

アウトライアを考慮した楕円検出の高性能化に関する研究
(Study on improvement of ellipse detection with outliers)

2018年3月

博士(工学)

益崎 智成

豊橋技術科学大学

目次

第1章	はじめに	1
1.1	背景	1
1.2	目的	2
1.2.1	楕円限定当てはめ	3
1.2.2	楕円弧点列の選択	3
1.2.3	楕円検出の高性能化	4
1.2.4	移動ロボットへの応用	4
1.3	構成	4
第2章	楕円限定当てはめ	5
2.1	楕円当てはめ	5
2.2	入力点に加わる誤差と特性	6
2.3	従来の楕円当てはめ法の問題点	7
2.4	楕円限定当てはめ	7
2.4.1	ランダムサンプリングを用いた楕円限定当てはめ	8
2.5	RANSAC	9
2.6	PROSAC	10
2.6.1	点群の成長とサンプリング回数の更新	11
2.7	点の分布を考慮したランダムサンプリング	12
2.8	正規乱数を用いたサンプリング法	14
2.9	入力点列を5等分するサンプリング法	16
2.10	実験	16
2.10.1	実験環境	16
2.10.2	パラメータ	17
2.10.3	シミュレーション用点列データ	17
2.10.4	RMS 誤差と KCR 下界	17
2.10.5	相対ノイズレベル	19
2.11	超精度くりこみ法と解の判定	19
2.12	ランダムサンプリングの収束回数の比較	22
2.12.1	点の分布の考慮の仕方による収束回数の比較	22
2.12.2	既存手法との収束回数の比較	24
2.13	楕円限定当てはめの精度比較	26
2.14	実画像を用いた楕円当てはめの評価	27
2.15	まとめ	29

第 3 章	効率的な楕円弧選択	31
3.1	既存の楕円検出	31
3.2	提案手法	31
3.3	部分弧と接部分弧	32
3.4	部分弧の拡張	33
3.5	誤差グラフの曲率に注目した楕円弧選択	33
3.5.1	誤差グラフの曲率を用いた楕円弧選択の流れ	33
3.5.2	誤差グラフの横軸の正規化	34
3.5.3	誤差グラフの曲率の評価	35
3.6	実験	37
3.6.1	シミュレーション実験	38
3.6.2	実画像実験	41
3.6.3	手法の問題点	43
3.7	各部分弧への楕円当てはめによる楕円弧選択	43
3.7.1	1 回目の部分弧選択	44
3.7.2	2 回目以降の部分弧選択	44
3.8	実験	45
3.8.1	楕円弧の選択過程の確認	45
3.8.2	楕円弧選択性能の向上確認	45
3.8.3	楕円弧選択性能の比較	49
第 4 章	同一楕円上の楕円弧点列の統合	51
4.1	幾何学的 MDL モデル選択	51
4.1.1	幾何学的 MDL	51
4.1.2	ノイズレベルの推定	52
4.2	幾何学的 MDL を用いた菅谷の統合判定	52
4.3	提案手法	52
4.4	楕円弧の分類	53
4.4.1	楕円の中心座標の導出	53
4.5	当てはめた楕円の弧長に対する楕円弧の占める割合を用いた選別	54
4.5.1	統合判定から除外する楕円弧の選択	54
4.5.2	楕円の長軸と短軸の計算	55
4.6	当てはめた楕円に対する楕円弧の位置関係を用いた選別	55
4.6.1	楕円領域の拡大	55
4.6.2	統合判定を行う組合せの選別	56
4.7	実験	57
4.7.1	楕円弧統合の効率向上の確認	57
4.7.2	楕円検出性能の比較	57
4.8	まとめ	60
第 5 章	移動ロボットへの応用	63
5.1	ロボットの環境認識	63
5.2	提案手法	63
5.3	壁認識システムの概要	63

5.4	楕円の交点を用いた壁と床の境界の検出	65
5.4.1	楕円の定義	65
5.4.2	2つの楕円の境界線	65
5.4.3	壁と床の境界検出	66
5.5	実験	66
5.5.1	壁と床の境界検出結果	66
5.5.2	移動ロボットの動作	67
5.6	まとめ	68
第6章 まとめ		69
発表論文		73
付録A 既存の楕円当てはめ		79
A.1	FNS法	79
A.2	HEIV法	80
A.3	射影ガウスニュートン法	81
A.4	最小二乗法と重み反復法	81
A.5	Taubin法とくりこみ法	82
A.6	超精度くりこみ法と超精度最小二乗法	82
付録B 既存の楕円限定当てはめ		85
B.1	Fitzgibbonらの方法	85
B.2	Szpakらの方法	86
付録C 楕円弧選択のための放物線当てはめによるアウトライア除去		87
付録D 壁とロボット間の距離推定		91
D.1	円形マーカによるカメラの位置姿勢推定	91
D.2	距離推定	93

第1章 はじめに

1.1 背景

シーン中の円形の物体を撮影すると画像中では楕円となり，その投影像からその物体の3次元位置が解析できる [11]．このため，画像から楕円を抽出することは視覚ロボットを含む広範な応用の基本的な処理の一つであり，楕円弧を抽出する種々の研究がなされている [1, 33]．そして，画像より複数の楕円を検出するさまざまな方法が研究されてきた．

画像から抽出したエッジ点列より楕円を検出するには，ハフ変換などによる投票に基づいたパラメータ推定方法と，入力点列から直接的に楕円なパラメータを推定する楕円当てはめ手法 [40, 5, 47, 48] が挙げられる．

近年では楕円当てはめに基づいた楕円検出法が注目されてきているが，楕円当てはめ手法は，入力として与えられる点列は楕円弧上の点にエッジ検出時の誤差が加わったような点列を仮定しており，そのような入力に対して，いかに高精度に楕円パラメータを推定するかに重点がおかれている．したがって，入力点列が楕円弧上の点列であれば，高精度に楕円パラメータを推定することが可能である一方，入力点列が直線などの楕円弧上の点列ではなかったり，楕円弧とそれ以外の形状が組み合わさった点列に対しては，正しい楕円を検出することが困難である．

楕円以外が含まれる入力点列に対する対策として，さまざまな研究も行われている．簡単な方法によるアプローチとしては連結したエッジ点列の曲率を計算し，曲率が大きく変化する点を楕円弧とそれ以外の点列の接続点とみなし，連結した点列を分割して楕円当てはめの入力にする方法がある．しかし，この方法では楕円弧とそうでない点列の接続部分の曲率によって楕円弧とそれ以外の点列を正しく分割できない場合がある．

容易な方法としてRANSAC[43]を用いた楕円検出が挙げられる．RANSACはアウトライアを含む入力データから正しいパラメータを推定するための手法であり，さまざまな当てはめ問題に対して有効である．しかし，解が十分に収束するまでに多くの反復計算を必要とし，アウトライアの割合が大きい場合に正しい解を求めることが出来ない問題がある．

菅谷は連結したエッジ点列を対象として，RANSACを用いてアウトライアを含んだ点列から楕円弧に楕円を当てはめを行い，幾何学的MDLを用いた同一楕円上の楕円弧を統合することにより，楕円上の点列が分裂して得られた場合でも正しい楕円の形状を推定可能な手法を提案した [40]．この場合のアウトライアとは楕円弧以外の点列を示す．

Yuらは，連結したエッジ点列にFitzgibbonらの楕円限定当てはめ [5] を適用し，当てはめ誤差の大きい点を除去することを繰り返して，アウトライアを含んだ点列から楕円を検出する手法 [47] を提案した．Yuらの手法では，反復処理の中で当てはめ誤差が大きく楕円弧候補から除外された点はその後の処理の対象とならないため，反復回数が多いと最終的に得られる点列が短くなり，正しい楕円よりも小さな楕円が検出される傾向がある．

Shaoらは，連結した点列を一定の長さや条件により分割し，楕円の当てはめ誤差が小さくなる点列の組み合わせを総当りで探索する手法 [38] を提案した．Wangらは，点列を短い線分ごとに直線セグメントとして分割し，楕円がよく当てはまる点列の組み合わせを選択する手法 [48] を提案した．しかし，これらの手法は，経験的なしきい値などを用いて入力点列を細かな部分弧に分割

するため、必要以上に多くの部分弧に分割してしまったり、それにより、最適な点列の組み合わせの探索に時間がかかることがある。

1.2 目的

本研究では、入力画像より得られるエッジ点列の中から楕円弧を抽出し、得られた楕円弧点列より画像中の楕円を検出する手法を提案する。提案手法は処理効率と検出性能の2つの観点から改良を行い、楕円検出法の性能を向上させる。

まず、画像から楕円を検出する流れについて図 1.1 を参照しながら説明する。

はじめに入力画像から濃淡値の変化の激しい点をエッジ点列として抽出する。抽出したエッジ点列には楕円弧上のものや直線上のもの、楕円弧と直線が結合したものなど様々であるので、連結した点列ごとに、その中から楕円弧のみを抽出する処理を行う。これは、楕円弧以外の点列を含んだ点列からは楕円パラメータを正しく推定することが困難だからである。さらに、抽出した複数の楕円弧の中から同一の楕円上の点列を検出して統合する。最後に得られた楕円弧に楕円当てはめを行い、楕円パラメータを推定する。

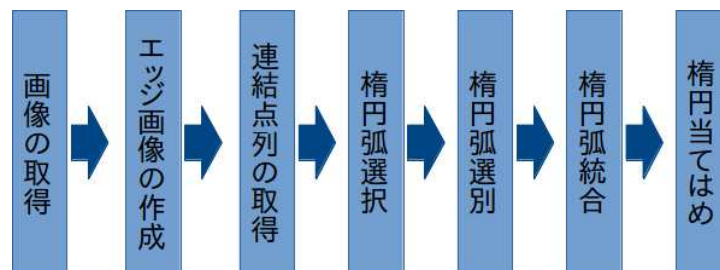


図 1.1: 楕円検出の流れ。

この楕円検出において、解決すべき問題点は大きく2つあり、まずは実際の入力画像には、ノイズや楕円弧以外の点列（これ以降アウトライアと呼ぶ）が含まれ、既存手法では正しい楕円を検出できない場合がある。

また、図 1.2 に示す拡張現実感システム [8] のようなリアルタイムで楕円を検出しなければならないシステムを実現するならば楕円検出法は高速であるほどよい。以上の問題を解決するため、

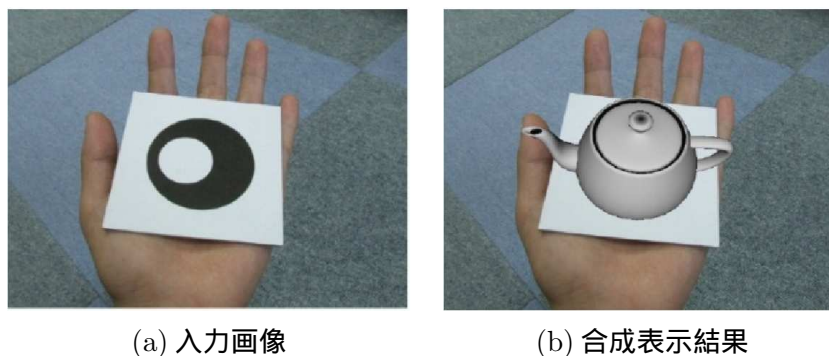


図 1.2: 円形マーカを用いた拡張現実感システム。

本論文では, ”楕円限定当てはめ”, ”楕円弧点列の選択”, ”楕円弧選別” についてそれぞれ新たな手法を示す. 各手法で想定する入力は異なり,

- 楕円弧点列の選択 :楕円弧点列にアウトライアが加わった点列を入力とする
- 楕円弧選別 :画像中のすべての楕円弧点列を入力とする
- 楕円限定当てはめ :楕円弧点列を入力とする

として, 問題を解決する. ここで, 本論文では, 楕円弧点列に加わるノイズと, 楕円以外の物体上の点列をアウトライアとして区別する.

本論文では, まず楕円当てはめの原理から提案した高精度な楕円限定当てはめについて 2 章で説明する. 次に楕円弧選択について 3 章で述べ, 4 章にて楕円弧点列の選別と統合処理を詳細を述べる. 5 章では提案した楕円検出法を用いた応用について述べる.

1.2.1 楕円限定当てはめ

楕円限定当てはめとは, 与えられた入力点列に楕円のみを当てはめる方法である. 本研究では, ノイズが楕円弧点列に含まれる場合でも高精度に楕円当てはめが可能な新たな手法を提案する.

楕円のみを当てはめる方法を最初に提案したものは Fitzgibbon ら [5] である. これは代数的距離の最小化であり, 単純な計算で楕円のみを当てはめることが可能である. Szpak ら [41] は楕円であるという制約を満たすようにサンプソン誤差を最小化する方法で, より精度の高い方法を発表した. しかし, 超精度くりこみ法の方がサンプソン誤差最小化より精度が高いことが確認されている [46], 超精度くりこみ法で楕円が当てはまればそれを解とするのが合理的である. 問題は, 楕円の解が得られなかった場合である. これに対し本研究では, 超精度くりこみ法の解の判定処理を加え, 楕円と判定された場合はその解を出力し, 楕円と判定されなかった場合はランダムサンプリングを用いた楕円のみを解とする手法を提案した [28].

本研究で提案する楕円検出では, 検出精度と検出効率それぞれの性能の向上を目的としている. しかし, 本節で述べた高精度な楕円限定当てはめは解を得るまでに多くの反復計算を必要とするために, 楕円検出処理には用いておらず, 既存の楕円限定当てはめを用いる. 応用として, 高精度な検出結果が必要となる場合には, 楕円弧の統合後の点列へ高精度な楕円限定当てはめを適用すればよい.

1.2.2 楕円弧点列の選択

提案した楕円限定当てはめ法はランダムに点列から点をサンプリングするのでサンプリングの無駄が多く, 解が収束するまでのサンプリング回数が多くなる. また, 基本的に入力点列に楕円弧点列以外の点が含まれることを考慮しておらず, 入力点列にアウトライアが含まれる場合に解の精度が低下する.

我々は, 楕円限定当てはめによって当てはめた楕円と入力点列の交点で点列を分割し, 当てはめ誤差の点列間での変化の傾向をもとに, 楕円当てはめに使用する点列の選択を繰り返し, 最終的に入力点列中の楕円弧点列のみを使用して楕円を当てはめる新たな手法を提案する. この手法では, 入力点列を当てはめた楕円との交点によって自動的に部分弧に分割するため, 経験的なしきい値などを必要としない. また, 選択した楕円弧に対して当てはめた楕円を用いた入力点列の再分割を繰り返すことで, 初期の部分弧分割が正しくなくても, 最終的に楕円弧上の部分弧が得られる.

ここで、楕円弧点列の選択に用いる楕円限定当てはめは、反復計算中で用いるため、高速に楕円の求まる Fitzgibbon ら [5] の手法を用いており、高精度な提案手法は用いていない。

1.2.3 楕円検出の高性能化

楕円上の全ての点列が一つに連結した入力点列として得られる状況は少なく、楕円弧点列が短い場合は、楕円の検出精度は低下する。この問題を解決するため、菅谷は同一楕円上の分裂または欠落して得られる楕円弧点列を統合して楕円を当てはめることで、より正しい形状の楕円を当てはめる方法を提案した [40]。しかし、画像中より得られる楕円弧点列の数が多くなればなるほど、楕円弧統合判定の組合せ数が多くなり、処理時間が増える。本研究では菅谷の同一楕円上の楕円弧点列の統合処理を採用し、楕円検出性能を向上させるとともに、楕円弧選別を行うことで、統合判定の組合せ問題を改善した楕円弧選別法を新たに提案する。

1.2.4 移動ロボットへの応用

移動ロボットの壁と床の境界認識システムへ、提案する楕円検出法を適用し、実際にリアルタイム処理での楕円検出の動作を確認し、その有効性を示す。

1.3 構成

本論文は以下の構成になっている。2章で楕円当てはめに必要なパラメータ表現と入力点に加わる誤差特性を説明し、既存の楕円当てはめと楕円限定当てはめ法について説明し、楕円限定当てはめの高精度化について検証した結果を示す。3章では、楕円検出のための楕円弧点列の選択法について述べ、4章では幾何学的 MDL を用いた同一楕円上の楕円弧点列の統合と提案手法による高性能化について述べる。5章では、楕円検出法を移動ロボットへ応用した壁と床の境界認識システムについて説明し、本研究の成果を示す。6章で本研究のまとめを述べる。また、付録にて既存の楕円当てはめ法、既存の楕円限定当てはめ法、移動ロボットのための距離推定法の詳細について述べる。

第2章 楕円限定当てはめ

この章ではまず、本研究の基礎である当てはめに用いる円錐曲線の式と入力点列に加わる誤差の特性の定義について説明し、楕円弧点列にノイズが加わった入力に対して、高精度な楕円限定当てはめを行う手法について述べる。

2.1 楕円当てはめ

本研究では得られた点列の座標情報より楕円を表す円錐曲線のパラメータを推定する処理を”楕円当てはめ”と呼ぶ。図 2.1 のように画像から抽出した点列に楕円を当てはめるとき、円錐曲線の 2 次式を用いて楕円を当てはめる。抽出した点列を用いた円錐曲線のパラメータの導出と、解が楕円を示すかの判定方法について説明する。

ここで、円錐曲線とは、円・楕円・放物線・双曲線およびその劣化(2直線など)を表す二次曲線であり、図 2.2 に示すような円錐の断面で表すことができる。

x, y の円錐曲線の 2 次式は

$$Ax^2 + 2Bxy + Cy^2 + 2f_0(Dx + Ey) + f_0^2F = 0 \quad (2.1)$$

で表され、式 (2.21) 中の f_0 はスケールを調節する定数である(本実験では $f_0 = 600$ とした)。式 (2.21) は楕円、放物線、双曲線、およびその退化(2直線など)を表す [15]。次の判定式が 0 より大きければ円錐曲線は楕円であるといえる。

$$AC - B^2 \begin{cases} > 0 & \text{楕円} \\ = 0 & \text{放物線} \\ < 0 & \text{双曲線} \end{cases} \quad (2.2)$$

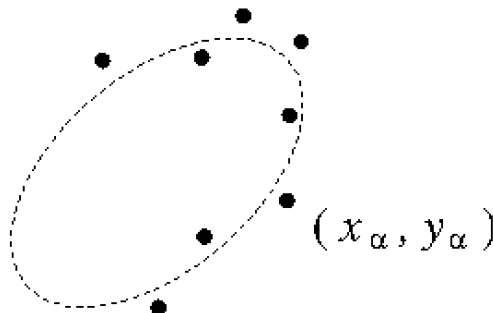


図 2.1: 点列に楕円を当てはめる。

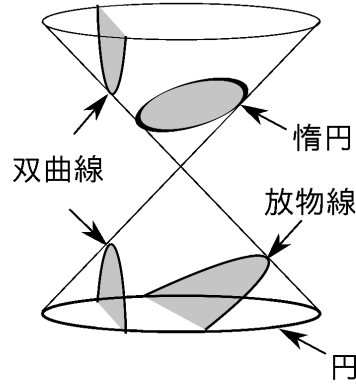


図 2.2: 円錐曲線の表現 .

しかし, 入力点にはノイズ (以下, 入力点に加わる誤差を「ノイズ」と呼ぶ) が含まれることを考慮すると, ある点列 $(x_1, y_1), \dots, (x_N, y_N)$ に楕円を当てはめることは $\alpha = 1, \dots, N$ に対して

$$Ax_\alpha^2 + 2Bx_\alpha y_\alpha + Cy_\alpha^2 + 2f_0(Dx_\alpha + Ey_\alpha) + f_0^2 F \approx 0 \quad (2.3)$$

となる係数 A, B, C, D, E, F を計算することである .

式を簡単化するため, 6次元ベクトル

$$\begin{aligned} \xi_\alpha &= (x_\alpha^2, 2x_\alpha y_\alpha, y_\alpha^2, 2f_0 x_\alpha, 2f_0 y_\alpha, f_0^2)^\top, \\ \theta &= (A, B, C, D, E, F)^\top \end{aligned} \quad (2.4)$$

を定義し, ベクトル a, b の内積を (a, b) と書けば式 (2.3) は,

$$(\xi_\alpha, \theta) \approx 0, \quad \alpha = 1, \dots, N \quad (2.5)$$

のように書ける .

θ には定数倍の不定性があるので, $\|\theta\| = 1$ と正規化できる .

2.2 入力点に加わる誤差と特性

誤差画像中から抽出した点列に楕円を高精度に当てはめる場合において重要な要因である . 本節では, 入力点列に加わる誤差の特性を考慮した正規化共分散行列の求め方を説明する .

図 2.1 で示されるようなデータ点 (x_α, y_α) は真の位置 $(\bar{x}_\alpha, \bar{y}_\alpha)$ に期待値 0, 標準偏差 σ の独立な正規分布に従うノイズ $\Delta x_\alpha, \Delta y_\alpha$ が加わると ξ_α の誤差は第 1 近似において次のようになる .

$$\begin{aligned} \Delta \xi_\alpha &= (2x_\alpha \Delta x_\alpha, 2\Delta x_\alpha y_\alpha + 2x_\alpha \Delta y_\alpha, 2y_\alpha \Delta y_\alpha, \\ &\quad 2f_0 \Delta x_\alpha, 2f_0 \Delta y_\alpha, 0)^\top \end{aligned} \quad (2.6)$$

仮定より $E[\Delta x] = E[\Delta y] = 0$, $E[\Delta x^2] = E[\Delta y^2] = \sigma^2$, $E[\Delta x \Delta y] = 0$ であるから, ξ_α の共分散行列は式 (2.7) のようになる .

$$V[\xi_\alpha] \equiv E[\Delta \xi_\alpha \Delta \xi_\alpha^\top] = \sigma^2 V_0[\xi_\alpha] \quad (2.7)$$

ただし, $V_0[\xi_\alpha]$ を式 (C.3) のように置き, これを「正規化共分散行列」と呼ぶ [14].

$$V_0[\xi_\alpha] = 4 \begin{pmatrix} x_\alpha^2 & x_\alpha y_\alpha & 0 & f_0 x_\alpha & 0 & 0 \\ x_\alpha y_\alpha & x_\alpha^2 + y_\alpha^2 & x_\alpha y_\alpha & f_0 y_\alpha & f_0 x_\alpha & 0 \\ 0 & x_\alpha y_\alpha & y_\alpha^2 & 0 & f_0 y_\alpha & 0 \\ f_0 x_\alpha & f_0 y_\alpha & 0 & f_0^2 & 0 & 0 \\ 0 & f_0 x_\alpha & f_0 y_\alpha & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (2.8)$$

理論的にはこれは真の位置 $(\bar{x}_\alpha, \bar{y}_\alpha)$ で評価すべきであるが, 実験によると観測値 (x_α, y_α) で評価しても結果に差がない. またこれは $\Delta x_\alpha, \Delta y_\alpha$ の第 1 次近似に基づいているが, 2 次以上の項を考慮しても最終結果に影響がないことが知られている.

2.3 従来の楕円当てはめ法の問題点

前述した楕円当てはめについて, 既存の楕円当てはめの方法とそれらの問題点について説明する.

