

美観と頑健性を両立させた複合現実感用 半人為的幾何位置合わせマーカの研究 (第 11 報) ～PM 方式におけるデザインルール体系の再構築～

Visually Elegant and Robust Semi-Fiducials for Geometric Registration in Mixed Reality (11)
- Reconstruction of Design Rule Scheme for SFINCS-PM -

久保裕紀子, 吉田大地, 柴田史久, 木村朝子, 田村秀行

Yukiko Kubo, Daichi Yoshida, Fumihisa Shibata, Asako Kimura and Hideyuki Tamura

立命館大学大学院 理工学研究科

(〒525-8577 滋賀県草津市野路東 1-1-1)

Abstract: We have proposed a geometric registration method, called SFINCS-PM, which employs posters as markers. This paper describes reconstruction of design rule scheme for SFINCS-PM. Our method verifies and identifies SFINCS-PM posters from a camera image based on the rules about the design of the posters, called “Design Rule”. Users who use SFINCS-PM method need to provide these rules according to the scheme for “Design Rule” which we prepared. However, users are difficult to describe design of various posters as rules according to the scheme because the conventional scheme has many restrictions on designing these rules. So, we reconstructed the design rule scheme which enables the users to design posters as SFINCS-PM markers more freely. Moreover, in order to improve the efficiency of the algorithm, we introduced concurrent processing of marker detection and tracking, concurrent processing of judgment with design rules, and image processing with GPU.

Key Words: *Mixed Reality, Semi-Fiducial Marker, Geometric Registration, Tracking*

1. はじめに

我々は, SFINCS (Semi-Fiducial INvisibly Coded Symbols) [半人為的, 目立たないよう符号化された記号群の意] という概念を提案し, 美観と頑健性を両立させた半人為的マーカによる位置合わせの研究を行っている [1][2]. この概念に沿った手法の 1 つとして, ポスタをマーカとして利用する位置合わせ手法である「ポスタ埋め込み方式 (Poster Masquerade 方式, 以下, PM 方式と略す)」を提案した. PM 方式では, ポスタ型マーカを独自のルールに従ってデザインすることで, 入力画像中からの検出・同定を可能としている. しかし, 従来のポスタ型マーカのデザインに関するルール体系では, ルールの記述能力が低く, マーカとして利用できるポスタデザインに制限があった. そこで本稿では, デザインに関するルールとその設計方法を見直し, より記述能力が高いルール体系を再構築した結果について報告する. また, この再構築に合わせて, マーカの検出・同定におけるルールの判定処理にスレッドの概念を導入し, マーカ検出・同定処理とマーカトラッキングの並行処理, GPU を利用した画像処理を導入することで処理能力の向上も実現した結果についても報告する.

2. SFINCS-PM

2.1 概要

PM 方式では, 位置姿勢推定に利用するポスタ型マーカは, 独自のルールに従って事前にデザインされている. こ

れによりカメラ画像中のポスタ型マーカを検出し, 検出したポスタ型マーカの ID を同定することができる. カメラの位置姿勢は, 検出・同定したポスタ型マーカのカメラ画像中における形状から推定する. PM 方式では, ポスタ型マーカのデザインをルールとして記述するための仕組みをデザインルール体系と定義している.

2.2 従来のデザインルール体系の問題点

従来のデザインルール体系には, 配置ルールスキームと配色ルールスキームの 2 種類が定義されている [1]. 従来の方式では, このルールスキームのどちらか一方に従い, ポスタ型マーカを検出・同定するためのルールを設計する. そのため, 異なるスキームで用意されたルールを同時に利用することはできない, 範囲を指定した配色ルールを利用できないなど, デザインをルールとして記述する能力が低く, ポスタを自由にデザインすることが難しかった.

3. デザインルール体系の再構築

従来のデザインルール体系において, ルールを記述する能力が低いという問題を解決するため, PM 方式のデザインルール体系を再構築した.

3.1 用語定義

デザインルール体系の再構築にあたって用語の定義を明確にする.

