

卒業論文概要書

Summary of Bachelor's Thesis

学科名 Department	情報理工	氏名 Name	小林大起	指導 教員 Advisor	渡辺 裕 印 Seal
研究指導 Research guidance	オーディオビジュアル情報処理研究	学籍番号 Student ID	CD 1W130212 - 6		
研究題目 Title	ペン型デバイスを用いたインタラクティブ AR コンテンツ構築 Research for Building Interactive AR Contents using Pen-type Device				

1. まえがき

近年, Augmented Reality (AR) と呼ばれる技術に対する需要が高まっている. AR とは現実空間に存在しないオブジェクトを重畳投影し, それらを合わせた新しい環境をユーザに提供するための技術体系を指す. ナビゲーションの分野など, AR 技術を中核とする新製品も急速に普及しつつある. しかし, こうした実用段階まで到達したシステムのほとんどはシステムがユーザに情報を提供する一方通行型である. それに比べ, 投影された 3D モデルをユーザ側から操作するといった, インタラクティブなタイプは未発達である[7]

本研究では, AR とユーザの相互関係構築のために, ユーザ操作を拡張現実として空間に投影することを目的としている. 具体的には, ペン型のデバイスを用いて空間に描画を行うシステムの構築を目的とする.

2. ペン型デバイスを用いたインタラクティブ AR コンテンツ構築

現実空間で提案システムを動作させるには 3 次元空間を再構築する必要がある. これを実行するためには, エピポーラ幾何に基づいた三角測量を行う SfM という高コストアルゴリズムが必要となる. しかし, 本研究では AR コンテンツをインタラクティブに扱うため, リアルタイム性が要求される. そこで, SfM をリアルタイム性に特化させた v-SLAM を用いる. 提案システムの動作は, (1)入力デバイスの描画点の 3 次元位置を毎フレーム推定 (2)現実環境の再構築データに追加 (3)登録データを逆投影, となる. このため, 再構築データは点群形式が適している. そこで, 特徴点ベースの代表的 v-SLAM を 2 種類取り上げ, その特性評価実験を行うことでシステムへの適合性を検討した.

3. 実験結果評価

実験ではまず特徴点ベースの v-SLAM の代表である PTAM に対して, 予め撮影された映像内の特徴的なシーンに対して特徴点検出と平面再構成を行い, その傾向を分析した. 次に, 最新の特徴点ベース v-SLAM である ORB-SLAM に対し, 同様の映像を入力して特性を分析した. 最後に, 直接カメラから撮影された映像に対して ORB-SLAM による再構成を行い, 先述の傾向分析をさらに詳細化した.

4. まとめ

本研究で提案するシステムの実装には現実空間の認識系に高輝度点, 回転運動, 高フレームレートなどの条件が必須となる. 実験結果から, ORB-SLAM はこれをほぼ完全に満たし, 提案システムに最も向いている v-SLAM であることを示した.

参考文献

[7] moguraVR ポケモン GO にも応用可能か MIT, AR 内の物体が現実干渉したように見せる技術を発表,

<http://www.moguravr.com/interactive-dynamic-video-ar/>, アクセス : 2017/01/24

[16] G. Klein and D. Murray: "Parallel Tracking and Mapping for Small AR Workspaces", 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), pp.1-10 (2007)

[18] Raul Mur-Artal, J. M. M. Montiel, Juan D. Tardos, ORB-SLAM: a Versatile and Accurate Monocular SLAM System, IEEE Transactions on Robotics (Volume: 31, Issue: 5, Oct. 2015)

[20]梅澤雄司, 浜本隆之, インタラクティブな 3D 描画に向けた visual SLAM ベースの AR システム, PCSJ/IMPS2016, P-4-19

早稲田大学基幹理工学部情報理工学科

Bachelor's Thesis

卒業論文

Title
論文題目

Research for Building Interactive AR Contents using
Pen-type Device
ペン型デバイスを用いたインタラクティブ
AR コンテンツ構築

Student ID 学籍番号	1W130212-6
Name 氏名	小林 大起

Supervisor 指導教員	渡辺 裕 印
--------------------	--------

2017年2月1日

2016 年度 卒業論文

ペン型デバイスを用いたインタラクティブ AR コンテンツ

構築

Research for Building Interactive AR Contents using

Pen-type Device

指導教員 渡辺 裕 教授

早稲田大学 基幹理工学部

情報理工学科

1W130212-6

小林大起

目次

1. 序論	1
1.1. 研究背景	1
1.2. アウトライン	2
2. 理論と従来手法	4
2.1. ARにおけるマーカ方式とマーカレス方式	4
2.2. マーカレス方式における基盤手法	9
2.2.1. エピポーラ幾何	9
2.2.2. Structure from Motion の仕組み	14
2.3. カメラキャリブレーション	18
2.4. 各種 SLAM の詳論	22
2.4.1. PTAM	22
2.4.2. ORB-SLAM	23
2.5. 提案手法の参考にした類似研究	25
3. 提案手法	27
3.1. 入力形態および使用カメラ	27
3.2. カメラ外部パラメータの取得	28
3.3. ワールド・カメラ・デバイス間の座標変換	28
3.4. v-SLAM のスケール不定性の解決	30
3.5. カメラ座標系とデバイス座標系間の変換	32
3.6. 描画の実行	35
4. 実験結果	37
4.1. カメラキャリブレーション	37
4.2. 各種 SLAM による実験環境の 3 次元再構成	39
4.2.1. PTAM の実装と考察	39
4.2.2. ORB-SLAM の実装と考察	42
4.3. 実装中のシステムの構築方針	54
5. まとめと今後の方針	56
5.1. 本論文のまとめ	56
5.2. 今後の研究方針	56
6. 参考文献等一覧	58
6.1. 引用注釈	58
6.2. その他参考文献等一覧	60
図表目次	61

1.序論

1.1. 研究背景

近年、AR 技術による新たなサービス分野の開拓が盛んに目指されている。AR とは”Augmented Reality (拡張現実)”の略語であり、現実空間に存在しないオブジェクトを重畳投影し、それらを合わせた新しい環境をユーザに提供するための技術体系を指す。

現在、AR 技術が用いられるアプリケーションが急増しており、身近な例では、AR 技術を用いたゲームがある。ニンテンドー3DS には初期から AR オブジェクトを重畳させるためのカードをカメラにより認識し、その映像上に仮想オブジェクトを重ね合わせるといったソフトが提供されていた。また、2016 年に大ヒットした Niantic のソーシャルゲーム「ポケモン GO」には、カメラを通して撮影している風景にポケモンの 3D モデルを重ね合わせ、あたかも日常風景にポケモンが存在しているかのような映像を提供している。

ゲーム以外の応用分野では、医療分野において外科手術中に術野からは臓器などに隠れて見えない器官のオブジェクトを投影することで疑似的な透視を可能にし、術者がより安全性の高い執刀を行うことができるようにする試みが行われている[1]。広告分野ではカタログ、あるいはプロモーション映像内に AR によるオブジェクト投影が行われている。顧客がより製品像を理解しやすくなるなどの利点があるため、従来のものに勝る集客・販促効果を発揮することが期待されている[2]。

その他、博物館や美術館で遺跡や絶滅した古代生物の 3D モデルによる実物大復元図を広いスペースに投影する[3]、タグ付けされた作品情報を表示する、ガイド用のオブジェクトを表示することで従来の案内板よりも直感的かつ汎用性の高いナビゲーションを利用者に提供する[4]などのシステムが検討されている。特にナビゲーションに関しては、増加する外国人観光客の影響もあり、駅構内や観光地など様々な場所においてこうした革新的なシステムを導入すべく積極的な議論が行われている[5]。AR 技術は映像的なオブジェクトによる案内だけではなく種々の言語による案内情報を省スペースにて提供することが可能になる。加えて、自動車運転におけるナビゲーションにも適用されている。具体的には、従来の独立端末に案内情報を表示するタイプではなく、戦闘機の HUD (Head Up Display) のようにフロントガラスに直接情報を投影することで、それ自体を巨大なディスプレイとして扱うものが Pioneer 社などから開発・発売されている[6]。

このように、AR 技術はあらゆる分野において、革新的なサービスやコンテンツを提供できる可能性を秘めており、利用プラットフォームの普及・高度化に伴って爆発的なニーズ拡大が予想される。上記の AR 利用例の一部は、2016 年現在の段階で既に実用化されているものが少なくない。

しかしながら、こうした実用段階まで達したシステムのほとんどはシステムがユーザに情報を提供する一方通行型である。それに比べ、例えば投影された 3D モデルをユーザ側から操作するといったインタラクティブなものはまだ発展途上である[7]。すなわち、現段階ではユーザの操作を重畳された仮想オブジェクトに、特にリアルタイムで反映するといった技術は未発達といえる。拡張された現実空間との相互的な操作インタフェースを実現することは、より高度な AR コンテンツ、あるいはその発展形である MR (Mixed Reality; 複合現実) への基盤となる。

本研究では AR とユーザの相互関係構築のために、ユーザ操作を拡張現実として空間に投影することを主眼目的として設定した。具体的には、ペン型のデバイスを用いて空間に描画を行う処理を目的とする。このシステムによりユーザが自在に空間上にオブジェクトを生成できれば、それを様々な用途に応用することが期待できる。

1.2. アウトライン

本論文のアウトラインを以下に示す。

- ・ 第 1 章 序論
 - 1 節：研究背景
 - 2 節：アウトライン

- ・ 第 2 章 理論と従来手法
 - 1 節：AR におけるマーカ方式とマーカレス方式
 - 2 節：マーカレス方式における基盤手法
 - 3 節：カメラキャリブレーション
 - 4 節：各種 SLAM の詳論
 - 5 節：提案手法の参考にした類似研究

- 第 3 章 提案手法
 - 1 節：入力形態および使用カメラ
 - 2 節：カメラ外部パラメータの取得
 - 3 節：ワールド・カメラ・デバイス間の座標変換
 - 4 節：v-SLAM のスケール不定性の解決
 - 5 節：カメラ座標系とデバイス座標系間の変換
 - 6 節：描画の実行

- 第 4 章 実験結果
 - 1 節：カメラキャリブレーションの結果
 - 2 節：各種 SLAM による実験環境の 3 次元再構成
 - 3 節：実装中のシステムの構築方針

- 第 5 章 まとめと今後の方針
 - 1 節：本論文のまとめ
 - 2 節：今後の研究方針

- 第 6 章 参考文献等一覧
 - 1 節：引用注
 - 2 節：その他参考文献

2.理論と従来手法

2.1. ARにおけるマーカ方式とマーカレス方式

本節では、現行のAR技術を実現するための手法を二つに大別し、それぞれについて説明する。

現在、ARオブジェクトを重畳するための手法として用いられているものは、その位置認識による手法から2種に分類できる。一つはマーカ方式と呼ばれる手法である。これはオブジェクトを重畳する位置にあらかじめマーカと呼ばれる認識用の図形を設置し、それをシステムが認識することで3次元空間上の位置を特定すると同時にオブジェクトを投影する。これに対し、位置認識用のマーカを用いず、撮影された3次元空間の映像のみからその構造を把握し、オブジェクトを重畳する位置をシステムが自己推定する手法がマーカレス方式である。

マーカ方式のARでは、基本的に用いられるマーカは四角形である。一般的な形式は、各辺を黒く塗ったものをベースに、内側部分に様々な図形が描かれているものである。

3次元空間内に置かれたマーカが撮影された映像内に映り、システムに認識されると、ソフトウェア内部においてまず二値化などの閾値変換によりマーカの黒色部分が抽出される。これにより、マーカの輪郭および4個のコーナー部分が検出され、システムに認識される処理後のマーカ画像は距離に比例した面積を持つ、視点角度に応じた台形となる。

次に、この台形と本来のマーカ画像を比較することにより、画面内に表示するオブジェクトの座標系を撮影されている現実空間の視点と一致させる。最後にマーカ内部の図形をパターンマッチングにより認識することで、投影すべきオブジェクトをシステムが選択し、計算された距離・角度で撮影画像内に重畳表示する。

以上の処理を毎フレーム行うことで、仮想のオブジェクトが投影されたAR画像がユーザに提供される。図2.1にマーカの実例、図2.2に上記の処理を図にまとめたものを提示する。

図 2.1 典型的な AR マーカの具体例[8]

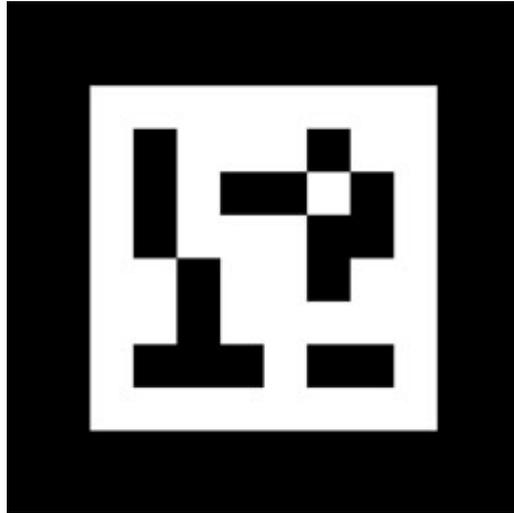
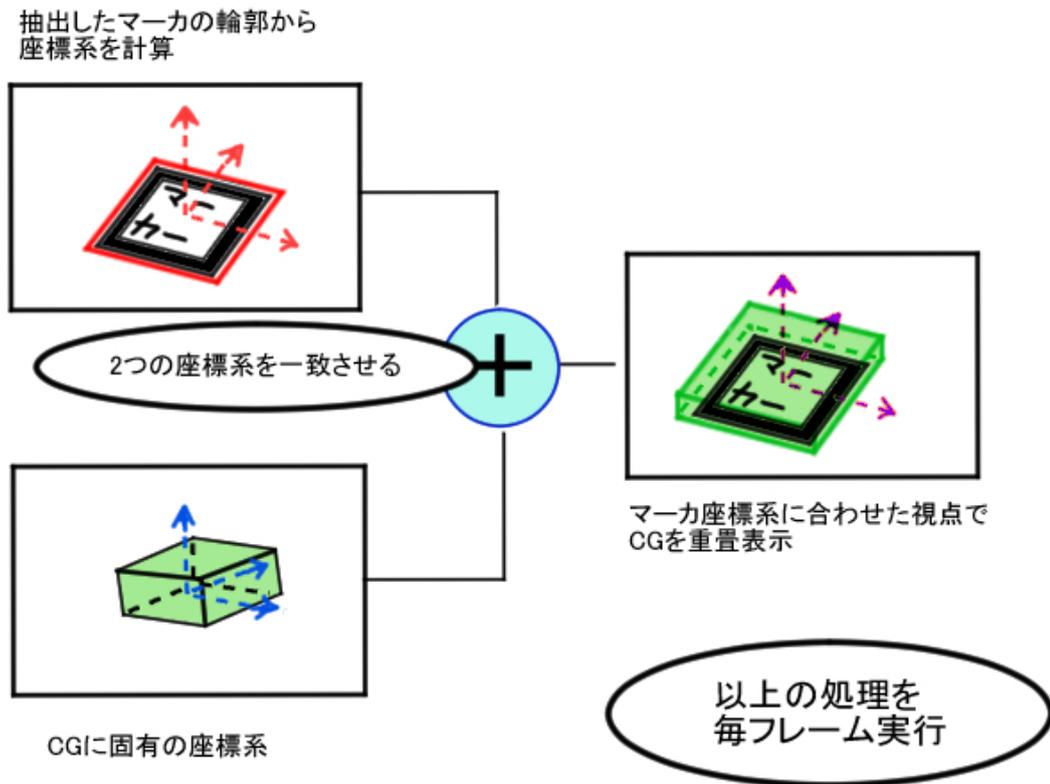


図 2.2 マーカ式 AR の処理手順[9]



一方、マーカレス方式を用いる場合、システムに提供される情報は撮影された 3 次元空間の映像のみとなるため、システムには映像のみから平面検出などの処理を行う必要が生じる。この場合、3 次元空間を撮影した映像から部屋の窓や机の隅、エッジなどの特徴的な部分を検出する必要が生じる。換言すると、このように撮影された映像から直接得られた特徴点群をマーカとして用いるものがマーカレス AR といえる。

マーカ有り AR と類似の環境下で使用する場合 (具体的には机や壁面など単純かつ比較的広い平面上にのみオブジェクトを投影する場合) はこれらの特徴点からそのまま座標計算を行うことも考えられるが、より複雑な環境下に自然な重畳投影を行う場合には、3 次元空間全体の構造をシステムが認識する必要がある。これを可能とするために用いられる技術が SfM: Structure from Motion と呼ばれる技術である。

SfM はマッチムーブ、すなわち撮影済みの映像に CG をリアルに合成するために用いられていた技術で、カメラの視点移動と映像内の 3 次元空間構成とを同時に算出する。中核処理は、映像を構成するフレーム画像のうち、同じ被写体を別角度から映した 2 枚を選び、それぞれの特徴点を抽出した後、これらの特徴点をエピポーラ幾何と呼ばれるステレオカメラ用の計算で分析することで実現される。

このように、マーカレス AR では 3 次元空間を特徴点、あるいは画素などのそれに準ずる手がかりによって再構成する計算が必須となる。

マーカ有り AR とマーカレス AR では、それぞれ一長一短のメリット・デメリットが存在する。両者を以下で比較する。

マーカ方式では、重畳位置の指定があらかじめ行われ、システム側は周囲の環境まで逆算することなくオブジェクトの重畳が可能である。そのため、実装アルゴリズムも比較的単純なもので十分となり、豊富なライブラリやツールが広く公開され、これらの敷居の低さから趣味・商用問わず広く普及しており、カメラを搭載した計算機であれば十分に動作するソフトウェアが多数公開されている。つまり、一般的な構成のモバイル端末なら AR コンテンツを利用できるということを意味する。したがって、必要なハードウェア的制約の観点からもマーカレス AR に勝る。

また、近年の画像処理技術の高度化に伴ってマーカ自体の自由度も高くなり、さまざまなテクスチャをマーカとして利用できるようになってきている。これにより、広告などにマーカを埋め込むのみならず、広告自体をマーカとして利用す

るケースも増えている。この点は、一見するとマーカレス AR より初歩的ではあるが、ユーザに対して AR コンテンツの存在を分かり易く示すという側面ではむしろマーカレス AR に優位となりうる。