楕円当てはめでは, 当てはめる楕円と点列との距離の二乗和 (再投影誤差) を最小にするよう計算する方法と 0 になるべき式 (2.3) の円錐曲線の式の二乗和 (代数的距離) を最小にするよう計算する方法がある. また, 再投影誤差や代数的距離を最小化するのではなく, 誤差解析によって精度がなるべく高くなるように計算方式を定める方法がある.

ここで, 再投影誤差とは, サンプソン誤差 [7] という関数で式 (C.3) で求めた正規化共分散行列を用いて式 (2.9) で近似される.

$$J = \frac{1}{N} \sum_{\alpha=1}^N \frac{(\xi_\alpha, \theta)^2}{(\theta, V_0[\xi_\alpha] \theta)} \quad (2.9)$$

上記の楕円当てはめ法では, 超精度くりこみ法が最も優れた手法と言われる [20, 21, 46]. しかし, これらの楕円当てはめ法は式 (2.2) を考慮せず円錐曲線の式のパラメータを求めるものであり, 入力点列に加わるノイズが大きくなると図 2.3 のような楕円以外が計算されることがある. また, 実際の画像データを用いた場合は式 (2.2) の左辺が厳密に 0 になることはないので双曲線が主に当てはまる. 既存の楕円当てはめの詳細については付録の 7.1 節にて説明する.

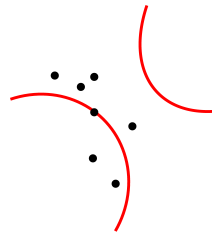


図 2.3: 双曲線が当てはまる例.

2.4 楕円限定当てはめ

本節では, 前述した楕円当てはめ法の問題について, 既存の楕円限定当てはめ法と提案するランダムサンプリングを用いた手法について説明する.

楕円当てはめの問題点を解決する手法として提案された手法が, Fitzgibbon らの方法 [5] である. Fitzgibbon らの手法は, 代数的距離の最小化であって計算は単純であるが精度が低い. より高精度な手法として Szpak らの方法 [41] が挙げられる. これらの詳細については付録の 7.2 節にて説明する. 本研究では超精度くりこみ法の問題点を解決し, Szpak らの手法よりも精度の高い楕円限定当てはめ法を提案する.

2.4.1 ランダムサンプリングを用いた楕円限定当てはめ

本研究では, 既存の楕円当てはめ法で最も精度が高いとされる超精度くりこみ法を採用する. 超精度くりこみ法では楕円を当てはめられない場合がある問題を解決するため, ランダムサンプリングを組み合わせた新たな手法を提案する.

まず, 図 2.4 (a) のようなエッジ画像中の赤色で示した点列を入力として選択すると, 選択した点列に対して超精度くりこみ法を用いた楕円当てはめを行う. 超精度くりこみ法の解が楕円であるかどうかの判定を行い, 解が楕円であった場合は処理を終了する. 図 2.4 (b) の青で示したような楕円以外の解であった場合, ノイズの大きく加わった点を除去して楕円当てはめを行うために, 点の分布を考慮したランダムサンプリングによる楕円当てはめを行う. 最終的に図 2.4 (c) のような楕円が当てはまった結果を出力する. 提案手法の具体的な手順を次に示す.

1. 入力点列 (x_α, y_α) , $\alpha = 1, \dots, N$ に超精度くりこみ法を用いて楕円を当てはめ, そのパラメータ θ を計算する.
2. θ が式 (2.2) を満たせば, それを解として終了する.
3. 超精度くりこみ法の反復が一定回数 (実験では 100 回を上限とした) 以内に収束しない, あるいは収束した解が式 (2.2) を満たさなければ, 点の分布を考慮したランダムサンプリングによる楕円当てはめを用いてパラメータ θ を計算する.

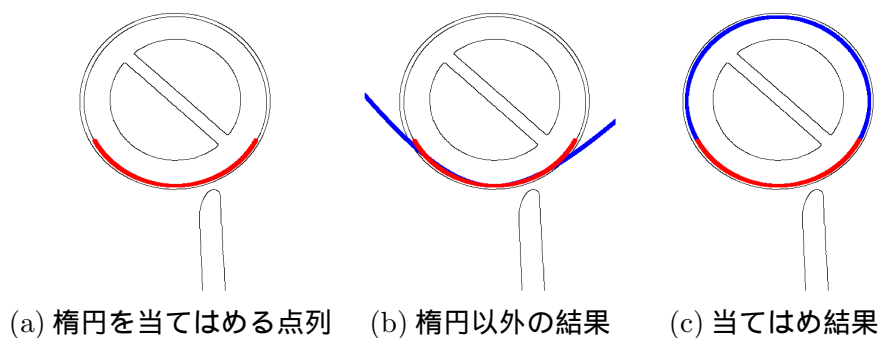


図 2.4: 楕円限定当てはめの実行例.

ランダムサンプリングによる楕円当てはめは, 入力点列の中からランダムに選択した点を用いて, 楕円当てはめを繰り返し最適な楕円当てはめ結果になるまで点の選択と当てはめの反復計算を行う手法であり, ノイズにロバストな楕円当てはめが可能である. 代表的な方法として用いられる, RANSAC[43] では様々な幾何学的当てはめに用いることができるが, ランダムに点列から

点を選択するので解の収束に必要な解の収束回数が多くなる．ここで収束回数とは，最適な楕円を当てはめるまでに行った入力点列のサンプリングの回数を示す．この RANSAC の問題に対して，当てはめる点列中の信頼度が高い点を優先してサンプリングし，収束回数の削減を行った手法が PROSAC[10] である．これに対し本研究では，サンプリングする点の分布を考慮することにより解の収束回数を削減する方法を提案する．本手法では解の収束回数を削減することにより，将来的な計算コストの削減を目指す．提案手法では，サンプリングする 5 点中 3 点を入力点列中の中央と両端付近より入力点ごとの相対距離を用いてそれぞれサンプリングする．これに対し，本研究では提案手法の有効性を確認するため，提案手法のサンプリング方法を，入力点ごとの相対距離に加えて正規乱数を用いて 3 点をサンプリングした場合と，入力点列を 5 等分したそれぞれの区間より 1 点ずつランダムにサンプリングした場合の方法についても比較する．

次節より，ランダムサンプリングの当てはめ方法として，既存手法である RANSAC と PROSAC と，提案手法の楕円当てはめの手順を説明し，正規乱数を用いたサンプリング法と入力点列を 5 分割したサンプリング法についての説明を行う．

2.5 RANSAC

RANSAC を用いた楕円当てはめの手順を次に示す．

1. 最小のサンブソン誤差を J_{\min} と定義し， $J_{\min} = \infty$ とする．
2. 入力点列の中からランダムに 5 点 (x_{α}, y_{α}) ， $\alpha=1, \dots, 5$ を選択する．
3. 選択した 5 点より，式 (2.4) のように ξ_{α} を作成し，行列 M を式 (2.17) で計算する．

$$M = \sum_{\alpha=1}^5 \xi_{\alpha} \xi_{\alpha}^{\top} \quad (2.10)$$

4. 行列 M の最小固有値に対する単位固有ベクトル θ を計算する．
5. θ が式 (2.11) を満たさない場合，手順 1 にもどる．

$$(\theta, N_F \theta) > 0 \quad (2.11)$$

6. 全ての入力点について当てはめた楕円とのサンブソン誤差 J を式 (2.12) より求める．

$$J = \frac{1}{N} \sum_{\alpha=1}^N \frac{(\xi_{\alpha}, \theta)^2}{(\theta, V_0[\xi_{\alpha}] \theta)} \quad (2.12)$$

7. $J_{\min} > J$ のとき J_{\min} を J に更新し，このときの θ を解候補とする．
8. 手順 2 から手順 7 を繰り返し J_{\min} が既定回 (本研究では 100 回) 更新されなければ，この時点での解候補になっている θ を解として処理を終了する．

楕円当てはめには最低 5 点の点を用いるの必要があり，RANSAC を用いた楕円当てはめでは，ランダムサンプリングにより得られる入力点列中の 5 点を用いる．ランダムにサンプリングした 5 点を用いて楕円を当てはめ，サンプリングした点の位置と当てはまった楕円について図 2.5 に示

す。図中の黒い楕円が真の楕円，選択した点を用いて最小二乗法により楕円を当てはめた結果を青色で示す。この例では楕円上の 30 点の入力点列に対し，期待値 0，標準偏差 1.05 を独立に加えた。図中では ■ がサンプリングされた入力点，■ がその他の入力点を示す。

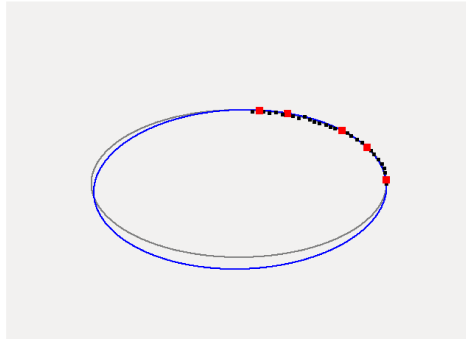
図 2.5 (a) のような点が選択された場合は真の形状に近い楕円が当てはまる。しかし，図 2.5 (b), 図 2.5 (c) のような入力点列中の偏った位置にある点群が選択された場合，最適解とは遠い楕円が当てはまり，収束するまでの繰り返し回数が増える。

2.6 PROSAC

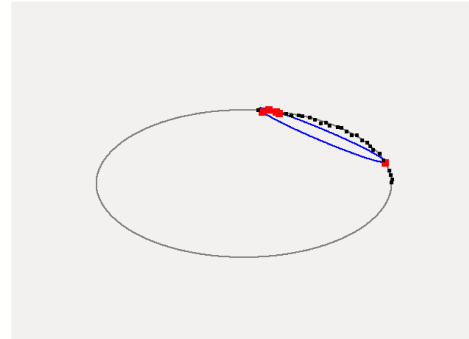
PROSAC は，楕円の当てはめ方は RANSAC と同様だが，サンプリングの方法が異なる。PROSAC では，当てはめる楕円との Sampson 誤差が小さい点を信頼度の高い点として，信頼度の高い点を優先して選択して当てはめを行っていくことでサンプリング回数を減らす。

次に PROSAC の楕円当てはめでのランダムサンプリング手順を示す。

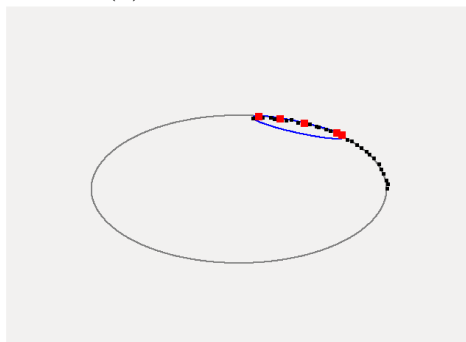
1. N 点の入力点からランダムに抽出した 5 点を用いて，楕円当てはめを行う。
2. 選択した 5 点を点群 U とする。
3. 全ての点に対して Sampson 誤差を式 (2.12) により計算し，Sampson 誤差が小さい順にソートする。
4. 点群 U の中からランダムに 4 点と，点群 U に含まれない Sampson 誤差が最も小さい点を用いて楕円を当てはめる。



(a) 点列全体から選択



(b) 両端から選択



(c) 片端から選択

図 2.5: ランダムサンプリングした点と当てはめ結果。

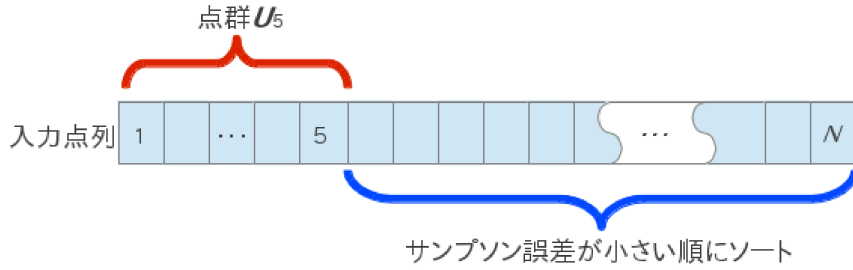


図 2.6: 点群 U_α の作成

5. 手順 3 から手順 4 を T 回目のサンプリングまで繰り返す．終了条件を満たしたら終了する．
6. 点群 U に含まれないサンプルソン誤差が最も小さい点を追加し，サンプリング回数 T を更新して，手順 4 に戻る．
7. 全ての入力点が点群 U に追加された場合，RANSAC と同様の処理を行う．

サンプリング回数 T の更新と点群 U の成長については，次で詳しく説明する．

2.6.1 点群の成長とサンプリング回数の更新

まず PROSAC では，図 2.6 のようにランダムに選択した 5 点を点群 U_5 として，この 5 点を用いて楕円を当てはめる．当てはめた楕円とのサンプルソン誤差が小さい順に点群 U_5 に含まれない点列をソートする．次に PROSAC では，図 2.7 (a) のように点群 $U_{\alpha-1}$ 中からランダムに選択された 4 点と， t 回目のサンプリングでの $U_{\alpha-1}$ に含まれないサンプルソン誤差が最も小さい入力点 $g(t)$ を用いて楕円当てはめを行う．ここで， α 点で構成される点群を U_α と示す．

この α を増加させて図 2.7 (b) のように点群 U_α を成長させる．サンプリングを行う毎に t を増加させ， T'_α 以上の回数になるまで α 値を変更せずにサンプリングと楕円当てはめを繰り返す．ここで， T'_α の初期値は 1 として与えられ， α を増加させ点群を成長させた場合に $T'_\alpha = T'_{\alpha+1}$ と更新される． $T'_{\alpha+1}$ は式 (2.15) で更新される．

$$T'_{\alpha+1} = T'_\alpha + \lceil T_{\alpha+1} - T_\alpha \rceil \quad (2.13)$$

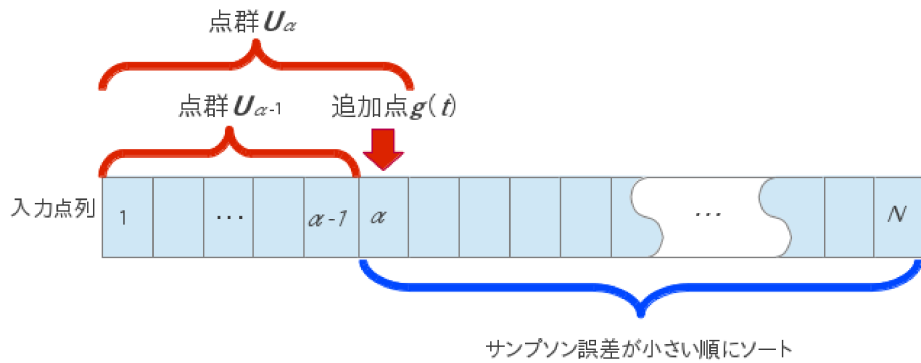
$$T_\alpha = T_N \prod_{i=0}^{\alpha-1} \frac{\alpha - i}{N - i} \quad (2.14)$$

$$T_{\alpha+1} = \frac{\alpha + 1}{\alpha + 4} \quad (2.15)$$

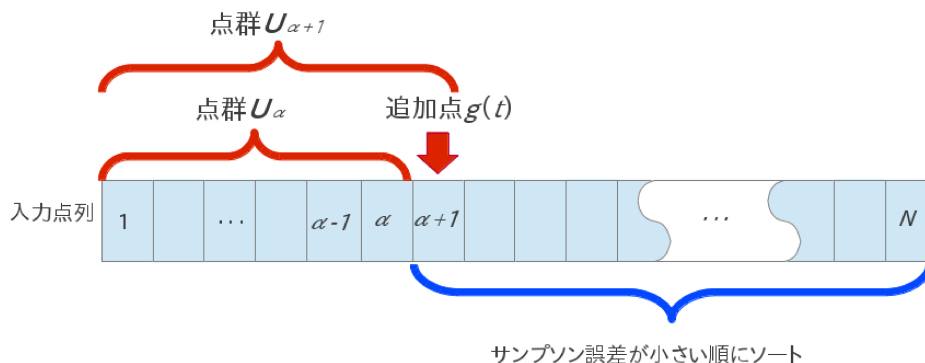
ここで， T_N とは，RANSAC が全ての入力点をサンプリングするのに必要な反復計算の回数の見込みを示す．本研究では， $T_N = 10$ とする．また， $\lceil x \rceil$ は天井関数であり，実数 x 以上の最小の整数を示す．

式 (2.15) より，ソートした α 番目の入力点に対して， T'_α 回のサンプリングと楕円当てはめを繰り返したあと $\alpha \leftarrow \alpha + 1$ として，点群 U_α を成長させる．

α が N 点以上になった場合は，RANSAC と同様に入力点中からランダムに 5 点選択する．



(a) U_α の選択点



(b) U_α の成長後

図 2.7: PROSAC の点群 U の成長 .

2.7 点の分布を考慮したランダムサンプリング

提案手法の点の分布を考慮したランダムサンプリングを用いた楕円当てはめでは, RANSAC[43] と同等の当てはめ結果をより少ない収束回数で求める .

そこで, 図 2.5 から経験的に入力点列中の偏りの無い位置から選択された点によって楕円が当てはめられた場合に, 最適な楕円の当てはめ結果が得られやすいと仮定し, 図 2.8 で示すような入力点列中の 3 つの赤丸領域よりそれぞれ 1 点ずつ選択した後, ランダムで 2 点選択する .

提案手法の具体的な手順を次に示す .

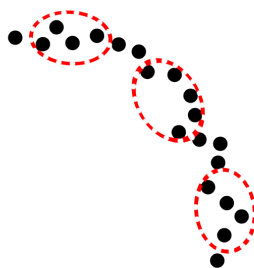


図 2.8: 3 点の選択領域 .

1. 最小のサンプルソン誤差を J_{\min} と置き, $J_{\min} = \infty$ とする.
2. ランダムに 1 点目の点 a を選択する.
3. 注目する点 a の座標を (x, y) とし, 他の入力点 (x_α, y_α) , $\alpha = 1, \dots, N-1$ (ただし, α は注目する点を除く) との距離 d_α を式 (2.16) で計算し, 昇順に d_α をソートする.

$$d_\alpha = \sqrt{(x - x_\alpha)^2 + (y - y_\alpha)^2} \quad (2.16)$$

4. 点 a から最も遠い点との距離を d_A とし, 昇順にソートされた d_α の α を 1 から増加させて, $d_A \times 0.8 < d_\alpha$ を初めて満たした点を 2 点目として選択し, 点 b とする.
5. 注目する点 b の座標を (x, y) とし, 式 (2.16) を用いて他の入力点との距離 d_α を計算し, 昇順に d_α をソートする.
6. 点 b から最も遠い点との距離を d_B とし, 昇順にソートされた d_α の α を 1 から増加させて, $d_B \times 0.8 < d_\alpha$ を初めて満たした点を 3 点目として選択し, 点 c とする.
7. 点 ac 間の距離を式 (2.16) と同様に求め, 予め決定したしきい値よりも小さい場合 (実験では 10 画素以下) には, 点 a について作成し, 昇順にソートした d_α の α を 1 から増加させて, $d_A \times 0.5 < d_\alpha$ を初めて満たした点を点 c とする.
8. 残り 2 点を a, b, c 点を除いた入力点列の中からランダムに選択する.
9. 選択した 5 点の座標をそれぞれ (x_β, y_β) , $\beta = 1, \dots, 5$ とし, 式 (2.4) のように ξ_β を作成し, 行列 M を式 (2.17) で計算する.

$$M = \sum_{\beta=1}^5 \xi_\beta \xi_\beta^\top \quad (2.17)$$

10. 行列 M の最小固有値に対する単位固有ベクトル θ を計算する.
11. θ が式 (2.18) を満たさない場合, 手順 2 にもどる.

$$(\theta, N_F \theta) > 0 \quad (2.18)$$

12. 全ての入力点について当てはめた楕円とのサンプルソン誤差 J を式 (2.19) より求める.

$$J = \frac{1}{N} \sum_{\alpha=1}^N \frac{(\xi_\alpha, \theta)^2}{(\theta, V_0[\xi_\alpha] \theta)} \quad (2.19)$$

13. $J_{\min} > J$ のとき J_{\min} を J に更新し, このときの θ を解候補とする.
14. 手順 2 から手順 13 を繰り返し J_{\min} が既定回 (本研究では 100 回) 更新されなければ, この時点での解候補になっている θ を解として処理を終了する.

提案手法では，点 a をランダムで選択した後，最も遠い点との距離が 8 割以上の点を選択するので，点 b は入力点列のどちらかの端点付近の点を選択される．また，点 b も同様に最も距離の遠い点との距離が 8 割以上の点を選択するので，点 b とは逆の端点付近の点を選ばれる．このとき，始めにランダムで選択された点 a が，端点付近の点だった場合については，点 c は点 a と点 b の中点付近の点に再設定されるため図 2.8 で示されるような領域からのサンプリングが見込める．提案手法では，入力点列の分布を考慮した 3 点を用いることで，図 2.5 (b), 図 2.5 (c) のように入力点列に沿わない楕円が当てはまりにくく，ランダムに 5 点選択するよりも少ない収束回数で，サンプソン誤差が最小の楕円の解を出力する．

2.8 正規乱数を用いたサンプリング法

N 点の入力点列に対して，期待値 μ を分布を考慮して選択し，標準偏差 σ の正規分布に従う乱数を用いて 3 点をサンプリングする手法について， μ の決定とサンプリング方法について次に説明する．

1. 入力点列 (x_α, y_α) , $\alpha = 1, \dots, N$ に対して x 座標の値が最小の点を点 A と置く．
2. 点 A の座標を (x_1, y_1) とし，式 (2.20) より点 A と入力点列 (x_α, y_α) との距離 d_α を求め，距離が近い順に入力点列をソートする．

$$d_\alpha = \sqrt{(x_1 - x_\alpha)^2 + (y_1 - y_\alpha)^2} \quad (2.20)$$

3. ソートした点列中から (点 A から最も遠い点との距離)/2 の距離に近い β_1 番目の点を求める．
4. ソートした点列中から期待値 β_1 , 標準偏差 $N/6$ の正規分布に従う乱数 γ_1 を求め， $[\gamma_1]$ 番目の点を 1 点目として選択する． $[\gamma_1] > N$ または $[\gamma_1] < 1$ ならば再選択する．
ここで， $[A]$ は天井関数を示し，実数 A に対して A 以上の最小の整数を示す．
5. ソートした点列中から (点 A から最も遠い点との距離)/5 の距離に近い β_2 番目の点を求める．
6. ソートした点列中から期待値 β_2 , 標準偏差 $N/6$ の正規分布に従う乱数 γ_2 を求め， $[\gamma_2]$ 番目の点を 2 点目として選択する．既に選択している点か， $[\gamma_2] > N$ または $[\gamma_2] < 1$ ならば再選択する．
7. ソートした点列中から (点 A から最も遠い点との距離) $\times 4/5$ の距離に近い β_3 番目の点を求める．
8. ソートした点列中から期待値 β_3 , 標準偏差 $N/6$ の正規分布に従う乱数 γ_3 を求め， $[\gamma_3]$ 番目の点を 3 点目として選択する．既に選択している点か， $[\gamma_3] > N$ または $[\gamma_3] < 1$ ならば再選択する．
9. 選択した 3 点を除いた入力点列中から，一様乱数を用いて残り 2 点を選択する．

点の分布を考慮してサンプリングする入力点列中の 3 点は， $\sigma = N/6$ とした場合， $\mu + N/2$ から $\mu - N/2$ 番目までの入力点を選択される確率は 99.73% , $\mu + N/6$ から $\mu - N/6$ 番目までの入力点を選択される確率は 68.27% で選択され， $\sigma = N/18$ とした場合は， $\mu + N/6$ から $\mu - N/6$ 番

目までの入力点が選択される確率は 99.73% , $\mu + N/18$ から $\mu - N/18$ 番目までの入力点が選択される確率は 68.27% の確率で選択される . 図 2.9 で示す楕円上の赤色で示す 30 点の入力点列に対し , 標準偏差 σ を $N/6$ と $N/18$ の場合でそれぞれ 10000 回サンプリングを行い 1 から 30 番目の点がそれぞれ何回選ばれるかその傾向を図 2.10 に示す . 図の縦軸はそれぞれの点がサンプリングされた割合 , 横軸は x 値が小さい端点からの距離近い順にソートされた α 番目の入力点を示す .