[SFINCS ポスタ]

PM 方式で使用するポスタ型マーカ

[デザインルール]

SFINCS ポスタを検出・同定する為のポスタ型マーカのデザインに関する規則。デザインルールは、以降で述べる Verify ルールと ID ルールの 2 種類から構成され、複数の Verify ルールと ID ルールを組み合わせ設計する。

[Verify ルール]

カメラ画像において SFINCS ポスタとその他の領域を区別するためのデザインルール。

[ID ルール]

Verify ルールにより SFINCS ポスタと判定された場合に、その ID を同定するために用いるデザインルール。

[ルールスキーム]

デザインルールを設計するための指標。ルールスキームのうち、Verify ルール用の指標を Verify ルールスキーム、ID ルール用の指標を ID ルールスキームと呼ぶ。各ルールスキームには、Verify ルールや ID ルールを作成するための雛形が定義されている。

[ルールセット]

ルールスキームに定義されたルールの雛形にパラメータを付与することで実体化したルールの集合。ルールセットのうち、Verify ルールの集合を Verify ルールセット、ID ルールの集合を ID ルールセットと定義する。

3.2 ルールスキームとデザインルール

PM 方式におけるデザインルール体系の概念図を図 1 に示す。ルールスキームとルールセットの関係は、オブジェクト指向プログラミングにおけるクラスとインスタンスの関係に相当する。以降、3.2.1 項では Verify ルール、3.2.2 項では ID ルールについて詳述する。なお、ルールスキームに含まれている各ルールの雛形は今後拡張が可能であり、様々なルールを導入することで、様々なポスタデザインを記述可能なデザインルールが設計できる。

3.2.1 Verify ルール

PM 方式では、2 種類の Verify ルールを定義している。

- (1) ポスタの構成要素の配置を利用したルール
- (2) ポスタの配色を利用したルール

(1) のルールでは、ポスタの構成要素である画像や文字の配置をルールとして記述する。現在、記述可能な構成要素は、画像要素、1 または 2 行の文字列であるタイトル要素、複数行の文字列であるテキスト要素の 3 種類である。

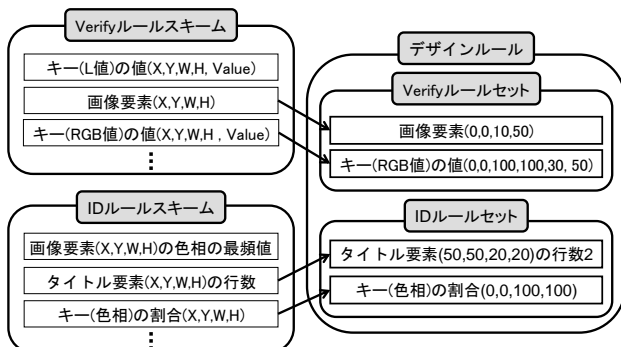


図1 デザインルール体系の概念図

(2) のルールでは、ポスタの色情報をルールとして記述する。色情報とは、ポスタ中の任意の範囲におけるヒストグラムから取得可能な色の値や度数に関する情報である。現在、記述可能な色情報は、無彩色とキー (RGB 値, 色相値, L 値) である。キーとは、ポスタ中で指定した範囲内における色ヒストグラムにおいて度数の高い色を指す。

次に Verify ルールの設計方法について説明する。まず、PM 方式の利用者は Verify ルールスキームから使用したい Verify ルールの雛形を選択する。Verify ルールスキームには (1) と (2) の両方の雛形が定義されており、組み合わせで使用することが可能である。次に、選択したルールの雛形にキーの値、構成要素の種類やルールの適用範囲等のパラメータを付与することでルールを実体化する。PM 方式の利用者は、必要に応じて複数の Verify ルールを選択し、Verify ルールセットを構築する。選択した Verify ルールの数が多いほど、SFINCS ポスタと他の領域との区別が容易になる。Verify ルールの具体例は 5 章で述べる。

