるのはカメラ-目間の相対位置 X_{CE} のみであるため、スクリーン-カメラの相対位置姿勢 W_{SC} が別途必要である。これらの行列の関係は以下の式 1 のようになる。相対位置姿勢は 4 × 4 行列であり、式中の \bar{X} は 3 次元座標 X の同次表現である。

$$\bar{X}_{SE} = W_{SC} \bar{X}_{CE} \quad (1)$$

本システムにおいて W_{SC} を固定させるということはカメラとプロジェクタの投影位置を常に固定しなければならないことになり設置に不便が生じる。よって本実験では必要に応じてカメラとプロジェクタを任意の場所に配置できるように投影前に簡易的なキャリブレーションを行うこととした。この位置関係はシステム実行中は変わらないものとする。

キャリブレーションの際にスクリーン中央に AR マーカを置き直接撮影すると、システム使用時に顔を映せる空間が十分に確保出来なくなるため、本研究では図 2 に示す固定した 2 つの AR マーカをキャリブレーションに用いた。マーカの固定台にはブックエンドを使い床の AR マーカを 1、背の AR マーカを 2 とした。キャリブレーションの様子を図 3 に示す。AR マーカ 1 と 2 の相対位置姿勢 W_{12} は、ARToolKitPlus [6] を使用し予め算出した。キャリブレーションはまず AR マーカ 1 をスクリーン中央に置き、カメラで AR マーカ 2 を撮影を行い、その画像を ARToolKitPlus を使用して解析し相

[†]Tokyo University of Science

[‡]KDDI R&D Laboratories, Inc.

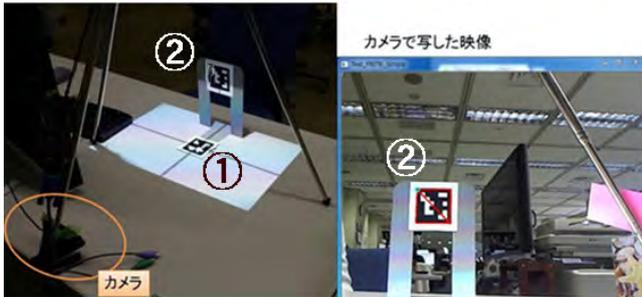


図 3: キャリブレーションの様子

対位置姿勢 W_{C2} を出す。相対位置姿勢 X_{12} 、 W_{c2} と各座標 X_{SE} 、 X_{CE} の関係は以下の式が成り立つ。

$$\bar{X}_{SE} = W_{12}W_{C2}\bar{X}_{CE} \quad (2)$$

ただし X_{SE} はスクリーンから見た目の3次元座標、 X_{CE} はカメラから見た目の3次元座標である。

キャリブレーション結果と顔認識結果をこの式に代入することで、リアルタイムで目-スクリーンの相対位置姿勢が分かり、投影映像を作成することができる。キャリブレーション後の処理の流れは以下の通りである。

1. 目の位置取得
2. カメラ 目間の相対3次元位置 X_{CE} を算出
3. 視点位置に応じた映像を生成・投影

2.2. 目の位置取得

asmLibrary を利用した顔認識でカメラ画像上の目の位置をピクセル座標として得る。本システムでは認識結果の揺らぎを抑えるため、この座標を時間軸方向に平滑を加えて使用している。平滑フレーム数は、物を眺める程度の速度で移動しながら観察した際に、表示ディレイを感じないという基準で10フレームに設定した。

2.3. カメラ 目間の相対3次元位置を算出

画像上の左右の目のピクセル位置から、カメラを原点とした目の3次元座標 X_{CE} を算出する。行列の各要素については $X_{CE} = [x_{CE}, y_{CE}, z_{CE}]^T$ とした。算出はカメラ画像と現実との比 $a[\text{mm}/\text{pixl}]$ を式(3)から求め、式(4)に代入することで行う。[1]

座標はカメラ画像における右目の座標を $R = [r_x, r_y]^T$ 、同じく左目を $L = [l_x, l_y]^T$ とし、カメラの焦点距離を f 、ユーザの両目の距離を $d[\text{mm}]$ とした。(図4)なお両目の距離 d は65として計算した。 c_x, c_y はカメラの光軸とピクセル座標原点とのズレである。 c_x, c_y, f は事前にキャリブレーションによって求めた。

$$a = \frac{d}{\sqrt{(r_x - l_x)^2 + (r_y - l_y)^2}} \quad (3)$$

$$\begin{cases} x_{CE} = (r_x - c_x) \times a \\ y_{CE} = -(r_y - c_y) \times a \\ z_{CE} = f \times a \end{cases} \quad (4)$$

