

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： 線分特徴を用いた絞り込み処理による  
リローカリゼーション手法の開発

指導教員： 下田 宏 教授

氏名： 徳丸 博紀

提出年月日： 平成28年2月10日(水)

## 論文要旨

題目：線分特徴を用いた絞り込み処理によるリローカリゼーション手法の開発

下田宏研究室， 徳丸博紀

要旨：

原子力発電プラントの保守・解体作業を支援するために拡張現実感 (Augmented Reality : AR) 技術を用いる研究がなされている。AR 技術を解体作業の支援に応用することで、効率的に解体作業を行えるようになる可能性がある。AR を実現するためには、カメラの位置・方向をリアルタイムで計測するトラッキングと呼ばれる技術が必要である。プラント内部のような広域で複雑な形状の物体が多く存在する環境下では、マーカレスでトラッキングを行えるようになることが望ましい。マーカレスで安定したトラッキングを実現するためには、現在のカメラ画像のみを用いてカメラの位置・方向を計測するリローカリゼーションと呼ばれる手法が必要である。これまで、様々なリローカリゼーション手法が提案されているが、プラント内部で安定かつ高速に利用可能な手法は開発されていない。

一方、プラント内部には、配管のエッジなど線分が多く存在する。この線分を利用すれば、リローカリゼーションの安定性や速度を向上させることができる可能性がある。本研究では、線分特徴を利用した絞り込み処理によるリローカリゼーション手法の開発と評価を目的とする。リローカリゼーションの安定性や速度を向上させることができれば、原子力発電プラント内部で AR を利用した作業の効率化が期待できる。

提案手法では、事前準備として画像から認識した線分の本数、線分同士がなす角度の分布、線分間の距離の分布などの情報をその画像を取得した時のカメラの位置・方向と組にして保存した線分情報データベースを作成する。リローカリゼーションを実行する際には、現在のカメラの入力画像から認識される線分特徴の情報と線分情報データベース内の情報を比較し、入力画像と類似した画像の候補をデータベース内から探索する。提案手法では最初に、計算負荷の小さい処理から順番に実行し類似画像の候補を段階的に絞り込み、最後まで残った画像を対象に計算負荷は高いが精度が高い処理を実行することで、類似画像の最終候補を 1 枚選出し、この画像と組になっていたカメラの位置・方向を現在のカメラの位置・方向とする。

提案手法では線分を認識するという処理が増えるが、候補画像を計算負荷の小さい処理で段階的に絞り込んでいき、最終的に候補として残った少数の画像のみを対象として計算負荷は高いが精度の高い画像の輝度値の差分を求める処理を行うことで、結果として処理時間が短く、精度の高いリローカリゼーションが実現できた。開発した提案手法の性能を画像からランダムに選んだピクセル同士で比較して類似画像を検索する Randomized Fern と呼ばれる手法の性能と比較した。性能を比較する際に使用した線分情報データベースは、新型転換炉ふげん発電所の純水装置室の内部を撮影した画像を使用した。性能の比較には、正答率と処理時間の 2 つの指標を用いた。その結果、正答率が同じとなる条件では、提案手法の方が、処理時間が短いという結果が得られた。

今後の課題としては、線分の認識を安定化させ、提案手法の性能を向上させることや、リローカリゼーションを実行する環境の特性に応じてパラメータを効率的に決定する方法を実現することなどが挙げられる。

# 目次

第 1 章 序論	1
第 2 章 研究の背景と目的	3
2.1 研究の背景	3
2.2 研究の目的と意義	11
第 3 章 線分特徴を用いた多段式スクリーニング法	14
3.1 既存のリローカリゼーション手法の概要	14
3.1.1 画像を用いたリローカリゼーション手法	14
3.1.2 正規化相互相関を用いたリローカリゼーション手法	15
3.1.3 Randomized Fern を用いたリローカリゼーション手法	16
3.2 提案手法の概要	22
3.3 データベースの構築	24
3.4 類似画像検索	38
第 4 章 提案手法の評価	46
4.1 評価の概要	46
4.2 提案手法のパラメータの決定	51
4.3 絞り込み処理の順番の決定	53
4.4 提案手法の性能の評価	56
4.5 評価結果の考察	59
4.5.1 提案手法のパラメータの考察	59
4.5.2 絞り込み処理の順番の考察	60
4.5.3 提案手法の性能の考察	62
第 5 章 結論	68
謝辞	70
参考文献	71



## 目 次

2.1	原子力発電プラント内部における AR 技術の使用例 <sup>[7][8]</sup> . . . . .	4
2.2	トラッキングの分類 . . . . .	4
2.3	新型転換炉ふげん発電所の純水装置室内部 . . . . .	6
2.4	人工マーカの例 . . . . .	7
2.5	環境に存在する点や線などの自然特徴の例 . . . . .	7
2.6	自然特徴を利用したトラッキングの枠組み <sup>[13]</sup> . . . . .	8
2.7	トラッキングとリローカリゼーションの関係 . . . . .	9
2.8	画像を用いたリローカリゼーション手法におけるデータベース . . . . .	10
2.9	異なる位置にある特徴点を同一特徴点として誤認識する例 . . . . .	12
3.1	画像を用いたリローカリゼーション手法の概要 . . . . .	14
3.2	画像を用いたリローカリゼーションにおける類似画像の探索 . . . . .	15
3.3	Randomized Fern における画像の符号化 . . . . .	17
3.4	Randomized Fern における画像間の類似度の計算 . . . . .	18
3.5	キーフレームのデータベースの作成 . . . . .	19
3.6	Randomized Fern のキーフレームのみのデータベースによる問題 . . . . .	21
3.7	提案手法のアルゴリズムの概要 . . . . .	22
3.8	原子力発電プラント内部で撮影した RGB 画像とデプス画像 . . . . .	23
3.9	提案手法の絞り込み処理による候補画像の絞り込み . . . . .	25
3.10	データベースの構築の処理の流れ . . . . .	26
3.11	RGB 画像のグレースケール画像への変換 . . . . .	27
3.12	原子力発電プラント内部で撮影した画像を対象に LSD を適用した結果 の例 . . . . .	27
3.13	線分の誤認識 . . . . .	28
3.14	デプス画像面の 1 点の実空間での位置 . . . . .	28
3.15	デプス画像面の 1 点の 3 次元空間上での X 座標 . . . . .	29
3.16	データベースに格納する線分総数法で必要な情報 . . . . .	31
3.17	画面上の位置による線分の本数の相違 . . . . .	31

3.18	画面分割法における画面の分割とデータベースの構成要素 (分割数を 6 とした場合) . . . . .	32
3.19	平行線法における線分の分類 . . . . .	33
3.20	水平・垂直な線分が数多く認識される場合の例 . . . . .	33
3.21	データベースに格納する平行線法で利用する情報 . . . . .	34
3.22	2D 距離法における距離の計算 . . . . .	35
3.23	2D 距離法におけるビンとデータベースに格納する情報 . . . . .	36
3.24	2D 長さ法におけるヒストグラム . . . . .	37
3.25	2D 角度法におけるヒストグラム . . . . .	37
3.26	事前準備で構築するデータベースに保存される情報 . . . . .	38
3.27	実際にリローカリゼーションを実行する際の処理の流れ . . . . .	39
3.28	線分の総数が少ない画像における線分の総数の差と差の割合 . . . . .	40
3.29	線分の総数が多い画像における線分の総数の差と差の割合 . . . . .	41
3.30	ヒストグラムの比較方法 . . . . .	43
4.1	候補画像データベースからの問題画像の抜き出し . . . . .	47
4.2	カメラの位置・方向は異なるが類似した画像が撮影される例 . . . . .	48
4.3	純水装置室内での撮影領域 . . . . .	50
4.4	画像の撮影に使用した Xtion PRO LIVE . . . . .	50
4.5	リローカリゼーションの処理時間 . . . . .	51
4.6	各処理段階の処理時間の平均 . . . . .	56
4.7	最適化後の処理の順番で実行した際の候補画像の組数の変化 . . . . .	57
4.8	最適化後の処理の順番で実行した際の候補画像の組数の変化の拡大図 . . . . .	58
4.9	2D 角度法における差の閾値と正答率の関係 . . . . .	60
4.10	線分の認識が不安定な場合の例 . . . . .	61
4.11	9 種類の絞り込み処理と処理時間の関係 . . . . .	63
4.12	提案手法の各絞り込み処理による候補画像の組数の変化 . . . . .	64
4.13	提案手法の各絞り込み処理による候補画像の組数の変化の拡大図 . . . . .	65
A.1	特徴点の追跡処理 <sup>[13]</sup> . . . . .	74
A.2	テンプレートマッチング . . . . .	75

## 表 目 次

4.1	Xtion PRO LIVE の仕様 <sup>[24]</sup> . . . . .	49
4.2	各絞り込み処理における正答率が最大になる閾値 . . . . .	53
4.3	各絞り込み処理におけるパラメータ . . . . .	54
4.4	絞り込み処理の順番の決定に使用した PC の仕様と使用した開発環境 . . . . .	54
4.5	各絞り込み処理の平均順位 . . . . .	55
4.6	最適化後の処理の順番で実行した時の候補画像組数 . . . . .	57
4.7	性能の評価に使用した PC の仕様 . . . . .	58
4.8	提案手法の性能の評価結果 . . . . .	58
4.9	Randomized Fern の性能の評価結果 . . . . .	59
4.10	9 種類の絞り込み処理と処理時間の平均 . . . . .	62
4.11	提案手法の各絞り込み処理後の候補画像の組数の平均 . . . . .	63
4.12	提案手法の各処理毎の処理時間 . . . . .	67

# 第 1 章 序論

2011年3月11日に発生した東日本大震災による東京電力福島第一原子力発電所の事故以降、日本において廃炉になる原子力発電所が増加している<sup>[1]</sup>。運転を終了した原子力発電プラントでは放射性物質が残存しているため、解体作業は通常のプラントよりもさらに安全かつ効率的に実施する必要がある。一方、原子力発電プラントの保守・解体作業を支援するために拡張現実感 (Augmented Reality : AR) 技術を用いる研究が行われている<sup>[2]</sup>。ARは、通常では見ることのできない情報をユーザに知覚させたり、現実世界の位置や方向をユーザが直感的に理解しやすいように提示できるなどの特長を有している<sup>[3]</sup>。このような特長を有するARを専門的な知識と技術が必要な解体作業の支援に用いることで、効率的に解体作業を行えるようになる可能性がある。

ARを実現するためには、現実世界の映像を撮影するカメラの位置・方向を計算するトラッキングと呼ばれる技術が必要である。原子力発電プラント内部のような広域で複雑な形状の物体が数多く存在する環境下では、カメラを利用したビジョンベースの手法を用いることが望ましい。ビジョンベースの手法には、マーカを利用したものがあるが、マーカを事前に設置する必要があり、準備作業に手間がかかる。また、作業時にマーカが邪魔になるという問題もある。そのため、事前にマーカを設置する必要のない自然特徴を利用した手法が有効であると考えられる。自然特徴を利用したトラッキングでは、多くの場合、現在と直前の連続する画像間のカメラの位置・方向の変化は小さいと仮定し、直前の画像におけるカメラの位置・方向を初期値として最適化問題を解くことで、トラッキングの安定性と計算速度の向上を図っている。しかし、カメラが急峻に動きカメラの位置・方向が大きく変化した場合などは、直前のカメラの位置・方向を初期値としたトラッキングを行うことができなくなる。この時、現在の画像のみを利用してカメラの位置・方向を計算する必要がある。この時に使用される方法をリローカリゼーション手法と呼ぶ。

原子力発電プラント内部は広域で複雑な形状の物体が多数存在しており、そのような環境で十分高速かつ高精度にリローカリゼーションできる手法はまだ開発されていない。一方、原子力発電プラント内部に多く存在する配管のエッジなどの線分を認識する手法を用いれば、リローカリゼーションの安定性を向上させられる可能性がある。また、原子力発電プラント内部は広域であるため、リローカリゼーションを実行する



際に使用する情報の量が膨大になる。リローカリゼーション実行時に、情報量を絞り込むことができれば、計算負荷が小さくなり、処理速度が向上すると考えられる。そこで本研究では、線分特徴を利用した絞り込み処理によるリローカリゼーション手法の開発と評価を目的とする。リローカリゼーションの速度や安定性を向上させられれば、原子力発電プラント内部での AR を利用した作業の効率化が期待できる。

本論文は、第1章の序論を含め、全5章で構成されている。第2章では、ARを実現するために必要なトラッキングとリローカリゼーションについて述べた後に、本研究の目的と意義を述べる。第3章では、既存のリローカリゼーション手法と本研究で提案する線分特徴を利用したリローカリゼーション手法のアルゴリズムについて説明する。第4章では、提案手法の評価について述べる。第5章では、本研究の結果をまとめ、今後の課題を述べる。

## 第 2 章 研究の背景と目的

本章ではまず、本研究の背景として、ARの概要、ARを実現するために必要なトラッキング手法、およびローカリゼーション手法について述べた後、本研究の目的と意義について述べる。

### 2.1 研究の背景

AR技術とは、コンピュータを用いて人の視覚や聴覚を拡張することにより、現実の世界を拡張する技術のことを指す。例えば、現実世界の映像にComputer Graphics(CG)を重畳表示すれば、実在しない物が、あたかも目の前に存在するかのよう感じさせることができる。ARを用いることでユーザの知覚を拡張、減衰、変化させることができ<sup>[5][6]</sup>、プラントの保守・点検・解体作業の支援に応用することにより、作業効率や安全性を向上させることができると期待されている。原子力発電プラントにおいてAR技術を利用した例<sup>[7][8]</sup>を図2.1に示す。図2.1(a)は、ARを用いて原子力発電プラントの解体作業を支援するアプリケーションを示している。実際の解体作業の現場で、解体対象物をどのように解体していくのかなどのシミュレーションを行うことができるシステムである。小型のタブレット端末の画面上に現実世界の映像を映し出し、その上に解体対象物のCGを重ねて表示している。図2.1(b)は、原子力発電プラント内部の映像に、放射線量の分布を重畳表示し、放射線量が高く危険なエリアを現場で視覚的に認知することができるアプリケーションの例である。このようなAR技術を実現するためには、ユーザの視野を撮影するカメラの位置・方向をリアルタイムで計算するトラッキングと呼ばれる技術が必要である。図2.2に示すように、トラッキングには、ジャイロセンサや加速度センサなどの慣性センサを用いたもの、全地球測位システム(Global Positioning System : GPS)を用いたもの、磁気センサや超音波センサを用いたもの、カメラ等のビジョンセンサを用いたものがある<sup>[9][10]</sup>。しかし、図2.3に示す実際の原子力発電プラント内部のような、複雑な形状の物体が多数存在している広大な屋内環境では、利用できるトラッキング技術に限られる。ジャイロセンサ・加速度センサを用いたトラッキングの場合、時間積分による誤差が蓄積していくため、ARによる原子力発電プラントの保守・解体作業の支援のような長時間の使用が想定される

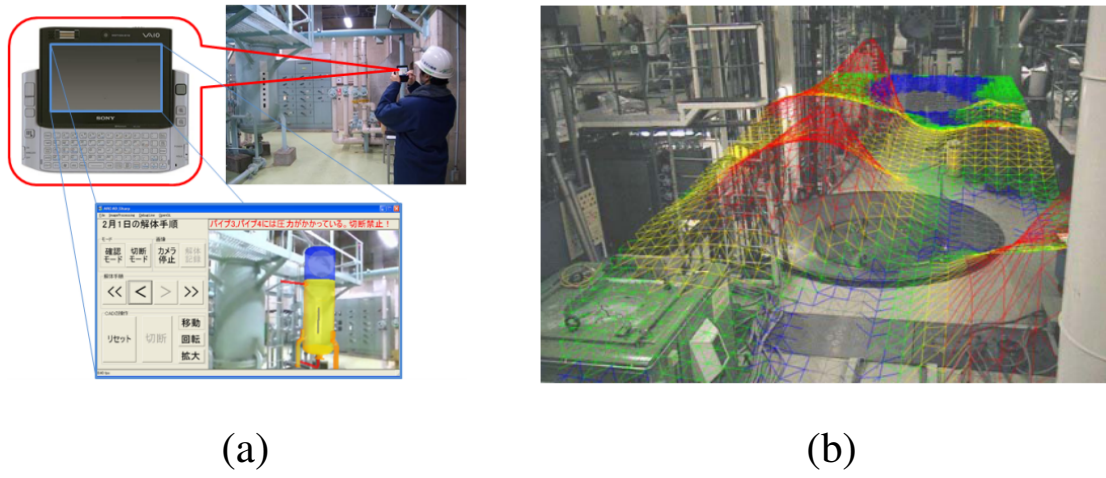


図 2.1: 原子力発電プラント内部における AR 技術の使用例 [7][8]

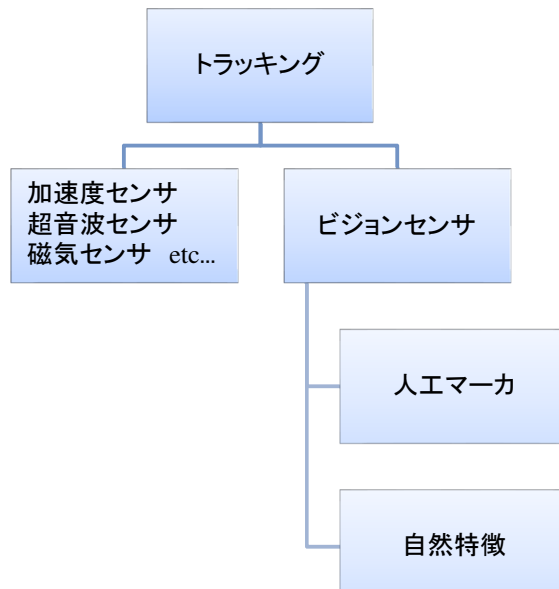
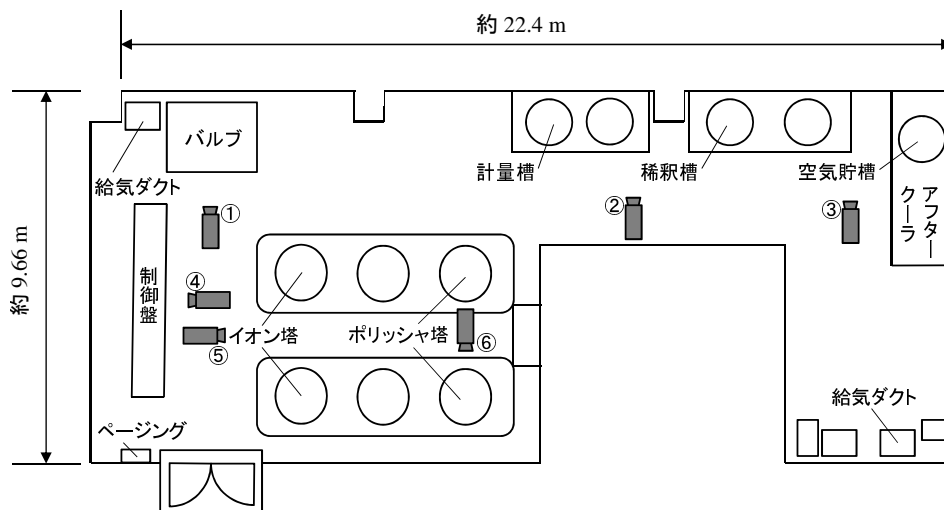


図 2.2: トラッキングの分類

場面では不向きである。また、GPSの場合、原子力発電プラント内部は人工衛星からの電波を十分に受信できず使用できない。また、磁気センサは、原子力発電プラント内部のような金属でできた物体の多い場所では使用できず、超音波センサを用いたトラッキングは、広域で複雑な形状の物体が多数存在する原子力発電プラント内部では正確な反射波を得ることができず、精度が不十分となる。原子力発電プラント内部では以上のような各種制限により、使用できるセンサが限られる。一方、ビジョンセンサは原子力発電プラント内部でも使用可能である。そこで、原子力発電プラント内部ではビジョンベースの手法が有効であると考えられる。ビジョンベースの手法は人工マーカを用いたトラッキングと、自然特徴を利用したトラッキングに大別できる。人工マーカを用いたトラッキングでは図2.4に示すような座標が既知である人工マーカをカメラで撮影し、画像処理と幾何計算によりマーカを基準としたカメラの位置・方向を求める。人工マーカを用いた手法では、広域な原子力発電プラント内部で使用する場合、大量のマーカを設置する必要があり、準備作業に手間がかかる。実際にARを利用して、原子力発電プラント内部での作業の効率化を図る場合、作業の手間が少ない方が良い。以上より、事前準備の手間の少ない自然特徴を利用したトラッキングを用いるのが望ましい。

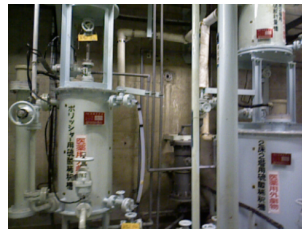
自然特徴を利用したトラッキングでは、環境に存在する点や線などの自然特徴を認識し、これらを利用してカメラの位置・方向を計算する<sup>[11][12]</sup>。環境中で認識される自然特徴の例を図2.5に示す。図2.5の例では、原子力発電プラント内部に存在するバルブを支えているフレームの角や側面部分が、それぞれ自然特徴点や線分特徴として認識されている。自然特徴を利用したトラッキングの枠組みを図2.6に示す。図2.6のように、自然特徴を利用したトラッキングでは、環境中に存在する自然特徴に関する情報を保存したデータベースを作成し、環境を撮影した画像から認識される自然特徴とデータベースに保存されている自然特徴を対応付けることで、カメラの位置・方向を計算する。一般的にはまず、現在の画像を撮影したときのカメラの位置・方向は直前の画像を撮影したときのカメラの位置・方向に近いと仮定し、データベース内に保存された自然特徴を現在のカメラの画像上に投影した際の2次元座標 $p$ を求める。この $p$ の周辺で、外見が最も似た自然特徴 $p'$ を探索し、対応する自然特徴とする。そして、対応づけられた $p$ と $p'$ の間の距離の差を最小化する最適化問題を解くことにより、現在のカメラの位置・方向を求める。以上の処理を、自然特徴を利用したトラッキングでは、時系列に並んだ連続画像に対して繰り返し行う。連続画像に対する画像間の対応付けを行うことによりトラッキングを継続させる処理を追跡処理という。追跡処理