Verify ルールの判定は、抽出した SFINCS ポスタの候補領域からルールに基づいて取得した各種の情報が、各 Verify ルールに適合しているか否かによって行う。定義された Verify ルールを 1 つでも満たさない場合は、その候補領域を SFINCS ポスタではないと判定する。

3.2.2 ID ルール

PM 方式では、2 種類の ID ルールを定義している。

- (1) ポスタの構成要素中の特徴を利用したルール
- (2) ポスタの配色を利用したルール

(1) のルールでは、ID ルールとしてポスタ中の構成要素の固有の特徴を記述する。特徴として、画像要素では最も度数が多い色相値、タイトル要素ではタイトルの行数と色、テキスト要素では文字の色を用いる。(2) のルールでは、ID ルールとしてポスタの色の特徴を記述する。すなわち、ポスタ中で範囲を指定し、その範囲の色ヒストグラムから抽出された各キーの色の値や度数を用いる。

ID ルールの設計は、Verify ルール同様、ID ルールスキームから任意のルールの選択し、実体化させることで行う。

ID ルールによる SFINCS ポスタの ID の同定は、定義された全ての ID ルールを SFINCS ポスタに適用した結果と事前に作成している ID データベースを照合することで行う。ID データベースは、定義された全ての ID ルールを、各 SFINCS ポスタに適用した結果を保持している。ID データベースの例を表 1 に示す。

4. アルゴリズムの改良

デザインルール体系の再構築に伴い、新たなルール体系に適した PM 方式のアルゴリズムを再設計した。デザイン

表1 ID データベースの例

	ID 1	ID 2	ID 3
ID ルール 1	True	True	False
ID ルール 2	False	True	False
ID ルール 3	False	False	True
ID ルール 4	True	False	True

ルールの判定処理にスレッドの概念を導入し、マーカ検出・同定とトラッキングの並行処理、GPU を利用した画像処理を導入したことにより、処理能力の向上を実現した。以降、4.1 節ではマーカ検出・同定とトラッキングの並行処理、4.2 節では GPU による処理について詳述する。

4.1 マルチコアによる並行処理

4.1.1 マーカ検出・同定とトラッキングの並行処理

従来の PM 方式では、SFINCS ポスタを検出・同定し、カメラの位置姿勢推定に成功した場合、次フレーム以降ではトラッキング処理のみを行うことで、処理の高速化を図っていた。しかし、トラッキング中に新たな SFINCS ポスタがカメラ画像に映った場合、検出・同定処理を行っていないため、新たな SFINCS ポスタの認識ができないことが問題であった。そこで、SFINCS ポスタ検出・同定処理とトラッキング処理を並行に実行することで、トラッキング中に新たに出現した SFINCS ポスタの検出・同定を可能にした。図 2 に SFINCS ポスタ検出・同定処理とトラッキング処理の並行処理の概念図を示す。SFINCS ポスタ検出・同定スレッドでは、カメラ画像からの SFINCS ポスタ候補領域抽出処理、その領域の正対処理と、その領域に対するデザインルールに基づいた判定処理を担当する。位置姿勢推定スレッドでは、カメラ画像中での SFINCS ポスタの形状による位置姿勢推定と、既に検出・同定した