2.4. 視点位置に応じた映像を生成・投影

目の位置座標からユーザに見えるであろう画像を OpenGL で作成する。本システムでは利目の関係から右目の座標を基準として画像生成を行った。

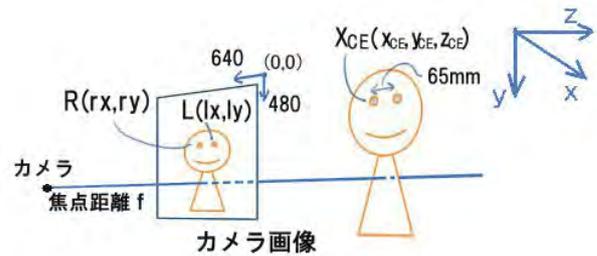


図 4: 座標の配置



図 5: AR マーカを取り付けた眼鏡

まず glFrustum 関数を使い、視線をスクリーンが存在する位置方向に向ける。 glFrustum 関数の視線の方向は画面枠を指定して確定するので、3D モデルを画面中央に表示すると引数 (*left, right, bottom, top*) はそれぞれ以下のように指定した。

$$\begin{aligned} \text{left} &= (x_{CE} - w_s/2) \times n/z_{CE} \\ \text{right} &= (x_{CE} + w_s/2) \times n/z_{CE} \\ \text{bottom} &= (y_{CE} - h_s/2) \times n/z_{CE} \\ \text{top} &= (y_{CE} + h_s/2) \times n/z_{CE} \end{aligned} \quad (5)$$

ただし w_s, h_s はそれぞれ投影スクリーンの幅と高さでありプログラム実行時に引数として与える。また n は OpenGL の描画設定に合わせるための定数である。

この後、 $\text{glTranslatef}(x_{CE}, y_{CE}, z_{CE})$ でモデルを移動させ、ユーザ視点の画像が作成できる。これを 3D レンダリングしてプロジェクタに出力し、投影する。

3. 精度の定量評価

提案したシステムを、顔認識精度・視点推定精度・投影誤差の3点から定量評価する。

3.1. 顔認識精度

顔認識システム asmLibrary の精度についてマーカ認識と比較して評価する。マーカ認識には ARToolKitPlus による AR マーカ認識を使用し、眼鏡上部に AR マーカを取り付けたもの(図5)をかけることで顔認識を行った。マーカの位置と目の位置とのズレに関しては計測し、修正した。実験条件は以下の通りである。

- ・人の顔をカメラ対し 30 度の角度をつけ固定する
- ・プログラムが計算した人の目の3次元位置を mm 単位の数値で出力
 - ・左右を x 方向、上下を y 方向、前後を z 方向とする
 - ・撮影レート 30、400 サンプル(約 13 秒間)

実験結果を図6に示す。ただし図6のグラフはどちらも平滑化済みであり、マーカ認識の方はマーカを検出できなかったフレームに関しては前のフレーム結果を保持する改善を加えている。

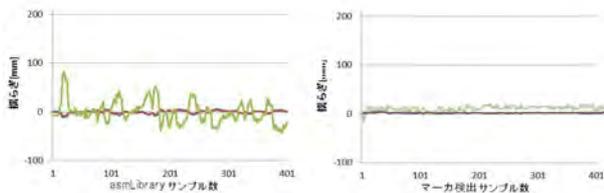


図6: asmLibrary とマーカ認識の精度比較

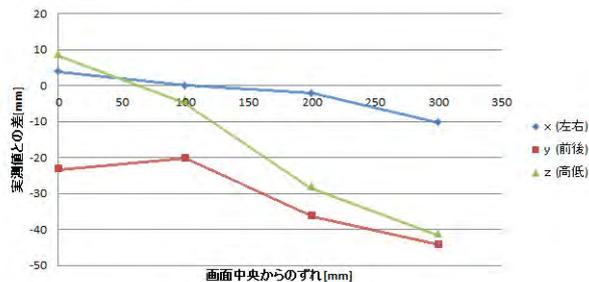


図7: スクリーン対 AR マーカの相対位置の検証

図6から分かるように asmLibrary の追跡はマーカ認識に比べ揺らぎが多いことがわかる。実際に投影像を見ても asmLibrary の方は常に揺れているのが視認でき、マーカ認識の方では静止状態に見えた。この実験結果より、本システムは最終的には裸眼での立体視を目的とするが、現時点においてはマーカ認識を使用して顔認識精度に依らない精度を検討することとした。より精度の高い顔認識手法の適用および評価については今後の検討課題とする。

