

AReyes : 協調型実世界マーカ読み取りシステム

宮寺和彦[†] 岩井将行^{††} 戸辺義人^{†††}

現実世界上のオブジェクトを、カメラにより認識するアプリケーションにおいて一般的に利用される ARToolkit は、認識したオブジェクト情報を複数台のコンピュータで相互に利用するようには設計されていない。また、ビジュアルマーカの一部分がカメラの視野から外れたり、角度が不適切なとき認識できなくなる問題が生じていた。そこで、われわれは複数台のカメラを利用し、認識したオブジェクト情報を、ネットワークを介してデータを相互に利用するシステムを開発した。

AReyes: A Cooperation Type Detection System for Visual Markers

Kazuhiko Miyadera[†] Masayuki Iwai^{††} Yoshito Tobe^{†††}

ARToolkit, which has been used a software tool for Augmented Reality application can recognize markers by a single camera. However ARToolkit is not designed to use plural object information to recognize by the computer of the stand mutually. In addition, the part of the visual marker deviated from the field of vision of the camera, this problems will happen when an angle is inappropriate. Therefore, we propose a system which uses parallel cameras which exchange the object information that can recognize mutually through a TCP/IP network.

1 はじめに

現実世界上のオブジェクトを、カメラにより認識するアプリケーションにおいて、一般的に利用される ARToolkit は、認識したオブジェクト情報を複数台のコンピュータで相互に利用するようには設計されていない。また、ビジュアルマーカの一部分がカメラの視野から外れたときや、角度が不適切なとき認識できなくなる問題が生じていた。そこで、われわれは複数台のカメラを利用し、認識したオブジェクト情報を、ネットワークを介してデータを相互に利用する AReyes を開発した。本稿では AReyes の設計と実装について報告する。

2 拡張現実と ARToolkit の制約

2.1 拡張現実

拡張現実とは現実環境にコンピュータを用いて、情報を付加提示する技術である。主にオブジェクトの認識には光学カメラが用いられることが多い。しかし、光学カメラを用いた場合、画像処理が必要となり、これは多大なコストがかかってしまう。

そこで認識するオブジェクトを自然物などではなく、2次元マーカに限定することにより、速度と精度を獲得する手法が存在する。それらのシステムを用いたアプリケーションの開発を行っている ARToolkit [1] や CyberCode [2] が存在する。ARToolkit は四角形の絵をマーカに用いているのに対し、CyberCode は2次元バーコードを用いることによりマーカ自体に情報を埋め込んでいる。また ARToolkit を発展させ CyberCode のように2次元バーコードを利用した ARTag [3] が開発された。

2.2 ARToolkit の制約

ARToolkit は拡張現実アプリケーションの実装を助けるC言語用ライブラリである。ARToolkit はコンピュータに接続されたカメラでビジュアルマーカを認識し、マーカの位置や向きなどの情報を計算し、OpenGL でカメラからの画像に3Dオブジェクトを上書きすることを可能にする。そのため、一般的に拡張現実のアプリケーションを作成するライブラリのデファクトスタンダードとなっている。しかし、1台のみのカ