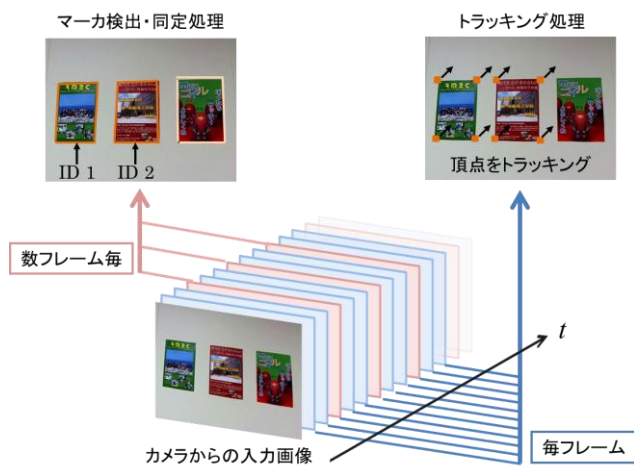


図2 並行化の概念図

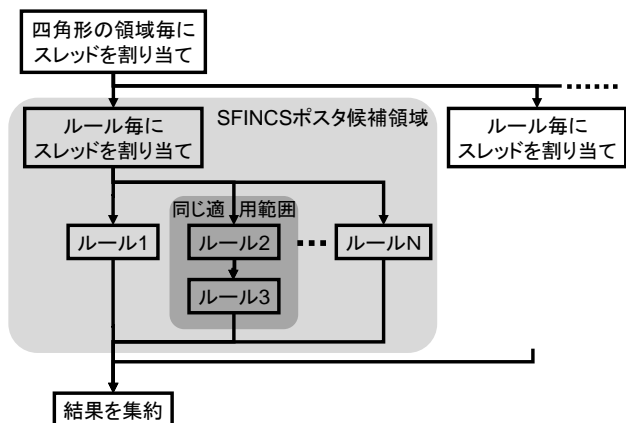


図3 判定処理のスレッド割り当て

SFINCS ポスタのトラッキングを担当する。

4.1.2 デザインルール判定の並行処理

Verify ルール・ID ルールの判定処理においても、複数のスレッドを用いることで処理効率化を図った。ルール判定における並行処理の概要を図 3 に示す。スレッドの割り当ては以下の 2 つの基準に従う。

- ・ SFINCS ポスタ候補領域ごとに割り当てる

PM 方式では、カメラ画像から抽出した複数の SFINCS ポスタ候補領域に対してデザインルールが適合するか否かを判定する。従来の PM 方式では、各領域に対する判定処理を逐次処理していたため、候補領域の増加に比例して、処理負荷も増加するといった問題があった。そこで、領域ごとにスレッドを割り当て、各領域に対する判定処理を並行に行うことで、この問題を解決した。

- ・ ルールセットに定義されたルールごとに割り当てる

デザインルールによる判定処理は、Verify ルール・ID ルールによる判定結果の集約であるが、各判定処理は独立しているため同時に処理できる。そこで、ルールごとにスレッドを割り当て、ルールセットに定義されたルールの判定処理を並行に行う。ただし、全てを並行に行うのではなく、ルールの適用範囲が同じルールは、同一スレッドで処理する。これは、ルールの適用範囲が同じ場合、他のルール判定で抽出した色ヒストグラム等の情報を流用できる場合が多く、並行に判定するよりも、処理負荷が大きいヒストグラムの生成回数を減らすことができるからである。

4.2 GPU による処理の高速化

SFINCS ポスタ検出・同定処理の中には、2 値化処理のように、各画素に対して単純な処理を施すことが多くある。この種の単純な処理を GPU によって並列処理することで処理能力の向上を実現する。今回 GPU による並列化を施した処理は、2 値化、グレースケール化、Harris オペレータによる特徴点抽出[3]、四角形領域抽出である。

5. 動作確認

デザインルール体系の再構築と、アルゴリズムの見直しにより、ルールの記述能力と処理能力が向上したことを確認する。動作環境を表 2、使用するポスタを図 4 に示す。

表2 実験環境

OS	Windows 7 Professional(x64)
CPU	Intel Core 2 Quad 2.66GHz
グラフィックボード	NVIDIA GeForce GTX 285
メモリ	4.00GB
カメラ	Logicool HD Webcam C910



(a) (b) (c)

図4 使用するポスタ

5.1 ルール記述能力の向上

デザインルール体系の再構築により、デザインルールの記述能力が向上したことを確認する。ルールの記述能力が向上したことを確認するために、(1) 従来の PM 方式で使用しているルールの記述が可能であること、及び (2) 従来の PM 方式では記述できなかったルールが記述可能であることを確認する。(1) で使用するデザインルールとして、ポストの構成要素の位置のみを利用したデザインルール A1 と、(2) で使用するデザインルールとして、配置ルールと配色ルールを併用したデザインルール A2 の 2 種類を設計した (表 3, 表 4)。デザインルール A1 では、図 4 にある (a) と (b) のポストを SFINCS ポスタとし、デザインルール A2 では図 4 にある (b) と (c) のポストを SFINCS ポスタとする。動作確認の結果を図 5 に示す。