一方、マーカ方式 AR のデメリットは対象となる現実空間にあらかじめマーカを準備する必要がある点に集約される。したがって、マーカが設置できない微細、あるいは複雑な形状の現実環境に適用することは難しい。また、前述した自由度の高いマーカを用いる場合でも、認識用のアルゴリズムに対象テクスチャを登録する作業はシステム構築者がその都度行う必要がある。さらに、AR オブジェクトの選択にテクスチャのパターン認識を要する以上、テクスチャが潰れて判別不能になる遠距離において適用することは困難である。これらの問題点は、初見の環境に AR を構築するケースにおいて非常に大きい制約となり、汎用性でマーカレス方式に劣るということに他ならない。

マーカレス AR の優位点は、上記のマーカ方式の問題点を解決できる点である。現実空間の認識はカメラ画像にのみ依存するため、初見の環境においても特徴点検出が適切になされる限りは性能が期待できる。複雑な形状の物体に重畳表示を行うことも特徴点利用のアルゴリズム次第では可能である。理論的にはカメラ画像に映る範囲全体に対して重畳できるため、マーカ方式や測距センサを用いる場合よりも距離に対してロバストである。

こうしたメリットが活かされる具体例として、序章で述べた外科手術への応用が挙げられる。患者の体内にマーカを事前に設置することはできないが、マーカレス方式なら重畳を行うことが可能である。

また、距離に対してロバストかつ 3 次元再構成を行うため、カーナビゲーションシステムなどに対してもマーカ方式より高度な機能を実装することが可能となる。さらに、3 次元再構成は AR のみならず、ロボティクスビジョンの分野でも盛んに応用が試みられている基幹技術であり、ドローンなど様々な最先端技術とマーカレス AR の相互転用による発展が期待される。

このように、あらゆる分野への AR の応用が目指されている現状を鑑みると、後述する実装の困難さを考慮してもなおマーカレス AR の汎用性は極めて優れた特徴といえる。

一方、マーカレス AR のデメリットとしては計算量の多さ・複雑さとハードウェア的制約の 2 点が挙げられる。3 次元再構成の主要技術である SfM は映

像内のシーンごとに処理が実行されるため、計算量が膨大である。そのため後述する PTAM が発表されるまで AR などのリアルタイム性を重視するコンテンツでは利用することが困難であった。また、リアルタイム性の問題を除いても、この PTAM の発表以前は距離センサ付きの RGB-D カメラなど高額な専用機器や処理能力の高い計算機が必須であった。SfM 用ソフトウェアも非常に高額なものが多く、マーカ式 AR よりハードウェア・ソフトウェア両面で極めて制約が大きかった。PTAM に代表される高速なアルゴリズム、および Kinect などの廉価な実験機器の発売によりこうした問題点はある程度解決されたが、それでもなおマーカ方式に比べライブラリ面での普及が遅れ、環境構築を困難にする一因となっている。

参考として、マーカ AR とマーカレス AR の現段階での具体例を次の図 2.3、2.4 に示す。前者はニンテンドー3DS の AR カード、後者はプレイステーション VITA の AR モードである。

図 2.3 ニンテンドー3DS のマーカ式 AR 機能[10]



図 2.4 プレイステーション VITA のマーカレス AR 機能[11]



本節では、マーカ AR、マーカレス AR のそれぞれについて基本構造の概要、およびそれぞれのメリット・デメリットを述べた。序章で述べたように、本研究はインタラクティブな AR コンテンツの構築を目指すものであり、主にマーカレス AR の手法に注目する。次節以降では、3 次元再構成技術やマーカレス AR の中核理論を説明する。

2.2. マーカレス方式における基盤手法

本節以降では、前節で説明したマーカレス AR に必要となる 3 次元再構成技術について述べる。

2.2.1. エピポーラ幾何

エピポーラ幾何とは、ステレオカメラ、あるいは二つの異なるカメラ画像から 3 次元的情報を復元するための幾何学である。人間の両眼による立体視と同一の原理であり、ある被写体を異なる視点から撮影した 2 枚の画像から奥行き情報を得る。

図 2.5 エピポーラ幾何概略図[12]

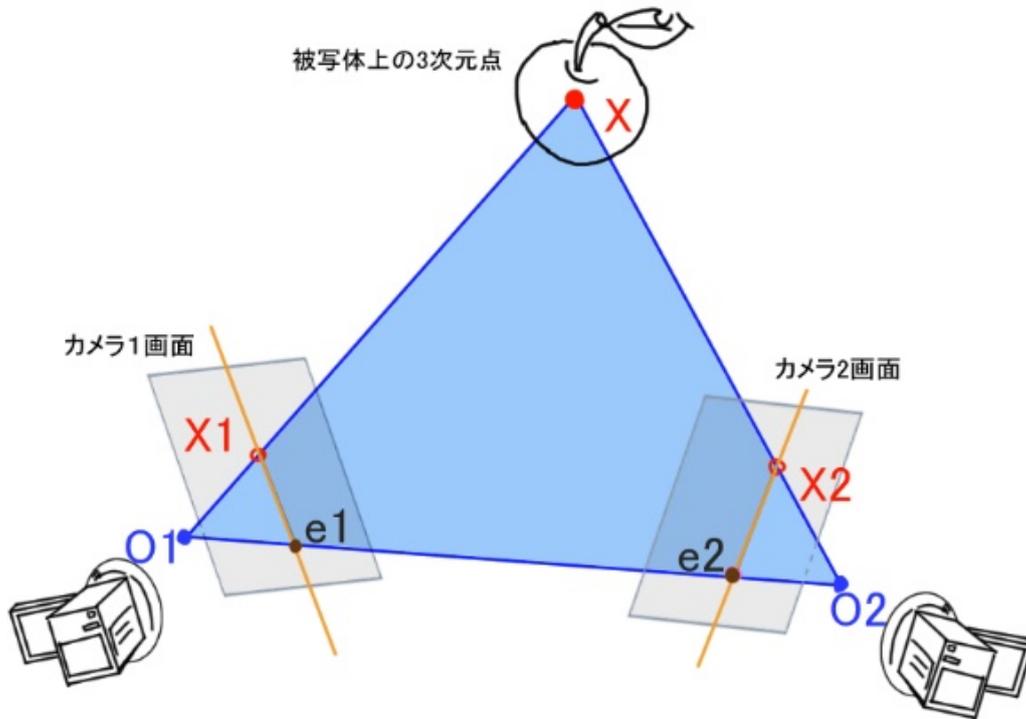


図 2.5 のように被写体、およびそれを写す 2 つのカメラ画像を考える。このとき、 X は被写体上の 3 次元点、 X_1 、 X_2 はそれぞれのカメラ画面に映った X の像であり、カメラの位置（光学的中心）はそれぞれ O_1 、 O_2 である。また、点 e_1 、 e_2 はそれぞれのカメラ画面内においてもう一方のカメラの中心が映る点であり、エピポールと呼ばれる。 X, O_1, O_2 を含む平面をエピポーラ平面と呼ぶ。また、直線 X_1e_1 、 X_2e_2 をエピポーラ線と呼び、これらはエピポーラ平面とそれぞれのカメラ画面の交線となる。左のカメラを基準に考えると、その画面に映されている点 X は右カメラ画面のエピポーラ線上のどこかに映されることが図から分かる。

ここで、上記のエピポーラ線上への 3 次元点の投影が何によって行われるかを考えると、それは基礎行列と呼ばれる 3×3 行列となることが知られている。この行列を F とすると、 F にはカメラの内外両方の情報が含まれることになる。したがって、 F を求めることがカメラ運動の逆算、すなわち SfM に直結する。

行列 F は数学的には以下のように計算される。いま、 X_1 、 X_2 の画面上

の位置ベクトルを同次座標系のベクトル \bar{x}_1, \bar{x}_2 とする。なお、同次座標とは、N次元空間の点の全ての移動・要素を表すために N+1次元の座標を用いて位置を定義する座標系で、本項の場合は3次元ベクトルになる。

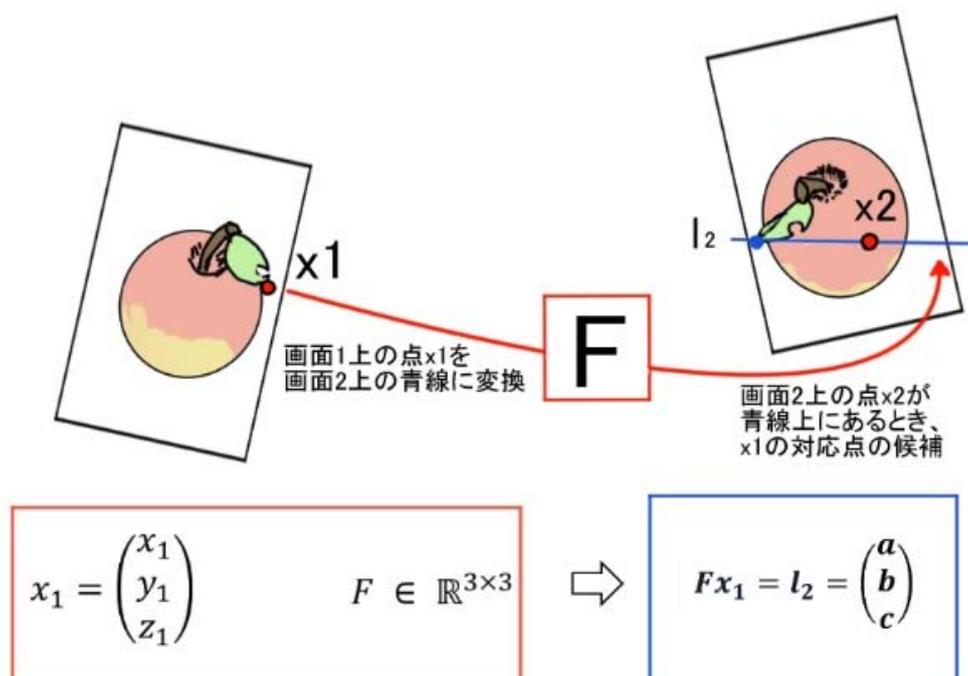
この \bar{x}_1, \bar{x}_2 を用いて前述の「三次元点 X がカメラ 1 画面上に点 X_1 として映ったとき、同じ点 X がカメラ 2 画面上のエピポーラ線上のどこか一点に対応して投影される」性質を表すと、以下の関係式として表現できる。

$$\bar{x}_2^T F \bar{x}_1 = 0 \quad (2.1)$$

この式はエピポーラ拘束式と呼ばれ、実際の3次元点の位置ベクトルによらず各画面上と基礎行列のみで定義される関係式である。F が 3×3 行列、 \bar{x}_1 は3次元ベクトルであるため、 $F \bar{x}_1 = [a, b, c]$ と表せる。エピポーラ線は直線であるので、直線の方程式の定義から式(2.2)のようにして拘束式の成立を証明できる。以上を図 2.6 に示す。

$$\bar{x}_2^T F \bar{x}_1 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} [a, b, c] = ax_2 + by_2 + cz_2 = 0 \quad (2.2)$$

図 2.6 エピポーラ拘束の概要図[12]



このエピポーラ拘束から F を求めるとき、一般的に、8点アルゴリズム

と呼ばれる計算が用いられる。式(2.1)を画面上のそれぞれの対応点 i に対してとると、

$$[x_2^i, y_2^i, w_2^i] \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_1^i \\ y_1^i \\ w_1^i \end{bmatrix} = 0 \quad (2.3)$$

となる。なお、図 2.6 までは直線の方程式を示すためあえて z を用いた座標表記を採用したが、本来同次座標における拡張次数部分に入るものはスケール定数である。そのため、これ以降は z ではなく w を用いて表す。

ここで、未知変数となる行列 F を求めるため、 n 個の対応点に対して式(2.3)を立て、それらを連立方程式とすることで F の要素をまとめ上げることを考える。すなわち、以下を立式する。

$$\begin{bmatrix} x_2^1 x_1^1 & x_2^1 y_1^1 & x_2^1 w_1^1 & \cdots & w_2^1 w_1^1 \\ x_2^2 x_1^2 & x_2^2 y_1^2 & x_2^2 w_1^2 & \cdots & w_2^2 w_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_2^n x_1^n & x_2^n y_1^n & x_2^n w_1^n & \cdots & w_2^n w_1^n \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ \vdots \\ f_{33} \end{bmatrix} = A \vec{f} \quad (2.4)$$

$$= 0$$

F は 3×3 行列ではあるものの、スケール不変という特性を持つ。したがって、式(2.4)を解くために必要となる $n=8$ である。すなわち、相異なる対応点を 8 個選択すれば F を求められる。

次に、式(2.4)に対して SVD : 特異値分解を施す。これは、ある行列 A を

$$A = U \Sigma V \quad (2.5)$$

なる形に変換する計算である。ただし、 U は $m \times m$ の直交行列、 Σ は非対角成分は 0、対角成分は非負で大きさの順に並んだ行列、 V は $n \times n$ の直交行列である。これを式(2.4)の f に対して適用する。

図 2.7 行列 Σ の構造[13]

σ_1 \cdot \cdot \cdot σ_r	0
0	0

$$\sigma_1 \geq \dots \geq \sigma_r \geq 0$$

ここで、 F のランクが 2 であることを利用すると、 Σ の最小対角要素を 0 とすることが可能である。以上の計算により、 F が導出される。

求められた F はカメラ外部・内部両方の情報を複合したものであることは冒頭で述べた。したがって、この F をさらに分解することで外部・内部の情報を切り分けることが可能である。

いま、カメラ外部情報を有する行列を基本行列 E と定義する。2 つのカメラ内部パラメータをそれぞれ K_1, K_2 とすると、

$$E = K_2^T F K_1 \tag{2.6}$$

なる式が成り立つ。ここで、 K はカメラ内部パラメータであり、後述するカメラキャリブレーションによって求められるため、ここでは既知のデータとして扱う。(同一カメラを用いる場合、 K は 1 種類である。)

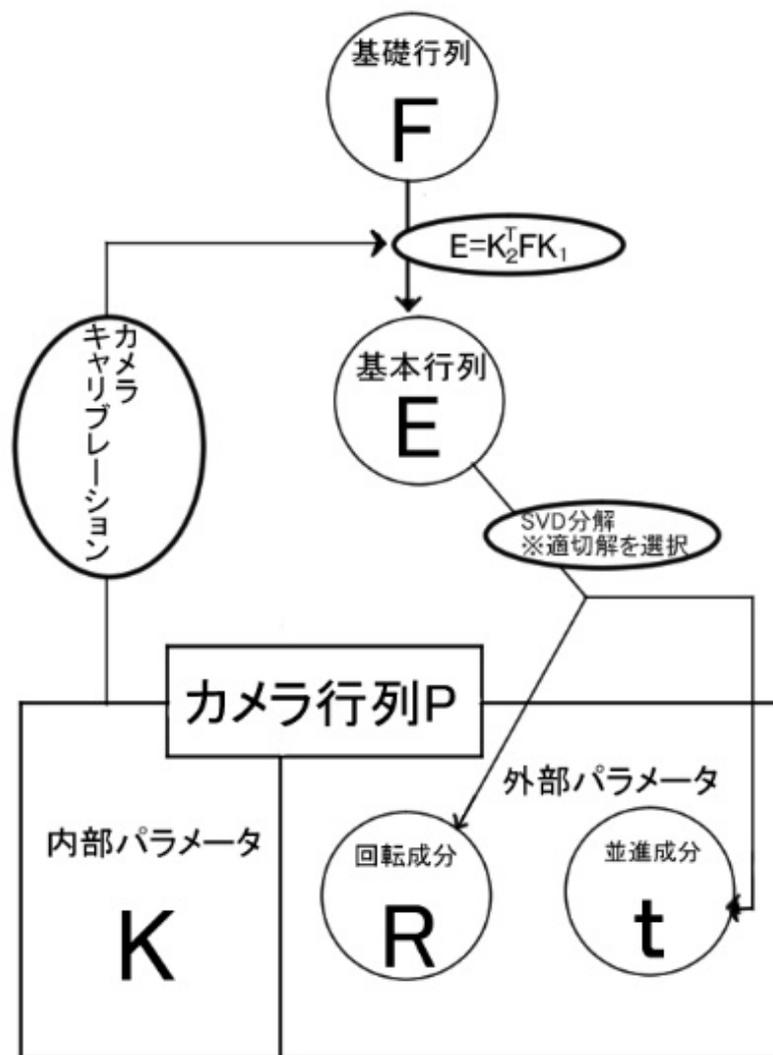
この E に対して、式(2.4)と同様に SVD による分解を施すと、カメラ外部パラメータが推定され、カメラ行列 P が外部・内部のパラメータの積の形で求められる。(なお、得られる解は四つあるが、カメラ画面と被写体の位置関係が現実空間に即したものとなるのは 1 種類であるため、一意に絞ることが可能である。)

このカメラ行列は現実空間の座標を基準にとった座標系 (ワールド座標系) を導入した際、ワールド座標上の点をカメラ画面の座標系へと直接変換する行列であり、この P を求めることがエピポラ幾何の最終目的である。カメラ行列 P の詳細は 2.3 節にて後述する。

2.2.2. Structure from Motion の仕組み

2.2.1.項において導出したカメラ行列 P を用いて現実世界の座標系を再現する手法が SfM である。図 2.8 に導出のフローチャートを示す。

図 2.8 カメラ行列導出のフローチャート[12]



このような手順で求めたそれぞれのカメラ行列 P_1, P_2 を用いて、前項の図 2.6 における 3 次元点 X のカメラ画面への投影を式で表す。簡単のため、点 X の世界座標系における位置ベクトルを \vec{X} 、それぞれのカメラ画面における x_1, x_2 の画像座標系での位置ベクトルをそれぞれ \vec{x}_1, \vec{x}_2 とす

る。(これらの座標系の差異については 2.3 節を参照)

これらを用いて立式すると、式(2.7)となる。

$$\begin{aligned}\lambda_1 \vec{x}_1 &= P_1 \vec{X} \\ \lambda_2 \vec{x}_2 &= P_2 \vec{X}\end{aligned}\tag{2.7}$$

式(2.7)を変形して、以下を得る。

$$\begin{bmatrix} P_1 & -\vec{x}_1 & 0 \\ P_2 & 0 & -\vec{x}_2 \end{bmatrix} \begin{bmatrix} \vec{X} \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = 0\tag{2.8}$$

式(2.8)は(2.4)と同様、 $A \vec{x} = 0$ の形式である。したがって、SVD による分解を \vec{x} に適用することで、3次元ベクトル \vec{X} を復元できる (こちらも前節と同様に、複数出現する解の中から現実空間とカメラの位置関係に合致するものを選ぶ)。すなわち、カメラ画像から 3次元空間の立体情報を推定することが可能となる。