図 2.10 では , 1 点目 , 2 点目 , 3 点目にサンプリングされる点の期待値 μ はそれぞれ 15 番目 , 6 番目 , 23 番目の点を選ばれる . 図 2.10 より , 正規分布に従いそれぞれの点を選択されていることが確認できる .

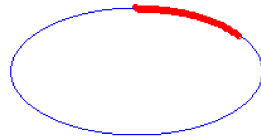


図 2.9: 入力点列 .

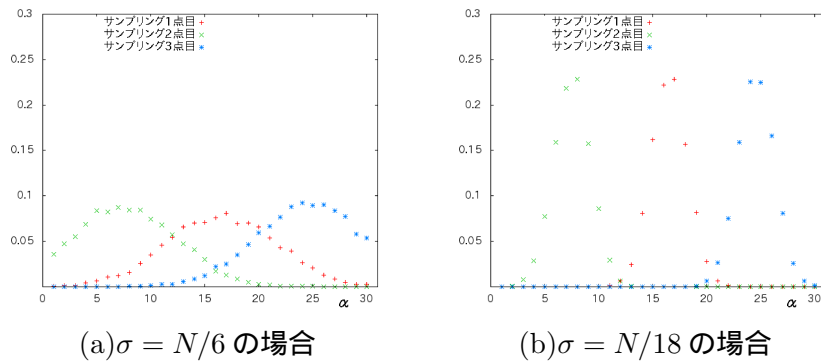


図 2.10: 1 点目 2 点目 3 点目がそれぞれサンプリングされる割合 .

2.9 入力点列を5等分するサンプリング法

サンプリングする5点を N 点からなる入力点列を5等分した区間より、一様乱数を用いてそれぞれ選択する。

次にサンプリング手順を示す。

1. 入力点列 (x_α, y_α) , $\alpha = 1, \dots, N$ に対して x 値が最小の端点を点 A と置く。
2. 点 A の座標を (x_1, y_1) とし、式 (2.21) より点 A と入力点列 (x_α, y_α) との距離 d_α を求め、距離が近い順に入力点列をソートする。

$$d_\alpha = \sqrt{(x_1 - x_\alpha)^2 + (y_1 - y_\alpha)^2} \quad (2.21)$$

3. 0 から $N/5$ までの一様乱数 γ を求め、 $\lceil \gamma \rceil$ 番目の点を1点目として選択する。
4. $N/5$ から $2N/5$ までの一様乱数 γ を求め、 $\lceil \gamma \rceil$ 番目の点を2点目として選択する。
5. $2N/5$ から $3N/5$ までの一様乱数 γ を求め、 $\lceil \gamma \rceil$ 番目の点を3点目として選択する。
6. $3N/5$ から $4N/5$ までの一様乱数 γ を求め、 $\lceil \gamma \rceil$ 番目の点を4点目として選択する。
7. $4N/5$ から N までの一様乱数 γ を求め、 $\lceil \gamma \rceil$ 番目の点を5点目として選択する。
8. 同じ点を選択していないか調査し、選択していたら手順2に戻る。

2.10 実験

この節では、まず実験環境と必要なパラメータについて説明する。次にシミュレーション実験により、既存手法の問題点の解決の確認として、超精度くりこみ法の解が楕円以外が当てはまる割合と提案手法の RMS 誤差の統計的な比較を行う。また、提案手法である点の分布を考慮したランダムサンプリングの有効性を確認するため、提案手法と、正規乱数を用いたサンプリング法、入力点列を5等分するサンプリング法の当てはめ結果を比較した後、超精度くりこみ法と組み合わせた RANSAC と PROSAC、提案手法の解の収束回数の比較実験を行う。さらに、楕円限定当てはめ法としての提案手法の有用性を確認するため、シミュレーションデータと実画像を用いて提案手法と Fitzgibbon らの方法と Szpak らの方法の楕円当てはめ結果を比較する。

本節では、実験を行った環境と各パラメータを示す。また、統計的な手法比較の際に用いる精度の理論限界である KCR 下界、RMS (平方平均二乗) 誤差、相対ノイズレベルについて説明する。

2.10.1 実験環境

次の実験環境で行った。

- Ubuntu 12.04 LTS
- Intel(R) Core2 Duo CPU E6850 @ 3.00GHz × 2
- メモリ 4GB

2.10.2 パラメータ

実験で用いた各手法は次のパラメータに設定した．

- 画像サイズ：640 × 480
- 超精度くりこみ法で解が収束しない場合の反復計算の上限：100 回
- RANSAC , PROSAC , 提案手法のランダムサンプリングによる楕円当てはめの終了条件： θ が 100 回更新されない
- PROSAC の反復回数設定用パラメータ： $T_N = 10$
- 提案手法のランダムサンプリングの点 a と点 c の距離が近すぎないか判定する閾値：10 画素

2.10.3 シミュレーション用点列データ

シミュレーション実験に用いる楕円のパラメータと用いる点列を示す．

楕円

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (2.22)$$

の弧長 (始点は (1,0)) の区間 $[s_0, s_1]$ を $N - 1$ 等分する N 点 $(\bar{x}_\alpha, \bar{y}_\alpha)$, $\alpha = 1, \dots, N$ をとる．図 2.11 に作成した点列を で示す．楕円の形状の変化させた入力データとして図 2.11 (a)~(c) , 入力データ (b), (c) に対して点列の位置を変えた入力データとして図 2.11 (d), 図 2.11 (e) のように作成した．

(a) $a = 100, b = 100, N = 30, [s_0, s_1] = [0, s(55^\circ)]$

(b) $a = 50, b = 100, N = 30, [s_0, s_1] = [0, s(55^\circ)]$

(c) $a = 50, b = 200, N = 30, [s_0, s_1] = [0, s(55^\circ)]$

(d) $a = 50, b = 100, N = 15, [s_0, s_1] = [s(60^\circ), s(100^\circ)]$

(e) $a = 50, b = 200, N = 15, [s_0, s_1] = [s(60^\circ), s(100^\circ)]$

ただし $s(\phi)$ は始点から測った偏角 ϕ の点までの弧長である (実験では弧長は数値積分で評価した) . 隣接点間の平均距離は図 2.11 に記入したようになる . これらの点列の各点の x, y 座標に独立に期待値 0 , 標準偏差 σ の正規分布に従う誤差を加えた点 (x_α, y_α) , $\alpha = 1, \dots, N$ に楕円を当てはめる .

2.10.4 RMS 誤差と KCR 下界

当てはめた楕円の精度を表す計算値 θ の精度を評価する RMS 誤差とそれに伴う精度の理論限界である KCR 下界について説明する . まず , θ の符号が不定であるため $(\theta, \bar{\theta}) < 0$ であれば , $\theta \leftarrow -\theta$ として符号を揃える . RMS 誤差は θ と真値 $\bar{\theta}$ は共に単位ベクトルであるから , その誤差 $\Delta\theta$ を θ の $\bar{\theta}$ に垂直な成分

$$\Delta^\perp\theta = P_{\bar{\theta}}\theta \quad (2.23)$$

で測る (図 2.12) . ただし $P_{\bar{\theta}}$ は $\bar{\theta}$ に垂直な空間への射影行列である [15] .

$$P_{\bar{\theta}} \equiv I - \bar{\theta}\bar{\theta}^\top \quad (2.24)$$

そしていろいろな σ に対して本実験では 10000 回独立に試行し , 次の RMS(平方平均二乗) 誤差 D を計算する .

$$D = \sqrt{\frac{1}{10000} \sum_{a=1}^{10000} \|\Delta^\perp \theta^{(a)}\|^2} \quad (2.25)$$

ここで $\theta^{(a)}$ は a 回目の試行の解である .

また , RMS 誤差の理論限界を表す KCR 下界 [14, 19] を , 真値を用いた行列

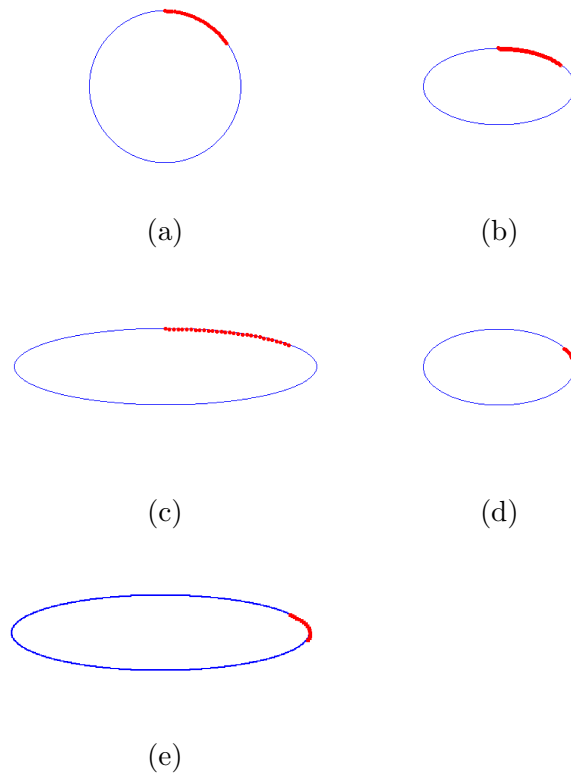


図 2.11: 実験に用いた楕円上の点列 .

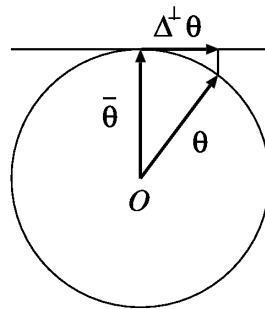


図 2.12: 計算値 θ の真値 $\bar{\theta}$ に垂直な成分 $\Delta^\perp \theta$.

$$M_{KCR} = \frac{1}{N} \sum_{\alpha=1}^N \bar{W}_\alpha \bar{\xi}_\alpha \bar{\xi}_\alpha^\top, \quad \bar{W}_\alpha = \frac{1}{(\bar{\theta}, V_0[\xi_\alpha] \bar{\theta})} \quad (2.26)$$

の固有値 $\mu_1 \geq \dots \geq \mu_6 (= 0)$ を計算する．式中の $V_0[\xi_\alpha]$ は真の値 $(\bar{x}_\alpha, \bar{y}_\alpha)$ を用いて計算する．このとき，RMS 誤差 D の KCR 下界は次のようになる．

$$D_{KCR} = \frac{\sigma}{\sqrt{N}} \sqrt{\frac{1}{\mu_1} + \dots + \frac{1}{\mu_5}} \quad (2.27)$$

2.10.5 相対ノイズレベル

入力点列に加わるノイズを増加させたときの RMS 誤差のグラフを描画する場合，入力点列の密度による解のスケールの違いを統一するため，ノイズの標準偏差 σ の隣接点間の平均距離の比，すなわち隣接点間の平均距離を 1 とする尺度で測ったものを「相対ノイズレベル」とする．

図 2.11 (a),(b),(c),(d),(e) の隣接点間平均距離はそれぞれ 2.96, 3.31, 5.72, 2.72, 3.44 であり，これで標準偏差 σ を割った相対ノイズレベル ϵ が RMS 誤差の横軸とする．

2.11 超精度くりこみ法と解の判定

本実験では，入力点列にノイズを加えていったときの超精度くりこみ法が双曲線を当てはめる割合を調査し，提案手法と超精度くりこみ法の当てはめ精度を比較することにより，提案手法の有用性を確かめる．そこで，図 2.11 で示す点列に対し，期待値 0，標準偏差 σ の正規分布に従う誤差を加えた点 (x_α, y_α) ， $\alpha = 1, \dots, N$ に超精度くりこみ法と提案手法を用いて楕円を当てはめ，いろいろな σ に対して 10000 回独立に試行した．また，超精度くりこみ法は双曲線が当てはまる場合も RMS 誤差を計算した．

図 2.13 にそれぞれの入力点列に対応した結果を示す．図の左列に RMS 誤差，右列に超精度くりこみ法で双曲線が当てはまる割合を示す．グラフの横軸は相対ノイズレベルを示し，RMS 誤差のグラフ中に KCR 下界も示す．

図 2.13 より，超精度くりこみ法はノイズが小さいときは楕円が当てはまり，KCR 下界に近い非常に高精度な結果を示していることが確認できる．しかし，図 2.13 の右列のグラフより，入力点列に加わるノイズが大きくなるほど双曲線の当てはまる割合が大きくなり，RMS 誤差が大きくなっていることを確認した．また，図 2.13 (a), (b), (c) の双曲線の当てはまる割合を比較すると，楕円が細くなるほど双曲線が当てはまり易くなっており，RMS 誤差の計測値も細い楕円ほど低くなっている．よって，細い楕円ほど提案手法が有効であるといえる．

図 2.13 (d) のような曲率が高い点列では，曲率が低い点列よりも，超精度くりこみ法，提案手法共に精度が落ちてしまうが，しかし図 2.13 (e) では，精度が大きく向上している．このことから，超精度くりこみ法，提案手法では，細い楕円の曲率の高い部分の点列が取得できれば高精度に楕円を当てはめられる傾向があることが分かった．これは，超精度くりこみ法でノイズが大きく加わった入力点列に楕円を当てはめる場合，真の楕円よりも小さい楕円が当てはまる傾向があり，入力点列全てに沿うように楕円を当てはめるので細い形状が当てはまる．このことより，真値が真円と細長い楕円の場合は，細長い楕円の方が形状が似ているため精度が向上したと考えられる．

また，ノイズが大きくなったときに KCR 下界を計測値が下回っている．提案手法では，ランダムサンプリングを用いているため入力点列に大きくノイズが加わった点をサンプリングしなけ

れば高精度に楕円を当てはめることができるため、KCR 下界を下回ることは十分考えられる。超精度くりこみ法が KCR 下界を下回ってしまった原因は、RMS 誤差の導出には図 2.12 で示すように単位ベクトルの誤差で評価しているため、誤差の上限が存在する。そのため、統計的に RMS 誤差を評価した場合に、ノイズが大きくなると KCR 下界を下回る結果が求まる。

提案手法は、超精度くりこみ法で双曲線が当てはまる割合が増加するとともに、RMS 誤差の方が超精度くりこみ法よりも低くなっていることを確認した。このことより、提案手法はノイズが大きい入力点列に対し、超精度くりこみ法の問題点を解決し、高精度に楕円を当てはめる手法として有効であるといえる。

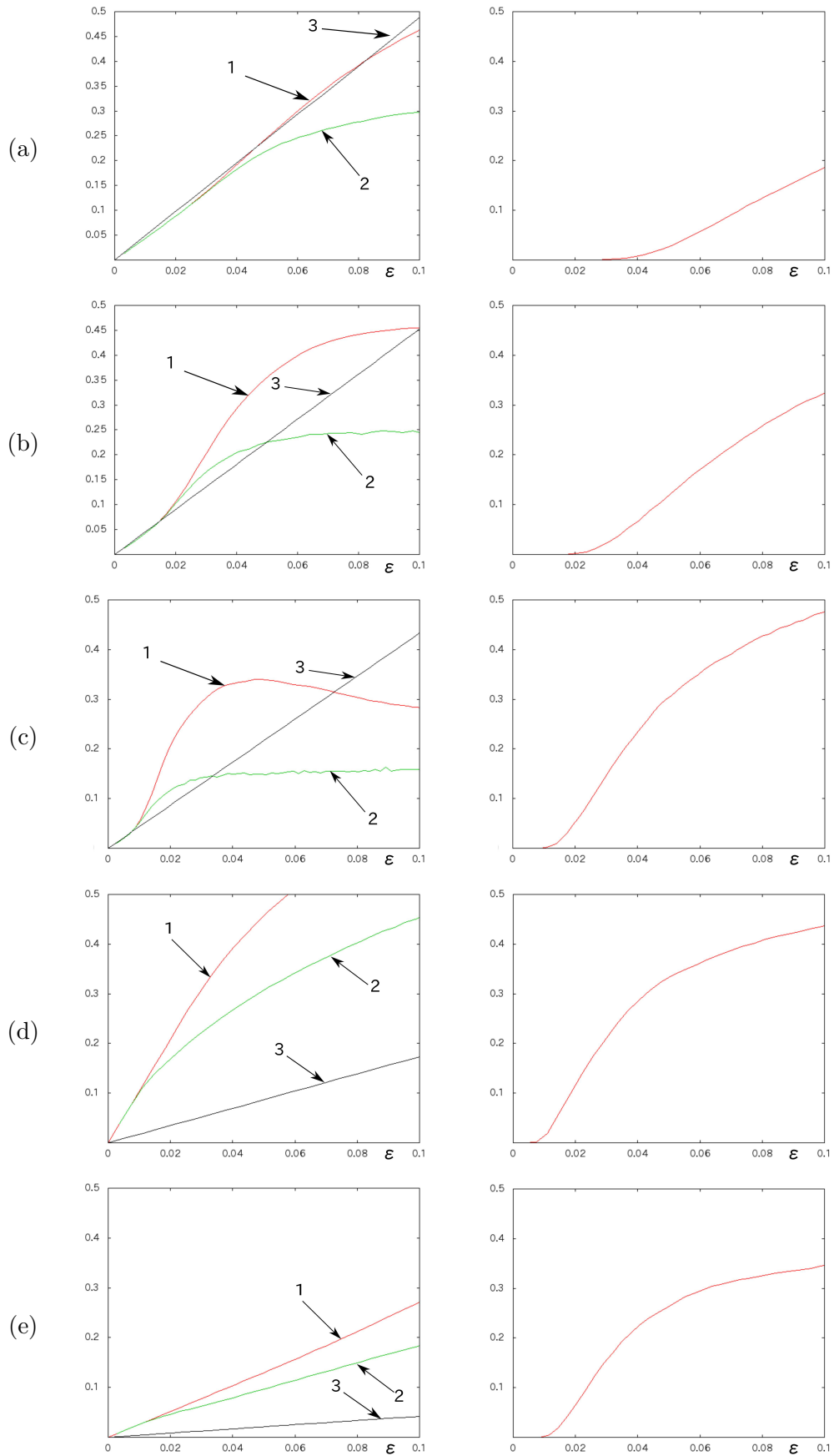


図 2.13: 左列: RMS 誤差 (1:超精度くりこみ法, 2:提案手法, 3:KCR 下界), 右列: 超精度くりこみ法で双曲線が当てはまる割合.

2.12 ランダムサンプリングの収束回数の比較

本実験では、まず提案手法のランダムサンプリング法について点の分布の考慮方法を変えた場合の検討するため、提案手法の方法、正規乱数を用いたサンプリング法、入力点列を5等分するサンプリング法についてRMS誤差と、解の収束回数を比較する。次に、超精度くりこみ法の解が双曲線だった場合のランダムサンプリングを用いた楕円当てはめについて、RANSACとPROSAC、提案手法の計算が終了するまでのサンプリング回数を比較し、提案手法の有効性を確認する。

図2.11で示す点列に対し、期待値0、標準偏差 σ の正規分布に従う誤差を加えた点 (x_α, y_α) 、 $\alpha = 1, \dots, N$ にそれぞれのサンプリング法を用いて楕円を当てはめ、いろいろな σ に対して10000回独立に試行した。

次にそれぞれの実験の結果を示す。

2.12.1 点の分布の考慮の仕方による収束回数の比較

提案手法の方法、正規乱数を用いたサンプリング法、入力点列を5等分するサンプリング法について、RMS誤差と解の収束回数を比較する。このとき、正規乱数を用いたサンプリング法の N 点の入力に対する正規乱数の標準偏差 $\sigma = N/6$ の場合と、 $\sigma = N/18$ の場合についても比較した。図2.14に左列にそれぞれの手法のRMS誤差を示し、右列にそれぞれの手法の収束回数の平均を示す。グラフの横軸は相対ノイズレベルを示す。

図2.14より、左列のRMS誤差については、提案手法、正規乱数を用いたサンプリング法では、同等の精度が得られていることが分かる。しかし、入力点列を5等分するサンプリング法がわずかに精度が高くなっている。これは、入力点列を5等分して、それぞれの区間より1点ずつ選択していることから、どのサンプリング結果でも点の分布に偏り無くサンプリングが行え、点列に沿った楕円が当てはまると考えられる。このため、真の楕円と大きく離れた楕円が試行中に当てはまらず、他のサンプリング方法よりも若干精度が高くなったと考えられる。

右列の解の収束回数については、提案手法がもっとも収束回数が少ないことがわかる。正規乱数を用いたサンプリング法は、標準偏差が小さい法が収束回数が少ないが、どちらの場合も提案手法ほど収束回数は少なくならない。これは標準偏差が大きい場合は、サンプリングする点のばらつきが大きく解が収束しないためだと考えられる。また、提案手法では、点の分布を考慮してサンプリングした点の距離が近い場合について対処しているが、正規乱数を用いた手法では考慮していないために、3点それぞれ離れた点がサンプリングされない結果が含まれたと考えられる。また、5分割するサンプリング法は、どの楕円上の点列でも、比較した手法の中で最も解の収束回数が多い。さらに、図2.14(d), (e)で示す短い入力点列に対する解の収束回数が極端に多くなっている。これは、サンプリングするときに入力点列が短いをさらに短い区間に分割してサンプリングしているため、どの試行でも同じような5点がサンプリングされやすく、たまたま速く最適解が見つかる場合も少なく、同じような楕円が当てはまり易いため、解の収束条件が満たされにくいためだと考えられる。

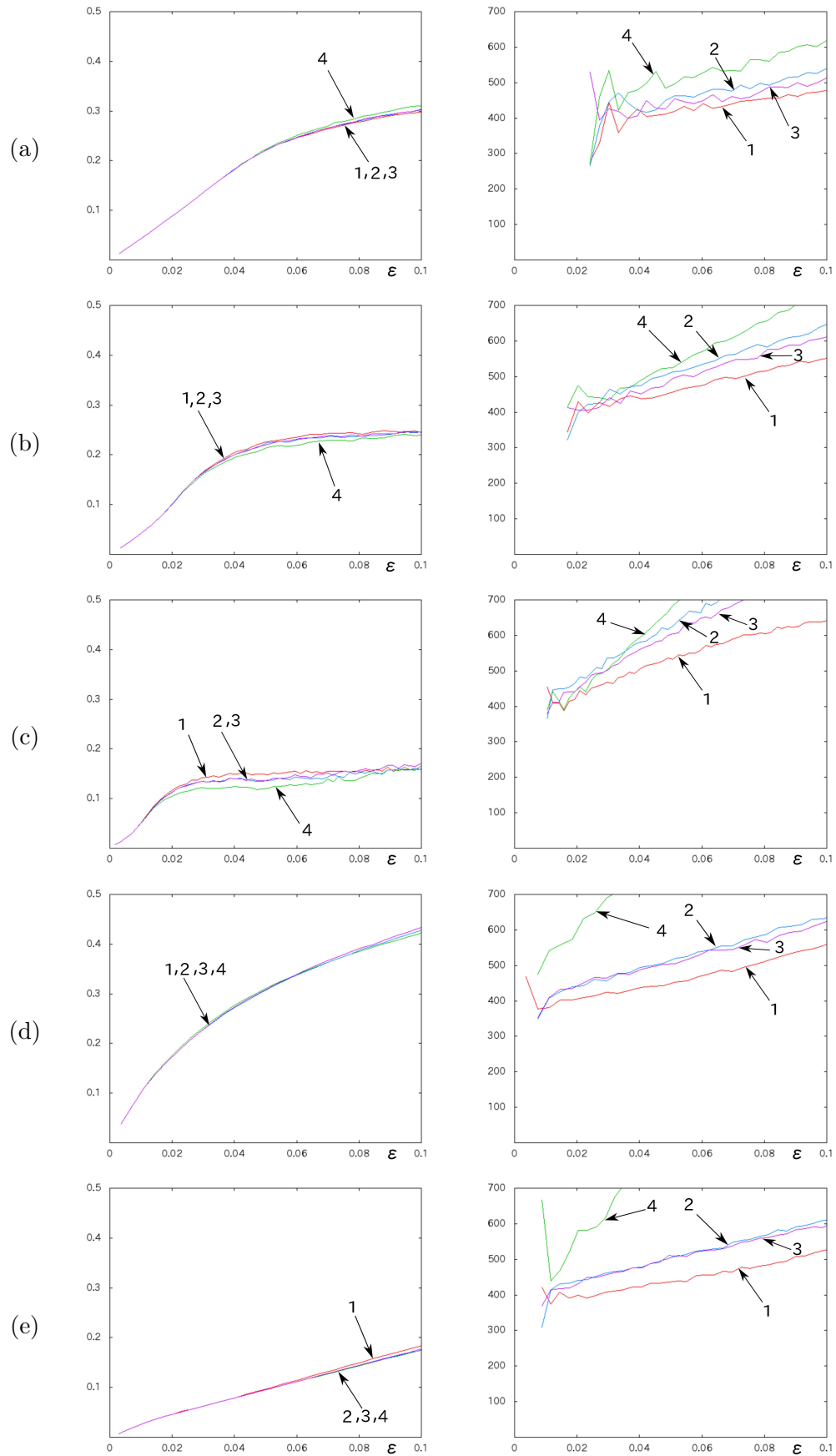


図 2.14: 左列: RMS 誤差, 右列: 解の収束回数 (1:提案手法, 2:正規乱数を用いたサンプリング法 ($\sigma = N/6$), 3:正規乱数を用いたサンプリング法 ($\sigma = N/18$), 4:入力点列を5等分するサンプリング法) .