①バルブ



②計量槽



③稀釈槽・空気貯槽



④制御盤



⑤イオン塔



⑥ポリッシャ塔周辺

図 2.3: 新型転換炉ふげん発電所の純水装置室内部



図 2.4: 人工マーカの例

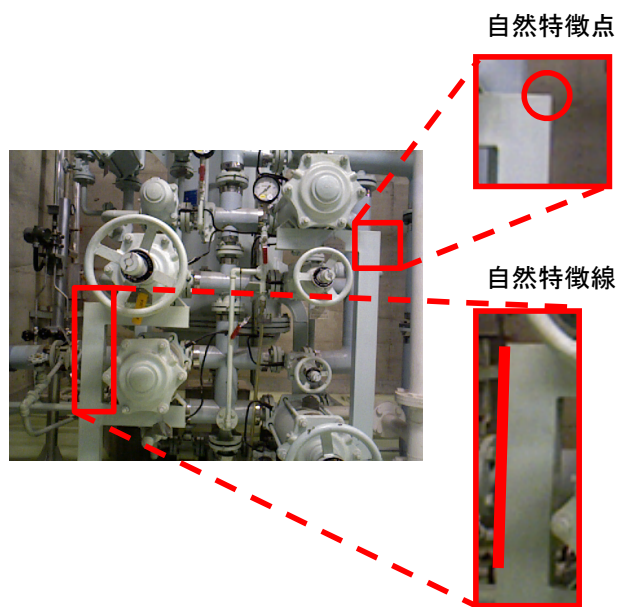


図 2.5: 環境に存在する点や線などの自然特徴の例

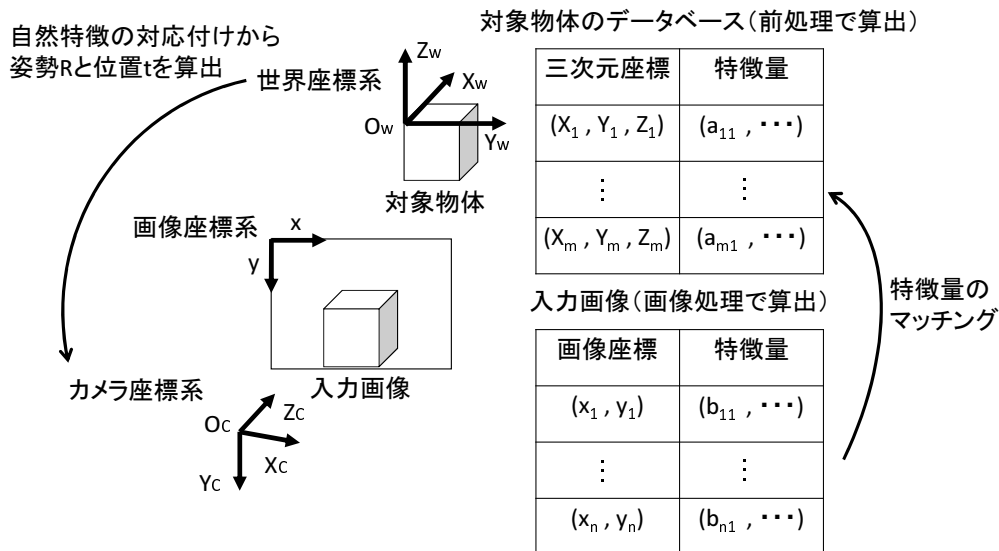


図 2.6: 自然特徴を利用したトラッキングの枠組み<sup>[13]</sup>

を行うことで、計算負荷が軽減し高速で継続したトラッキングが可能となる。追跡処理の詳細を付録 A に示す。

しかし、カメラが急峻に動くなどした場合、最適化問題を解いて算出されたカメラの位置・方向が、実際のカメラの位置・方向と大きく異なる可能性がある。図 2.7 に示すように、最小化された誤差の値が、予め定めた閾値よりも大きくなった場合には、トラッキングが失敗したと判断する。トラッキングが失敗した場合に、現在の画像のみの情報を利用してカメラの位置・方向を推定することをリローカリゼーションという。リローカリゼーションには、様々な手法が提案されているが<sup>[14]</sup>、よく利用されているのは、トラッキングに成功している間に、カメラ画像とそのカメラ画像を取得した際のカメラの位置・方向の組をデータベース内に格納しておきリローカリゼーションの際に利用する、画像を用いたリローカリゼーション手法である。図 2.8 に示すように、カメラで環境を撮影し、その時のカメラの位置・方向を表す並進ベクトル (Translation vector :  $T$ ) と回転行列 (Rotation matrix :  $R$ )、画像から認識される自然特徴、縮小した画像などを組にしてデータベースに格納しておく。トラッキングに失敗したときに現在の入力画像とデータベースに格納した画像を比較し、入力画像と似ていると判断された画像がデータベースに存在すれば、現在のカメラの位置・方向は、その画像が

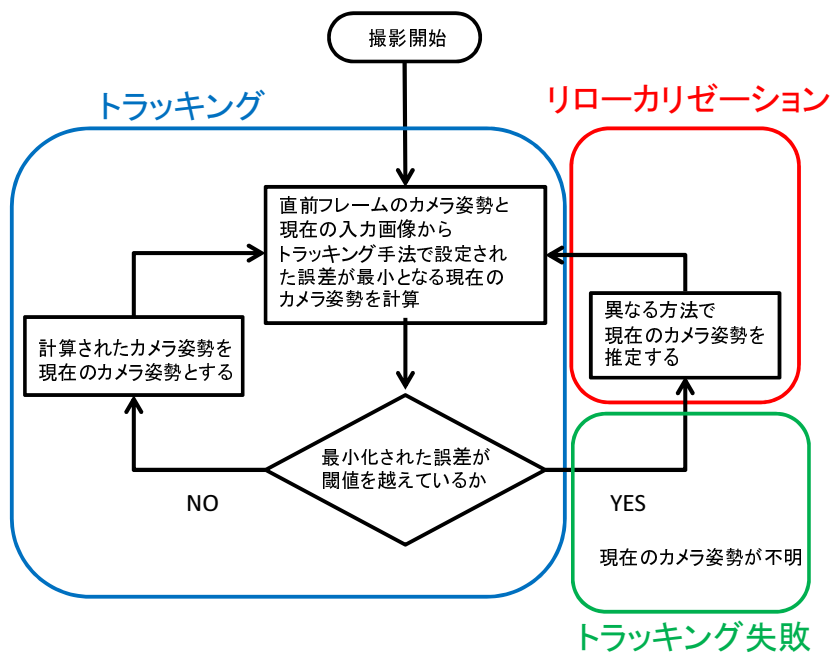


図 2.7: トラッキングとリローカリゼーションの関係



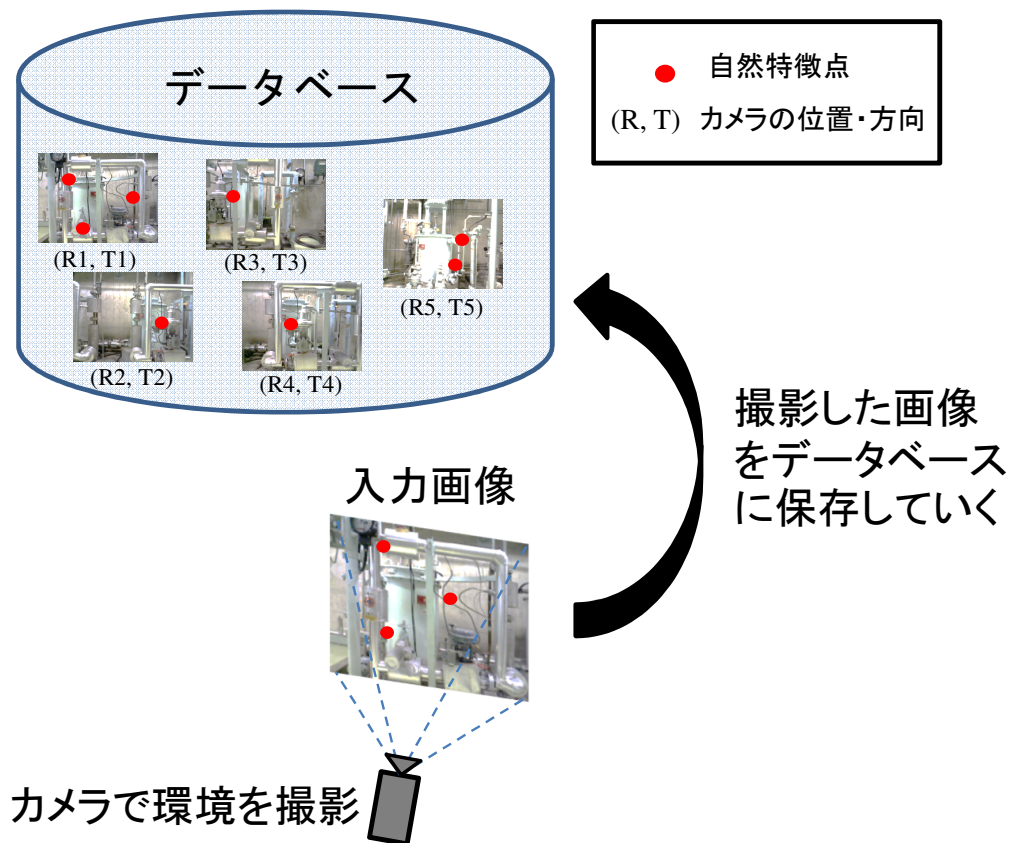


図 2.8: 画像を用いたリローカリゼーション手法におけるデータベース

撮影された時のカメラの位置・方向に近いと判断される。そして、似ていると判断されたデータベース内の画像を取得した際のカメラの位置・方向を初期値として誤差最小化の最適化問題を解くことにより再びトラッキングを試みる。原子力発電プラント内部は広域であるため事前に撮影する画像の枚数が多く、データベースに格納される情報も膨大になる。したがって、トラッキングに失敗した場合、入力画像と似た画像を膨大なデータベース内から探索する必要があり処理に時間がかかる。

そこで、原子力発電プラント内部でARによる効率的な作業支援を行うには、トラッキングに失敗しても短時間でトラッキングを再開できる安定したリローカリゼーションを実現する必要がある。全ての類似画像の候補に対して、精度は高いが計算負荷が高い画像全体の比較などを行う前に、画像の一部分のみや画像を代表する情報を用いる計算負荷が小さい処理を実行し候補画像を減らすことで、精度が高く処理時間が短いリローカリゼーションを実現できる可能性がある。

原子力発電プラント内部では、画像を代表する情報として自然特徴を利用することが有効であると考えられる。しかし、図 2.9 に示すように、プラント内部の配管が重なっている箇所などで、3次元空間上での位置が異なる特徴点であっても、外見が似ているため同一の特徴点と誤認識してしまう場合がある。このような場合、その特徴点の外見などの情報を用いて類似画像検索を行うと、実際とは異なるカメラの位置・方向と組になってデータベースに保存されている画像が選出され、結果としてリローカリゼーションに失敗してしまう可能性がある。しかし、類似画像検索の精度を上げるため、特徴点の外見を比較する領域を広げると、結果として精度は高いが処理時間が長いリローカリゼーション手法となる。

## 2.2 研究の目的と意義

自然特徴の外見だけではなく、原子力発電プラント内部に多く存在する配管などの人工物のエッジなどの線分特徴を利用すれば、精度が高く処理時間が短いリローカリゼーションが実現できる可能性がある。本研究では、線分特徴を利用した絞り込み処理によるリローカリゼーション手法の開発と評価を目的とする。提案手法では、類似画像の候補を絞り込む際に、計算負荷が小さい処理から順に適用することにより、段階的に候補画像を絞り込む。先に計算負荷がより小さい処理を実行し、後になるにつれて計算負荷が高く精度が高い処理を実行することで、結果として処理時間が短く精度が高い手法の実現が期待できる。本研究の目的が達成され、精度が高く処理時間の

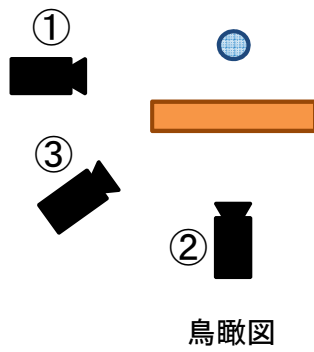
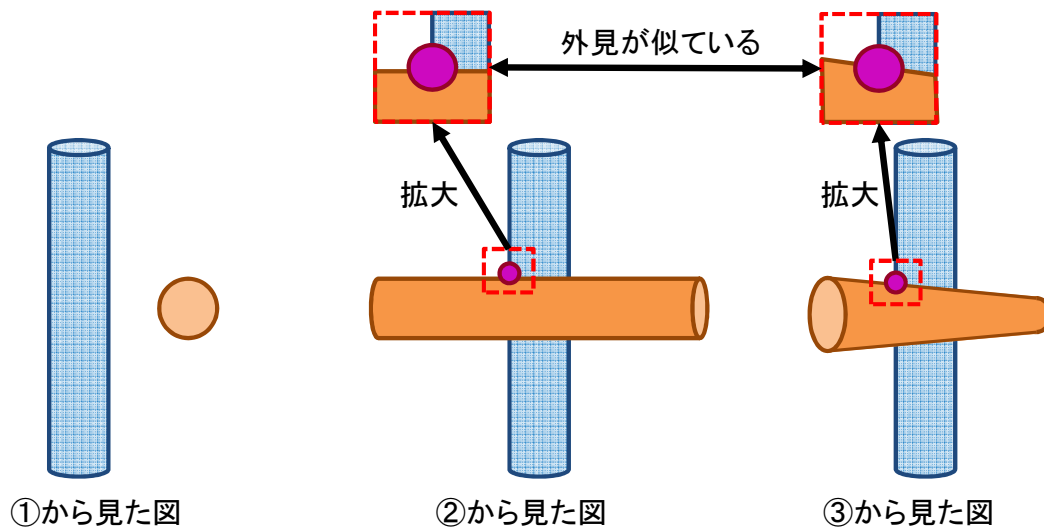


図 2.9: 異なる位置にある特徴点を同一特徴点として誤認識する例

短いリローカーゼーションを実現できれば、トラッキングに失敗した場合でも短時間でトラッキングを再開させることができ、ARを用いた作業支援システムが使いやすくなり、結果として原子力発電プラント内部の保守・点検・解体作業の安全性や効率を向上させることができると期待される。

# 第 3 章 線分特徴を用いた多段式スクリーニング法

本章では最初に既存のリローカリゼーション手法の概要について述べる。次に、提案手法の概要と課題について述べる。最後に、提案手法のアルゴリズムの詳細について述べる。

## 3.1 既存のリローカリゼーション手法の概要

### 3.1.1 画像を用いたリローカリゼーション手法

画像を用いたリローカリゼーション手法の典型的な処理の流れを図 3.1 に示す。図 2.8 に示したように、画像を用いたリローカリゼーションでは、事前もしくはトラッキングが成功している際に撮影した画像と、その画像を撮影した時のカメラの位置・方向を組にしてデータベースに保存しておく。トラッキングに失敗し、リローカリゼーションを行う際には、図 3.2 に示すように、現在のカメラからの入力画像とデータベース内の画像の類似度を計算する (以下、データベース内の画像を候補画像と呼ぶ)。画

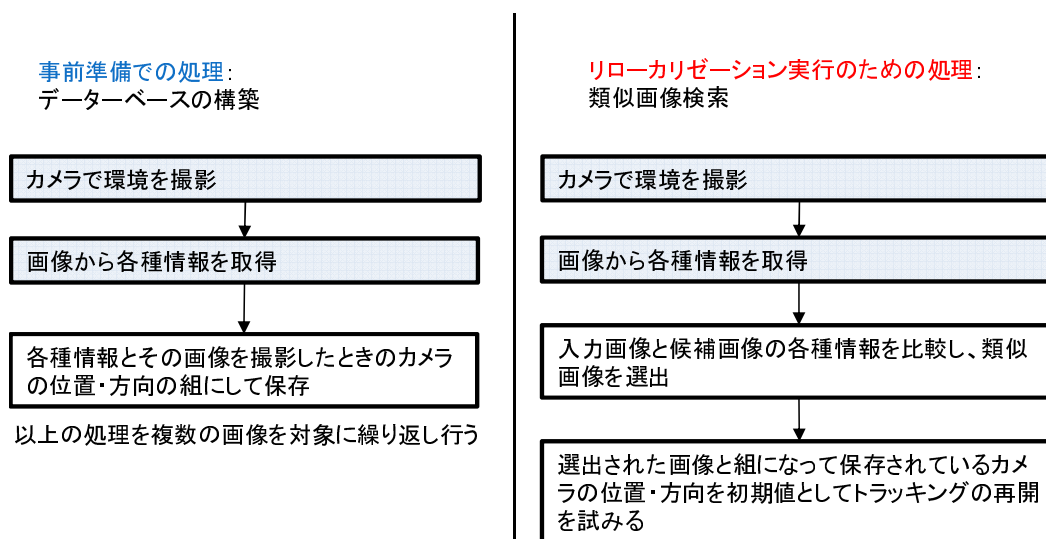


図 3.1: 画像を用いたリローカリゼーション手法の概要

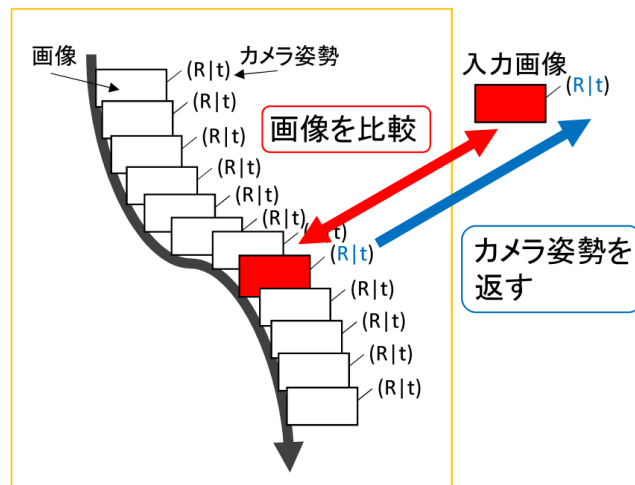


図 3.2: 画像を用いたリローカリゼーションにおける類似画像の探索

像の類似度が、設定した閾値以上であれば、入力画像と候補画像が類似していると判断される。この際、閾値以上の類似度の画像がデータベース内に見つからなかった場合には、現在の入力画像を用いたリローカリゼーションは失敗したものとし、次の入力画像を対象に同じ処理を繰り返す。一方、閾値以上の類似度の画像が見つかった場合には、それらの画像の中で最も類似度が高い画像を1枚、もしくは上位複数枚を選出する。ここで、類似度が高い画像同士は、撮影時のカメラの位置・方向が互いに近い可能性が高いと仮定し、選出した画像と組になっていたカメラの位置・方向を利用して、トラッキングの再開を試みる(類似画像を複数枚選出している場合には、それら全てを対象にトラッキングの再開を試みる)。

しかし、この方法では候補画像が撮影された位置・方向にのみ復帰することが可能であるため、広い範囲を対象にリローカリゼーションすることを可能にするためには、それらの領域全体を抜け落ちなく撮影し、その時のカメラの位置・方向を求めておく必要がある。

### 3.1.2 正規化相互相関を用いたリローカリゼーション手法

画像の全てのピクセルを用いて類似度を計算する方法としては、Sum of Squared Difference(SSD: ユークリッド距離二乗和)、Sum of Absolute Difference(SAD: 市街地距離和)、Normalized Cross-Correlation(NCC: 正規化相互相関)などがある<sup>[15][16]</sup>。正規化相互相関を用いたリローカリゼーション手法では、画像の類似度を正規化相互相

関で表し、入力画像と類似した画像をデータベースから選出する。正規化相互相関は、入力画像の輝度値の値を  $T(x', y')$ 、候補画像の輝度値の値を  $I(x + x', y + y')$  とすると式 3.1 で表される。ただし、 $(x, y)$  は入力画像の候補画像上での座標を表すとする。

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{(\sum_{x', y'} T(x', y')^2 \sum_{x', y'} I(x + x', y + y')^2)}} \quad (3.1)$$

式 3.1 では、正規化相互相関の値は 0 から 1.0 の値で表される。値が 0 に近いほど類似度が高く、1 に近い程類似度が低い。正規化相互相関を用いたリローカリゼーション手法は、精度は高いが処理時間が長いという問題がある。

### 3.1.3 Randomized Fern を用いたリローカリゼーション手法

Randomized Fern<sup>[17][18]</sup> を用いたリローカリゼーション手法では、画像の中からランダムに選んだピクセルの輝度値を、ランダムな閾値で二値化して符号化し、入力画像と候補画像の符号化された結果を比較する。画像全体のピクセル同士を比較する正規化相互相関を用いたリローカリゼーション手法などに比べて、画像の一部のピクセルの輝度値を二値化したコード同士の比較のみを行うため、計算量が少なく処理速度が速いという特徴がある。図 3.3 に示すように、符号化の際にはまず、画像からランダムに複数のピクセルを選ぶ。<sup>[17][18]</sup> で使用が推奨されているように、撮影対象物までの距離も同時に取得できる RGB-D カメラを用いた場合、画像中の各ピクセルは、Red(R)、Green(G)、Blue(B) の 3 つのチャンネルの輝度値に加えて、深度の値 Depth(D) を持つ。図 3.3 に示すように、RGBD の 4 つの各チャンネルのそれぞれの輝度値が、ランダムに決定された 4 つの閾値  $\tau_r, \tau_g, \tau_b, \tau_d$  により 0 か 1 に二値化される。1 つのピクセルを 4bits で符号化することになり、例えば、画像中からランダムに選ばれるピクセルの数を 500 個とすると、1 枚の画像は  $500 \times 4\text{bit}$  の二値化されたコードで表される。各画像 1 枚につき、この 2000bit のコードがデータベースに保存される。Randomized Fern では、この生成したコードを一定の bit 数毎に分けて扱う。この分けた後の bit の集まりを Fern と呼ぶ。分割する際の bit 数は適用する対象に応じて調整することが推奨されているが、多くの場合、1 ピクセル毎、すなわち 4bits 毎にコードを分ける。

図 3.4 に示すように、Randomized Fern では、2 枚の画像の類似度を、Blockwise Hamming Distance(BlockHD) で表す。BlockHD は、2 つの画像間での Fern を比較した際の、全体の Fern 数に対する互いに異なる値を取る Fern 数の割合であり、0 から 1 の値を取る。BlockHD の定義を式 3.2 に示す。ただし、 $m$  は Fern 数、 $q_{IJ}$  は画像 I と J で等しいコードを持つ Fern の個数を表す。

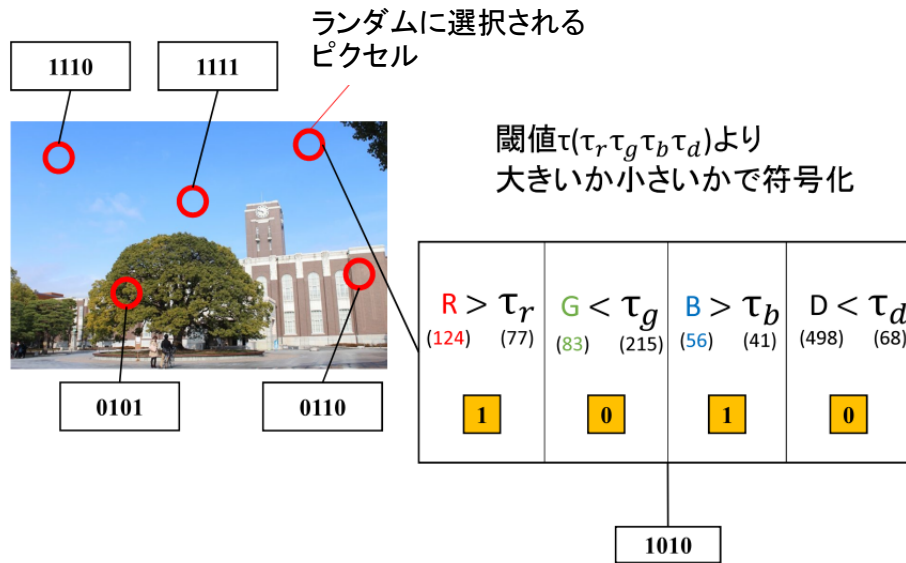


図 3.3: Randomized Fern における画像の符号化

$$BlockHD = \frac{m - q_{IJ}}{m} \quad (3.2)$$

図 3.4 の例の場合、等しいコードを持つ Fern の数は 4 個であるため、 $q_{IJ} = 4$  となる。また、全体の Fern の個数は 5 個であるため、 $m = 5$  となる。従って、図 3.4 における 2 枚の画像間の BlockHD は、

$$BlockHD = \frac{5 - 4}{5} = 0.2 \quad (3.3)$$

となる。BlockHD の値が 0 に近いほど類似度が高く、1 に近いほど類似度が低いと判断される。

次に、Randomized Fern におけるデータベースについて述べる。データベース作成の際には、入力画像と既にデータベースに格納されている画像との BlockHD を計算し、BlockHD の値が予め定めた閾値以上となる画像のみを、撮影時のカメラの位置・方向と組にしてデータベースに格納する。つまり、図 3.5 に示すように、データベース内の候補画像との類似度が低い画像のみがキーフレームとしてデータベースに格納され、類似度の高い画像はデータベースには格納されない。これは、データベースに保存される情報量を少なくし、リローカリゼーションを行う際の処理時間を短縮するためである。ただし、図 3.5 における KF は、キーフレーム (Key Frame : KF) を表している。