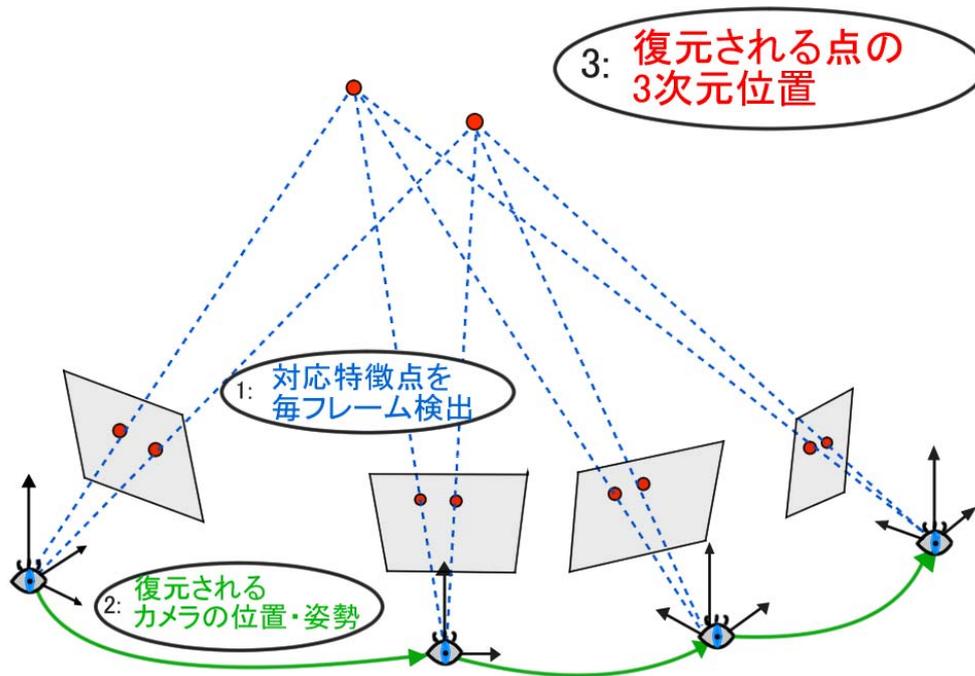
この計算は三角測量と呼ばれ、SfM による 3次元再構成の最小単位を表している。

これら一連の計算の流れを異なる 2 フレームの映像の組ごとに対して行い、映像を構成する全フレームに対して処理が完了されるまで繰り返す処理が SfM である。

実際には、図 2.8 に示したフローチャート冒頭の基礎行列 F の算出には毎回誤差が生じる (再投影誤差)。そのまま逐次算出し続けた場合、再投影誤差は一方向的に広がり続ける。これを防ぐため、実際には算出用のフレームの組をある程度蓄えておき、それらを全て用いて一度に基礎行列を計算する方式が採られている。

以上の内容を図 2.9 にまとめる。なお、純粋な SfM では図中の特徴点検出の部分には再投影誤差を最小化するため、各特徴点の誤差を予測しつつ再帰的計算でそれらを最小化する bundle adjustment と呼ばれる高コストの最適化が施されている。

図 2.9 最終的な SfM の動作[14]



3次元再構成技術において SfM は主要技術である。しかし、元来 SfM はマッチムーブと呼ばれる分野での利用が主であり、映像に CG を後から重畳することを目的としている。この用途では事前に全ての映像情報が提供されている前提が成立し、あるフレームにおいて CG を重畳するとき、システムはそのフレームより後の映像情報も使用して計算することが可能である。つまり、重畳計算に過去・現在のみならず未来の情報も用いることが可能である。また、最終的にユーザに提供する映像を作成するためにかかる時間的な制約がない。

一方、AR では現実空間を拡張してユーザに提供するという関係上、リアルタイム性が重視されることになる。したがって、純粋な SfM のように未来の情報を用いるなど、計算量を増やして重畳精度を迫及することは不可能である。

こうした SfM の問題点を解決するため、SfM と共通する計算手法を用いてリアルタイム処理を行う技術が SLAM: Simultaneous Localization And Mapping である。このうちマーカレス AR に用いられるものはカメラベースの手法になるため、一般に Visual-SLAM と呼ばれる。

これは従来ロボティクス分野で用いられている技術であり、「未知環境にて動作する自走式ロボットに搭載したカメラで周囲の様子を撮影し、その情報を

地図化すると同時にその中での自己位置を推定する」という一連の動作を連続的に繰り返す。

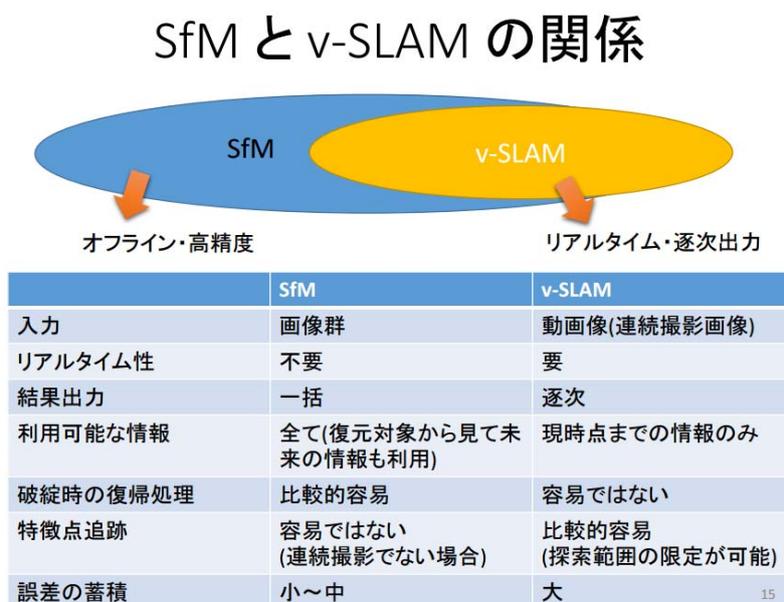
ロボットと人間が保持するカメラとでは本体に対する視点位置が固定されていたり、車輪の回転数などの動きの規則性に関する扱いやすい情報を記録することが可能であったり、といった相異が存在する。そのため、(人間が手に持つデバイスで実行する用途の) マーカレス AR ではロボットに用いる場合に比べればシステムに提供できるデータが少なく、高精度化の手段が限られる。

しかし、近年発表された SLAM はドローンなど 3 次元的な動きを行うロボットに搭載することも考慮し、視点変更や動きの複雑さに対するロバストさを追求したものが多い。また、同様の理由で、従来のようにオムニビューカメラなど特殊かつ価格・重量両面でコストの高い計測機器の利用を前提とせず、かつソフトウェアを動作させるための計算機の処理能力を小型端末クラスに抑えた手法が盛んに研究されている。

これらの傾向により、近年の SLAM はマーカレス AR に要求されるアルゴリズムの動作条件と一致するものが増えている。

参考として、奈良先端科学技術大学院の佐藤智和氏によってまとめられた SLAM と SfM の比較を図 2.10 に示す。

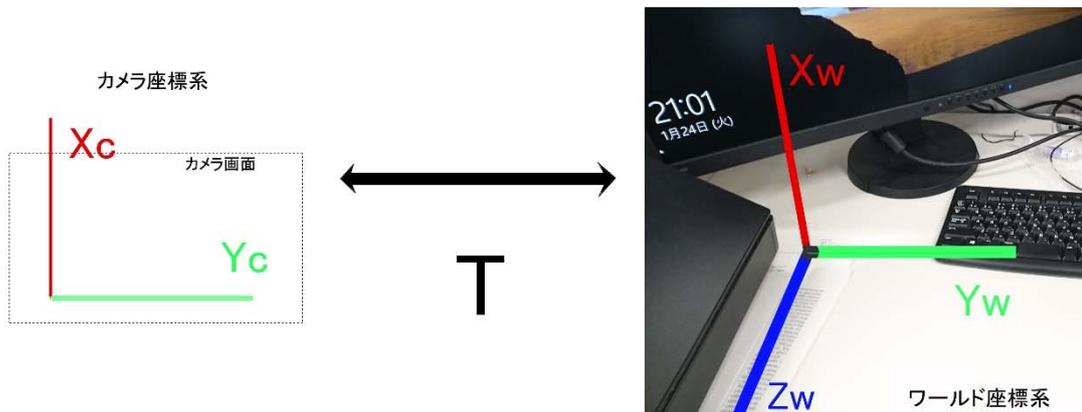
図 2.10 SLAM と SfM の比較[15]



2.3. カメラキャリブレーション

カメラとは現実世界の画像を平面に投影するためのデバイスであり、そのためには基準となる座標系が必要となる。そこで、現実空間を表す 3 次元座標をワールド座標、カメラ画面を表す座標をカメラ座標と定義した上で、現実空間上の点がどのようにカメラ画面に投影されるかを考える。なお、これらの座標は同次座標系である。

図 2.11 ワールド座標とカメラ座標、およびその変換



図中にある T は 2.2 節で述べたカメラ行列内の外的要素を格納した行列であり、カメラ外部パラメータと呼ばれる。この T は回転を表す成分と並進運動を表す成分から成り、それぞれを R 、 t という行列にまとめることで以下のように表現できる。

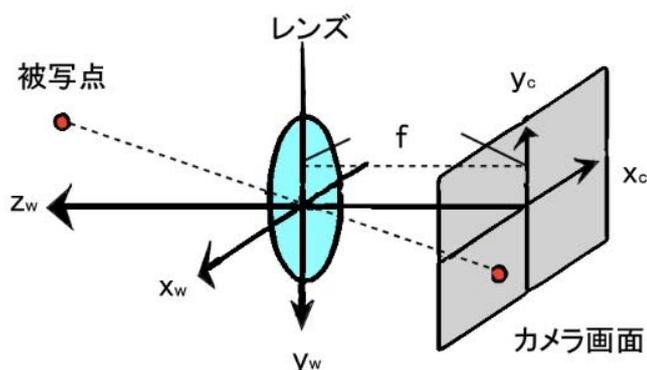
$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & t \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (2.9)$$

ここで、 R は 9 自由度、 t は 3 自由度であるが、回転成分をそれぞれの軸方向にのみに制限することで自由度を 3 自由度に落とすことができる。よって、この T 全体の自由度は 6 である。式(2.9)をワールド座標内の点に乗算することで視点移動による実世界上の点の相対的な移動を表せる。(エピポラ幾何において述べた基本行列 E はこれらを用いて $E = RT$ と定義される。すなわち、 E の SVD 分解によって回転成分と並進成分を分離できる。)

行列 T は視点移動によって時々刻々と変化する。この値を逐次計算することが 2.1~2.2 節で述べたエピポラ幾何・SfM の目的である。次に、同節で述べたカメラ内部パラメータについて述べる。

現実空間に静止している被写体を静止カメラ画面に投影する場合の理想的モデルを考える。これは、スケール値 λ 、各軸方向への焦点距離の成分 f_x , f_y を用いて以下のように表せる。

図 2.12 理想的な投影モデル



$$\lambda \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2.10)$$

しかし、実際のカメラ画像にはアスペクト比 a 、スキュー比 sq 、画像中心 c のカメラ座標といったパラメータにより撮影された画像に歪みが生じる。これらを是正するため、式(2.10)の投影モデルにそれぞれのパラメータを追加表現すると(2.11)となる。

$$\lambda \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sq & c_x \\ 0 & a & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2.11)$$

ここで、 f はカメラに用いるレンズによって規定され、現在の工業規格精度から x_c , y_c 成分が等しくなるとみなせる。したがって式(2.11)内に存在する未知数は5個となる。これらをまとめてカメラ内部パラメータと呼ぶ。(つまり、式(2.11)の行列部分が該当する。)

以上により一般的な投影モデルが求められたため、カメラが移動する場合を含めた計算式を導出できる。カメラ移動による3次元点の相対的な位置変化はワールド座標にカメラ外部パラメータを乗算することで変換可能であるた

め、式(2.11)に式(2.9)を代入すると式(2.12)を得る。

$$\lambda \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sq & c_x \\ 0 & a & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2.12)$$

式(2.12)は「カメラの製品的特性による歪み、およびその位置姿勢を考慮した現実空間内の被写体の見え方」を表し、エピポーラ幾何や SfM が目的とする計算結果と一致する。すなわち、2.2 節の理論は式(2.12)を行列の SVD 分解によって求めることに帰着する。

2.1~2.2 節で述べた理論は、式(2.12)右辺のカメラ外部パラメータ取得を可能とする。そのためには式(2.12)内のカメラ内部パラメータ部分が既知という前提が成り立つ。

カメラ内部パラメータを構成する未知数はカメラが組み立てられたときに規定され、撮影者が故意に変更を加えない限り以降変化することはない。これは、一度でもカメラ内部パラメータを推定できればそれを定数として扱えることを意味する。したがって、以降の本節ではカメラ内部パラメータの取得方法について述べる。

カメラ内部パラメータを算出するため、式(2.12)を展開・整理して式(2.13)を得る。なお、この計算を行う時点ではまだどの未知数も算出されていないため、現実世界上の位置ベクトルに乗算する行列要素を全て C_{ij} と置き換えて略記した。

$$\lambda \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2.13)$$

式(2.13)を展開すると、3本の連立方程式が得られる。そのうち1つは $\lambda = \dots$ であるため、これを残り2本に代入することで λ が消去できる。よって、

$$\begin{aligned} C_{31}x_w x_c + C_{32}y_w x_c + C_{33}z_w x_c + x_c \\ &= C_{11}x_w + C_{12}y_w + C_{13}z_w + C_{14} \\ C_{31}x_w y_c + C_{32}y_w y_c + C_{33}z_w y_c + y_c \\ &= C_{21}x_w + C_{22}y_w + C_{23}z_w + C_{24} \end{aligned} \quad (2.14)$$

を得る。この式(2.14)における未知数の数はカメラ内部・外部パラメータの自由度の和となるため、11である。

式(2.14)が成り立つことを考慮して、カメラによって撮影された外部環境の

映像内から対応する特徴点を n 個見つけ、その各座標データを式(2.14)の対応変数に代入すると、未知数 11、式 $2n$ 本の連立方程式を得る。これを行列とベクトルによる表記に直すと以下が得られる。

$$\begin{bmatrix} x_{W1} & y_{W1} & z_{W1} & 1 & 0 & 0 & 0 & 0 & -x_{W1}x_{C1} & -y_{W1}x_{C1} & -z_{W1}x_{C1} \\ 0 & 0 & 0 & 0 & x_{W1} & y_{W1} & z_{W1} & 1 & -x_{W1}y_{C1} & -y_{W1}y_{C1} & -z_{W1}y_{C1} \\ \vdots & \vdots \\ x_{Wn} & y_{Wn} & z_{Wn} & 1 & 0 & 0 & 0 & 0 & -x_{Wn}x_{Cn} & -y_{Wn}x_{Cn} & -z_{Wn}x_{Cn} \\ 0 & 0 & 0 & 0 & x_{Wn} & y_{Wn} & z_{Wn} & 1 & -x_{Wn}y_{Cn} & -y_{Wn}y_{Cn} & -z_{Wn}y_{Cn} \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{12} \\ C_{13} \\ C_{14} \\ \vdots \\ C_{21} \\ \vdots \\ C_{33} \end{bmatrix} = \begin{bmatrix} x_{C1} \\ y_{C1} \\ \vdots \\ x_{Cn} \\ y_{Cn} \end{bmatrix} \quad (2.15)$$

以上の式を簡単のために

$$A \vec{C} = \vec{c} \quad (2.16)$$

とすると、

$$\vec{C} = (A^T A)^{-1} A^T \vec{c} \quad (2.17)$$

が成り立つ。なお、(2.25)は行列計算への最小二乗法の応用によって導かれる。式(2.25)を計算システムによって解くことは回帰分析により画像上の対応点間の距離二乗和を最小化することを意味する。この解法は様々な手法が存在するが、前節で述べた **bundle adjustment** もここで用いる非線形最適化計算である。

以上により、行列 A の疑似的な逆行列 $(A^T A)^{-1} A^T$ が算出され、それを用いてカメラパラメータ \vec{C} が求められる。得られた \vec{C} の各要素を式(2.13)に代入すると、

$$\lambda \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \end{bmatrix} \begin{bmatrix} x_W \\ y_W \\ z_W \\ 1 \end{bmatrix} \quad (2.18)$$

となる。ただし、(2.21)と異なりここでの C_{ij} は全て既知定数である。式(2.26)右辺のカメラ行列 P のうち、左 3×3 をグラム・シュミットの正規直交化法によって直交化することで式(2.27)の形に分解可能である。

$$\lambda \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = \begin{bmatrix} f & sq \cdot f & c_x \\ 0 & a \cdot f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x_W \\ y_W \\ z_W \\ 1 \end{bmatrix} \quad (2.19)$$

式(2.12)と比べると、同次座標とするために行列 R 、 t に追加した部分が削れているものの、未知数が全て求められていることが分かる。

式(2.19)の右辺左側の 3×3 行列はカメラ内部パラメータであり、先述の通り同じカメラを用いる限り全てのアプリケーションに対して不変である。

本節で述べた外部画像によるカメラ内部パラメータを推定する処理をカメラキャリブレーションと呼ぶ。なお、式(2.19)右辺右側の 3×4 行列はあくまで

キャリブレーション時のみのカメラの位置姿勢を表す外部パラメータである。しかし、他のシーンに対してもカメラ内部パラメータの不変性から、キャリブレーションを一度でも事前実行しておけば理論的には SfM によってあらゆる映像間でのカメラの位置姿勢を逆算できる。

2.4. 各種 SLAM の詳論

本節では、リアルタイム性を重視した SfM を行う SLAM と呼ばれる技術について、その特徴をまとめる。

2.4.1. PTAM

マーカレス AR において、従来の SfM では計算コストが高く、リアルタイム処理に応用することが困難とされてきた。しかしながら、G.Klein、D.Murray によって 2007 年に提唱されたこの PTAM[16]はいくつかの画期的な手法によりそれを克服したブレークスルーである。

PTAM とは **Parallel Tracking and Mapping** の略であり、文字通り **Tracking** (視点追跡・特徴点对応) と **Mapping** (検出特徴点による再構成) を別スレッドに分けて動作させる。これによって高速・高精度なマーカレス AR の動作環境が実現された。要求コストが低く、発表当時一般的であったデュアルコア CPU、RAM4GB 程度の計算機で動作する。(スレッドを分けて **Tracking** と **Mapping** の並列処理を行う関係上、必ずコア数は 2 以上要求される。) その上、撮影機器はそれまでの SLAM と異なり、一般的な単眼カメラを用いて動作させることが可能である。さらに、この PTAM は一定の条件下 (GPL v3) において 2008 年よりフリーウェアとして公開された。このような実装条件の容易さにより多くの研究人口を獲得した結果、PTAM の基本的構造はこれ以降の多くの高精度 SLAM の根幹となっている。

