

エネルギー科学研究科
エネルギー社会・環境科学専攻修士論文

題目： 振り子運動する RGB-D カメラを
用いた 3次元再構成手法の開発

指導教員： 下田 宏 教授

氏名： 二神 繁暁

提出年月日： 令和7年2月5日（水）

論文要旨

題目：振り子運動する RGB-D カメラを用いた 3 次元再構成手法の開発

エネルギー情報学分野 (下田研究室), 二神繁暁

要旨：

近年、現実環境を模した 3 次元モデルが、原子力発電所での作業支援や VR を用いた災害体験などの様々な分野で活用されている。現実環境を模した 3 次元モデルを作成する手法の一つに、カメラを用いて環境を撮影した画像から 3 次元モデルを作成する手法がある。この手法は、レーザスキャナなどの大掛かりな機材を必要とせず、カメラのみを用いるため利用しやすいという利点がある。この手法では、多くの場合、環境を撮影した複数の画像について、各画像を取得した瞬間のカメラの 3 次元座標と向きを含むカメラ姿勢を画像中の特徴を手掛かりに推定する。そして、推定されたカメラ姿勢に基づいて 3 次元モデルを作成する。しかし、特徴が少ない画像からカメラ姿勢を推定することは困難であるため、単色の平坦な壁など、特徴が少ない領域は作成される 3 次元モデルから抜け落ちてしまうという問題がある。そこで本研究では、振り子の先端に RGB-D カメラを取り付けた振り子カメラを用いて環境を撮影し、画像の撮影時刻と振り子の運動法則を利用してカメラ姿勢を推定することで、特徴の少ない領域でも抜け落ちなく 3 次元再構成できる手法を開発することを目的とする。

この方法では、まず振り子カメラを振動させて環境を連続的に撮影する。次に、撮影した画像の中で十分な特徴点を持つ画像のみでトラッキングを行い、振り子運動の一部でのカメラ姿勢を推定する。その後、推定したカメラ姿勢を用いて振り子の運動方程式のパラメータを推定する。そして、撮影時刻と推定したパラメータを用いた運動方程式から、撮影した全画像のカメラ姿勢を推定する。最後に、推定したカメラ姿勢を用いて 3 次元再構成を行い、3 次元モデルを取得する。

提案手法が目的を達成していることを確認するため、Unity で作成した理想的な仮想環境と現実環境の両方で評価実験を行った。仮想環境で行った実験では、提案手法は、全画像のカメラ姿勢で真値に近い値を推定でき、既存手法よりも推定精度が高いことを確認できた。一方、現実環境で行った実験でも、特徴の少ない領域でも抜け落ちなく 3 次元再構成できたが、作成したモデルは既存手法に比べて精度が低かった。そこで、振り子を吊るす支柱部分の質量も考慮するように運動方程式を修正し、改めて実験を行った。その結果、作成したモデルの詳細が鮮明になった。以上より、提案手法は仮想環境と現実環境の両方で、特徴の少ない領域でも 3 次元再構成ができることを確認できた。

目次

第 1 章 研究の背景と目的	1
1.1 研究の背景	1
1.2 研究の目的	2
1.3 論文の構成	3
第 2 章 関連研究	4
2.1 レーザスキャナを用いた 3 次元再構成	4
2.2 画像を用いた 3 次元再構成	5
2.2.1 カラー画像を用いた 3 次元再構成	5
2.2.2 カラー画像とデプス画像を併用した 3 次元再構成	5
2.2.3 機械学習を用いた 3 次元再構成	6
2.3 特徴の少ない領域に注目した 3 次元再構成	7
2.4 本研究の位置づけ	7
第 3 章 振り子運動する RGB-D カメラを用いた 3 次元再構成手法の提案	8
3.1 提案手法の概要	8
3.2 提案手法で使用する座標系	10
3.3 運動方程式のパラメータ推定のための多特徴点画像のトラッキング	12
3.3.1 多特徴点画像の抽出	12
3.3.2 SfM を用いたカメラ姿勢のトラッキング	13
3.4 運動方程式のパラメータ推定と全画像のカメラ姿勢の推定	13
3.4.1 トラッキング結果を用いた振り子の角度の推定	15
3.4.2 運動方程式のパラメータの最適化	15
3.4.3 運動方程式を用いた全画像のカメラ姿勢の推定	21
3.5 3 次元再構成の実行	23
第 4 章 提案手法の実装	24
4.1 ソフトウェアの実装	24

4.1.1	画像の撮影と保存を行うソフトウェアの実装	24
4.1.2	画像の処理と3次元再構成を実行するソフトウェアの実装	25
4.2	シミュレーションを用いたパラメータの最適化順の決定	26
4.2.1	使用したシミュレーション環境と画像の撮影条件	27
4.2.2	シミュレーション方法	29
4.2.3	パラメータの最適化順の決定	30
4.3	シミュレーションによる実装ソフトウェアの動作検証	39
4.4	ハードウェアの実装	45
第 5 章	現実環境を撮影した画像セットによる提案手法の評価	52
5.1	評価の目的	52
5.2	評価に用いた環境とその環境の撮影方法	52
5.2.1	評価に用いた環境と振り子カメラの設置位置	52
5.2.2	環境の撮影方法と取得した画像の前処理	54
5.3	運動方程式と提案手法の処理に必要なパラメータの設定	55
5.4	結果と考察	57
5.4.1	取得した画像と多特徴点画像の抽出結果	57
5.4.2	カメラ姿勢の推定結果と3次元再構成結果および考察	57
5.5	支柱の質量を考慮した場合の3次元再構成	62
5.5.1	運動方程式の修正	62
5.5.2	3次元再構成を行った結果と考察	62
第 6 章	本研究のまとめと今後の展望	67
6.1	本研究のまとめ	67
6.2	今後の展望	68
	謝 辞	69
	参 考 文 献	70
	付録 A 支柱の質量を考慮した運動方程式の導出	74

目次

1.1	カメラ姿勢を用いた3次元再構成の原理	2
3.1	提案手法の概要	9
3.2	振り子カメラの概要	10
3.3	提案手法で使用する座標系	11
3.4	ステップ1の処理の流れ	12
3.5	特徴の多い画像と少ない画像の例	13
3.6	ステップ2の処理の流れ	14
3.7	振り子の角度の定義	15
3.8	最適化に使用する振り子の角度	16
3.9	持ち上げ角度の定義	17
3.10	ステップ2-2の処理の流れ	19
3.11	数値計算で画像の撮影時刻における振り子の角度を求める処理の流れ	20
3.12	振り子の長さ L の最適化の流れ	22
4.1	画像の撮影と保存を行うソフトウェアの処理の流れ	24
4.2	画像間の差分が少ない画像	26
4.3	シミュレーションで使用した3次元モデルの外観	27
4.4	振り子の長さの真値からのずれとエラー値の関係	31
4.5	抵抗係数の真値からのずれとエラー値の関係	31
4.6	持ち上げ角度の真値からのずれとエラー値の関係	32
4.7	初期角速度の真値からのずれとエラー値の関係	32
4.8	振り子の長さ・抵抗係数とエラー値の関係	33
4.9	振り子の長さ・持ち上げ角度とエラー値の関係	34
4.10	振り子の長さ・初期角速度とエラー値の関係	35
4.11	抵抗係数・持ち上げ角度とエラー値の関係	36
4.12	抵抗係数・初期角速度とエラー値の関係	37
4.13	持ち上げ角度・初期角速度とエラー値の関係	38

4.14 最適化したパラメータを用いた多特徴点画像の振り子の角度の推定結果	40
4.15 カメラ姿勢の真値を用いた仮想環境の3次元再構成結果	42
4.16 既存手法 (COLMAP) によって得られた仮想環境の3次元再構成結果	42
4.17 提案手法による仮想環境の3次元再構成結果	43
4.18 最適化したパラメータを用いた全画像の振り子の角度の推定結果	43
4.19 各再構成結果の壁部分を真上から見た結果の比較	44
4.20 カメラ姿勢の真値を用いた仮想環境の再構成結果と提案手法による仮想 環境の再構成結果の比較	45
4.21 実装した振り子カメラの全体図	46
4.22 振り子カメラの構成図	49
4.23 支点部分の模式図	50
4.24 振り子カメラの先端部分	51
5.1 評価に用いた環境の見取り図	53
5.2 評価に用いた環境	53
5.3 振り子カメラの初期位置	56
5.4 全画像の振り子の角度の推定結果	58
5.5 既存手法 (COLMAP) で得られた現実環境の3次元再構成結果	58
5.6 提案手法による現実環境の再構成結果	59
5.7 既存手法 (COLMAP) と提案手法の現実環境の3次元再構成結果の比較	59
5.8 COLMAP のトラッキング結果を用いた多特徴点画像の現実環境の3次 元再構成結果	60
5.9 多特徴点画像の振り子の角度の推定結果	61
5.10 運動方程式修正前後の3次元再構成結果の比較 (全体)	63
5.11 運動方程式修正前後の3次元再構成結果の比較 (扉部分)	64
5.12 運動方程式修正後の多特徴点画像の振り子の角度の推定結果	65
A.1 仮定した振り子の図	74

表 目 次

4.1	シミュレーションで使用する RGB-D カメラの仕様	28
4.2	運動方程式のパラメータの変化範囲	30
4.3	各パラメータの初期値と終了条件	39
4.4	運動方程式のパラメータの最適化結果	40
4.5	撮影に使用した RGB-D カメラの仕様	47
4.6	使用したバッテリーの仕様	48
4.7	使用したメイン PC の仕様	48
5.1	カラー画像およびデプス画像の詳細	54
5.2	運動方程式のパラメータの初期値	55
5.3	各パラメータの最適化後の値	62
5.4	運動方程式修正後の各パラメータの最適化結果	65

第 1 章 研究の背景と目的

1.1 研究の背景

近年、デジタルツインが普及するとともに、現実環境を計測して作成した 3 次元モデルが様々な分野で活用されている。原子力発電所では、発電所全体を 3 次元モデル化することで、仮想現実感 (VR) を活用した作業のシミュレーション^[1]や、遠隔からの稼働設備の監視・検査が可能になった^[2]。また、3 次元モデルは建物の設計の際の必要要件の確認や地震応答の解析^[3]などにも活用されている。さらに防災分野では、実在する環境での災害発生を模擬体験できる VR システムの実現に用いられたり^[4]、ドローンを活用した災害発生時の被害状況の把握や 2 次災害の防止に活用されたりしている^[5,6]。そのほか、3 次元モデルは都市計画^[7]や文化財の保存^[8]など幅広く活用されている。

このような現実の物体や環境の 3 次元モデルを作る手法は 3 次元再構成と呼ばれる。3 次元再構成の手法には、レーザスキャナを用いる手法^[8-11]、カラー画像を用いる手法^[12-15]、各ピクセルが被写体との距離を表すデプス画像とカラー画像を併用する手法^[16-18]、画像に機械学習を適用して必要な情報を推定する手法^[19-21]などがあり、作成する 3 次元モデルの規模やモデルに求められる精度に応じた手法が使用される。画像を用いる手法は、カメラで環境を撮影し、取得した画像をもとに 3 次元再構成を行う。この手法は、環境を撮影するためにレーザスキャナなどの大掛かりな機材を必要とせず、カメラのみを用いるため利用しやすいという利点がある。画像を用いる手法の一般的な処理の流れを以下に示す。まず、カメラで撮影された画像から画像処理で認識した特徴点を利用して、画像が撮影された瞬間のカメラの位置と方向 (以下、カメラの位置と方向を合わせて「カメラ姿勢」と呼ぶ) を推定する。次に、図 1.1 に示すように、推定したカメラ姿勢と画像内の特徴点の位置から三角測量の原理を用いて、複数の画像に共通して写る特徴点の 3 次元位置を求める。これを全ての特徴点について行うことで、3 次元モデルを作成する。しかし、単一色の平坦な壁など特徴の少ない領域のみが写った画像は、十分な数の特徴点を認識できず、画像処理によってカメラ姿勢を推定することは困難であるため、3 次元再構成に使用できない。そのため、特徴の少ない領域は作成する 3 次元モデルから抜け落ちてしまうという問題がある。

この問題を解決するためには、画像内の情報以外の情報を活用してカメラ姿勢を推定し、3次元再構成を行う必要がある。画像内の情報以外の情報として活用できると考えられるものの1つに、カメラが従う物理運動法則がある。既知の運動法則に従ってカメラを運動させて環境を撮影するとともに、各画像の撮影時刻を記録すれば、運動法則と撮影時刻からカメラの姿勢を求められる可能性がある。特に、比較的単純な物理法則に従う振り子を用いれば、運動方程式と撮影時刻から、各画像を撮影した際の正確なカメラ姿勢が推定できる可能性がある。

ただし、各画像を撮影した際のカメラ姿勢が推定できただけでは、カラー画像のみを使用する場合、三角測量の基準となる画像間で共通する特徴点を識別できない。そのため、三角測量を行うことができず、特徴の少ない領域を3次元再構成することができない。そこで本研究では、カラー画像とデプス画像を併用する手法を採用することで、3次元再構成を行う。

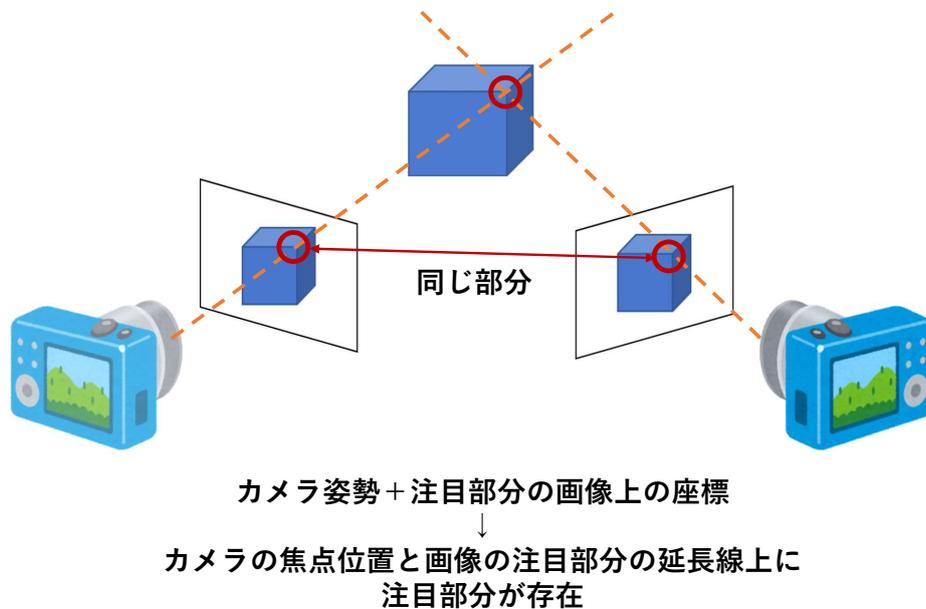


図 1.1: カメラ姿勢を用いた3次元再構成の原理

1.2 研究の目的

本研究では、振り子の先端に、カラー画像とデプス画像の両方を同時に撮影可能なRGB-Dカメラを取り付けた振り子カメラを用いて環境を撮影し、撮影時刻と振り子の運動法則を利用してカメラ姿勢を推定することで、特徴の少ない領域でも抜け落ちなく3次元再構成できる手法(以下、提案手法)を開発することを目的とする。

提案手法を用いることで、特徴の少ない領域を含む3次元モデルを得られるだけでなく、以下のような利点がある。

- 振り子カメラは3次元再構成に適した速度でカメラを振動させながら自動的に環境を撮影できる。人がカメラを持って環境を撮影する場合には、3次元再構成を精度高く実行するために、「カメラを速く動かさない」など注意すべきことが多いが、振り子カメラを用いれば、撮影時の注意点を完全に理解していない人でも容易に利用できる^[22]。
- カメラの移動方向が限定されるため、モデルの質を低下させる要因となる手振れによる画像のブレなどを後処理で排除しやすい^[23]。
- カメラの姿勢を少しずつ変化させながら同じ領域を繰り返し撮影することが容易になるため、超解像処理と組み合わせることにより、被写体の詳細な3次元モデルを構築することが可能になる^[24]。

1.3 論文の構成

本論文は全6章で構成される。以下、第2章では、関連研究について述べる。第3章では、提案手法の概要について述べた後、提案手法の各ステップにおける処理の詳細を述べる。第4章では、まず、ソフトウェアの実装について述べる。次に、Unityを用いたシミュレーションと実装ソフトウェアの動作検証について述べる。その後、ハードウェアの実装について述べる。第5章では、提案手法を評価する為に実施した現実環境での実験とその結果について述べる。最後に、第6章では本研究のまとめと、今後の課題を述べる。

第 2 章 関連研究

本章では、既存の代表的な 3 次元再構成手法として、レーザスキャナを用いる手法と画像を用いる手法について述べる。また、本研究と同様に、特徴の少ない領域でも抜け落ちなく 3 次元再構成することを目指した手法についても述べる。

2.1 レーザスキャナを用いた 3 次元再構成

レーザスキャナを用いた 3 次元再構成は、文化財の保存や環境の計測など様々な場面で活用されている^[8,9]。大阪大学適塾記念センターは、レーザスキャナを活用して適塾の建物の 3 次元計測を行い、測定結果をもとに 3 次元モデルを作成することで、適塾の建物を後世に守り伝える取り組みを行っている^[8]。このようなレーザスキャナを用いる手法では、測定したい環境内にレーザスキャナを設置した上で、測定対象に照射したレーザが反射して戻るまでの時間を計測することで、測定対象までの距離を算出する。この処理を測定対象の表面に密に繰り返すことにより、測定対象の表面形状の情報を得ることができる。この手法は、現実の環境を高精度かつ正確に再構築することができるというメリットがある反面、高性能なレーザスキャナは高価であり、手軽に利用しにくいというデメリットがある。また、屋内など複雑な形状をした環境では、抜け落ちのないモデルを作成するために、死角が生じないように何度もレーザスキャナを移動させながら測定する必要があり、データ収集に手間がかかる。

そのため、近年では大掛かりなレーザスキャナに代わり、LiDAR スキャナを用いた 3 次元再構成手法が開発され、実際に活用されている^[10,11]。LiDAR スキャナは大掛かりなレーザスキャナに比べて価格が安く、サイズも小さいため、手軽に利用することができる。しかし、LiDAR スキャナはレーザスキャナに比べて空間解像度が低く、環境の 3 次元再構成における精度の面で劣る。

いずれの方法も、原子力プラント内の管理区域など、長期間の滞在が難しく、取扱いに注意が必要な設備の多い環境では、詳細な環境の計測が難しい。また、レーザスキャナも LiDAR も複数回に分けて環境を計測した場合、その結果を統合するための処理が必要であるため、計測結果だけを用いて外見や形状に特徴のない領域を統合することは困難である。

2.2 画像を用いた3次元再構成

画像を用いた3次元再構成には、カラー画像のみを用いる手法、カラー画像とデプス画像を併用する手法、取得した画像に機械学習を適用する手法など、様々な手法が開発されている。以下では、機械学習を適用する以外の手法の詳細を述べた後、機械学習を適用する手法について述べる。

2.2.1 カラー画像を用いた3次元再構成

カラー画像を用いた3次元再構成手法には、Structure from Motion(以下, SfM), Multi View Stereo(以下, MVS), および Visual Simultaneous Localization and Mapping(以下, Visual SLAM) などがある。SfMとは、異なる視点から撮影された画像を用いて3次元構造を再構築する技術であり、同時にカメラ姿勢を推定することもできる。MVSとは、多視点の画像とカメラ姿勢から、より密な3次元構造を復元する技術である。SfMとMVSを組み合わせることで、入力画像から精度の高い3次元モデルを作成できる。一方でVisual SLAMは、動画などの連続して入力される画像に対してカメラ姿勢を推定したり、環境の3次元モデルを作成したりする技術であり、リアルタイム性を重視している点が特徴である。SchönbergerらはSfMとMVSを組み合わせた汎用的な3次元再構成システムであるCOLMAPを開発した^[12,13]。これは、大規模なシーンや連続していない画像セットの入力など様々な場面で活用でき、詳細なパラメータの設定を行うことが可能である。また、Gaoらはドローンとロボットを組み合わせることで屋内環境の再構成を行う手法を開発した^[14]。さらに、Mur-Artalらは室内、屋外環境などの様々な環境下で活用可能で、リアルタイムに動作するORB-SLAMを開発した^[15]。

以上のように、カラー画像を用いた3次元再構成手法には様々なものが開発されているが、これらの手法は画像内に存在する特徴点をもとにカメラ姿勢を推定するため、特徴が少ない領域を写した画像では特徴点を抽出できず、カメラ姿勢が推定できない。また、カラー画像のみでは原理的に被写体のスケールを推定できないため、追加の処理でスケールを推定する必要がある。

2.2.2 カラー画像とデプス画像を併用した3次元再構成

カラー画像に加えてデプス画像を用いた3次元再構成手法では、デプス画像を利用することで被写体のスケール情報を含めた3次元モデルを取得できる。また、デプス画像の情報を用いることで、安定性の向上や計算コストの削減にもつながる。さらにIntel

社の RealSense^[25] などの低コストな RGB-D カメラが登場したことによって、この手法は広く使用されるようになった。Choi らは室内を撮影したカラー画像とデプス画像の画像セットをセグメント毎に分割して 3 次元再構成を行い、セグメント間で共通して存在するものを利用して分割したモデルを合成することで、部屋全体の高精度な 3 次元モデルを作成した^[16]。Mur-Artal らは、単眼カメラのみに対応していた ORB-SLAM を RGB-D カメラやステレオカメラにも対応させ、精度を向上させた ORB-SLAM2 を開発した^[17]。Dai らの開発した BundleFusion は、大規模な屋内環境でリアルタイムに 3 次元モデルを作成することができ、オフラインの手法と比較しても優れた精度を示している^[18]。

しかし、これらの手法もカラー画像を利用した手法と同様に、画像処理を用いてカメラ姿勢を推定しているため、特徴の少ない領域を写した画像のカメラ姿勢は推定できない。そのため、カメラ姿勢の推定には画像内から得ることのできる特徴点やデプスの値以外の情報を用いる必要がある。

2.2.3 機械学習を用いた 3 次元再構成

近年、取得した画像に機械学習を適用することで、3 次元再構成を行う手法の開発が進んでいる。Neural Radiance Fields(NeRF)^[19] は機械学習を利用した 3 次元再構成の技術として注目を集めている。これは、入力画像から精密な 3 次元モデルを作成できるだけでなく、入力画像には存在しない新たな視点からの詳細な画像を生成できる。さらに、Yu らは NeRF の精度を維持しつつその計算速度を大幅に向上させる手法を開発した^[20]。また、Barron らはレンダリングの効率化を行うことで、高効率で高速な再構成が可能な手法を開発する^[21] など、NeRF の性能や汎用性向上のための様々な研究が行われている。

しかし、機械学習を用いた 3 次元再構成手法では、計算能力の向上や、複雑な環境を詳細に再構成することに主眼がおかれており、特徴の少ない領域に注目した研究は行われていない。また、機械学習を用いて作成される 3 次元モデルは、本来存在するはずのものが欠落したり、逆に存在しないはずのものが生成されたりする可能性がある。この問題は、機械学習の研究が進み、その精度が上がるにつれて改善はされるが、完全に排除することは困難である。そのため、原子力プラント内部のモデル化など、高い正確性が求められる場面で利用することは難しい。

2.3 特徴の少ない領域に注目した3次元再構成

特徴の少ない領域の3次元再構成手法についてもいくつか研究されている。Zhangらは、LiDARセンサで得たフロアマップから室内の3次元モデルを作成し、テクスチャ情報を持つ画像をマッピングすることで3次元モデルを作成する手法を提案した^[26]。しかし、この手法は単純な屋内環境を想定しているため、屋内環境が複雑な場合には適用できない。また、Gaoらは大規模建造物の抜け落ちが少ない3次元再構成のために、レーザスキャンで得られたデータと、地上と空中からそれぞれ撮影した画像を組み合わせた3次元再構成手法を開発した^[27]。この手法は特徴の少ない領域や複雑な領域など、画像による再構成が難しい領域をレーザスキャンで得られたデータで補うことにより、シーンの抜け落ちが少ない再構成を実現する。一方で、この手法は建造物を外から撮影することを想定しており、屋内環境での検証は行われていない。また、これら2つの手法は画像とレーザスキャナによるスキャン結果を利用しており、画像のみを用いた特徴の少ない領域の3次元再構成は行われていない。

2.4 本研究の位置づけ

第2.1節から第2.3節で紹介したように、3次元再構成手法にはレーザスキャナやLiDARセンサ、カラー画像やデプス画像を用いた手法など様々な手法が開発されているが、特徴の少ない環境で使用可能な、カメラのみを用いた手法は存在しない。そこで本研究では、振り子の運動法則を利用した、特徴の少ない領域でも抜け落ちなく3次元再構成できる手法を開発することを目的とする。本研究により、カメラのみで簡単に特徴の少ない領域を含むモデルを作ることができるようになる。また、振り子カメラは一度運動を開始すると自動で撮影を継続するため、環境撮影の手間を軽減することもできる。そのため、長時間の滞在が難しく、環境の計測が困難な原子力発電所内部などの領域でも、抜け落ちのない3次元モデルを取得できる可能性がある。

第 3 章 振り子運動する RGB-D カメラを用いた 3次元再構成手法の提案

本章では，振り子カメラを用いた3次元再構成手法の概要について説明し，その後，提案する手法の詳細について説明する．

3.1 提案手法の概要

提案する振り子カメラを用いた3次元再構成手法の概要を図 3.1 に示す．

提案手法では，カラー画像とデプス画像の両方を取得できる RGB-D カメラを使用して環境を撮影する．カメラは図 3.2 に示すように，紐ではなく細い支柱を用いて吊るす．カメラを紐で吊るすと，カメラ本体が紐を軸にして回転し，撮影したい方向とは異なる方向を撮影してしまう可能性がある．そのため，細い支柱を用いてカメラを吊るすことで，カメラ本体が意図しない方向に回転することを防ぐ．一方，振り子運動のための支柱の回転に関しては，支柱の回転中心にベアリングを用いることで，振動時の摩擦を低減する．これにより，カメラの振動軌道が平面上に制限され，後述するカメラ姿勢の推定を簡易にすることもできる．