次に、Randomized Fern を用いて実際にリローカリゼーションを行う方法について述べる。リローカリゼーションの際には、入力画像とデータベースに格納されている



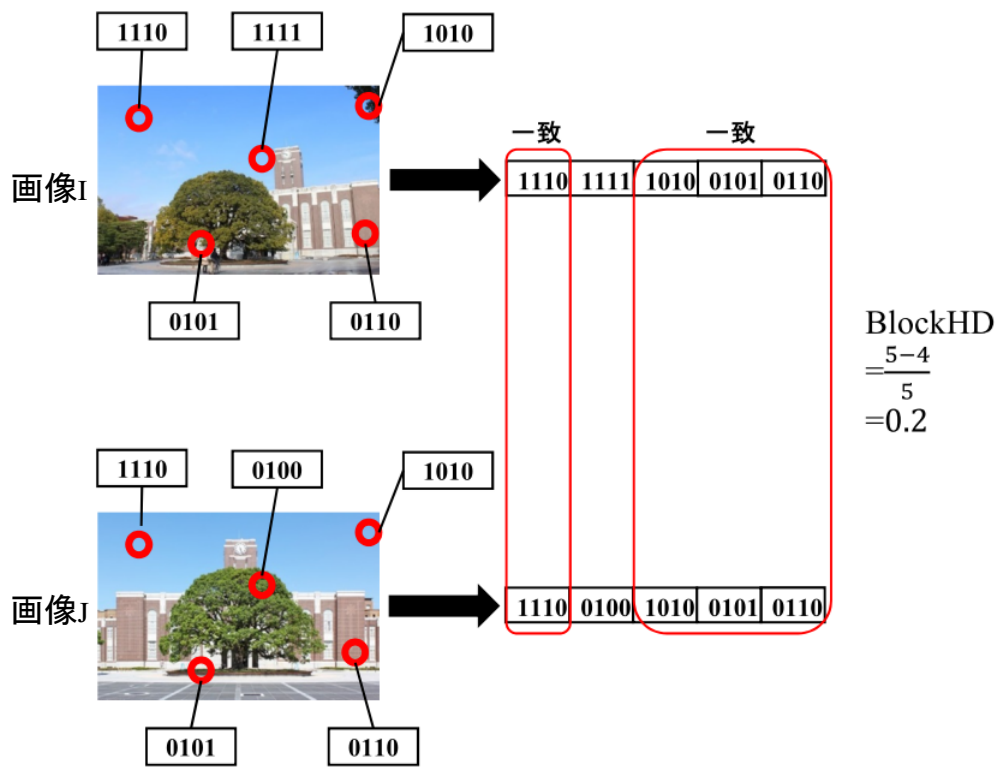


図 3.4: Randomized Fern における画像間の類似度の計算

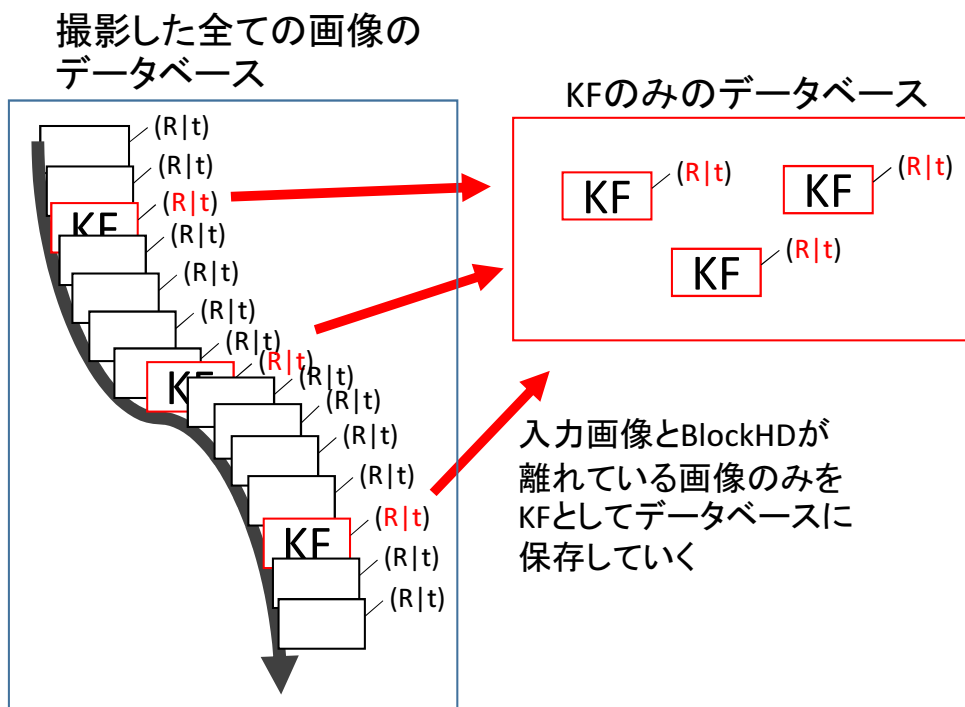


図 3.5: キーフレームのデータベースの作成

画像の BlockHD を計算し、類似度が高い画像上位 5 枚を選出する。上位 5 枚のうち、一番類似度の高い画像から順番に、その画像と組になって保存されているカメラの位置・方向を現在のカメラの位置・方向と仮定してトラッキングを試みる。トラッキングが成功すればリローカリゼーションが成功したことになる。トラッキングが失敗すれば、2 枚目の画像と組になって保存されているカメラの位置・方向をトラッキングの初期値として用いる。トラッキングが成功するまで、以上の処理を繰り返す。上位 5 枚のいずれの画像でもトラッキングが成功しなかった場合は、次の入力画像を対象に、上記と同様の処理を繰り返す。

次に、Randomized Fern を用いたリローカリゼーション手法の利点について述べる。この手法では、正規化相互相関を用いたリローカリゼーション手法のように、画像中の全てのピクセルを用いて類似度を計算するのではなく、ランダムに選ばれた一部のピクセルの情報のみを用いて類似度を計算するため、処理速度が速いという利点がある。また、先に述べたようにデータベースには BlockHD が予め定めた閾値以上の画像の情報のみが格納されているため、全ての画像に関する情報を格納したデータベースに比べて容量が小さく、より高速に類似画像を検索できる。また、画像中からランダムに選ばれたピクセルをコード化するには、ランダムに決定された閾値より大きい小さいかという条件のみで判断される。そのため、同じ箇所を撮影した場合でもピクセルの輝度値やデプスの値が異なる場合があるが、その場合でもコードは同じものになる可能性が高くなる。したがって、外乱の影響を受けにくく、過去に撮影した環境と同じ環境を撮影していれば、同じ環境を撮影していると認識できる可能性が高くなる。

しかし、Randomized Fern を用いたリローカリゼーション手法は、<sup>[17]</sup> に示されているパラメータのままでは、データベースに過去に撮影した画像のうち一部の画像の情報のみしか保存されない。そのため、カメラがデータベースに保存されていない位置・方向にある時は、リローカリゼーションによって得られるカメラの位置・方向と実際のカメラの位置・方向との差が大きくなり、精度が下がる。その様子を図 3.6 に示す。リローカリゼーションの精度を上げるにはには、Fern 数、BlockHD の閾値などのパラメータを<sup>[17]</sup> に示されている値から調整する必要がある。

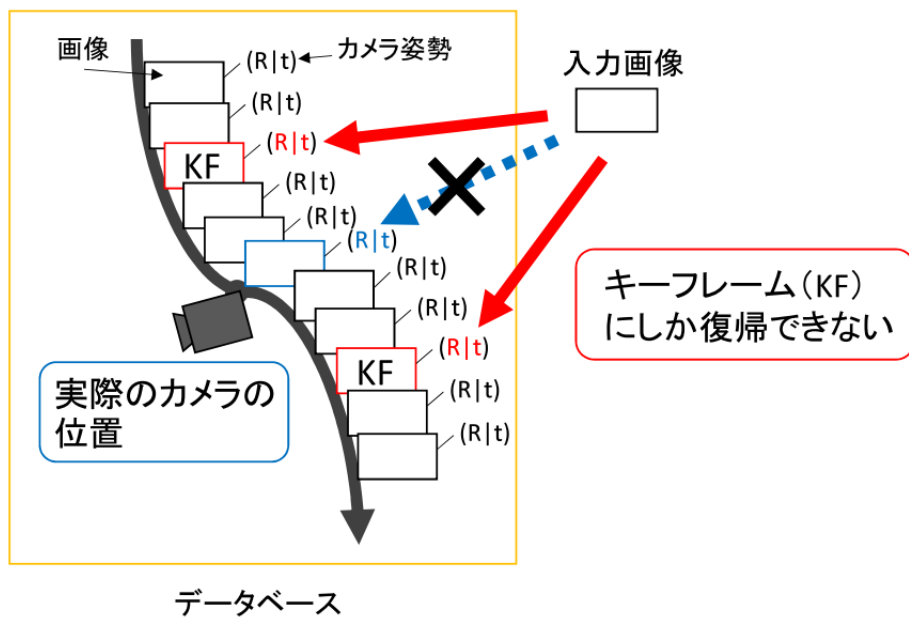


図 3.6: Randomized Fern のキーフレームのみのデータベースによる問題

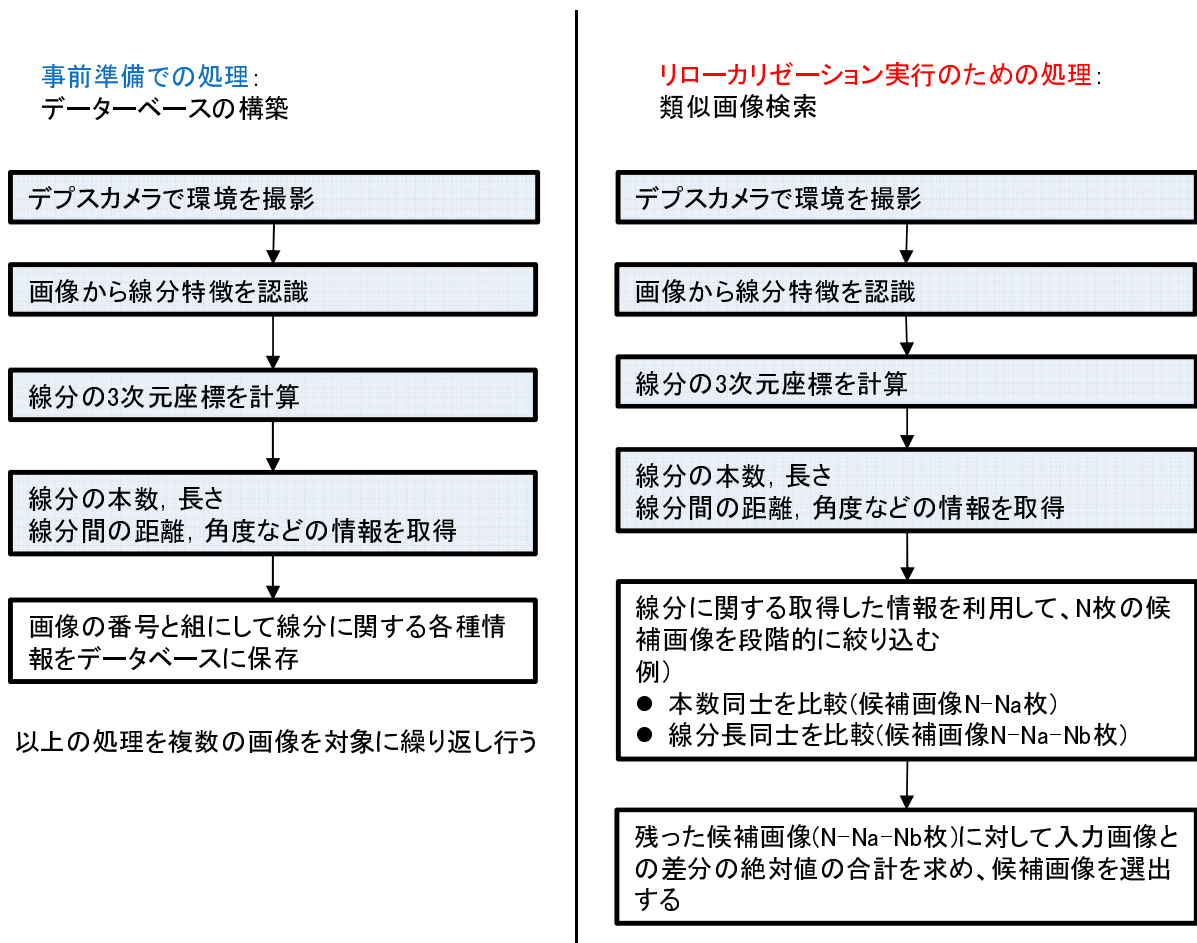


図 3.7: 提案手法のアルゴリズムの概要

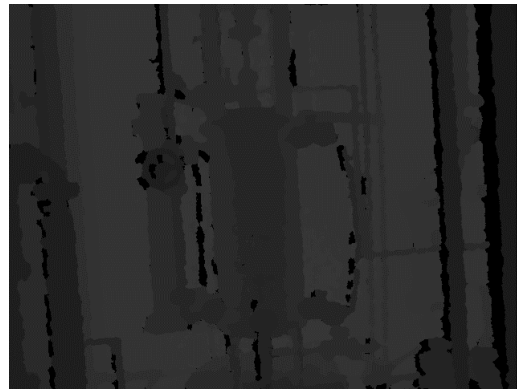
### 3.2 提案手法の概要

線分特徴を用いた多段式スクリーニング法では、線分特徴を利用して候補画像を絞り込む。本手法は、図 3.7 に示すように、事前準備としてデータベースを構築する処理と、データベースに保存された情報を利用して入力画像と類似した画像の候補を絞り込む処理に分かれている。類似画像を検索する処理では、最初に、入力画像で認識される線分特徴から得られる情報を利用して候補画像を絞り込む。この時、線分特徴に関する複数の情報を計算負荷が小さい順に利用し、段階的に候補画像を絞り込む。候補を絞り込んだ後に、計算負荷は高いが精度が高い処理を用いて正しい類似画像を求める。

事前準備の処理では、リローカリゼーション実行時の類似画像検索の処理に必要な各種情報を予め取得し、データベースに保存しておく。より多くの情報を利用可能にし



RGB画像



デプス画像

図 3.8: 原子力発電プラント内部で撮影した RGB 画像とデプス画像

て、処理制度を上げることを可能にするために、提案手法ではカラー画像と対象物までの距離が分かるデプス画像を撮影できる RGB-D カメラを使用する。最初に、RGB-D カメラで環境を撮影し、RGB 画像上で線分特徴を認識した後、デプス画像を用いて認識された線分特徴の 3 次元座標を求める。原子力発電プラント内部で撮影した RGB 画像とデプス画像の例を図 3.8 に示す。その後、線分の本数、長さ、直線間の距離、角度などの線分に関する各種情報を取得し、画像を取得した順番につけた画像の番号及び縮小画像とを組にしてデータベースに保存する。

リローカリゼーション実行時の処理では、現在の入力画像上で認識された線分の本数、長さ、直線間の距離、角度などの情報を取得し、データベースに保存した線分特徴の情報と比較し、候補画像を段階的に絞り込む。計算負荷が小さい情報を用いて候補画像を絞り込んだ後に、候補として残った画像のみを対象に、計算負荷は高いが、正確に類似画像を見つけることができる処理 (縮小画像を対象とした SAD による類似画像検索) を実施する。図 3.7 の右側に示した例では、 $N$  枚の候補画像を対象に、一段階目の絞り込み処理として線分の本数による絞り込みの処理を適用することで、 $N - N_a$  枚まで絞り込む。次に、 $N - N_a$  枚の候補画像を対象に、二段階目の絞り込み処理として線分の長さによる絞り込み処理を適用することで、 $N - N_a - N_b$  枚まで絞り込む。最後に、 $N - N_a - N_b$  枚の候補画像を対象に、入力画像の縮小画像と候補画像との SAD を求め、類似度の高い画像を選出する。この際、候補画像を  $N$  枚から  $N - N_a - N_b$  まで絞り込むのに必要な処理負荷が、 $N_a + N_b$  枚の候補画像の SAD を求める処理負荷よりも小さければ、結果として、全体の処理時間を短縮できる。

3.1.2 項で述べた正規化相互相関を用いたリローカリゼーション手法と比べ線分特徴を用いた多段式スクリーニング法では、画像上の線分を認識する処理が増えるが、線分の本数、長さ、距離、角度の比較は、正規相互相関を求める処理に比べ短時間で実行できるため、全ての候補画像に対して正規化相互相関処理を実施する場合と比べて検索の高速化が可能になると期待される。

候補画像の絞り込みに利用可能な線分情報としては、様々な情報が考えられるが、本研究では、処理負荷が比較的小さいと予想される、画面全体で認識されている線分の本数を比較する方法(線分総数法)、画面を複数の領域に分割し、領域毎に認識された線分の本数を比較する方法(画面分割法)、画像の x 軸、y 軸に平行な線分の本数を比較する方法(平行線法)、カメラ画像上での線分の長さの分布を比較する方法(2D 長さ法)、実空間での線分の長さの分布を比較する方法(3D 長さ法)、カメラ画像上での線分間の距離の分布を比較する方法(2D 距離法)、実空間での線分間の距離の分布を比較する方法(3D 距離法)、カメラ画像上での線分間の角度の分布を比較する方法(2D 角度法)、実空間での線分間の角度の分布を比較する方法(3D 角度法)の9種類の方法を実装する。図 3.9 の赤色の点線で囲まれた部分に示すように、9種類の絞り込み処理を順番に適用し、候補画像を絞り込んでいく。候補画像を絞り込んだ後、入力画像と候補画像の縮小画像の SAD を求めることにより類似画像を選出する。但し、スクリーニング処理を多段階とするにあたり、適用する絞り込み処理の順番によって、最終的な処理時間が変化する可能性がある。また、絞り込みが進んだ段階で余分な絞り込み処理を追加すると、返って処理時間が長くなる可能性もある。そこで、本研究では、原子力発電プラント内を撮影して得た画像を対象に、様々な順番・組み合わせで絞り込み処理を適用し、適切な処理の順番を検討する。この処理の順番の検討については、4.3 節で述べる。

### 3.3 データベースの構築

データベースを構築する際の処理の流れを図 3.10 に示す。始めに、トラッキングを行う対象の環境を、RGB-D カメラで撮影して RGB 画像とデプス画像を取得する。この時、カメラはゆっくりと動かし、後の処理で線分特徴を認識しやすくするために、ぼやけが少ない画像を撮影することが望ましい。次に、図 3.11 に示すように RGB 画像をグレースケール画像に変換する。次に、グレースケール画像に対して Line Segment Detector(LSD)<sup>[19][20][21][22]</sup> を適用し線分を認識する。原子力発電プラント内部で撮影し