PTAM の概要を以下にまとめる。

- **Tracking** と **Mapping** 処理を分割し、別スレッドにて実行
- 特徴点検出アルゴリズムは **FAST** を採用
- **Tracking** は毎フレームごとに実行
- **Mapping** はパッチ処理により、数フレームに一度実行

- Map 初期化には 2 つのフレーム画像に 5points-algorithm を適用
- Mapping にはエピポーラ幾何に基づいたアルゴリズムが採用
- Mapping に用いる特徴点数は数百～数千単位

ロボットに搭載されたカメラで 3 次元再構成を行う場合と人間の保持したカメラから重畳を行うマーカレス AR を比べると、カメラ移動の自由度が高い関係上 Tracking にかかるコストが高くなるといわれている [16]。PTAM は Tracking 処理を単独で動作させるよう分離させたことにより、常に一定以上のリソースをその処理に保証している。そのため、検出される特徴点数を多数要求されるアルゴリズムを実装することができ、一定のパフォーマンスを得ることが可能となる。結果として、位置・姿勢ともに高い推定精度、頑健性を獲得している。 [16][17]

PTAM に高速性をもたらす要因の一つが FAST と呼ばれる特徴点検出法である。これは、注目画素周辺の 16 輝度値と注目画素の輝度値の差をベースに darker, similar, brighter の 3 値をもつ 16 次元ベクトルを生成し、あらかじめ構築した決定木に基づいて特徴点か否かの判定を行う手法である [15]。

また、疎密的アルゴリズムと呼ばれる画像探索法によって画像の探索範囲を限定していることも特徴といえる。元画像に対し、4 画素を平均化した値を持つ 1 画素に置き換えることで解像度を下げた画像を生成し、これを複数回行った後低解像度の画像から順に探索していく。

2.4.2. ORB-SLAM

ORB-SLAM [18] は 2015 年に発表された、最新の特徴点ベース v-SLAM の 1 つである。PTAM と同じく単眼カメラで動作する SLAM であり、その中でも最高レベルの精度を誇る。

ORB-SLAM の概要は以下の通りである。

- 特徴点検出手法に ORB を採用
- システムを 3 スレッドに分割・並行動作することで高速化
- それぞれのスレッドは Tracking, Local Mapping, Loop Closing
- Tracking はフレームごとにカメラの位置測定を行うと同時に新たなキーフレーム挿入のタイミングを決定

- **Mapping** は新規追加されたキーフレーム内の情報を処理し、局地的な **bundle adjustment** の実行によりカメラ位置周辺を最適化された三角測量で再構成
- **Loop Closing** は視点移動の経路内のループ部分を毎フレーム検索、ループ発見時はループ両端の同一特徴点群を結合してループ閉鎖
- システムはキーフレーム、現フレーム、全特徴点、現フレーム内特徴点から構成される視覚化グラフ情報を複数保持
- 保持グラフをそれぞれのスレッドで動作最適化・ロスト時からのリカバーに使用

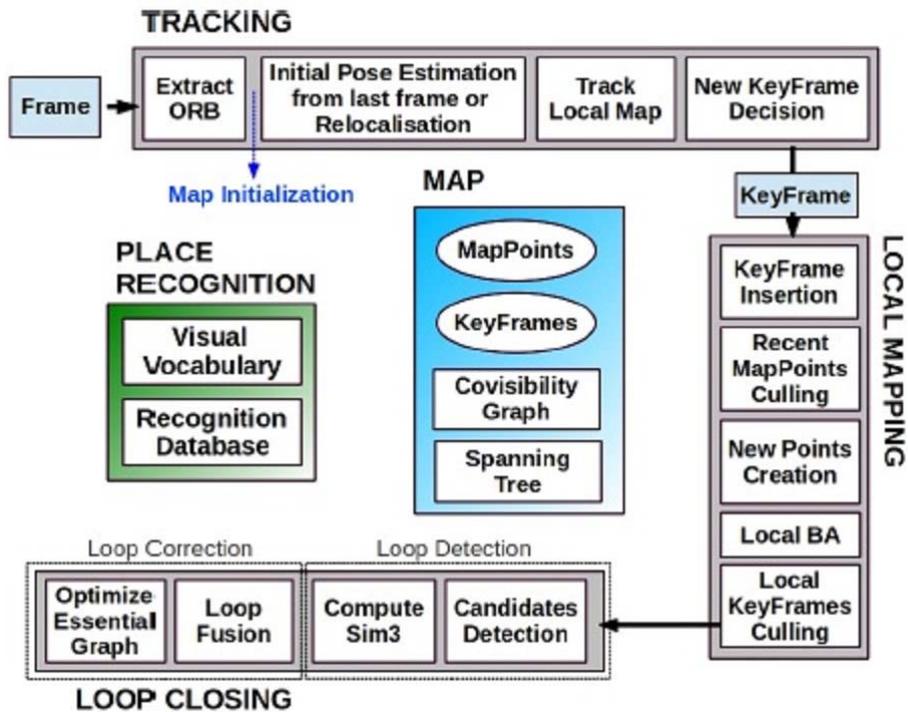
ORB とは **FAST** に回転に対する普遍性を付与するため、キーポイントのオリエンテーション算出機能を付加した検出法である[19]。

システムの構成自体は **PTAM** のスレッド分割手法をさらに発展させたものであることが概要から明らかであり、同一の場所を繰り返し撮影する場合に 3 次元再構築の精度向上と余剰特徴点の削除を兼ねたループ検出機構が追加されている。

また、システム全体の動作結果として得られたマップそのものを各スレッドが自らの次フレームの処理に活用する構造も非常に効率的といえる。

上記の設計により、**ORB-SLAM** は視点移動に対するロバストさと生成マップ、つまり 3 次元再構築結果の高精度・効率化を達成している。図 2.13 にシステム全体の概要を示す。

図 2.13 ORB-SLAM 概要図[18]

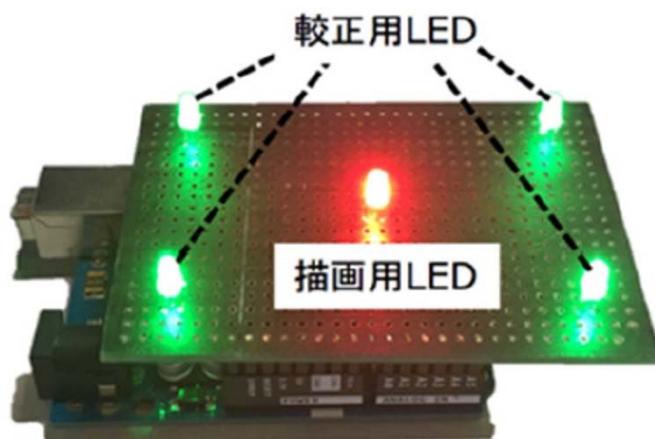


2.5. 提案手法の参考にした類似研究

本研究の目標であるインタラクティブな AR コンテンツの制作ツールは、ペン型デバイスによる空中描画システムである。その参考とした先行研究に、「インタラクティブな 3D 描画に向けた visual SLAM ベースの AR システム」がある。その概要を以下に示す[20]。

先行研究では入力形態として図 2.14 に示すデバイスを用いている。これは、Arduino UNO Rev3 の四隅に緑色 LED ランプを、中心に赤色 LED ランプを配置した正方形型のデバイスである。

図 2.14 先行研究での入力デバイス[20]



このデバイスのうち、中心の赤いランプは描画用、四隅の緑色ランプはカメラとデバイスの相対的な位置姿勢を検出する軸正用である。

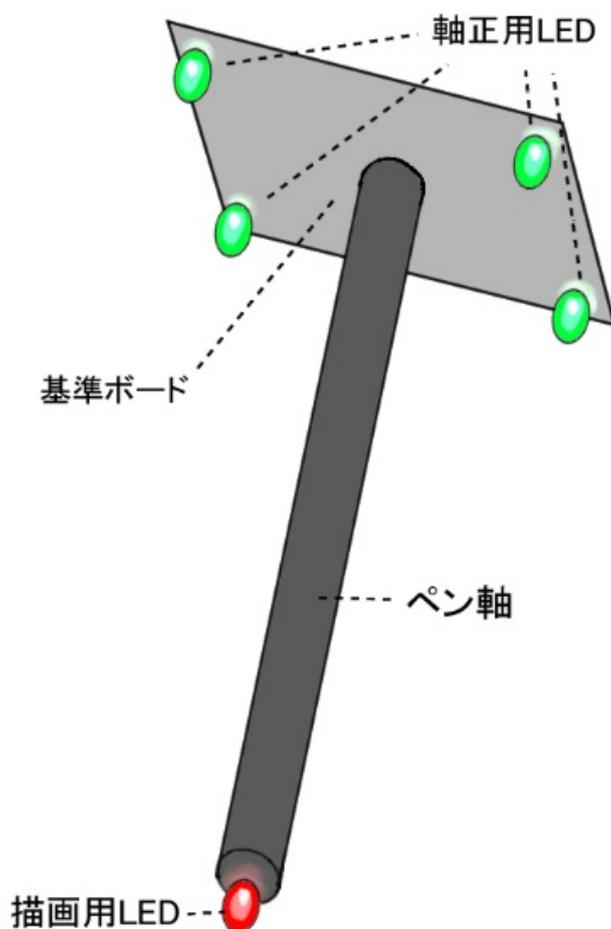
先行研究で提唱されたシステムの動作は、概略すると次の流れとなる。まず、3次元空間を再構成した後、デバイス中央のLED位置を推定し、それを再構成データに追加する。その後、登録されたLED位置を示す点群データを現在フレームに逆投影する。以上の動作を毎フレーム連続的に実行することで描画用LEDの軌跡を視覚化し、空間描画を実現する。

3. 提案手法

3.1. 入力形態および使用カメラ

梅澤氏の先行研究では、入力デバイスとして正方形型のデバイスを用いていた。この方法は入力するデバイスを平面と見なせるため計算が簡単になる利点を持つが、ユーザが自分の手で保持して図形を自在に描画するためには少々扱いづらい。そこで、本研究においては入力デバイスを図 3.1 に示すようなペン型とした。このデバイスなら、描画という目的に特化する上でユーザが平時より扱い慣れているペンと同じ感覚による入力が可能になるため、より自由かつ自然なインタフェースを提供することが期待できる。

図 3.1 ペン型デバイスの概形



また、周囲環境を撮影するカメラはシステムをできる限り一般的な環境で実

行可能にするため、入手の容易さから単眼の WEB カメラを使用した。

3.2. カメラ外部パラメータの取得

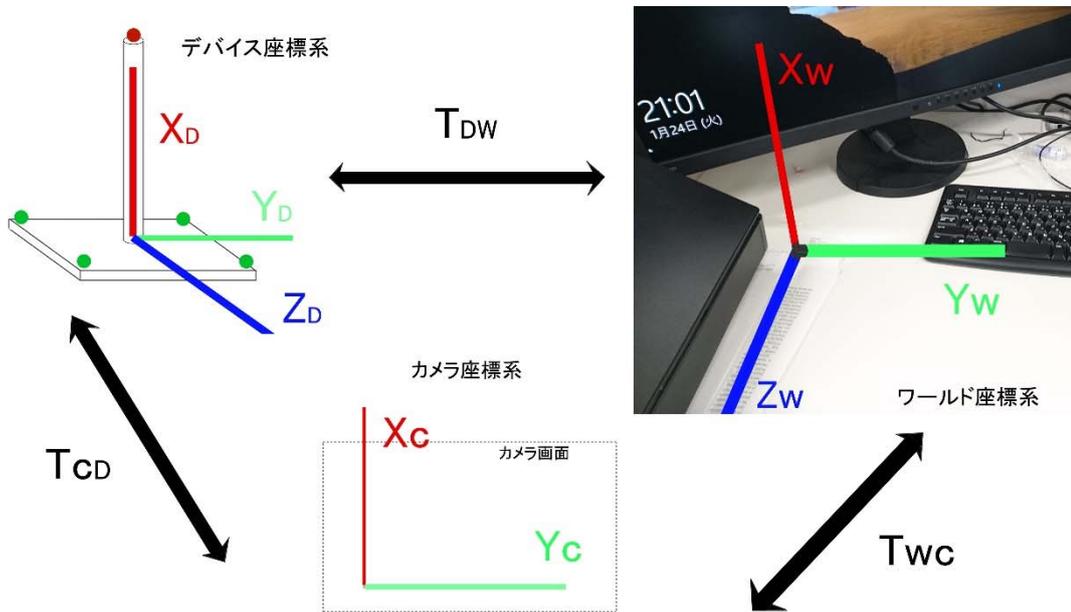
第 2 章で述べたように、カメラ外部パラメータは 3 次元環境を再構成する際に SfM によって算出される。したがって、システムを実行する周囲環境に対して何らかの v-SLAM を動作させることでカメラ外部パラメータが取得されることになる。

本研究のシステムにおいては、後述するようにペン先の描画用 LED の位置を点として保存する必要があるため、周囲環境の再構成データも点群 (point cloud) によって表現されている方が好ましい。そのため、3 次元再構成に使用する SLAM は特徴点ベースのものを採用した。4 章に示す特性実験の結果から、その中でも 2016 年現在で最高精度を持ち、かつ出力される点群データも現在最も普及している点群処理用ライブラリである point cloud library (pcl) で扱いやすい ORB-SLAM をメインとしてシステムを構築した。

3.3. ワールド・カメラ・デバイス間の座標変換

提案システムではカメラを固定し、デバイスを手で握って空間を動かすことが前提となる。そのため、現実空間におけるデバイスのワールド座標系での絶対位置は一定ではなく、毎フレーム変化する。したがって、このままでは現実空間が固定されている前提で再構成を行う SLAM はデバイス位置までは点群化できない。この問題を解決するため、デバイス固有の 3 次元座標系となるデバイス座標系を新たに導入することで第 2 章に述べたカメラと現実空間の座標系理論を拡張する。

図 3.2 各種座標系の関係



なお、第 2 章で述べた理論が基本となるため、各種座標系内の点の表現に関しては以降断り無しに同次座標を用いる。

T_{WC} は第 2 章で述べたカメラ外部パラメータであり、回転成分と並進成分から成るカメラの位置姿勢を表す 6 自由度行列である。ここで、基準となっている座標系はワールド座標系であるため、より正確に表記するならば「現実空間に対するカメラの位置姿勢」となることに注意する。同様に、 T_{CD} は「デバイスに対するカメラの位置姿勢」、 T_{DW} は「現実空間内でのカメラの位置姿勢」を表す行列であることがいえる。

また、AR として重畳する線を構成するものは描画用 LED の軌跡であるため、その各フレーム内での各座標系における位置をそれぞれ P_W , P_C とする。なお、 P_D に関してはデバイスが変形することはないため常に定数として扱える。

これらの位置姿勢行列を用いて、空間内に描画した結果を重畳表示することを考えると、最終的に映像がユーザに提供される場所はカメラ画面であるため、求めるべきはカメラ座標系におけるデバイスの位置であるといえる。すなわち、 P_C をワールド座標系経由で得ることができれば、奥行き方向を含めた立体的な描画を正しくカメラ画面に重畳することが可能となる。

現実空間、すなわちワールド座標系におけるデバイスの位置姿勢は、以下のように計算することで算出可能である。

$$T_{WD} = T_{CW}T_{CD}^{-1} \quad (3.1)$$

すなわち、

$$\begin{bmatrix} R_{WD} & t_{WD} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{CW} & t_{CW} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{CD}^T & -R_{CD}^T t_{CD} \\ 0 & 1 \end{bmatrix} \quad (3.2)$$

となる。ただし、**v-SLAM** に共通する問題点として、スケールが一定でないことが挙げられ、式(3.2)においてもワールド座標系のカメラの並進成分 t_{CW} とデバイス座標系のカメラの並進成分 t_{CD} のスケールは一致しない。したがって、どちらかを基準としてもう一方を表す必要がある。ここではワールド座標系におけるスケール、つまり第2章における λ を基準スケールとする。

式(3.2)から、各成分について展開すると(3.4)、(3.5)を立式できる。

$$R_{WD} = R_{CW}R_{CD}^T \quad (3.3)$$

$$t_{WD} = -\lambda R_{CW}R_{CD}^T t_{CD} + t_{CW} \quad (3.4)$$

これらにより P_D を P_W へと変換することを考えると、回転移動の距離は回転成分にスケール分を乗算した値となり、並進移動の成分は(3.4)でスケール変換済であるため、

$$P_W = \lambda R_{WD}P_D + t_{WD} \quad (3.5)$$

という式が導かれる。

第2章より、**ORB-SLAM** によって現実空間が再構築された段階で T_{CW} は算出されることが自明であるため、式(3.5)において未知であるのはスケール λ 、および R_{CD} 、 t_{CD} のみである。これらを算出できれば式(3.5)によって現実空間でのフレームごとのデバイス位置が推定可能である。

3.4. **v-SLAM** のスケール不定性の解決

基準とするスケールを λ としたため、これを求めるために必要なデバイスの描画用 LED、軸正用 LED の3次元位置は **ORB-SLAM** を用いればよい[20]。この計算を行う間デバイスを固定することで、現実環境と同様に3次元的位置が **ORB-SLAM** によって算出可能だからである。こうして算出されたデバイスの3次元位置を P_{SLAM} とする。デバイス座標系とワールド座標系のスケールを一致させるためには、 P_{SLAM} と式(3.5)の P_W との間の誤差を最小化すればよい。

近年、一般に 3 次元空間上の 2 つの点群データ間の位置合わせやスケール合わせに使用されるアルゴリズムの 1 つに ICP アルゴリズム[21][22]が存在する。ICP アルゴリズムは計算量の多さから、比較を行う 2 つのデータが大まかに位置合わせされていることが前提となるが、v-SLAM では基本的に直前のフレーム内の 3 次元点群データを取得できるためこの条件をクリアできる。

ある点群データに対するスケール変換とは、後述の P および f 、 S を用いて

$$f(p_i; S, p_c) = S(p_i - p_c) + p_c \quad (3.6)$$

なる計算を行うことに他ならない。ただし、この時 p_c はスケール変換の中心となる点を表す。

ICP アルゴリズムは、式(3.6)に現れるスケール変換行列 S を以下の入力から最適化することで、2 つの点群データ間の誤差を最小化する。

入力：

- 座標変換する 3 次元点群データ $P = \{p_i \in \mathbb{R}^3 | i = 1, 2, \dots, |P|\}$
- スケール合わせ対象の 3 次元点群データ $Q = \{q_j \in \mathbb{R}^3 | j = 1, 2, \dots, |Q|\}$
- 初期のスケール変換行列 S^0
- 収束判定閾値 τ 、反復上限回数 L