RGB-D カメラを用いて環境を撮影した後の処理は，以下の3つのステップに分けられる．まず，振り子の長さや抵抗係数などの振り子の運動方程式のパラメータを推定するため，自然特徴点が多く写っている画像（以下，多特徴点画像）のカメラ姿勢を求める（以下，ステップ1）．次に，トラッキング結果から運動方程式のパラメータを推定し，推定したパラメータと画像の撮影時刻から運動方程式を用いて全ての画像のカメラ姿勢を推定する（以下，ステップ2）．最後に，推定したカメラ姿勢，カラー画像，デプス画像を用いて3次元再構成を行う（以下，ステップ3）．

次節以降では，まず提案手法で使用する座標系について説明した後，各ステップの詳細を説明する．

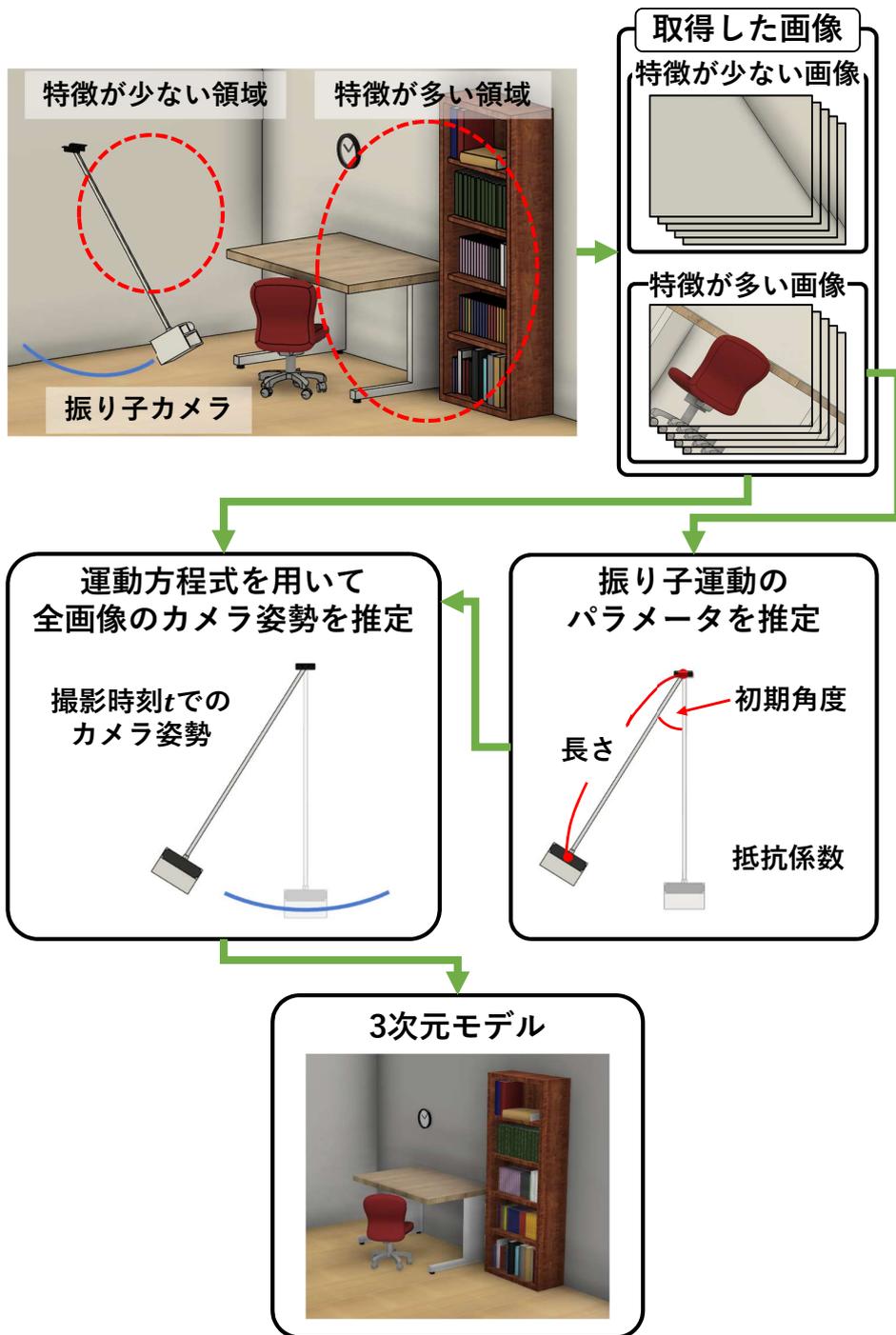


図 3.1: 提案手法の概要

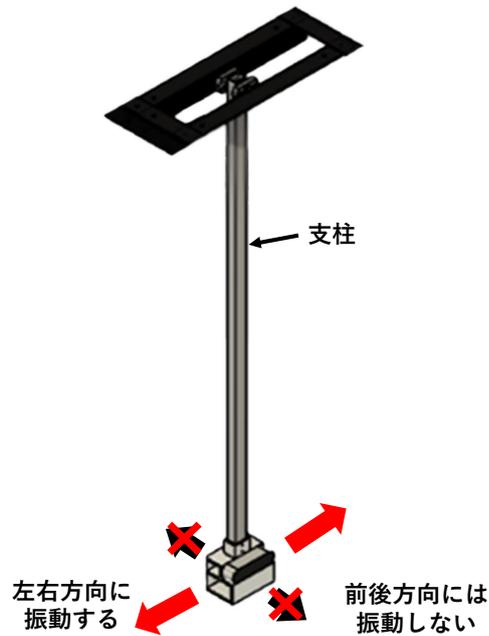


図 3.2: 振り子カメラの概要

3.2 提案手法で使用する座標系

提案手法で使用する座標系を図 3.3 に示す. 使用する座標系は世界座標系 W , 振り子座標系 P , カメラ座標系 C_n であり, すべての座標系は右手系である.

世界座標系 W とは, トラッキングの際に読み込んだ 1 枚目の画像のカメラ位置を原点とし, 画像上の y 軸方向の正の方向と同じ方向を y 軸正の方向, カメラの光軸方向を z 軸正の方向とする座標系である.

振り子座標系 P とは, 振り子の支点位置を原点とし, 世界座標系の原点へ向かう方向を y 軸正の方向, カメラの光軸方向を z 軸正の方向とする座標系である.

カメラ座標系 C_n とは, 画像ごとに異なる座標系であり, 各画像のカメラ位置を原点, 画像上の x 軸方向の正の方向と同じ方向を x 軸, y 軸方向の正の方向と同じ方向を y 軸, カメラの光軸の方向を z 軸とする支柱軸方向に回転しない座標系である.

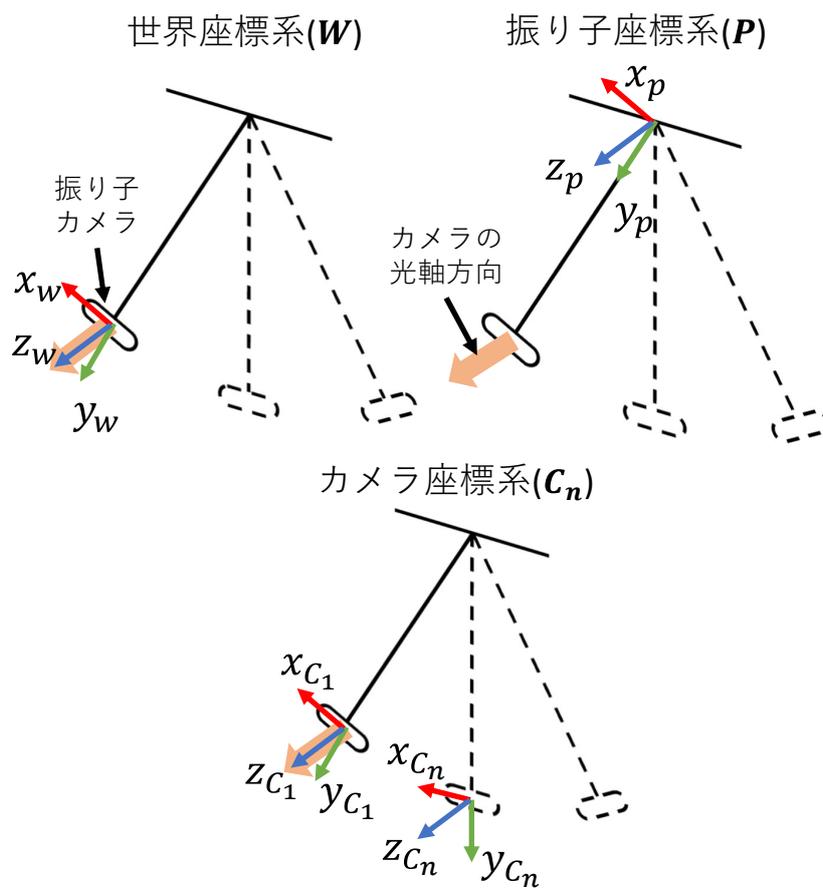


図 3.3: 提案手法で使用する座標系

3.3 運動方程式のパラメータ推定のための多特徴点画像のトラッキング

ステップ1の処理の流れを図3.4に示す。ステップ1ではまず、取得したすべてのカラー画像から多特徴点画像を抽出する(以下、ステップ1-1)。その後、抽出した多特徴点画像を用いて、カメラ姿勢を推定するためのトラッキングを行う(以下、ステップ1-2)。以下、それぞれのステップの詳細を説明する。

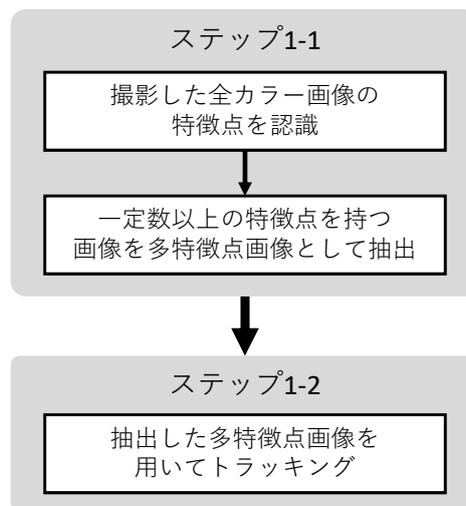


図 3.4: ステップ1の処理の流れ

3.3.1 多特徴点画像の抽出

ステップ1-1では、すべてのカラー画像上の特徴点を認識し、その後、多特徴点画像を抽出する。本研究は、特徴の少ない領域でも3次元再構成を実行できるようにすることを目的としているため、図3.5に示すように、環境を撮影した画像には特徴点の多い画像と少ない画像が含まれる。しかし、ステップ1-2でトラッキングを実行する際、画像に含まれる特徴点が十分でない場合、トラッキングが適切に実行できない場合がある。そこで提案手法では、特徴点の数が予め定めた閾値を下回る画像をステップ1-2の処理対象から除外することで、トラッキングの安定性を向上させる。特徴点の認識にはOriented FAST and Rotated BRIEF(以下、ORB)^[28]を使用する。ORBは、Scale-Invariant Feature Transform(以下、SIFT)^[29]やSpeeded Up Robust Features(以下、SURF)^[30]などの他の特徴点の認識手法と同程度の精度を持ち、それらより高速に処理を実行できる。また、特許による利用制限もないため、提案手法ではORBを採用



図 3.5: 特徴の多い画像と少ない画像の例

した. 多特徴点画像の抽出基準となる閾値の適切な値は, 撮影する環境によって異なるが, 本研究では環境に応じて適切な値を手動で設定するものとする.

3.3.2 SfM を用いたカメラ姿勢のトラッキング

ステップ 1-2 では, ステップ 1-1 で抽出したカラー画像を用いてトラッキングを行い, カメラ姿勢を推定する. 提案手法では, 既存の SfM 手法を用いて多特徴点画像のトラッキングを行う. 既存の SfM 手法には COLMAP^[12,13] や ORB-SLAM2^[17] などがあり, 基本的にはどの手法を使用することも可能である. 提案手法では, SfM によって得られるカメラ姿勢 D は式 (3.1) に示すように 4×4 の同次行列として表す. ただし R はカメラの方向を表す 3×3 の回転行列, \mathbf{x} はカメラの位置を表す 3 次元ベクトルである. また, カメラ姿勢は世界座標系を用いて表現される.

$$D = \begin{bmatrix} R & \mathbf{x} \\ \mathbf{0} & 1 \end{bmatrix} \quad (3.1)$$

3.4 運動方程式のパラメータ推定と全画像のカメラ姿勢の推定

ステップ 2 ではステップ 1 で得た多特徴点画像のトラッキング結果を基に振り子の運動方程式のパラメータを推定し, その後, 全ての画像について撮影されたときのカメラ姿勢を推定する. ステップ 2 の処理の流れを図 3.6 に示す. まず, ステップ 1 で得られた多特徴点画像のトラッキング結果を用いて, 撮影画像の 1 枚目のカメラ姿勢を基準とした角度を推定する (ステップ 2-1). 次に, 推定した振り子の角度から, 振り子の

運動方程式のパラメータを最適化する(ステップ2-2)。最後に、最適化したパラメータを用いて振り子の運動方程式の数値計算を行い、全ての画像が撮影されたときのカメラ姿勢を推定する(ステップ2-3)。以下、それぞれのステップの詳細を説明する。

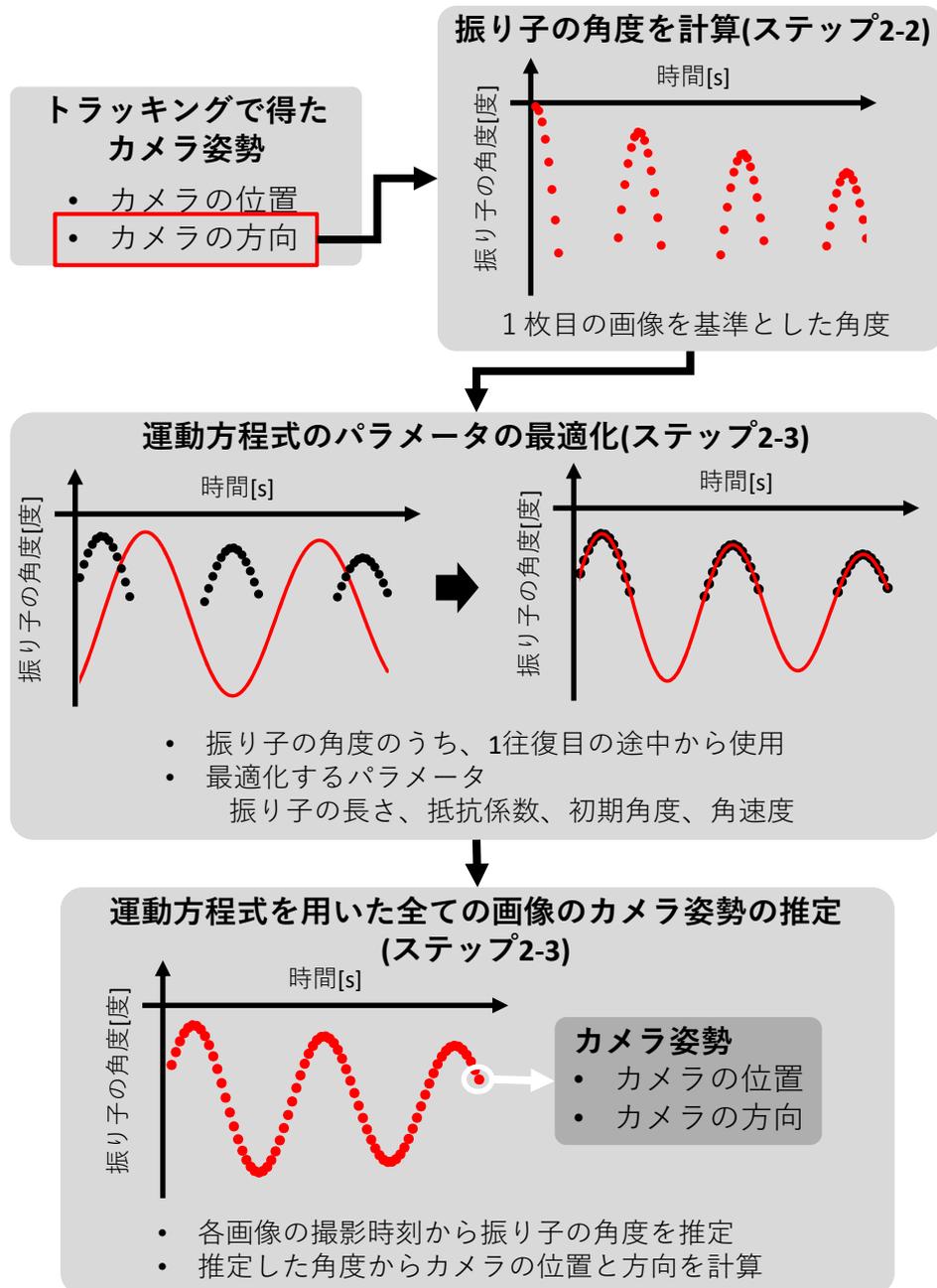


図 3.6: ステップ2の処理の流れ

3.4.1 トラッキング結果を用いた振り子の角度の推定

ステップ2-1では、トラッキングによって得られた多特徴点画像のカメラ姿勢から、各画像の撮影時刻における振り子の角度を計算する。ここで、振り子の角度とは、図3.7に示すように、1枚目の画像取得時点を基準とした各画像取得時の振り子のなす角を指す。提案手法では、振り子カメラは常に同一平面上を運動する設計としているため、振り子の角度は振り子座標系における z 軸周りの回転によって表される。また、振り子座標系は振り子カメラの支点位置が原点になるよう世界座標系を平行移動したものであるため、カメラの方向を表す回転行列は世界座標系と共通である。そのため、SfMを用いて得られたカメラ姿勢のうち、カメラの方向を表す回転行列から、世界座標系における z 軸周りの回転を求めることで振り子の角度を求めることができる。回転行列 R から振り子座標系における z 軸周りの回転 θ_{z_p} を求める式を式(3.2)に示す。

$$\theta_{z_p} = \arctan\left(\frac{R_{21}}{R_{11}}\right) \quad (3.2)$$

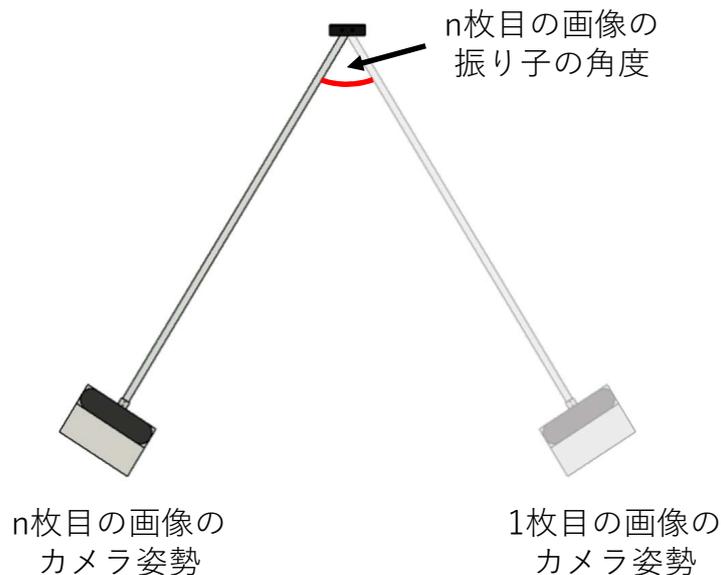


図 3.7: 振り子の角度の定義

3.4.2 運動方程式のパラメータの最適化

ステップ2-2では、ステップ2-1で計算した振り子の角度を利用して、振り子の運動方程式を定めるために必要なパラメータを最適化する。ただし、図3.8に示すように、最初の1往復の前半部分の振り子の角度は最適化に使用しないものとする。提

案手法では、振り子カメラを用いて環境を撮影する際、振り子の運動を開始した瞬間に正確に撮影を開始させることが困難であるため、時刻0で角速度が0であることを仮定できない。また、振り子カメラは手で一定の高さまで持ち上げ、手を離すことで運動を開始するため、開始直後は、手が干渉することによって振り子運動に従わない可能性がある。従って、多特徴点画像の最初の数枚は最適化に用いる画像から除外し、残った画像の1枚目を新たに時刻0とする。また、新たに定めた時刻0における角速度は測定できないため、これも最適化対象とする。

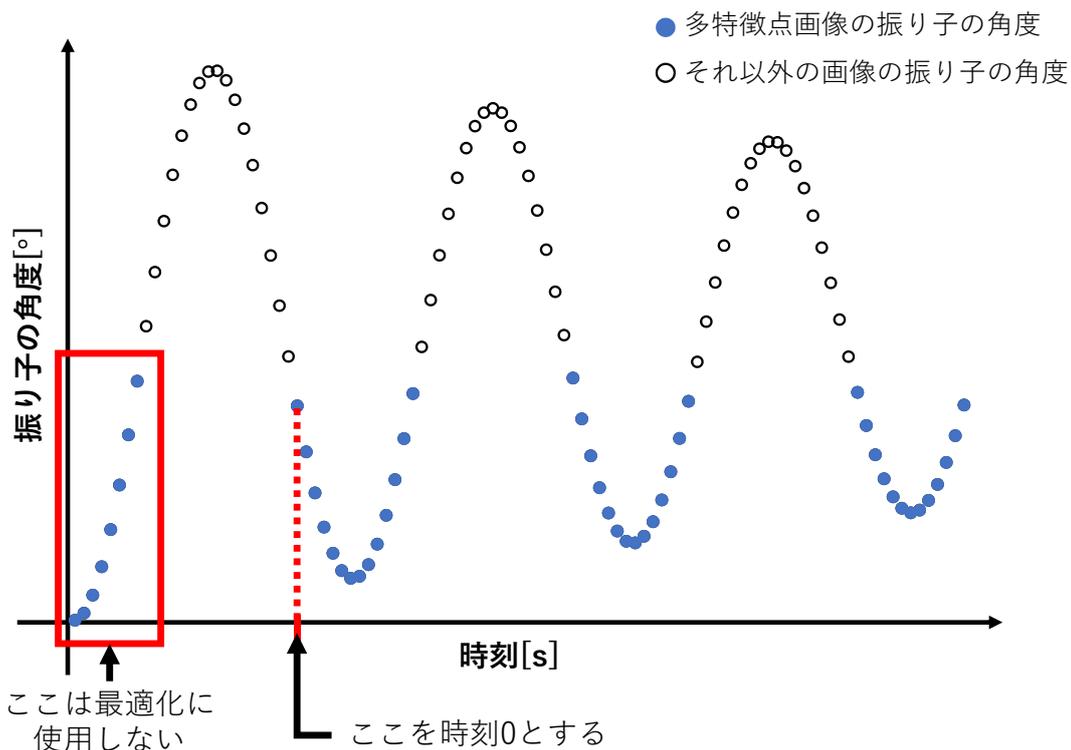


図 3.8: 最適化に使用する振り子の角度

提案手法で使用する運動方程式を式 (3.3) に示す。

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L} \sin(\theta - \theta_{\text{up}}) - \frac{b}{m} \frac{d\theta}{dt} \quad (3.3)$$

ただし、 θ は振り子の角度、 g は重力加速度、 L は振り子の支点からカメラの焦点位置までの長さ (以下、振り子の長さ)、 θ_{up} は図 3.9 に示すように、振り子座標系における1枚目の画像と振動中心との振り子の角度の差 (以下、持ち上げ角度)、 b は抵抗係数、 m は振り子の質量を表す。また、新たに定めた時刻0における振り子の角度と角速度の初期条件をそれぞれ θ_0 、 ω_0 とする。

式 (3.3) の運動方程式を定めるために必要なパラメータは、振り子の長さ L 、抵抗係

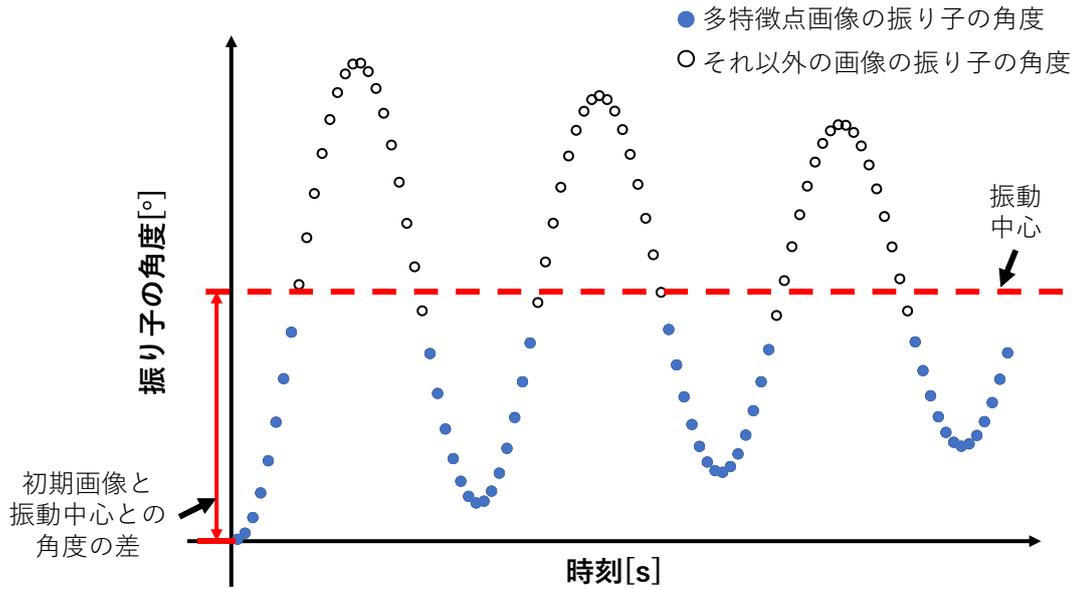


図 3.9: 持ち上げ角度の定義

数 b , 持ち上げ角度 θ_{up} , 重力加速度 g , 振り子の質量 m , 初期角度 θ_0 , 初期角速度 ω_0 の 7 つである. このうち, 重力加速度 g は定数であり, 振り子の質量は測定した値を使用する. また, 初期角度 θ_0 はトラッキング結果から計算した振り子の角度を用いることができるため, 推定が必要なパラメータは全部で 4 つとなる. ただし, 振り子の長さは, 事前にある程度計測可能である. しかし, カメラの焦点位置はカメラ内部であり, 正確な値を測定することはできないため, 最適化によって推定する. 最適化のための各パラメータの初期値について, 振り子の長さは計測で得た支柱の長さを, 持ち上げ角度は運動開始時に振り子を持ち上げた際の, 最下点を基準にした角度を別途計測して初期値とする. 抵抗係数は 0 を初期値とする. 角速度の初期値の決定方法を以下に述べる. 空気抵抗を考慮しない運動を考えた場合, 力学的エネルギー保存の法則より, 運動開始時と振り子が最下点に到達した瞬間について式 (3.4) が成り立つ. ただし, θ_{start} は運動開始時に振り子を持ち上げた際の, 最下点を基準にした角度であり, ω_{under} は最下点における角速度である.