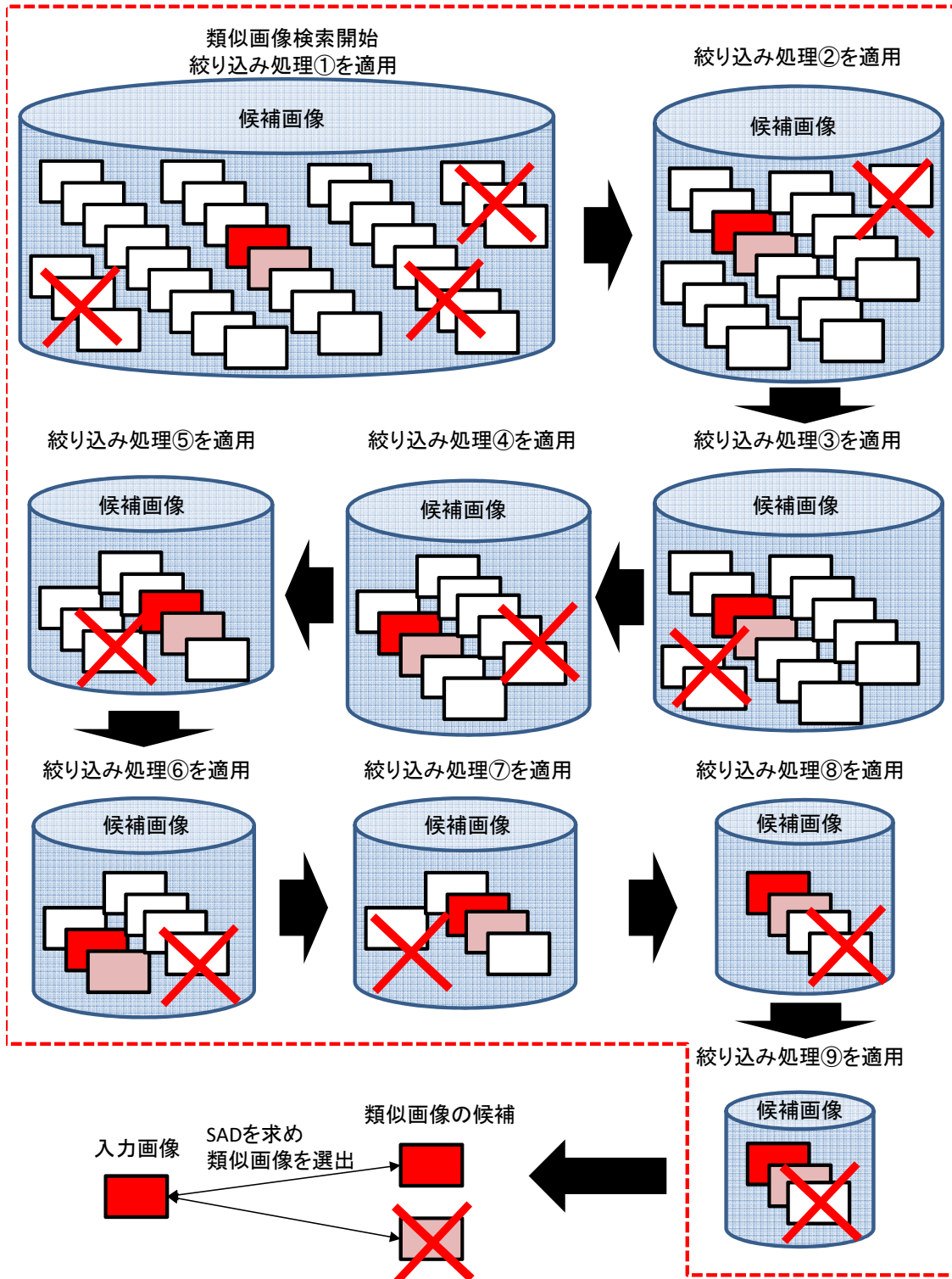


図 3.9: 提案手法の絞り込み処理による候補画像の絞り込み



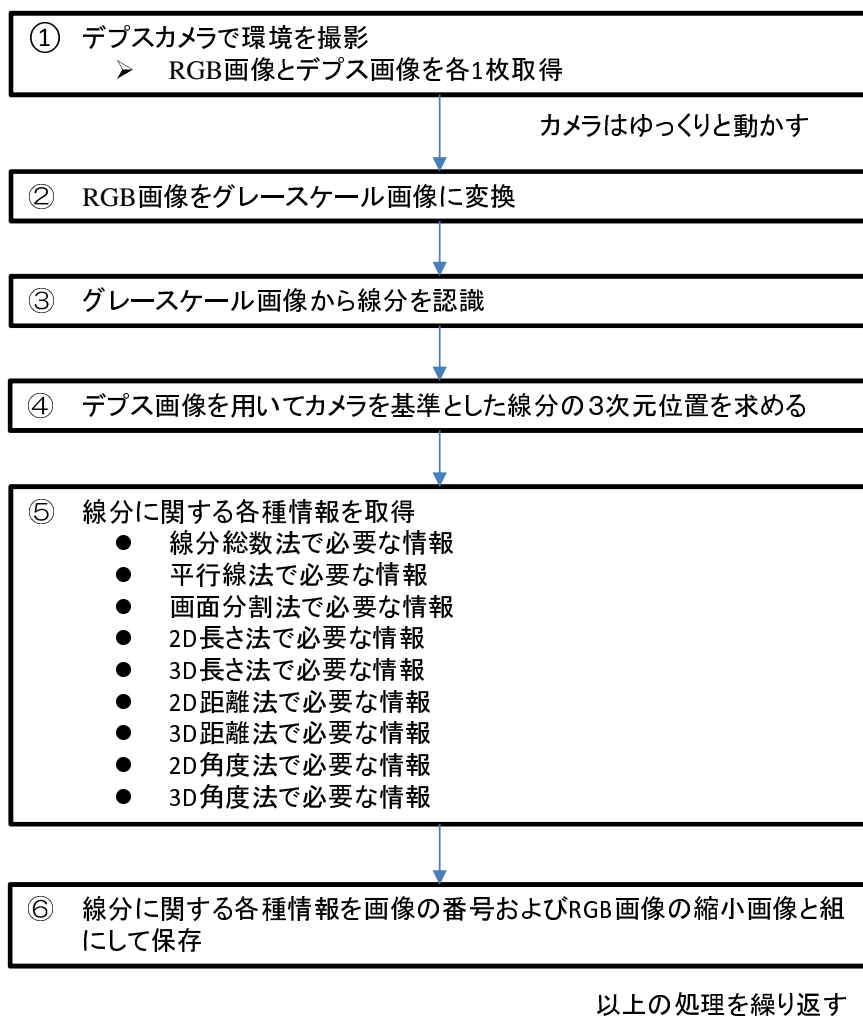


図 3.10: データベースの構築の処理の流れ

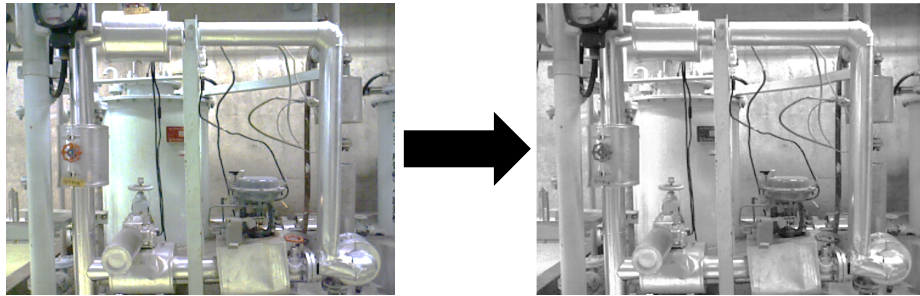


図 3.11: RGB 画像のグレースケール画像への変換

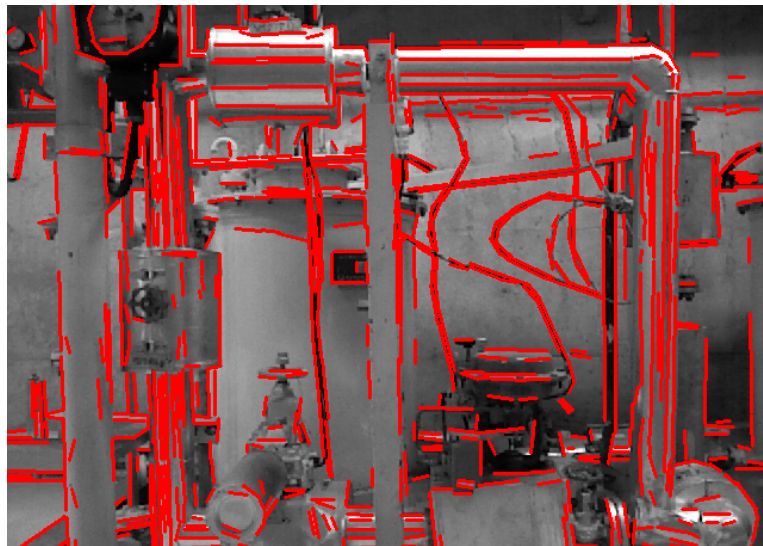


図 3.12: 原子力発電プラント内部で撮影した画像を対象に LSD を適用した結果の例

た画像を対象に LSD を適用した結果の例を図 3.12 に示す。

認識された線分のうち、配管が交差する箇所付近などで認識されたものは、図 3.13 に示すように、実空間上では繋がっていないにも係らず、2次元平面上で繋がっているかのように認識されてしまう場合がある。このような線分の誤認識を防ぐため、認識された線分を対象に Random Sample Consensus(RANSAC)<sup>[23]</sup> による線分認識安定化を試みる。具体的には、線分上の 2 点をランダムにサンプリングし、デプス画像を用いて、その 2 点のカメラの焦点を基準とした 3 次元座標を求める。図 3.14 に示すよう

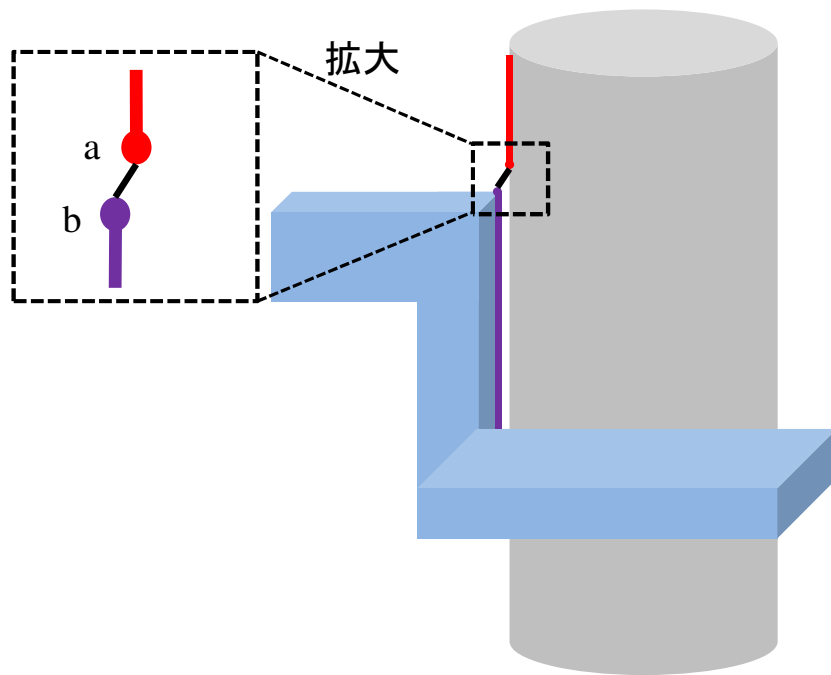


図 3.13: 線分の誤認識

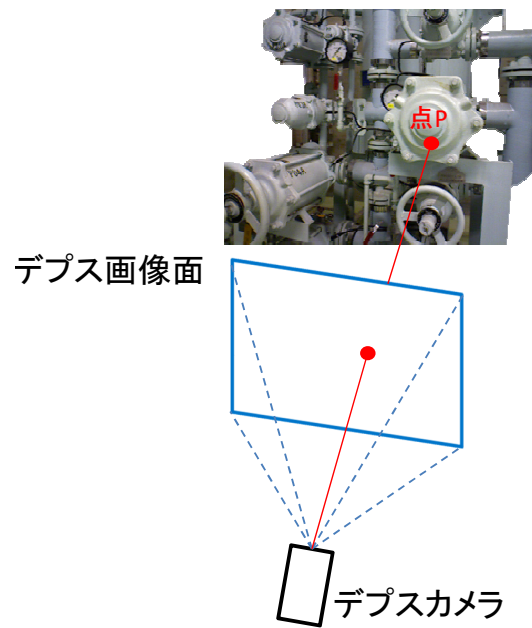


図 3.14: デプス画像面の 1 点の実空間での位置

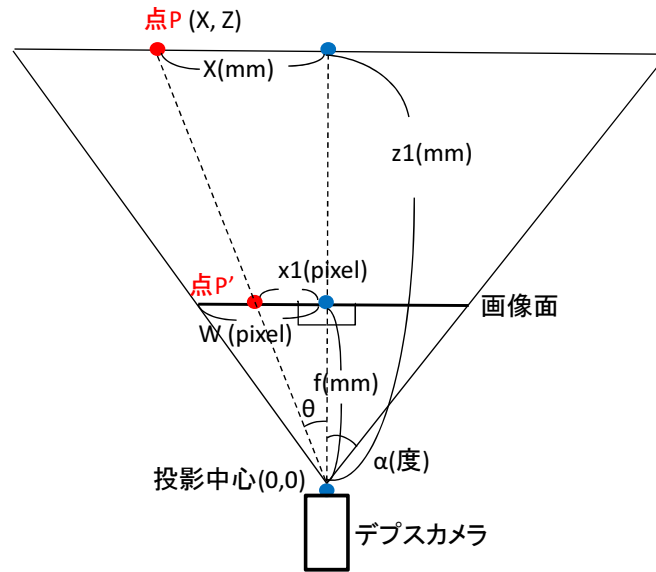


図 3.15: デプス画像面の 1 点の 3 次元空間上での X 座標

に、デプス画像の各ピクセルは、現実世界の物体の 1 点に対応している。RGB-D カメラの視野角と解像度、各ピクセルにおけるデプスの値 ( $z$  値) が分かっているならば、RGB-D カメラの焦点位置を基準とした 3 次元座標系において、線分上の点の水平方向 ( $X$  軸)、垂直方向 ( $Y$  軸)、奥行き方向 ( $Z$  軸) の座標値を計算できる。具体的には、水平方向 ( $X$  軸)、垂直方向 ( $Y$  軸) の座標値は、次のようにして計算できる。ただし、被写体表面の点  $P$  の実空間での 3 次元座標を  $(X, Y, Z)$  とする。RGB-D カメラの焦点距離を  $f$ 、投影中心からカメラ画像の中心への直線とカメラ画像の法線とが成す角を  $\theta$ 、カメラ画像上での線分の端点を  $P'$ 、画像の中心から点  $P'$  までの  $x$  軸方向のピクセル単位の距離を  $x1$ 、被写体表面  $P$  までの奥行き距離を  $z1$  とする。また、本研究で使用する RGB-D カメラは、水平方向の視野角が 57 度、解像度は  $640 \times 480$  であるため、図 3.15 における  $W$  は 320[pixel]、 $\alpha$  は 28.5[度] である。焦点距離  $f$  は  $W$  と  $\alpha$  を用いて式 3.4 で表される。

$$f = \frac{W}{\tan \alpha} \quad (3.4)$$

図 3.15 より、 $\tan \theta$  は式 3.5 で表される。

$$\tan \theta = \frac{x1}{f} = \frac{X}{z1} \quad (3.5)$$

したがって、 $X$  は式 3.6 で表される。

$$X = z1 \cdot \frac{x1}{f} \quad (3.6)$$

ただし、 $\theta$  は式 3.7 で表される。

$$\theta = \tan^{-1} \frac{x_1}{f} \quad (3.7)$$

$Y$  も同様に計算して、式 3.8 のように求めることができる。

$$Y = z_1 \cdot \tan \theta' \quad (3.8)$$

ただし、 $\theta'$  は式 3.9 で表される。

$$\theta' = \tan^{-1} \frac{y_1}{f} \quad (3.9)$$

以上のようにして、画像面の任意の位置の 3 次元座標を計算できる。次に、そのランダムに選ばれた 2 点を通る 3 次元空間内の直線を算出する。そして、算出された直線から予め定めた距離  $d$  以内に入る 3 次元空間における線分上の点の数を数える。以上の処理を予め定めた回数  $n$  だけ繰り返し、点の数が最も多かった線分を、正しい線分とする。本研究では、 $d=0.1[\text{m}]$ 、 $n=20$  とした。以上のように RANSAC を適用することで図 3.13 の例のような誤認識を減らすことができる。

線分の両端の座標を計算できれば、線分の長さなども計算できる。ここで、認識された線分のうち画面上での長さが 20pixel 以下の線分は、利用しないこととする。20pixel 以下の短い線分は、環境中に多く存在し、それらをリローカリゼーションの情報として利用しようとするデータベースの情報量が多くなり処理負荷がかかるためである。

次に、提案手法における 9 種類の絞り込み処理で使用するための情報をデータベースに保存する処理について述べる。

始めに、線分総数法で必要な情報について述べる。線分総数法では、画面上で認識される線分の総本数を利用するため、画像ごとに線分の総本数をデータベースに保存する。このとき、図 3.16 に示すように、総本数の情報は画像の番号及び RGBD 画像の縮小画像と組にして保存する。

次に、画面分割法で必要な情報について述べる。画面分割法では、図 3.18 の上側に示すように、画像を複数の領域に分割した場合の、それぞれの領域内に存在する線分の本数の情報を利用する。画面を分割する際の適切な領域の数は、処理対象とする環境の特徴により変化することが予想される。その為、領域の数を変化させて絞り込みの性能を調べ、適切な領域の数を決定する必要がある。図 3.17 に示すように、画面の領域ごとの線分本数の情報を利用した方が、画面全体の線分の数だけの情報を利用した場合よりも画像間の違いを識別しやすくなることが期待できる。図 3.17 における左右 2 枚の画像の線分の総数は共に 20 本であるが、画面を 2 分割した場合の線分の本数

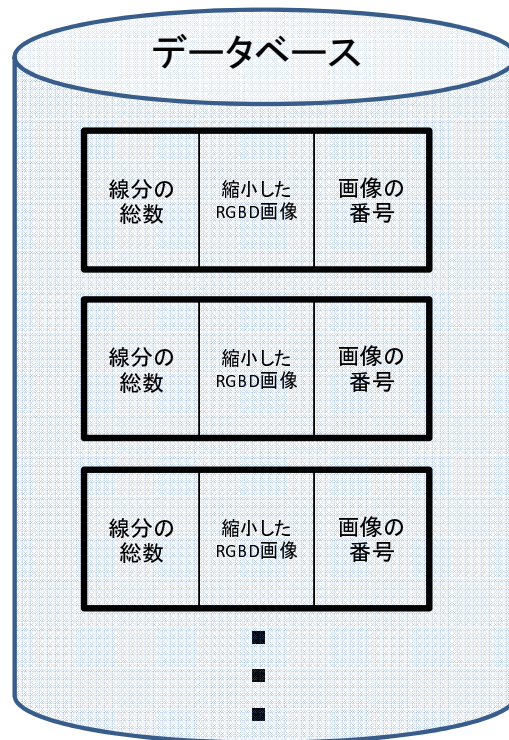


図 3.16: データベースに格納する線分総数法に必要な情報

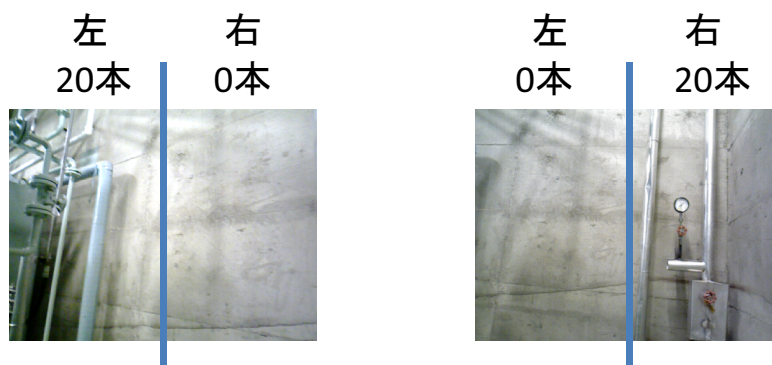


図 3.17: 画面上の位置による線分の本数の相違

領域①	領域②	領域③
10	14	33
8	10	3
領域④	領域⑤	領域⑥

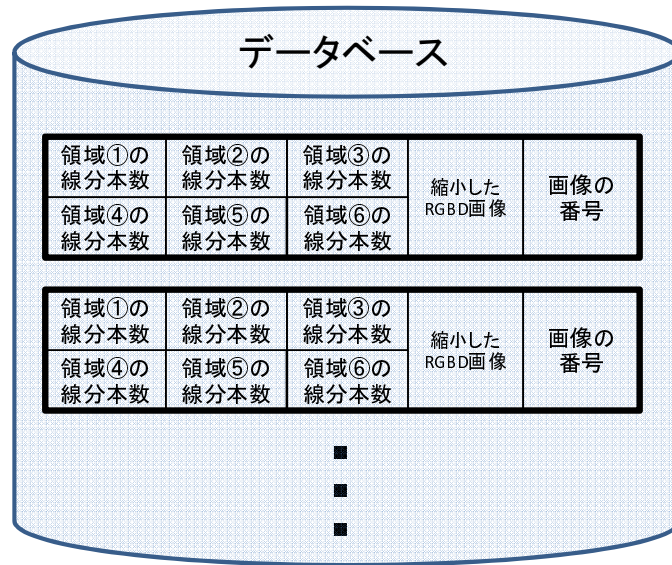


図 3.18: 画面分割法における画面の分割とデータベースの構成要素 (分割数を6とした場合)

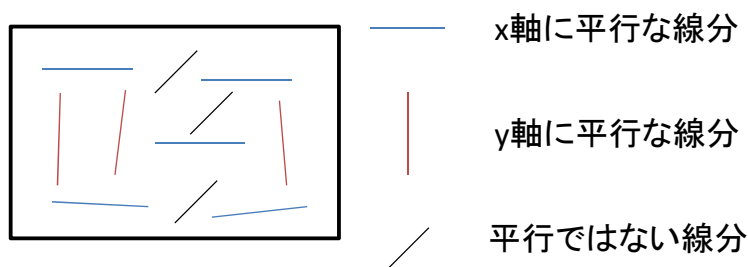


図 3.19: 平行線法における線分の分類



図 3.20: 水平・垂直な線分が数多く認識される場合の例

を見ると、左右の画像で、各領域の線分の本数が異なることが分かる。図 3.18 の上側に示すように画面を複数の領域に分割し、領域毎に含まれる線分の本数を RGBD 画像の縮小画像及び画像の番号と組にしてデータベースに保存する。

次に、平行線法で必要な情報について述べる。平行線法では、図 3.19 に示すように、画面上で認識される線分のうち、画面の横軸 (x 軸) に平行な線分と縦軸 (y 軸) に平行な線分の本数の情報を利用する。これは、図 3.20 に示すように原子力発電プラント内部には配管や機器、タンクなどの人工物が多く、その大半が地面に対して水平、垂直に作られており、それらの画像上の輪郭から認識された線分の画像の x 軸・y 軸に対する方向のずれの分布を、類似画像検索の情報として利用できること期待されるためである。x 軸、y 軸のそれぞれに対して予め定めた閾値  $\theta$  以内の角度の線分を、x 軸・y 軸に平行な線分と見なす。図 3.19 の例では、画面の x 軸に対して平行な線分が 5 本、画面の y 軸に対して平行な線分が 3 本である。図 3.21 に示すように、これらの x 軸と y 軸に対して平行な線分の本数を画像の番号および RGBD 画像の縮小画像と組にしてデータベースに保存する。



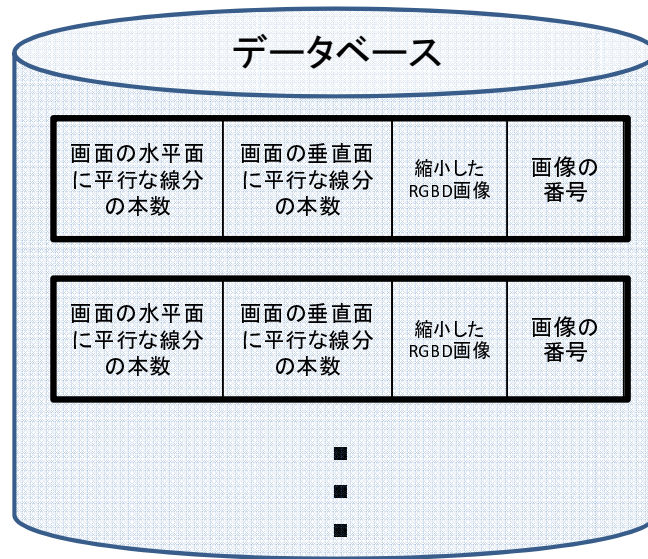


図 3.21: データベースに格納する平行線法で利用する情報

次に、2D 距離法で必要な情報について述べる。LSD で線分を認識すると、線分の両端の画面上での2次元座標を得ることができる。図 3.22 に示すように、ここでは、2本の線分の最も近い2つの端点の間の距離を線分の全ての組み合わせで計算する。次に、図 3.23(a) に示すように、線分間の距離が 10[pixel] 以上 30[pixel] 未満の数、30[pixel] 以上 50[pixel] 未満の数のように、距離に応じて複数のグループに分け (以下、グループをビンと称する) ビンごとに数を数える。図 3.23(b) に示すように、ビン毎の数を画像の番号および RGBD 画像の縮小画像を組にしてデータベースに保存する。

次に、3D 距離法で必要な情報について述べる。まず、各線分の両端の3次元座標を計算した後、その両端を通る3次元の直線を求める。その後、全ての直線の組み合わせで直線間の最短の距離を計算する。次に、2D 距離法と同様に、計算された直線間の距離を予め定めたビンの閾値毎に割り振り、ビン毎の数を求める。計算した直線間の距離のビンごとの数を画像の番号および縮小画像と組にしてデータベースに保存する。

次に、2D 長さ法で必要な情報について述べる。まず、各線分の画面上での長さを計算する。この時、図 3.24 に示すように、計算した線分の画面上での長さを予め定めたビンの閾値ごとに割り振り、ビン毎の数を求める。計算した線分の画面上での長さのビン毎の数を画像の番号および縮小画像と組にしてデータベースに保存する。

次に、3D 長さ法で必要な情報について述べる。まず、線分の両端の点の3次元座標を計算し、線分の実空間での長さを計算する。ここで、2D 長さ法と同様に、計算された線分の長さを予め定めたビンの閾値ごとに割り振り、ビン毎の数を求める。この線