初期化：

- 反復回数 $k=0$ として、 S^0 により P^0 の座標変換

$$P^0 = \{p_i^0\} = \{f(p_i; S^0, p_c)\} \quad (3.7)$$

処理：反復回数を k 回目とする。

- 1) 点 p_i^k を以下のコスト関数を最小化する点 q_j に対応付けする。得られた対応点 q を q_i とする。なお、 $w(p_i^k)$ は与えられた位置ベクトルに対する単位法線ベクトルである。

$$c(p_i^k, q_j) = |p_i^k - q_j|^2 + |w(p_i^k) - w(q_j)|^2 \quad (3.8)$$

- 2) スケール変換行列 S^k を求める。最急降下法によって次の誤差評価関数を最小化する。求められた S を S^k とする。

$$E_S^k = \sum_{i=1}^{|P|} |q_i - f(p_i; S, p_c)|^2 \quad (3.9)$$

- 3) 2) で求められた S^k を用いて P^0 を座標変換し、 $k+1$ 回目の反復となる

座標変換を行う。

$$p^{k+1} = \{p_i^{k+1}\} = \{f(p_i^0; S^k, p_c^0)\} \quad (3.10)$$

4) (3.10)を(3.9)に代入し、 E_S^{k+1} を計算する。ここで、

$$E_S^k - E_S^{k+1} < \tau \text{ or } k > L \quad (3.11)$$

なら処理終了、そうでないなら 1)に戻り反復処理を続行する。

以上のアルゴリズムを本節に適用すると $p_i = P_{Di}$, $q_i = P_{SLAMi}$, $f =$ 式(3.5)である。したがって、スケール変換行列 S に対応するものは λ となり、それぞれを(3.9)に代入することでその計算式が(3.12)に導かれる。また、比較するのはデバイス上の描画用・軸正用の LED5 つの座標なので、 $MAX(i)=5$ である。スケール中心には各点群のそれぞれの座標系での平均位置を用いている。

$$\lambda^* = arg \min_{\lambda} \sum_{i \in S} |(P_{SLAMi} - \overline{P_{SLAM}}) - sR_{WD}(P_{Di} - \overline{P_D})|^2 \quad (3.12)$$

3.5. カメラ座標系とデバイス座標系間の変換

デバイス座標系におけるカメラの相対運動が推定できれば 3.3~3.4 節の理論により現実世界におけるデバイスのワールド座標がシステムに認識させられることが確かめられた。よって、本節では T_{CD} 、すなわち R_{CD} , T_{CD} を取得する方法を 2 通り示す。

まず一つ目に考えられるものは、マーカ AR のライブラリを転用する手法である。現在、マーカ式 AR の機能をソフトウェアに盛り込むための有名なライブラリに ARToolKit と呼ばれるものがある。ARToolKit は 2015 年にオープンソース化されており、利用しやすくなっている。また、マーカの作成、およびシステムへの認識も容易であり、2.1 節で挙げたマーカ形式に則った画像ならばカメラで撮影して登録可能である。

マーカ式 AR の動作は、カメラで撮影したマーカの距離・角度をパターンマッチングによって計算し、それに合致した角度に傾けたオブジェクトを重畳することで完了される。これは、カメラ座標系からマーカ座標系への座標変換に他ならない。したがって、ペン型デバイス自体をマーカとして登録できれば、ライブラリ内の関数によって各フレームでの R_{CD} , t_{CD} が計算されることになる。

ここで、図 3.1 に挙げたペン型デバイスの構造を確認すると、軸正用の 4 つの LED ランプを設置する箇所は正方形の板になっている。この部分の LED 設置面側にマーカを配置すると、マーカ座標系と図 3.2 のデバイス座標系が一致する。つまり、このときライブラリによって検出されるカメラ座標系とマーカ座標系間の変換行列が T_{CD} となることがいえる。したがって、計算されたマーカ座標系の各要素を参照することで R_{CD} , t_{CD} が取得されることになる。

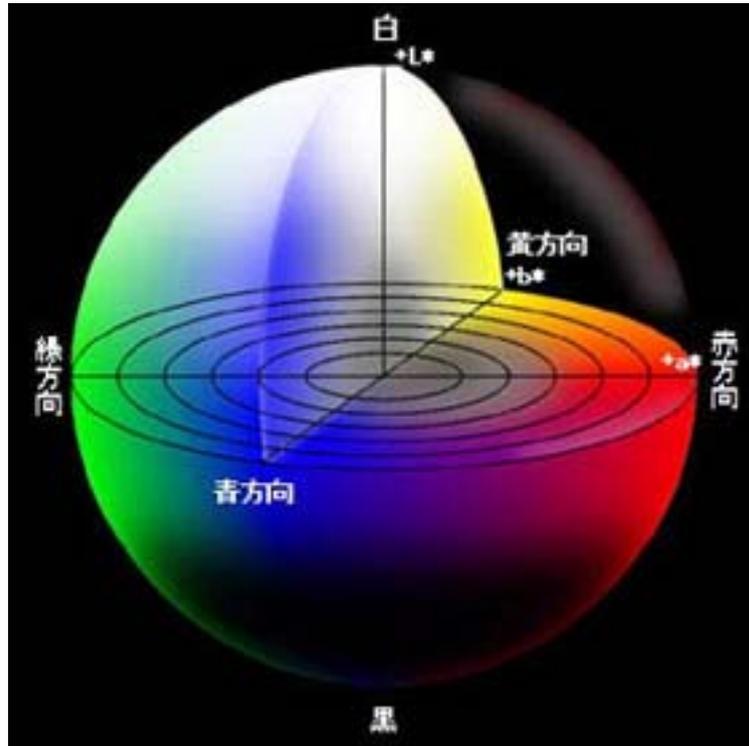
もう一方の手法は、LED によりデバイス座標系とスクリーン座標系で対応する同一点を 4 個検出することで R_{CD} , t_{CD} を計算する方法である。デバイス座標系での各 LED の位置は簡単に算出できるため、これを実現するためにはカメラ座標系での各 LED 位置を求められればよいことになる。

LED の中心部は高輝度であり、撮影された画像内では他の箇所より白飛びしていることは自明である。そこで、画像内の白飛び箇所を楕円形にトリミングする。トリミング後の画像に対して適切に関心領域 (Region of Interest; ROI) を作成し、その内適切なもの (つまり LED ランプ部分を写した ROI のみ) を抽出し、その中心座標を算出して各 LED のカメラ画面上の座標とする。

ここで、LED には描画用と軸正用の 2 種があり、これらは区別して認識されなければならない。そのため、LED ランプの色を利用する。

知覚的にほぼ均等な歩度と評価される色空間の 1 つに、CIE が 1976 年に推奨した L^*a^*b 色空間が挙げられる。これは、図 3.3 に示す XYZ 型の表色系で、 L は輝度、すなわち色の明るさであり、0~100 までで評価される。 a は赤色の強さを表し、正数値なら赤み、負数値なら補色である緑の強さを表現する。 b は黄色の強さであり、正数値なら黄色、負数値なら青色の強さを表せる。 a, b ともに 0 ならば色彩なしとなる。

図 3.3 L*a*b 色空間モデル[23]



LED の色による描画用・軸正用 LED の区別を行うには、まずそれぞれの LED のテンプレート画像を用意しておく必要がある。トリミングしたそれぞれの領域において、楕円の長径方向、短径方向の直線状の画素値を準備したテンプレート画像と比較する。このとき、描画用と軸正用では発光が a^* 軸方向の補色の関係となるため、明確な区別が可能となる。

以上により各トリミング領域と適切なテンプレート画像とを分布関数に入力して比較すると、領域内の ROI 作成箇所によって様々な値をもつズレが生じる。このとき、テンプレート画像と分布のズレが最も小さくなる ROI を選択することでカメラ画面上の LED による発光部分を検出できる。簡潔に例示して説明すると、蛍光灯や反射光なども高輝度部分としてトリミングされるが、 L^*a^*b 色比較によってテンプレート画像と類似した光り方の ROI のみをその中から抽出できる、ということである。

本節ではこれまでに 2 通りの R_{CD} , t_{CD} 算出法を示した。これらはどちらも一長一短である。

前者の方法は比較的実装が容易で認識精度が高いものの、先行研究よりも多くのライブラリを追加する必要がある。また、マーカが認識できない場合はシ

システム全体の計算が途切れる可能性も生じる。また、LED 設置面にはペン軸も接着されているため、角度によってはマーカの一部がペン軸の陰となり隠されることも考えられる。

後者はライブラリ数を節約でき、また LED 自体が高輝度で画像内にて目立つためシステムによる認識力が高い。しかし、ペン軸方向の回転を検出することが困難である上に、算出までに複数のアルゴリズムを組み合わせる動作させねばならない。

また、両者ともにカメラに対して LED 面が映らない、具体的にはカメラに LED もしくはマーカ設置板の裏面を向けている場合はデバイス位置の推定が不可能となる。

そこで、本研究のシステムでは、現段階では後者の手法を実装中であるが将来的にこれら 2 種の手法を併用することを考えている。具体的には以下の通りである。

LED 設置面にマーカも同時に配置し、また設置板の裏面にも別のマーカを配する。そして、裏面のマーカをシステムが認識した場合、デバイス座標系の y 座標値を反転させる。LED・マーカ両者がともに認識されている場合はその平均値を R_{CD} , t_{CD} とし、片一方の場合は利用可能なものを用いる。

これらの仕組みを追加することによって、ペン軸の回転、およびカメラに LED 設置板の裏面が映っている場合でも想定されるペン先の位置を算出可能になり、位置推定精度も先行研究より向上されることが期待できる。

ペン軸方向の回転が活用される高度なアプリケーションに応用する場合を考えると、上記の仕組みの追加が大きな役割を果たすことができる。例えば蛍光ペンのようにペン先の形状を円形以外の形状に設定するとき、ペン軸の描画面に対する傾きによる線幅の変化までを含めて 3 次元空間上にトレースすることが可能になると想定される。

3.6. 描画の実行

3.5 節で R_{CD} , t_{CD} が算出されたことにより、3.4 節のスケール計算、次いで 3.3 節内式(3.5)によるデバイスの描画用 LED のワールド座標 P_D が計算可能となる。この一連の計算により、システムの最終目的である描画用 LED のカメラ画面への投影計算が可能となる。その計算式は第 2 章の式(2.12)に各座標を代入することで式(3.13)として整理可能である。(ただし、本節の T_{CW} はカメラ座標をワールド座標に変換するため、その逆行列が現実空間からカメラへの投

影に対応していることに注意)

$$P_C = K T_{CW}^{-1} P_W \quad (3.13)$$

上記した一連の手順でシステムに登録された P_W , P_C はフレームが移り変わっても破棄せず、次フレーム以降でも投影し続ける。これは、 P_W を **ORB-SLAM** によって生成された現実環境の点群データに追加登録することで実現可能である。以上によって描画用 **LED** の軌跡が **AR** として表示されることになる。

4. 実験結果

本章に示す実験の実行環境を次の表 4.1 にまとめる。

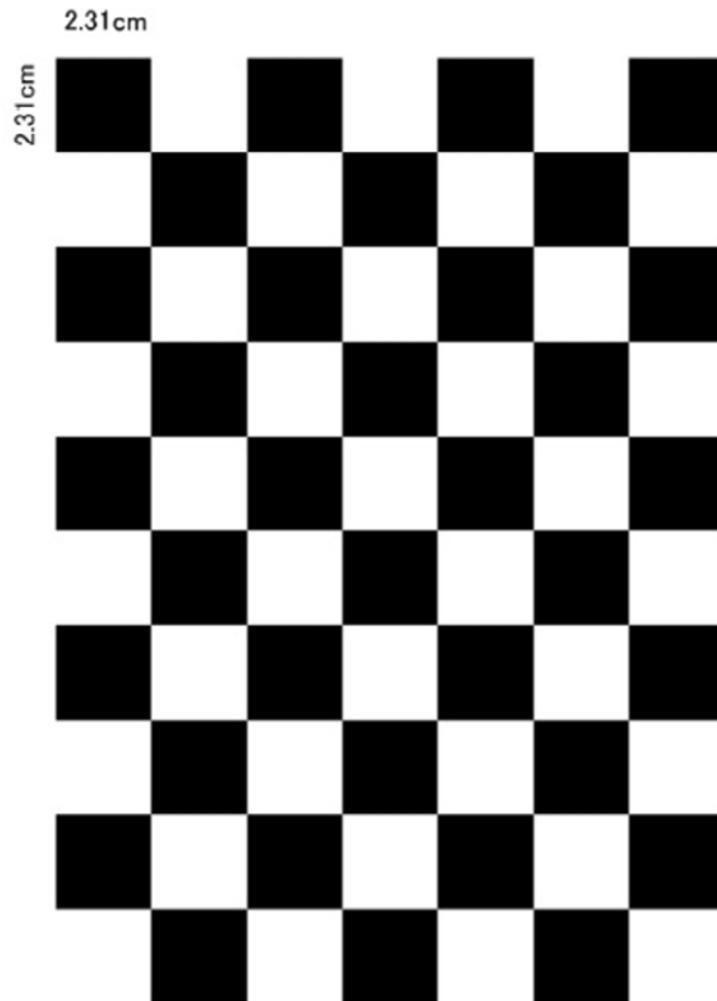
表 4.1 実験環境

OS	Windows10+Ubuntu14.04LTS
プラットフォーム	ROS indigo
CPU	Intel core i7
グラフィックボード	
メモリ	3GB
ディスク	
使用外部デバイス	USBWEB カメラ (機種後述)

4.1. カメラキャリブレーション

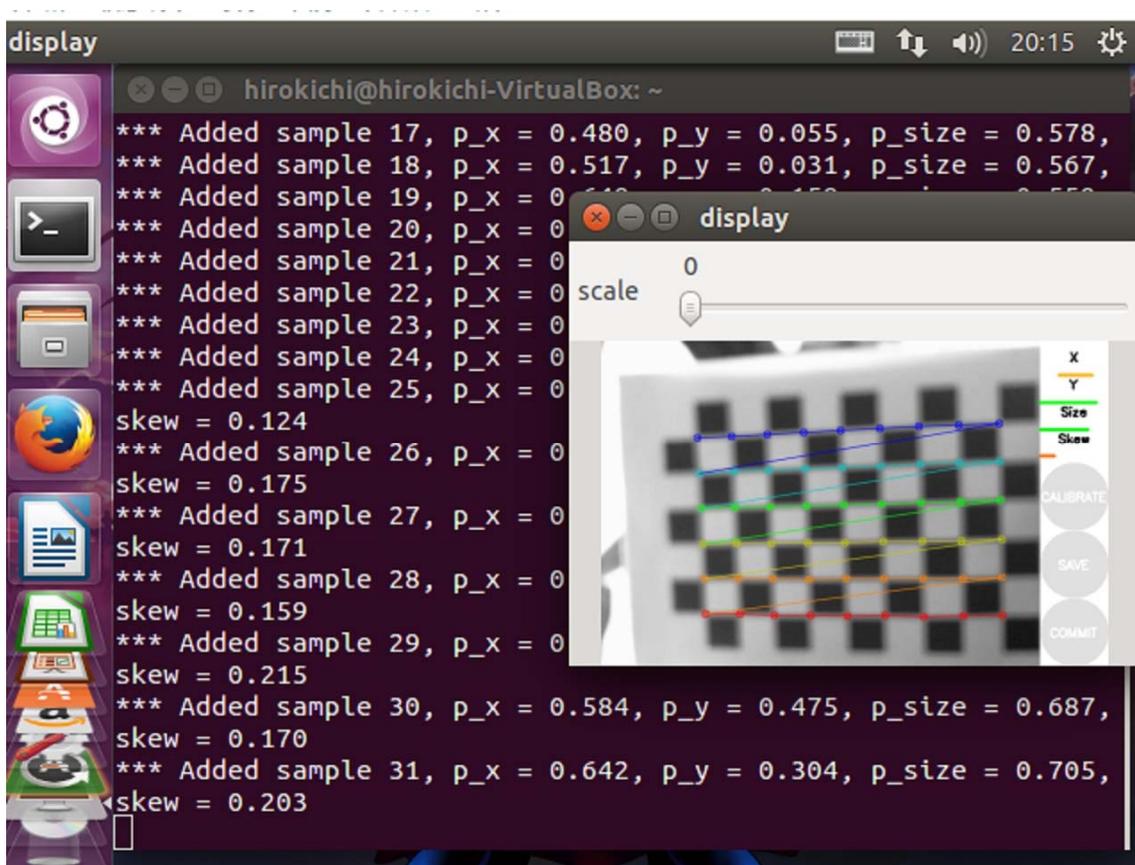
本節では、第 3 章の理論に基づいたカメラキャリブレーションの結果を示す。図 4.1 はキャリブレーションに用いたチェッカーボードの詳細である。縦横のマス数は 7×10 、各正方形のサイズは計測したところ、1 辺の長さが 2.31[cm]であった。

図 4.1 使用したキャリブレーション用チェッカーボード



このボードを用いて、web カメラ(Logicool Webcam C615)のキャリブレーションを行った。次の図 4.2 にその様子を示す。

図 4.2 カメラキャリブレーションの様子



キャリブレーションソフトには ROS 上で動作する WEB カメラ用のパッケージを使用している。チェックから取得されたラインがほぼ直線になっていることから、カメラ内部パラメータの取得によって歪みが是正されていることが分かる。

4.2. 各種 SLAM による実験環境の 3 次元再構成

4.2.1. PTAM の実装と考察

マーカレス AR や SLAM に用いられるアルゴリズムの多くは複数のライブラリによる複雑な依存関係によって構成されている。したがって、用いるライブラリのバージョンが作成者の実装環境とわずかにでも異なるとソフトウェアのインストールに失敗することがあり、これがマーカレス AR における研究入門をマーカ式のものに比べて困難にしている一因

であることは第 2 章でも既に述べた。PTAM もその例外ではなく、2007 年の発表時から多くのライブラリがバージョンアップしたこともあり、現行のバージョンでは多くのエラーが発生した。特に致命的だったのは `fltk` というライブラリの 2.x 系の開発が中断され、現時点で入手不可能となっていた点である。