5.2 処理能力の向上

アルゴリズムの改良により、PM 方式の処理能力が向上したことを確認する。以降、5.2.1 項では並行処理、5.2.2 項では GPU による処理について詳述する。

5.2.1 並行処理

今回、マルチコアによる並行処理により、トラッキング中に新たな SFINCS ポスタの検出・同定を可能にした。動作確認として、1 枚の SFINCS ポスタのトラッキング中に新たに SFINCS ポスタが出現した際に、それを認識できることを確認する。動作確認の結果を図 6 に示す。

5.2.2 GPU と CPU の処理速度比較

GPU による処理に導入したことにより、処理速度が向上したことを確認する。使用したポストは図 4 の 3 枚のポストを、デザインルールには A2 を用いた。また、本確認

表3 デザインルール A1

Verify ルールセット		
・タイトル位置が上部		
・画像要素の位置が中央		
ID ルールセット		
ID ルール/ポスト ID	1	2
画像要素の色相の最頻値が 190-220(青)	○	×
画像要素の色相の最頻値が 0-10(赤)	×	○

表4 デザインルール A2

Verify ルールセット		
・ポストの下部で最も多い色相値が 0-20 ∨ 340-359(赤)		
・ポスト全体の無彩色の割合が 40%以下		
ID ルールセット		
ID ルール/ポスト ID	1	2
画像要素の位置が中央	○	×
ポストの上部で最も多い色相値が 95-120(緑)	×	○



図6 複数のマーカ検出時のトラッキングの実行結果



(a)デザインルール A1 (b)デザインルール A2

図5 ルールの記述能力向上実行結果

表5 GPU・CPU によるメソッド単位での処理時間

	CPU	GPU
2 値化処理	13.05ms	7.51ms
グレースケール化処理	69.93ms	13.21ms
特徴点抽出処理	167.92ms	30.16ms
四角形領域抽出処理	27.08ms	22.03ms

表6 本手法の全体の処理時間

	CPU のみ	GPU 導入後
初期位置合わせ	454.52ms	193.77ms
トラッキング	36.08ms	34.58ms

では、同じ画像シーケンスにおける 100 フレームでの平均処理時間を計測した。メソッド単位での CPU と GPU による処理時間の違いを表 5 に示す。また、CPU のみと GPU を導入した本手法の全体の処理時間を表 6 に示す。ただし、使用する CPU がサポートするスレッド数にもよるが、ルールセットに定義されるルールの数が増えれば、それに伴って処理時間も増加する。

6. むすび

本研究では、PM 方式のポストを作成するためのデザインルールとその設計方法の再構築し、従来よりもルールの記述能力が高いことを確認した。また、この再構築に合わせて、マルチスレッドによるルールの判定処理、マーカ検出・同定処理とトラッキングの並行処理、GPU を利用した画像処理を導入することにより、処理能力を向上させた。

今後は、ポストの一部が隠れてしまうような場合やカメラの動きが速い場合でも継続可能なトラッキング機構の構築を検討している。

参考文献

- [1] 天目 他：“ポストを利用した複合現実感用幾何位置合わせ”，日本バーチャルリアリティ学会論文誌, Vol.14, No.3, pp.381 - 389, 2009.
- [2] 茂地 他：“複合現実感のための同系色ツートンカラーマーカにおける ID 符号体系と誤り検出・訂正”，日本バーチャルリアリティ学会論文誌, Vol. 16, No. 2, pp. 201 - 210, 2011.
- [3] C. Harris *et al.*: “A combined corner and edge detector,” Proc. 4th Alvey Vision Conf., pp. 147 - 151, 1988.