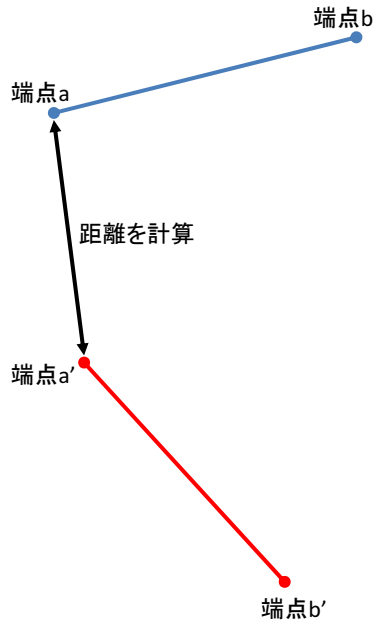


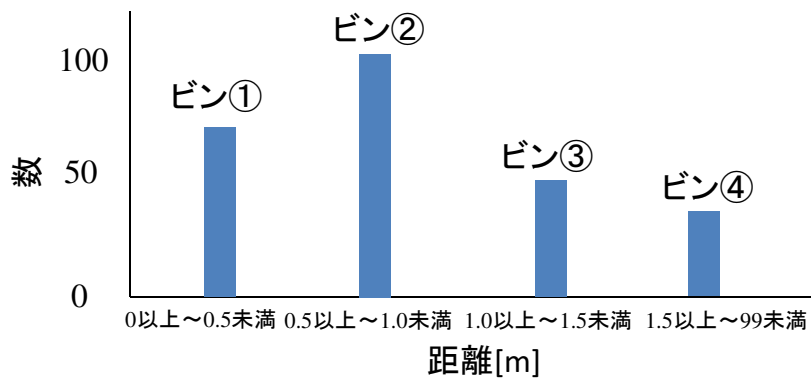
図 3.22: 2D 距離法における距離の計算

分の長さのビンごとの数を画像の番号および縮小画像と組にしてデータベースに保存する。

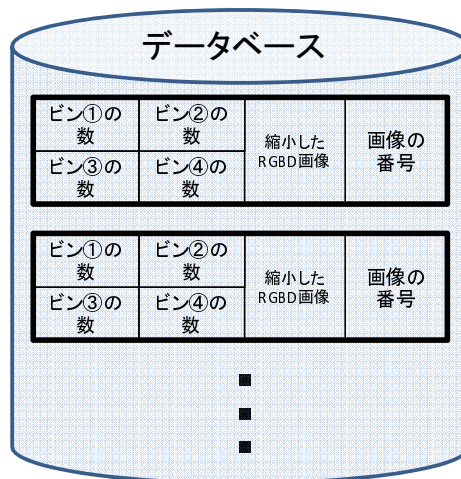
次に、2D 角度法で必要な情報について述べる。まず、各線分の両端を始点、終点とするベクトルを求め、そのベクトルのなす角度として、全ての線分の組み合わせで線分間の角度を求める。その際、求められた角度  $a$  が  $90$  度を超える場合には、 $180 - a$  を求めることにより、全ての結果が  $0$  度から  $90$  度に収まるようにする。ここで、図 3.25 に示すように、計算された角度を予め定めたビンの閾値毎に割り振り、ビン毎の数を求める。計算した 2 次元のベクトル間の角度のビンごとの数を画像の番号および縮小画像と組にしてデータベースに保存する。

次に、3D 角度法で必要な情報について述べる。まず、各線分の両端の 3 次元座標を計算し、その 2 点を始点、終点とする 3 次元ベクトルを求め、そのベクトルのなす角度として、全ての線分の組み合わせで線分間の角度を求める。その際、2D 角度法と同じように、求められた角度  $a$  が  $90$  度を超える場合には、 $180 - a$  を求めることにより、全ての結果が  $0$  度から  $90$  度に収まるようにする。計算した角度を予め定めたビンの閾値ごとに割り振り、ビン毎の数を求める。計算した 3 次元のベクトル間の角度のビンごとの数を画像の番号および縮小画像と組にしてデータベースに保存する。

最後に、入力画像と候補画像の縮小画像の輝度値の SAD を求めて類似画像を 1 枚選出する処理で必要な情報について述べる。環境を撮影した際の RGBD 画像を縮小した



(a) 線分間の距離のビンごとの分類



(b) データベースに格納する2D距離法の情報

図 3.23: 2D 距離法におけるビンとデータベースに格納する情報

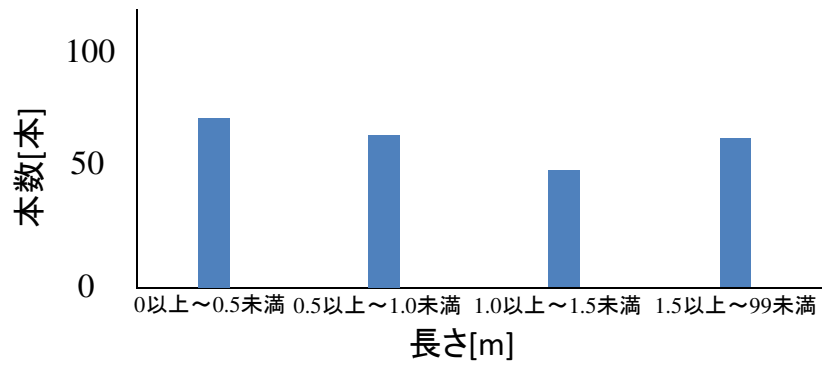


図 3.24: 2D 長さ法におけるヒストグラム

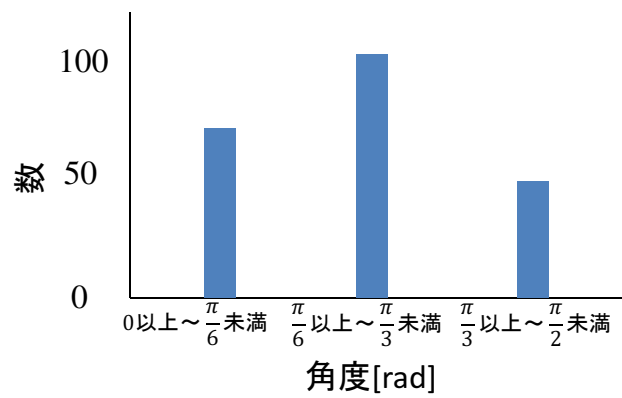


図 3.25: 2D 角度法におけるヒストグラム

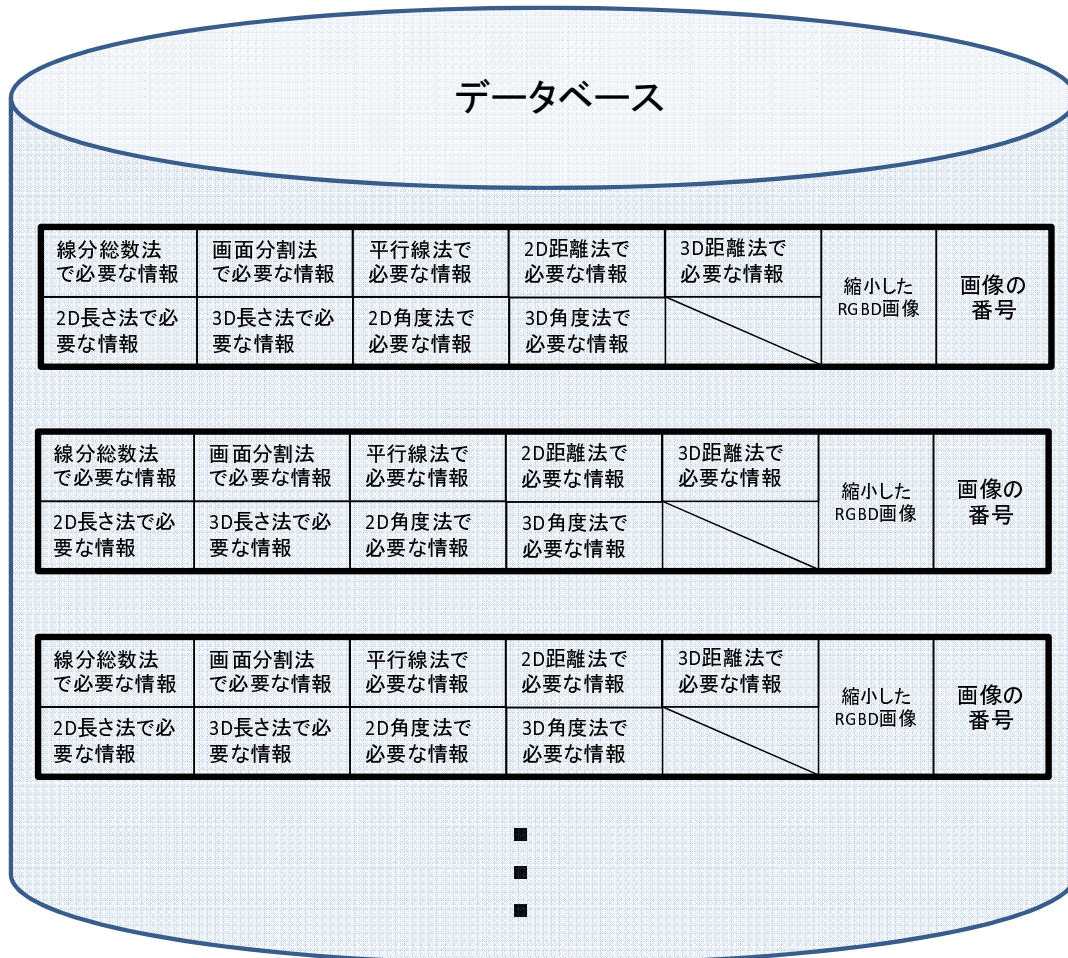


図 3.26: 事前準備で構築するデータベースに保存される情報

画像をそれぞれ画像の番号と組にしてデータベースに保存しておく。ただし、縮小画像はピクセルごとに R、G、B、D の 4 つの輝度値で表現されている。

以上に述べた方法で図 3.26 に示すように、データベースに線分総数法、画面分割法、平行線法、2D 距離法、3D 距離法、2D 長さ法、3D 長さ法、2D 角度法、3D 角度法で必要な情報を、縮小した RGBD 画像と画像の番号とを組にして保存する。

### 3.4 類似画像検索

実際にリローカリゼーションを実行する際の処理の流れを図 3.27 に示す。類似画像検索の処理では、最初に図 3.10 における 1 から 5 番と同じ処理を行い、入力画像の線分に関する各種情報を取得する。

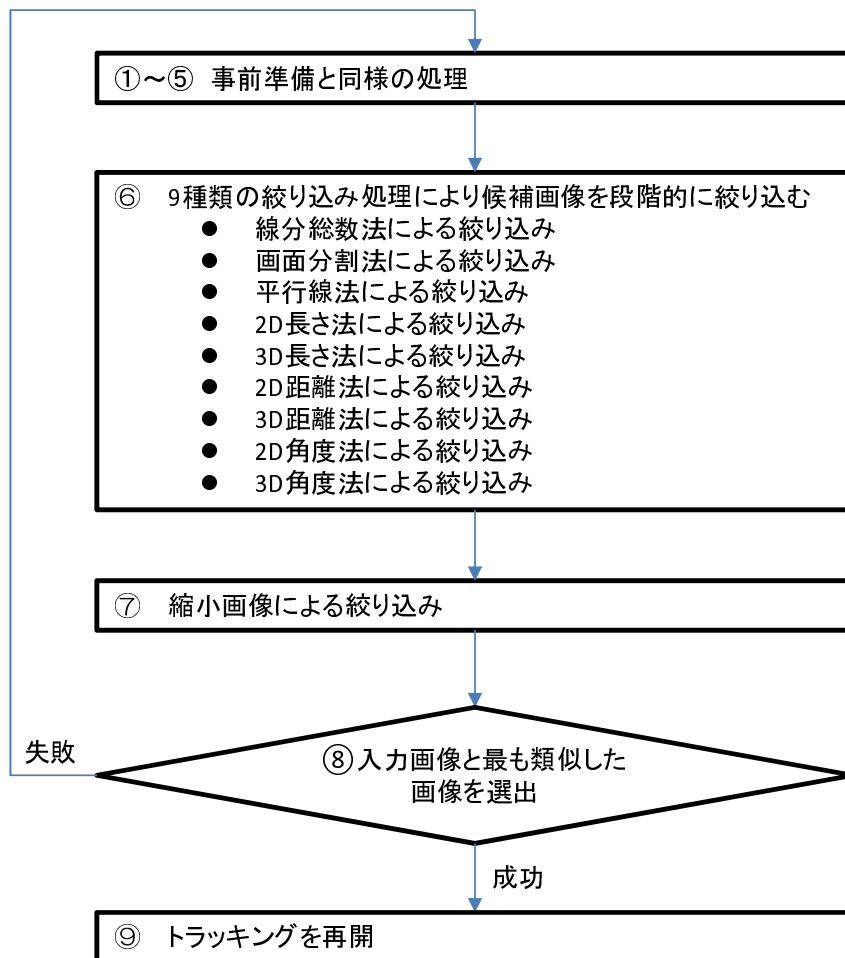
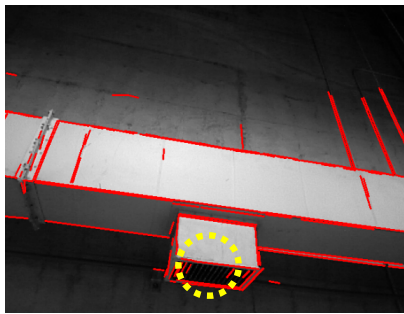
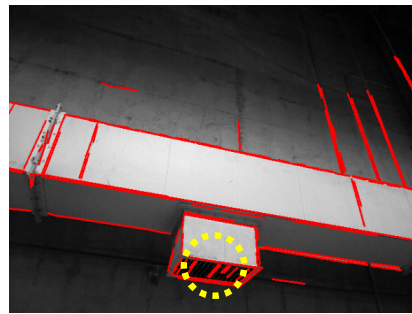


図 3.27: 実際にリローカリゼーションを実行する際の処理の流れ



(例)線分の総数20



(例)線分の総数40

線分の総数の差は20本  
線分の総数の差の割合は50%

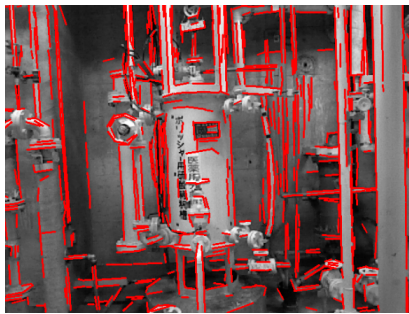
図 3.28: 線分の総数が少ない画像における線分の総数の差と差の割合

次に、入力画像から得られた線分に関する各種情報を利用して、データベース内にある候補画像を段階的に絞り込む処理を行う。提案手法の絞り込みの処理は前述のように9種類存在し、以下ではそれらを用いた絞り込み処理の詳細について述べる。

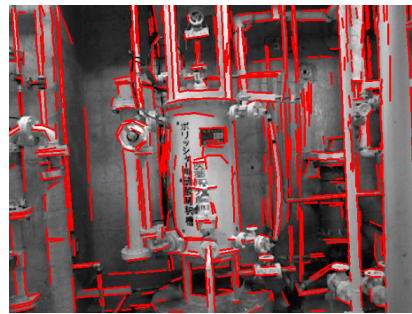
最初に、線分総数法を用いた候補画像の絞り込み処理について述べる。この処理ではまず、入力画像から認識された線分の本数と、データベースに保存されている候補画像から認識された線分の総数を比較する。この時、両者の線分の総数の差の絶対値が予め定めた閾値 numABS 以内、または総数の差の絶対値の入力画像上での線分の総数に対する割合が予め定めた閾値 numRAT 以内となる候補画像を探索する。ここで、線分の総数の差の割合を式 3.10 に示す。

$$\text{総数の差の割合} = \frac{\text{候補画像上の線分の総数} - \text{入力画像上の線分の総数}}{\text{入力画像上の線分の総数}} \quad (3.10)$$

線分の総数の差だけでなく、割合も用いて類似画像を探索するのは、図 3.28 に示すように、線分の総数の差の割合のみで入力画像と候補画像が類似しているかどうかを判断した場合、認識される線分の総数が少ない画像の場合は、撮影している位置・方向が近いにも関わらず、線分の総数の差の割合が大きくなる場合があり、本来であれば類似していると判断すべき画像を類似していないと判断してしまうことが起きてしまうためである。逆に、図 3.29 に示すように、画面上で認識される線分の総数が多かった場合に、線分の総数の差のみで入力画像と候補画像が類似しているかどうか判断すると、画像から認識される線分の総数が多い場合に、少しのカメラの位置・方向の変化が、認識される線分の本数に大きな変化を及ぼす場合があり、その場合にも、本来で



(例)線分の総数500



(例)線分の総数520

線分の総数の差は20本  
線分の総数の差の割合は4%

図 3.29: 線分の総数が多い画像における線分の総数の差と差の割合

あれば類似していると判断すべき画像を類似していないと判断してしまうことが起きてしまう。その為、線分の総数の差だけでなく線分の総数の差の割合も用いる。図 3.28 と図 3.29 ではどちらも線分の総数の差は 20 本であるが、線分の総数の差の割合は異なっている。以上のように、画面上で認識される線分の総数を用いる場合、線分の総数の差と線分の総数の差の割合の両方で類似画像の候補を残す必要があることが分かる。実際には類似している画像にも関わらず、認識される線分の本数に差が生じるのは、図 3.28 の黄色の丸で囲まれた部分に示すように、線分の認識が輝度値やデプスの小さな変化に影響され、線分の認識が不安定になる場合があるためである。以上に述べたように、線分総数法では、線分の総数の差の絶対値と、総数の差の絶対値の入力画像上の線分の増数に対する割合を用いて候補画像と入力画像が似ているか否かを判断し、似ていると判断された画像のみを候補に残す。

次に、画面分割法を用いた候補画像の絞り込み処理について述べる。図 3.18 に示したように画面を複数の領域に分割し、各領域ごとに線分の本数を数え、入力画像における各領域毎の線分の本数と候補画像の各領域毎の線分の本数を比較する。このとき、両者の線分の本数の差の絶対値が予め定めた閾値  $areaABS$  以内または、両者の線分の本数の差の絶対値の入力画像上の線分の総数に対する割合が予め定めた閾値  $areaRAT$  以内になる領域を互いに似ている領域として、その数を数える。ここで、領域における本数の差の割合を式 3.11 に示す。

$$\text{領域における本数の差の割合} = \frac{\text{候補画像の領域 } i \text{ の線分の本数} - \text{入力画像上の領域 } i \text{ の線分の総数}}{\text{入力画像の領域 } i \text{ の線分の総数}} \quad (3.11)$$



次に、似ていると判断された領域の数が予め定めた閾値  $sepaABS$  以内または、似ていると判断された領域の数の全体の領域数に対する割合が予め定めた閾値  $sepaRAT$  以内になる画像を候補画像として残す。 $sepaABS$  の閾値は式 3.12 のように、画面の分割数に応じて決定する。

$$sepaABS = \text{画面の縦の分割領域数} \times \text{画面の横の分割領域数} - 1 \quad (3.12)$$

図 3.18 に示した例の場合、画面の縦の分割領域数が 3、画面の横の分割領域数が 2 であるため、 $sepaABS$  は 5 となる。したがって、6 つの領域の内 5 つの領域で線分の本数が類似していれば、その画像は候補画像として残る。一方、図 3.18 では、 $sepaRAT$  の値は 0.5 としている。したがって、領域数 6 つの 0.5 である 3 つの領域の線分の本数が類似していれば、その画像は候補画像として残る。候補画像として残った画像以外は候補から除外する。

次に、平行線法を用いた候補画像の絞り込み処理について述べる。平行線法では、入力画像の  $x$  軸と  $y$  軸に平行な線分の本数をそれぞれ数え、候補画像における  $x$  軸、 $y$  軸に平行な線分の本数とそれぞれ比較する。この時、入力画像と候補画像の  $x$  軸に平行な線分の本数の差が予め定めた閾値  $paraABS$  以内、または  $y$  軸に平行な線分の本数の差が  $paraABS$  以内、または  $x$  軸に平行な線分の本数の割合が予め定めた閾値  $paraRAT$  以内、または  $y$  軸に平行な線分の本数の差の割合が  $paraRAT$  以内となる画像を候補として残し、残りを候補から除外する。ここで、画像の  $x$  軸に平行な線分の本数の割合を式 3.13 に、画像の  $y$  軸に平行な線分の本数の割合を式 3.14 に示す。

$$x \text{ 軸に平行な線分数の割合} = \frac{\text{候補画像の } x \text{ 軸に平行な線分数} - \text{入力画像の } x \text{ 軸に平行な線分数}}{\text{入力画像の } x \text{ 軸に平行な線分数}} \quad (3.13)$$

$$y \text{ 軸に平行な線分数の割合} = \frac{\text{候補画像の } y \text{ 軸に平行な線分数} - \text{入力画像の } y \text{ 軸に平行な線分数}}{\text{入力画像の } y \text{ 軸に平行な線分数}} \quad (3.14)$$

次に、2D 長さ法を用いた候補画像の絞り込み処理について述べる。2D 長さ法では、まず、認識された線分の画面上での長さを 3.3 節で述べた方法と同様の方法で計算する。次に、予め定めた閾値ごとにビンに割り振りビンごとに含まれる数を計算し、データベースに保存されている候補画像の情報と比較する。ビンに含まれる数を比較する様子を図 3.30 に示す。入力画像と候補画像のビンに含まれる数をそれぞれ比較する際に、数の差が予め定めた閾値  $2DLenABS$  以内、または数の差の割合が予め定めた閾値  $2DLenRAT$  以内のものを候補として残し、残りを候補から除外する。ここで、入力画

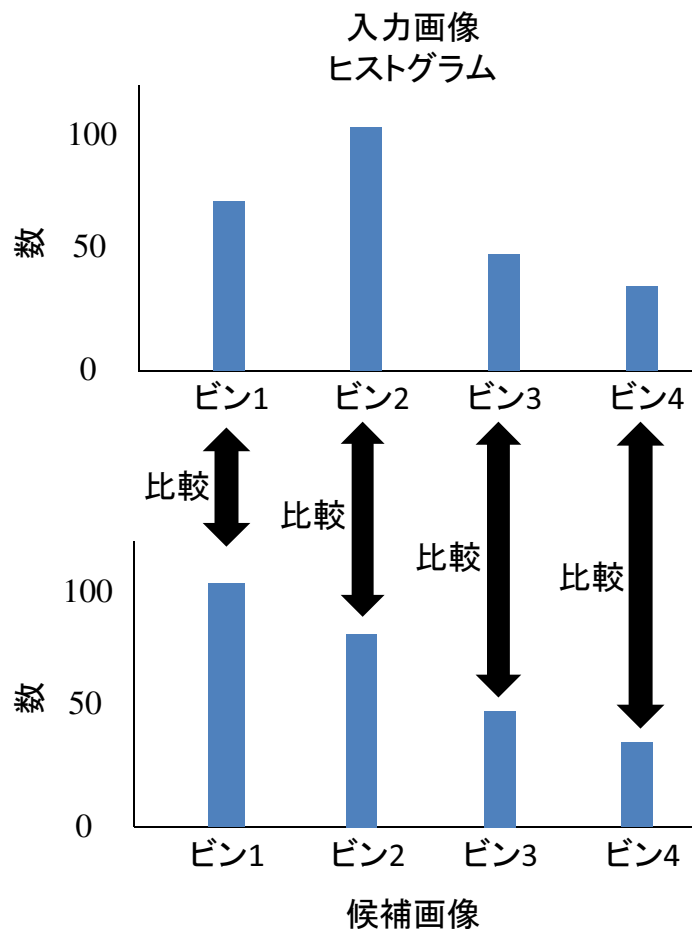


図 3.30: ヒストグラムの比較方法

像と候補画像のビンに含まれる線分の数の差の割合を式 3.15 に示す。

$$\text{ビンに含まれる線分の数の差の割合} = \frac{\text{候補のビンに含まれる線分の数} - \text{入力ビンに含まれる線分の数}}{\text{入力ビンに含まれる線分の数}} \quad (3.15)$$

次に 3D 長さ法を用いた候補画像の絞り込み処理について述べる。3D 長さ法は、入力画像から認識された線分の 3 次元空間内での長さを 3.3 節で述べた方法と同様の方法で計算し、2D 長さ法と同様に計算した長さを予め定めたビンの閾値ごとに割り振りビンごとに含まれる数を計算し、データベースに 3D 長さ法の情報として保存されているビンに含まれる数と比較する。入力画像と候補画像のビンに含まれる数を比較する際に、それぞれのビンに含まれる数の差が予め定めた閾値 3DLenABS 以内、または数の差の割合が予め定めた閾値 3DLenRAT 以内のものを候補として残し、残りを候補から除外する。ただし、差の割合は式 3.15 で表されるものとする。

