

Fisheye Marker を用いた AR システムに関する研究

A Study of Augmented Reality System with Fisheye Marker

本車田匡隆 † 太田正哉 † 山下勝己 †
 Masataka Motokurumada Masaya Ohta Katsumi Yamashita
 † 大阪府立大学大学院 工学研究科
 Graduate School of Engineering, Osaka Prefecture University

1 はじめに

近年、スマートフォンなどの情報端末の高性能化に伴い、これらを用いた拡張現実 (AR) アプリの開発が盛んに行われている [1, 2, 3, 4] . 特にスマートフォンやタブレット端末では、ディスプレイの高解像度化が急速に進んでおり、今後これに合わせてカメラのキャプチャ映像も高解像度化が進むと考えられる . マーカ型 AR ではカメラ映像からマーカをリアルタイムに検出する必要があり、高解像度映像から高速にマーカを検出する手法が求められる . とところで、プラットフォーム (OS) 依存性が少なく、ネイティブアプリより開発が容易なことから、ブラウザ上で動作する Web アプリの開発が広く行われている . 近年の JavaScript エンジンの高速化により、AR のような高負荷なアプリでもブラウザ上で実行可能となっている [5] .

そこで本研究では JavaScript ベースで高速なマーカ検出が可能なマーカ型 AR システムについて検討する . マーカはラスタ走査で検知できる Fisheye (ジャノメ) で、矩形のような形状を検出するより高速な検出が可能である .

2 従来の矩形マーカの検出

マーカ型 AR ではさまざまなマーカパターンが用いられており、それぞれのシステムでマーカに応じたマーカ検出法が提案されている . ARToolkit[6] に代表される矩形マーカを用いるシステムでは、OpenCV の API, FindContours, ApproxPoly が広く用いられる . FindContours は、2 値化画像から輪郭を検出し、そのリストを作成する [7] . ApproxPoly は、作成されたリストの輪郭を折れ線近似する . これらの処理の後、矩形であるか否かを判定し、マーカの候補とする . 特に FindContours は画像中の画素を走査して輪郭を抽出するため計算時間がかかる .

3 Fisheye マーカを用いる AR システム

本研究では図 1 に示すマーカを用いる . マーカの位置を検知するためにポジションパターンとして Fisheye (ジャノメ) を用いる . 図 2(a) に示す通り、このパターンは中心を通る水平線上で、黒、白、黒、白、黒と続く線分の長さの比が 1:1:3:1:1 となる同心円で構成され

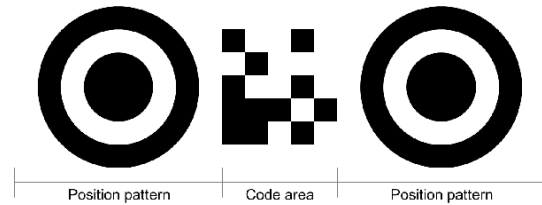


図 1 提案するマーカ

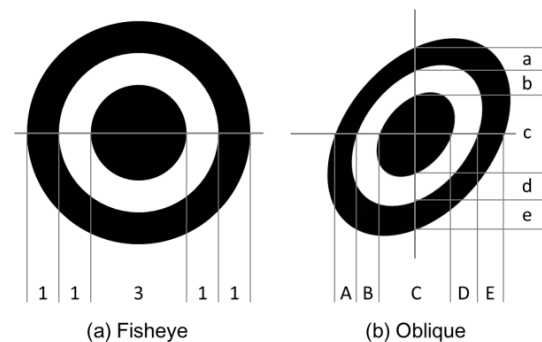


図 2 Fisheye パターン

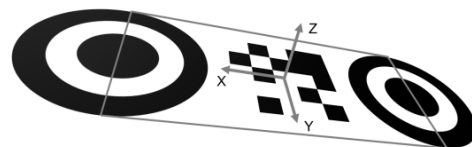


図 3 位置姿勢検出

る . 図 2(b) からわかる通り、このマーカは任意の角度から撮影しても上述の比がほぼ変わらない特徴を有する . そこでキャプチャ画像をラスタ走査し、この比 ($A : B : C : D : E$) を求め、上述の比とほぼ同程度と判断できるものをマーカの候補とする . これは QR コードのポジションパターンの検出方法と同じであり、上述の比は QR コードのポジションパターンにおけるそれと同じとしている .

この方法で得られた候補にはマーカでないものが含まれるため、ここで得られた候補について、そのマーカの中央を通る垂直線上で、黒および白の線分の比 ($a : b : c : d : e$) が上述の比とほぼ等しいと判断できない候補を除去する .