3.2. 視点推定精度

顔認識精度に依らないシステムの視点推定精度の検討する。評価はスクリーン対 AR マーカの相対位置について実測とプログラム算出結果の誤差を測定、評価した。測定の際、マーカの位置をできる限り正確に固定するため眼鏡は使わずに AR マーカのみをそのまま追跡、実測した。結果を図7に示す。図7より以下の2点が分かる。

- ・撮影画像中央からの距離に対し+5%の誤差がある
- ・上と別に y(距離) は常に 2cm の誤差がある

誤差の程度が撮影画像の場所に依存することから、この原因はレンズの歪みとレンズと撮像素子の距離差と考えられる。レンズ歪みはキャリブレーションによって校正することが可能である。またレンズと撮像素子の差は、プログラムは撮像素子を原点において算出するが実測では素子手前のレンズ部分からしか測定できないことによる誤差であり、プログラム中では撮像素子を原点として扱うことで統一した。

3.3. 投影像の誤差評価

視点追跡の誤差が使用時にどの程度の影響を及ぼすか検討するため、使用が想定される距離・角度の視点位置から、投影画像と実物との見た目の誤差を計測した。評価はスマートフォンのカメラを目の代わりとして、実際にユーザが観測するであろう投影画像を撮影し、行った。図8は撮影の際に使用したスマートフォン (Galaxy SC-04E) である。なお実物と映像の位置合わせについては手前下の頂点が重なるように実物を配置することで行った。

投影画像と実物との誤差について映像と実物の比較を図9に示す。画像左が映像、中央が実物、右重ね合わせ (映像: 青、



図8: 背面に AR マーカを取り付けたスマートフォン



図9: 映像と実物の比較

実物: 赤) となっている。結果として合わせてみると実物とのズレが視認できるが、実物なしで歪んでいると感じるほどではなかった。これは実用可能範囲内の誤差だといえる。

4. 被験者による立体感の評価

以下の項目を4段階でユーザ評価を行った。被験者は20~50代の男女11名である。評価にあたって被験者は視点追跡のための AR マーカを取り付けた眼鏡を使用し、実物モデルは使用せず映像のみを見て判断を行った。

1. 両目/右目のみ状態で見て立体的に見えるか
2. 見え方/モデルごとの立体感の差について
3. どの程度の誤差が許容範囲か

結果として以下の知見が得られた。図10は比較した見え方、図11は比較したモデル、図12は結果をまとめたグラフである。

4.1. 両目/右目のみ状態で見て立体的に見えるか

図12(1)から片目で見た方がより立体感が得られることが分かる。これは視点位置を右目として画像を生成していることが原因として考えられる。左目でなく右目を選んだのは右目が利目の人が多いからである。

一方で両目の中心を視点位置とした画像も生成したが図12(1)と同様に両目より片目の方が立体的に感じられるとの回答もあり、両目で見た際に左右の目による視差も立体感に影響を及ぼしていると考えられる。

プロジェクタでの投影では、眼鏡などを使わない限り左右の目に違う映像を映す技術は確立しておらず、両眼視差はコンテンツ作成の際の制限となると考えられる。

4.2. 見え方/モデルごとの立体感の差について

図12(2)より a の見え方が最も立体的に見えることがわかる。特に c の床についていない状態の見え方は一般的に有りうる状況ではないので想像しづらく、歪んで (或いは床に張り付いて) 見えると考えられる。また a と b の差に関しては正面近くから評価検討している被験者が多かったため、正面において確実に立方体とわかる a が選ばれたと考えられる。実際に第一印象としては a が立体的に見えたが動くとき大差ないとする意見も得られた。

図12(3)より、モデル間での立体感の差は評価は際立った片寄りはなく、立体感の差は人それぞれであると思われる。また自由記述回答としては次のような真逆な意見が得られた。セイロンについて

- ・モデルが実在するので立体感がある



図 10: 比較した見え方



図 11: 比較したモデル

- ・モデルが実在する分、それと比べて違和感があるプリンについて
- ・角がない分、幾何的な違和感が少ない
- ・角がない分、はっきりとした奥行を感じられない

上記の回答から分かるように同じ理由でも立体感を感じる被験者とそうでない被験者がいる。特にセイロンとプリンの比較では“どちらかといえば”で答えた被験者はおらず半々になっており、立体感は個人差があることが見受けられる。