次に 2D 距離法を用いた候補画像の絞り込み処理について述べる。2D 距離法は、入力画像から認識された線分同士の画面上での距離を 3.3 節で述べた方法と同様の方法で計算し、2D 長さ法と同様に計算した距離を予め定めた閾値でビンごとに割り振りビンごとに含まれる数を計算し、データベースに 2D 距離法の情報として保存されているビンに含まれる数と比較する。入力画像と候補画像の距離のビンに含まれる数を比較する際に、それぞれのビンに含まれる数の差が予め定めた閾値 2DDisABS 以内、またはそれぞれの数の差の割合が予め定めた閾値 2DDisRAT 以内のものを候補として残し、残りを候補から除外する。ただし、差の割合は式 3.15 で表されるものとする。

次に 3D 距離法を用いた候補画像の絞り込み処理について述べる。3D 距離法は、入力画像から認識された線分同士の 3 次元空間内での距離を 3.3 節で述べた方法と同様の方法で計算し、2D 長さ法と同様に計算した距離を予め定めた閾値でビンごとに割り振りビンごとに含まれる数を計算し、データベースに 3D 距離法の情報として保存されているビンに含まれる数と比較する。入力画像と候補画像の距離のビンに含まれる数を比較する際に、それぞれのビンに含まれる数の差が予め定めた閾値 3DDisABS 以内、またはそれぞれの数の差の割合が予め定めた閾値 3DDisRAT 以内のものを候補として残し、残りを候補から除外する。ただし、差の割合は式 3.15 で表されるものとする。

次に 2D 角度法を用いた候補画像の絞り込み処理について述べる。2D 距離法は、入力画像から認識された線分同士の画面上での角度を 3.3 節で述べた方法と同様の方法で計算し、2D 長さ法と同様に計算した角度を予め定めた閾値でビンごとに割り振りビンごとに含まれる数を計算し、データベースに 2D 角度法の情報として保存されているビンに含まれる数と比較する。入力画像と候補画像の角度のビンに含まれる数を比較す

る際に、それぞれのビンに含まれる数の差が予め定めた閾値  $2DAngABS$  以内、またはそれぞれの数の差の割合が予め定めた閾値  $2DAngRAT$  以内のものを候補として残し、残りを候補から除外する。ただし、差の割合は式 3.15 で表されるものとする。

次に 3D 角度法を用いた候補画像の絞り込み処理について述べる。3D 角度法は、入力画像から認識された線分同士の 3 次元空間内での角度を 3.3 節で述べた方法と同様の方法で計算し、2D 長さ法と同様に計算した角度を予め定めた閾値でビンごとに割り振りビンごとに含まれる数を計算し、データベースに 3D 角度法の情報として保存されているビンに含まれる数と比較する。入力画像と候補画像の角度のビンに含まれる数と比較する際に、それぞれのビンに含まれる数の差が予め定めた閾値  $3DAngABS$  以内、またはそれぞれの数の差の割合が予め定めた閾値  $3DAngRAT$  以内のものを候補として残し、残りを候補から除外する。ただし、差の割合は式 3.15 で表されるものとする。

最後に、候補として残った画像のみを対象として、入力画像の縮小画像と候補画像の縮小画像の SAD を求め、最も類似した画像 1 枚を選出する。

以上が実際のリローカリゼーションを行う際の処理であり、一般的なりローカリゼーション手法では類似画像として選出された候補と組になって保存されているカメラの位置・方向を現在のカメラの位置・方向の初期値としてトラッキングの再開を試みる。トラッキングが失敗すれば再び以上と同様のリローカリゼーション実行時の処理をトラッキングが成功するまで行う。

## 第 4 章 提案手法の評価

本章では、まずリローカリゼーション手法の評価の概要について述べる。次に、線分特徴を利用した多段式スクリーニング法における 9 種類の絞り込み処理それぞれにおける閾値の決定方法について述べる。その後、決定した閾値を用いて絞り込み処理の順番を決定する方法について述べ、決定した結果を用いて線分特徴を用いた多段式スクリーニング法の性能を評価し、Randomized Fern と比較する。最後に、評価結果の考察を述べる。

### 4.1 評価の概要

本研究では、実際の原子力発電プラント内で取得した画像を対象にリローカリゼーション手法を適用し、その成否と処理に要する時間で、リローカリゼーションの性能を評価する。ただし、リローカリゼーションの成否の判断は、カメラが正しい位置・方向に復帰できたか否かで判断するのではなく、類似画像を正しく選択できたか否かで判断する。具体的には、リローカリゼーションの性能を評価するにあたり、絞り込みの元になる候補画像を格納する候補画像データベースと、リローカリゼーションが必要な時の現在のカメラからの入力画像に相当する画像（以下、問題画像と呼ぶ）を複数格納する問題画像データベースを作成する。その際、図 4.1 に示す様に、先に候補画像データベースを作成し、その候補画像データベースからランダムに画像を抜き出すことにより問題画像データベースを作成する。なお、ランダムに抜き出した画像は、候補画像データベースからは削除するため、候補画像データベースと問題画像データベースで重複して存在する画像はないものとする。これは実際にリローカリゼーションを行う場合、リローカリゼーションに必要な情報が格納されている線分情報データベースを作成する時に取得した画像と完全に一致する画像がカメラから入力されることは非常に稀であることを模擬するためである。

最初にリローカリゼーションの成否の判断方法について述べる。リローカリゼーションの成否の判断は、問題画像を入力としてリローカリゼーションを行ったときに選出された画像が予め別の方法で求めておいた正解画像（以下、真値と呼ぶ）と一致しているかどうかで判断する。3.1.1 項で述べたように、一般的な画像を用いたリローカリゼー

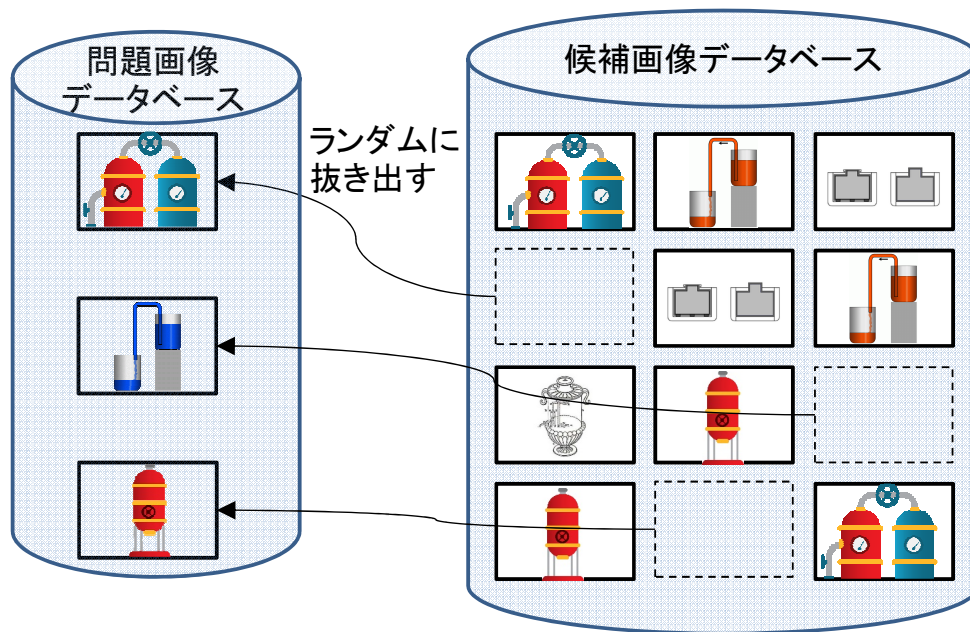
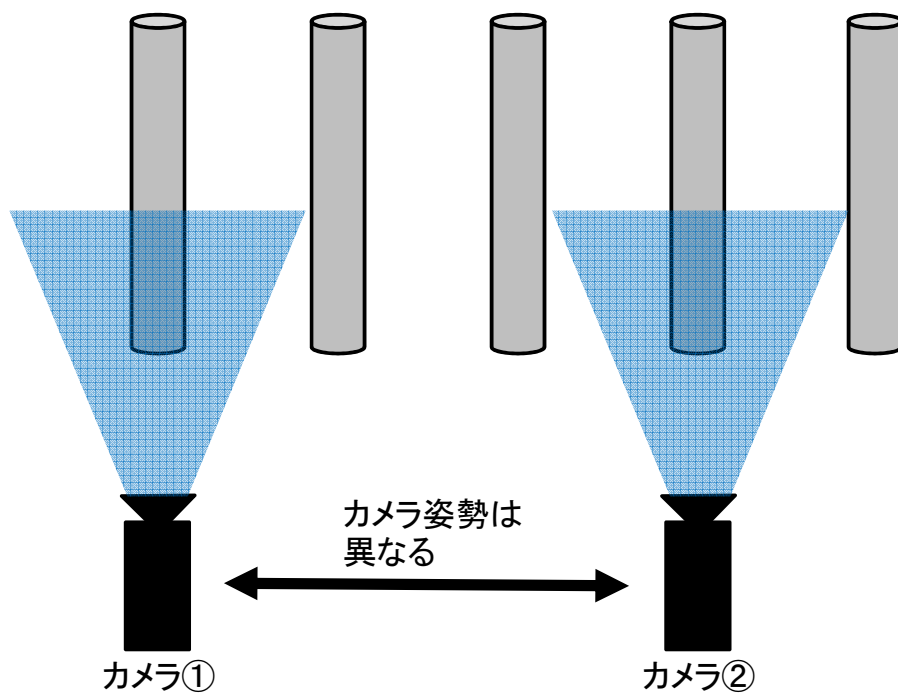


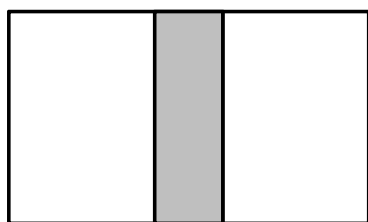
図 4.1: 候補画像データベースからの問題画像の抜き出し

シジョン手法では、トラッキングに失敗したときに現在のカメラの画像のみの情報を利用して、カメラ画像と類似した画像をデータベース内から探索し、選出された画像と組になって保存されているカメラの位置・方向を初期値としてトラッキングの再開を試みる。選出された画像と組になって保存されているカメラの位置・方向が正しければトラッキングが再開でき、リローカリゼーションが成功したことになるが、保存されているカメラの位置・方向が間違っていれば、トラッキングが再開できず、リローカリゼーションが失敗したことになる。つまり、画像を用いたリローカリゼーションの成否の判断は、通常はカメラの位置・方向を使用して判断するが、本研究では、最終的に選ばれた画像が正しいか否かでリローカリゼーションの成否を判断する。これは、原子力発電プラント内部の広域で複雑な環境下で多数のカメラの位置・方向の真値を得ることが現時点では技術的に難しいためである。また、リローカリゼーションの成否を現在のカメラの入力画像と類似した画像を選出できたか否かで判断した場合、図 4.2 に示すように、実際にはカメラの位置・方向が異なるにも関わらず外見は類似した画像を選出してしまう可能性がある。しかし、本研究では図 4.2 に示す様な状況が発生する画像の数は、作成する候補画像データベースに含まれる画像の数と比べて非常に小さいと予想されるため、今回の評価では影響は考慮しないものとする。

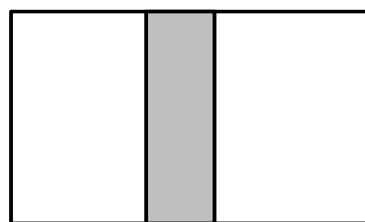
次に、評価に用いる真値の作成方法について述べる。本研究では、各問題画像に対する真値を求めるにあたり、計算負荷は非常に大きいが高精度な正規化相互相関を用



カメラ①が撮影した画像



カメラ②が撮影した画像



画像は類似している

図 4.2: カメラの位置・方向は異なるが類似した画像が撮影される例

表 4.1: Xtion PRO LIVE の仕様<sup>[24]</sup>

搭載センサ	RGB センサ、深度センサ、ステレオマイク
電源仕様	USB バスパワー
深度センサ有効距離	0.8m から 3.5m
深度センサ有効範囲	水平 58°、垂直 45°、対角 70°
深度センサ解像度	640 × 480 ドット (VGA) /30fps 320 × 240 ドット (QVGA) /60fps
プラットフォーム	Intel x86 プラットフォーム AMD プラットフォーム
対応 OS	Windows7(64bit/32bit) /XP(64bit/32bit) Ubuntu 10.10 以降 (Intel x86 ベースの 64bit/32bit 環境)
インターフェース	USB 2.0
使用可能環境	室内
サイズ	180 × 35 × 50mm

いることにした。すなわち、問題画像データベースに含まれる各画像毎に、候補画像データベースに含まれる全ての画像と正規化相互相関を求め、その結果が0.1以下となる全ての候補画像をその問題画像に対する真値とすることにした。したがって、1つの問題画像に対して複数の真値となる画像の番号が存在する可能性がある。正規化相互相関を求める際には OpenCV の `matchTemplate` 関数を `CV_TM_SQDIFF_NORMED` モードで使用した。このモードでは、正規化相互相関は式 3.1 で表される。候補画像データベースは RGB 画像とデプス画像を合わせて 236552 組のものを使用し、そこからランダムに 1000 組の画像を抜き出し、問題画像データベースを作成した。画像のサイズは  $320\text{pixel} \times 240\text{pixel}$  とした。

次に、候補画像データベースを作成する際の撮影条件について述べる。候補画像データベース内の画像は全て、新型転換炉ふげん発電所の純水装置室全体を撮影して作成した。候補画像データベースの画像を撮影した領域を図 4.3 に示す。撮影は図 4.4 に示す ASUS 社の Xtion PRO LIVE と呼ばれるデプスカメラを用い、撮影のフレームレートは 30fps とした。Xtion PRO LIVE の仕様を表 4.1 に示す。撮影の際にはカメラをゆっくりと動かし、できる限りぼやけが少ない画像を撮影するように努めた。また、カメラが物体や壁に近づきすぎないように撮影し、デプス値が正しくとれるように努めた。



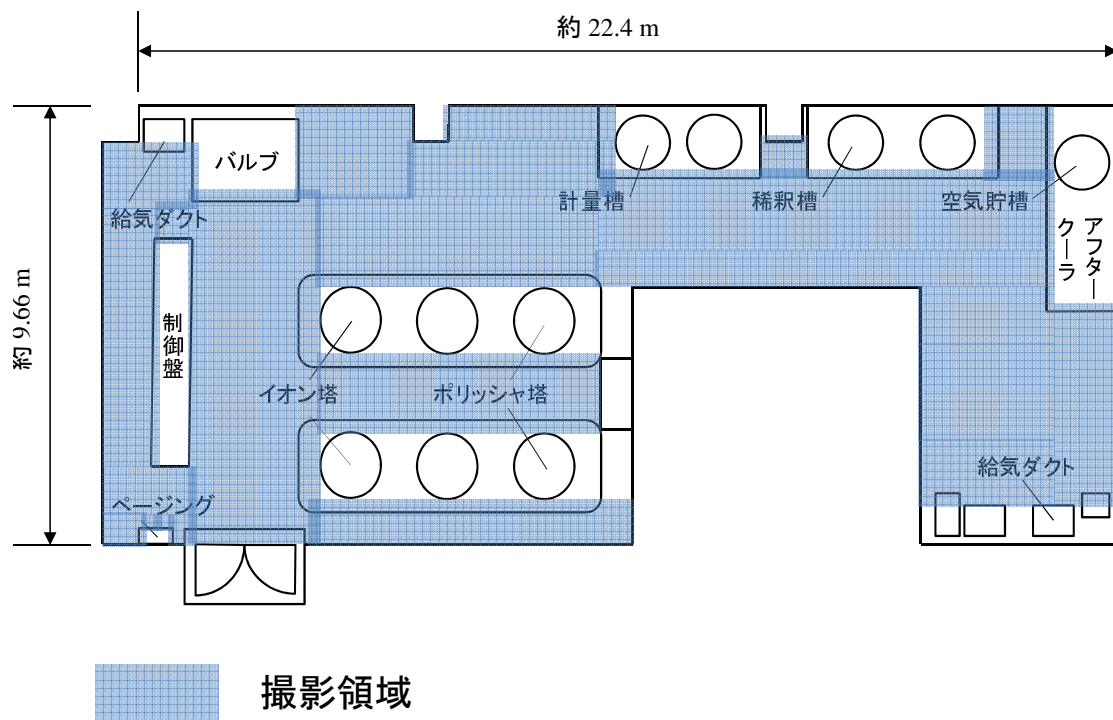


図 4.3: 純水装置室内での撮影領域



図 4.4: 画像の撮影に使用した Xtion PRO LIVE

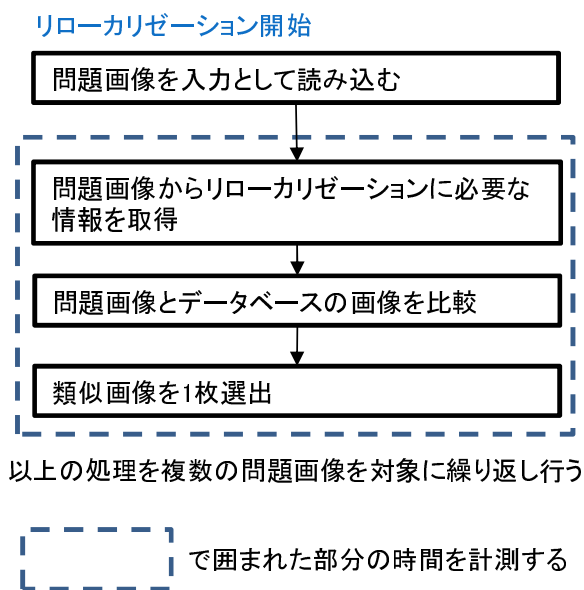


図 4.5: リローカリゼーションの処理時間

次に、リローカリゼーション手法の性能の評価の指標について述べる。リローカリゼーション手法の性能は、正答率と処理時間の2つの指標によって評価する。正答率をリローカリゼーションに成功した割合と定義し、本研究における正答率の定義を式 4.1 に示す。ただし、リローカリゼーションを行った回数は本研究では問題画像の枚数とする。

$$\text{正答率} = \frac{\text{リローカリゼーションに成功した回数}}{\text{リローカリゼーションを行った回数}} \quad (4.1)$$

リローカリゼーション1回の処理時間は、図 4.5 に示すように画像を読み込んだ後から類似画像が選出されるまでの時間と定義する。

本研究では、画像を用いたリローカリゼーションの代表的な手法である Randomized Fern と提案手法の性能を比較する。

## 4.2 提案手法のパラメータの決定

本節では、線分特徴を用いた多段式スクリーニング法の9種類の絞り込み処理で使用するパラメータの決定方法について述べる。9種類の全ての絞り込み処理で3.4節で述べたように、数の差と、数の差の割合の2種類の閾値を設定する必要がある。また、画面分割法では画面の分割数、平行線法では画面のx軸とy軸にそれぞれ平行であると判定する線分の角度範囲、2D長さ法・3D長さ法・2D距離法・3D距離法・2D角度法・

3D 角度法のビンの数および各ビンに線分を割り振る際の長さ範囲、距離範囲、角度範囲などのパラメータを決定する必要がある。これらのパラメータは、リローカリゼーションの性能に影響を与えると考えられるが、最適なパラメータの値は、リローカリゼーション処理の対象とする環境の複雑さに大きく依存すると予想される。そのため、高い性能のリローカリゼーションを実現するためには、リローカリゼーションの対象とする環境全体を事前に撮影して候補画像データベースを作成し、それを対象に様々なパラメータ値を用いてリローカリゼーションを実行し、最も高い性能となるパラメータ値を見つける処理(最適化処理)を実行することが望ましい。しかし、原子力発電プラント内部などの広い環境を対象とする場合、取得される画像の数が膨大になるため、その全てを用いて最適化処理を実行することは困難である。そこで本研究では、対象となる環境を撮影して得た画像の一部を用いて最適化処理を実行し、その結果得られた値を利用する手法を提案する。パラメータを決定する際の候補画像データベースは、4.1 節で述べた 236552 組の画像から 1000 組ずつランダムに 5 つ抜きた合計 5000 組の画像を使用した。作成した 5000 組の候補画像から、RGB 画像とデプス画像をランダムに 100 組抜き出し問題画像データベースを作成した。問題画像を 100 組抜き取ったため、候補画像データベースの画像は 4900 組となった。最適化処理を実行するにあたり、線分情報データベースに格納する縮小画像のサイズは事前検討でリローカリゼーションの精度が大きくは低下しない最小のサイズであることが確認された  $32pixel \times 24pixel$  とした。

各絞り込み処理における閾値決定の際には、3.4 節で述べた 9 種類の絞り込み処理をそれぞれ 1 種類ずつ実行し、その後、入力画像と候補画像の Sum of Absolute Difference(SAD : 市街地距離和) を求めることで類似画像を 1 枚に絞り込み、正答率を求める。この時、各絞り込み処理における数の差の閾値、数の差の割合の閾値、ビンの数、画面分割法における画面の分割数、平行線法における画面の x 軸と y 軸にそれぞれ平行であると判定する線分の角度範囲をそれぞれ変化させる。数の差の閾値と数の差の割合の閾値は一定のステップで増加させていき、正答率が一番高く処理時間が短い閾値を調べる。数の差の閾値は、10 から 100 まで 10 ステップで増加させ、数の差の割合の閾値は 0.05 から 0.9 まで 0.05 ステップで増加させる。さらに、画面分割法における画面の分割数、平行線法における平行線の許容角度、2D 長さ法・3D 長さ法・2D 距離法・3D 距離法・2D 角度法・3D 角度法のビンの数および長さ、距離、角度それぞれをビンに割り振る際の長さ・距離・角度の閾値を適宜変化させた。その結果、提案手法の正答率が一番高いものの中で処理時間が一番短くなった各絞り込み処理における閾

表 4.2: 各絞り込み処理における正答率が最大になる閾値

絞り込み処理	数の差の閾値	数の差の割合の閾値
線分総数法	numABS=10	numRAT=0.05
画面分割法	areaABS=10	areaRAT=0.4
平行線法	paraABS=10	paraRAT=0.25
2D 長さ法	2DLenABS=10	2DLenRAT=0.05
3D 長さ法	3DLenABS=10	3DLenRAT=0.05
2D 距離法	2DDisABS=20	2DDisRAT=0.45
3D 距離法	3DDisABS=30	3DDisRAT=0.45
2D 角度法	2DAngABS=60	2DAngRAT=0.75
3D 角度法	3DAngABS=20	3DAngRAT=0.15

値を表 4.2 に示す。また、そのときの各絞り込み処理におけるパラメータを表 4.3 に示す。表 4.2 に示す閾値よりも閾値を大きくすれば緩い閾値となり、各絞り込み処理後の候補画像の枚数が増加する。また、表 4.2 に示す閾値より値を小さくすれば厳しい閾値となり、各絞り込み処理後の候補画像の枚数が減少する。

### 4.3 絞り込み処理の順番の決定

本節では、提案手法の 9 種類の絞り込み処理の順番の決定方法について述べる。3.4 節で述べたように、提案手法では 9 種類の絞り込み処理を計算負荷が小さいものから順番に実行し、候補画像を段階的に絞り込んでいき、最終的に残った候補画像と入力画像の SAD を求めることで、類似画像を 1 枚選出する。以上のように、最初に計算負荷が小さい処理で候補画像を絞り込み、後に計算負荷は高いが精度が高い処理を実行することで、処理時間が短く精度の高いリローカリゼーションが実現できると期待される。絞り込み処理の順番を決定する際の候補画像データベースは、4.2 節で述べた 4900 組の候補画像データベースを用いた。また、絞り込み処理の順番を決定する際に使用した問題画像データベースも 4.2 節で述べたも問題画像データベースを用いた。提案手法の絞り込み処理の順番の決定に用いた Personal Computer(PC) の仕様を表 4.4 に示す。線分情報データベースに格納する縮小画像のサイズは  $32\text{pixel} \times 24\text{pixel}$  とした。