2.12.2 既存手法との収束回数の比較

RANSAC と PROSAC, 提案手法について, 図 2.15 に左列にそれぞれの手法の RMS 誤差を示し, 右列にそれぞれの手法の収束回数の平均を示す. グラフの横軸は相対ノイズレベルを示す.

図 2.15 より, 左列の RMS 誤差はどのグラフでも, PROSAC と提案手法が RANSAC にほぼ一致しており, 同等の精度で楕円当てはめができていたことが確認できた. また, 右列の収束回数は, ノイズが小さい場合に値が安定していないが, これは超精度くりこみ法で双曲線が当てはまる割合が少ないため, ランダムサンプリングできた結果が少ないためだと考えられる. ノイズが大きくなった場合には, 提案手法が最も収束回数が少なくなっていることを確認した. また, PROSAC の収束回数が RANSAC と同等かそれ以上の結果になってしまった原因は, PROSAC は本来大量の入力点列中に当てはめたい物体とは関係ない点も含まれる場合を想定しているため, 今回の実験のような楕円上の少ない入力点列のみの入力データでは, RANSAC と同等程度の結果しか得られなかったと考えられる.

図 2.15 (a), (b), (c) の収束回数の変化をみると細い楕円になるほど提案手法とその他の手法との収束回数に差が生じるが図 2.15 (d) と (e) では収束回数に大きな差が見られない, これより提案手法は細長い楕円の短軸上の入力点列において最も有効であると考えられる.

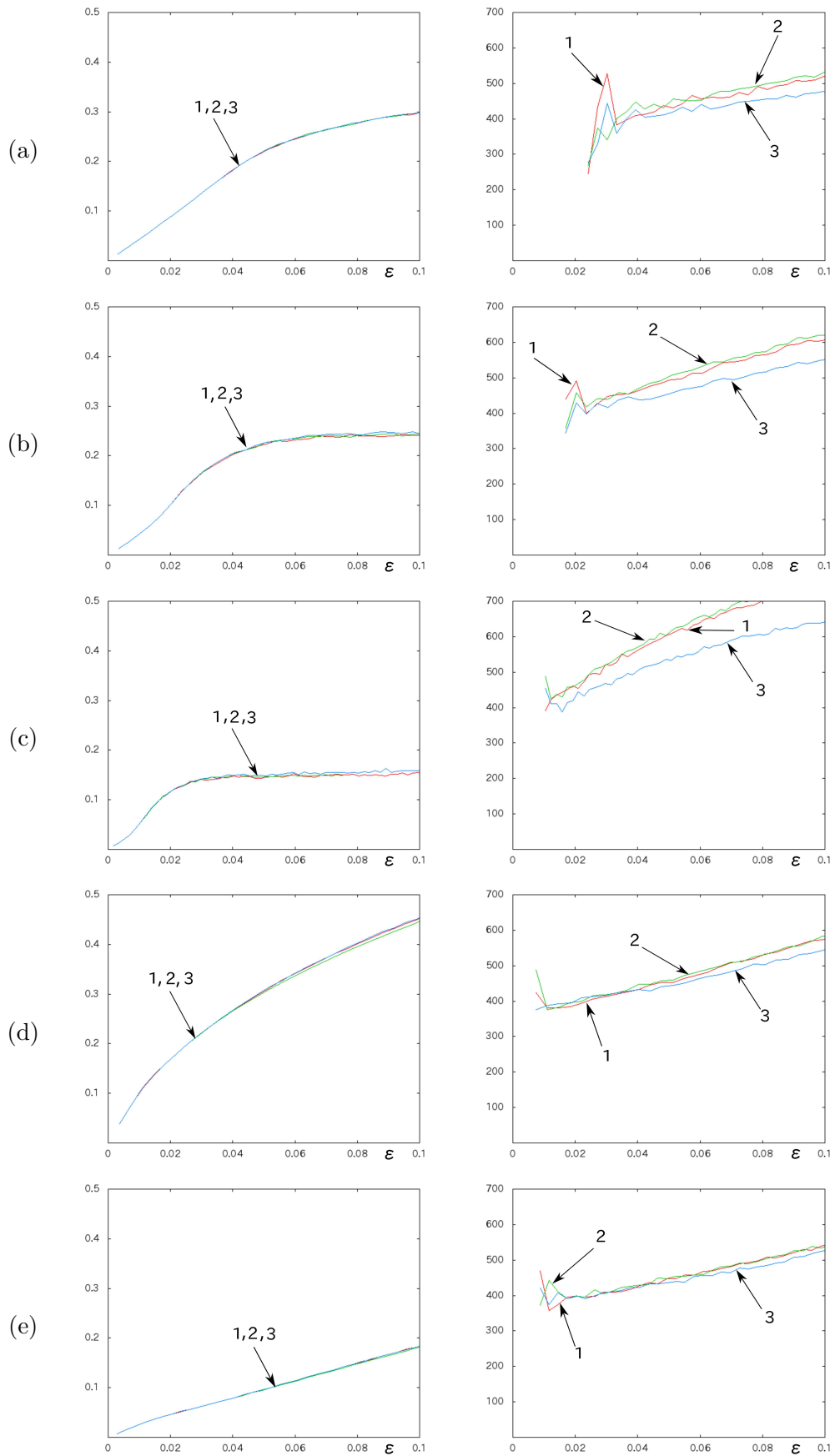


図 2.15: 左列: RMS 誤差, 右列: 解の収束回数 (1:RANSAC, 2:PROSAC, 3:提案手法)。

2.13 楕円限定当てはめの精度比較

楕円限定当てはめとしての提案手法の有効性を示すため、既存手法である Fitzgibbon らの方法と Szpak らの方法と当てはめ精度の比較を行う。

図 2.11 (a)~(e) に示す点列に対し、期待値 0、標準偏差 σ の正規分布に従う誤差を加えた点 (x_α, y_α) , $\alpha = 1, \dots, N$ に超精度くりこみ法、提案手法、Fitzgibbon らの方法、Szpak らの方法を用いて楕円を当てはめ、いろいろな σ に対して 10000 回独立に試行した。図 2.16 に図 2.11 (a)~(e) に対応する各手法の RMS 誤差を示す。グラフの横軸は相対ノイズレベルを示す。

図 2.16 (a), (b) では、提案手法が最も RMS 誤差が小さいことが確認できた。また、Fitzgibbon らの方法はノイズが低い場合は比較手法の中で最も RMS 誤差が大きくなるが、図 2.16 (b), (c) で示すように、曲率が低い点列の当てはめに対してはノイズが大きいと RMS 誤差が Szpak らの方法よりも小さくなった。Szpak らの方法は、ノイズが小さい場合は提案手法と同等の RMS 誤差が得られるが、図 2.16 (b), (c) で示すように、曲率が低い点列の当てはめではノイズが大きくなると RMS 誤差が大きくなった。しかし、図 2.16 (d), (e) で示すように曲率が高い点列では RMS 誤差が提案手法と同等またはそれ以上に小さくなった。

図 2.17 に標準偏差 1.15 で超精度くりこみ法で双曲線が当てはまる場合のそれぞれの手法楕円の当てはめ結果の一例を示す。また、図中の点線は当てはめたい真の楕円を示す。

図 2.17 (a), (b) では、提案手法が真の形状に近い楕円が当てはまっているが、図 2.17 (c), (d), (e) では、求めたい楕円とは形状が違う楕円が当てはまった。しかし、用いた手法の中では、もっとも真の形状に近いと言える。また、Fitzgibbon らの方法は、どの入力点列に対しても真の形状より小さい楕円が当てはまり、Szpak らの方法は、比較手法の中では、最も大きな楕円が当てはまった。提案手法はその中間の楕円が当てはまった。

当てはまる楕円の形状にそれぞれの手法で傾向が見られるのは、楕円を当てはめる為の計算中の制約条件によるものと推測される。解の楕円の条件は $AC - B^2 > 0$ から楕円を表すのに対し、Fitzgibbon らの方法では、条件 $AC - B^2 = 1$ のもとで Sampson 誤差を最小化しているため、より細い楕円が当てはまりやすいと考える。また、Szpak らの方法では解が放物線になる条件 $AC - B^2 = 0$ に近づく用に、Sampson 誤差を最小化するため、双曲線に近い大きな楕円が当てはまりやすいと考える。

実験より、提案手法は曲率の低い点列図 2.11 (b), (c) では Fitzgibbon らの方法と同等かそれ以上の精度になり、曲率の高い点列図 2.11 (d), (e) では Szpak らの方法に近い精度になっており、どの入力点列のパターンでも高い精度の当てはめが可能であるといえる。

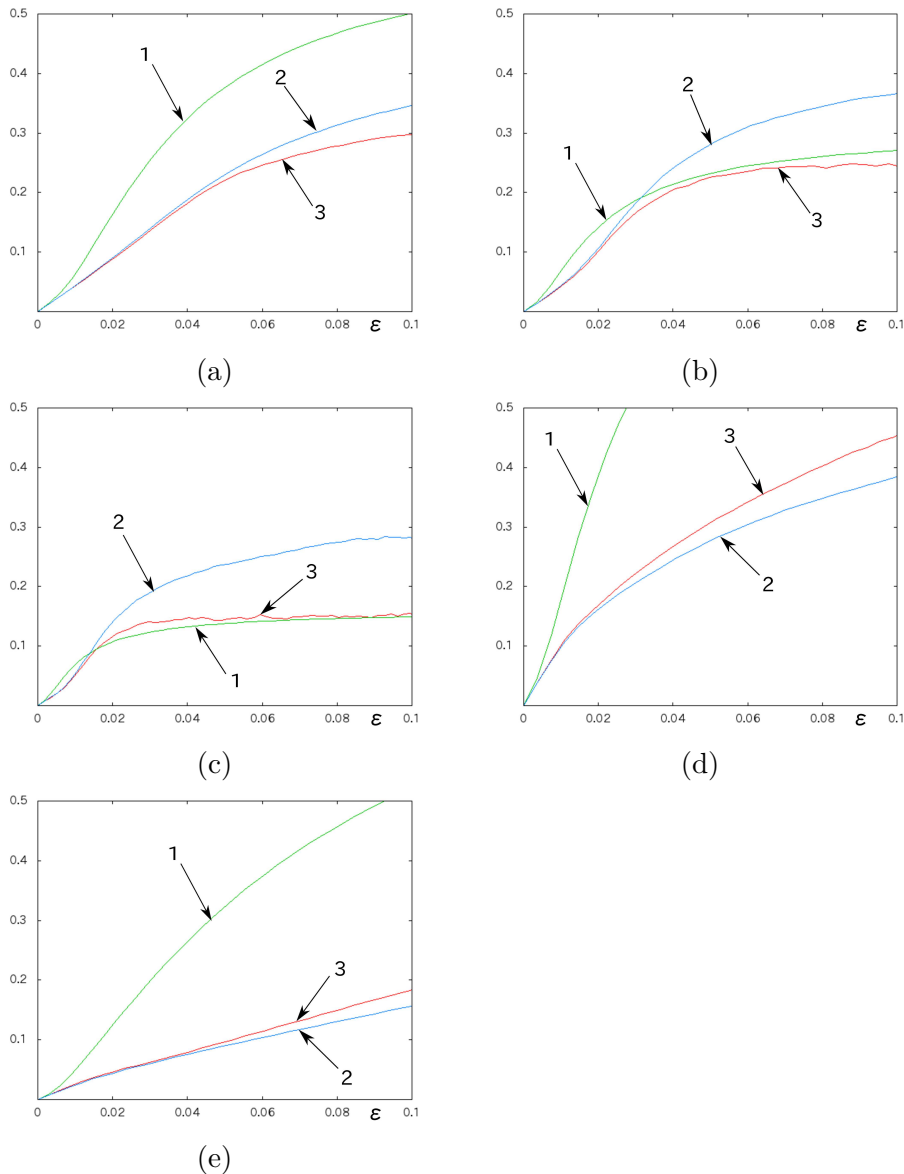


図 2.16: RMS 誤差 (1:Fitzgibbon らの方法, 2:Szpak らの方法, 3:提案手法)。

2.14 実画像を用いた楕円当てはめの評価

本節では、実画像より抽出した点列に、超精度くりこみ法、提案手法、Fitzgibbon らの方法、Szpak らの方法を当てはめ、提案手法の有用性を確認する。図 2.18 (a)~(d) に当てはめ結果を示す。図 2.18 の左列に実験に用いた入力点列とエッジ画像を示す。図中の ■ が用いた点列を表す。また、図 2.18 の右列に各手法で当てはめた楕円を入力画像に描画した結果を示す。

図 2.18 (a) は曲率が低く長い点列を用いて楕円を当てはめた。超精度くりこみ法で楕円が当てはまり提案手法も同じ楕円が当てはまる。超精度くりこみ法、提案手法、Szpak らの方法で真の楕円に重なる楕円が当てはまった。Fitzgibbon らの方法は若干小さい楕円が当てはまった。図 2.18 (a) は入力点数が実験結果の中で最も多く、提案手法が最も真の形状に近い結果で当てはめられたことから、入力点数多い点列ほど高精度に楕円が当てはめられることがわかる。

図 2.18 (b) も図 2.18 (a) と同じような曲率が低い点列を用いて楕円を当てはめたが、他の実験結果と比べ提案手法が真の形状より遠い楕円が当てはまった。これは、入力点列が背景の建物の

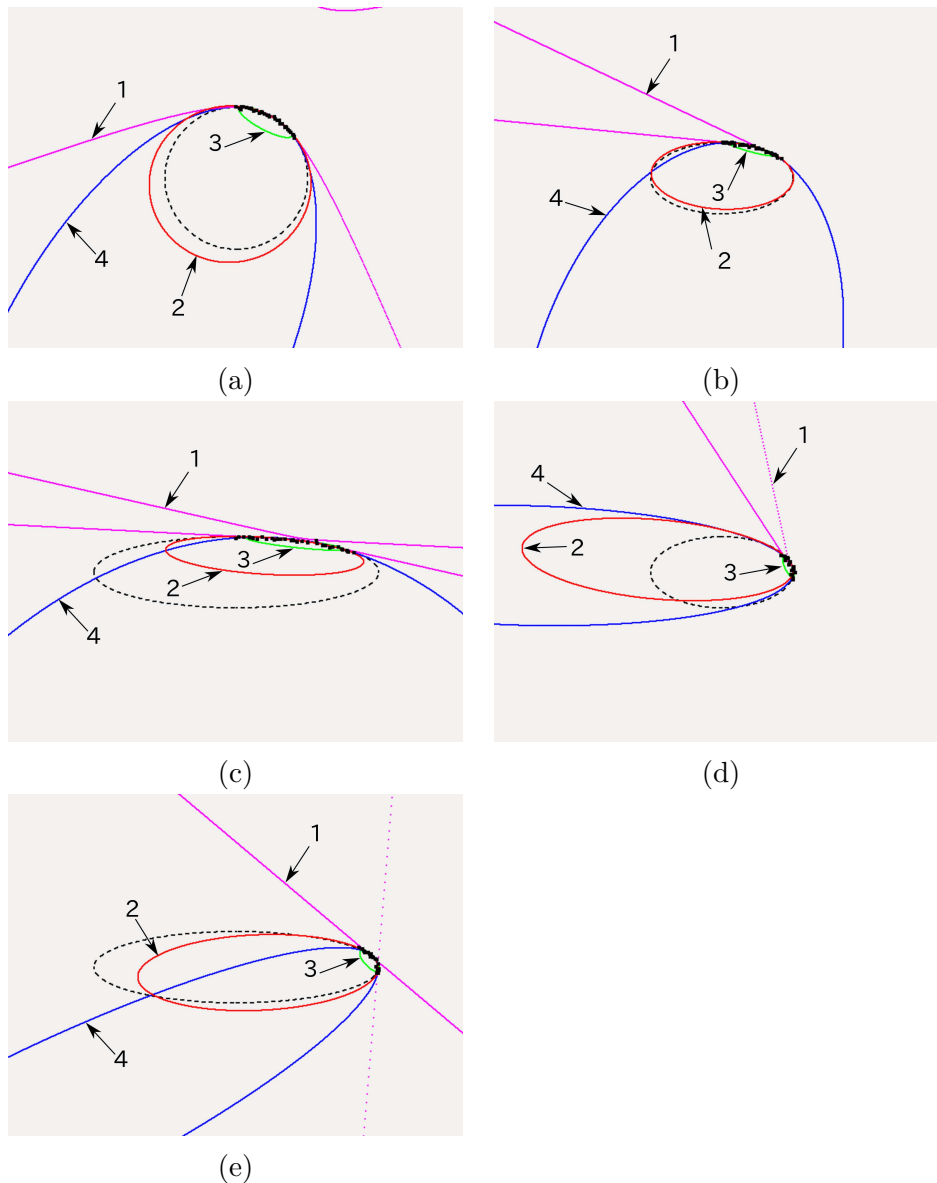


図 2.17: 超精度くりこみ法で双曲線が当てはまる場合の結果 (1:超精度くりこみ法, 2:提案手法, 3:Fitzgibbon らの方法, 4:Szpak らの方法)。

エッジに影響を受けノイズが若干大きくなってしまったためだと考えられる。

図 2.18 (c) は楕円の長軸上の曲率が高く短い点列を用いて、楕円を当てはめた。超精度くりこみ法で双曲線が当てはまり、Szpak らの方法も双曲線に近い大きな楕円、Fitzgibbon らの方法は小さい楕円が当てはまり、提案手法はその中間の楕円が当てはまった。

図 2.18 (d) はさらに細長い楕円の長軸上の短い点列を用いて楕円を当てはめた。超精度くりこみ法で双曲線が当てはまり、Szpak らの方法は双曲線に近い大きな楕円、Fitzgibbon らの方法は小さい楕円が当てはまり、提案手法は最も真の形状に近い楕円が当てはまった。

このことから、Szpak は大きな楕円、Fitzgibbon ら小さな楕円が当てはまる傾向があり、提案手法はその中間の楕円が当てはまる傾向があることが分かった。また、入力点列の長さが長い程、当てはめ精度は向上するが、図 2.18 (b) のように、入力点列に加わる大きな背景等のノイズが加わると精度が落ちるので、実画像を用いた楕円当てはめでは、ノイズを除去した点列をできるだけ長く取得することが重要である。

2.15 まとめ

本章では、金谷らの超精度くりこみ法に点の分布を考慮したランダムサンプリングを組み合わせた新しい楕円限定当てはめ法を提案した。従来の楕円当てはめ法の中でも優れた楕円当てはめ