$$mgL(1 - \cos \theta_{start}) = \frac{1}{2}m(L\omega_{under})^2 \quad (3.4)$$

式 (3.4) から最下点における角速度 ω_{under} は式 (3.5) と書けることがわかる.

$$\omega_{under} = \sqrt{\frac{2g(1 - \cos \theta_{start})}{L}} \quad (3.5)$$

提案手法では、振り子運動中に外部から力を加えないため、最下点における角速度の絶対値は ω_{under} を超えることはない。そのため、角速度の初期値は最初に持ち上げる方向を考慮して正負を決定した $\pm|\omega_{\text{under}}|$ のいずれかとする。

ステップ2-2の一連の流れを図3.10に示す。このステップでは、ステップ2-1で計算した振り子の角度と、運動方程式を数値計算して得られた振り子の角度の2乗誤差の和(以下、エラー値と呼ぶ。)が最小になるようにパラメータを最適化する。ただし、エラー値の計算に用いる振り子の角度は度数法で表すものとする。以下に、ステップ2-2の処理の詳細について述べる。まず、各パラメータの初期値を用いてエラー値を計算する。以降、この値を基準エラー値とする。次に、図3.10に示すように、振り子の長さ、初期角速度、抵抗係数、持ち上げ角度の順に、各パラメータを1つずつ最適化する。そして、一通りパラメータの最適化を行った後、再度エラー値(終了時エラー値)を計算する。以下のいずれかの条件を満たした場合にはこのステップを終了する。

- 終了時エラー値が閾値以下
- 基準エラー値と終了時エラー値の差が閾値以下
- 繰り返し回数が閾値以上

これらの条件を満たさない場合、終了時エラー値を次の基準エラー値として処理を繰り返す。なお、最適化するパラメータの順番については、第4.2節で後述するシミュレーションにより決定した。

次に、エラー値を求めるために必要な、運動方程式を数値計算して画像の撮影時刻における振り子の角度を求める手法について述べる。処理の流れを図3.11に示す。運動方程式の数値計算には4次のルンゲ・クッタ法を用いるが、この運動方程式の数値計算では、任意の時刻における振り子の角度を直接計算することはできない。そのため、まず、0.1ms毎に振り子の角度を計算し、時刻と振り子の角度を紐付けたデータを作成する。次に、1枚目の画像から順に、撮影時刻における振り子の角度を計算する。このとき、撮影時刻と同じ時刻が最初に計算したデータに存在する場合は、その時刻に対応する振り子の角度をその画像の振り子の角度とする。存在しない場合は、撮影時刻の直前と直後のデータを用い、線形補完によって計算された値をその画像の振り子の角度とする。この処理を最後の画像まで繰り返すことで、運動方程式を用いて計算した振り子の角度を取得する。

次に、各パラメータごとの最適化の詳細について述べる。ここでは具体例として振り子の長さ L の最適化について述べる。振り子の長さ以外のパラメータも同様の方法

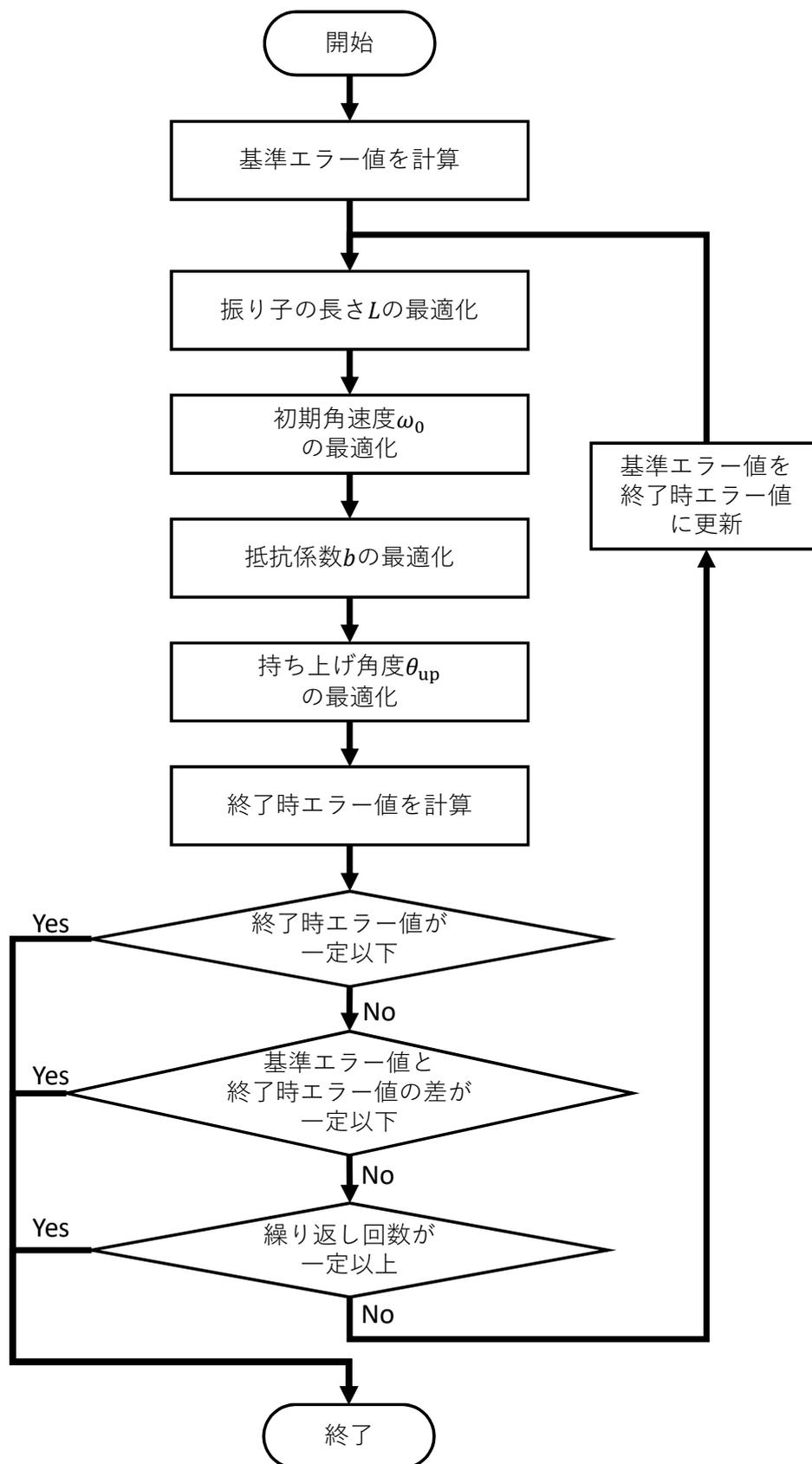


図 3.10: ステップ 2-2 の処理の流れ

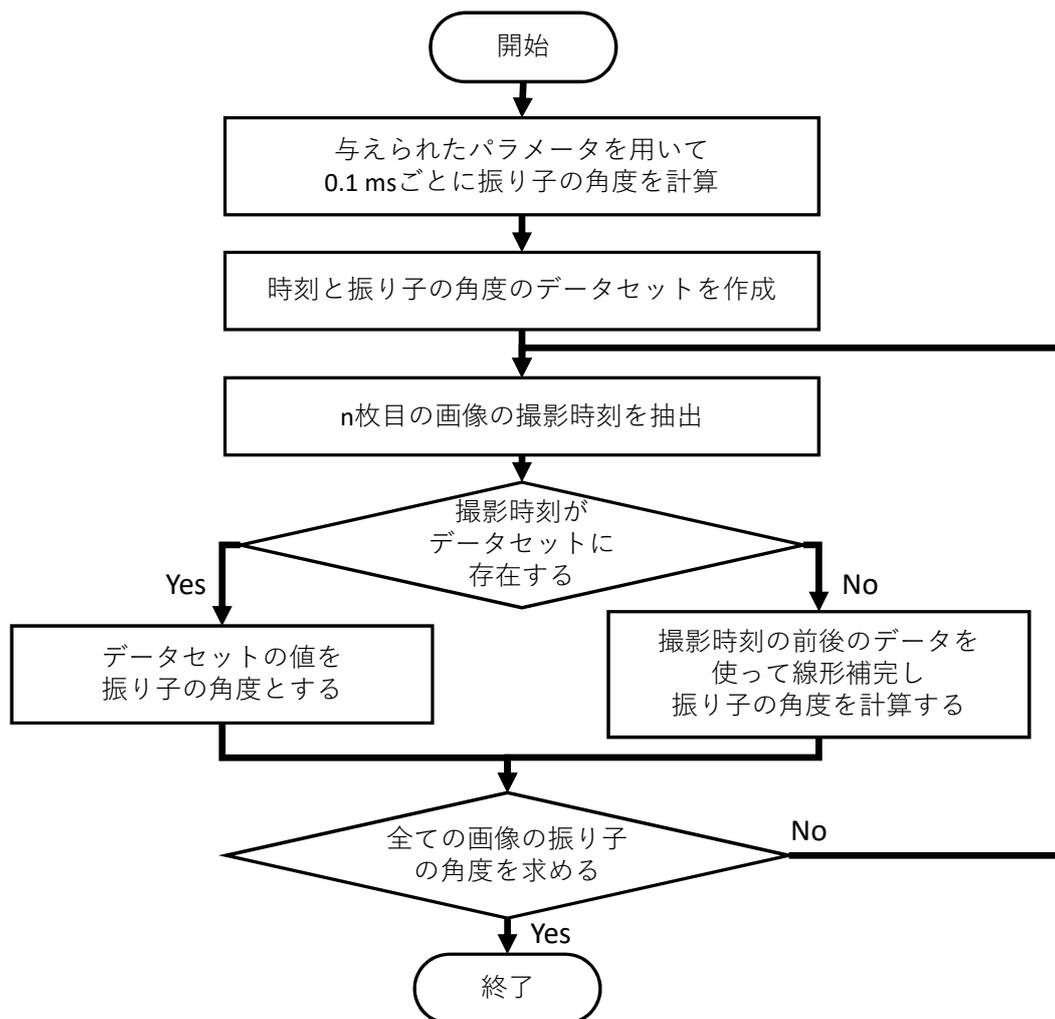


図 3.11: 数値計算で画像の撮影時刻における振り子の角度を求める処理の流れ

で最適化する。処理の流れを図3.12に示す。最適化には山登り法を用いる。まず、最適化する4つのパラメータの初期値を用いてエラー値の計算を行い、これを基準エラー値 E_{base} とする。次に、振り子の長さ L を正と負の両方向に δ だけ変化させた $L_+(=L+\delta)$ と $L_-(=L-\delta)$ を用い、それぞれのエラー値 E_+ と E_- を計算する。そして、 E_+ と E_- の小さいものを E_{min} とし、 E_{min} と E_{base} を比較する。 E_{min} が E_{base} より小さい場合、 E_{base} の値を E_{min} に、 L を E_{min} を求めるために使用した δ だけ変化させたものに更新する。そうでない場合、 δ を $1/10$ にする。この操作を繰り返し、 E_{min} の値、 E_{base} と E_{min} の差、 δ の値のいずれかが予め定めた閾値よりも低くなるか、予め定めた繰り返し回数に達した場合に一連の処理を終了する。 δ の値、終了条件はパラメータ毎に適切な値を設定する。

3.4.3 運動方程式を用いた全画像のカメラ姿勢の推定

ステップ2-3では、ステップ2-2で最適化したパラメータを用いて全ての画像のカメラ姿勢を求める。まず、最適化したパラメータを用いて、ステップ2-2と同様の方法で運動方程式の数値計算を行い、全ての画像の撮影時刻における振り子の角度を計算する。次に、計算した振り子の角度を用いて世界座標系における回転行列と位置を表す3次元ベクトルを求める。 n 枚目の画像の振り子の角度を θ_n 、振り子の長さを L とするとき、回転行列 R_n と3次元ベクトル \mathbf{x}_n はそれぞれ式(3.6)、式(3.7)と表される。

$$R_n = \begin{bmatrix} \cos \theta_n & \sin(-\theta_n) & 0 \\ \sin \theta_n & \cos \theta_n & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

$$\mathbf{x}_n = \begin{bmatrix} -L \sin \theta_n \\ L(\cos \theta_n - 1) \\ 0 \end{bmatrix} \quad (3.7)$$

最後に、これらの結果を組み合わせ、トラッキングによって得られたカメラ姿勢と同様に 4×4 の同次行列としてカメラ姿勢を求める。

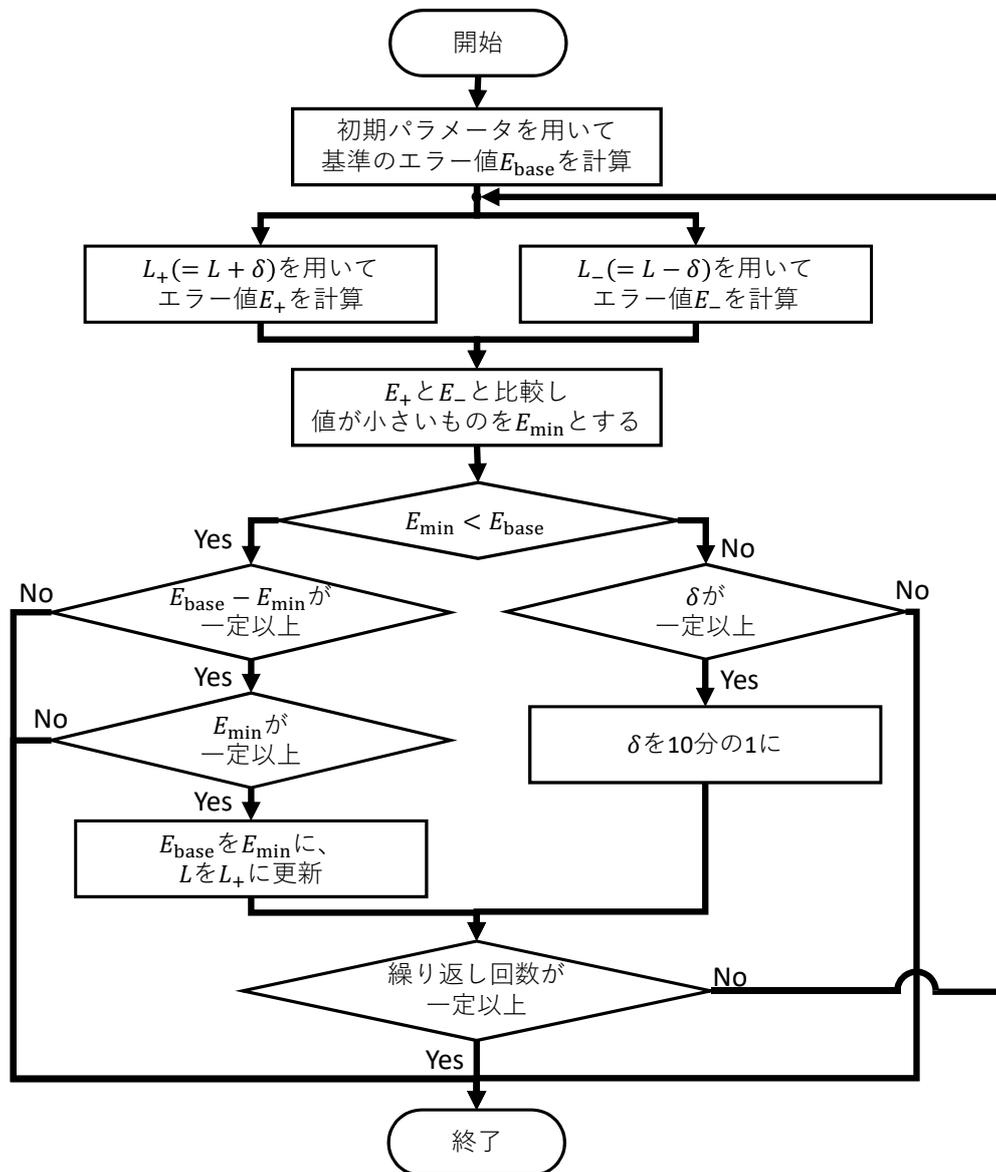


図 3.12: 振り子の長さ L の最適化の流れ

3.5 3次元再構成の実行

ステップ3では、ステップ2で求めたカメラ姿勢、環境を撮影したカラー画像およびデプス画像を用いて3次元再構成を行い、3次元モデルを取得する。以下で3次元再構成の処理の流れを説明する。

まず、1枚のデプス画像の1つのピクセル(画像を構成する画素)について、画像の中心を原点とする2次元座標 $\mathbf{i}_c = (x, y)^T$ [pixel]、カメラの焦点距離 f [pixel]、そのピクセルのデプス値 d_{i_c} [mm] を用いてカメラ座標系における3次元座標を求める。デプス値が0の場合は、デプスの値が取得できていないため、そのピクセルをスキップし、次のピクセルの処理に移る。 \mathbf{i}_c と f 、 d_{i_c} を用いて、カメラ座標系における3次元座標 $(X_c, Y_c, Z_c)^T$ を求める式を式(3.8), (3.9), (3.10)に示す。

$$X_c = x \frac{d_{i_c}}{f} \quad (3.8)$$

$$Y_c = y \frac{d_{i_c}}{f} \quad (3.9)$$

$$Z_c = d_{i_c} \quad (3.10)$$

次に、求めたカメラ座標系における選択したピクセルの3次元座標と、世界座標系におけるカメラ姿勢を用いてカメラ座標系における3次元座標を世界座標系に座標変換する。そして、デプス画像に対応するカラー画像のピクセルからカラー情報を取得し、再構成時のカラー情報とする。これをすべての画像のすべてのピクセルについて行うことで、3次元モデルを得る。

第 4 章 提案手法の実装

本章では，第 3 章で述べた提案手法の実装について述べる．まず，ソフトウェアの実装について述べる．次に，第 3.4.2 項で述べた，パラメータを最適化する際の順番を決定するために実施したシミュレーションについて述べる．その後，実装したソフトウェアの動作確認を Unity を用いて行った結果について述べる．最後に，ハードウェアの実装について述べる．

4.1 ソフトウェアの実装

4.1.1 画像の撮影と保存を行うソフトウェアの実装

本項では，振り子カメラを用いて環境を撮影し，取得した画像を保存するためのソフトウェアの実装について述べる．ソフトウェアの処理の流れを図 4.1 に示す．環境を撮影した画像は，カメラを制御する小型 PC に一時的に保存し，撮影終了後に提案手法を実行するメイン PC に転送する方法を採用した．これにより，小型 PC とメイン PC の間で常に通信を行い，画像の取得と送信を 1 枚ずつ繰り返し行う手法に比べ，送信失敗などによる画像枚数の減少を防ぐことが可能である．また，小型 PC に保存する際

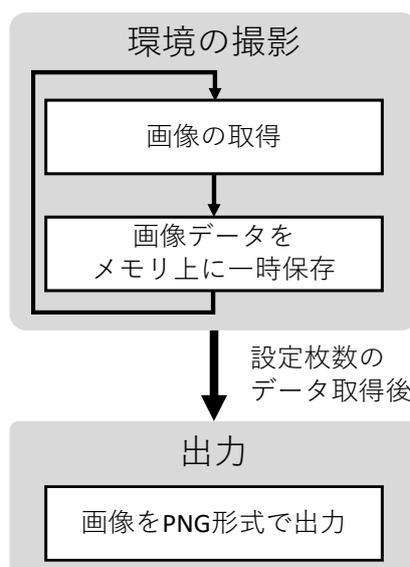


図 4.1: 画像の撮影と保存を行うソフトウェアの処理の流れ

にも、まず、画像データをメモリ上に保存し、全ての画像データを取得した後に記憶デバイスに書き出す。これは、画像データを1枚ずつ記憶デバイスに書き出すと、その処理によってフレームレートが低下したり、画像の撮影間隔が一定にならない可能性があり、そのような問題が起こらないようにするためである。ソフトウェアは、事前に指定した枚数の画像データを取得後に撮影を終了し、PNG形式で画像を保存する。

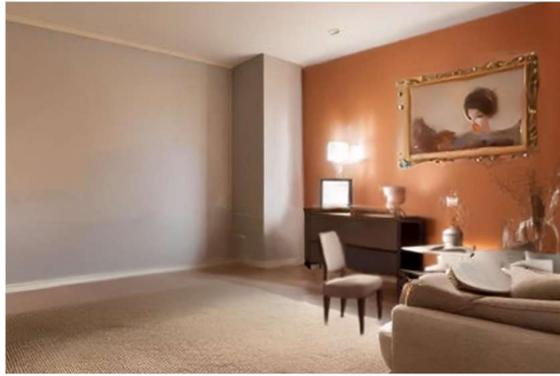
このソフトウェアは、開発環境としてMicrosoft社のVisualStudio 2017、ライブラリとしてRealSense SDK 2.0(Ver.2.50.0)とOpenCV(Ver.4.5.5)を用いてC++開発した。

4.1.2 画像の処理と3次元再構成を実行するソフトウェアの実装

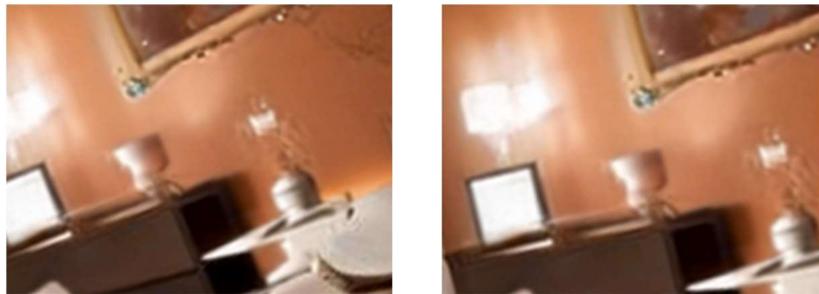
本項では、取得した画像に第3章で述べた提案手法を適用し、3次元再構成を実行するためのソフトウェアの実装について述べる。

本研究では、提案手法のステップ1-2である多特徴点画像のトラッキングに、COLMAP(Ver.3.7)^[12,13]を用いた。COLMAPは特徴点の認識と特徴量の計算にSIFTを使用しており、画像の回転やスケールの変化に対して頑健である。また、カメラ姿勢や特徴点の3次元座標の推定誤差を減らすBundle Adjustmentを行うため、誤差の少ないトラッキング結果を得ることができる。以上の特徴を踏まえ、本研究では多特徴点画像のトラッキングにCOLMAPを採用した。COLMAPでは、まず入力された全てのカラー画像の特徴点を抽出する。次に各特徴点の特徴量を計算し、それを基に画像間で特徴点のマッチング(以下、特徴点マッチング)を行う。そして、特徴点マッチングの結果を用いてカメラ姿勢の推定と3次元再構成を行う。本研究では、COLMAPのデフォルトの設定に加え、特徴点の抽出の際にestimate affine shapeとdomain size poolingのオプションを、特徴点マッチングの際にはguided matchingのオプションをそれぞれ有効にする。これにより、処理に要する時間は増加するが、結果として得られる画像間の特徴点マッチングの数を増やすことができ、より正確なトラッキングが可能になる。COLMAPを実行する際には、初期化(トラッキングの起点となる環境内の3次元点の座標の推定)に使用する2枚の画像を決定する必要がある。この際、画像間の差分が小さすぎる場合には初期化に失敗し、それ以降の処理が実行できない。振り子カメラは同じ環境を何度も撮影するため、図4.2のように、多特徴点画像の中には画像間の差分が非常に小さい組が存在する。そのような画像が選択されることを防ぐため、最初に入力する画像は手動で選択する。

このソフトウェアは、第4.1.1項と同様に、開発環境としてMicrosoft社のVisualStudio 2017、ライブラリとして画像の読み込みと特徴点の認識、特徴量の計算に