最初に、問題画像 100 組を入力画像として 9 種類の絞り込み処理をそれぞれ 1 種類

表 4.3: 各絞り込み処理におけるパラメータ

絞り込み処理	パラメータ			
画面分割法	画面分割数			
	縦	横		
	2	2		
平行線法	平行と判定する角度			
	水平面	垂直面		
	0度以上10度未満	80度以上90度未満		
2D長さ法	画面上での長さのビン			
	0pixel以上0.5pixel未満	0.5pixel以上1.0pixel未満	1.0pixel以上1.5pixel未満	1.5pixel以上100pixel未満
3D長さ法	実空間での長さのビン			
	0m以上0.5m未満	0.5m以上1.0m未満	1.0m以上1.5m未満	1.5m以上100.0m未満
2D距離法	画面上での距離のビン			
	0pixel以上30.0pixel未満	30.0pixel以上100.0pixel未満		
3D距離法	実空間での距離のビン			
	0m以上1.0m未満	1.0m以上100.0m未満		
2D角度法	画面上での角度のビン			
	0度以上22.5度未満	22.5度以上45度未満	45度以上67.5度未満	67.5度以上90度未満
3D角度法	実空間での角度のビン			
	0度以上22.5度未満	22.5度以上45度未満	45度以上67.5度未満	67.5度以上90度未満

表 4.4: 絞り込み処理の順番の決定に使用した PC の仕様と使用した開発環境

プロセッサ	Intel Core i7-4700MQ 2.40 GHz
RAM	16 GB
OS	Windows7 Professional 64 bit
開発環境	Visual Studio 2010
開発言語	C++
使用ライブラリ	OpenCV3.1.0、boost1.5.9

表 4.5: 各絞り込み処理の平均順位

絞り込み処理	平均順位
2D 角度法	1.0
3D 角度法	2.0
3D 長さ法	3.2
2D 長さ法	4.1
平行線法	4.8
画面分割法	6.3
3D 距離法	7.6
2D 距離法	7.7
線分総数法	8.3

ずつ実行し、処理時間を計測し、問題画像 100 組の処理時間の平均を求め、処理時間の平均が一番短かったものを 1 段階目の絞り込み処理とする。次に、2 段階目の絞り込み処理として残り 8 種類の絞り込み処理をそれぞれ 1 種類ずつ実行し、8 種類の処理時間の平均を比較し、一番処理時間の短かったものを 2 段階目の絞り込み処理として採用する。以上を絞り込み処理が残り 1 種類になるまで繰り返し行い、絞り込み処理の順番を決定する。

以上の方法で処理の順番を決定する場合、候補画像データベースと問題画像データベースが同じであれば、結果は常に同じになるはずであるが、処理を実行している間の OS のバックグラウンドの処理の影響などで、実際には繰り返し実行すると結果が変動する。そこで本研究では、10 回処理を繰り返し、順番の平均を求めることにした。結果を表 4.5 に示す。表 4.5 に示した平均順位の結果より、提案手法の絞り込み処理の順番は、1 段階目が 2D 角度法、2 段階目が 3D 角度法、3 段階目が 3D 長さ法、4 段階目が 2D 長さ法、5 段階目が平行線法、6 段階目が画面分割法、7 段階目が 3D 距離法、8 段階目が 2D 距離法、9 段階目が線分総数法となる。ここで、処理段階毎の処理時間の 10 回の平均を図 4.6 に示す。図より、処理時間に注目すると 6 段階目の処理で処理時間が長くなっているが、7 段階目以降の処理で再び処理時間が減少していることが分かる。ここで、決定した順番で提案手法を実行した時の各絞り込み処理後の候補画像の枚数を表 4.6 に示す。また、候補画像の枚数の変化の様子を図 4.7 に示す。さらに、図 4.7 の赤色の丸で囲まれた部分を拡大したグラフを図 4.8 に示す。図 4.7、図 4.8 より、

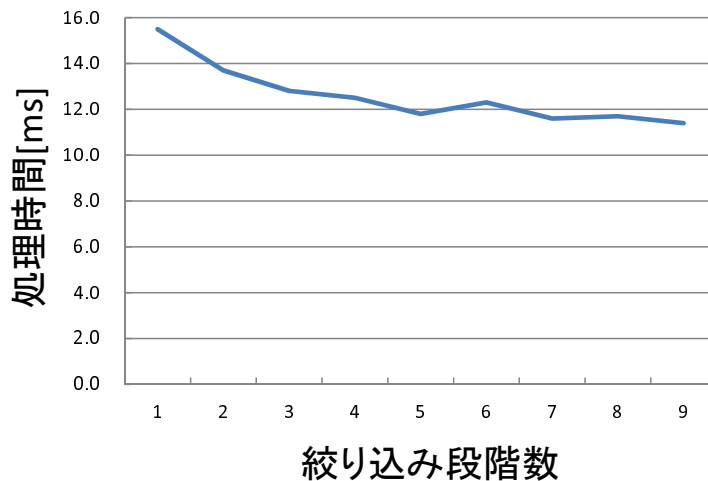


図 4.6: 各処理段階の処理時間の平均

絞り込み処理を適用する毎に、段階的に候補画像が絞り込めていることが分かる。

#### 4.4 提案手法の性能の評価

本節では、線分特徴を用いた多段式スクリーニング法の性能の評価について述べる。提案手法と Randomized Fern の性能を比較し、最後に考察を述べる。性能の評価には、4.1 節で述べた 236552 組の候補画像データベースを使用した。また、問題画像は RGB 画像とデプス画像を合わせて 1000 組とした。性能の評価に使用した PC の仕様を表 4.7 に示す。線分情報データベースに格納する縮小画像のサイズは  $16\text{pixel} \times 12\text{pixel}$  とした。

4.3 節で決定した 9 種類の絞り込み処理の順番を適用した提案手法を、1000 組の問題画像を入力として実行したときの正答率と処理時間の平均 (1 組あたりの所要時間) を表 4.8 に示す。表 4.8 に示す提案手法の性能と、Randomized Fern の性能を比較する。<sup>[17]</sup> の論文で提案されている Randomized Fern では、リローカリゼーションの際に入力画像と候補画像の BlockHD を計算し、BlockHD が小さい上位 5 つを入力画像と類似している画像の候補として選出するが、本研究では、上位 1 つのみを類似画像として選出した。

最初に、<sup>[17]</sup> の論文で提案されている通りのパラメータを用いて Randomized Fern を実行し、原子力発電プラント内部で撮影した大量の画像を使用した時に、パラメータの調整をせずにどの程度の処理時間と正答率を示すか調べた。その結果表 4.9 に示すよ

表 4.6: 最適化後の処理の順番で実行した時の候補画像組数

手法	候補画像組数
なし	4900
(1段階目)2D 角度法	346.9
(2段階目)3D 角度法	196.0
(3段階目)3D 長さ法	132.0
(4段階目)2D 長さ法	110.7
(5段階目) 平行線法	91.5
(6段階目) 画面分割法	85.1
(7段階目)3D 距離法	84.4
(8段階目)2D 距離法	83.9
(9段階目) 線分総数法	82.1

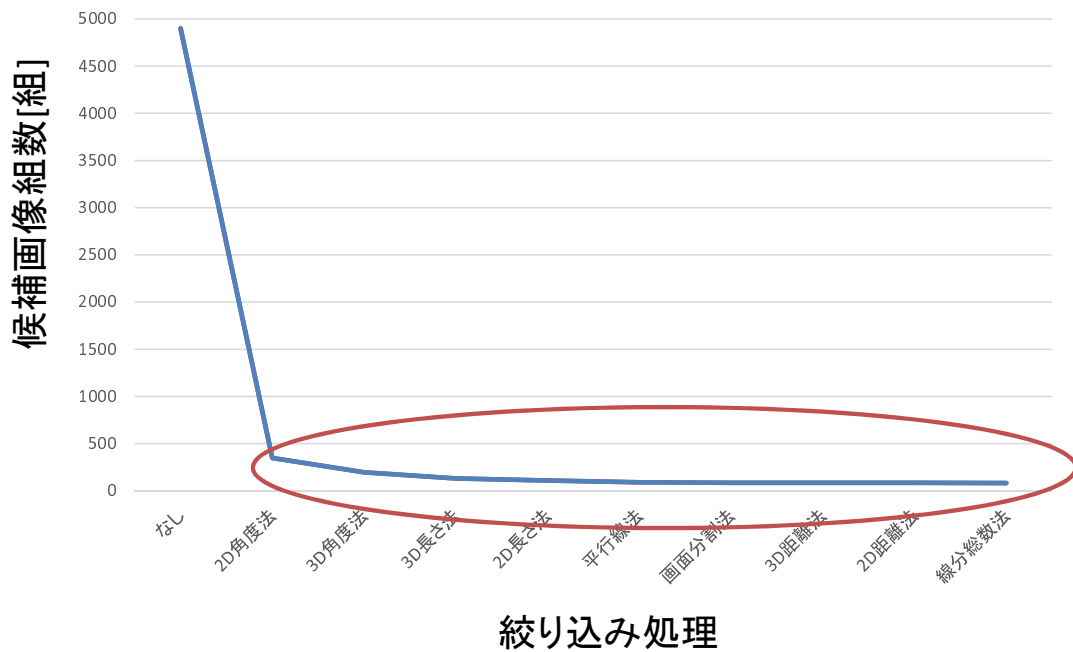


図 4.7: 最適化後の処理の順番で実行した際の候補画像の組数の変化



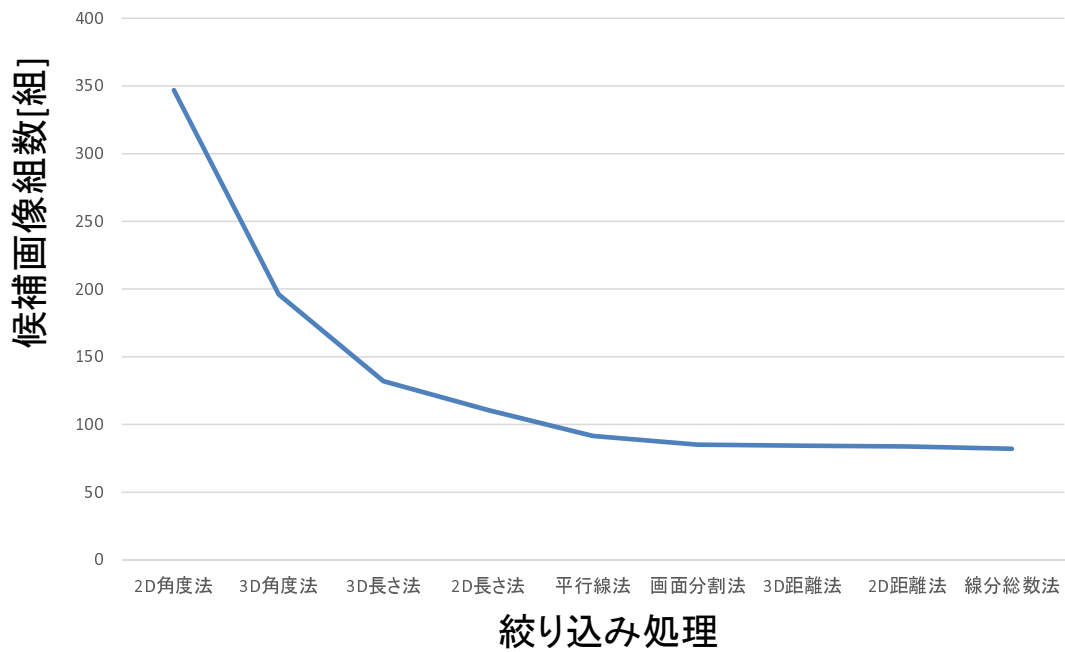


図 4.8: 最適化後の処理の順番で実行した際の候補画像の組数の変化の拡大図

表 4.7: 性能の評価に使用した PC の仕様

プロセッサ	Intel Core i7-4700S 3.10 GHz
RAM	24 GB
OS	Windows7 Professional 64 bit
開発環境	Visual Studio 2010
開発言語	C++
使用ライブラリ	OpenCV3.1.0、boost1.5.9

表 4.8: 提案手法の性能の評価結果

処理時間 [ms]	正答率 [%]
45.7	93.6

表 4.9: Randomized Fern の性能の評価結果

条件		結果	
fern 数	BlockHD	処理時間 [ms]	正答率 [%]
500	0.1	16.3	61.9
500	0.0	49.6	93.6

うに正答率は 61.9% と低く、このままではプラント内での作業支援に使用できない。

そこで、Randomized Fern の正答率を上げるため、<sup>[17]</sup> で提案されているパラメータのうち、BlockHD の値を 0.0 に変更して Randomized Fern を実行し、処理時間と正答率を計測した。その結果、表 4.9 に示すように、正答率が 93.6% で提案手法と同様となったが、処理時間は 49.6ms で提案手法よりも長くなった。

## 4.5 評価結果の考察

本節では評価結果の考察を述べる。最初に、提案手法におけるパラメータの考察について述べる。次に、提案手法の絞り込み処理の順番に関する考察を述べる。最後に、提案手法の性能の考察を述べる。

### 4.5.1 提案手法のパラメータの考察

表 4.2 に示した閾値は、4900 組の候補画像データベースで提案手法を実行したときに、正答率が最も高く、処理時間が短くなる値であり、データベースの画像の枚数や撮影環境が変化した場合には、適切な閾値の値が異なると考えられる。したがって、この閾値をそのまま使用した場合、候補画像データベースの枚数などが変化した場合、正答率が下がる可能性がある。そこで、表 4.2 に示す閾値の値よりも閾値を緩くすることで、各絞り込み処理によって真値の画像が候補から除外される可能性が低くなることが期待できる。

閾値を厳しくしすぎると正答率が下がるのは、除外される候補画像が多くなり、その除外された候補画像の中に真値の画像が含まれる可能性が高くなるためであると考えられる。しかし、閾値を緩くした場合、絞り込み処理で候補画像が十分に絞り込まれず、最後に行う縮小画像の差分を求める計算負荷の高い処理の対象となる画像が増加し、処理時間が長くなる。2D 角度法における差の閾値の変化による正答率の変化の

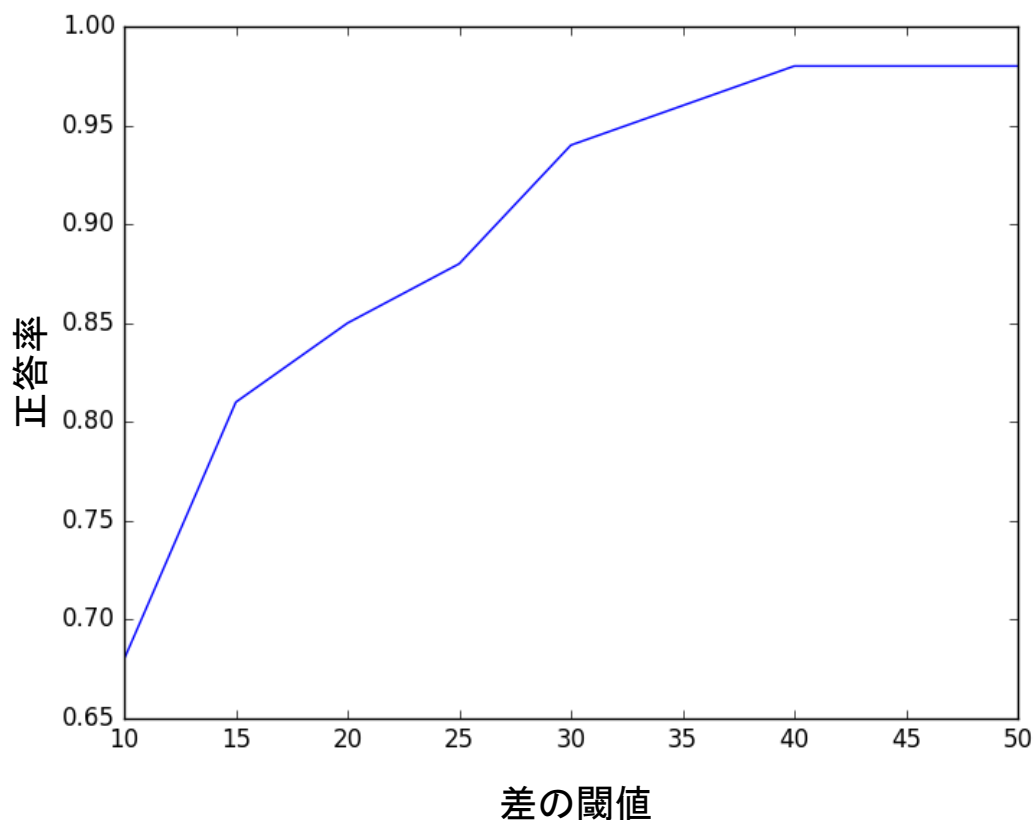


図 4.9: 2D 角度法における差の閾値と正答率の関係

様子を図 4.9 に示す。図 4.9 より、閾値を緩くするほど正答率が上昇しているのが分かる。他の絞り込み処理においても同様の現象が確認できた。

#### 4.5.2 絞り込み処理の順番の考察

7 種類の処理のうち、平均順位が安定して 1 位、2 位となったのはそれぞれ 2D 角度法と 3D 角度法であるが、これは 2D・3D 角度法では、線分情報データベースに保存する情報を作成する処理が線分の認識の不安定さにほとんど影響されないためであると考えられる。LSD で線分を認識すると、図 4.10 に示すように、同じ場所を撮影した画像であっても少しの輝度の変化の違いで線分が切れたりつながったりする場合がある。このように線分の認識が不安定になると、線分の長さは不安定になる。一方、線分間の角度は、角度を計算する際に 3.3 節で述べたように提案手法では線分を直線に変換し、

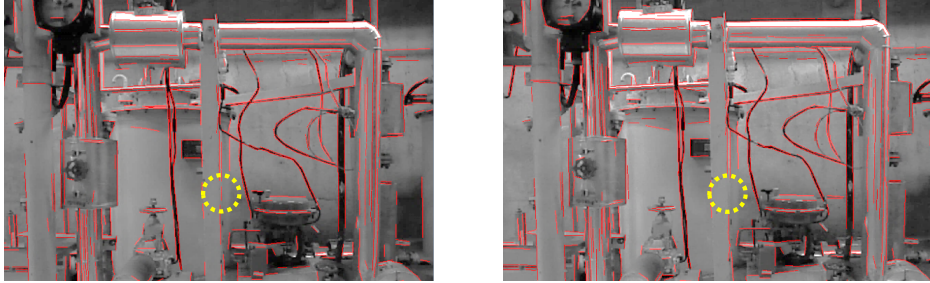


図 4.10: 線分の認識が不安定な場合の例

直線間の角度を計算しているため、線分が切れたりつながったりした場合であっても、角度の値に変化はない。つまり、似たようなカメラの位置・方向で撮影した画像間では同じ線分の組の角度はほぼ同じになり安定している。したがって、2D・3D 角度法の閾値を厳しくすることができ、閾値を厳しくすると 2D・3D 角度法を実行した後の候補画像の枚数を減らすことができる。その結果、提案手法の最後に行う SAD を求めて類似画像を 1 枚選出する処理の対象が減り、計算負荷が低くなり、平均順位が 1 位及び 2 位になったと考えられる。

次に、平均順位が 3.2 となり高いのが 3D 長さ法である。カメラの位置や方向が変化した場合でもカメラが写している対象の実世界の線分の長さは変化せず、安定して 3 次元空間上での線分の長さを計算できる。そのため、3D 長さ法の閾値を厳しくすることができ、閾値を厳しくすると 3D 長さ法を実行した後の候補画像の枚数を減らすことができる。その結果、提案手法の最後に行う縮小画像の比較によって類似画像を 1 枚選出する処理の対象が減り、計算負荷が低くなり、平均順位が高くなったと考えられる。しかし、撮影に使用した Xtion PRO LIVE ではデプスの値が不安定になりやすく、線分の実世界での長さを安定して計算することが難しく、閾値を厳しくしすぎると正答率が下がってしまうため、角度法より閾値を緩くしなければならない。その結果、縮小画像の比較によって類似画像を 1 枚選出処理の対象が、角度法よりも多くなり、平均順位が 3 位になったと考えられる。

表 4.10: 9種類の絞り込み処理と処理時間の平均

手法	処理時間平均 [ms]
(1段階目)2D 角度法	11.95
(2段階目)3D 角度法	4.87
(3段階目)3D 長さ法	2.95
(4段階目)2D 長さ法	0.93
(5段階目) 平行線法	0.39
(6段階目) 画面分割法	1.95
(7段階目)3D 距離法	0.40
(8段階目)2D 距離法	0.35
(9段階目) 線分総数法	0.22

### 4.5.3 提案手法の性能の考察

提案手法の性能の評価で使用した RGB 画像とデプス画像を合わせて 236552 組の候補画像データベースで提案手法を実行し、問題画像を 1000 組としたときの各絞り込み処理の処理時間の平均 (1 組あたりの処理時間) を表 4.10 に示す。また、各絞り込み処理と処理時間の関係を図 4.11 に示す。図 4.11 から、画面分割法の処理では処理時間が長くなっていることが確認できる。ここで、各絞り込み処理後の候補画像の組数の平均を表 4.11 に示す。表 4.11 における候補画像の組数は全て、RGB 画像とデプス画像の組数を表している。各絞り込み処理による候補画像の組数の変化を図 4.12 に示す。図 4.12 より、提案手法による絞り込み処理により候補画像の枚数を大幅に減少させることができていることが分かる。図 4.12 の赤色の部分を拡大したものを図 4.13 に示す。図 4.13 からわかるように、7段階目以降ではほとんど候補画像を絞り込めていないことが分かる。これは、絞り込み処理が進むにつれて、線分に関する各種情報だけの比較では入力画像と候補画像で似ているものが多くなり、入力画像と候補画像の違いを判別できなくなる可能性が高くなるためであると考えられる。

次に、表 4.8 に示した結果の際の提案手法の各処理毎の処理時間の平均を表 4.12 に示す。表 4.12 から分かるように、提案手法の処理のうち、最も処理時間が長い処理が最後に行う問題画像と候補画像の縮小画像を比較する処理で、全体の処理時間の約 28.0% を占めている。次に処理時間が長い処理が 2D 角度法の処理であり、全体の約 26.3% を

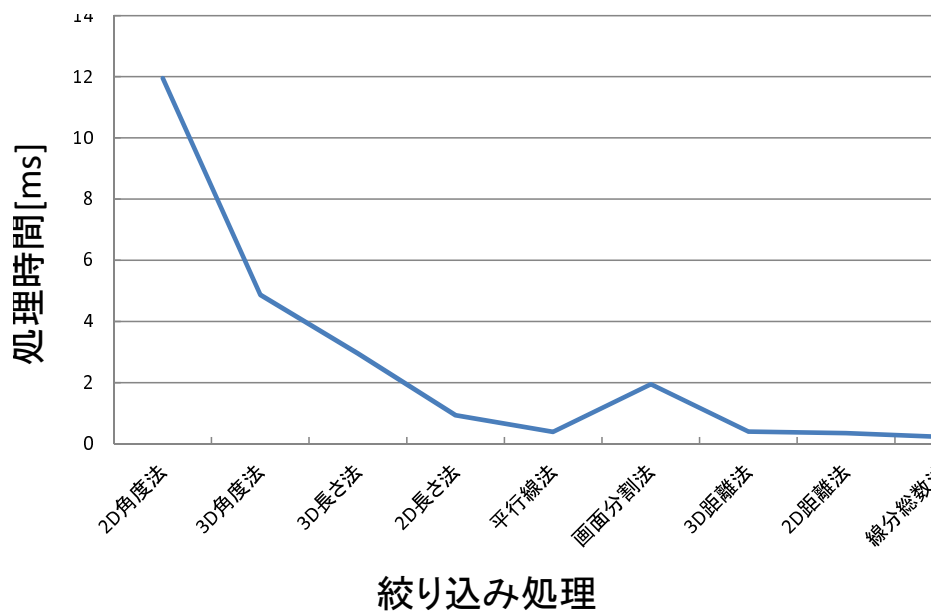


図 4.11: 9 種類の絞り込み処理と処理時間の関係

表 4.11: 提案手法の各絞り込み処理後の候補画像の組数の平均

手法	候補画像組数
なし	236552
(1 段階目)2D 角度法	12982.0
(2 段階目)3D 角度法	7149.9
(3 段階目)3D 長さ法	4664.1
(4 段階目)2D 長さ法	3933.6
(5 段階目) 平行線法	3361.3
(6 段階目) 画面分割法	3116.4
(7 段階目)3D 距離法	2837.7
(8 段階目)2D 距離法	2740.9
(9 段階目) 線分総数法	2707.4