[†] 東京電機大学大学院 未来科学研究科 情報メディア学専攻

^{††} 東京大学 生産技術研究所

^{†††} 東京電機大学 未来科学部

メラでビジュアルマーカを認識しなければならず、図1のように一部分がカメラの視野外にある場合や、角度が不適切な場合はマーカを認識することが不可能である。

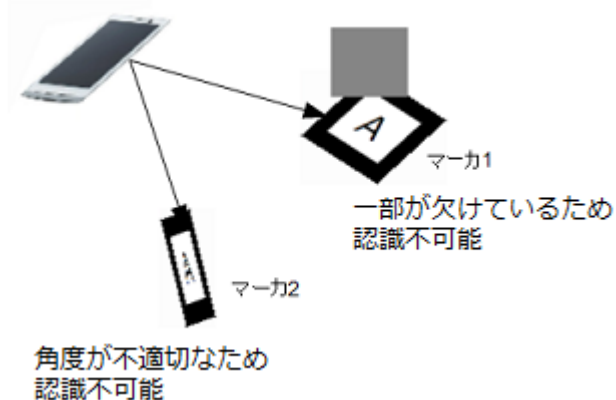


図1. 角度によるマーカ認識のブラインド問題

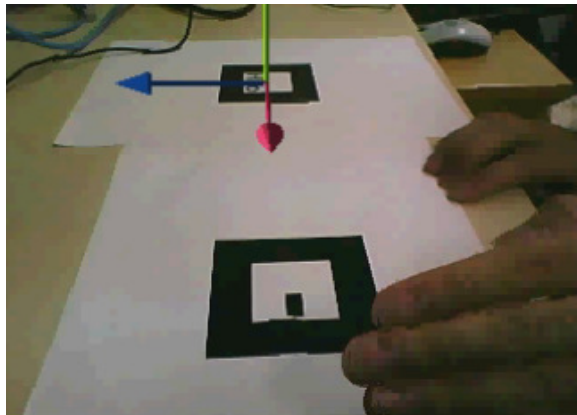


図2. 障害物によりマーカの一部分が欠けている画像

3 AReyes

3.1 ARToolKit における変換行列

ARToolKitでマーカを認識すると、マーカを中心とした座標系の行列 M から、カメラ画像を中心とした座標系の行列 C へ変換するための 4×4 の変換行列 T が求められる。

M , C , T は,

$$C = TM \cdots (1)$$

$$M = T^{-1}C \cdots (2)$$

と表すことができる。 T の逆行列である T^{-1} は、 T から計算して求めることができる。また、変換行列 T は

$$T = \begin{pmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdots (3)$$

と表すことができる。 r_1 から r_9 までは回転成分、 (t_x, t_y, t_z) は並進成分を表している。

並進成分はカメラ画像を中心とした右手系の直交座標で表されている。

3.2 AReyes の提案手法

1つのカメラで捉えることができないマーカを複数のコンピュータとカメラを用いて認識する。コンピュータの処理をサーバとクライアントに分け、クライアントのコンピュータが認識したマーカのデータはネットワークを介してサーバに送られる。サーバはマーカのデータを各クライアントのコンピュータに送る。クライアントのコンピュータは自分がマーカを認識できない場合、サーバから送られたデータを用いてマーカを認識する。

複数台のカメラとマーカを用いたとき、カメラ i を中心とした座標系の行列を C_i 、カメラ j を中心とした座標系の行列を C_j 、マーカ p を中心とした座標系の行列を M_p 、マーカ q を中心とした座標系の行列を M_q とおくと、以下の8つの式が求められる。

$$C_i = T_{ip} M_p \cdots (4), \quad M_p = T_{ip}^{-1} C_i \cdots (5)$$

$$C_i = T_{iq} M_q \dots (6), \quad M_q = T_{iq}^{-1} C_i \dots (7)$$

$$C_j = T_{jp} M_p \dots (8), \quad M_p = T_{jp}^{-1} C_j \dots (9)$$

$$C_j = T_{jq} M_q \dots (10), \quad M_q = T_{jq}^{-1} C_j \dots (11)$$

カメラ i からマーカ p を認識することができない場合、式(4) 式(5)を求めることができない。ここで、式(6)、式(11)から

$$C_i = T_{iq} T_{jq}^{-1} C_j \dots (12)$$

が求められる。同様に式(12)は、式(8)により

$$C_i = T_{iq} T_{jq}^{-1} T_{jp} M_p \dots (13)$$

と変形することができる。

式(13)はマーカ p を中心とした座標系の行列 M_p から、カメラ i を中心とした座標系の行列 C_i へ変換する式と同じであるため、式(4)の代わりに式(13)を使用することができる。したがって、カメラ i からマーカ p を直接認識できない場合でも、カメラ j 、マーカ q によって間接的にマーカ p を認識することが可能となる。

3.3 AReyes のデータ通信

クライアントはサーバに接続すると、まず `int` 型のノード ID を受け取り、これを保存する。このノード ID により、サーバからデータを受信したとき、どのクライアントのカメラからのデータか識別することができる。各クライアントはマーカを識別すると、識別した時間、自身のノード ID、あらかじめ決めてある `int` 型のマーカ ID、変換行列の回転成分、変換行列の並進成分をサーバに送信する。送信パケットを Pt とすると

$$Pt = \{Time, ID_{node}, ID_{marker}, r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, t_x, t_y, t_z\} \dots (14)$$