4.3. 誤差の許容範囲

評価は実物体と比べて歪み無く見えるか、視点とのズレや違和感はないかの観点から4段階評価で行った。目安として1はどこから見てもズレている、2は視点によってはズレている、3はズレてはいるがさほど気にならない、4はズレていないとした。投影映像に加えた誤差は5cm刻みに4段階で大きくした。

図12(4)よりズレの気にならない誤差は5cm以内と分かる。よって今後、視点追跡のシステムをARマーカから顔認識へと変更する際は、現在のシステムに加えて5cm以下のズレを許容誤差として比較検討する必要があるという知見を得られた。

5. まとめ

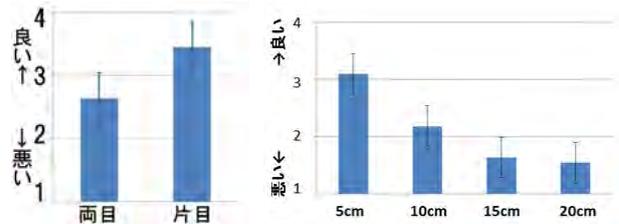
本研究ではリアルタイム視点追跡による立体視を用いたプロジェクション型ARシステムの提案を行った。提案システムを用いて性能評価を行い、視点追跡精度の誤差を定量的に明らかにしその改善策を考察した。またユーザ評価も行い立体感の得やすいアングルおよびモデル形状を考察した。今後の課題としては以下の点が挙げられる。

- ・顔認識精度の改善

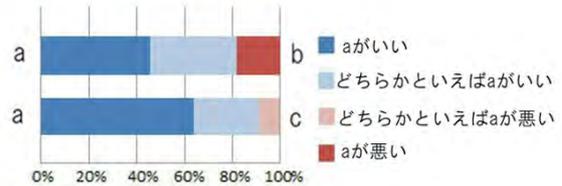
横顔・俯き等の顔角度が大きい場合に検出性が低く、ユーザの使用感でもこの点が多く挙げられた。これは現在使用しているasmLibraryの特性であり、別のライブラリまたはセンサに置き換えるか併用することによって改善が可能である。システム検証結果である許容誤差5cm以内を目安に、現在使用できる技術を模索中である。

- ・立体感の改善

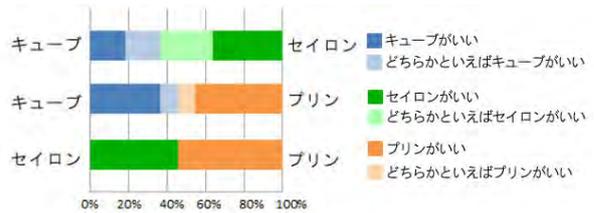
現在右目を視点に画像生成を行っているため両目で見ただけに立体感が得られていないが両目で見ただけにも立体感を感じられるよう改善する必要がある。両目で見ただけにどの位置から見ていると感じているのかを考慮した画像生成にすること



(1) 両目/右目のみでの立体感 (4) 誤差の許容



(2) 見え方ごとの立体感



(3) モデルごとの立体感

図 12: 結果のグラフ

とで、より立体感を感じられる画像提示が可能になると考えられる。

- ・ユーザ評価の再調査

今回の被験者による評価では、特に立体感を感じるかという点で一貫した傾向が得られなかったため項目を変更して再度評価する必要がある。

参考文献

- [1] 鵜沼由布子ら, “デブスカメラを用いたシースルーモバイルARシステム,” 第20回画像センシングシンポジウム, IS2-38, 2014
- [2] Lee, “Head Tracking for Desktop VR Displays using the WiiRemote” IEEE Prevasive computing p.p.39-45, 2008
- [3] Francone, “Using the User’s Point of View for Interaction on Mobile Devices.” IHM ’11, 24-27, 2011
- [4] 近藤ら, “自由曲面への投影を用いたバーチャル解剖模型” 日本バーチャルリアリティ学会論文誌, p201-208, 2005
- [5] Y.Wei, “asmLibrary Library of Active Shape Model,” <http://code.google.com/p/asmLibrary/>
- [6] Daniel Wagner and Dieter Schmalstieg: ARToolKitPlus for Pose Tracking on Mobile Devices. In Proc. 12th Computer Vision Winter Workshop (CVWW’07), February 2007.