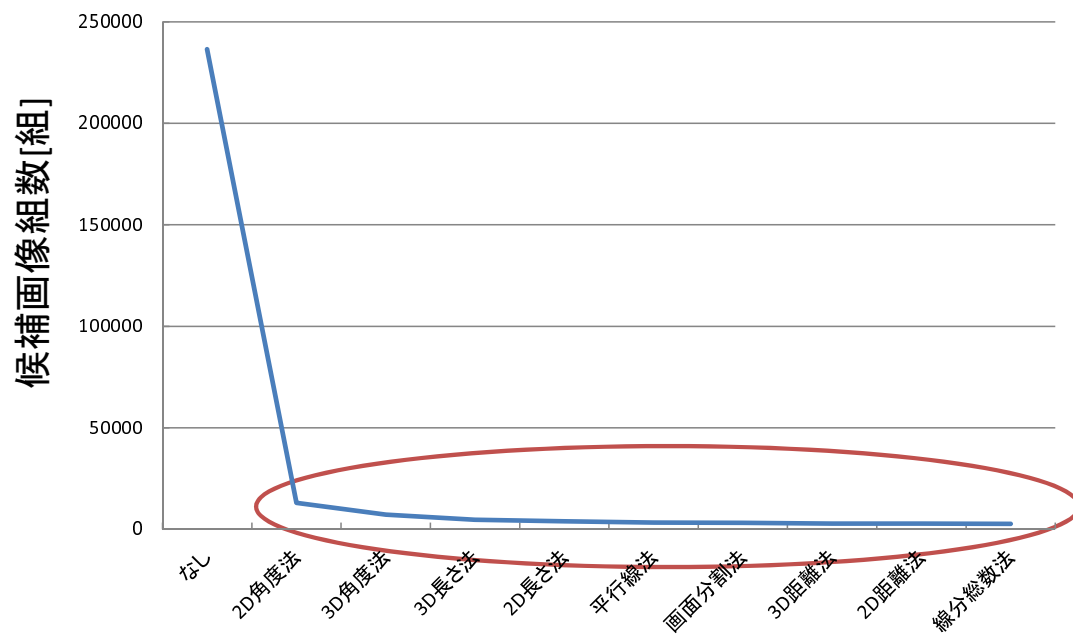


図 4.12: 提案手法の各絞り込み処理による候補画像の組数の変化

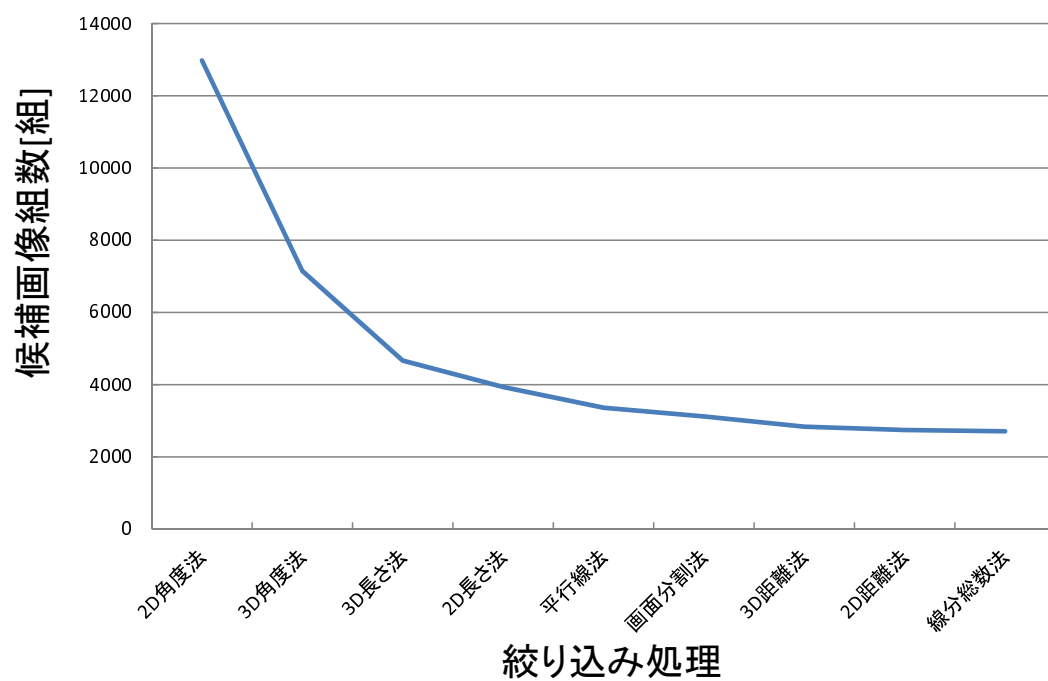


図 4.13: 提案手法の各絞り込み処理による候補画像の組数の変化の拡大図



占めている。3番目に処理時間が長い処理が問題画像から線分を認識する処理などリローカリゼーションに必要な情報を取得する処理で、全体の処理時間の約19.5%を占めている。以上の3つの処理を合わせて約73.8%となる。最も処理時間が長い縮小画像比較の処理の時間を短くするためには、処理時間の短い9種類の絞り込み手法により候補画像をなるべく多く絞り込み、縮小画像を比較する処理の回数を減らす必要がある。2番目に処理時間が長い処理が2D角度法となっているが、これは、2D角度法は提案手法の絞り込み処理の1段階目の処理であり、絞り込む対象の候補画像の枚数が多いためであると考えられる。性能の評価の際に2D角度法が処理の対象とした画像はRGB画像とデプス画像を合わせて236552組である。一方、1番処理時間が長かった縮小画像の比較の処理が対象とした画像の枚数の平均はRGB画像とデプス画像を合わせて約3585組であった。したがって、2D角度法は処理する画像の枚数が縮小画像の比較で処理する枚数の約66倍多い。しかし、2D角度法の処理時間は縮小画像の比較時間とほぼ同じである。したがって、2D角度法は非常に高速で画像を処理していることが分かる。また、2D角度法以外の他の8種類の絞り込み処理も、全体の処理時間にほとんど影響を与えない。以上より、最も処理時間が長い縮小画像の処理を軽くできれば、提案手法の処理時間の短縮が期待できる。

4.4節で述べたように、提案手法ではRandomized Fernよりも処理時間が短くなったが、これは提案手法では計算負荷が小さい処理で問題画像と類似した画像の候補を段階的に絞り込んでいき、最後に残った少数の候補画像のみを対象に計算負荷は高いが精度の高いSADを求めて類似画像を選出しているため、全体として処理時間が短く、正答率が高いリローカリゼーションを実現できたと考えられる。しかし、提案手法では線分を認識する処理があり、表4.12に示すように提案手法の中で3番目に処理時間が長い処理であるが、この処理時間は候補画像の数に関わらず一定の長さとなる。そのため、候補画像の枚数が少ない場合はRandomized Fernの方が処理時間が短くなると考えられる。したがって、リローカリゼーションを行う環境の大小に応じて、適切なリローカリゼーション手法を選択する必要がある。

表 4.12: 提案手法の各処理毎の処理時間

処理	処理時間 [ms](全体処理時間に対する割合)
線分情報を取得	8.9(19.5%)
(1段階目)2D 角度法	12.0(26.3%)
(2段階目)3D 角度法	4.9(10.7%)
(3段階目)3D 長さ法	3.0(6.6%)
(4段階目)2D 長さ法	0.9(2.0%)
(5段階目) 平行線法	0.4(0.9%)
(6段階目) 画面分割法	2.0(4.4%)
(7段階目)3D 距離法	0.4(0.9%)
(8段階目)2D 距離法	0.4(0.9%)
(9段階目) 線分総数法	0.2(0.4%)
縮小画像比較	12.8(28.0%)
全体処理時間	45.7(100%)

## 第 5 章 結論

原子力発電プラント内部は広域で複雑な形状の物体が多数存在しており、そのような環境で十分高速かつ高精度にリローカリゼーションできる手法はまだ開発されていない。本研究では、原子力発電プラント内部に数多く存在する配管のエッジなどの線分を認識する手法を用いれば、リローカリゼーションの安定性を向上させられる可能性があることに着目し、線分特徴を利用した絞り込み処理によるリローカリゼーション手法を開発することを目的とした。

開発した提案手法の性能を、画像を用いたリローカリゼーションの代表的な手法である Randomized Fern の性能と比較した。性能を比較する際に使用したりローカリゼーションの候補画像データベースには、新型転換炉ふげん発電所の純水装置室の内部を撮影した RGB 画像とデプス画像を合わせて 236552 組の画像とした。性能の比較には、正答率と処理時間の 2 つの指標を用いた。正答率とは、リローカリゼーションを行った回数に対してリローカリゼーションが成功した回数を表している。処理時間とは、入力画像を撮影してからリローカリゼーションが終わるまでの時間を表している。リローカリゼーションの成否を判断するための真値には、事前にデータベースからランダムに 1000 組抜き出した問題画像と類似した画像を正規化相互相関を用いて探索して使用した。

提案手法と Randomized Fern の性能を比較した結果、正答率が同じとなる条件では、提案手法の方が、処理時間が短いという結果が得られた。提案手法では現在のカメラ画像から線分を認識する処理が増えるが、候補画像を計算負荷が小さい処理で段階的に絞り込んでいき、最終的に候補として残った少数の画像のみを対象として計算負荷が高い Sum of Absolute Difference(SAD : 市街地距離和) を求める処理を行うことで、結果として処理時間が短く、精度の高いリローカリゼーションが実現できた。データベースの画像の枚数が少ない場合には、Randomized Fern の方が処理時間が短くなるが、今回の研究では原子力発電プラント内部などの広域な環境を想定しているため、そのような環境では提案手法が有効であることがわかった。

本研究では絞り込み処理のパラメータの決定を RGB 画像とデプス画像を合わせて 4900 組の小さい候補画像データベースを用いて行った。提案手法の性能の評価に使用した 236552 組の候補画像データベースでは提案手法の絞り込み処理の最適な閾値が異

なるはずである。したがって、使用する環境ごとに提案手法の絞り込み処理における最適な閾値を決定すれば、提案手法の性能はさらに向上することが期待できる。

また、本研究では線分の認識を安定化させるために RANSAC を使用したが、他にも線分の認識を安定化させる処理を加えれば、提案手法の性能はさらに向上することが期待できる。

今後は、原子力発電プラント内部の広域な環境下でデータベースの枚数に応じた最適な閾値を効率的に決定する方法を開発し、提案手法の準備作業の手間を最小限に抑えると共に、計算負荷が小さく線分の認識を安定化させる処理を提案手法に組み込み、提案手法の安定性や速度を向上させる工夫が必要である。

## 謝 辞

本研究を進めるにあたり、丁寧かつ熱心なご指導を頂きました下田宏教授に感謝の意を表します。就職活動の際には、ご多忙にも関わらず毎朝面接のご指導を頂きましたことにも合わせて御礼申し上げます。

本研究のテーマのアイデア、提案手法のアルゴリズムの構築、プログラミングの指導、論文の執筆方法、プレゼンテーションの技法、研究生活全般等の様々な面で、平日休日、昼夜問わず多くのご助力とご指導を頂きました石井裕剛助教に、心から感謝の気持ちと御礼を申し上げたく、謝辞にかえさせていただきます。

同じ AR チームとして、多くの刺激と示唆を頂きましたエネルギー科学研究科修士 2 回生の藤井巧哉君、Yongxin Wang 君、エネルギー科学研究科修士 1 回生の木村太郎君、遠藤竜太君、工学部電気電子工学科 4 回生の大橋由暉君に感謝いたします。

本論文の執筆にあたり、メインサポーターとして数々の助言や添削など、惜しみないご協力を頂きました木村太郎君には、改めて御礼申し上げます。また、サブサポーターとして非常に丁寧かつ的確な添削を下さったエネルギー科学研究科修士 1 回生の浦山大輝君には、感謝の念がたえません。本当にありがとうございました。

同じ研究室の先輩として、エネルギー科学研究科博士課程の北村尊義さん、エネルギー科学研究科修士課程の上東大裕さんには、日頃から研究の進み具合を気にかけていただき、終始適切な助言を賜りました。心より感謝いたします。

2 年間の大学院での研究生活を愉快地に過ごせたのは、同期の皆様の御陰です。同期の金川英弘君、藤井巧哉君、古田真也君、Yongxin Wang 君に御礼申し上げます。

上下関係なく仲良くして接してくれた後輩のエネルギー科学研究科修士 1 回生浦山大輝君、木村太郎君、下中尚忠君、遠藤竜太君、杉田耕介君、工学部電気電子工学科 4 回生の上田樹美さん、大橋由暉君、辻雄太君に感謝いたします。

英語のネイティブスピーカーと会話するという貴重な機会を与えて下さった留学生の Husni Razana さん、Bingrong Huang 君に感謝しております。Thank you very much.

細かな事務手続きや研究生活全般にわたり、親身になって支えて下さった普照郁美秘書に深謝いたします。

研究にあたり何度も足を運ばせていただいた原子炉廃止措置研究開発センターの皆様に、改めて御礼申し上げます。

## 参考文献

- [1] 経済産業省: 資源エネルギー庁ホームページ, <http://www.enecho.meti.go.jp/about/whitepaper/2015pdf/>, Accessed December 29, 2015.
- [2] W. Yan, S. Aoyama, H. Ishii, H. Shimoda, T. Sang, S. Inge, T. Lygren, Jo. Terje, M. Izumi: Development and Evaluation of a Temporary Placement and Conveyance Operation Simulation System Using Augmented Reality, *Nuclear Engineering and Technology*, 44(5), pp.507-522 (2010).
- [3] 石井裕剛: 応用2: プラント保守作業支援, *情報処理*, 51(4), pp.392-397 (2010).
- [4] T. Kimura, H. Tokumaru, H. Ishii, H. Shimoda, Y. Kouda: Evaluation and Improvement of a Relocalization Method for Camera Tracking in Nuclear Power Plants, *Proceedings of International Symposium on Socially and Technically Symbiotic Systems (STSS)*, pp.407-416 (2015).
- [5] I. Rabbi and S. Ullah: A Survey on Augmented Reality Challenges and Tracking, *Acta Graphica*, 1(2), pp.29-46 (2013) .
- [6] B. H. Thomas: A Survey of Visual Mixed, and Augmented Reality Gaming, *ACM Computers in Entertainment*, 10, pp.16-52 (2012).
- [7] 顧穎成: 自然特徴点を用いた広域トラッキングのための環境モデル更新手法, エネルギー科学研究科修士論文, 京都大学 (2014).
- [8] R. Vabo, L. Piotrowski, G. Rindahl: 3D representation of radioisotopic dose rates within nuclear plants for improved radioprotection and plant safety, *International Journal of Nuclear Safety and Simulation*, 1(2), pp.127-133 (2010).
- [9] 唐津豊、中澤仁、高汐一紀、徳田英幸: 加速度センサを利用したマーカレス AR によるセンサ情報可視化システム, *電子情報通信学会信学技報*, 110, pp.79-84 (2010).

- [10] J. Newman, D. Ingram, A. Hopper: Augmented Reality in a Wide Area Sentient Environment, Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR) 2011, pp.77-86 (2011).
- [11] H. Kato, M. Billinghurst: Marker Tracking and HMD Calibration for a Video based Augmented Reality, 2nd IEEE and ACM International Workshop, pp.85-94 (1999).
- [12] A. J. Davison, I. D. Reid, N. D. Molton, O. Stasse: MonoSLAM: Real-Time Single Camera SLAM, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(6), pp.1052-1067 (2007) .
- [13] 内山英昭, 北原格: AR(拡張現実) 技術の基礎・発展・実践, 科学情報出版株式会社, pp. 38-55 (2015).
- [14] J. Pacheco, J. Ascencio, J. Mancha: Visual simultaneous localization and mapping: a survey, Artificial Intelligence Review 43(1), pp. 55-81 (2015).
- [15] H. Li, H. Duan, X. Zhang: A novel image template matching based on particle filtering optimization, Pattern Recognition Letters, 31(13), pp.1825-1832 (2010).
- [16] J. Sarvaiya, S. Patnaik, S. Bombaywala: Image Registration by Template Matching Using Normalized Cross Correlation, Advances in Computing, Control, & Telecommunication Technologies, 2009, ACT'09, International Conference on, IEEE, pp.819-822 (2009).
- [17] B. Glocker, S. Izadi, J. Shotton, A. Criminisi: Real-Time RGB-D Camera Relocalization, Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on, 1(4), pp.173-179 (2013).
- [18] B. Glocker, J. Shotton, A. Criminisi, S. Izadi: Real-Time RGB-D Camera Relocalization via Randomized Ferns for Keyframe Encoding, Visualization and Computer Graphics, IEEE Transactions on, 21(5), pp. 571-583 (2015).
- [19] R. Gioi, J. Jakubowicz, J. Morel, G. Randall: LSD: a Line Segment Detector, Published in Image Processing On Line on 2012, pp.35-55 (2012).

- [20] R. Gioi, J. Jakubowicz, J. Morel, G. Randall: LSD: A Fast Line Segment Detector with a False Detection Control, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4), pp.722-732 (2008).
- [21] R. Gioi, J. Jakubowicz, J. Morel, G. Randall: On Straight Line Segment Detection, *Journal of Mathematical Imaging and Vision*, 32(3), pp.313-347 (2008).
- [22] V. Patraucean, P. Gurdjos, R. Gioi: A Parameterless Line Segment and Elliptical Arc Detector with Enhanced Ellipse Fitting, *Computer Vision ECCV 2012*, Springer Berlin Heidelberg, pp.572-585 (2012).
- [23] M. Fischler, R. Bolles: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Communications of the ACM*, 24(6), pp.381-395 (1981).
- [24] ASUSTeK Computer Inc.: ASUS Xtion PRO LIVE, [https://www.asus.com/jp/3D-Sensor/Xtion\\_PRO\\_LIVE/specifications/](https://www.asus.com/jp/3D-Sensor/Xtion_PRO_LIVE/specifications/), Accessed February 1, 2016.
- [25] 池田 光二, 吉田 昌司, 中島 啓介, 浜田 長晴, 依田 晴夫: 正規化相関演算の単調関数化による高速テンプレートマッチング, *電子情報通信学会論文誌 J83-D-(9)*, pp.1861-1869 (2000).
- [26] A. Goshtasby: Template Matching in Rotated Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 7(3), pp.338-344 (1985).



## 付録 A トラッキングの追跡処理

追跡処理の流れを図 A.1 に示す。具体的に追跡処理とは、時刻  $t$  の画像で認識された自然特徴に対して、時刻  $t+1$  の画像上で同一の自然特徴の位置を計算する処理のことである。時刻  $t+1$  の画像において、時刻  $t$  の画像で認識された自然特徴の周囲を探索範囲として設定する。次に、探索範囲内で時刻  $t$  の画像で認識された自然特徴をテンプレートとしてテンプレートマッチング<sup>[25][26]</sup>を行う。テンプレートマッチングとは、図 A.2 に示すように、画像の中の特定の一部分であるテンプレートと類似している部分を画像の中から探し出す処理のことである。図 A.2 では、青色の四角で囲まれた部分がテンプレートの画像と一類似していると判断される部分を表している。最後に、テンプレートマッチングにより最も似ていると判断された画素を算出し、時刻  $t+1$  の画像における自然特徴の位置とする。時刻  $t$  の画像において認識された自然特徴の 3次元座標が算出できている場合、自然特徴の 3次元座標と時刻  $t+1$  の画像における自然特徴の画像上の 2次元座標を用いて最適化問題を解くことで、時刻  $t+1$  におけるカメラの位置・方向を計算することができる。以上のような追跡処理を行うことで計算量を削減し、トラッキングの安定性の向上を図っている。

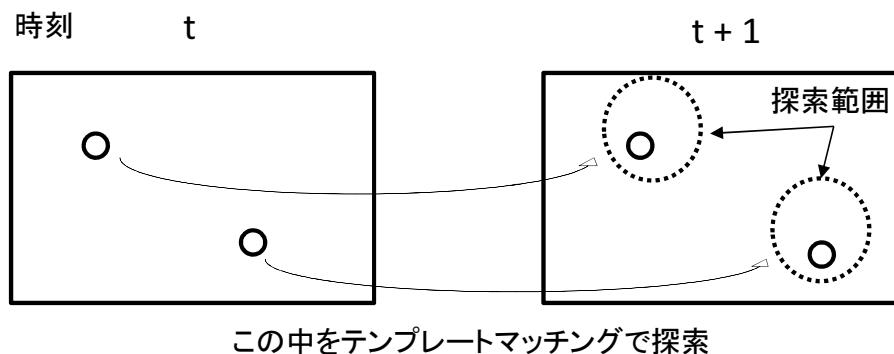


図 A.1: 特徴点の追跡処理<sup>[13]</sup>

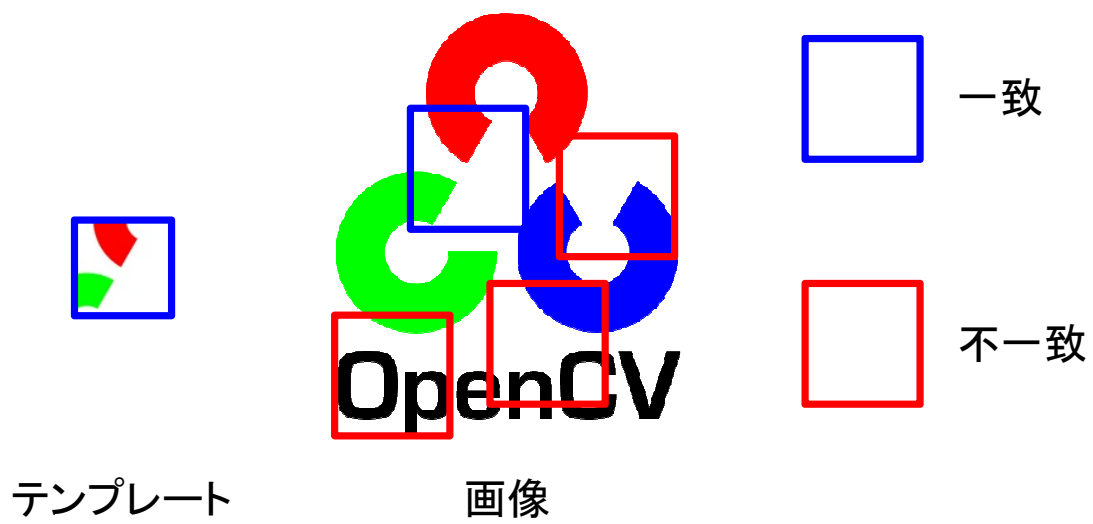


図 A.2: テンプレートマッチング