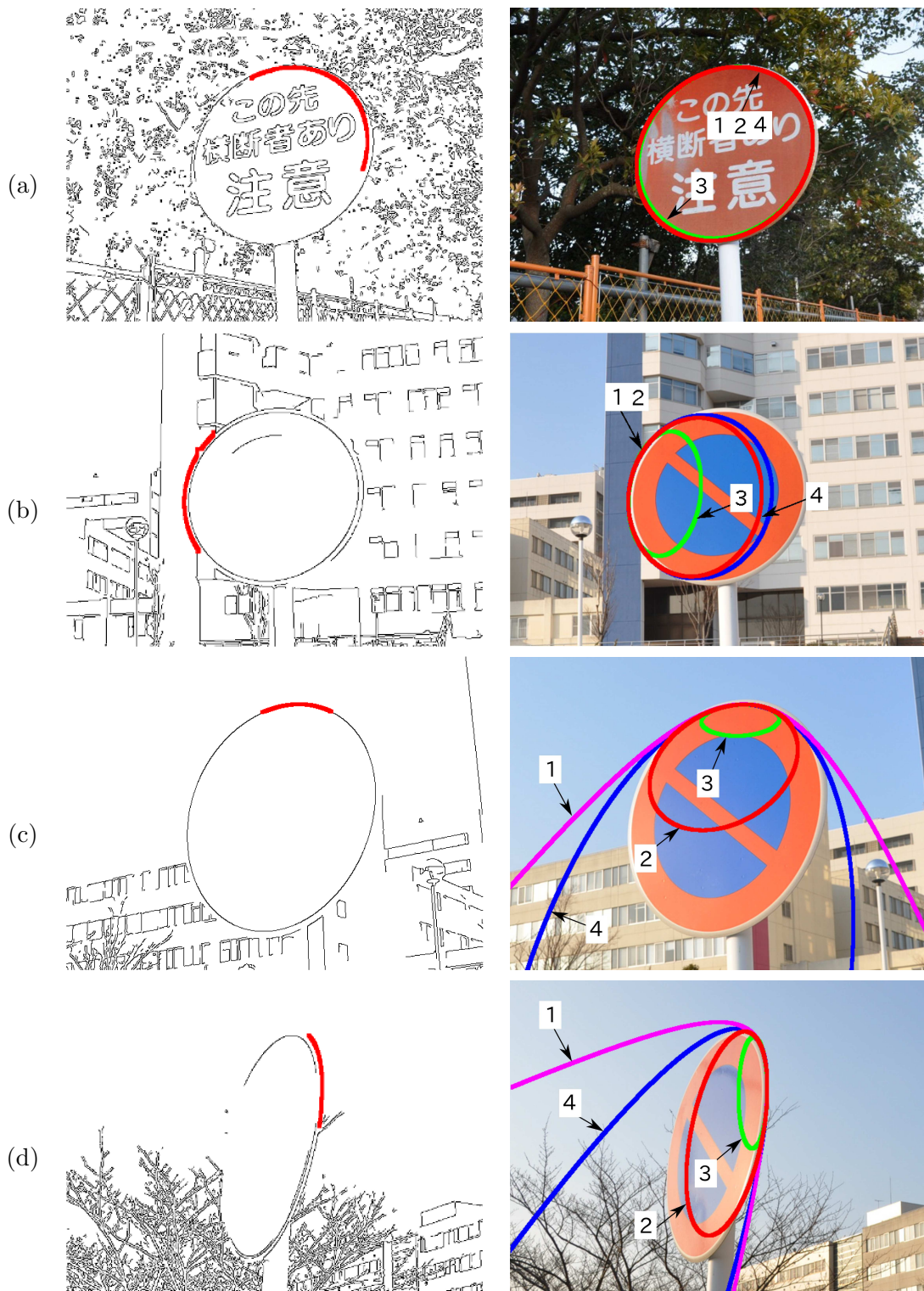


図 2.18: 左列: エッジ画像と入力点列 ((a):326 点, (b):230 点, (c):106 点, (d):149 点), 右列: 入力画像と当てはめ結果 (1:超精度くりこみ法, 2:提案手法, 3:Fitzgibbon らの方法, 4:Szpak らの方法)。

法だと言われる超精度くりこみ法であるが、ノイズが大きいと双曲線が当てはまることがある。提案手法はそのような場合に入力点列のランダムサンプリングによって点列に最も近い楕円を当てはめる手法を示した。実験により、入力点列に加わるノイズが大きくなり、超精度くりこみ法が双曲線を当てはめる割合が増加した場合に提案手法により精度が向上することを確認した。

これまでにランダムサンプリングの収束回数を削減する方法として PROSAC が挙げられるが、超精度くりこみ法で双曲線が当てはまる場合の入力点列に対し、RANSAC、PROSAC、提案手法の解の収束回数をそれぞれ比較した実験により提案手法は PROSAC と同等またはそれ以上に収束回数の削減ができることを示した。また、提案手法の収束回数の削減効果は、曲率が低い入力点列で楕円が細くなるほど効果が上がることを確認した。

第3章 効率的な楕円弧選択

前章では、ノイズの含んだ楕円弧点列に対して高精度に楕円限定当てはめを行う手法について述べた。しかし、リアルタイム処理上での楕円検出では、入力点列に楕円弧以外の点列も含まれており、楕円限定当てはめはそれらのアウトライアには対応していない。この章では、効率的な楕円検出のために、アウトライアが含まれる点列から楕円弧点列のみを効率的に抽出する手法について既存手法と提案手法についてそれぞれ説明し、実験によって提案手法の有効性を示す。本論文では、楕円弧点列中に含まれる楕円弧上から外れた点をノイズ、入力点列に含まれる楕円弧以外の点列をアウトライアと呼び分ける。

3.1 既存の楕円検出

画像より抽出した点列から楕円を検出する一般的な方法として Hough 変換を用いた楕円当てはめ法を用いる方法 [30] が挙げられる。しかし、Hough 変換はパラメータ空間への投票に基づく方法であるため、楕円の 5 次元パラメータではメモリ量や計算量が非常に大きくなる。

これに対し、Yu らは、画像より得られた連結した点列に Fitzgibbon らの楕円限定当てはめ [5] と当てはまり具合の悪い点の除去を繰り返し、点列中から楕円を検出する方法 [47] を提案した。Fitzgibbon らの楕円限定当てはめは、反復計算が必要なく、楕円弧以外の点列（これ以降アウトライアと呼ぶ）が含まれる点列に対しても必ず楕円を当てはめることができる。しかし、Yu らの方法では、一回の楕円候補を得るまでの点列除去の繰り返し中で、除去した点はそれ以降使用しないため、誤って楕円上の点を除去してしまった短い点列に対し、小さな楕円が当てはまってしまう場合がある。Shao らは、連結した点列を一定の長さや条件により分割し、楕円の当てはめ誤差が小さくなる点列の組み合わせを総当りで探索する手法 [38] を提案した。これらは、経験的なしきい値などを用いて入力点列を細かな部分弧に分割するため、必要以上に多くの部分弧に分割してしまい解を得るまでに多くの探索が必要となる。

菅谷らは RANSAC による楕円を当てはめによって楕円上の点列を選択し、短い点列を選択してしまった場合でも、得られた同一楕円上の点列をモデル選択によって統合し、楕円を再当てはめを行う方法 [40] を提案した。これにより、他の物体に隠された欠損のある画像中の楕円物体に対しても正しく楕円を検出することが可能となる。ここで用いるモデル選択とは、モデルの当てはめ誤差とモデルパラメータの自由度の両方を考慮して、モデルに対する当てはまり具合を評価するものであり、代表的な幾何学的 MDL [16] を用いて同一楕円上の点列統合判定を行う。

しかし、楕円弧の全組み合わせごとに楕円当てはめが必要であり、ランダムサンプリングを用いた楕円当てはめでは計算量が多くなり、計算時間が多くなる。

3.2 提案手法

本研究では、図 3.1 (a) に示すように、入力点列は連結しており、楕円弧上の点列とそれ以外の点列が結合しているものとし、このような入力点列に楕円を当てはめると、入力点列と交差する

ような楕円が得られると仮定する。我々は，入力点列を楕円の当てはめ誤差をもとに部分弧に分割して，部分弧ごとにインライアとアウトライアを判別する方法を提案する。本研究では部分弧を次の二つのアプローチによって選択する。

1. 入力点列へ楕円を当てはめ，部分弧ごとの誤差グラフの曲率を用いた点列選択
2. 部分弧ごとに楕円を当てはめ，それぞれの当てはめ誤差を用いた点列選択

手法 1 の選択は，部分弧ごとの誤差の変化のみに注目すればよく，ほかの指標を再計算する必要がなく効率的に部分弧を選択できる。手法 2 は手法 1 の選択方法よりも計算量は増すが，得られた部分弧の中で最も楕円弧らしい点列を選択可能である。提案方法の入力点列を部分弧に分割し，その中から楕円弧点列を選択していくアプローチは，すべての点を調べるのではなく，部分弧ごとに楕円弧の判定を行うため，入力点を個々に評価するよりも効率的であり，反復計算では全ての部分弧を対象として判定を繰り返すため，Yu らの方法のように反復の過程で対象となる点列が減少することがない。また，RANSAC のようにランダムサンプリングを行わないので，収束までの反復回数も少ない。

次節より，提案手法の前処理である，部分弧と接部分弧への点列分割，部分弧の拡張について述べ，2 つの楕円弧選択方法について説明する。

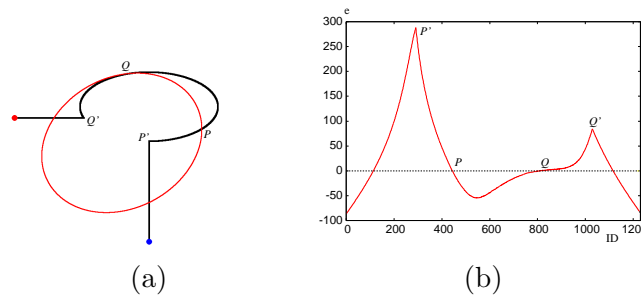


図 3.1: アウトライアを含む点列への楕円当てはめと誤差のグラフ。(a) 入力点列 (黒点) と当てはめた楕円 (赤)。入力点列は青丸を始点，赤丸を終点とする順番に連結性を保って並んでいる。(b) 当てはめた楕円に対する入力点列との誤差。横軸:(a) の青丸を始点とする点の番号。縦軸:符号づけした楕円の当てはめ誤差。

3.3 部分弧と接部分弧

円錐曲線を表す式 (3.1) の左辺の x, y に任意の点 $p(x, y)$ の座標を代入すると，その点が楕円弧上の点であれば 0，楕円の外側もしくは内側であれば正の値か負の値となる (符号は反対の場合もある)。

$$Ax^2 + 2Bxy + Cy^2 + 2f_0(Dx + Ey) + f_0^2F = 0, \quad (3.1)$$

式 (3.1) を簡単化するため，6次元ベクトル

$$\xi_\alpha = (x_\alpha^2, 2x_\alpha y_\alpha, y_\alpha^2, 2f_0 x_\alpha, 2f_0 y_\alpha, f_0^2)^\top, \theta = (A, B, C, D, E, F)^\top \quad (3.2)$$

を定義し、ベクトル a, b の内積を (a, b) とし、式 (3.1) を、

$$(\xi_\alpha, \theta) \approx 0, \quad \alpha = 1, \dots, N \quad (3.3)$$

のように書く。

当てはめた楕円と入力点列の交点は、式 (3.1) の左辺の符号が変化する点として検出することが可能である。この交点で分割された点列を以降の説明では、「部分弧」と呼ぶ。ただし、点列が楕円に接する場合には、式 (3.1) の左辺の値が厳密に 0 になることはないので、隣接点間で符号の変化が激しく起こり、非常に短い部分弧が多数生成されることがある。これらの部分弧は楕円の当てはめ誤差が非常に小さい楕円上の点列と推定できる。以降の説明ではこの部分弧を「接部分弧」と呼ぶことにする。

3.4 部分弧の拡張

図 3.1 の例では、当てはめた楕円により入力点列が 5 つの部分弧に分割される。そして、楕円弧のみを含むインライアである部分弧のグラフでは、誤差の変化が滑らかなのに対して、アウトライア点を含む部分弧のグラフでは、誤差がピークの周辺で急激に変化していることが確認できる。また、インライアである部分弧 PQ の両端の弧 PP' と弧 QQ' もインライアであることがわかる。点 P', Q' はインライアと判定した部分弧 PQ に隣接した部分弧の誤差のピーク点である。

対象の部分弧が楕円弧であった場合、楕円弧の部分弧に隣接する点列も楕円弧である可能性が高い。楕円当てはめでは楕円上を占める割合が大きい点列を使用するほど正しい楕円の形状を推定できる。従って、インライアと判定された部分弧のみではなく、その両端の部分弧の誤差のピーク点まで含めた部分弧（図 3.1 の例では部分弧 $P'Q'$ ）をインライアとして、次の楕円当てはめに使用することで、より正しい形状の楕円を当てはめる。

3.5 誤差グラフの曲率に注目した楕円弧選択

本手法では、図 3.1 (b) で示すように誤差の変化は連結点列の曲率の変化を擬似的に表しており、点列が滑らかな曲線であればその誤差変化も滑らかになるため、この誤差の変化をもとにインライアを選択する。これは、入力点列をその曲率で分割することと同じ発想であるが、すべての点を調べるのではなく、分割した部分弧ごとの誤差がピークとなる点の誤差グラフの曲率のみを調べ、その曲率が大きい部分弧をアウトライア、それ以外の部分弧をインライアと判定するのですべて点の曲率を調べる方法よりも効率的である。

3.5.1 誤差グラフの曲率を用いた楕円弧選択の流れ

この手法では、入力点列は連結しており、楕円弧上の点列とそれ以外の点列が結合しているものとし、このような入力点列に楕円を当てはめると、入力点列と交差するような楕円が得られると仮定する。また、点列が楕円弧であった場合は誤差グラフは滑らかに変化し、アウトライアであった場合は誤差グラフの変化がピーク値付近で大きくなると仮定し、部分弧を繰り返し選択し、入力点列中の楕円弧のみを用いた楕円を当てはめる。手法の流れを以下に示す。

1. 点列に Fitzgibbon らの方法で楕円を当てはめる。

2. 当てはめた楕円のパラメータを用いて各点ごとに式 (3.1) の左辺の値を計算し, その符号が変化する点で点列を部分弧に分割する.
3. 分割した部分弧ごとに, その中で最大誤差をもつ点とその周辺の点とのなす角 ϕ を計算する.
4. 1 回目の楕円当てはめであればステップ (a), そうでなければステップ (b) を行なう.
 - (a) 部分弧の長さがしきい値よりも長い部分弧の中で計算した角度が最大の部分弧とそれに隣接する部分弧の誤差のピークまでの弧をインライアとして楕円を当てはめる.
 - (b) 計算した角度がしきい値以上の部分弧をインライアとして楕円を当てはめる.
5. 得られるインライア点数が変化しなくなるまでステップ 2 からステップ 4 の処理を繰り返す.

インライアとして検出する部分弧の選択基準が, 1 回目とそれ以降で異なる. 誤差グラフのインライアと判定された部分弧が本当にインライアであれば, それに隣接する部分弧の一部もインライアであるから, これらをすべて使用して楕円を当てはめると効率的に正しい楕円が得られる. しかし, 本来アウトライアである部分弧を誤ってインライアと判定した場合には, それに隣接するアウトライアをさらに追加してしまうため, 正しい楕円が得られにくくなるため, 2 回目以降では, インライアと判定された部分弧をすべてインライアとして採用するが, インライアの部分弧に隣接した部分弧の一部はインライアとしては追加しない. これは, 入力点列のうちインライアのみ, もしくはほとんどがインライアである点に楕円を当てはめた場合には, 当てはめた楕円と点列の交点がインライアとアウトライアの境界となるので, アウトライア点を誤ってインライア点として追加しないためである.

3.5.2 誤差グラフの横軸の正規化

提案手法は分割した部分弧ごとに誤差が最大となる点を検出し, その点と前後の点を結ぶ線分のなす角を計算する. その角度がしきい値以下であれば, その部分弧をアウトライア, しきい値以上である部分弧をインライアとするが, この角の大きさは, 誤差のグラフの横軸のスケールに依存する. 例えば, 同じ形状の点列でも, その大きさが異なる場合, 大きさが大きい方, すなわち, 点列の数が多く対応する部分の角度が大きくなる. そこで, グラフの横軸をすべての入力点中の最大誤差を基準に正規化して, 次の 2 次元ベクトルで表現する.

$$\mathbf{Q}_\alpha = \left(\frac{\lambda e_{max} \alpha}{N}, e_\alpha \right)^\top \quad (3.4)$$

ここで, e_{max} はすべての入力点中の楕円の当てはめ誤差の最大値, λ は正規化のための定数¹, e_α は式 (C.2) で計算される点 α の当てはめた楕円に対する誤差である.

$$e_\alpha = \sqrt{\frac{(\xi_\alpha, \theta)}{(\theta, V_0[\xi_\alpha] \theta)}} \quad (3.5)$$

また, 当てはめ誤差の最大値が極端に大きい場合には式 (3.4) の正規化がうまく働かないため, 誤差の最大値のしきい値 E_{max} を定めて², e_{max} がしきい値 E_{max} よりも大きい場合には, e_{max} を E_{max} に置き換えて正規化する.

¹実験では $\lambda = 2.0$ とした.

²実験では (入力点の座標の最大値 / 3) とした.

M 個の部分弧 $R_\kappa(i, j) = \{Q_\beta, \beta = i, \dots, j\}, \kappa = 1, \dots, M$ に分割した点 $Q_\alpha, \alpha = 1, \dots, N$ に対して, 次の手順でインライアとアウトライアの判定を行う. アルゴリズム中で説明する記号を図示したものを図 3.2 に示す.

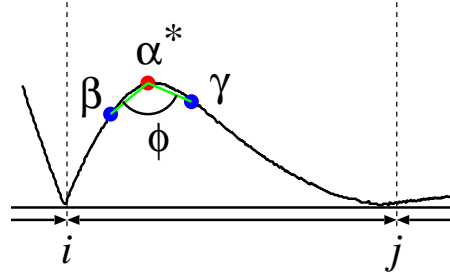


図 3.2: 分割した点列の誤差変化の指標

3.5.3 誤差グラフの曲率の評価

1. 部分弧 R_κ 内の誤差の最大値を探索し, その点番号を α^* とする. ここで, この部分弧内の誤差の最大値がしきい値 E_{min} ³ より小さい場合には, その部分弧を接部分弧としてインライアに分類して終了する.
2. 点 α^* から d だけ離れた点を点 β, γ とし, 次のように計算する.

$$\beta = \alpha^* - d, \gamma = \alpha^* + d, d = (j - i)/r \quad (3.6)$$

ここで, r は最大誤差を取る点と隣接点との距離を定めるパラメータである⁴. もし, β と γ がともにその部分弧内になれば, どちらかが部分弧内となるまで, $d \leftarrow d - 1$ とする.

3. 次の三つの条件に従い, 二つのベクトル $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}$ を計算する. ここで, $x_b^{(a)}$ はベクトル $\mathbf{x}^{(a)}$ の第 b 成分を示す.

(a) Q_β と Q_γ がともに部分弧内の点の場合:

$$\mathbf{x}^{(1)} = Q_{\alpha^*} - Q_\beta, \mathbf{x}^{(2)} = Q_{\alpha^*} - Q_\gamma \quad (3.7)$$

(b) Q_β のみが部分弧内の場合:

$$\mathbf{x}^{(1)} = Q_{\alpha^*} - Q_\beta, \mathbf{x}^{(2)} = (-x_1^{(1)}, x_2^{(1)})^\top \quad (3.8)$$

(c) Q_γ のみが部分弧内の場合:

$$\mathbf{x}^{(2)} = Q_{\alpha^*} - Q_\gamma, \mathbf{x}^{(1)} = (-x_1^{(2)}, x_2^{(2)})^\top \quad (3.9)$$

³実験では $E_{min} = 1.5$ とした.

⁴実験では $r = 8$ とした.

4. 次の式でベクトル $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}$ のなす角 ϕ を計算する .

$$\phi = \cos^{-1} \left(\frac{(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})}{\|\mathbf{x}^{(1)}\| \|\mathbf{x}^{(2)}\|} \right) \quad (3.10)$$

5. ϕ がしきい値 $\hat{\phi}$ よりも小さい場合 , その部分弧をアウトライア , そうでない場合にはインライアとする .

この手法の一番最初のインライア検出では, 上記のアルゴリズムで検出された接部分弧と, それ以外で長さがしきい値よりも大きいインライア弧の中で誤差のピークの角度が最大となる部分弧, その部分弧に隣接する部分弧のうち誤差のピーク点から隣接するインライア弧の端点までを次の楕円当てはめに使用するインライアとする . ただし, インライア弧に隣接する部分弧が接部分弧であった場合は, さらにそれに隣接する部分弧を調べて接部分弧ではない部分弧のピーク点から隣接する接部分弧の端点までをインライアとする .

3.6 実験

本実験では、誤差グラフと曲率に注目した楕円弧選択方法について、正しく提案手法によって楕円弧が選択できることを示し、既存の楕円限定当てはめ法とインライアの選択精度と楕円を当てはめるまでの反復計算回数を比較し、提案手法の有効性を示す。

インライア選択

図 3.3, 図 3.4 に示すアウトライアを含む入力に対して提案手法の楕円弧の選択過程を示す。提案手法では、楕円当てはめと、点列の選択を繰り返し行ない、最終的な入力点列中の楕円弧点列を選択している。図 3.3, 図 3.4 では、提案手法のステップごとの選択した点列と当てはめた楕円、楕円の当てはめ誤差をそれぞれ示す。

図 3.3 (1)-(b) は全ての入力点を用いて Fitzgibbon らの手法で楕円を当てはめており、図 3.3 はその符号付けした誤差グラフを示している。横軸は整列した入力点を示しており、縦軸で入力点に対応する符号付けした楕円の当てはめ誤差を示す。

提案手法ではまず入力点を全て用いて楕円を当てはめ、図 3.3 (1)-(c) の誤差グラフより図 3.3 (2)-(a) で示す青い点列を選択し、さらに隣接した部分弧の誤差のピーク値までの緑色で示した点列を選択した。選択した点列を用いて当てはめた楕円を図 3.3 (2)-(b) に示す。ここで、入力内の楕円弧がほぼ選択され、図 3.3 (2)-(c) にて選択したインライア点列が変化しないため、3 ステップ目で反復計算を終了し、楕円弧点連列が正しく選択できることが確認できた。

図 3.4 は 3 つの楕円が重なり、連結した複数の楕円弧点列が得られる入力である。図 3.4 (1) では全ての入力点を使用して楕円を当てはめ、そのときの誤差グラフを示している。この入力点列は閉じているため、グラフの左端と右端は連結した区間だと考える。図 3.4 (2)-(a) は、図 3.4 (1)-(c) の誤差グラフ中で最も変化角度が小さい点列を拡張した点列が選択されており、正しく一つの楕円弧が選択出来ていることが確認できる。しかし、図 3.4 (2)-(b) で示すように、このステップで使用した点列のみでは正しい楕円が当てはめられていない。図 3.4 (2)-(c) の誤差グラフより、図 3.4 (3)-(a) に示す同一楕円上の対角に存在する点列を選択でき、3 ステップ目で図 3.4 (3)-(b) に示す正しい楕円を当てはめることができた。これらの結果より誤差グラフの曲率により入力点列中の楕円弧を選択し、提案手法によって正しい楕円を当てはめることが可能であることを確認した。

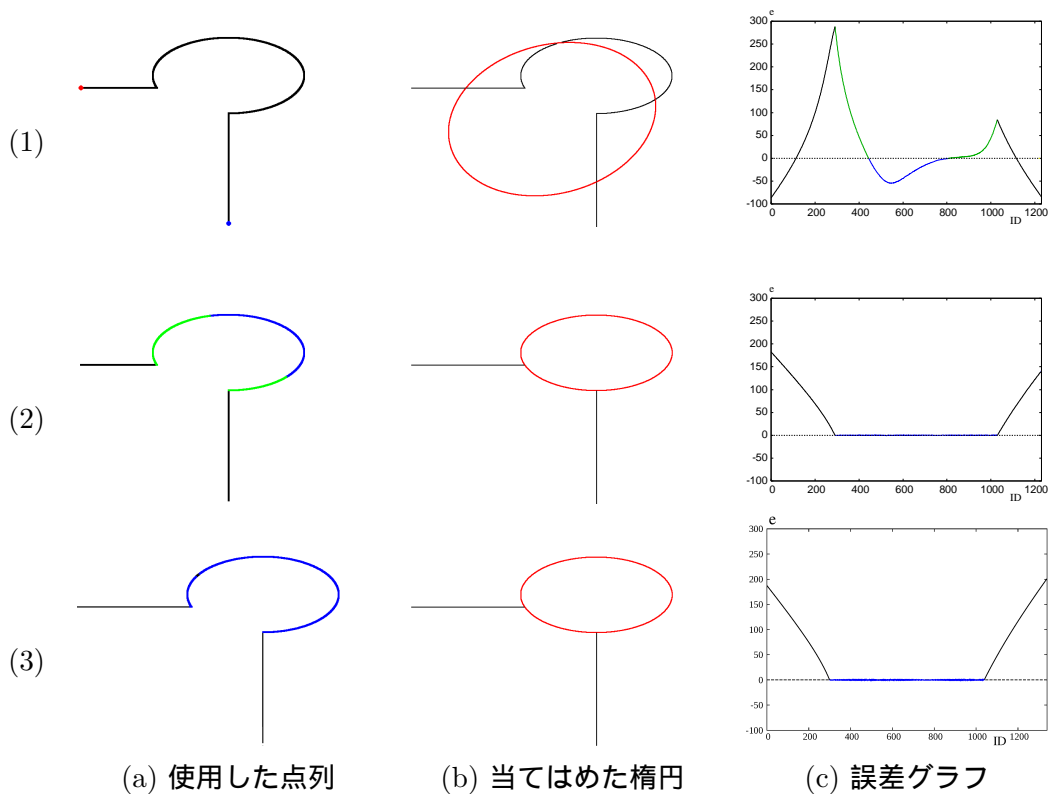


図 3.3: 誤差グラフの曲率を用いた楕円弧選択. (a) 青線:選択したインライア点列. 緑線:選択した点列から拡張した点列. (c) 横軸:整列した入力点列. 縦軸:楕円の内外で符号付けした楕円の当てはめ誤差. 青線と緑線:次の行の (a) で選択する点列.