撮影環境の例



画像間の差分が少ない画像の例

図 4.2: 画像間の差分が少ない画像

OpenCV(Ver.4.5.5) を、行列の計算に Eigen(Ver.3.4.0) を用いて C++ 言語で開発した。

4.2 シミュレーションを用いたパラメータの最適化順の決定

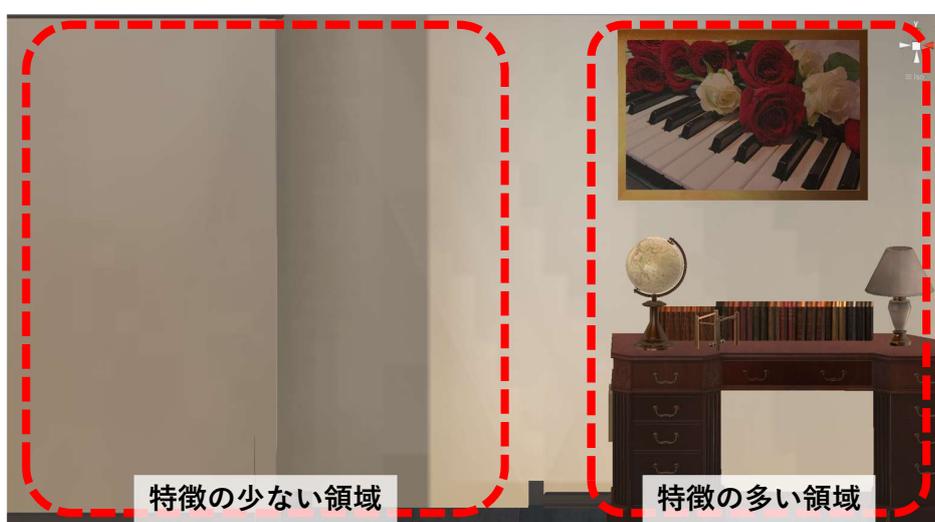
第 3.4.2 項で説明したパラメータの最適化において、最適化の対象となる 4 つのパラメータである振り子の長さ、抵抗係数、持ち上げ角度、初期角速度の推定誤差は、エラー値に与える影響が異なると考えられる。そのため、局所最小値(ローカルミニマム)に陥ることなく、より速く最適化を行うためには各パラメータの最適化順を適切に設定する必要がある。そこで本節では、Unity を用いて理想的な仮想空間でシミュレーションを行い、各パラメータがエラー値に与える影響を検証し、その結果を基にパラメータの最適化順を決定する。Unity を用いるシミュレーションでは、振り子運動を模擬した際にカメラが取得するカラー画像とデプス画像だけでなく、各画像を取得した瞬間のカメラ姿勢の真値を取得できる。そのため本節では、第 3.4.2 項で述べた多特徴点画像のカメラ姿勢に、COLMAP を用いたトラッキング結果ではなく、シミュレーション

で生成された真値を用いる。これにより、より正確な検証を行うことが可能になる。

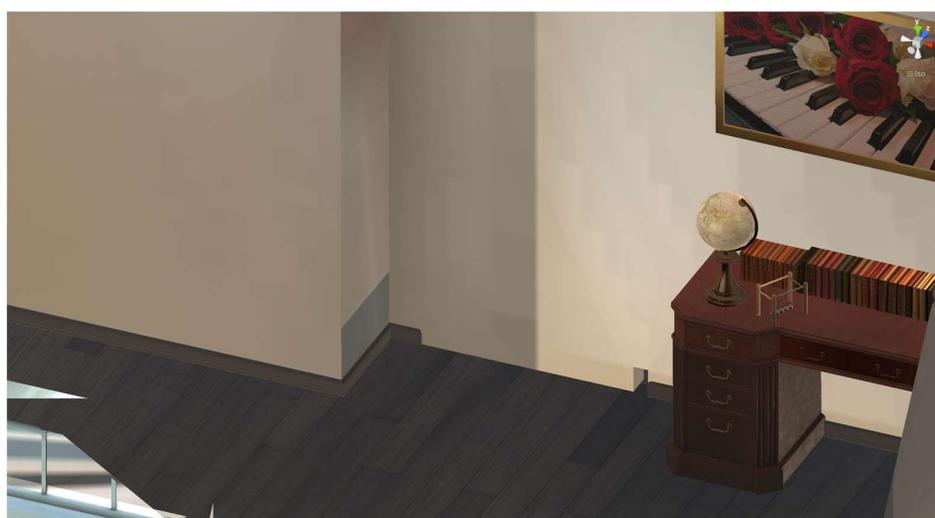
4.2.1 使用したシミュレーション環境と画像の撮影条件

シミュレーションに使用した3次元モデルの外観を図4.3に示す。このモデルはUnity (Ver.2020.2.6f1) を用い、HQ Suburban House^[31] を基に作成した。作成したモデルでは、図4.3に示すように、右側を机や絵画などのオブジェクトを配置した特徴の多い領域、中央から左側にかけてをオブジェクトを配置しない特徴の少ない領域とした。

次にUnity内で作成した振り子カメラについて述べる。カメラの各パラメータの設



正面から見た図



斜め上から見た図

図 4.3: シミュレーションで使用した3次元モデルの外観

定を表 4.1 に示す。振り子運動は、運動方程式を用いて再現する。Unity で使用可能な物理エンジンを用いて振り子の運動を再現したところ、運動開始からの経過時間と振り子の角度の関係が現実の運動法則に従っていない部分があり、シミュレーションとして適切ではなかった。そのため、本研究では運動方程式を数値計算することで振り子運動を再現した。使用した運動方程式は式 (3.3) と同様である。運動方程式のパラメータはそれぞれ、 $L = 1.20$ m, $g = 9.80$ m/s², $b = 5.00 \times 10^{-2}$ kg/s, $m = 1.00$ kg とした。また、振り子運動は、特徴の多い領域が運動開始時に写るように 45.0° 持ち上げて開始し、運動開始時の角速度 ω_{start} は 0.00 rad/s とした。このとき、振動の中心は最下点であり、振り子を 45.0° 持ち上げて運動を開始するため、持ち上げ角度 θ_{up} は 45.0° となる。また、1 往復目の前半を除いた後の振り子の角速度の初期条件 ω_0 (時刻 0 での角速度) は、時刻 0 の振り子の角度の真値 $\theta_{0\text{true}}$ と、前後の画像を取得した時の振り子の角度の真値 $\theta_{-1\text{true}}$ および $\theta_{1\text{true}}$ 、各画像を取得した時間差 Δt_{-1} および Δt_1 を用いて、以下の式 (4.1) で求めたものを使用する。

$$\omega_0 = \frac{\frac{\theta_{0\text{true}} - \theta_{-1\text{true}}}{\Delta t_{-1}} + \frac{\theta_{1\text{true}} - \theta_{0\text{true}}}{\Delta t_1}}{2} \quad (4.1)$$

表 4.1: シミュレーションで使用する RGB-D カメラの仕様

項目	仕様
Forcal Length	385 pixel
Sensor Size	640×480 pixel
Clipping Planes(Near)	0.1 m
Clipping Planes(Far)	10 m

続いて、Unity におけるカメラ姿勢の計算方法について述べる。カメラ姿勢は、運動方程式を数値計算して得られた角度 θ_t を用いて計算した。運動方程式の数値計算には第 3.4.2 項と同様に 4 次のルンゲ・クッタ法を用いた。まず、Unity での世界座標系におけるカメラ姿勢を計算する。カメラの位置 $(x, y, z)^T$ は、 θ_t 、支点の座標 $(x_f, y_f, z_f)^T$ 、振り子の長さ L を用いて、式 (4.2) に従って求める。

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_f + L \sin \theta_t \\ y_f - L \cos \theta_t \\ z_f \end{bmatrix} \quad (4.2)$$

また、カメラの方向は localEulerAngles 関数を用いて $(0, 0, \theta_t)$ と表す。このカメラの位置と方向をもとに localToWorldMatrix 関数を用いて 4×4 の同次行列を求める。次に、計算したカメラ姿勢を右手系に変換する。これは、Unity は左手系を用いて座標や回転を表現している一方、提案手法では右手系で座標系を定義しており、Unity で得られた結果をそのまま利用できないためである。最後に、実世界の世界座標系に変換するため、1 枚目の画像のカメラ姿勢を表す同次行列の逆行列を全てのカメラ姿勢を表す同次行列に掛ける。

4.2.2 シミュレーション方法

本項では、Unity で得られた画像とカメラ姿勢の真値を用いて、各パラメータを変化させた際にエラー値がどのように変化するかを調べるための、シミュレーションの方法について述べる。まず、提案手法と同様に、第 3.3.1 項で述べたステップ 1-1 を実行し、多特徴点画像を抽出する。次に、抽出した多特徴点画像のカメラ姿勢の真値から振り子の角度を計算する。そして、最適化を行う 4 つのパラメータのうち、3 つのパラメータを真値で固定し、残った 1 つのパラメータを変化させてエラー値を計算する。さらに、4 つのパラメータのうち 2 つを固定し、残りの 2 つのパラメータを変化させた際に、エラー値がどのように変化するかも調べる。これは、複数のパラメータを変化させた際にローカルミニマムに陥る可能性がどの程度あるかを調べるために行う。以上のシミュレーションによって得られたエラー値の結果から、パラメータを最適化する順番を決定する。

式 (4.1) を用いて振り子の角速度の初期条件 ω_0 を計算した結果、真値は -2.04 となった。各パラメータを変化させる範囲を表 4.2 に示す。振り子の長さは、実際に計測した長さから 0.3 m 以上離れることはないと考えられるため、真値に対して ± 0.3 m を範囲とした。持ち上げ角度についても、計測する際に 10° 以上ずれることは無いと考えられるため、真値に対して $\pm 10^\circ$ を範囲とした。抵抗係数については、減衰が必ず起こるため、正の値とするべきであるが、適切な範囲を理論的に決定することは難しいため、十分大きな正の値として、 0.1 を範囲の最大とした。初期角速度は、第 3.3.2 項の式 (3.5) を基に、時刻 0 における角速度は負であるので、 -2.19 から 0 までを範囲とした。各範囲をそれぞれ 100 分割し、変化を確認した。

表 4.2: 運動方程式のパラメータの変化範囲

パラメータ	真値	変化させる範囲	変化量
振り子の長さ (L)[m]	1.20	0.90 – 1.50	6.00×10^{-3}
抵抗係数 (b)[kg/s]	5.00×10^{-2}	0.00 – 0.100	1.00×10^{-3}
持ち上げ角度 (θ_{up})[°]	45.0	35.0 – 55.0	2.00×10^{-1}
初期角速度 (ω_0)[rad/s]	-2.04	-2.19 – 0.00	2.19×10^{-2}

4.2.3 パラメータの最適化順の決定

Unity を用いたシミュレーション結果について述べる。1つのパラメータを変化させた結果を図 4.4 から図 4.7 に示す。各図の赤い線は真値を表す。図 4.4 は振り子の長さ、図 4.5 は抵抗係数、図 4.6 は持ち上げ角度、図 4.7 は初期角速度をそれぞれ変化させた結果を示している。また、2つのパラメータを変化させた結果を図 4.8 から図 4.13 に示す。図 4.8 から図 4.13 のそれぞれについて、上段はエラー値の変化を立体的に表現したグラフを、下段には各値のエラー値の大小を色で表現したものを示す。図 4.8 は振り子の長さ と 抵抗係数、図 4.9 は振り子の長さ と 持ち上げ角度、図 4.10 は振り子の長さ と 初期角速度、図 4.11 は抵抗係数 と 持ち上げ角度、図 4.12 は抵抗係数 と 初期角速度、図 4.13 は持ち上げ角度 と 初期角速度をそれぞれ変化させた結果を示している。

まず、図 4.4 から図 4.7 を比較すると、エラー値の変化の幅が最も大きいのは振り子の長さであった。また、振り子の長さ と 他の 1つのパラメータを変化させた図 4.8 から図 4.10 の結果から、振り子の長さが真値から離れている場合、他のパラメータを変化させてもエラー値が小さくならないことがわかる。さらに、振り子の長さは他のパラメータの値によらず、真値付近で急激にエラー値が減少することも確認できる。これらの結果から、まず振り子の長さを最適化し、真値に近い値にすることが必要であると考えられる。

次に、持ち上げ角度に注目する。図 4.6 の結果から、持ち上げ角度のみを変化させた場合、他のパラメータに比べてエラー値の変化の幅が小さいことがわかる。また、図 4.11 と図 4.13 を見ると、図 4.13 の初期角速度が真値よりも大きい場合を除いて、他のパラメータの値によらず、持ち上げ角度を変化させた際のエラー値の変化の幅も小さい。以上の結果から、持ち上げ角度がエラー値に与える影響は他のパラメータに比べて小さいと考えられ、優先して最適化する必要は無いと考えられる。

残りの抵抗係数と初期角速度について、図 4.5 と図 4.7, 図 4.11, 図 4.13 から、ローカルミニマムに陥ることなくエラー値が変化していることがわかるため、両者の順番が最適化結果に与える影響の差は小さいと考えられる。一方で、図 4.12 より、初期角速度が真値から離れている場合、抵抗係数の値を変化させてもエラー値があまり小さくならないことがわかる。そのため、より速く最適な値に収束させるために、本研究では初期角速度を先に最適化する。

以上より、パラメータの最適化は、振り子の長さ、初期角速度、抵抗係数、持ち上げ角度の順に行うこととする。

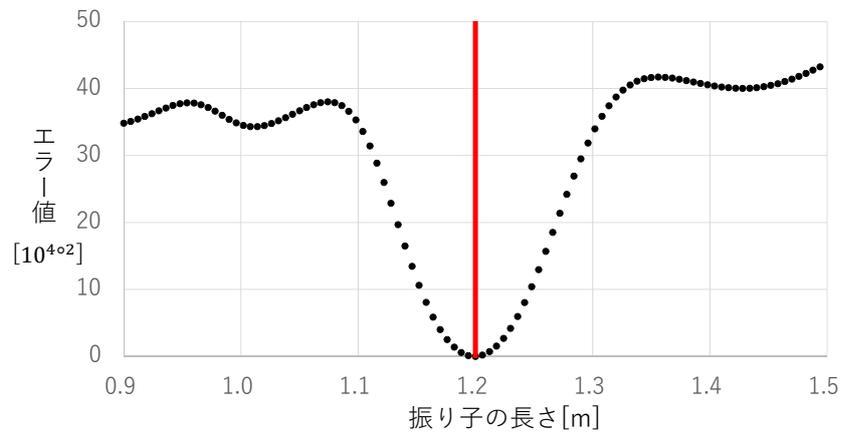


図 4.4: 振り子の長さの真値からのずれとエラー値の関係

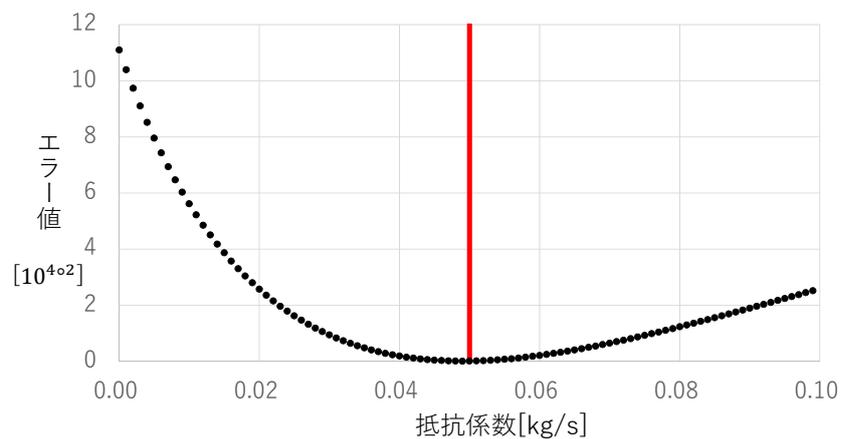


図 4.5: 抵抗係数の真値からのずれとエラー値の関係

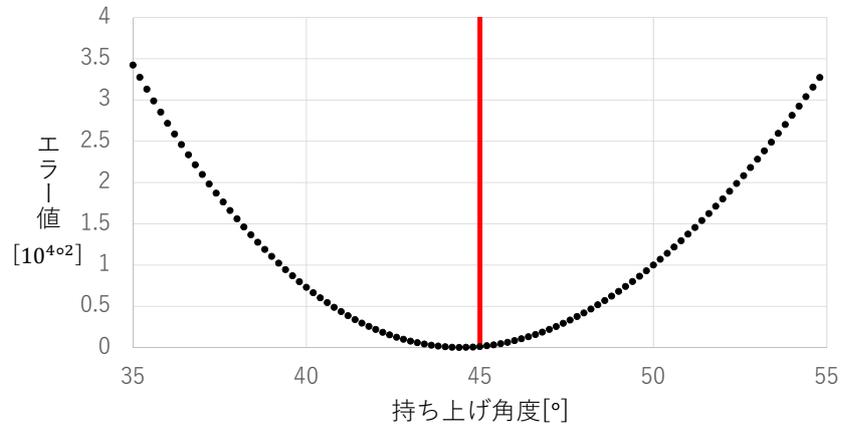


図 4.6: 持ち上げ角度の真値からのずれとエラー値の関係

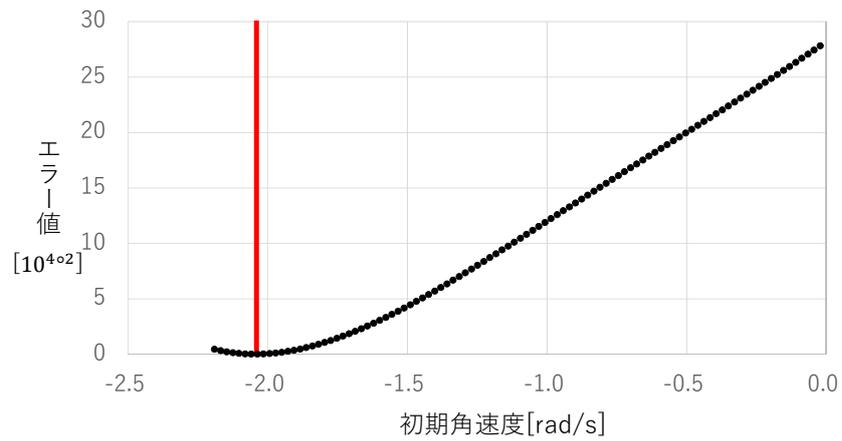


図 4.7: 初期角速度の真値からのずれとエラー値の関係

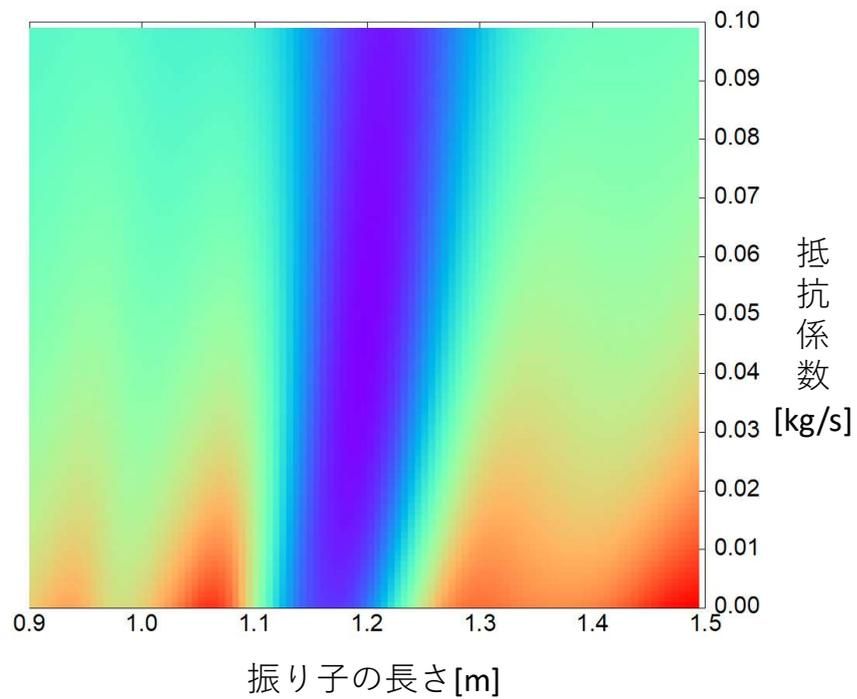
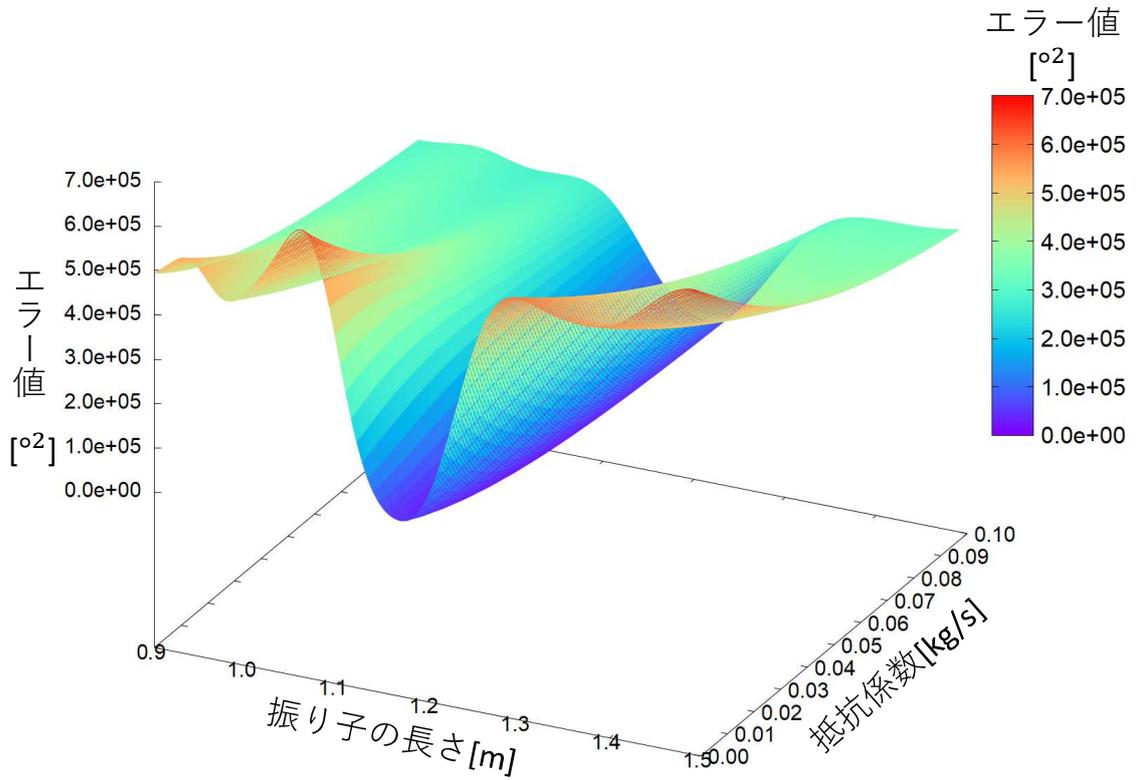


図 4.8: 振り子の長さ・抵抗係数とエラー値の関係

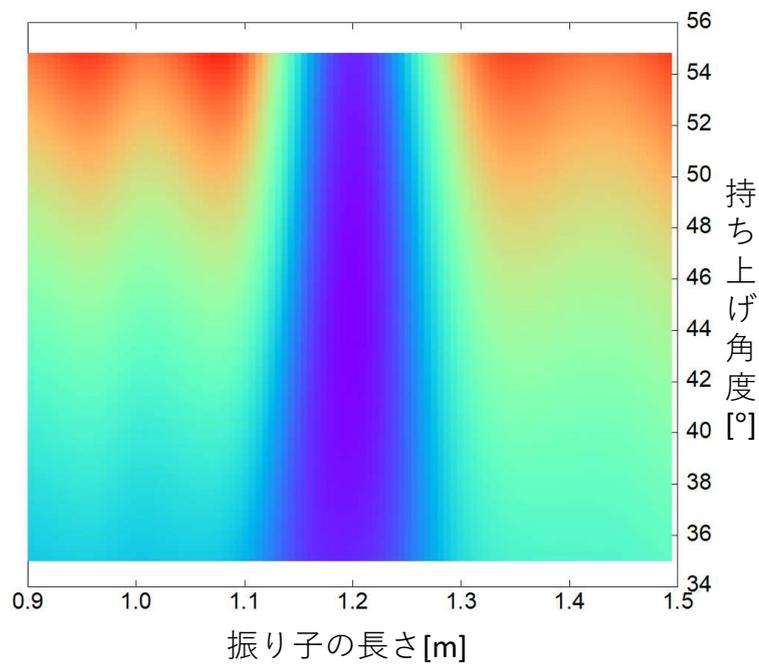
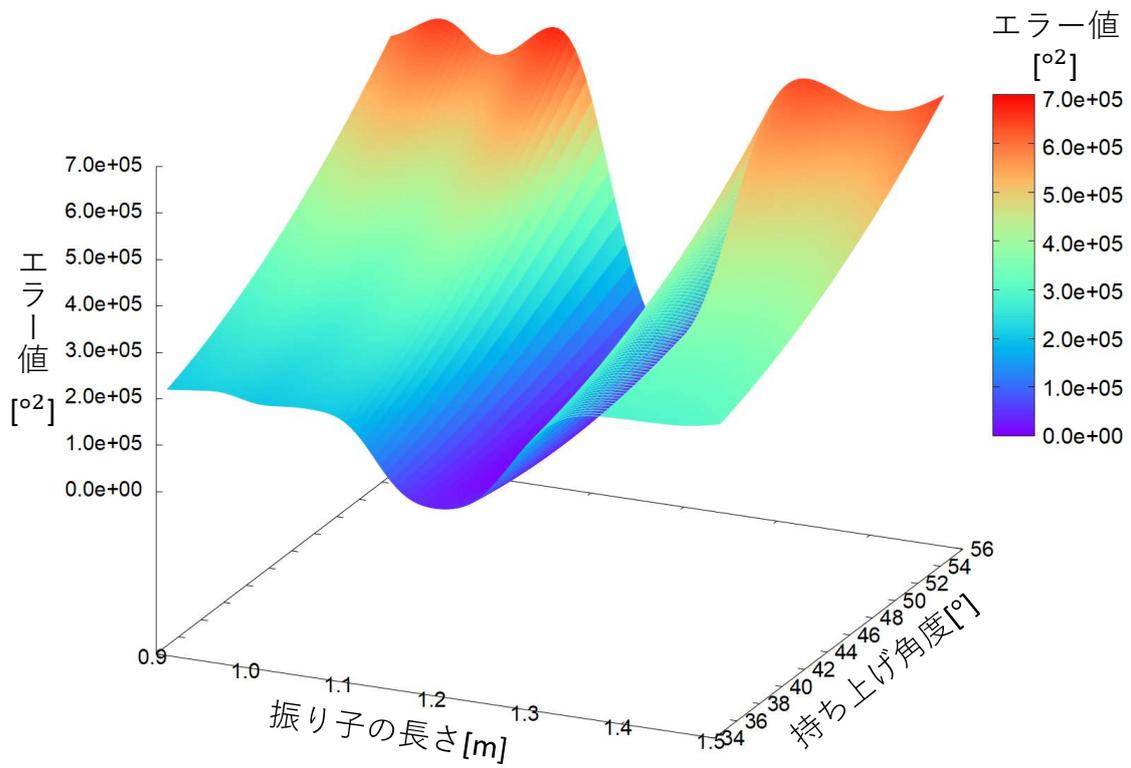