と表すことができる。

AReyes のデータ通信は主に送信スレッド、受信スレッド、描画スレッドの3つのスレッドから成り立っており、擬似コードで表すと図3のようになる。描画スレッドは描画を繰り返すだけなので、ここでは省略する。

サーバはクライアントから受信したデータをすべてのクライアントにそのまま送信する。単純なエコーサーバで実装している。各クライアントは受信したデータから、ノード ID を比較して、自身以外のノードのデータを格納する。

```

MakerList = CreateMakerList(); // マーカリスト

Thread LoopLocalDetect{ // 送信スレッド
    Loop{
        D = DetectMarker(); // マーカ認識 式(13)
        if(D != NULL){ // ローカル認識成功
            NodeList.send(D); // ネットワーク送信
            marker = MakerList.select (D.IDmarker); // リストからマーカを選定
            marker.updatePosition(D); // 式(1)
            maker.setSelfDetection(time() ); // 最終ローカル認識時間の更新
        }
    }
}

Thread LoopWait{ // 受信スレッド
    Loop{
        receive(D); // パケット受信
        marker = MakerList.select (D.IDmarker); // マーカの選定
        if(!marker.isSelfDetection (D.time)){ // ローカル認識失敗の確認
            tx = translate(D); // 式(12)
            marker.updatePosition(tx); // 位置情報の更新
            marker.setRemoteDetect(D.time); // 最終リモート認識時間の更新
        }
    }
}

```

図3. AReyes 通信アルゴリズム

3.4 耐マーカ移動補完

今までも、マーカが障害物に隠れたときや、カメラの画面外に出たとき、角度が不適切なとき認識できない問題を、複数マーカを利用して相対距離からマーカの位置を算出する研究は存在した。しかし、それらの研究の大半はマーカの相対距離が固定されているため、個々のマーカを自由に動かすことができない。

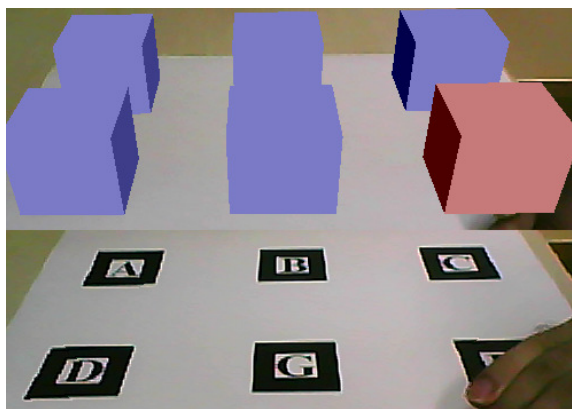


図4. ARToolKitにおける複数マーカを利用したアプリケーション

本稿の AReyes では各マーカ間の相対距離は式(13)により、計算で求めることができるため、マーカを自由に動かすことができる。

3.5 耐カメラ移動補完

1台のコンピュータに2台のカメラを接続し、データを共有することで死角を補う手法も存在する。しかし、1台のコンピュータにつながっているため、カメラを自由に動かすことができない。また、ハードウェアを限定する形となり、アプリケーションに利用することが難しくなる。個々のローカルで完結しているため、それ以上の発展が見込められない。

本稿の AReyes では複数台のコンピュータに1台のカメラを接続しているため、各々のカメラは自由に動かすことができる。また、カメラが1台接続していればいため、携帯電話などのアプリケーションに応用することも可能となる。ネットワークでデータを相互利用しているため、ローカルでカメラの台数を増やさなくても、ユーザの数だけカメラが存在し、お互いにデータを補完しあうことができる。

4 実装

Microsoft Visual C++ 2005 と ARToolKit を使用して実装を行った。

2台の Intel Atom CPU 330 1.60GHz を搭載したノート PC とカメラを用いた実験を行った。1台のコンピュータにつき1台のカメラを接続し、それぞれをカメラ i 、カメラ j とする。2つのマーカをそれぞれマーカ p 、マーカ q とし、カメラから認識できる範囲で自由に配置した。カメラもマーカも固定せず、自由に動かすことができる。