3.6.1 シミュレーション実験

本手法の有効性を確認するため、既存手法との比較し、その有効性を示す。本実験では、提案手法、RANSAC、Yu らの手法、Shao らの手法とシミュレーションデータを用いて結果を比較した。図 3.5 にそれぞれの楕円の当てはめ結果を示す。図 3.5 (a), (c), (e), (g) はそれぞれの手法で当てはめた楕円を赤線で示す。図 3.5 (b), (d), (h) は提案手法、RANSAC、Shao らの手法で選択したインライア点列を青点でそれぞれ示す。図 3.5 (f) は Yu らの手法で結果として得られる楕円当てはめに用いた点列を青点で示す。

図 3.5 (1), (2) の入力データは同形状でスケールが異なるが、全ての手法で正しく楕円を当てはめることができた。図 3.5 (3) はスケールの違う同じような形状の楕円が 2 つ接している入力である。提案手法と RANSAC、Shao らの手法は外側の点列に楕円を当てはめたが、Yu らの手法は、内側の点列に楕円を当てはめた。

図 3.5 (4) は楕円弧の内側に直線の点列が連結している入力データである。全ての手法によって楕円弧を選択することが出来たが、図 3.5 (4)-(f) の Yu らの手法は短い楕円弧点列しか選択されておらず、他の手法と比較し、小さい楕円が当てはまっている。

図 3.5 (5) は直線点列と楕円弧点列が滑らかに連結している入力データである。このような入力に対して入力点列の曲率によってアウトライアを除去する手法ではうまく動作しないが、提案手法は楕円弧のみを正しく選択できている。

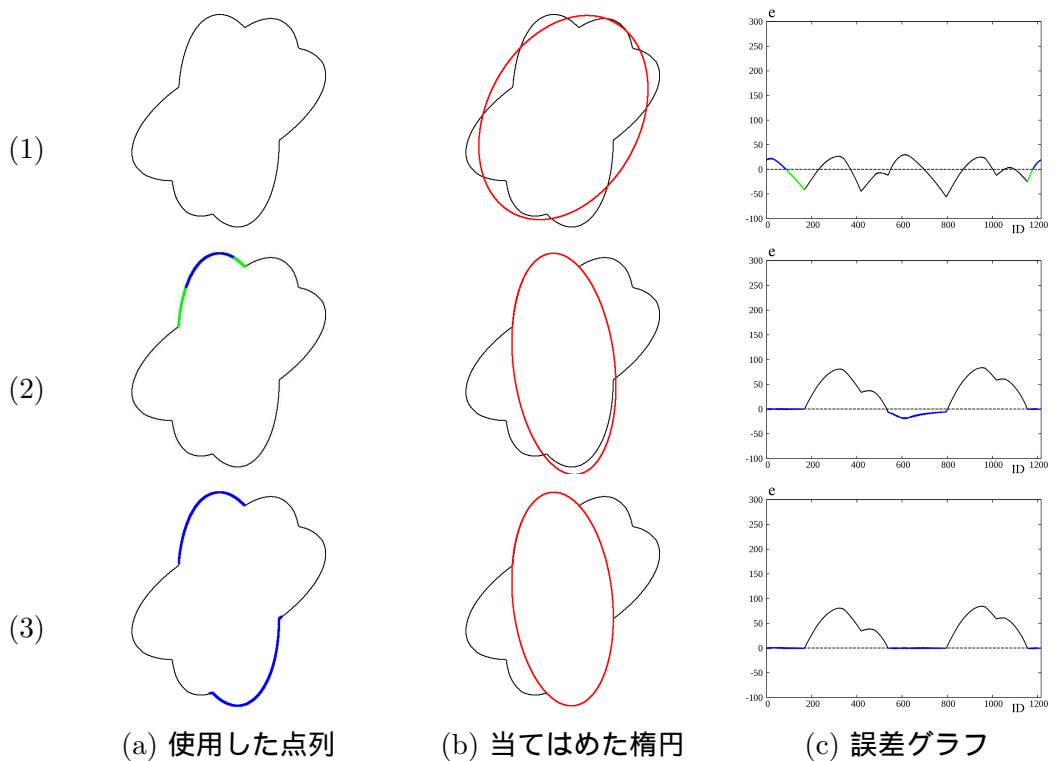


図 3.4: 誤差グラフの曲率を用いた楕円弧選択. (a) 青線:選択したインライア点列. 緑線:選択した点列から拡張した点列. (c) 横軸:整列した入力点列. 縦軸:楕円の内外で符号付けした楕円の当てはめ誤差. 青線と緑線:次の行の (a) で選択する点列.

図 3.5 (6), (7) は楕円弧点列中に異なる形状の点列を含んだ入力データである. 提案手法と RANSAC はそれぞれ正しい楕円を当てはめることができた. Yu らの手法は, 図 3.5 (7)-(e) で示すように短い点列を選択し, 小さな楕円を当てはめた. Shao らの手法は多くのアウトライアを含んだ点列の選択結果となった.

図 3.5 (8) は楕円弧点列よりも多くのアウトライア点を含んだ入力データである. この場合に提案手法と Yu らの手法は正しい楕円を当てはめることができたが, RANSAC では点数の多いアウトライアの点列をインライアとして選択された.

表 3.1 は図 3.5 の入力データに対応した, 各手法のインライアの選択割合とアウトライアの選択割合を示す. 手で計測したインライア点とアウトライア点と各手法で選択した点を比較した割合を示す. 表 3.1 より提案手法と RANSAC のインライアの選択割合はほとんどの場合同等の値が得られるが, アウトライア点数が楕円弧点数より多い場合に提案手法が優れていることがわかる. また, Yu らの手法は多くの入力データに対してインライアの選択割合が小さい. 図 3.5 (3), (7) の入力データでは Yu らの手法は他の手法と異なる楕円上の点列を選択するため, 他の手法とは異なる楕円上の点列を用いて評価値を示す.

表 3.2 はそれぞれの手法の反復回数と計算時間をまとめたものである. 実験環境は, CPU は Intel Core 2Duo 3.00GHz \times 2, 主メモリ 4GB, OS Ubuntu 12.04 を用いた. RANSAC については, 50 回続けて解が変化しなかったときに収束したと判定して, 収束までの総反復回数を求めた. 乱数により試行のたびに反復回数が変わるため, 同じデータに対して 10 回の試行を行った平均を計算した. 提案手法と RANSAC, Yu らの手法は C++にて作成したシステムを用いたが, Shao らの手法は matlab 環境での動作のため, 評価から除外した. 提案手法は数回の反復で収束し, 計算

時間も高速であることがわかる。それに対して RANSAC では解が収束するまでに多くの反復が必要なことから提案手法と比べて多くの計算時間がかかっている。また Yu らの方法は多くの反復と計算時間が必要であることがわかる。

表 3.1: インライアの選択割合 / アウトライアの選択割合.

	提案手法	RANSAC	Yu らの手法	Shao らの手法
(1)	0.99 / 0.00	0.99 / 0.00	0.49 / 0.01	0.80 / 0.00
(2)	0.99 / 0.00	1.00 / 0.01	0.49 / 0.02	0.81 / 0.00
(3)	0.99 / 0.04	1.00 / 0.07	0.40 / 0.04	0.93 / 0.09
(4)	1.00 / 0.00	1.00 / 0.01	0.36 / 0.00	0.90 / 0.05
(5)	0.99 / 0.00	1.00 / 0.04	0.47 / 0.00	0.93 / 0.04
(6)	0.99 / 0.00	1.00 / 0.01	0.48 / 0.01	0.61 / 0.45
(7)	0.90 / 0.00	1.00 / 0.02	0.45 / 0.06	0.54 / 0.68
(8)	0.50 / 0.00	0.02 / 0.98	1.00 / 0.10	0.51 / 0.63

表 3.2: 計算時間と反復回数の比較. 計算時間 [ミリ秒](反復回数)

	提案手法	RANSAC	Yu らの手法
(1)	4 (5)	41 (111)	80 (20)
(2)	8 (5)	61 (113)	148 (48)
(3)	12 (7)	79 (137)	208 (17)
(4)	4 (3)	49 (152)	288 (106)
(5)	8 (9)	42 (148)	600 (178)
(6)	4 (4)	59 (218)	216 (72)
(7)	8 (8)	68 (134)	48 (10)
(8)	4 (3)	64 (143)	424 (157)

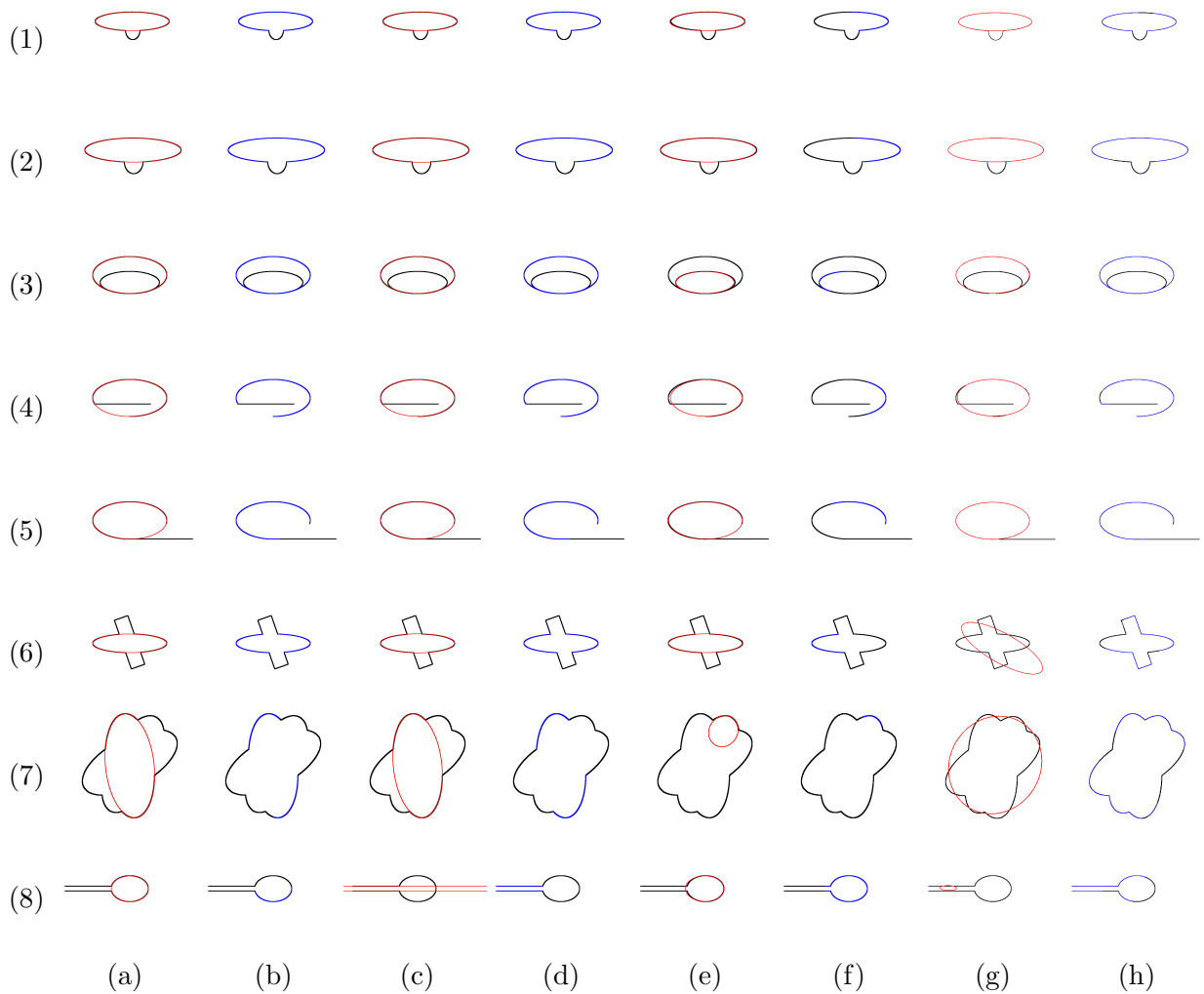


図 3.5: 楕円弧選択の比較. (a) 提案手法. (c)RANSAC. (e)Yu らの手法. (g)Shao らの手法位. 赤線:当てはめた楕円. (b), (d), (h) 各手法で選択した点列. (f)Yu らの手法で使用した点列.

3.6.2 実画像実験

図 3.6 に実画像実験結果を示す. 図 3.6 (a) は canny オペレータによって抽出したエッジ点を示す. 本実験では, 連結点数が 50 点以下の点列は除去した. 図 3.6 (a) の赤点は楕円を当てはめるために手動で選択した点列である. 図 3.6 (b), (c), (d), (e) はそれぞれ提案手法, RANSAC, Yu らの手法, Shao らの手法で選択した点列と当てはめた楕円を示す. 緑点が選択した点を示し, 赤線で当てはめた楕円を示す. 提案手法と RANSAC はどの入力データに対しても正しい楕円を当てはめることが出来たが, Yu らの手法は小さな楕円をどの入力データに対しても当てはめた. これより, Yu らの手法は入力データに対し, 小さい楕円を当てはめる傾向にあると考えられる. Shao らの手法は, どの入力データに対しても多くのアウトライアを含んだ点列の選択結果となった. Shao らの手法は入力点列を分割して, 使用する楕円弧点列の組合せを推定するとき, 分割した点列の当てはめ誤差と選択した点列の長さしか考慮しないため, 楕円弧の形状を考慮したアウトライアの分離が正しくできないためだと考えられる.

表 3.3 に図 3.6 に対応した各手法のインライアの選択割合とアウトライアの選択割合を示す. 表

より提案手法とRANSACは同様のインライア性能であるが、アウトライアの選択割合はRANSACに比べ提案手法の方が低く、正しい楕円弧選択ができていることがわかる。

表 3.4 に図 3.6 に実装環境の異なる Shao らの手法を除いた、各手法の反復計算回数と計算時間を示す。表 3.4 より、提案手法はどの入力データに対しても、比較手法よりも少ない反復計算で高速に楕円を当てはめることが可能であると言える。

表 3.3: インライアの選択割合 / アウトライアの選択割合.

	提案手法	RANSAC	Yu らの手法	Shao らの手法
(1)	1.00 / 0.00	1.00 / 0.02	0.49 / 0.08	0.95 / 0.23
(2)	1.00 / 0.00	1.00 / 0.02	0.00 / 1.00	0.71 / 0.46
(3)	0.83 / 0.02	0.83 / 0.05	0.27 / 0.00	0.36 / 0.48

表 3.4: 計算時間と反復回数の比較. 計算時間 [ミリ秒](反復回数)

	提案手法	RANSAC	Yu らの手法
(1)	3 (4)	70 (157)	228 (65)
(2)	1 (6)	16 (141)	3 (5)
(3)	4 (10)	49 (208)	56 (35)

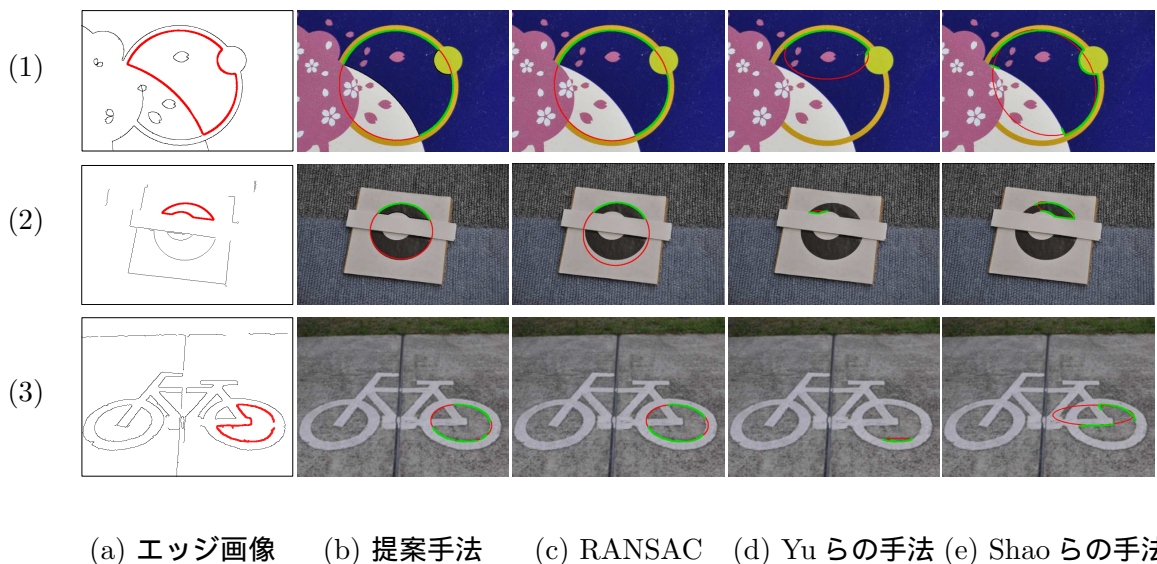


図 3.6: 実画像実験結果. (a) 赤点:選択した入力点列. (b)-(e) 赤線:当てはめた楕円, 緑:選択した点.

3.6.3 手法の問題点

この手法では、アウトライアを含む部分弧の形状によって誤差グラフのピーク値付近の曲率が小さくなる場合に、誤差グラフの曲率のみではインライアとアウトライアの判定ができない場合がある。図 3.7 にその一例を示す。図 3.7 (a) に全ての入力点列を用いて楕円を当てはめた結果、図 3.7 (b) に対応した誤差グラフを示す。矩形点列部分の部分弧の誤差グラフが凹形状になり、その区間のグラフが緩やかに変化する。そのため、図 3.7 (c) 中の青点で示す点列を誤って選択する。この問題を解決する手法として次節で述べる部分弧ごとへ楕円を当てはめる手法であるが、筆者は誤差グラフへ放物線を当てはめることにより問題を解決するアプローチも検討した。この手法については付録の 7.3 節にて詳細を述べる。

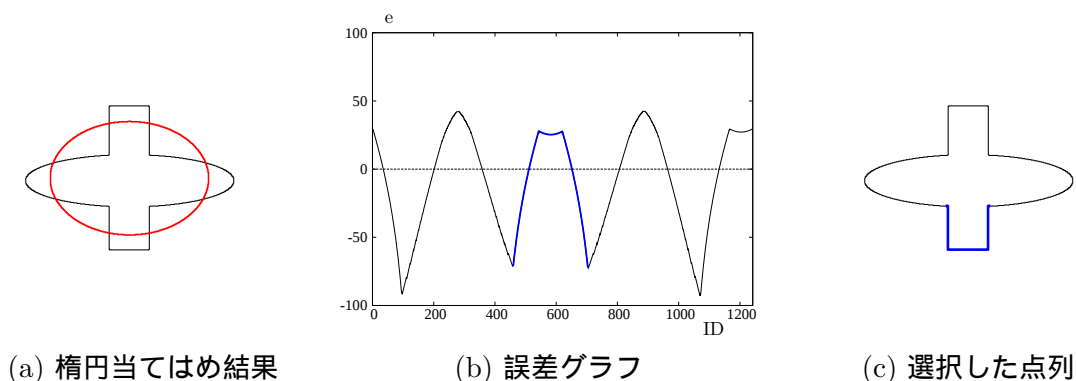


図 3.7: 誤った点列選択結果. (a) 黒線:入力点列, 赤線:当てはめた楕円. (b) 青線:選択した部分弧. (c) 青点:選択した点列.