図 4.9: 振り子の長さ・持ち上げ角度とエラー値の関係

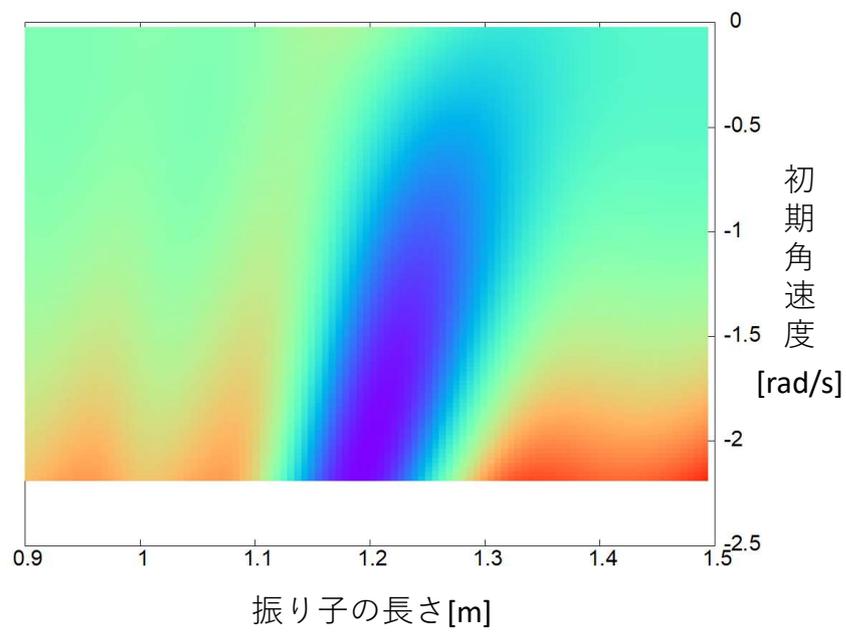
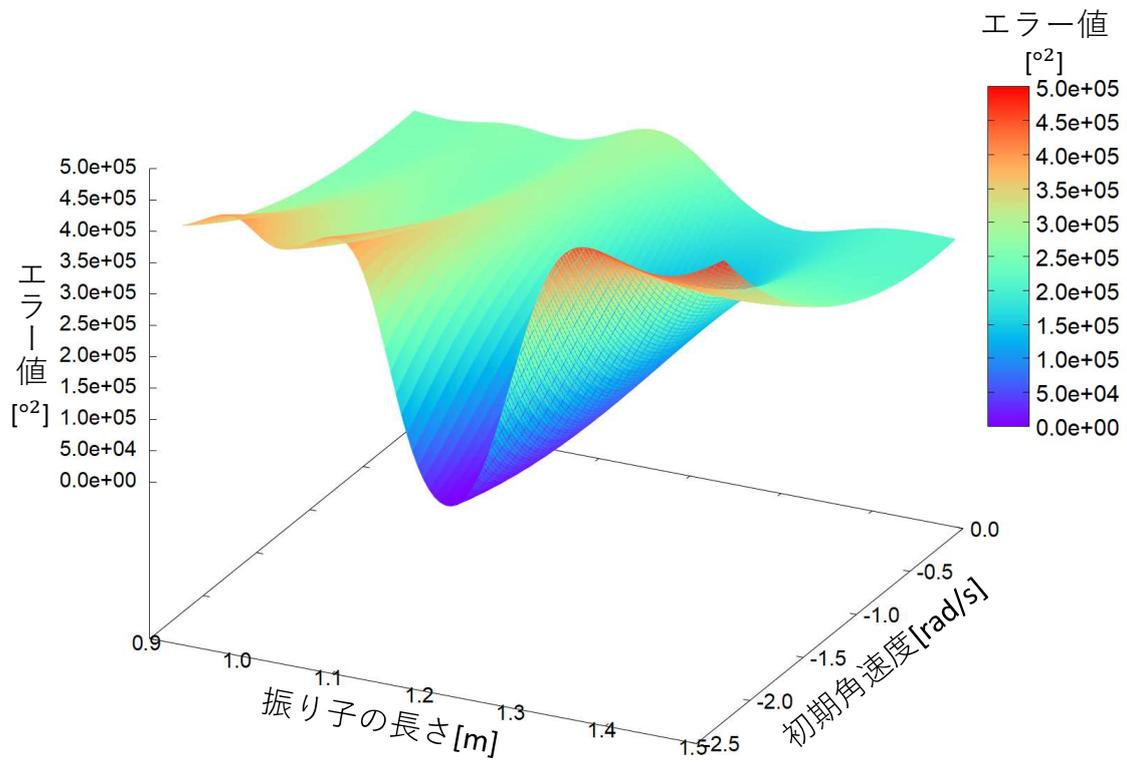


図 4.10: 振り子の長さ・初期角速度とエラー値の関係

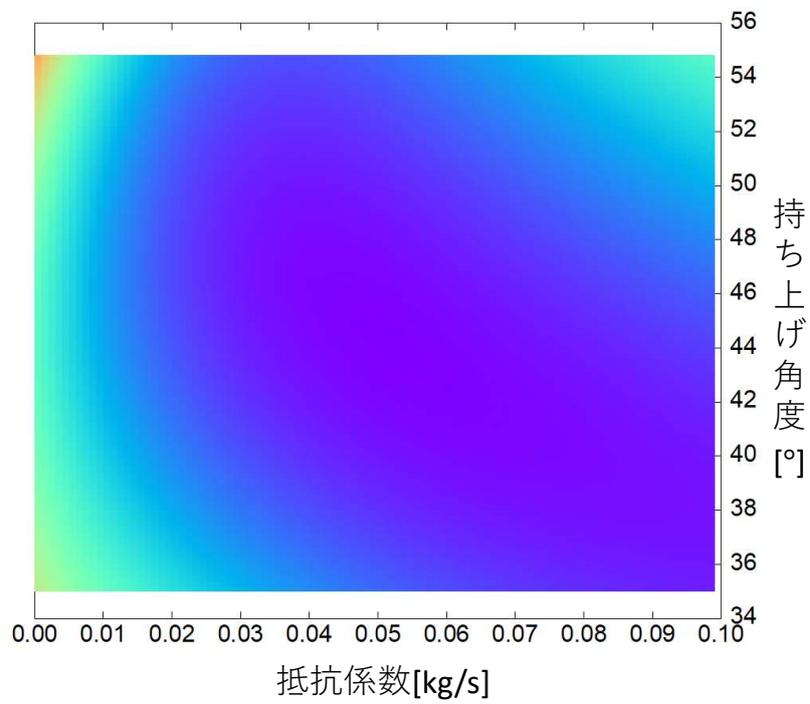
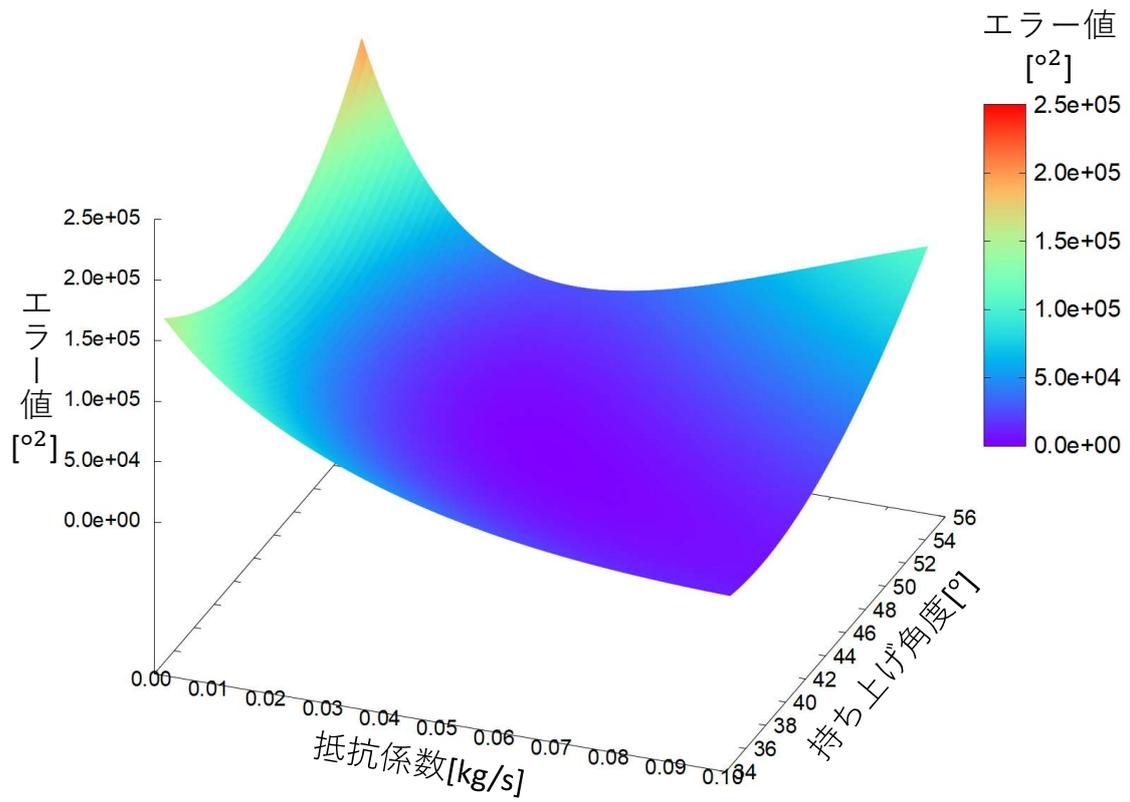


図 4.11: 抵抗係数・持ち上げ角度とエラー値の関係

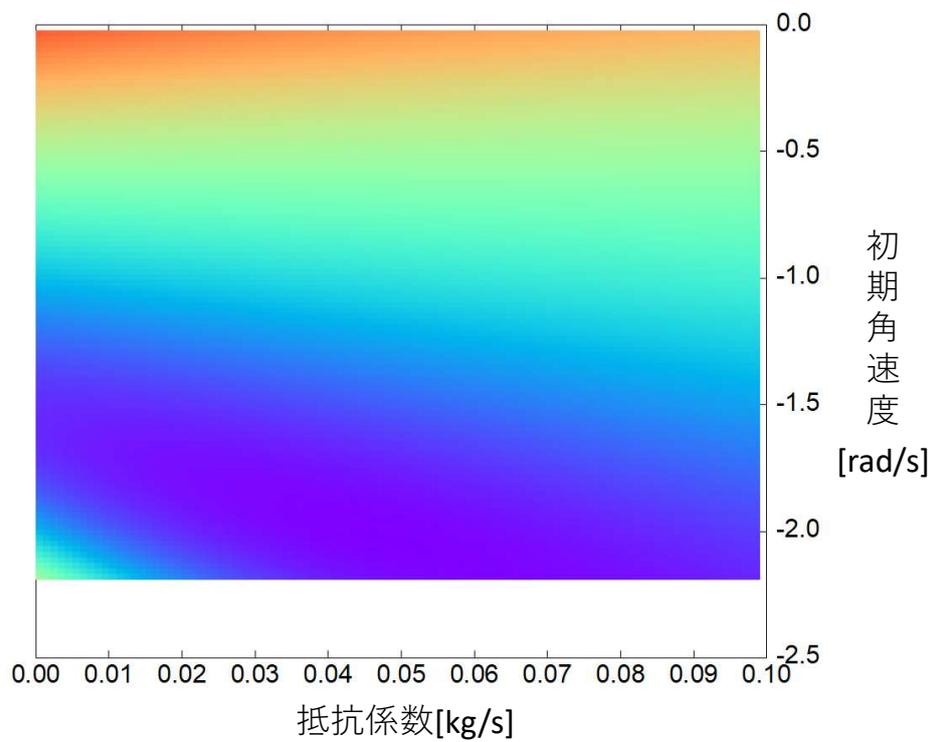
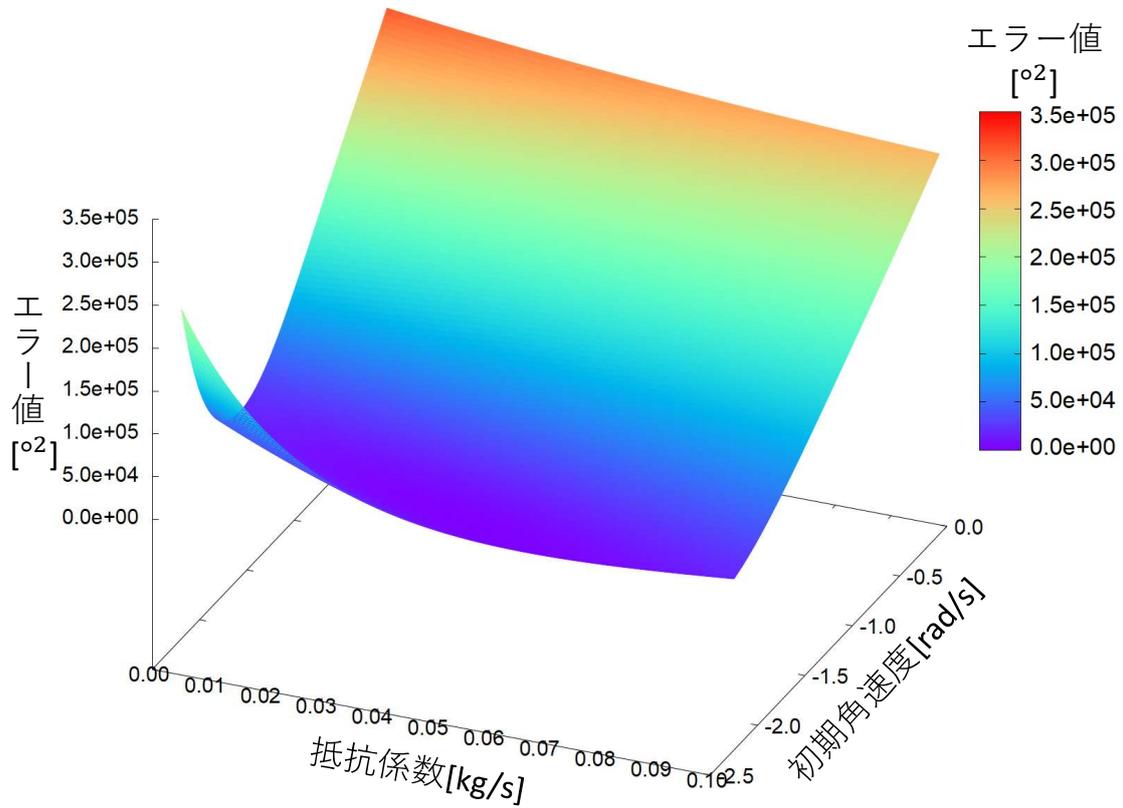


図 4.12: 抵抗係数・初期角速度とエラー値の関係

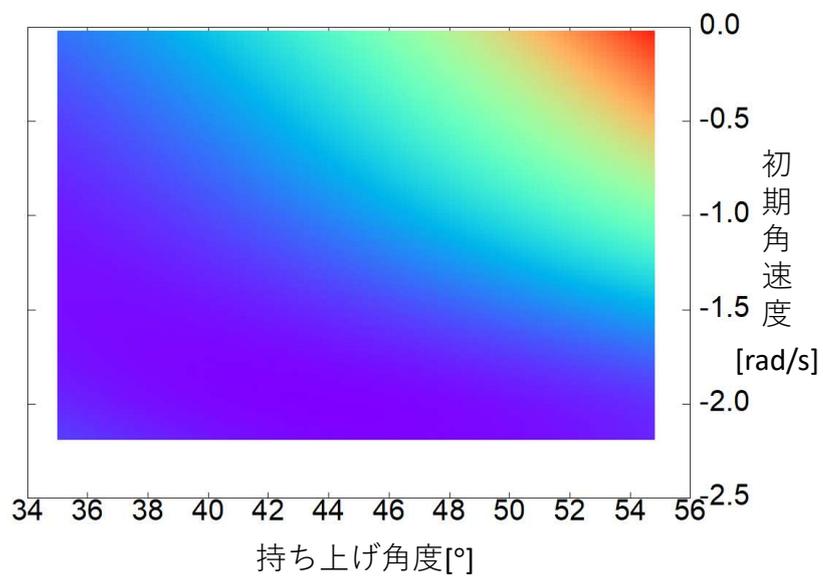
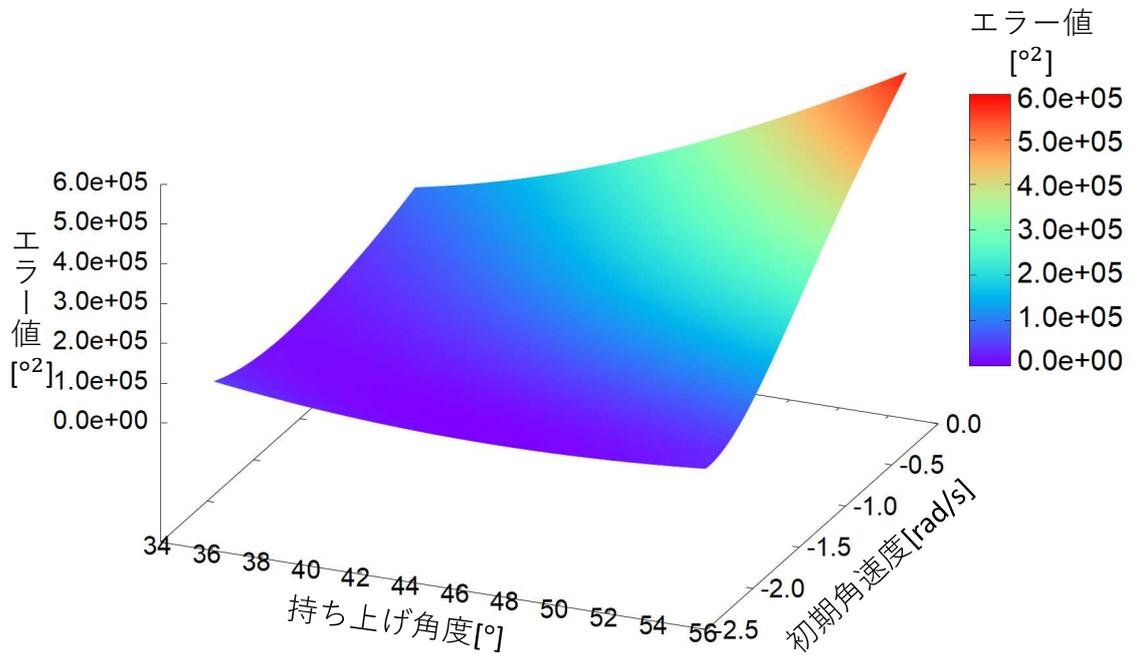


図 4.13: 持ち上げ角度・初期角速度とエラー値の関係

4.3 シミュレーションによる実装ソフトウェアの動作検証

本節ではシミュレーションにより振り子運動を模擬して生成したカメラ画像を用いて、実装ソフトウェアが正しく動作し、特徴の少ない領域でも抜け落ちのない3次元再構成が実現できるか検証する。ただし、本検証はあくまで理想的な環境において提案手法が動作するかを検証するものである。これは、Unityで作成した環境では、振り子カメラは仮定した運動方程式に従った完全に平面上の運動を行い、デプス画像にはノイズが含まれないなど、現実環境と異なるためである。Unityで作成した環境と環境の撮影方法は第4.2節と同様である。

本検証ではUnityで作成した仮想環境で振り子カメラによりシミュレーションを行うことで、1,000組のカラー画像とデプス画像を取得した。ステップ1-1で多特徴点画像を抽出するための閾値となる特徴点の数を800個とした結果、抽出された画像は412組となった。このうち、1往復目の前半部分を除いた398組の画像をステップ2のパラメータの最適化に利用した。最適化に用いた各パラメータの初期値、最適化の際に使用する各パラメータの変化量 δ の初期値および終了条件の閾値を表4.3に示す。初期値の決定方法は第3.4.2項に従う。ただし、振り子の長さを持ち上げ角度は、真値に対して測定で起こりえる誤差を付与した1.25 mと38°を初期値とした。ステップ2-2の終了条件は、終了時エラー値が 1.0×10^{-2} を下回る、基準エラー値と終了時エラー値の差が 1.0×10^{-3} を下回る、または繰り返し回数が500回に達するのいずれかを満たした場合とした。また、各パラメータの最適化処理の終了条件も、ステップ2-2と同様に $E_{\min} < 1.0 \times 10^{-2}$ 、 $E_{\text{base}} - E_{\min} < 1.0 \times 10^{-3}$ 、または繰り返し回数500回のいずれかを満たした場合とした。

まず、運動方程式のパラメータの最適化結果を表4.4に示す。また、図4.14に最適化したパラメータを用いた多特徴点画像の振り子の角度の推定結果を示す。図4.14の黒

表 4.3: 各パラメータの初期値と終了条件

パラメータ	初期値	δ の初期値	δ の終了条件
振り子の長さ (L)[m]	1.25	0.10	1.0×10^{-5}
抵抗係数 (b)[kg/s]	0.00	1.0×10^{-2}	1.0×10^{-6}
持ち上げ角度 (θ_{up})[°]	38.0	1.0	1.0×10^{-4}
初期角速度 (ω_0)[rad/s]	-2.19	0.10	1.0×10^{-5}

表 4.4: 運動方程式のパラメータの最適化結果

パラメータ	真値	最適化後の値
振り子の長さ (L)[m]	1.20	1.199
抵抗係数 (b)[kg/s]	5.0×10^{-2}	4.990×10^{-2}
持ち上げ角度 (θ_{up})[$^{\circ}$]	45.0	44.47
初期角速度 (ω_0)[rad/s]	-2.04	-2.045
エラー値 [$^{\circ 2}$]	—	0.3352

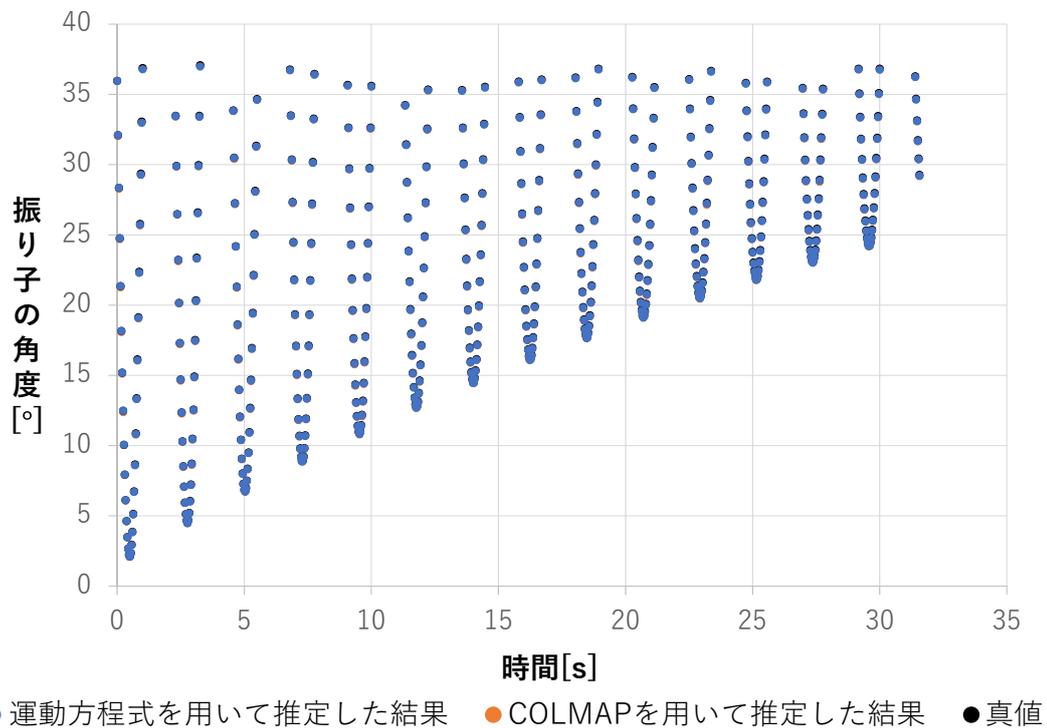


図 4.14: 最適化したパラメータを用いた多特徴点画像の振り子の角度の推定結果

色は振り子の角度の真値，橙色は COLMAP のトラッキングにより得られた振り子の角度，青色は提案手法で推定した振り子の角度であるが，それぞれの結果が近い値であるため，一番上に描いた青色の点のみが見えている状態となっている．表 4.4 より，振り子の長さ，抵抗係数，初期角速度については，誤差 1% の範囲内であり，精度高く推定できていることがわかる．持ち上げ角度のみ，誤差 1% を超える結果となった．一方で，ステップ 2 - 2 が終了した時点のエラー値は，0.3352 と小さい値になっている．また，図 4.14 の青色の点は橙色の点が見えない程重なっている．さらに，多特徴点画像のトラッキング結果から計算した振り子の角度と，提案手法を用いて推定した振り子の角度の，誤差の絶対値の平均は $2.26 \times 10^{-2}^\circ$ ，標準偏差は $1.82 \times 10^{-2}^\circ$ となった．これらの結果から，最適化によって COLMAP を用いて得たトラッキング結果と整合するパラメータを推定できることがわかる．次に，提案手法を用いて推定した振り子の角度とその真値を比較すると，誤差の絶対値の平均は $5.26 \times 10^{-2}^\circ$ ，標準偏差は $1.65 \times 10^{-2}^\circ$ となり，トラッキング結果から計算した振り子の角度を用いた場合と比較すると値は大きい，依然として差が小さいことがわかる．以上の結果から，理想的な仮想環境において，最適化によって真値に近いパラメータを推定できることが確認された．