図5から図7に実験結果の画像を示す。図2において、マーカ p の一部分が障害物によって隠れているため認識することができないが、図5ではカメラ j からのデータを用いて表示することができている。また、図6ではカメラ i を移動させ、角度が変わったが、マーカ p が画面外に出てもカメラ j からのデータを用いて表示することができている。



図5. AReyes を使用したカメラ i の画像



図 6. カメラ i を移動させた画像

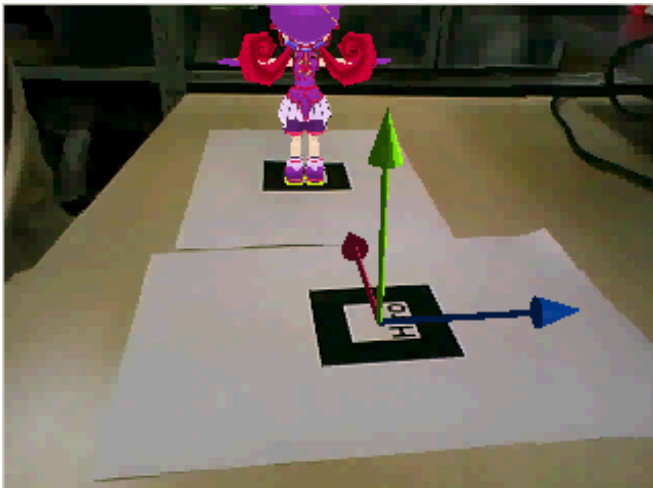


図 7. カメラ j の画像

5 評価

5.1 評価環境

初期配置として 2 台のカメラと 2 つのマーカを図 8 のように配置し AReyes を実行した。実行中にカメラ i, カメラ j, マーカ p, マーカ q をランダムに動かし, マーカが障害物に隠れる状況を作った。

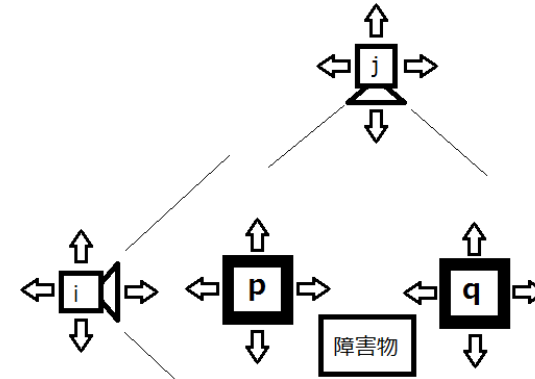


図 8. 評価実験の初期配置図

5.2 結果

AReyes を使用せずに 1 台のカメラで実験を行ったときのマーカの認識率と, AReyes を使用して 2 台のカメラで実験を行ったときのマーカ認識率を計測し比較した。6 回の実験を行った結果のグラフを, 以下の図 9 から図 11 に示す。