3.7 各部分弧への楕円当てはめによる楕円弧選択

図 3.7 の例でも示したように、誤差グラフのピーク付近の誤差変化では正しく楕円上の部分弧を選択できない可能性がある。そこで我々は、誤差グラフのピーク付近の誤差変化を調べる代わりに、分割した部分弧に楕円を当てはめ、1点当たりの平均当てはめ誤差が小さい点列を選択する。

この手法では、すべての部分弧の組合せに対して楕円当てはめを行うのではなく、あくまでも単一の部分弧ごとに楕円当てはめを行い、一つの部分弧を選択する。これは、以下のような理由からである。

- 当てはめた楕円との交点によって分割した部分弧は、楕円上の点列とそれ以外の点列が組み合わせられたものである可能性があるため、Shao らの方法のような組合せ探索では正しい楕円弧点列を検出できない。
- 前節で述べた方法は、楕円弧選択と楕円当てはめを繰り返して、その過程で得られる部分弧も変化し、最終的に入力点列は楕円上の点列とそれ以外に分割される。そのための初期選択では、楕円上の点列が一部でも得られればよい。

また、この手法では前述した誤差の曲率を用いた楕円弧選択手法の考え方を引き継ぎ、1回目の点列選択と2回目以降の点列選択では、次で述べるような異なる手法で部分弧選択を行う。

3.7.1 1 回目の部分弧選択

1 回目の部分弧選択の結果はそれ以降の結果に大きな影響を与えるため、より信頼度の高い楕円弧点列を選択したい。そのため、楕円との交点で分離した部分弧だけではなく、隣接する部分弧の誤差のピーク点までの点列を追加した点列を用いた楕円当てはめと平均誤差の計算を行う。これは対象とする部分弧が楕円上の点列である場合には、それに隣接する追加対象とした点列も楕円上の点列であることが多い事実に基づいている。具体的な手順を以下にまとめる。

1. 入力点列に当てはめた楕円との交点により入力点列を部分弧に分割する。
2. 各部分弧とそれに隣接する部分弧の誤差のピーク値までの点列に楕円を当てはめる。ただし、隣接する部分弧が接部分弧⁵であった場合は、さらに隣接した部分弧の誤差のピーク値までの点列を追加する。
3. 当てはめた楕円に対する 1 点あたりの平均誤差 \bar{J} を計算し、平均誤差が最小となる部分弧を楕円弧として選択する。楕円との当てはめ誤差の計算方法は益崎らの方法と同様である。

ノイズ（以下、データの誤差をノイズと呼ぶ）のある点列 $(x_1, y_1), \dots, (x_N, y_N)$ に楕円を当てはめることは $\alpha = 1, \dots, N$ に対して

$$Ax_\alpha^2 + 2Bx_\alpha y_\alpha + Cy_\alpha^2 + 2f_0(Dx_\alpha + Ey_\alpha) + f_0^2 F \approx 0 \quad (3.11)$$

となる A, B, C, D, E, F を計算することである。式 (3.11) 中の f_0 はスケールを調節する固定した定数である⁶。式 3.11 の要素を 6 次元ベクトル

$$\xi_\alpha = (x_\alpha^2, 2x_\alpha y_\alpha, y_\alpha^2, 2f_0 x_\alpha, 2f_0 y_\alpha, f_0^2)^\top, \theta = (A, B, C, D, E, F)^\top \quad (3.12)$$

と定義すると N 点の部分弧の 1 点あたりの楕円の当てはめ誤差 \bar{J} は

$$\bar{J} = \frac{1}{N} \sum_{\alpha=1}^N \sqrt{\frac{(\xi_\alpha, \theta)^2}{(\theta, V_0[\xi_\alpha]\theta)}} \quad (3.13)$$

より求まる。

3.7.2 2 回目以降の部分弧選択

2 回目以降の点列選択では、一つ前のステップで選択した部分弧とそれ以外の部分弧を一つずつ組み合わせたペアに対して、楕円当てはめを行い、平均当てはめ誤差が最小なペアを選択する。以下に、2 回目以降の部分弧選択の手順を示す。

1. 一つ前のステップで選択した部分弧と、その点列に当てはめた楕円によって分割される部分弧の組み合わせに対して、楕円を当てはめる。
2. 平均当てはめ誤差を計算し、最小となる部分弧を選択する。
3. 選択した部分弧に含まれる点数が、一つ前のステップで選択したものより少なければ、前のステップで選択した部分弧とその楕円パラメータを出力して終了する。また、他に選択できる部分弧がない場合は、現在選択している部分弧と楕円パラメータを出力して終了する。そうでなければ、手順 1 に戻る。

⁵点列に当てはめた楕円と点列が接しており、部分弧内の最大の当てはめ誤差が閾値（実験では 1.5 以下の当てはめ誤差とした。）以下の部分弧を接部分弧と定義している。

⁶実験では $f_0 = 600$ とした。

3.8 実験

提案手法の有効性を示すために、次に示す3つの実験を行った。実験では、検証事項が明確に判断できるように簡単な図形を描画した入力画像を用いた。

1. 楕円弧の選択過程の確認
2. 楕円弧選択性能向上の確認
3. Shao らの手法との楕円弧選択性能の比較

3.8.1 楕円弧の選択過程の確認

本実験では提案手法による点列選択の動作過程を示す。図 3.8 に入力データと1回目の楕円当てはめ結果を示す。この当てはめ結果より、各部分弧を左右の部分弧の誤差のピーク値となる点まで拡張した点列に楕円を当てはめた結果とそれぞれの当てはめ誤差平均値を図 3.9 に示す。同様にして、図 3.10 に3回目の点列選択過程、図 3.11 に4回目の点列選択過程をそれぞれ示す。

図 3.9 では、誤差平均が最小の(4)が選択され、このときの接部分弧(121点)を選択する。図 3.9 (4)で得られた点列とそのときに得られる部分弧の組み合わせを求めた結果を図 3.10 に示す。

図 3.10 (5)が誤差平均が最小の当てはめ結果となり、このときの接部分弧の点数が121点から279点に増えているため、選択される。

図 3.11 (1)が最小の誤差平均になるが、このときの接部分弧点数が0点になるため、この入力での点列選択は図 3.10 (5)の接部分弧を選択して楕円弧選択処理を終了する。

図 3.10 (5)より入力点列中の楕円弧を正しく選択できることを確認した。

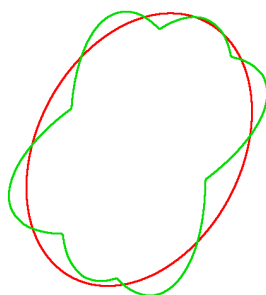


図 3.8: 1回目の楕円当てはめ(緑点:選択点, 赤線:当てはめた楕円)

3.8.2 楕円弧選択性能の向上確認

前節で提案した部分弧ごとの誤差の変化よりグラフの曲率を求め楕円弧を選択する手法と提案手法による改良手法との楕円弧選択性能の比較を行った。図 3.12 (a) は入力点列とそれに当てはめた初期楕円である。また、図 3.12 (b) は、初期楕円に対する入力点列の当てはめ誤差をプロットしたグラフである。図 3.12 (c-1) は、誤差の変化に基づく手法で1回目の反復で得た楕円弧であるが、誤差の変化に基づく判定では楕円弧ではない点列が選択されていることがわかる。一方、図

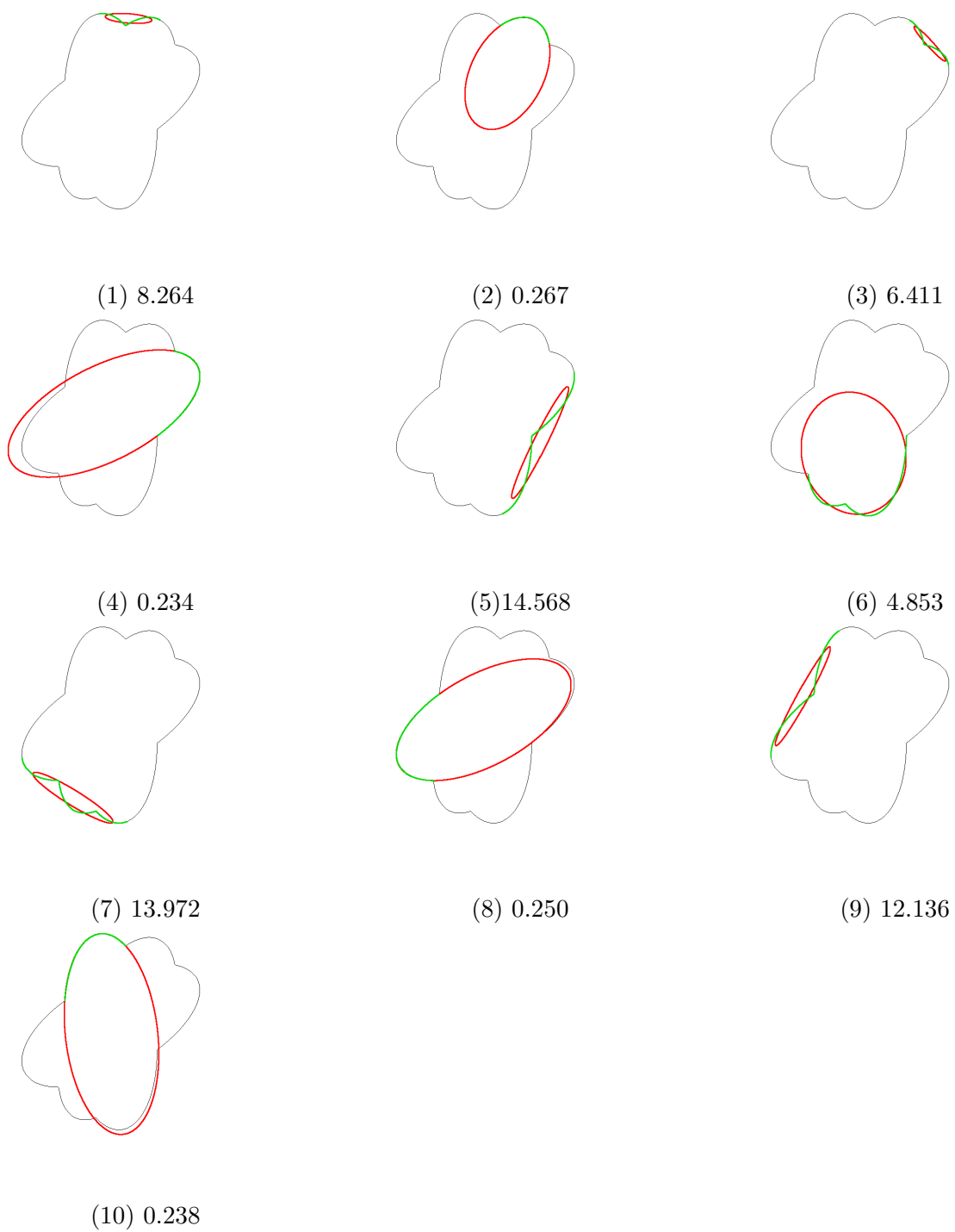


図 3.9: 1 回目の点列選択過程 (使用した点列の当てはめ誤差平均)

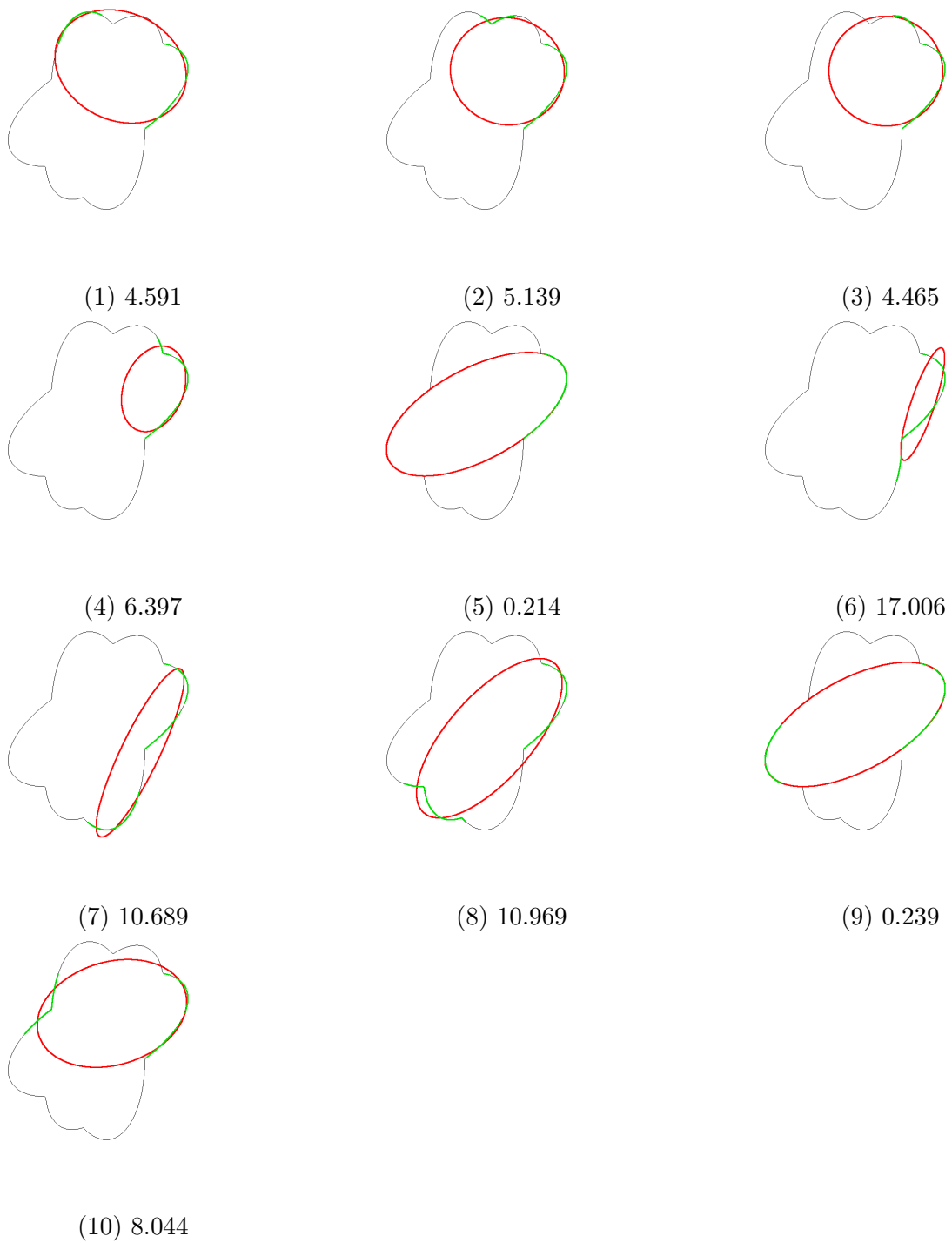


図 3.10: 2 回目の点列選択過程 (使用した点列の当てはめ誤差平均)

3.12 (d-1) は、改良手法によって選択した楕円弧である。楕円弧ごとの楕円の当てはめ誤差に基づいて判定することにより、楕円上の点列が得られている。また、図 3.12 (c-2), (c-3) は、誤差の変化に基づく手法で最終的に抽出した楕円弧とそれに当てはめた楕円である。1 回目の反復で間違っただ点列を選択した影響で、最終的に正しい楕円弧を抽出できず、正しい楕円が得られていない。しかし、改良手法では正しい楕円弧を抽出し、それにより正しい楕円が得られている（図 3.12 (d-2), (d-3)）。

図 3.13 は、改良手法による 1 回目の楕円弧選択での選択対象点列とそれに当てはめた楕円およびその平均当てはめ誤差である。図 3.13 (c) の点列はオリジナルの手法で選択された点列であるが、楕円の当てはめ誤差は大きいことがわかる。そして、楕円当てはめの平均誤差に基づいて図 3.13 (d) の点列が正しく選択されている。

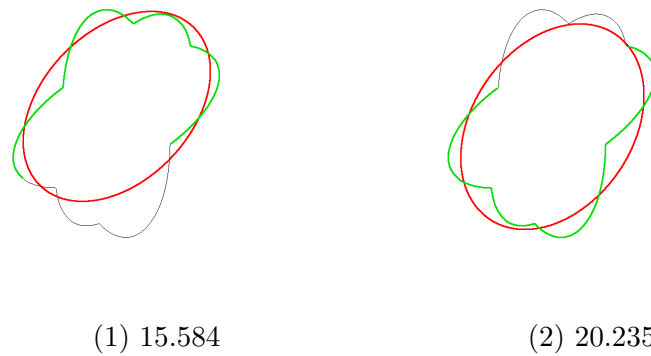


図 3.11: 3 回目の点列選択過程 (使用した点列の当てはめ誤差平均)

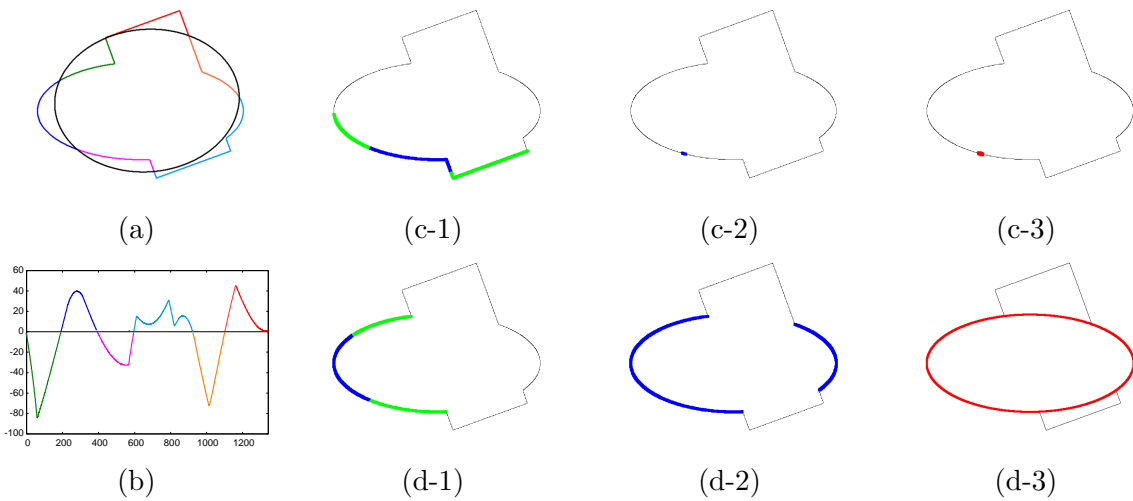


図 3.12: 楕円弧選択性能の比較. (a) 入力点列とそれに当てはめた初期楕円, (b) 初期楕円の当てはめ誤差グラフ, (c-1), (d-1) 初期選択楕円弧 (改良前の手法, 改良後の手法), 青の点列は選択した部分弧, 緑の点列は拡張した隣の部分弧, (c-2), (d-2) 最終的に得られた楕円弧 (改良前の手法, 改良後の手法), (c-3), (d-3) 得られた楕円弧に当てはめた楕円 (改良前の手法, 改良後の手法)

3.8.3 楕円弧選択性能の比較

Shao らの手法との比較を行い、楕円弧抽出の効率と性能の面で提案手法が優れていることを示す。図 3.14 (1)-(a) は Shao らの手法によって分割した入力点列を分割した点列ごとに色分けして表示したものである。Shao らの手法では、分割したこれらの点列の組合せに対して楕円当てはめ誤差が最小になる組合せを総当りで計算する。その結果、得られた点列が図 3.14 (2)-(a) の青色の点列である。この例では 12 個の部分弧が得られ、楕円当てはめは $\sum_{i=6}^{12} {}_{12}C_i = 2510$ 通りの部分弧の組合せに対して行った。また、最終的に抽出した点列は楕円上の点列の一部のみであったが、正しい楕円が得られている。

一方、図 3.14 (1)-(b) は入力点列を提案手法によって分割した点列であり、入力点列に当てはめた楕円との交点により分割を行っている。提案手法では、当てはめた楕円との交点により機械的に点列を分割しているが、点列選択と楕円当てはめを繰り返すたびに点列の分割を行うため、最終的に図 3.14 (2)(b) のように楕円上の点列をもれなく抽出することができており、それに当てはめた楕円も正しい結果となっている。また、提案手法では、全体で 25 回の楕円当てはめが行われ、Shao らの手法と比べて点列の分割の処理コストと、楕円当てはめの回数の面で効率的であることがわかる。

図 3.15 は別の入力点列に対する実験結果である。Shao らの手法では、異なる楕円上の点列を抽出しており、正しい楕円が得られていない。一方、提案手法では、正しい楕円弧を抽出しており、その結果、正しい楕円が得られている。図 3.15 (1) の初期の点列分割結果を見ると、二つの手法でおおよそ似た分割結果を得ている。Shao らの手法は、この点列分割結果に完全に最終結果が依存するが、提案手法では、反復処理のたびに点列の分割を繰り返すため、反復のたびに得られる分割結果が実際の楕円弧に一致したものに自動的に近づいていく利点がある。

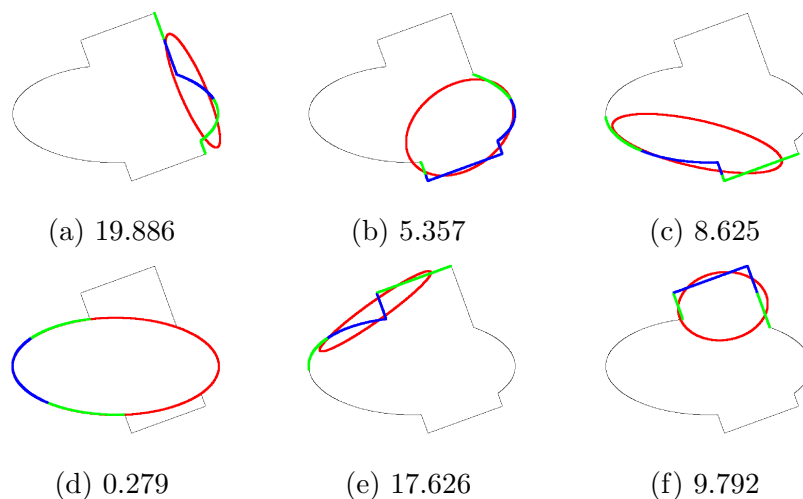


図 3.13: 提案手法における初期楕円弧の選択過程。青の点列は選択した部分弧、緑の点列は拡張した隣の部分弧、赤の楕円は青の点列と緑の点列に当てはめた楕円。数値は当てはめた楕円に対する入力点列の平均誤差。

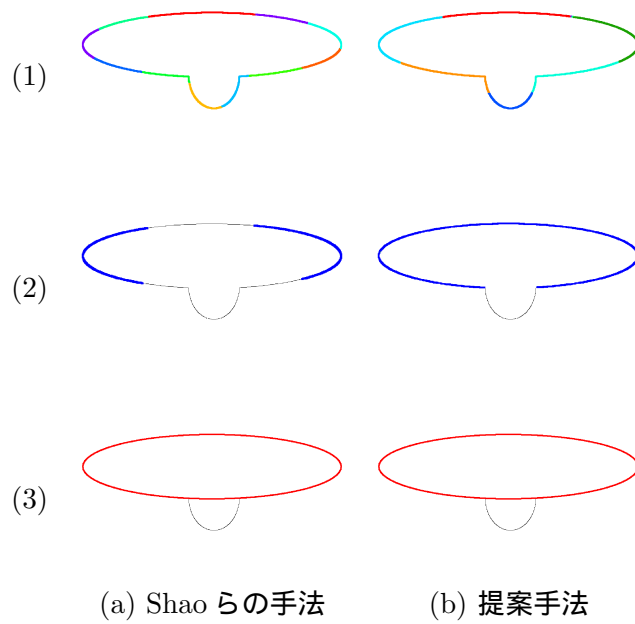


図 3.14: 楕円弧選択性能の比較 1. (1) 分割した部分弧, (2) 最終的に抽出した点列, (3) (2) の点列に当てはめた楕円.

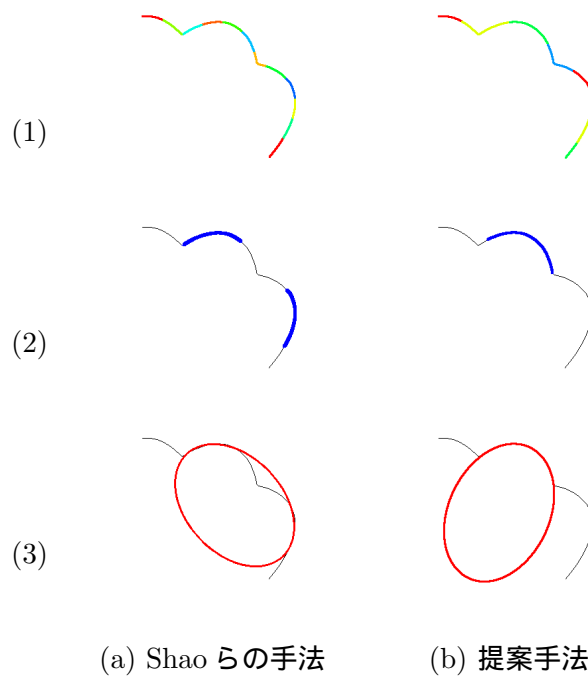


図 3.15: 楕円弧選択性能の比較 2. (1) 分割した部分弧, (2) 最終的に抽出した点列, (3) (2) の点列に当てはめた楕円.