次に，取得した全ての画像を用いた 3 次元再構成の結果について述べる．カメラ姿勢の真値を用いて 3 次元再構成した結果を図 4.15 に，既存手法 (SfM) で 3 次元再構成した結果を図 4.16 に，取得した画像に提案手法を適用した結果を図 4.17 に示す．ここで SfM には，ステップ 1 - 2 と同様に COLMAP を用いた．全画像を入力とした結果，それぞれ使用された画像の組は，真値を用いたものは 1,000 組，COLMAP を利用したものは 757 組，提案手法を適用したものは 948 組であった．提案手法では，1 往復目の後半から運動方程式の数値計算を行っており，それ以前の画像を除いているため，949 組となっている．また，COLMAP を用いた場合，特徴が少ない領域のみを写した画像を使用できないため，他の手法に比べて使用する画像の組が少なくなった．また，提案手法で使用した全画像の振り子の角度の推定結果を図 4.18 に示す．図 4.18 は，多特徴点画像のみの場合のグラフと同様に，黒色は振り子の角度の真値，橙色は COLMAP のトラッキングにより得られた振り子の角度，青色は提案手法で推定した振り子の角度である．

まず，図 4.16 と図 4.17 を比較すると，それぞれの図の赤い四角で示す領域からわかるように，提案手法は COLMAP に比べてより広く特徴の少ない領域を再構成できていることがわかる．一方で，両方の手法で 3 次元再構成ができる領域については，

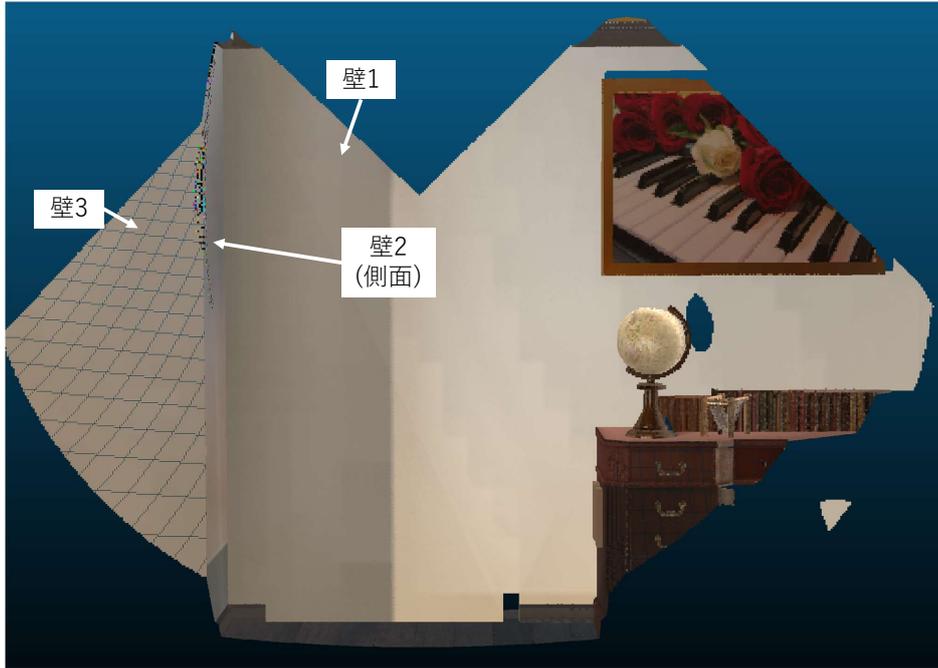


図 4.15: カメラ姿勢の真値を用いた仮想環境の3次元再構成結果

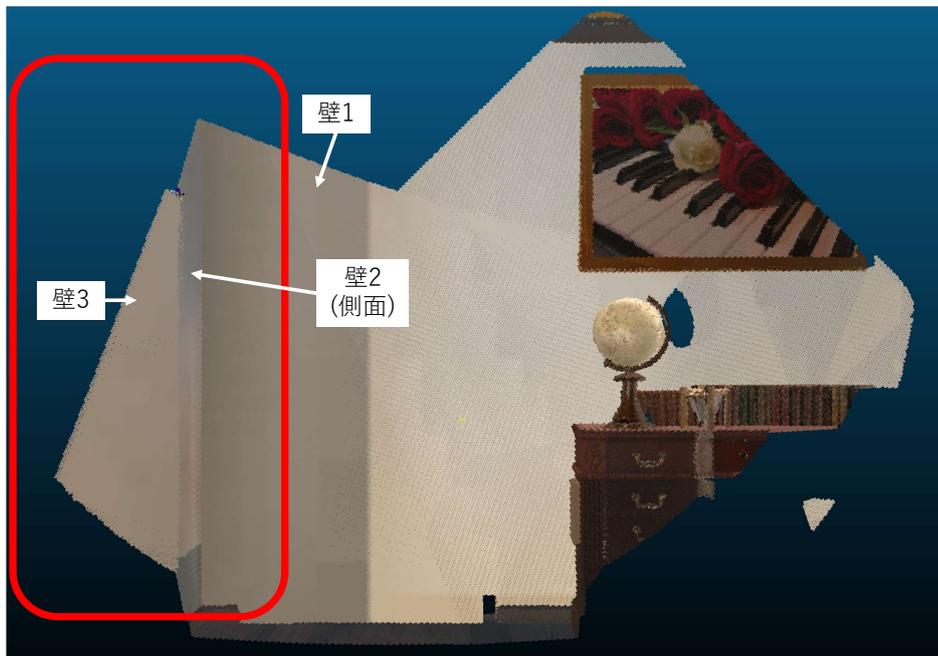


図 4.16: 既存手法 (COLMAP) によって得られた仮想環境の3次元再構成結果

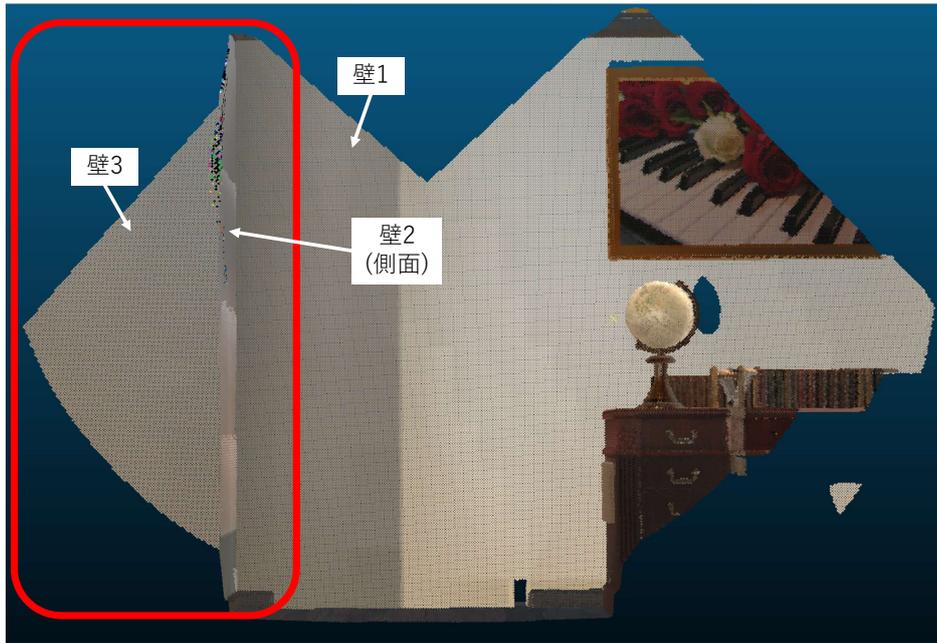


図 4.17: 提案手法による仮想環境の3次元再構成結果

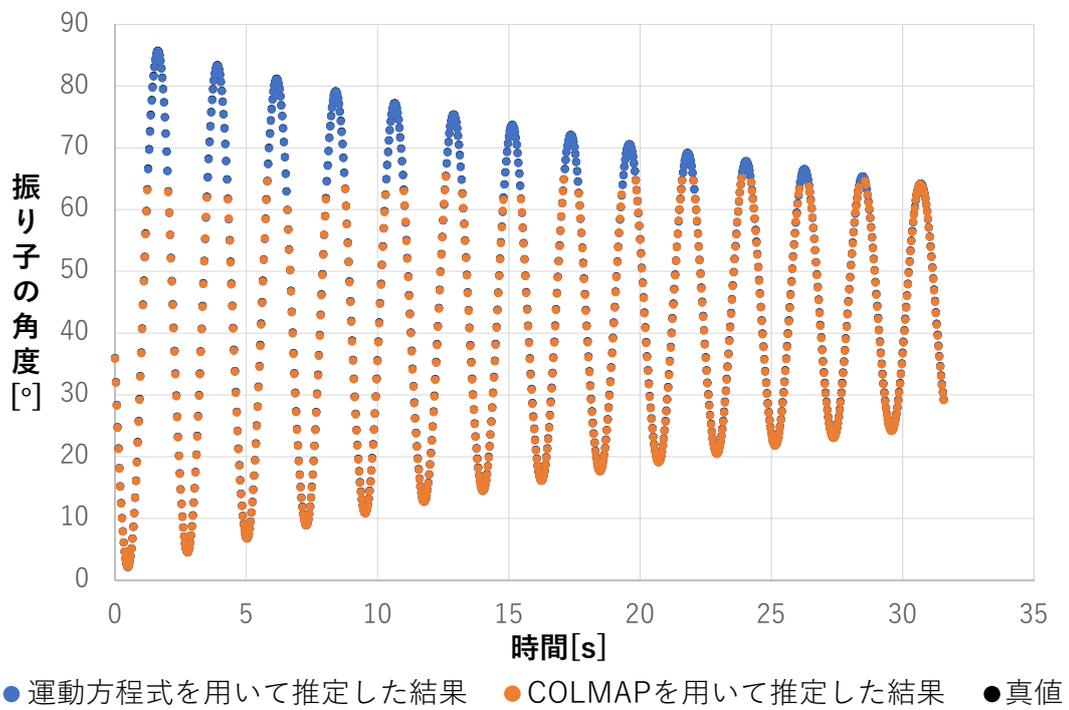


図 4.18: 最適化したパラメータを用いた全画像の振り子の角度の推定結果

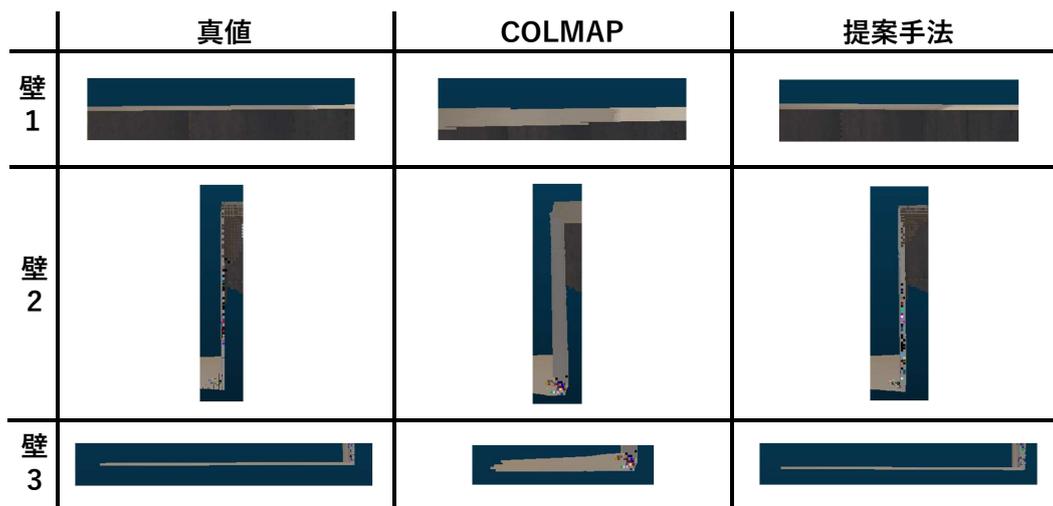


図 4.19: 各再構成結果の壁部分を真上から見た結果の比較

2つのモデルの精度の差はほとんど見られない。また、図 4.18 から、両方がともに振り子の角度を推定できる領域については差が小さいとわかる。しかし、各画像の振り子の角度の推定結果と真値との絶対誤差は、提案手法が平均 $5.94 \times 10^{-2}^\circ$ 、標準偏差 $1.62 \times 10^{-2}^\circ$ であるのに対し、COLMAP は平均 0.114° 、標準偏差 $4.66 \times 10^{-2}^\circ$ であり、提案手法は COLMAP よりも振り子の角度の推定精度が高い。これは、COLMAP などの SfM 手法は、画像処理により特徴点を認識し、その結果を用いて Perspective n-points Problem (PnP) 問題などを解くことで、カメラの姿勢を推定しているが、カラー画像にノイズやレンズ歪みが含まれないような理想に近い場合でも、特徴点が十分に写っていない画像を処理に含めてしまうと推定結果に誤差が含まれることを示している。一方、提案手法を用いることにより、カメラの運動が運動方程式に従うという拘束条件を付加することができ、より高精度にカメラの姿勢を推定できることを示している。また、各 3次元再構成結果の壁部分を真上からみたものを比較した結果を図 4.19 に示す。3次元再構成で得られるモデルは物体の表面を観察して得た情報を基に処理をしており、Unity で作成した 3次元モデルはそれぞれの壁を 1枚の平面としてモデル化しているため、推定結果が真値に近いほど、図 4.19 の真値の結果のように壁が薄い平面になる。しかし、COLMAP を利用したものは提案手法と比べ、どの壁でも 3次元再構成結果が太くなっており、対象ピクセルの 3次元位置を正しく推定できていないことがわかる。これは、特徴点が少ないため、COLMAP によるカメラ姿勢の推定が適切に行われていないことが原因であると考えられる。

次に、図 4.15 と図 4.17 の一部を拡大したものを図 4.20 に示す。図 4.20 から、Unity

	真値	提案手法
机上		
絵画		

図 4.20: カメラ姿勢の真値を用いた仮想環境の再構成結果と提案手法による仮想環境の再構成結果の比較

で作成した環境においては、提案手法は真値を用いて作成したモデルと遜色ないモデルを作成できている。

以上の結果から、提案手法を実装したソフトウェアは正しく動作し、振り子カメラが平面上でぶれなく振動し、デプス画像にノイズが含まれないような理想的な環境においては、特徴の少ない領域の抜け落ちのない3次元再構成が実現できることが示された。

4.4 ハードウェアの実装

本研究で実装した振り子カメラの全体図を図 4.21 に示す。振り子カメラは、全体を支える台とカメラを吊るすための支柱、環境を撮影するための RGB-D カメラ、RGB-D カメラを制御するための小型 PC、それらを駆動するためのバッテリーで構成される。

提案手法では、画像を取得した時刻を基に振り子カメラの姿勢を求めるため、画像を撮影した時刻を正確に取得できる必要がある。また、振り子カメラがある程度速く



図 4.21: 実装した振り子カメラの全体図

運動しても、歪みのない画像を取得できることが重要である。一般に、電子式のカメラの撮影方式には、ローリングシャッター方式とグローバルシャッター方式の2種類が存在する。ローリングシャッター方式は、カメラに搭載されるイメージセンサの上から順番に光を取り込むため、移動する物体を撮影する場合、画像に歪みが発生する。一方、グローバルシャッター方式は、全てのイメージセンサが同時に光を取り込むため、移動する物体を撮影しても画像に歪みは発生しない。そのため、振り子カメラに使用するカメラはグローバルシャッター方式であることが望ましい。そこで本研究では、画像を撮影した時刻を正確に取得でき、グローバルシャッター方式を採用している RGB-D カメラとして、Intel 社の RealSense D455^[32](以下、RealSense) を用いた。RealSense の仕様を表 4.5 に示す。実際に使用した解像度等は第 5 章で述べる。

表 4.5: 撮影に使用した RGB-D カメラの仕様

項目	仕様
カラー画像解像度	1280×800 pixels (30 fps)
RGB センサ FOV	水平 90°, 垂直 65°, 対角 98°
デプス画像解像度	1280×720 pixels (90 fps)
深度測定範囲	0.6 ~ 6 m
深度センサ FOV(16:9)	水平 87°, 垂直 58°, 対角 95°
深度センサ FOV(4:3)	水平 75°, 垂直 62°, 対角 89°

RealSense を制御する小型 PC には、Raspberry Pi 4 Model B(以下、Raspberry Pi) を用いた。また、Raspberry Pi の電源としては、Raspberry Pi および RealSense を駆動するのに必要な電流 (3 A) を供給可能な、Novoo PowerCube 5000 mAh モバイルバッテリーを使用した。モバイルバッテリーの仕様を表 4.6 に示す。提案手法の実行には表 4.7 に示す PC を使用した。

振り子カメラの構成図を図 4.22 に示す。1 台の振り子カメラで撮影する範囲を広くするためには、振り子カメラを吊るす支柱を長くする必要があるが、そのためには振り子を吊るすための台を高くし、支点の位置を高くしなければならない。本研究では屋内に設置することのできる高さの台として高さ 2.17 m の台を作成した。この時、床から振り子カメラの支点までの高さは 2.15 m となった。支柱部分には幅 15mm、奥行 20 mm、長さ 1.3 m、厚さ 1.5 mm、重さ 316 g のアルミ角パイプを用いた。RealSense

表 4.6: 使用したバッテリーの仕様

項目	仕様
バッテリータイプ	Li-ion バッテリー
電池容量	5000 mAh
USB-C 入力 (PD)	DC 5 V 2.1 A / 9 V 1.4 A / 12 V 1 A
USB-C 出力 (PD)	DC 5 V 3 A / 9 V 2 A / 12 V 1.5 A (18 W 最大)
USB-A 出力	DC 5 V 2.1 A / 9 V 2 A / 12 V 1.5 A (18 W)
合計出力	18 W
サイズ	76×35×27 mm

表 4.7: 使用したメイン PC の仕様

項目	仕様
OS	Windows 10 Pro 64 bit
CPU	Intel(R) Core(TM) i7-11700
RAM	32.0 GB

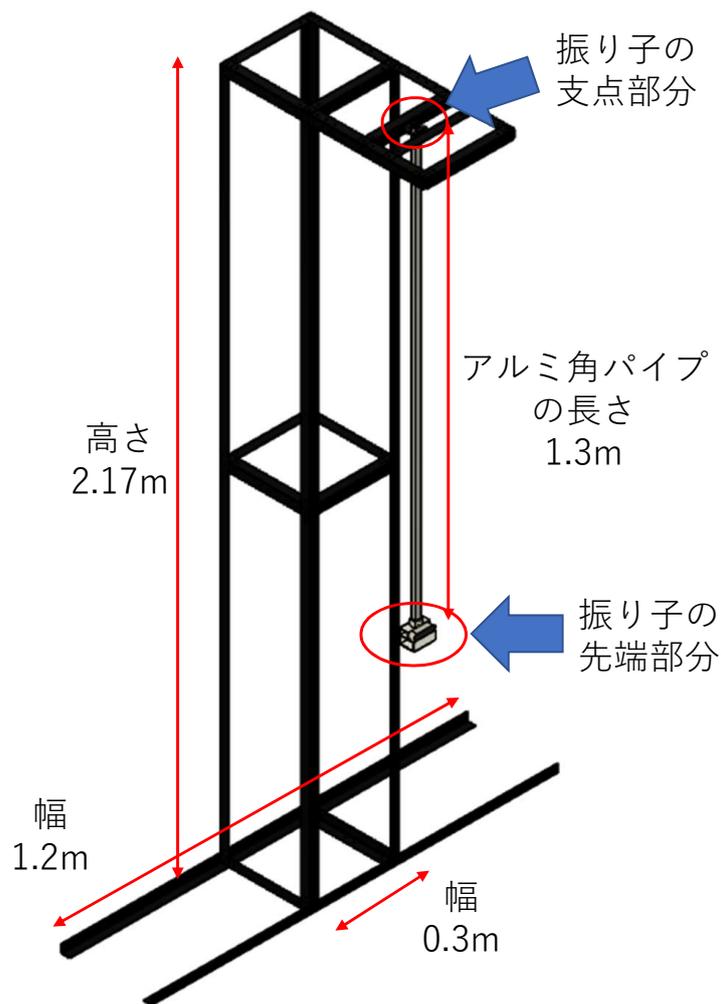


図 4.22: 振り子カメラの構成図

は広角であり、1枚の画像で広い範囲を撮影できるため、支柱の長さを長くしすぎると取得画像中の地面の割合が大きくなる。この点を考慮し、支柱の長さは台の高さの半分より少し長くなるように決定した。

次に振り子カメラの各部分の詳細について述べる。振り子カメラの支点部分の模式図を図 4.23 に示す。振り子カメラの支点部分では、アルミ角パイプを軸に通し、両側をボールベアリングと接続して滑らかに回転するようにした。この構造により、紐を用いて吊るした振り子と比べて、カメラが光軸方向に対して揺れることが少なくなるため、カメラが常に正面を向きながら、同一平面上を運動できる。また、ボールベアリングを使用しているため、支点部分で発生する摩擦を低減し、振り子運動に与える摩擦力の影響を小さくできるという利点もある。振り子の先端部分は、3D プリンタを用いて作成したケースに RealSense, Raspberry Pi, モバイルバッテリーを取り付け、

そのケースを支柱の先に固定した。実際に作成した振り子の先端部分の写真を図 4.24 に示す。筐体はアールエスコンポーネンツ社の DesignSparkMechanical を用いて CAD データを作成し、3D プリンタである MakerBot 社の Replicator+ を用いて印刷した。作成した振り子カメラの先端部分のサイズは図 4.24 に示す通りであり、重さは 378g であった。

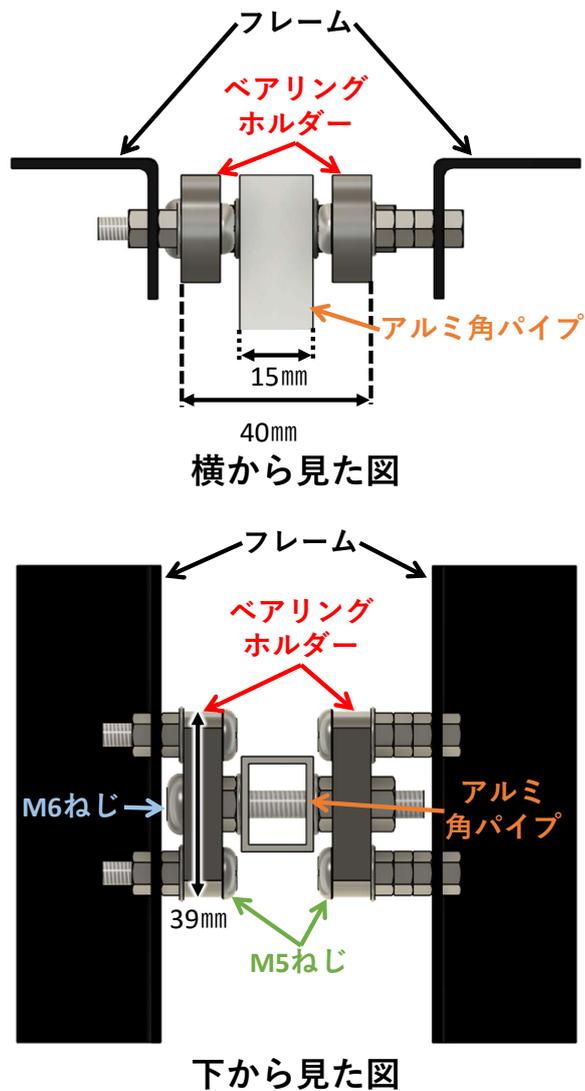


図 4.23: 支点部分の模式図

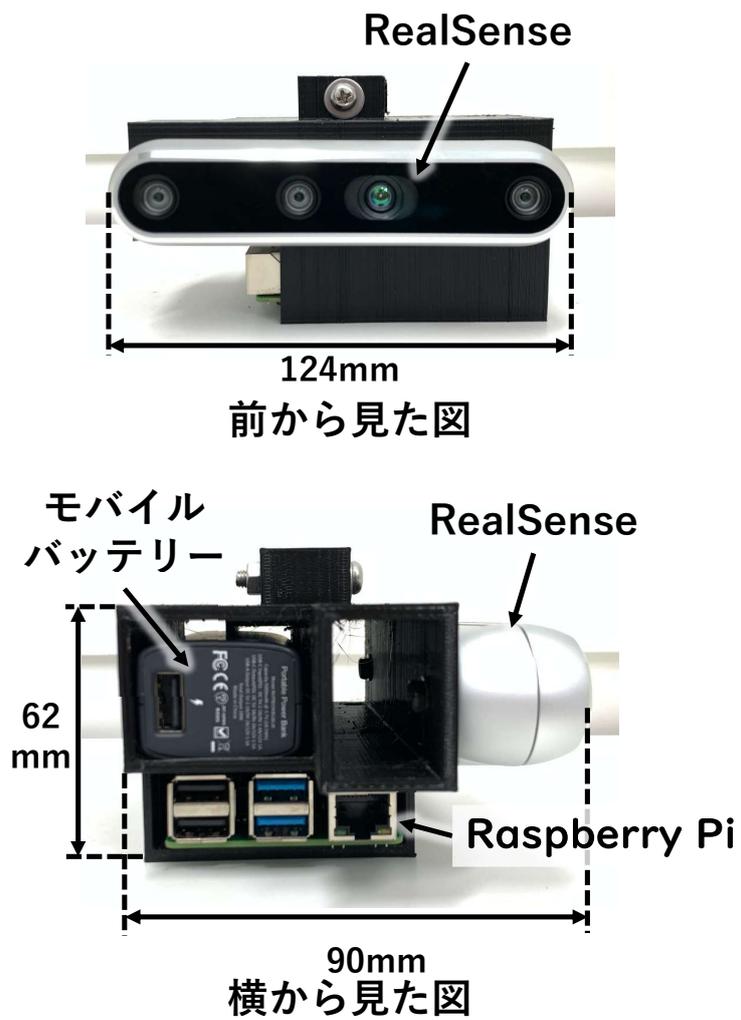


図 4.24: 振り子カメラの先端部分

第 5 章 現実環境を撮影した画像セットによる提案手法の評価

提案手法を評価するため、第 4 章で実装したソフトウェアとハードウェアを使用し、現実環境を対象に 3 次元再構成を行う。本章では、評価の目的と方法、そして評価結果について述べる。