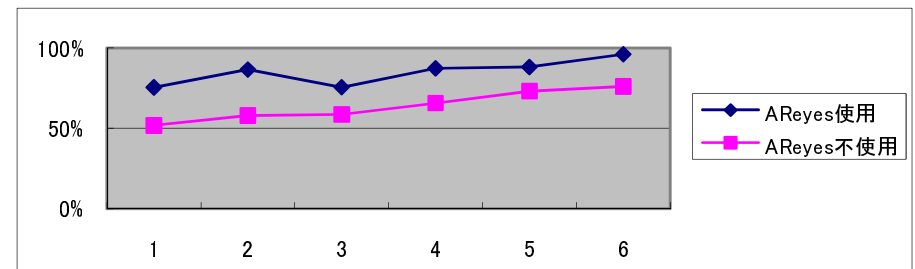


図 9. マーカ p の認識率

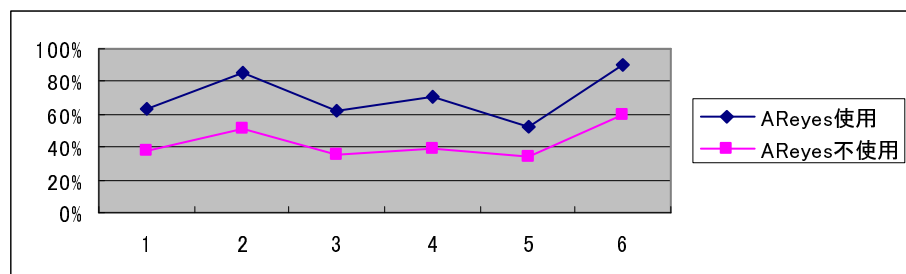


図 10. マーカ q の認識率

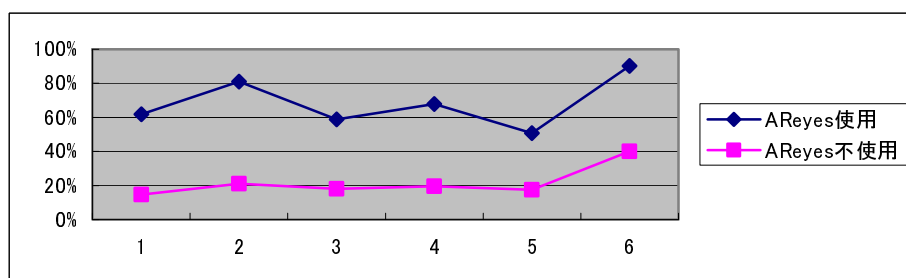


図 11. マーカ p, マーカ q を 1 画面に同時に認識した割合

AReyes を使用したとき、マーカ p において認識率は最低で 15.0%, 最高で 28.5 %, 平均で 20.9% 上昇した。同様にマーカ q において認識率は最低で 19.2%, 最高で 34.4%, 平均で 27.9% 上昇した。全体平均でマーカの認識率は 24.4% 上昇している。また、マーカ p, マーカ q を同時に認識した割合は最低で 33.4%, 最高で 59.9%, 平均で 46.6% 上昇している。

マーカが移動した場合、およびカメラが移動した場合においても、安定して認識率が向上していることがわかる。

6 議論

今回の実験では 2 台のコンピュータとカメラ, 2 枚のマーカを使用した。今後はコンピュータの台数やマーカの枚数を増やして, 規模を大きくしていきたい。

このとき, 複数のユーザからデータが送られたとき, どのデータを利用するかを判

断をおこなうための信頼度が必要となり, 信頼度の算出方法の検討が必要である。また, 本稿ではサーバはエコーサーバを利用しているが, 大規模化した場合トラフィックが大きくなりすぎてしまう。各クライアントに送るデータをまとめるか, クライアントからの要求に応じてデータを送るかなどの検討が必要となる。

また, コンピュータとマーカの数を増やすことで,

$$C_i = T_{is} T_{ms}^{-1} T_{mr} T_{lr}^{-1} \cdots T_{kq} T_{jq}^{-1} T_{jp} M_p \cdots (15)$$

のように複数のデータを経由して計算することが可能と考えられる。しかし, すべての組み合わせを計算することはコストがかかりすぎてしまい, 実用的とはいえなくなる。効率のいい組み合わせの検索方法の検討が必要となる。

ネットワークを介して得たデータを使用して表示するとき, 数 cm 程度のずれが生じることがある。これは単一のシステムではわずかな誤差だったが, 協調型にしたため計算回数が多くなり, 誤差が大きくなってしまったと考えられる。

誤差の修正方法として, 複数のユーザから受け取ったデータの平均をとる方法や, マーカの移動履歴を保存し, マーカの位置を推測する方法が考えられる。

7 まとめ

本研究では ARToolKit におけるビジュアルマーカの認識の限界を解決するために, 複数のコンピュータとカメラを使用して複数のマーカを認識し, ネットワークを介してデータを共有する AReyes を提案し開発した。これにより, マーカの認識率を平均で 46.6% 上昇させることに成功し, ARToolKit を利用して作成できるアプリケーションの可能性が広がった。

参考文献

- [1] Kato, H., Billinghurst, M. (1999). Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System. In Proceedings of the 2nd International Workshop on Augmented Reality (IWAR 99).
- [2] Jun, R., Yuji, A. (2000). CyberCode. Designing Augmented Reality Environments with Visual Tags, Designing Augmented Reality Environments (DARE 2000).
- [3] Fiala, M. (2005). ARTag, a fiducial marker system using digital techniques. IEEE Computer Society Conference on Computer Vision and Recognition, 2005(CVPR 2005), Vol.2, pp. 590-596.