こうした問題を解決するため、本研究ではオリジナルの PTAM ではなく、ロボティクス系で使用される ROS というプラットフォームを入手し、その ROS 系で動作するよう移植された `ethzasl-PTAM` というパッケージを導入した。

以下、`ethzasl-PTAM` に同梱されていた `bag` ファイルを ROS の再生機能を用いて PTAM に入力し、再構成精度を確認した。図 4.3~4.4 に結果を示す。図 4.3 は平面が画面上の大部分を占める比較的単純な部分に対する再構成結果、図 4.4 はキーボードなど、前者より入り組んだ環境に対する再構成結果である。

図 4.3 PTAM の動作結果 1/2 (入力: `bag` ファイル)

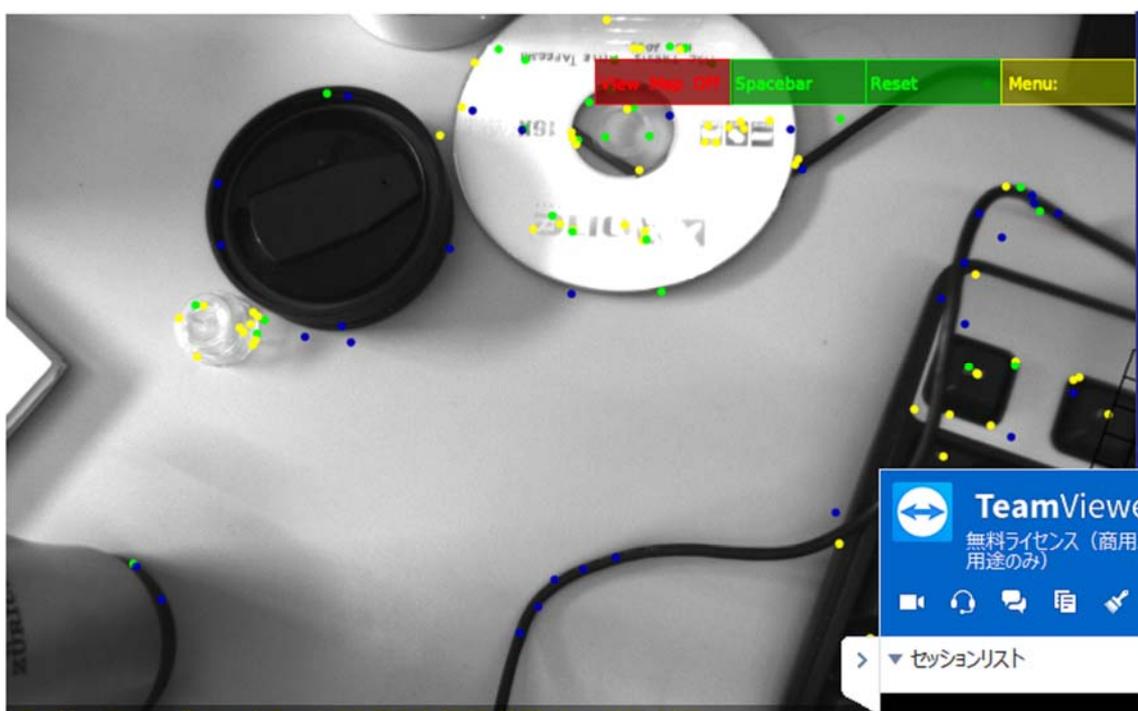


図 4.4 PTAM の動作結果 2/2 (入力: bag ファイル)

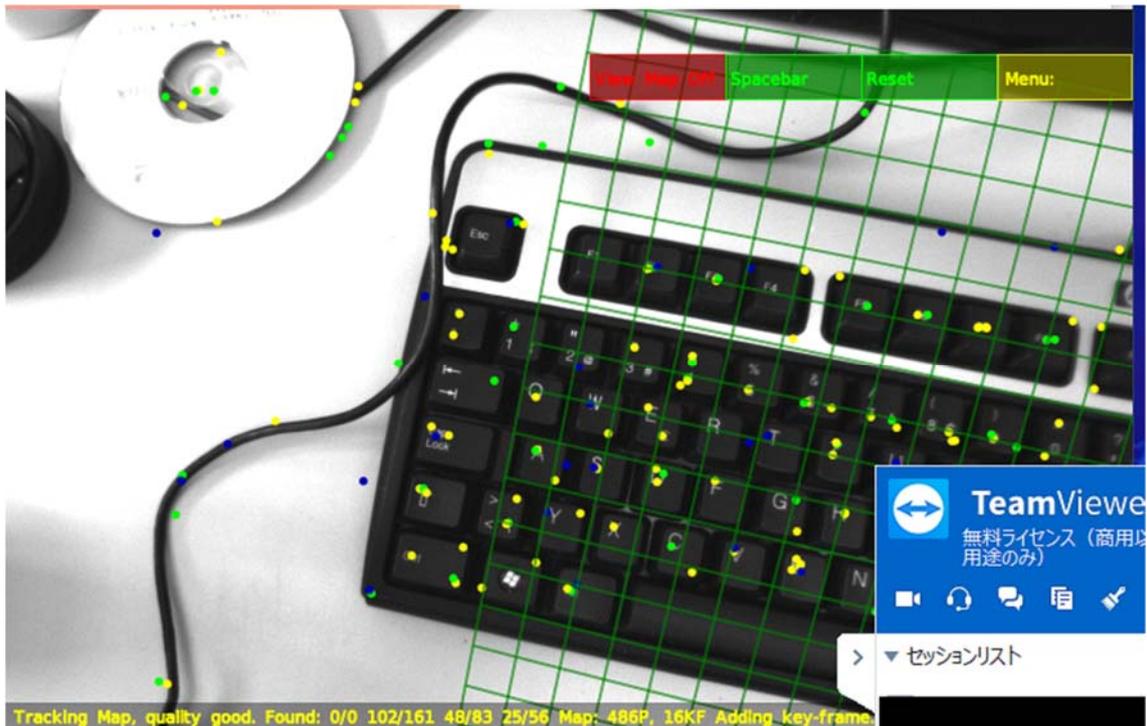


図 4.3 を見ると、平面が画面の大部分を占め、明確なコーナ一点に乏しいシーンでの特徴点検出は上手くいかず、平面の重畳も行えていない。CD や水筒などの明確な立体物に対しても特徴点検出量が少ないことは明確である。特徴点を検出できているところは小物が重なっている部分や CD とケーブルの重なっている部分、およびケーブルが強く曲がっている部分が挙げられる。

一方、図 4.4 を見ると基準平面の重畳が行えている。キーボードのキーとグリッド線が（キーボード自体がやや湾曲していることを考慮して）ほぼ平行であることから、 x 軸、 y 軸、 z 軸方向に対するキーボードの傾きを比較的正確に検出し、それに合致した平面を再構成できていると評価できる。

この後のシーンにおける動作結果は、カメラの動きが大きくなる部分では特徴点のロストがしばしば発生した。また、高輝度部分に対しては検出力が ORB-SLAM より劣っていた。

これらの結果から、PTAM はカメラの視点移動、特に回転成分が大きいシーンや白色部分（あるいは高輝度部分）の多いシーンに対しては精度が期待できないといえる。一方で、キーボードのキーやキャリブレーション

ンボードのようにコーナー箇所が多い部分に対しては正確な認識力を発揮できる。また、キーや CD の印字部分など、周囲とのコントラストが強い小規模点を特徴点として認識しやすい傾向にあった。こうした特性は主に PTAM に搭載された、コーナー検出に特化した FAST を特徴点検出機構に採用していることに起因すると考えられる。

4.2.2. ORB-SLAM の実装と考察

本研究で ROS 環境を構築した理由は PTAM の移植版パッケージの導入以外にもう 1 つあり、現状の高精度 SLAM の多くが ROS 環境向けに開発されている、という点による。ORB-SLAM もその一つであり、こちらは発表時期が新しいこと、また著者らによるインストール手順の説明が丁寧に記載された readmemo が添付されていたことから比較的容易に実装することができた。以下、PTAM と同様に bag ファイルの映像に対して 3 次元再構成を実行した結果を示す。なお、ORB-SLAM にも著者らが実行用のサンプルとして撮影した映像ファイルが入手可能であるが、PTAM との比較を行うために前項と同一の bag ファイルを入力として使用した。

図 4.5 ORB-SLAM による 3 次元特徴点検出 1/5(入力: bag ファイル)



図 4.6 上図時点での 3 次元再構成結果 1/5

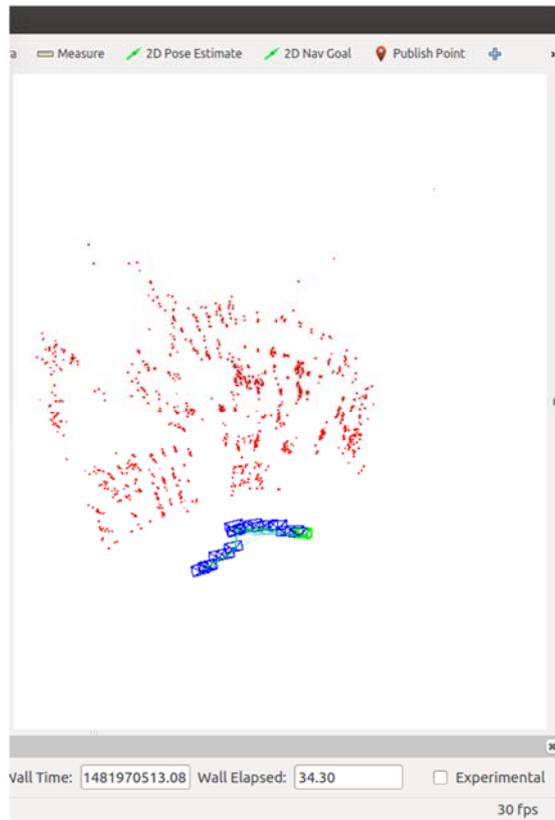


図 4.7 ORB-SLAM による 3 次元特徴点検出 2/5(入力: bag ファイル)



図 4.8 上図時点での 3 次元再構成結果 2/5

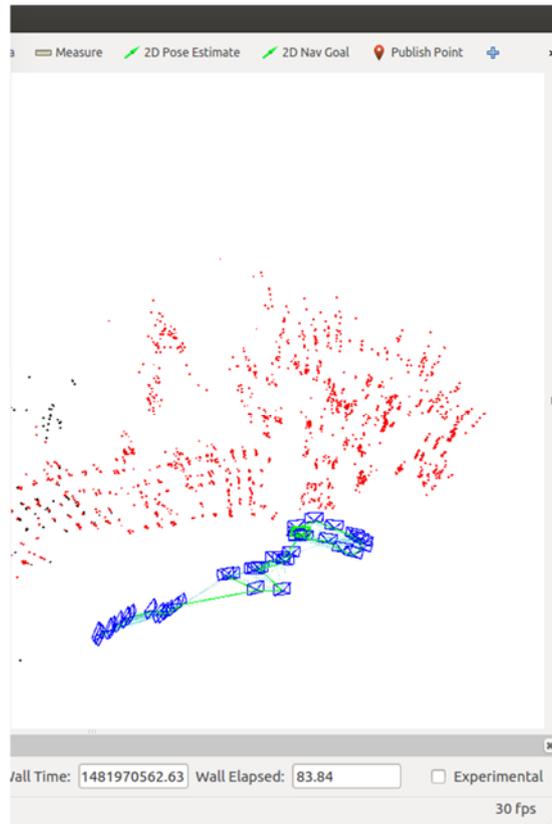


図 4.9 ORB-SLAM による 3 次元特徴点検出 3/5(入力: bag ファイル)

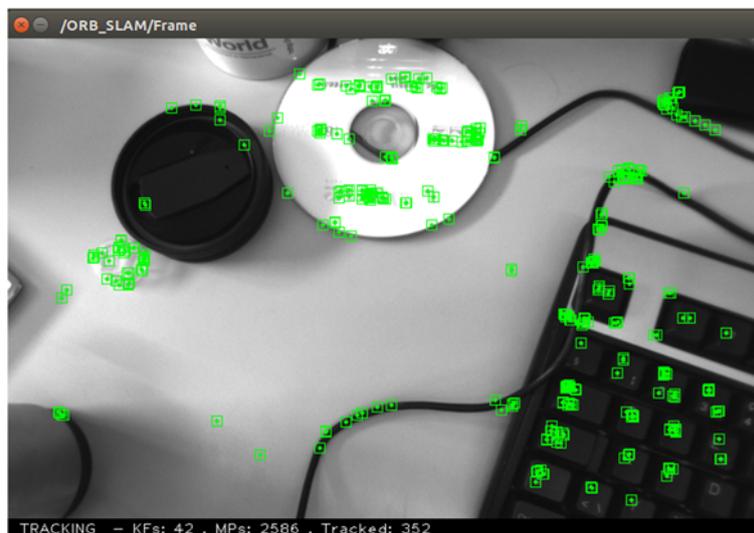


図 4.10 上図時点での 3 次元再構成結果 3/5

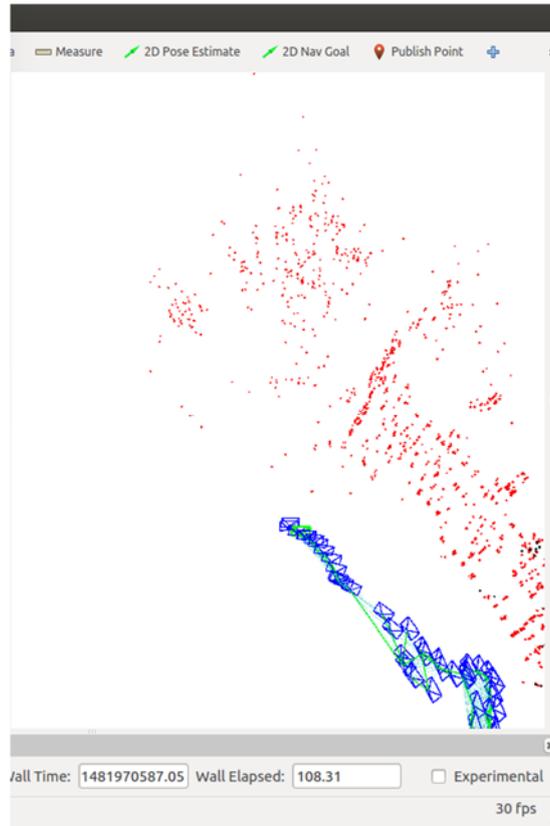


図 4.11 ORB-SLAM による 3 次元特徴点検出 4/5(入力: bag ファイル)

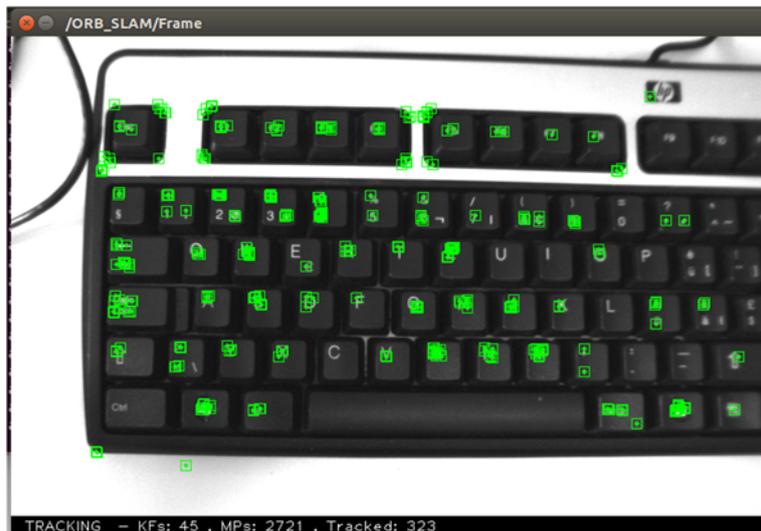


図 4.12 上図時点での 3 次元再構成結果 4/5

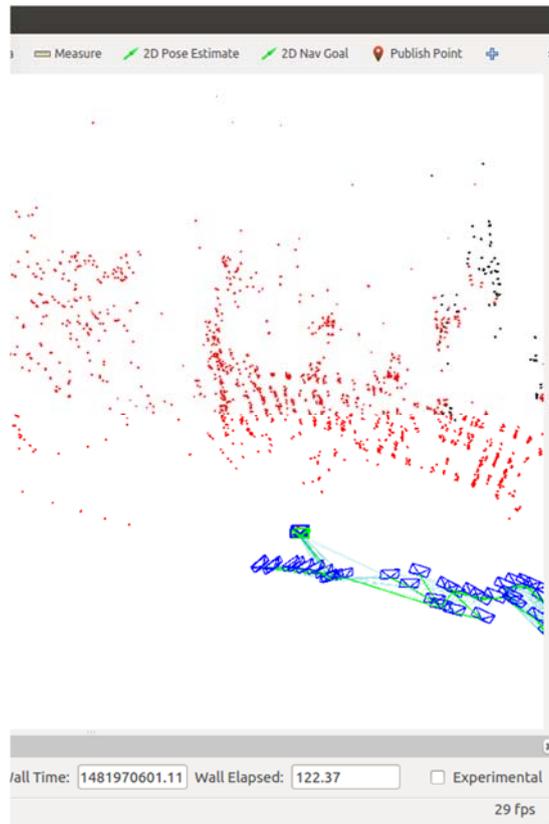


図 4.13 ORB-SLAM による 3 次元特徴点検出 5/5(入力: bag ファイル)



図 4.14 上図時点での 3 次元再構成結果 5/5

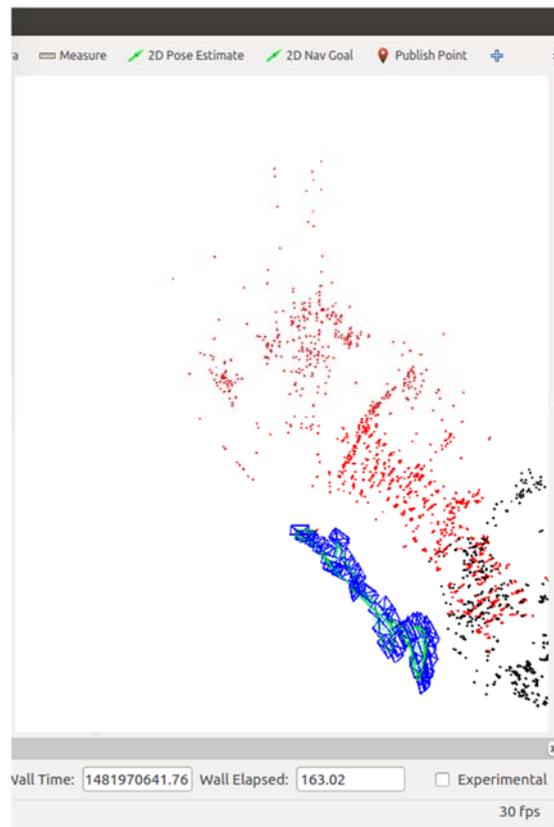


図 4.9 は PTAM の検出精度が悪かったシーンとほぼ同じシーンに対して ORB-SLAM を動作させた結果である。検出特徴点の数が明らかに上回っており、かつその精度も高いことが分かる。PTAM では検出できなかったケーブルの緩やかな曲がり角、および CD や小物の輪郭といった曲線部分に対しても特徴点を一定数検出できている。