5.1 評価の目的

本評価は、提案手法が現実環境においても、特徴の少ない領域が抜け落ちることなく 3 次元再構成できるかを評価することを目的とする。そのため、第 4 章で実装したハードウェアを用いて現実環境を撮影し、実装したソフトウェアで取得した画像を処理して 3 次元再構成を行う。本評価では、取得した画像を既存手法を用いてトラッキングした結果のみを利用して 3 次元再構成した場合 (以下、既存手法) と、提案手法を用いて 3 次元再構成した場合のそれぞれで得られた 3 次元モデルを用い、3 次元再構成が可能な領域の広さと作成したモデルの精度を比較することで提案手法を評価する。

5.2 評価に用いた環境とその環境の撮影方法

5.2.1 評価に用いた環境と振り子カメラの設置位置

評価で使用した環境の見取り図を図 5.1 に、環境の写真を図 5.2 に示す。特徴が少ないために既存手法では 3 次元再構成が困難となる領域を含む環境でも、特徴の少ない領域が抜け落ちることなく 3 次元再構成できることを示すため、特徴の多い領域と少ない領域の両方を含む環境を用意した。具体的には、特徴が少ない環境として、京都大学本部構内総合研究 10 号館 231 号室付近の廊下を選定し、特徴の多い領域は、図 5.2 の右側に示すように、扉に様々な色のポスターを貼ることで作成した。

振り子カメラは、図 5.1 の赤い四角で示す領域に、特徴点が多い扉側も撮影できるように設置した。振り子カメラの設置位置は、振り子カメラが最下点に来た時に撮影した画像に、特徴の多い領域が極力写らなくなるように決定した。これは、特徴の多い

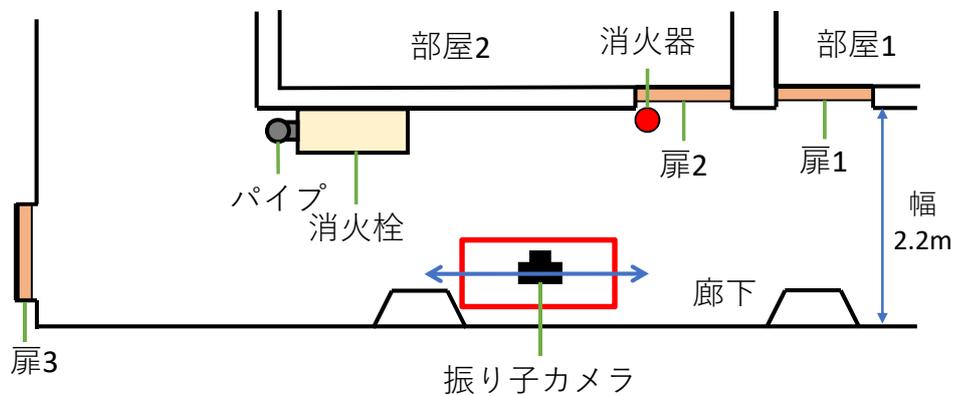


図 5.1: 評価に用いた環境の見取り図



図 5.2: 評価に用いた環境

領域を写した画像を一定枚数確保した上で、特徴の少ない領域をできるだけ広く撮影するためである。

5.2.2 環境の撮影方法と取得した画像の前処理

振り子カメラを用いた環境の撮影方法について述べる。まず、特徴の多い領域が運動開始時に写るように振り子カメラを持ち上げる。次に、カメラを制御する Raspberry Pi を操作し、撮影用ソフトウェアの処理を開始する。その後、カメラが撮影を開始したことを確認し、振り子カメラを持ち上げた手を放して振り子運動を開始させる。続いて、撮影が終了し、Raspberry Pi の記録メディアに画像の書き出しが開始されたことを確認した時点で振り子カメラの運動を停止させる。画像の書き出しが全て終了した後、全ての画像をメイン PC に転送する。メイン PC に画像を転送した後、デプス画像の焦点距離と焦点位置をカラー画像の焦点距離と焦点位置に合わせるアラインメントを行う。最後に、振り子カメラが運動を開始する前の画像を除き、使用する画像セットを作成する。

次に、デプス画像のアラインメントについて詳細を述べる。本評価で使用した RealSense は、カラー画像を取得する RGB カメラとデプス画像を取得するデプスカメラで、焦点距離および焦点位置が異なる。第 3.5 節で説明したステップ 2 – 3 で 3 次元再構成を行う際、カラー画像とデプス画像のピクセルを対応させる必要があるため、画像取得後にアラインメントを行う。本評価では、RealSenseSDK に含まれる alignment 関数を用いてアラインメントを実行した。最終的に取得したカラー画像とデプス画像の詳細を表 5.1 に示す。

次に、取得した画像から、運動を開始する前の画像を除いた画像セットを作成する方法を述べる。まず、取得した画像のうち、運動開始時付近の画像を目視で比較し、初

表 5.1: カラー画像およびデプス画像の詳細

項目	仕様
解像度	640×480 pixel
焦点距離	385 pixel
撮影フレームレート	60fps
画像フォーマット (カラー画像)	RGB 3 チャンネル, 1 byte/pixel
画像フォーマット (デプス画像)	1 チャンネル, 2 byte/pixel

めて変化が見られる画像を特定する。この画像を運動開始直後の画像とし、その直前の画像を運動開始時の画像とする。そして、運動開始時以降の画像を提案手法で使用する画像セットとする。

5.3 運動方程式と提案手法の処理に必要なパラメータの設定

本節では、運動方程式と提案手法の処理に必要なパラメータの初期値や閾値、および最適化の終了条件について述べる。最適化する各パラメータの初期値を表 5.2 に示す。振り子の長さは、振り子カメラの支点位置から先端部分のケースの底面までの長さが 1.35 m であり、これ以上の長さにはならないと考えられるため、この長さを初期値とした。また、振り子カメラは、図 5.3 に示すように、地面から約 1.24 m の地点まで持ち上げて運動を開始したため、最下点を基準に 47.6° 持ち上げられたと計算できる。これを基に、持ち上げ角度の初期値は 47.6° とした。角速度の初期値は、式 (3.5) より、 $L = 1.35$ m, $\theta_{\text{start}} = 47.6^\circ$ を用いて計算し、 -2.18 rad/s とした。また、今回作成した振り子カメラは、先端の質量が 378 g であるため、 $m = 0.378$ kg とした。各パラメータの最適化に使用する変化量 δ の初期値と閾値、ステップ 2-2 の終了条件と各パラメータの最適化の終了条件については、第 4.3 節と同じ値を使用した。

表 5.2: 運動方程式のパラメータの初期値

パラメータ	初期値
振り子の長さ (L)[m]	1.35
抵抗係数 (b)[kg/s]	0.00
持ち上げ角度 (θ_{up})[$^\circ$]	47.6
初期角速度 (ω_0)[rad/s]	-2.18

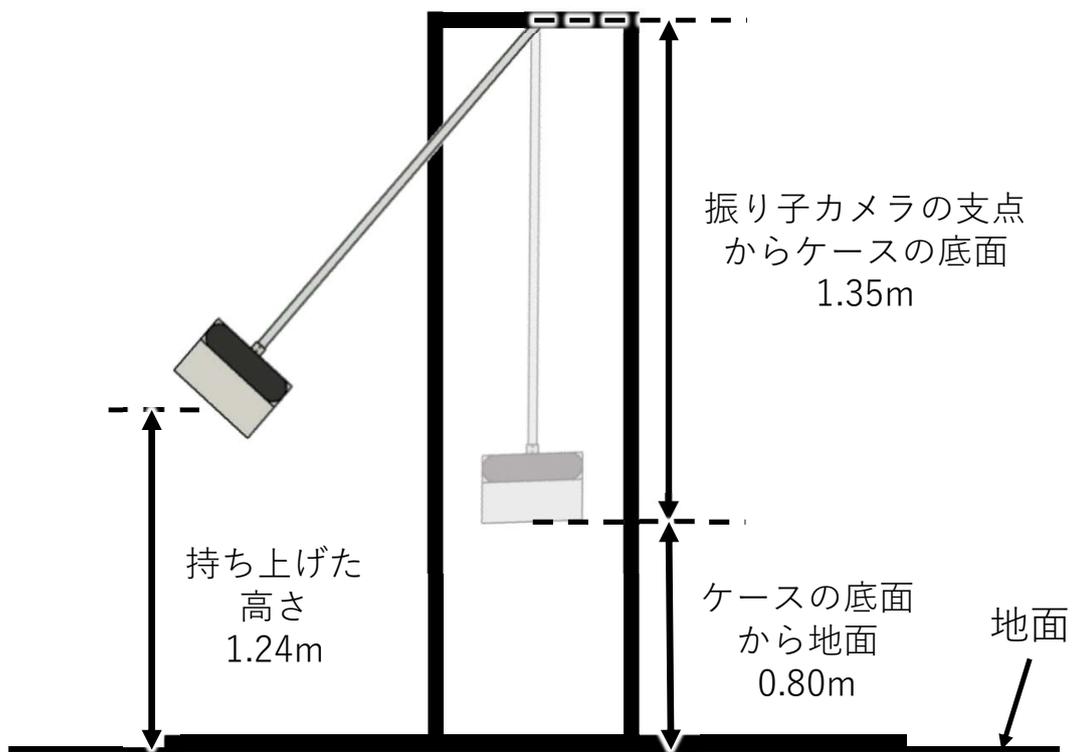


図 5.3: 振り子カメラの初期位置

5.4 結果と考察

5.4.1 取得した画像と多特徴点画像の抽出結果

本評価では、Raspberry Pi のメモリに余裕をもって一時保存できる撮影画像の組数として、1500 組のカラー画像とデプス画像を取得した。このうち、運動開始以前の 46 組の画像を除いた 1454 組の画像を提案手法で使用する画像セットとした。提案手法のステップ 1-1 で多特徴点画像を抽出する際の閾値とする特徴点の数を、第 4.3 節と同じ 800 個とした結果、434 組の画像が抽出された。このうち、1 往復目の前半部分を除いた 407 組の画像をステップ 2 のパラメータの最適化に利用した。

5.4.2 カメラ姿勢の推定結果と 3 次元再構成結果および考察

本評価では、既存手法として、第 4.3 章と同様に COLMAP を用いる。

まず、全ての画像のカメラ姿勢の推定結果を図 5.4 に示す。図 5.4 において、橙色の点はステップ 1-2 によって得られた多特徴点画像の振り子の角度、青色の点は提案手法で推定した振り子の角度を表す。図 5.4 から、提案手法は、多特徴点画像のトラッキング結果をもとに全ての撮影時刻における振り子の角度を推定できることがわかる。

次に、既存手法 (COLMAP) と提案手法の 3 次元再構成結果をそれぞれ図 5.5 および図 5.6 に示す。全画像である 1454 組の画像を入力とした結果、3 次元再構成に使用された画像の組は COLMAP が 976 組、提案手法が 1336 組であった。COLMAP は、第 4.3 節と同様に特徴の少ない領域のみを写した画像を使用できないため、提案手法に比べて使用される画像が少なかった。また、提案手法も、ステップ 2-2 で新たに定義した時刻 0 から運動方程式の数値計算を行っているため、それ以前の画像が使用されず、使用される画像の組数が減っている。図 5.5 と図 5.6 の赤い四角の領域を比較すると、提案手法の再構成結果には消火栓の端部分が存在するのに対し、COLMAP で再構成した結果には消火栓が映っていない。この結果から、提案手法は既存手法が 3 次元再構成できない特徴の少ない領域も 3 次元再構成できることがわかる。

以上の結果より、提案手法を用いることで、特徴の少ない領域のみを写した画像でもカメラ姿勢を推定でき、その結果を基に特徴の少ない領域の抜け落ちのない 3 次元再構成ができるといえる。

一方で、図 5.5 と図 5.6 のそれぞれにおいて黄色い四角で囲んだ扉部分を拡大したものを図 5.7 に示す。図 5.7 に注目すると、COLMAP の再構成結果では、黄色の四角で

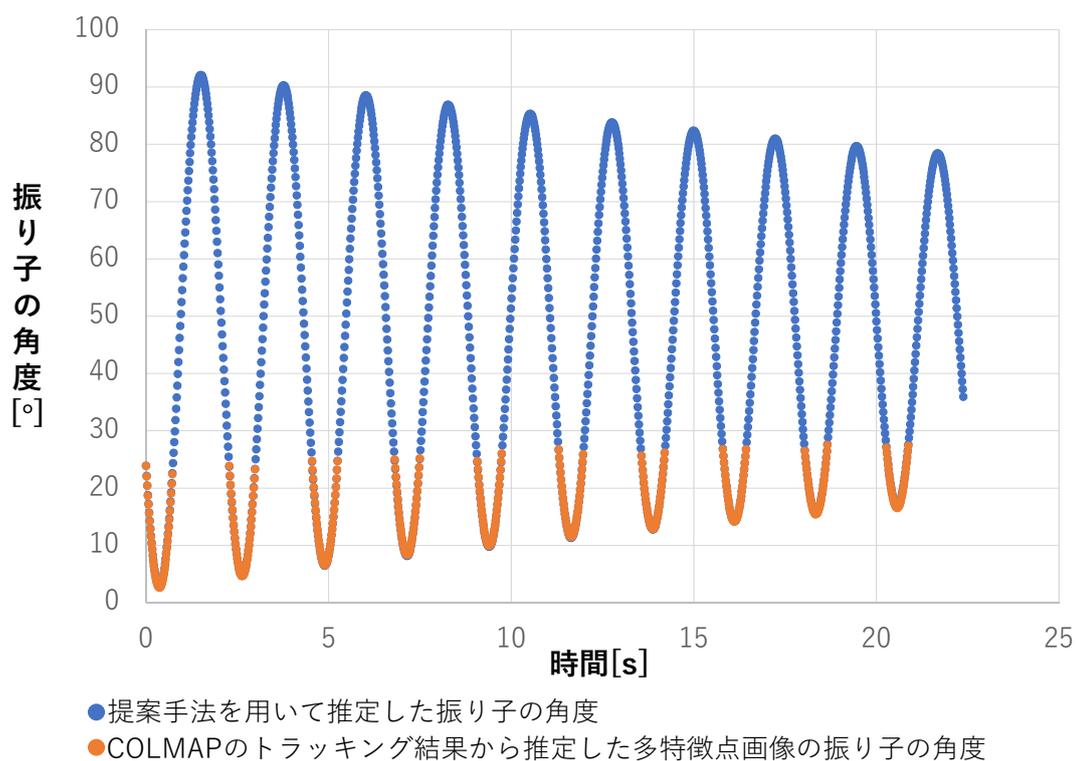


図 5.4: 全画像の振り子の角度の推定結果



図 5.5: 既存手法 (COLMAP) で得られた現実環境の3次元再構成結果



図 5.6: 提案手法による現実環境の再構成結果

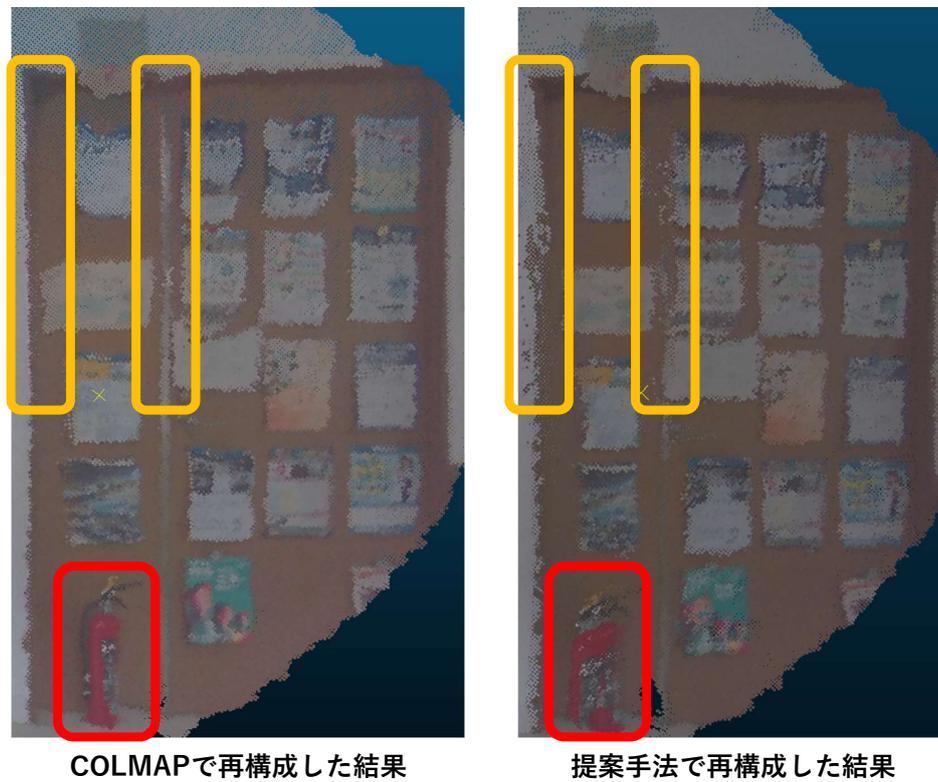
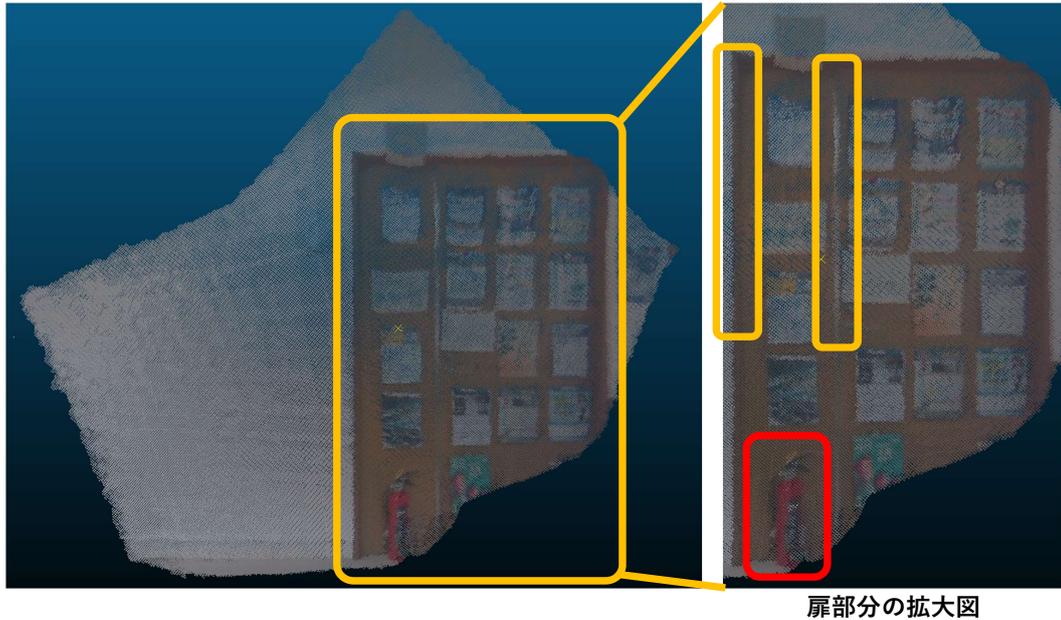


図 5.7: 既存手法 (COLMAP) と提案手法の現実環境の 3 次元再構成結果の比較



扉部分の拡大図

図 5.8: COLMAP のトラッキング結果を用いた多特徴点画像の現実環境の 3 次元再構成結果

囲んだドアの枠や銀色の縦の線が明瞭に映っているのに対し、提案手法ではぼやけている。同様に、赤い四角で囲んだ消火器の部分を比較しても、提案手法は COLMAP に比べて精度が低いことがわかる。これはカメラ姿勢の推定結果の誤差が大きいが原因だと考えられる。

ここで、カメラ姿勢の推定結果の誤差が大きくなる原因を究明するため、運動方程式のパラメータの最適化結果と、多特徴点画像の振り子の角度の推定結果を詳しく分析する。図 5.8 に COLMAP によるトラッキング結果を用いて、多特徴点画像のみで 3 次元再構成を行った結果を示す。また、図 5.9 に、図 5.4 の結果のうち、多特徴点画像の結果のみを抽出したものを示し、表 5.3 に各パラメータの最適化結果を示す。

図 5.8 を見ると、COLMAP によるトラッキング結果を利用した、多特徴点画像のみの 3 次元再構成は精度高く行われていることがわかる。この結果から、トラッキング結果のカメラ姿勢は信頼できる値であると考えられるため、このカメラ姿勢を用いて計算する同範囲の振り子の角度も信頼できると考えられる。次に、図 5.9 に注目すると、それぞれの振り子の角度は概形が酷似している。また、トラッキング結果により得られた振り子の角度と提案手法で推定した振り子の角度の差の絶対値の平均は 0.131° 、標準偏差は $7.27 \times 10^{-2}^\circ$ となり、振り子の角度の差は小さいことがわかる。以上の結果より、カメラ姿勢のうち、振り子の角度から計算されるカメラの方向は正しく推定で

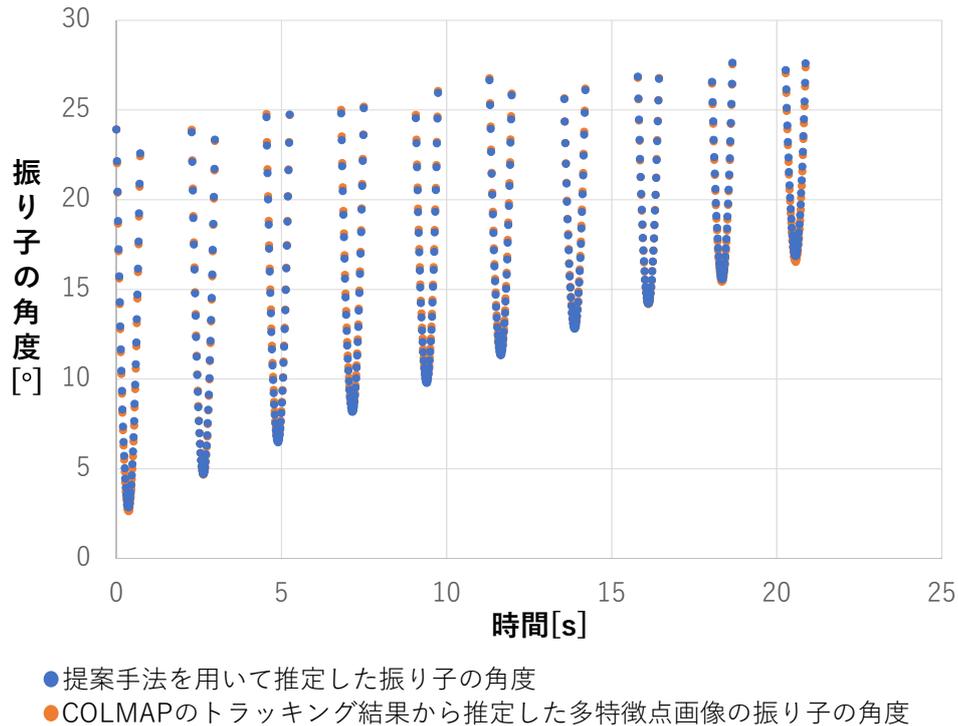


図 5.9: 多特徴点画像の振り子の角度の推定結果

きていると考えられる。最後に、表 5.3 に注目すると、振り子の長さが測定した値から大きくずれていることがわかる。カメラなどの機器を取り付けるケースの高さは、図 4.24 に示すように 62 mm であり、振り子カメラの支点から取り付けたケースの底面までの長さが 1.35 m であるため、振り子の長さは 1.29 m から 1.35 m の間の値になると考えられるが、最適化結果はそれよりも短い 1.184 m という結果になった。カメラ姿勢のうち、カメラの位置は式 (3.7) に示すように推定した振り子の角度と振り子の長さを用いて計算する。そのため、振り子の長さが真値から離れた値となったことが原因で、カメラの位置の推定が正しく行われず、3次元再構成を行って得られるモデルの精度が低下したと考えられる。

ここで、精度高く 3次元再構成できた第 4.3 節のシミュレーションと本評価の違いについて考える。2つの違いは、カメラを吊るす支柱部分にあると考えられる。第 4.3 節のシミュレーションでは、振り子を吊るす支柱の質量は考慮せず、カメラ部分のみ質量を持つとしていた。しかし、実際に作成した振り子カメラは支柱部分が先端部分と同程度の質量をもつため、カメラ部分のみ質量をもつとする仮定が成り立たないと考えられる。これを踏まえ第 5.5 節では、支柱の質量を考慮するように運動方程式を修正した上で再度評価を行う。

表 5.3: 各パラメータの最適化後の値

パラメータ	初期値	最適化後の値
振り子の長さ (L)[m]	1.35	1.184
抵抗係数 (b)[kg/s]	0.00	1.360×10^{-2}
持ち上げ角度 (θ_{up})[°]	47.6	47.95
初期角速度 (ω_0)[rad/s]	-2.18	-1.862
エラー値 [° ²]	-	9.147