最後に図 3 に示すように、Code 領域の境界線と平行な各 Fisheye の直径と、これらの両端をつないでできる矩形を求め、これより Code 領域の Code を復号し、

さらにマーカ座標系とカメラ座標系の変換行列 [8] を求め、AR コンテンツを描画する。以上のシステムはすべて HTML5/JavaScript を用いて実装できる。

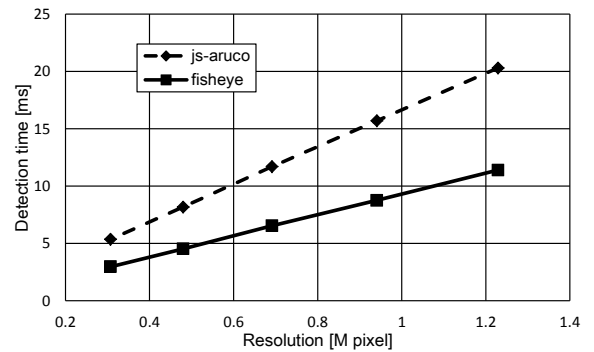
4 性能評価

提案システムを実装し、その性能を評価した。提案法と同じく HTML5/JavaScript で実装された js-aruco[5] を比較手法とする。js-aruco は ARToolkit と同様な矩形のマーカを OpenCV の FindContours を用いて検知する AR システムである。まず、キャプチャ映像の解像度に対するマーカの検出速度を計測した。計測に用いた機器として、PC は ASUS 社製 ZENBOOK UX31E (CPU: Intel Core i7 2677M, 1.8GHz, RAM: 4GB, OS: Windows 7 Home Premium 64bit), ブラウザは Google Chrome 23.0.1271.97-m, Web カメラは Microsoft 社製 LifeCam Studio Q2F-00020 (HD1080p, 800 万画素), またタブレット端末は、TOSHIBA 社製 REGZA Tablet AT570 (CPU: NVIDIA Tegra 3, 1.30GHz, RAM: 1GB, OS: Android 4.0.3, Web カメラ: 500 万画素), ブラウザは Opera Mobile ver.12.10 を用いた。解像度は、640×480 (307,200 画素), 800×600 (480,000 画素), 960×720 (691,200 画素), 1120×840 (940,800 画素), 1280×960 (1,228,800 画素) とし、マーカを真上から撮影した。図 4(a) および (b) に結果を示す。図より、PC およびタブレットの双方において、提案法は js-aruco より高速であり、PC では解像度が高くなるにしたがって、その効果が顕著になることがわかる。

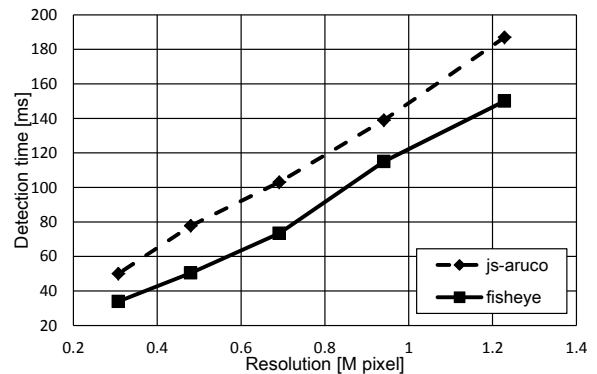
次にキャプチャ画像に映るマーカのサイズと、その認識率を計測した。マーカのサイズとは、js-aruco では正方形のマーカの面積 (画素数)、提案法では Fisheye パターン 2 つとその間にはさまれる領域 (オーバル状) の面積 (画素数) である。キャプチャ画像のサイズは 1280×960 とした。測定結果を図 4(c) に示す。図より、提案法は js-aruco と同じ認識率 (90% 以上) を得るために、マーカ面積をおよそ 3.1 倍大きくする必要があることがわかる。これは、マーカ面積を等しくした場合、カメラとマーカの距離を 43% 程度短くすることに相当する。Fisheye マーカは内側に細い白線があり、これがつぶれると検出が困難となる。

5 まとめ

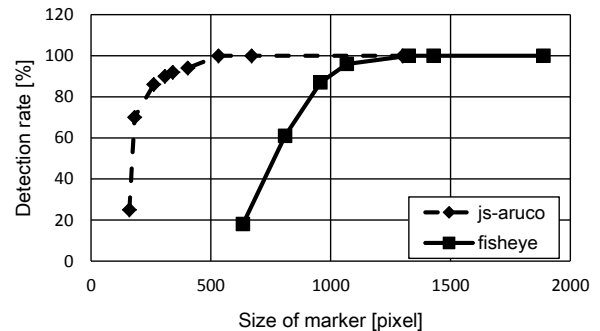
本研究では JavaScript ベースで高速なマーカ検出が可能なるマーカ型 AR システムについて検討した。Fish-eye パターンを用いたマーカによって高速な検出が可能であることを確認した。ただしマーカサイズを従来法と等しくした場合、約 4 割程度カメラをマーカに近づける必要があることがわかった。



(a) 検出時間 (PC)



(b) 検出時間 (Tablet)



(c) 検出率

図 4 実験結果

参考文献

- [1] Junaio, metaio, <http://www.junaio.com/>.
- [2] Vuforia, Qualcomm, <https://developer.qualcomm.com/mobile-development/mobile-technologies/augmented-reality>.
- [3] AR-media, <http://www.inglobetechnologies.com/>.
- [4] Unity3D, <http://unity3d.com/>.
- [5] js-aruco, <http://code.google.com/p/js-aruco/>.
- [6] ARToolkit, <http://www.hitl.washington.edu/artoolkit/>.
- [7] S.Suzuki and K.Abe, "Topological structural analysis of digitized binary images by border following," Computer Vision, Graphics, and Image Processing., Vol.30, No.1, pp.32-46, 1985.
- [8] H. Kato, M. Billinghurst, "Marker Tracking and HMD Calibration for A Video-based Augmented Reality Conferencing System," Proc. IWAR '99, pp.85-94, 1999.