また、図 4.13 は入力映像の終盤で図 4.7 とほぼ同じ場所にカメラが戻ってきたシーンであり、図 4.9 と比べると曲線部分に対する特徴点検出数がさらに若干増えている。一方で、図 4.9 で既に検出されていた特徴点に対して同一箇所にもう一度同じ特徴点を検出し、重複するといった現象はほぼ見られなかった。

同じ個所を繰り返し撮影することで精度が向上する傾向は PTAM にも見られるが、ORB-SLAM は屋外環境での動作も視野に入れ、検出した特徴点群からループまで検出し、同一と思われる特徴点を自動的にマージする機構が実装されている。上記の結果はこのアルゴリズムによりまだ登録されていない特徴点のみが優先的に点群に追加された結果だと評価できる。

PTAM が高い検出率を示したシーンに対しても、ORB-SLAM の検出率は決して劣らないものとなった。図 4.12 を参照すると、キーの印字部分に加えて、キー自体のコーナーも検出されている。図 4.5 や図 4.7 にも同様の傾向がより顕著に表れており、モニターの角やノート PC のタッチパッド部分、および貼られているラベルなどについてほぼ完全にコーナーを検出できている。

これらの図にはもう 1 つ評価可能な点がある。それは、キーボード上おかれた動物のペーパークラフトに対しても正確にほぼ全てのコーナーを検出し、その立体構造が把握できることである。このシーンにおいてはキーボードの法線方向とカメラの向きが近く、肉眼で見てもペーパークラフトの立体感はやや掴みづらいものとなっている。しかし、ORB-SLAM の動作結果はキーボードとペーパークラフトの両者に対して十分な数の特徴点を検出しており、平面と立体物とに明確に分離できるものとなった。本研究で目指す 3 次元描画において、カメラから立体感の判断が難しい視点での奥行きを正確に把握可能という利点は筆跡のトレースをより正確に行うことに繋がると期待できる。

図 4.6, 4.8, 4.10, 4.12, 4.14 は特徴点から再構成された 3 次元再構築結果を時系列順に並べたものである。それぞれ 1 つ前の図に示された特徴点検出結果と同フレームに対応している。これらを見ると、時系列に沿ってカメラ前方に特徴点が追加されていく様子が分かる。最終的な結果である図 4.14 を参照すると、キーボードやモニターの概形は点群データのみでの表示でもある程度把握でき、特にキーボードが緩やかに湾曲している様子はかなりの密度で再現されている。以上の結果から、入力映像から正確に 3 次元情報を復元できていると判断した。壁や平面の存在する位置がある程度密な点群で表現され、空間と区別できる以上、提案システム用の SLAM として採用するに十分な精度といえる。

次に、キャリブレーションを行ったカメラにより撮影された動画像に対して同様に ORB-SLAM による 3 次元再構成を行った。撮影対象は研究室のデスク付近である。前実験での入力映像は PTAM の動作確認用として用意されただけあり、v-SLAM にとって理想的なカメラワークが採用されていたため、こちらの実験ではなるべく乱雑な環境をある程度大きくカメラを傾けるシーンも含めながら入力画像を撮影した。

図 4.15 ORB-SLAM による 3 次元特徴点検出 1/4(入力: カメラ画像)



図 4.16 上図時点での 3 次元再構成結果 1/4

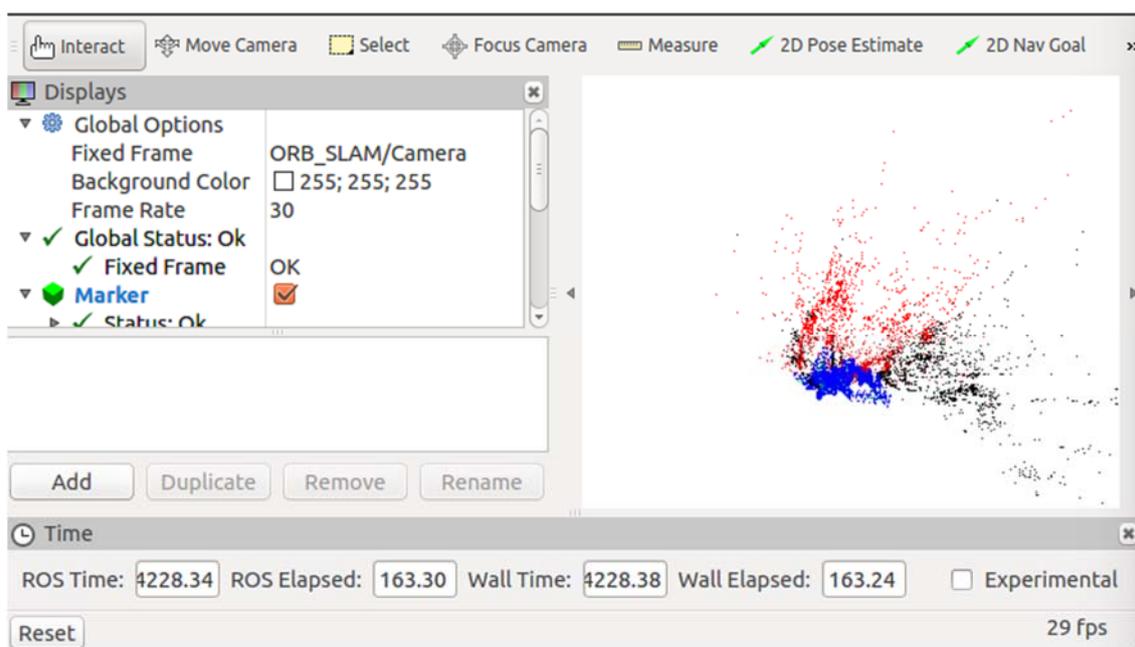


図 4.17 ORB-SLAM による 3 次元特徴点検出 2/4(入力: カメラ画像)



図 4.18 上図時点での 3 次元再構成結果 2/4

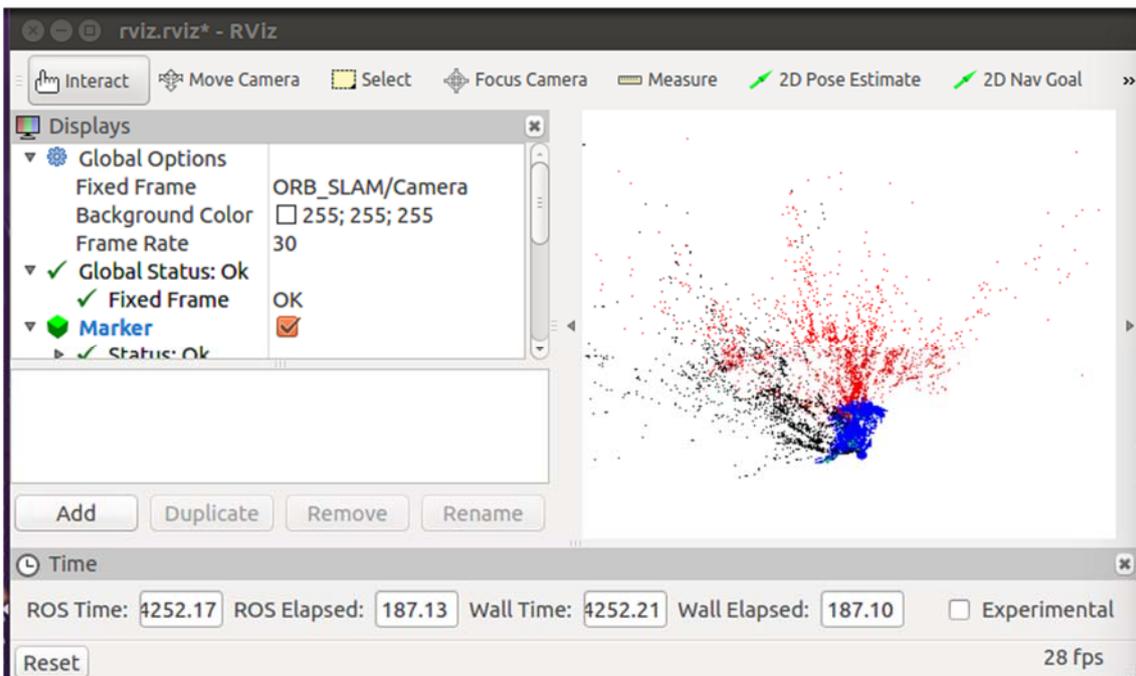


図 4.19 ORB-SLAM による 3 次元特徴点検出 3/4(入力: カメラ画像)

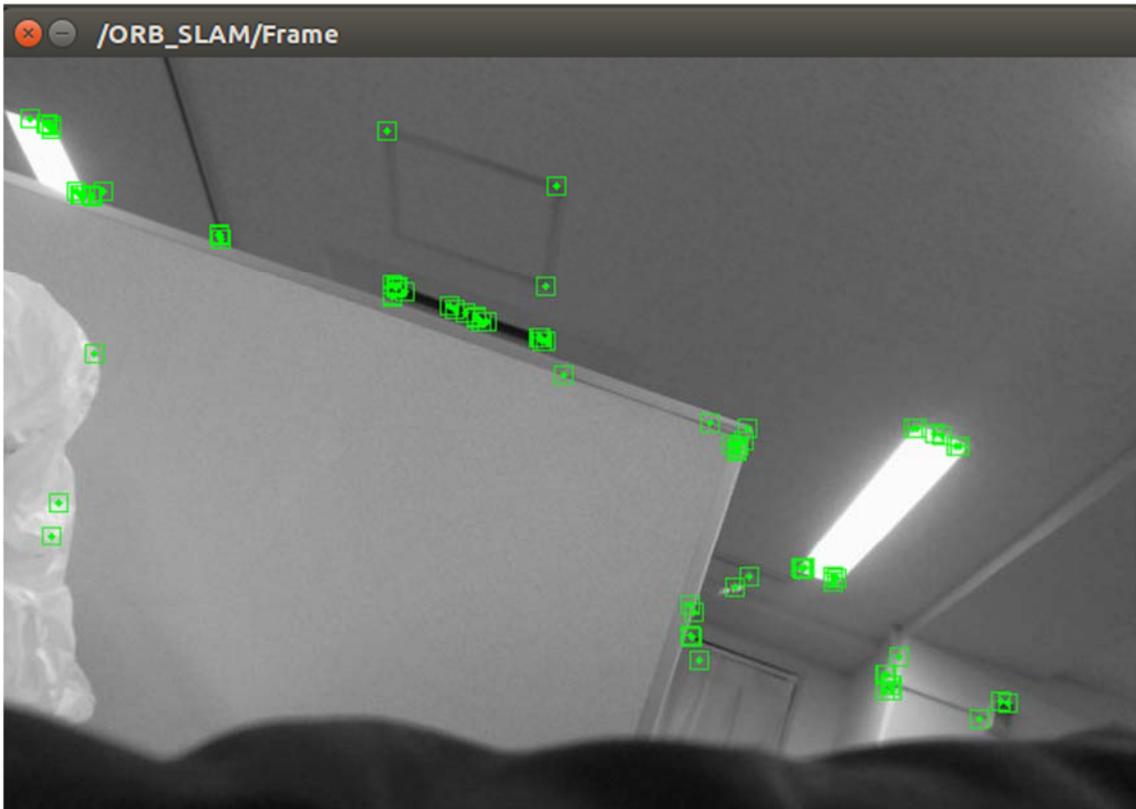


図 4.20 上図時点での 3 次元再構成結果 3/4

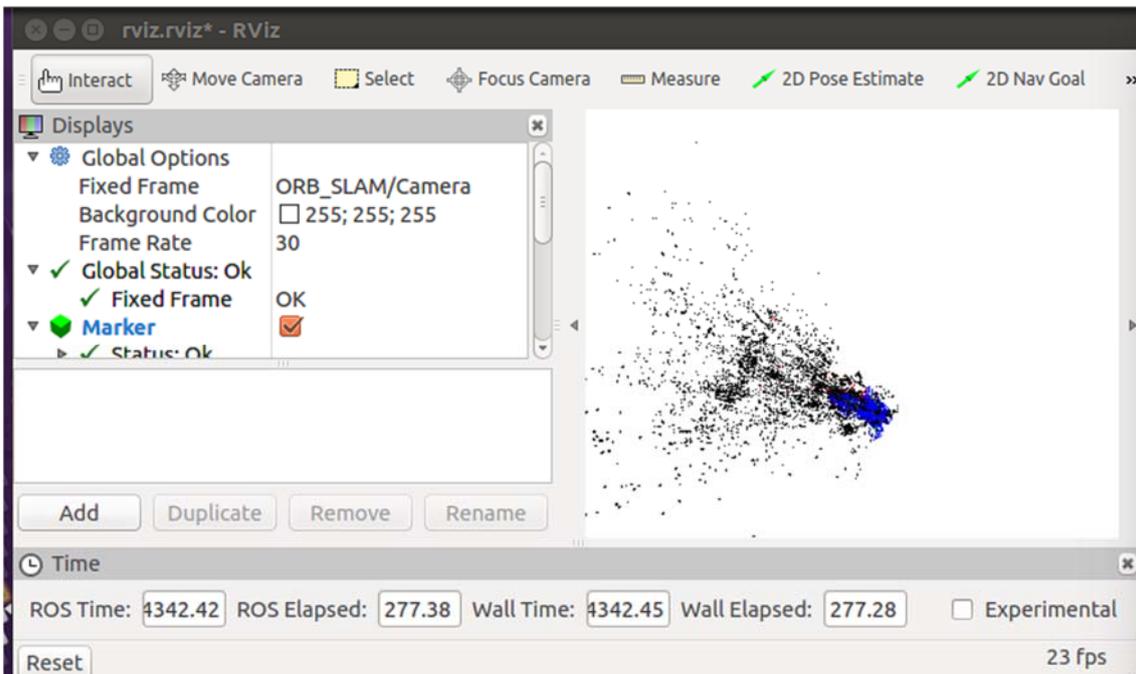


図 4.21 ORB-SLAM による 3 次元特徴点検出 4/4(入力: カメラ画像)

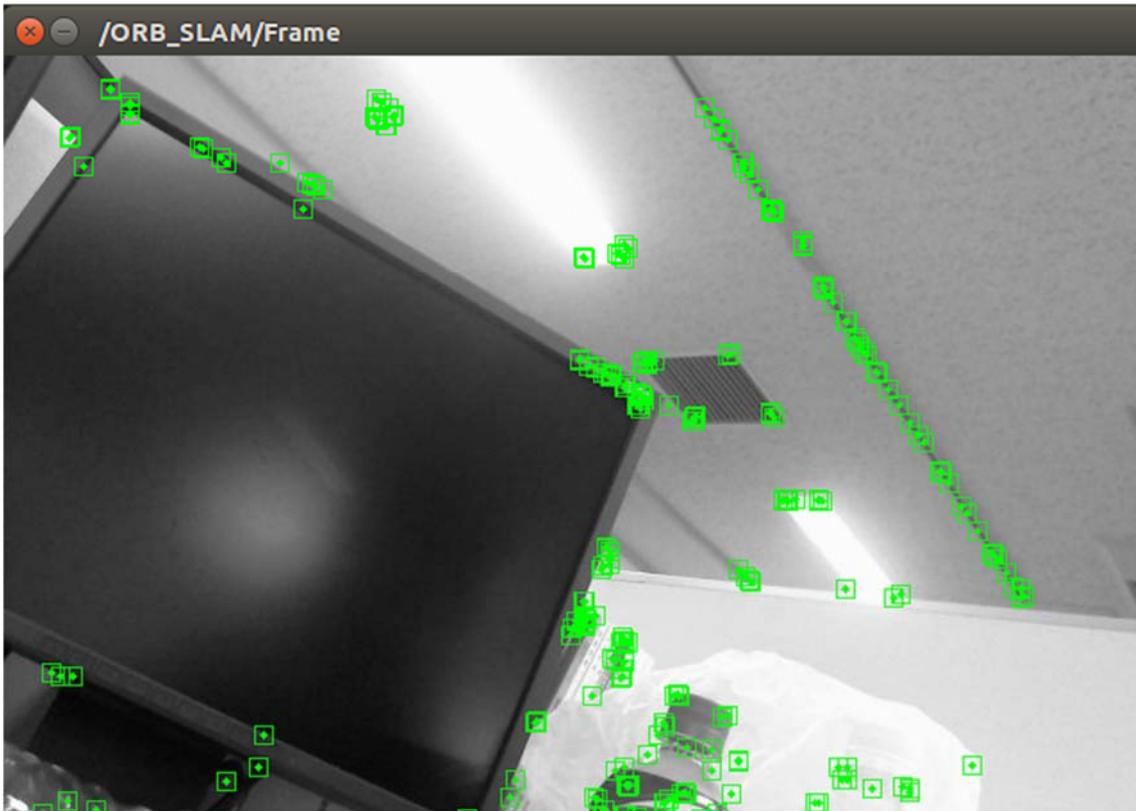
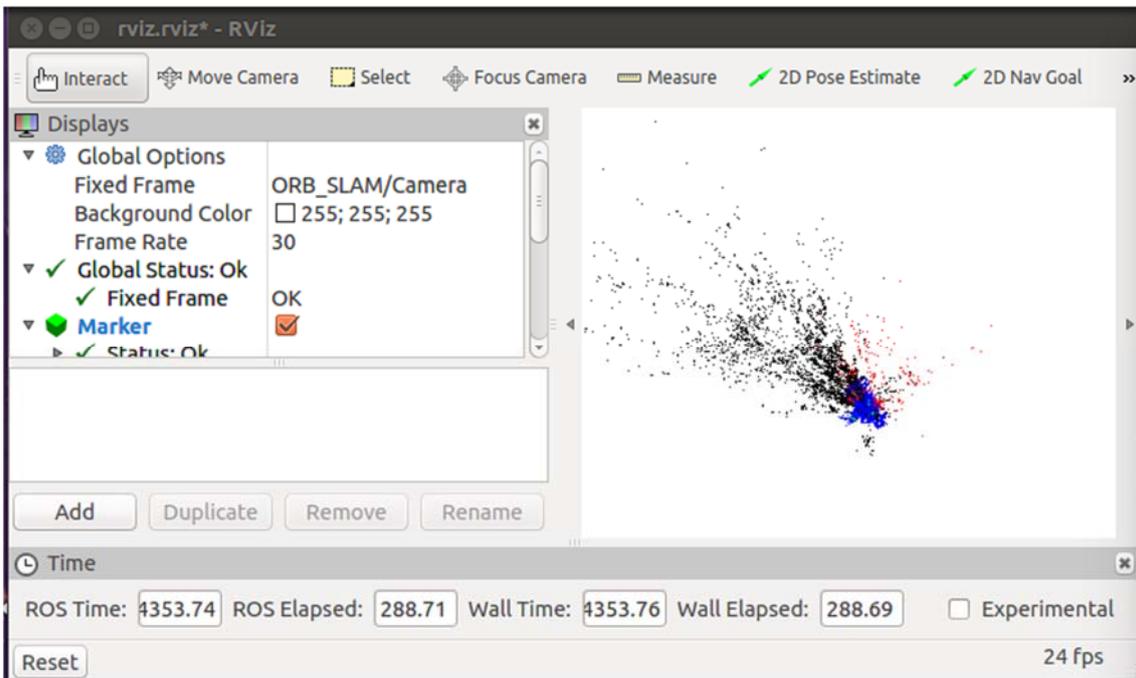