5.5 支柱の質量を考慮した場合の3次元再構成

5.5.1 運動方程式の修正

本項では、支柱が一様な質量を持つと仮定した場合について考える。振り子の減衰を一旦無視して考えると、支柱が一様な質量を持つ場合の振り子の運動方程式は式 (5.1) とあらわすことができる。ただし、支柱の質量を M 、先端部分の質量を m とする。運動方程式の導出方法については付録 A で述べる。

$$\frac{d^2\theta}{dt^2} = -\frac{\frac{1}{2}M + m}{\frac{1}{3}M + mL} g \sin(\theta - \theta_{\text{up}}) \quad (5.1)$$

また、式 (3.3) の減衰項である $-\frac{b}{m} \frac{d\theta}{dt}$ について、支柱の質量を考慮した場合でも、抵抗は角速度に比例するという仮定は妥当であり、質量以外を修正する必要はないと考える。そのため、支柱が一様な質量を持つと仮定し、減衰を考慮した運動方程式は式 (5.2) となる。

$$\frac{d^2\theta}{dt^2} = -\frac{\frac{1}{2}M + m}{\frac{1}{3}M + mL} g \sin(\theta - \theta_{\text{up}}) - \frac{b}{M + m} \frac{d\theta}{dt} \quad (5.2)$$

第 5.5.2 項では、以上のように修正した運動方程式を用いて再度 3 次元再構成を行う。

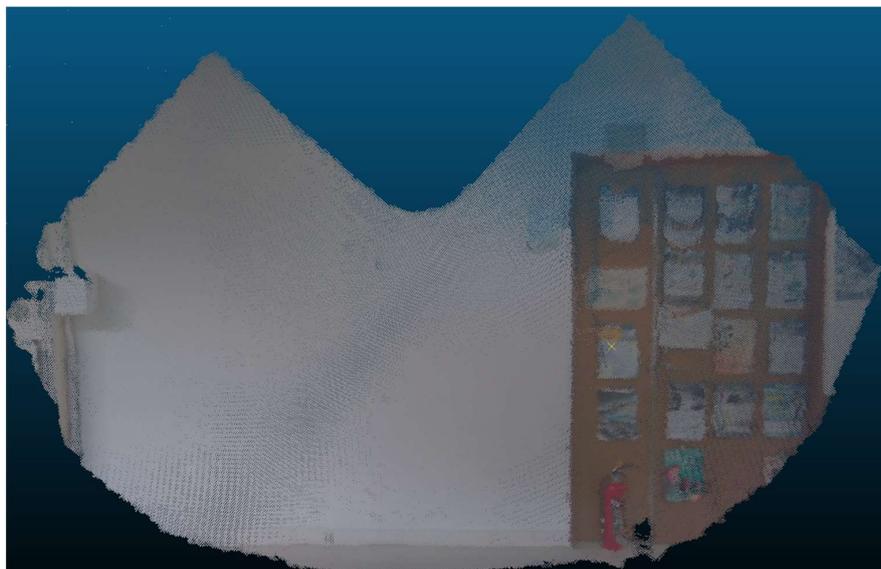
5.5.2 3次元再構成を行った結果と考察

本項では、3次元再構成を行った結果とその考察について述べる。式 (5.2) に用いる支柱の質量は $M = 0.316$ とした。

修正した運動方程式を用いて再度 3 次元再構成を行った結果を図 5.10 の下段に示す。また、図 5.10 の扉部分を比較した結果を図 5.11 に示す。図 5.10 より、運動方程式の修正後も特徴の少ない領域の再構成が可能であると確認できる。さらに、図 5.11 の比較



運動方程式修正後の再構成結果



運動方程式修正後の再構成結果

図 5.10: 運動方程式修正前後の 3 次元再構成結果の比較 (全体)

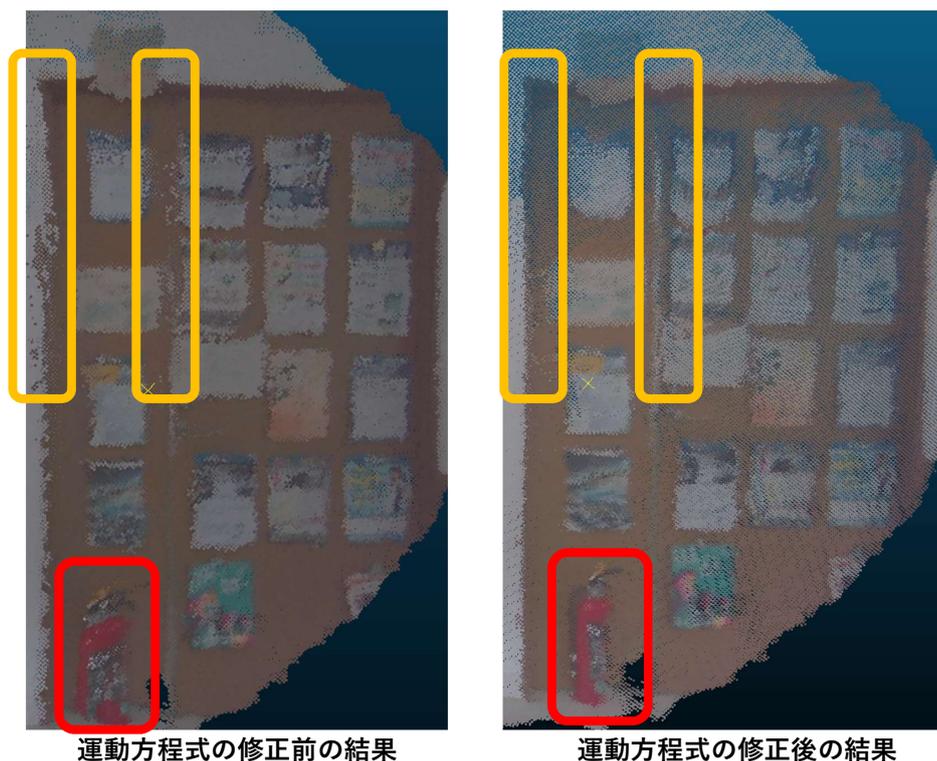


図 5.11: 運動方程式修正前後の 3 次元再構成結果の比較 (扉部分)

結果を見ると、運動方程式の修正前は扉の枠が歪んでいたのに対し、修正後は枠の部分が直線に近くなっている。さらに、消火器の輪郭が鮮明になるなど、3次元再構成の精度が向上している。

次に、多特徴点画像の振り子の角度の推定結果を図 5.12 に示す。また、最適化後の各パラメータの値を表 5.4 に示す。COLMAP を用いたトラッキング結果により得られた振り子の角度と提案手法で推定した振り子の角度の差の絶対値の平均は 0.131° 、標準偏差は $7.25 \times 10^{-2^\circ}$ となり、運動方程式修正後の振り子の角度の推定結果は修正前と同じである。また、図 5.9 と図 5.12 を比較しても違いが見られない。そのため、カメラの方向は修正前と同様に正しく推定できていると考えられる。一方で、振り子の長さの推定結果は 1.312 m となり、測定した値に近くなった。また、抵抗係数も修正前と変化しているが、これは支柱の質量も考慮したことで全体の質量が変化した結果であり、振り子の角度の推定精度には影響しないと考えられる。この結果から、式 (3.7) を用いて計算されるカメラの位置の推定精度が向上した結果、3次元再構成で得られるモデルの精度が向上したと考えられる。

以上の結果より、運動方程式修正後の提案手法を用いることで、現実環境においても、特徴の少ない領域が抜け落ちることなく 3次元再構成できることが示された。

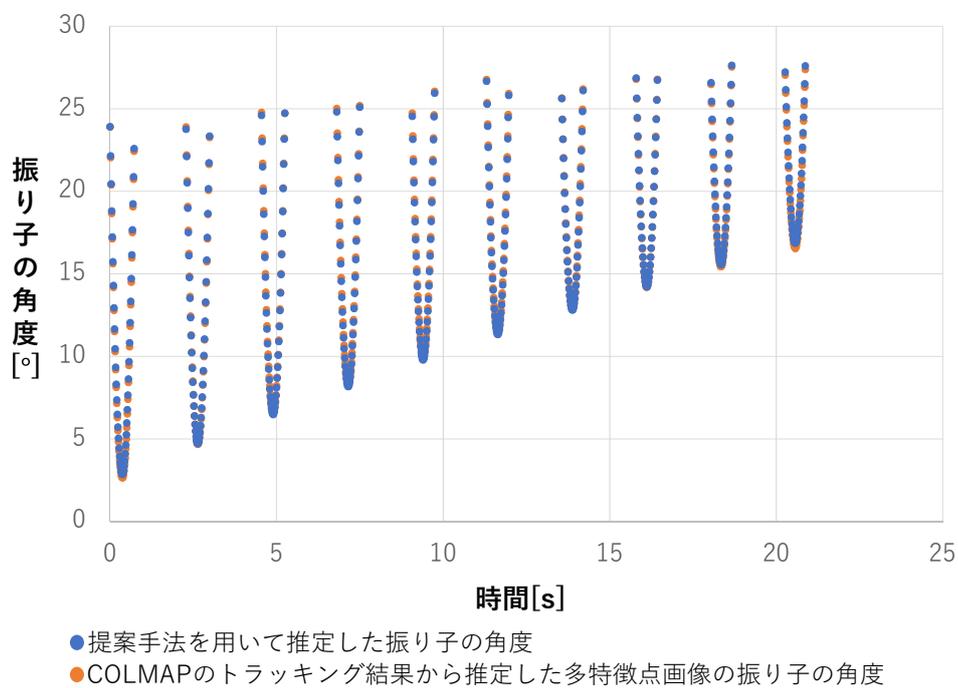


図 5.12: 運動方程式修正後の多特徴点画像の振り子の角度の推定結果

表 5.4: 運動方程式修正後の各パラメータの最適化結果

パラメータ	運動方程式修正前	運動方程式修正後
振り子の長さ (L)[m]	1.184	1.312
抵抗係数 (b)[kg/s]	1.360×10^{-2}	2.497×10^{-2}
持ち上げ角度 (θ_{up})[°]	47.95	47.94
初期角速度 (ω_0)[rad/s]	-1.862	-1.862
エラー値 [° ²]	9.147	9.138

なお，作成するモデルの精度をさらに向上させるために，支柱が一様な質量を持つと仮定したうえで減衰を考慮した，より厳密な運動方程式を立てる必要があると考えられる．さらに，提案手法では定数であると仮定した重力加速度についても，測定場所によって値が変わるため，最適化することでカメラ姿勢の推定精度が向上し，作成するモデルの精度も向上できる可能性がある．

第 6 章 本研究のまとめと今後の展望

6.1 本研究のまとめ

本研究では、RGB-D カメラを振り子の先端に取り付けた振り子カメラを用いて環境を撮影し、撮影時刻と振り子の運動法則を利用してカメラ姿勢を推定することで、特徴の少ない領域でも抜け落ちなく 3次元再構成できる手法(以下、提案手法)を開発することを目的とした。

提案手法では、まず、振り子カメラを用いて環境を撮影する。次に、取得した画像のうち、特徴点が十分に含まれる画像を抽出し、それらの画像から画像が撮影されたときのカメラ姿勢を SfM を利用して推定する。その後、カメラ姿勢の推定結果を用いて振り子の運動方程式のパラメータを推定し、それらを用いて運動方程式を数値計算することで全ての画像のカメラ姿勢を推定する。最後に、推定したカメラ姿勢と取得した画像を用いて 3次元再構成を行い、環境の 3次元モデルを得る。

提案手法の実装後、提案手法が目的を達成していることを確認するため、Unity で作成した理想的な仮想環境と現実環境の両方で評価実験を行った。まず、仮想環境での評価では、提案手法で行った最適化により、運動方程式のパラメータは真値に近い値を得ることができた。また運動方程式を用いて推定したカメラ姿勢の推定精度は、SfM のみを用いる既存手法よりも高いことを確認できた。これにより、特徴の少ない画像でもカメラ姿勢を正しく推定でき、特徴の少ない領域でも 3次元再構成が可能であることが示された。次に、現実環境で提案手法を評価した結果、仮想環境と同様に、提案手法は特徴の少ない領域を再構成することができたが、作成したモデルは既存手法に比べて精度が低かった。これは、提案手法で仮定した運動方程式では、振り子を吊るす支柱の重さを考慮していなかったが、実際に作成した振り子カメラでは支柱が先端部分と同程度の重さを持っており、現実の運動方程式から乖離していたためであると考えられた。そのため、支柱部分の質量を考慮するように運動方程式を修正して再度評価を実施した。その結果、特徴の少ない領域でも 3次元再構成でき、さらに精度も向上した。以上より、提案手法は現実環境においても、特徴の少ない領域でも 3次元再構成が可能であることが示された。

6.2 今後の展望

本研究では、振り子カメラを用いて環境を撮影し、振り子の運動法則を用いることで、特徴の少ない領域でも抜け落ちなく3次元再構成できる手法を開発した。しかし、本研究で作成した振り子カメラは、カメラを紐ではなく支柱を用いて固定しており、常に一定の方向を向いているため、カメラを最初に向けた方向以外の3次元再構成は困難である。また、1台の振り子カメラで撮影できる範囲にも限界がある。そのため、複数の振り子カメラを組み合わせたリ、既存の手法と組み合わせたりするなど、より広い範囲の再構成を可能にする方法を実現する必要がある。

また、本研究の振り子カメラは、支柱で吊るしており、摩擦を低減させるために支点部分にベアリングを用いているため、設置するために台が必要となり、可搬性が低い。さらに支柱の長さが固定されているため、撮影環境に応じて支柱を交換しなければならず、環境撮影のための準備に手間がかかる。そのため、カメラを取り付けたケースを吊るす部分を支柱ではなく紐にするなど、環境の撮影が容易なシステムに改良する必要もある。

謝 辞

この研究は、関係者の皆様のご協力に支えられ、成し遂げることができました。この場を借りて、深く感謝申し上げます。

下田宏教授には、研究会などで不十分な点に対するご指摘をいただき、またハードウェアに関して部品の選定や工具の使い方など様々な助言をいただきました。石井裕剛准教授には研究テーマの立案から論文執筆まで、研究活動全般に関して様々な支援をいただきました。特に、提案手法の開発にあたって非常に多くの時間を割いていただき、また困ったときには親身に相談に乗っていただき、適切な助言を多数いただきました。上田樹美助教授は、研究に関する助言だけでなく日常生活でも常に気にかけて下さりました。秘書の普照郁美さんは研究に関する事務手続きをして下さり、研究がスムーズに進むよう支援していただきました。

さらに、下田研究室の皆様にも大変なご協力を賜りました。修士1回生の尾籠君、阿部さんには、多忙な中で論文の執筆に協力し、添削など多大なサポートを提供していただきました。また、博士課程のOrchidaさん、修士1回生の田中君、荒木君、4回生の尾崎君、神田君、宮本君、山崎君には研究会で貴重な意見をいただき、また、普段の研究生活も楽しく過ごすことができました。同期の松岡君と佐々君、修士1回生の松永君、博士課程の竹内さん、OBの深田さん、野村さんとは、研究活動だけではなく、研究の合間に雑談や食事に行くなど、心の支えとなりました。また、研究を進める上で互いに協力し合った同期の小野君、山田さん、Sun君にも深く感謝します。

最後に、日常生活において支えてくださった家族や友人にも、深く感謝申し上げます。

参考文献

- [1] Hirotake Ishii: Plant maintenance and dismantling work support based on three-dimensional scanning technology, *International Journal of Nuclear Safety and Simulation*, 4(2), pp.97–104 (2013).
- [2] Ahmad K. Sleiti, Jayanta S. Kapat, Ladislav Vesely: Digital twin in energy industry: Proposed robust digital twin for power plant and other complex capital-intensive large engineering systems, *Energy Reports*, 8, pp.3704–3726 (2022).
- [3] 崔炳賢, 西田明美, 川田学, 塩見忠彦, Y. LI: 原子炉建屋の3次元有限要素モデルを用いた地震応答解析手法に関わる標準的解析要領(受託研究), 日本原子力研究開発機構 JAEA-Research(2021-017) (2022).
- [4] Wataru Asaba, Kimi Ueda, Hirotake Ishii, Hiroshi Shimoda: Evaluation of the Effect of VR Disaster Experience in Familiar Environment, *Integrating People and Intelligent Systems* (2023). Vol. 69.
- [5] 静岡県: 静岡県が進める VIRTUAL SHIZUOKA 構想 とは? (2024).
<https://www.pref.shizuoka.jp/machizukuri/1049255/1052183.html> (Retrieved January 2, 2025).
- [6] TERRA LABO: ドローン空撮「共通状況図(ベースマップ)」を作成、災害対策本部へ提供 (2021).
<https://terra-labo.jp/eventnews/2572/> (Retrieved January 2, 2025).
- [7] 国土交通省: PLATEAU[プラトー].
<https://www.mlit.go.jp/plateau/> (Retrieved January 30, 2025).
- [8] 大阪大学適塾記念センター: 適塾からのお知らせ 1(適塾建物の3Dモデル完成) (2024).
https://www.tekijuku.osaka-u.ac.jp/ja/special_news/copy4_of_paj907 (Retrieved December 28, 2024).

- [9] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, Duane Fulk: The digital Michelangelo project: 3D scanning of large statues, pp.131–144 (2000).
- [10] Brent Schwarz: Mapping the world in 3D, *Nature Photon*, **4**, pp.429–430 (2010).
- [11] 日本原子力研究開発機構：廃炉現場の汚染分布を3次元マップで”見える化”—見えない汚染を仮想空間で把握し、作業員の被ばくを低減—。
<https://www.jaea.go.jp/02/press2021/p21051403/> (Retrieved December 28, 2024).
- [12] Johannes Lutz Schönberger, Jan-Michael Frahm: Structure-from-Motion Revisited, *Conference on Computer Vision and Pattern Recognition*, pp.4104–4113 (2016).
- [13] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, Jan-Michael Frahm: Pixelwise View Selection for Unstructured Multi-View Stereo, *European Conference on Computer Vision*, pp.501–518 (2016).
- [14] Xiang Gao, Lingjie Zhu, Hainan Cui, Zhanyi Hu, Hongmin Liu, Shuhan Shen: Complete and Accurate Indoor Scene Capturing and Reconstruction Using a Drone and a Robot, *IEEE Sensors Journal*, **21**(10), pp.11858–11869 (2021).
- [15] Raul Mur-Artal, J. M. M. Montiel, Juan D. Tardos: ORB-SLAM: A Versatile and Accurate Monocular SLAM System, *IEEE Transactions on Robotics*, **31**(5), pp.1147—1163 (2015).
- [16] Sungjoon Choi, Qian-Yi Zhou, Vladlen Koltun: Robust reconstruction of indoor scenes, *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pp.5556–5565 (2015).
- [17] Raúl Mur-Artal, Juan D. Tardós: ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras, *IEEE Transactions on Robotics*, **33**(5), pp.1255–1262 (2017).

- [18] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, Christian Theobalt: BundleFusion: Real-Time Globally Consistent 3D Reconstruction Using On-the-Fly Surface Reintegration, *ACM Transactions on Graphics*, **36**(3), pp.1–18 (2017).
- [19] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng: NeRF: representing scenes as neural radiance fields for view synthesis, *Communications of the ACM*, **65**(1), pp.99–106 (2021).
- [20] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, Angjoo Kanazawa: PlenOctrees for Real-time Rendering of Neural Radiance Fields , 2021 IEEE/CVF International Conference on Computer Vision, pp.5732–5741 (2021).
- [21] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, Pratul P. Srinivasan: Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields, 2021 IEEE/CVF International Conference on Computer Vision, pp.5835–5844 (2021).
- [22] Yuki Harazono, Hirotake Ishii, Hiroshi Shimoda, Yasuyoshi Taruta, Yuya Kouda: Development of AR-based scanning support system for 3D model reconstruction of work sites, *Journal of Nuclear Science and Technology*, **59**(7), pp.934–948 (2022).
- [23] Yuxin Zhao, Hirotake Ishii, Hiroshi Shimoda: Development of a Camera Motion Estimation Method Utilizing Motion Blur in Images, *HCI International 2023 – Late Breaking Posters*, pp.125–131 (2024).
- [24] Masahiro Murayama, Toyohiro Higashiyama, Yuki Harazono, Hirotake Ishii, Hiroshi Shimoda, Shinobu Okido, Yasuyoshi Taruta: Depth Image Noise Reduction and Super-Resolution by Pixel-Wise Multi-Frame Fusion, *IEICE Transactions on Information and Systems*, **E105.D**(6), pp.1211–1224 (2022).
- [25] Intel: インテル®RealSense™テクノロジー.
<https://www.intel.co.jp/content/www/jp/ja/architecture-and-technology/realsense-overview.html> (Retrieved January 9, 2025).
- [26] Xuan Zhang, Zheng Fang, Zhengda Lu, Jun Xiao, Xiaolong Cheng, Xiaopeng Zhang: 3D Reconstruction of Weak Feature Indoor Scenes Based on Hector SLAM

and Floorplan Generation, 2021 IEEE 7th International Conference on Virtual Reality, pp.117–126 (2021).

- [27] Xiang Gao, Shuhan Shen, Lingjie Zhu, Tianxin Shi, Zhiheng Wang, Zhanyi Hu: Complete Scene Reconstruction by Merging Images and Laser Scans, IEEE Transactions on Circuits and Systems for Video Technology, **30**(10), pp.3688–3701 (2020).
- [28] Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski: ORB: An efficient alternative to SIFT or SURF, 2011 International Conference on Computer Vision, pp.2564–2571 (2011).
- [29] David G. Lowe: Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, **60**(2), pp.91–110 (2004).
- [30] Herbert Bay, Tinne Tuytelaars, Luc Van Gool: SURF: Speeded Up Robust Features, Computer Vision – ECCV 2006, pp.404–417 (2006).
- [31] Unity Asset Store: HQ Suburban House.
<https://assetstore.unity.com/packages/3d/environments/urban/hq-suburban-house-81890> (Retrieved January 16, 2025).
- [32] Intel: Depth Camera D455 - Intel RealSense.
<https://www.intelrealsense.com/depth-camera-d455/> (Retrieved January 16, 2025).

付録 A 支柱の質量を考慮した運動方程式の導出

本章では第 5.5.1 項で述べた運動方程式の導出方法について述べる。

ここでは、図 A.1 に示すように長さ L 、質量 M の一様な棒と質量 m の重りで構成される振り子を考える。このとき、振り子の振れ角を θ とすると、運動方程式は式 (A.1) と書ける。ただし、 I は棒と重りを合わせた支点まわりの慣性モーメント、 $\tau(\theta)$ は支点回りの重力によるトルクを表す。

$$I \frac{d^2\theta}{dt^2} + \tau(\theta) = 0 \quad (\text{A.1})$$

次に、支点回りの慣性モーメントについて考える。質量 M の一様な棒の端を固定し、支点としたとき、支点回りの慣性モーメント I_{rod} は式 (A.2) と表される。

$$I_{\text{rod}} = \frac{1}{3}ML^2 \quad (\text{A.2})$$

また、重りの慣性モーメント I_{mass} は、支点から距離 L の位置に質量 m の重りがあると考え、式 (A.3) と表される。

$$I_{\text{mass}} = mL^2 \quad (\text{A.3})$$

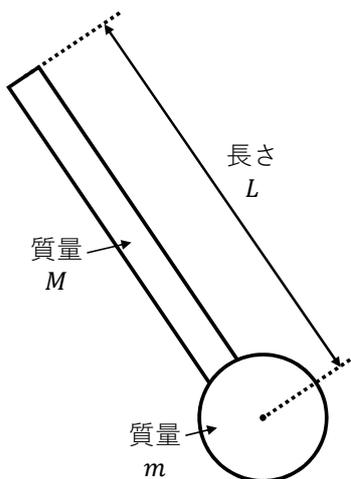


図 A.1: 仮定した振り子の図

従って、棒と重りを合わせた慣性モーメント I は式 (A.4) と表される。

$$I = I_{\text{rod}} + I_{\text{rod}} = \left(\frac{1}{3}M + m \right) L^2 \quad (\text{A.4})$$

続いて、重力によるトルクについて考える。重力によるトルクは支点回りの力と支点からの長さの積で表される。棒の質量は棒の重心にあり、棒の重心は支点から $\frac{L}{2}$ の位置にあると考えられる。よって棒によるトルク $\tau_{\text{rod}}(\theta)$ は式 (A.5) と表される。ただし、重力加速度を g とする。

$$\tau_{\text{rod}}(\theta) = \frac{MgL}{2} \sin \theta \quad (\text{A.5})$$

重りによるトルク $\tau_{\text{mass}}(\theta)$ は式 (A.6) と表される。

$$\tau_{\text{mass}}(\theta) = mgL \sin \theta \quad (\text{A.6})$$

よって合計のトルク $\tau(\theta)$ は式 (A.7) と表される。

$$\tau(\theta) = \tau_{\text{rod}}(\theta) + \tau_{\text{mass}}(\theta) = gL \left(\frac{1}{2}M + m \right) \sin \theta \quad (\text{A.7})$$

以上より、運動方程式は式 (A.8) と表される。

$$\left(\frac{1}{3}M + m \right) L^2 \frac{d^2\theta}{dt^2} + gL \left(\frac{1}{2}M + m \right) \sin \theta = 0 \quad (\text{A.8})$$

これを变形して式 (A.9) を得る。

$$\frac{d^2\theta}{dt^2} = -\frac{\frac{1}{2}M + m}{\frac{1}{3}M + m} \frac{g}{L} \sin(\theta) \quad (\text{A.9})$$

最後に、振り子の振れ角 θ を振り子座標系に変換して運動方程式は式 (A.10) のように表される。

$$\frac{d^2\theta}{dt^2} = -\frac{\frac{1}{2}M + m}{\frac{1}{3}M + m} \frac{g}{L} \sin(\theta - \theta_{\text{up}}) \quad (\text{A.10})$$