図 4.22 上図時点での 3 次元再構成結果 4/4



まず、図 4.15 を見ると、ケーブルが強く曲がっている部分や重なり合っている部分は **bag** ファイル入力時の **PTAM** と同様、カメラ映像に対しても実時間で検出されたことが分かる。また、モニターのコーナーについても前実験と同様に正確に検出されている。モニターのスタンド部分は円形であり、曲線状であるが、こちらに対しても輪郭が再現されるほどの特徴点数を検出することができた。

一方で、前実験と比べて特徴的であった点は、モニターに表示されたデスクトップ画像やペットボトルに反射した蛍光灯の光などを特徴点として認識した点である。この原因についての考察は後述する。

次に、図 4.17 を参照する。図 4.15 に比べて非常に乱雑なシーンを映した映像であるが、こちらはやはり図 4.15 に比べて検出箇所の規則性に劣る。傾向を分析するなら、腕時計のベルトなど非常に強く曲がっているコーナー部分の他にはマウスのボタン横のフレーム、ケーブル類、ペットボトルのラベルのロゴなど線状に映像に映っている箇所を優先していることが分かる。また、視点に近い部分ほど特徴点の検出数が多いこともいえる。

図 4.19, 図 4.21 については、**PTAM** が苦手としていた回転成分の強い視点に対して **ORB-SLAM** がどれほどの検出率を示せるかを試したシーンである。結果としては、図に示した通りコーナーやエッジ部分に対して極めて良好な精度を示した。2 枚の図を比べて分かる通り、意図的に視点を短時間で強く動かしたが、何もない場所に特徴点を検出する、直前に映っていた特徴点を大幅にロストするといった問題は生じていない。図 4.21 右下部分には図 4.17 に映っていたペットボトルの上部が映っているが、視点が大きく変化していても図 4.17 にてラベル内に検出された特徴点があらぬ箇所に移動するといったことは認められない。さらに、図 4.21 では新たに検出された特徴点によって図 4.17 より輪郭の再現度が向上していることが分かる。

以上の実験結果から、**ORB-SLAM** がカメラの回転に対してロバストであると評価できる。さらに、この実験から提案システムに **ORB-SLAM** を利用するに当たって大きなメリットとなる事項を発見した。それは、高輝度箇所に対する特徴点検出力の高さである。

図 4.19, 4.21 内には映像内に蛍光灯によって白飛びしている部分が存在するが、そのコーナー点やエッジに対して **ORB-SLAM** は特徴点を正

確に検出できている。また、図 4.21 では蛍光灯とデスクの間仕切りがカメラ画面上では交差している部位があるが、その部分にも特徴点が検出されている。以上を考慮すると、ORB-SLAM には高コントラストの箇所を優先的に分析する傾向が強いといえる。

図 4.15 にて起動中のデスクトップ PC の壁紙が映っているが、キャラクターの目や髪の毛のハイライト部分に特徴点を検出していること、ペットボトルに反射した蛍光灯の光点に特徴点を検出していることはこの特性によるものと考えられる。

なお、ORB-SLAM は机の下などの光量に乏しい環境下においては特徴点を 1 つも検出できなかった。この原因も、上記のコントラストに対する特性に繋がると考えられる。

本研究で提案するシステムでは、回転に対するロバストさはもちろん、デバイスに LED ランプを搭載しそれを SLAM によって位置推定することでスケールを固定する関係上、高輝度点に対して正確な検出力を誇ることは極めて大きい利点である。また、液晶画面に表示した光点を認識できる可能性がある点から、5.2 節に述べる入力デバイスのスマートフォンによる簡略化を着想した。

以上の実験結果を以て、当面の間は提案システムの開発に ORB-SLAM を用いることを決定した。

4.3. 実装中のシステムの構築方針

本論文執筆時点では、第 3 章で述べたシステム系のうち、周囲環境を点群データに変換し、3 次元再構成を行う部分までを実装した。本節では、一部 3.5 節と重複するものの、今後の研究を見据え現在実装中のシステムの残りの部分に関する現段階での設計方針をまとめる。

ペン型デバイスのシステムへの認証部分には、マーカ式 AR 用のライブラリを組み合わせる方法を採用する。これにより、ペン型デバイス後端の 4 つの認証・軸補正用 LED を設置した板状の部分にテクスチャを描き、LED ライトによる位置認識に加えてテクスチャによる角度計算を実現することを意図する。

そして、システム内で最も重要となる描画用 LED の位置登録については、あらかじめペン型デバイスの 3D モデルを作成し、それを先述した角度算出の

手法に基づいて傾けることでペン先の描画用 LED の位置を割り出す方法を採用予定である。なお、事前に現実物体やシーンの 3D モデルを作成し、それを用いて幾何学的位置合わせを行う手法は複雑な周囲環境の再構成を目的とした SLAM 分野で T.Dummond らによって 2002 年に提唱されている[24]。マッチングする形状は Dummond らの場合と異なりペン型デバイスという単純な構造の物体ではあるものの、本手法はそこに発想を得た。

以上によって算出された LED 位置を point cloud 上に登録し、それらを毎フレーム連続的に逆投影し続けることでペン先の軌跡を画面上に再現する。この関係上、作成する 3D モデルは pcd 形式で保存が可能なソフトウェアによって作成する。pcd は point cloud を扱う上でのデファクトスタンダードライブラリである pcl に互換性のあるフォーマットで、これにより他形式ファイルのインポートに必要なライブラリを削り、わずかでも移植性を向上することを目標とする。

5. まとめと今後の方針

5.1. 本論文のまとめ

本論文では、インタラクティブな AR コンテンツの開発に向け、ペン型デバイスによる入力軌跡の重畳投影を目指した研究の現段階での進捗と設計方針の提案をまとめた。また、それを実現するためのマーカ・マーカレス AR 両者の説明やその構成技術、ならびに 3 次元空間をデータとして取り込む手法について述べ、各種 3 次元再構成技術の実験結果とその特性について考察した。その結果を以下の表 5.1 に示す。

表 5.1 ORB-SLAM、PTAM の各種特性評価

	PTAM	ORB-SLAM
単純な平面	×	○
パターンのある平面	○	◎
平面上の立体物	△	◎
エッジ	○	○
曲線	×	○
コーナー	◎	◎
高輝度点	△	◎
低輝度点	×	×
視点回転	×	○
計算コスト	○	△

現時点での提案手法にはまだ改善・比較の余地が多々残されており、またシステム全体のうち現状で構築が完了した部分は現実空間の点群化処理部分に留まる。そのため、高精度化を目指し、関連研究を参考に残りのシステム構築、および次節に示す機能向上案を追加する必要がある。

5.2. 今後の研究方針

今回の研究で扱った SLAM は全て特徴点ベースの手法であった。今後の研究課題として提案手法部分を独立ソフトとして設計し、ORB-SLAM だけでなく LSD-SLAM などの画素ベースの方式にも組み合わせたいと考えている。これにより、PTAM と ORB-SLAM、LSD-SLAM をそれぞれ 3 次元再構成ツールとして用い、提案手法に対する精度、速度、動作コストなどを比較する。

加えて、本研究では使用カメラは単眼カメラであったが、近年では kinect (Microsoft 製) の登場により現在距離センサ付きの RGB-D が安価に入手可能となっている。対応 SLAM に関しても、ORB-SLAM の研究チームから RGB-D カメラ対応版の ORB-SLAM2 もリリースされており、基本構造が共通する。この利点を活かし、RGB-D カメラに対する提案システムの移植により、精度を向上させることが期待できる。

また、上記の内容とは別に、入力デバイスの単純化も検討している。4.2 節で述べたように ORB-SLAM はコーナー点だけでなく、映像内の白飛びに近い高輝度部分に対しても高精度の認識が可能であった。さらに、映像内に映ったディスプレイが表示している画面といった非常に間接的な箇所に対しても、輝度が高い部分ならば認識することが分かった。これらから、提案システムの発展形として第 3 章の関連研究で用いられていた長方形上の入力デバイスをスマートフォン画面上の光点表示などに置き換え、認識を行うことが挙げられる。

むろん、実際には LED ランプは平面上から一定の高さだけ飛び出した立体物であり、それを完全に平面である画面表示にて単純に代用できるとは考えにくい。この問題の解決案には、4.3 節で示したマーカ式 AR の仕組みを SLAM と融合させたシステム構築が完了した際に、その理論を転用しスマートフォン自体をマーカ化させることで認識精度を補うことが考えられる。

今後、本研究と類似の方法で多種多様な AR アプリケーションが考案され、それらが高度化していくためには、スマートフォンという高い普及率を誇るデバイスを利用することが大きな一助となる。「誰もがどこでも利用できる」ことが拡張「現実」という概念の要点である。本研究で示すシステムをスマートフォンで実現できればその理念に大きく合致するものとなる。

一方で、ペン型デバイスというユーザにとって扱いやすい入力体系ならではのより高度なアプリケーションやコンテンツの構想も欠かさずに追及していきたい。現時点で考えている内容の 1 つは空中ディスプレイへの転用である。空間を動いたペンの軌跡をシステムが認識できるのならば、VR ゴーグルなどで空中投影したディスプレイに対しての操作に転用可能であると期待される。

6.参考文献等一覧

6.1. 引用注釈

- [1] 中口 俊哉、拡張現実感(AR)技術による外科手術支援の現状と課題 (特集 急速な実用化が進む注目技術がもたらす可能性とは? AR (拡張現実)技術がもたらすビジネスチャンスと新しい研究開発テーマ)
研究開発リーダー 10(6), 4-8, 2013-09、技術情報協会
- [2] AR サービス「ココアル」、<https://www.coco-ar.jp/>、
アクセス：2017/01/09
- [3] 神山朗, AR を用いた仮想展示システム開発に向けての基礎研究、琉球大学主催、工学部情報工学科、卒業研究発表会, 2011
- [4] 東京国立博物館 「トーハクナビ」について、
http://www.tnm.jp/modules/r_free_page/index.php?id=1467,
アクセス：2017/01/10
- [5] 国土交通省 AR による訪日外国人旅行者への案内情報の提供に向けて、
http://www.mlit.go.jp/sogoseisaku/region/sogoseisaku_region_tk_000008.html,
アクセス：2017/01/17
- [6] Pioneer ナビシステムアップ ND-HUD10,
http://pioneer.jp/carrozzeria/system_up/navi/nd-hud10/,
アクセス：2017/01/08
- [7] moguraVR ポケモン GO にも応用可能か MIT、AR 内の物体が現実干渉したように見せる技術を発表,
<http://www.moguravr.com/interactive-dynamic-video-ar/>,
アクセス：2017/01/24
- [8] AR ナビキャラ、https://www.arnavichara.com/ar_about2.html、
アクセス：2017/01/09
- [9] マイナビニュース コンピュータビジョンのセカイ - 今そこにあるミライ
6 拡張現実感「AR」(2) - マーカ有り AR とマーカレス AR の仕組み、
http://news.mynavi.jp/series/computer_vision/006/,
アクセス：2017/01/09
一部改変
- [10] 任天堂 AR ゲームズ、<https://www.nintendo.co.jp/hardware/3ds/ar/index.html>、
アクセス：2017/01/10
- [11] SEGA 週間ディーヴァステーション、<http://info.miku.sega.jp/914>、
アクセス：2017/01/10

- [12] 【コンピュータビジョン】ネコと学ぶエピソード幾何、
<http://qiita.com/ykoga/items/14300e8cdf5aa7bd8d31>、
 アクセス：2016/12/29
 一部改変
- [13] 高校数学の美しい物語 特異値分解の定義，性質，具体例、
<http://mathtrain.jp/svd>、アクセス：2017/01/12
- [14] マイナビニュース コンピュータビジョンのセカイ - 今ここにあるミライ 2
 動画編集技術「マッチムーブ」(1)、
http://news.mynavi.jp/series/computer_vision/002/、
 アクセス：2017/01/13
 一部改変
- [15] 佐藤智和，特徴点追跡による動画像からの
 逐次三次元復元とその応用
 一座標系の基礎から応用事例・最新研究動向まで一，SSII2015 チュートリアル
<http://yokoya.naist.jp/paper/datas/1407/SSII%E3%83%81%E3%83%A5%E3%83%BC%E3%83%88%E3%83%AA%E3%82%A2%E3%83%AB%E4%BD%90%E8%97%A4.pdf>、
 アクセス：2017/01/12
- [16] G. Klein and D. Murray: "Parallel Tracking and Mapping for Small AR Workspaces", 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), pp.1-10 (2007)
- [17] 亀田 能成、マーカレス AR, 映像情報メディア学会誌：映像情報メディア 66(1), 45-51, 2012-01-01
- [18] Raul Mur-Artal, J. M. M. Montiel, Juan D. Tardos, ORB-SLAM: a Versatile and Accurate Monocular SLAM System, IEEE Transactions on Robotics (Volume: 31, Issue: 5, Oct. 2015)
- [19] 香山沙織ら，コーナー特徴を用いた画像処理と環境モデルの比較に基づくロケーションシステム，システム制御学会論文誌、vol28, No6, pp249-257, 2015
- [20] 梅澤雄司、浜本隆之、インタラクティブな 3D 描画に向けた visual SLAM ベースの AR システム, PCSJ/IMPS2016、P-4-19
- [21] T.Ziner, J.Schmidt, H.Niemann:” Point Set Registration with Integrated Scale Estimation” , In PRIP 2005, pp.116~119,2005 :
- [22] 本澤宏明，ステレオ照合顔面三次元データの標準モデル下 ICP アルゴリズム利用補正，大学院研究年報 理工学研究科篇，第 43 号
 ISSN: 1345-2428 /2013
- [23] 日本色研 いろのはなし L*a*b 表色系，

<http://www.sikiken.co.jp/colors/colors11.html>, アクセス : 2017/01/20

- [24] T.Dummond et al., Real-Time Visual Tracking of Complex Structures, PAMI, 2002

6.2. その他参考文献等一覧

- A 佐藤弘康, 同次座標系と透視投影, 東京電機大学情報環境学部「情報数学 III」
-
- B 北山 洋幸, OpenCV 3 基本プログラミング, ISBN 978-4-87783-398-5

図表目次

図 2.1	典型的な AR マーカの実例[8]	5
図 2.2	マーカ式 AR の処理手順[9]	5
図 2.3	ニンテンドー3DS のマーカ式 AR 機能[10]	8
図 2.4	プレイステーション VITA のマーカレス AR 機能[11]	8
図 2.5	エピポーラ幾何概略図	10
図 2.6	エピポーラ拘束の概要図	11
図 2.7	行列 Σ の構造[12]	13
図 2.8	カメラ行列導出のフローチャート[13]	14
図 2.9	最終的な SfM の動作[14]	16
図 2.10	SLAM と SfM の比較[15]	17
図 2.11	ワールド座標とカメラ座標、およびその変換	18
図 2.12	理想的な投影モデル	19
図 2.13	ORB-SLAM 概要図[18]	25
図 2.14	先行研究での入力デバイス[20]	26
図 3.1	ペン型デバイスの概形	27
図 3.2	各種座標系の関係	29
図 3.3	L^*a^*b 色空間モデル[23]	34
表 4.1	実験環境	37
図 4.1	使用したキャリブレーション用チェッカーボード	38
図 4.2	カメラキャリブレーションの様子	39
図 4.3	PTAM の動作結果 1/2 (入力: bag ファイル)	40
図 4.4	PTAM の動作結果 2/2 (入力: bag ファイル)	41
図 4.5	ORB-SLAM による 3次元特徴点検出 1/5(入力: bag ファイル)	42
図 4.6	上図時点での 3次元再構成結果 1/5	43
図 4.7	ORB-SLAM による 3次元特徴点検出 2/5(入力: bag ファイル)	43
図 4.8	上図時点での 3次元再構成結果 2/5	44
図 4.9	ORB-SLAM による 3次元特徴点検出 3/5(入力: bag ファイル)	44
図 4.10	上図時点での 3次元再構成結果 3/5	45
図 4.11	ORB-SLAM による 3次元特徴点検出 4/5(入力: bag ファイル)	45
図 4.12	上図時点での 3次元再構成結果 4/5	46
図 4.13	ORB-SLAM による 3次元特徴点検出 5/5(入力: bag ファイル)	46
図 4.14	上図時点での 3次元再構成結果 5/5	47
図 4.15	ORB-SLAM による 3次元特徴点検出 1/4(入力: カメラ画像)	49

図 4.16	上図時点での 3 次元再構成結果 1/4	49
図 4.17	ORB-SLAM による 3 次元特徴点検出 2/4(入力: カメラ画像)	50
図 4.18	上図時点での 3 次元再構成結果 2/4	50
図 4.19	ORB-SLAM による 3 次元特徴点検出 3/4(入力: カメラ画像)	51
図 4.20	上図時点での 3 次元再構成結果 3/4	51
図 4.21	ORB-SLAM による 3 次元特徴点検出 4/4(入力: カメラ画像)	52
図 4.22	上図時点での 3 次元再構成結果 4/4	52
表 5.1	ORB-SLAM、PTAM の各種特性評価	56