第4章 同一楕円上の楕円弧点列の統合

本章では, 前章で述べた手法により得られる楕円弧点列に対し, さらに高精度に楕円を当て編めるための楕円弧点列の統合手法について述べる. まず統合のための幾何学的 MDL モデル選択について述べ, 既存の楕円検出法と比較して, 提案手法による改良による性能の向上について詳しく説明する.

4.1 幾何学的 MDL モデル選択

二つのエッジ点列にそれぞれ異なる楕円を当てはめた場合と一つの楕円を当てはめた場合では, 二つの楕円を当てはめる方がパラメータの自由度が高くなり, よく当てはまる. このような複数のモデルに対する当てはまり具合を比較する場合に, 当てはめ誤差だけでなくモデルのパラメータの自由度も考慮しなければならない. これをモデル選択と呼び, その尺度として式 (4.1) で示す幾何学的 MDL[16] を求める.

$$\text{G-MDL} = J - (dN + p')\sigma^2 \log \left(\frac{\sigma}{L} \right)^2 \quad (4.1)$$

ここで, J はモデルの当てはめ誤差, N はデータ数, d はモデルの次元, p' はモデルパラメータの自由度, σ^2 はノイズレベル (誤差の分散), L は任意の定数を示す.

4.1.1 幾何学的 MDL

式 (4.1) で示した幾何学的 MDL を楕円当てはめに適用する.

データ数 N は対象とするエッジ点数とする. 本研究で当てはめる円錐曲線の式のデータ空間は xy 平面の 2 次元であり, 一つの拘束条件として式 (4.2) がある.

$$Ax_\alpha^2 + 2Bx_\alpha y_\alpha + Cy_\alpha^2 + 2f_0(Dx_\alpha + Ey_\alpha) + f_0^2 F \approx 0 \quad (4.2)$$

$$\xi_\alpha = (x_\alpha^2, 2x_\alpha y_\alpha, y_\alpha^2, 2f_0 x_\alpha, 2f_0 y_\alpha, f_0^2)^\top, \theta = (A, B, C, D, E, F)^\top \quad (4.3)$$

拘束条件はデータ空間上の曲線を定義するので, モデルの次元 d は 1 である. このモデルの次元で, 式 (4.3) に示す 6 次元ベクトルで表される楕円パラメータ θ の自由度 p' は 5 である.

モデル当てはめの誤差 J は,

$$J = \sum_\alpha^N \sqrt{\frac{(\xi_\alpha, \theta)^2}{(\theta, V_0[\xi_\alpha]\theta)}} \quad (4.4)$$

より求める.

以上より楕円当てはめに関する幾何学的 MDL は式 (4.5) となる.

$$\text{G-MDL} = J - (N + 5)\sigma^2 \log \left(\frac{\sigma}{L} \right)^2 \quad (4.5)$$

4.1.2 ノイズレベルの推定

モデル選択の計算をするには、ノイズレベル σ^2 を求める必要がある。二つのエッジ点列にはそれぞれ楕円が当てはまっており、それらが同一の楕円上の点列であるかを判定する。このとき、二つのエッジ点列は常に同一楕円上にある点列であるとは限らないので、ノイズレベルを二つのエッジ点列にそれぞれ楕円を当てはめた場合のモデルより求める。

$$\sigma^2 = \frac{J_1 + J_2}{N_1 + N_2 - 10} \quad (4.6)$$

ここで二つの点列それぞれに楕円を当てはめたときの当てはめ誤差を J_1, J_2 とし、それぞれのエッジ点数を N_1, N_2 とする。

4.2 幾何学的 MDL を用いた菅谷の統合判定

二つのエッジ点列にそれぞれ異なる楕円を当てはめた場合と一つの楕円を当てはめた場合は、二つの楕円を当てはめる方がパラメータの自由度が高くなり、よく当てはまる。このような複数のモデルに対する当てはまり具合を比較する場合に、当てはめ誤差だけでなくモデルのパラメータの自由度も考慮しなければならない。これをモデル選択と呼び、その尺度として幾何学的 MDL[16] を求める。

二つの楕円を当てはめた場合の幾何学的 MDL を $G\text{-MDL}_{1,2}$ とし、一つの楕円を当てはめた場合の幾何学的 MDL を $G\text{-MDL}_{1+2}$ とすると、式 (4.7) となる。

$$\begin{aligned} G\text{-MDL}_{1,2} &= J_1 + J_2 - (N_1 + N_2 + 10)\sigma^2 \log\left(\frac{\sigma}{L}\right)^2 \\ G\text{-MDL}_{1+2} &= J_{1+2} - (N_1 + N_2 + 5)\sigma^2 \log\left(\frac{\sigma}{L}\right)^2 \end{aligned} \quad (4.7)$$

式 (4.7) を比較し、 $G\text{-MDL}_{1,2} > G\text{-MDL}_{1+2}$ であれば二つの点列を統合する。ここで二つの点列それぞれに楕円を当てはめたときの当てはめ誤差を J_1, J_2 とし、それぞれのエッジ点数を N_1, N_2 とする。また、 σ^2 は誤差の分散、 L は任意の定数を示す。

4.3 提案手法

菅谷 [40] は二つの点列が同一楕円上の点列であるかを、二つの点列に別々に楕円を当てはめた残差と一つの楕円を当てはめた残差の比較にモデルの自由度を考慮した幾何学的 MDL によるモデル選択を導入することで、しきい値を用いない楕円弧統合判定を提案した。この方法では、1) 与えられた楕円弧のすべての組合せに対して楕円を当てはめ、2) その当てはめ誤差の小さい順に楕円弧の組を並べ替える。そして、当てはめ誤差の小さい順に幾何学的 MDL による統合判定を行い、楕円弧の統合が行われた場合には、統合した楕円弧と残りの楕円弧に対して上記の処理を行い、これを統合される楕円弧がなくなるまで繰り返す。そのため、入力楕円弧の数が多いと楕円弧の組合せ数が大きくなり、非常に処理時間がかかってしまう。

本研究では、すべての組合せに対して楕円当てはめを行い、当てはめ誤差計算を行うのではなく、楕円弧の形状や位置などの幾何学的な情報を用いて統合判定を行うべき組合せを選別することで、菅谷の手法よりも統合判定回数を削減する。

楕円弧統合を行う組合せを選別するために、以下の特徴を考慮する。

1. 当てはめた楕円の弧長に対する楕円弧の占める割合

例えば、当てはめた楕円の弧長に対して楕円弧点列の数がその大部分を占めていれば、入力点列に対してほぼ正しい楕円が得られていると考えられる。そのため、この割合が大きい楕円弧は統合判定から除外する。

2. 当てはめた楕円に対する楕円弧の位置

楕円弧どうしの位置が当てはめた楕円から見て離れた位置にある場合には、それらは別々の楕円上の点列である可能性が高い。また、楕円弧がある程度の長さを持つ場合には、当てはめた楕円の右上に位置する弧どうしのように、楕円上で似た位置関係にある楕円弧が同一の楕円上の点列である可能性は低い。これらを考慮して、同一楕円上にない可能性がある楕円弧どうしの統合判定は行わない。

以上を考慮した楕円弧の選別を行うために、当てはめた楕円に対する入力楕円弧の位置に基づく楕円弧選別手法を提案する。

4.4 楕円弧の分類

当てはめた楕円の中心を原点として、水平方向および垂直方向に軸を取り、その軸によって楕円を図 4.1 に示す 4 つの領域に分割する。そして、楕円当てはめに使用した楕円弧点列と楕円中心の座標を比較して、その大小関係により、楕円弧点列を 4 つの領域に分類する。

図 4.1 の青点 (x, y) の例では、楕円中心 (x_c, y_c) との大小関係を見ると、 $x - x_c < 0, y - y_c > 0$ であるから、領域 2 に分類される。そして、各領域 $\mathcal{R}_i, i = 1, \dots, 4$ に含まれる点の数を $N(\mathcal{R}_i)$ とする。

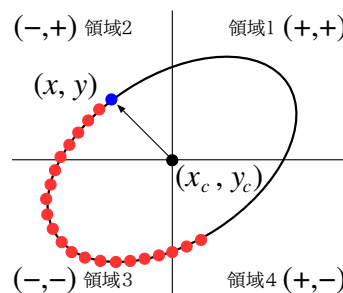


図 4.1: 楕円弧点列の分類. 楕円を楕円中心を原点とした軸により 4 つの領域に分割する. 領域 1 の $(+, +)$ はこの領域上の点の X, Y 座標から楕円中心の座標 (x_c, y_c) を引いた値がそれぞれ正になることを表す.

楕円の中心点の導出について以下に述べる.

4.4.1 楕円の中心座標の導出

中心座標の導出のため、楕円を含む 2 次曲線の式 (2.21) を次の用に表現を変える. 3 次元ベクトル \mathbf{x} , 3×3 対称行列 Q を

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ f_0 \end{pmatrix}, Q = \begin{pmatrix} A & B & D \\ B & C & E \\ D & E & F \end{pmatrix} \quad (4.8)$$

と置くと、楕円の方程式は次のように書ける。これは、 x に関する 2 次形式と呼ばれる。

$$(x, Qx) = 0 \quad (4.9)$$

さらに、2 次元ベクトル r, c , 2×2 対称行列 S を

$$r = \begin{pmatrix} x \\ y \end{pmatrix}, c = \begin{pmatrix} D \\ E \end{pmatrix}, S = \begin{pmatrix} A & B \\ B & C \end{pmatrix} \quad (4.10)$$

と置くと、楕円の方程式は次のように書ける。

$$(r, Sr) + 2f_0(c, r) + Ff_0^2 \quad (4.11)$$

楕円中心 (x_c, y_c) を、2 次元ベクトル $r_c = (x_c, y_c)^\top$ で表す。式 (4.9) の楕円を $-r_c$ だけ平行移動すると次のように書ける。

これを展開すると

$$(r, Sr) + 2(Sr + f_0c, r) + (r_c, Sr_c) + 2f_0(c, r_c) + Ff_0^2 = 0 \quad (4.12)$$

この楕円は、原点を中心に行っているはずであるから、

$$Sr_c + f_0c = 0 \quad (4.13)$$

でなければならない。楕円（および双曲線）では、行列 S は正則行列であり、中心 r_c が次のように計算できる。

$$r_c = -f_0S^{-1}c \quad (4.14)$$

4.5 当てはめた楕円の弧長に対する楕円弧の占める割合を用いた選別

前節の方法により楕円弧点列ごとに各領域に含まれる点の数を調べ、その情報を用いて、楕円弧統合判定の対象から除外する楕円弧の検出と、統合判定を行う組合せの選別の 2 段階の処理により、楕円弧統合判定を行う楕円弧を絞り込む。

4.5.1 統合判定から除外する楕円弧の選択

提案手法では以下の条件を満たす楕円弧を統合判定の対象から除外する。

1. 3 つ以上の領域に点が存在する場合

正しい楕円を得るのに十分な点が存在するものとして、楕円弧統合判定の対象から除外する。

2. 楕円の長軸方向の 2 領域に点が存在する場合

楕円の両側の点列から楕円を当てはめると正しい楕円が得られる可能性が高いため、楕円弧統合判定の対象から除外する。ただし、楕円の短軸方向の 2 領域に点が存在する場合は、長軸方向に楕円がずれる可能性があるため、楕円弧統合判定の対象からは除外しない。長軸方向の点の導出については次節で詳細を述べる。

4.5.2 楕円の長軸と短軸の計算

前節で述べた楕円の長軸方向の2領域に点列が存在するかどうかの判定について、その判定方法について説明する。

式(4.13)を用いて楕円の式(4.9)は次のように書ける。

$$(\mathbf{r} - \mathbf{r}_c, \mathbf{S}(\mathbf{r} - \mathbf{r}_c)) = c, c = (\mathbf{r}_c, \mathbf{S}\mathbf{r}_c) - Ff_0^2 \quad (4.15)$$

ここで、

$$\mathbf{S}' = \mathbf{S}/c \quad (4.16)$$

とすると、次のように書きなおせる。

$$(\mathbf{r} - \mathbf{r}_c, \mathbf{S}'(\mathbf{r} - \mathbf{r}_c)) = 1 \quad (4.17)$$

行列 \mathbf{S}' の固有値 λ_1, λ_2 に対応する固有ベクトルの正規直交系を $\mathbf{u}_1, \mathbf{u}_2$ とすると、楕円をその主軸方向 $\mathbf{u}_1, \mathbf{u}_2$ が座標軸に平行になるように中心 \mathbf{r}_c の周りに回転すると、その方程式は次のように書ける。

$$\lambda_1(x - x_c)^2 + \lambda_2(y - y_c)^2 = 1 \quad (4.18)$$

ここで、

$$a = \frac{1}{\sqrt{|\lambda_1|}}, b = \frac{1}{\sqrt{|\lambda_2|}} \quad (4.19)$$

と置くと、式(4.18)は次のように書ける。

$$\frac{(x - x_c)^2}{a^2} + \frac{(y - y_c)^2}{b^2} = 1 \quad (4.20)$$

従って、楕円の長軸と短軸は行列 \mathbf{S}' の固有値を計算すればよいことになる。

前節で求めた4領域中の2領域に点列が存在する点列の中から、導出した長軸方向のベクトル \mathbf{u}_1 を用いて、長軸方向の2領域に点列が存在するものを組み合わせから除外する。

4.6 当てはめた楕円に対する楕円弧の位置関係を用いた選別

残った楕円弧の統合判定候補に対して、それぞれの位置関係を用いた選別を行う。まず、4つの領域の中で含まれる点の数が最大な領域をその楕円弧の代表領域として定め、それぞれ代表領域が異なる楕円弧どうしのみを統合判定の対象とする。また、楕円弧どうしの画像中の位置関係を考慮するために、上記の条件を満たす組合せの中で、一方の楕円弧に当てはめた楕円の領域内¹にもう一方の楕円弧が存在する場合に、その組合せを統合判定の対象とする。楕円領域の拡大について、次に述べる。

4.6.1 楕円領域の拡大

提案手法では、楕円弧に当てはめた楕円周辺に使用した楕円弧と同じ楕円物体上の点列が存在すると仮定し、当てはめた楕円を拡大した領域内の点列を点列統合判定の候補とする。楕円弧に当てはめた楕円に対して、各主軸方向に δ だけ拡大した楕円 $Q^{(+)}$ は次のように計算できる。

¹実験では、楕円弧に楕円を当てはめた楕円に対して長軸と短軸をそれぞれ $\delta = 20$ 画素だけ拡張した楕円領域内を対象とした [33]。

1. 行列 S , ベクトル c , スカラー c を次のように定義する.

$$S = \begin{pmatrix} A & B \\ B & C \end{pmatrix}, c = \begin{pmatrix} D \\ E \end{pmatrix}, c = (c, S^{-1}c) - F \quad (4.21)$$

2. 行列 S の固有値 λ_1, λ_2 とし, 対応する単位固有ベクトルを u_1, u_2 とする.

3. $\lambda_1^{(+)}, \lambda_2^{(+)}$ を次のように計算する.

$$\lambda_1^{(+)} = \frac{c}{(\sqrt{c/\lambda_1} + \delta/f_0)^2}, \lambda_2^{(+)} = \frac{c}{(\sqrt{c/\lambda_2} + \delta/f_0)^2} \quad (4.22)$$

4. 次のように行列 $S^{(+)}$ を計算する.

$$S^{(+)} = U \begin{pmatrix} \lambda_1^{(+)} & 0 \\ 0 & \lambda_2^{(+)} \end{pmatrix} U^T \quad (4.23)$$

ただし, U は u_1, u_2 をこの順に列とする 2×2 行列である.

5. 次のように行列 $Q^{(+)}$ を計算する.

$$Q^{(+)} = \begin{pmatrix} S^{(+)} & S^{(+)}S^{-1}c \\ (S^{(+)}S^{-1}c)^T & F + (c, S^{-1}(S^{(+)} - S)S^{-1}c) \end{pmatrix} \quad (4.24)$$

4.6.2 統合判定を行う組合せの選別

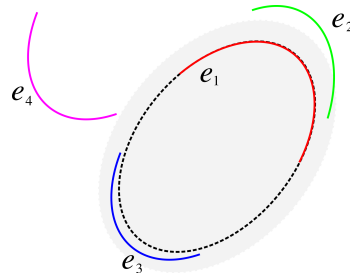


図 4.2: 統合判定を行う組合せの判定.

本節で述べた選別方法の流れを, 例として図 4.2 の楕円弧 e_1 の統合判定を対象に述べる. 楕円弧 e_2 は楕円弧 e_1 と代表領域が同じため, $\{e_1, e_2\}$ は統合判定の対象にしない. 次に楕円弧 e_3 は代表領域が異なり, 更に楕円弧 e_1 に当てはめた楕円 (図中の点線で示した楕円) 領域を拡大した領域 (図中の網掛けの楕円領域) に楕円弧 e_2 の点が含まれるため, $\{e_1, e_3\}$ は統合判定の対象とする. 最後に, e_4 は代表領域は楕円弧 e_1 と異なるが, 網掛けの楕円領域内に点が含まれないため, $\{e_1, e_4\}$ は統合判定の対象にしない.

以上の方法で選別した楕円弧組に対して, 幾何学的 MDL による統合判定を行い, 楕円弧の統合を行う.

4.7 実験

提案手法の有効性を示すために、次に示す2つの実験を行った。

1. 楕円弧統合効率の向上確認
2. 楕円検出性能の比較

4.7.1 楕円弧統合の効率向上の確認

図 4.3 (a) の入力点列を用いて、提案手法によって楕円弧統合の効率が向上することを確認する実験を行った。図 4.3 (b) は入力楕円弧に楕円を当てはめた結果である。従来のアルゴリズムでは、この8個の点列のすべての組合せに対して楕円当てはめを行い、当てはめ誤差の小さい順に統合判定処理を行っていく。図 4.3 (c) は、提案手法の第1段階で統合判定対象から除外した点列である。

次に残った点列に対して、図 4.1 に示した4つの領域の中から代表領域を割り当てた結果を代表領域ごとに色分けして図 4.3 (d) に示す。この結果をもとに統合判定を行う点列の組合せを求めたところ、5組の点列が統合判定の対象となり、オリジナルの方法よりも統合判定対象となる点列を効率的に削減することができている。最後に図 4.3 (e) が楕円弧の統合結果を示す。同じ色の点列が同一楕円上の点列として判定されたものである。そして、図 4.3 (f) がそれに当てはめた楕円である。なお、オリジナルの方法でも最終的には同じ結果を得たが、提案手法により処理の効率を上げながら、間違った統合結果を減らす効果も期待できる。

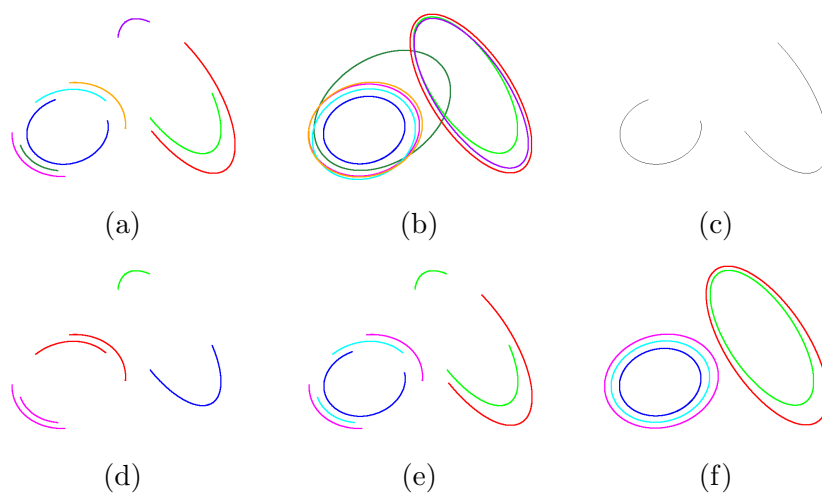


図 4.3: 楕円弧統合の効率向上の確認. (a) 実験に使用したエッジ点列, (b) (a) に当てはめた楕円, (c) 統合対象から除外した点列, (d) 統合対象点列の領域割り当て結果, (e) 楕円弧の統合結果, (f) 最終的に得られた楕円.

4.7.2 楕円検出性能の比較

最後に、実画像を入力²として様々な楕円検出手法との楕円検出性能の比較を行った。

²入力画像の一部は、Caltech256[6] データセットを利用した。

楕円検出手法には提案手法の他に, Shao らの手法 [38], Yu らの手法 [47], Fornaciari らの手法 [4] を用いた. Yu らの手法のパラメータである, 楕円と判定するための当てはめた楕円に対する使用した点列が占める楕円弧角度を 60 度以上とした. また実験は, CPU Intel Core i5-4690 3.50GHz, 主メモリ 16GB, OS Ubuntu 12.04 LTS の環境で行った.

楕円の検出結果

図 4.4 に実験結果を示す. 図 4.4 (a) は入力画像, 図 4.4 (b) はそれぞれの楕円検出手法に入力した点列, 図 4.4 (c), (d), (e), (f) はそれぞれ, 提案手法, Shao らの手法, Yu らの手法, Fornaciari らの手法によって検出した楕円である. ただし, すべての手法で, 短軸と長軸の長さの比が 0.3 よりも小さい細長い楕円と, 短軸の長さが 10 ピクセルより短い小さな楕円は誤検出と判断して描画していない. 図の結果より, すべての手法で概ね良好な楕円検出結果が得られたが, 提案手法では図 4.4(2) のお盆の縁のように短い点列がばらばらに得られた部分に対しても正しい楕円が検出されていることがわかる. また, 図 4.4(3) では, 同一楕円上の点列が複数に分離されていたり, 楕円上の点列以外の点列と連結している場合でも, 提案手法による点列の分割と複数楕円の統合によって正しい楕円が検出できている. 一方, Fornaciari らの手法では, 楕円の統合が行われない影響で 1 つの円形部分に複数の楕円が検出されていることがわかる.

楕円の精度評価

検出した楕円の精度評価と再現率の評価を行い, 提案手法の有効性を確認した. 検出した楕円の精度評価は Prasad らの論文 [35] で用いられている式 (4.25) を用いて行った.

$$D = 1 - \frac{\text{count}(\text{XOR}(\bar{\mathcal{L}}, \mathcal{L}_\alpha))}{\text{count}(\text{OR}(\bar{\mathcal{L}}, \mathcal{L}_\alpha))} \quad (4.25)$$

ここで, $\bar{\mathcal{L}}$ は正解の楕円内に含まれる画素の集合, \mathcal{L}_α は検出した楕円 α 内に含まれる画素の集合を表し, $\text{count}(\text{XOR}(\bar{\mathcal{L}}, \mathcal{L}_\alpha))$ は $\bar{\mathcal{L}}$ と \mathcal{L}_α のどちらか一方の領域にのみ含まれる点の数, $\text{count}(\text{OR}(\bar{\mathcal{L}}, \mathcal{L}_\alpha))$ は, $\bar{\mathcal{L}}$ と \mathcal{L}_α のいずれかの領域に含まれる点の数を表す. 従って, D は正解の楕円と検出した楕円の重なり具合を表し, 値が 1 に近いほど正しい楕円が得られたことを意味する.

図 4.4 (a) の入力画像上に描画した赤い楕円は, 検出した楕円の精度評価を行うために指定した楕円である. これらは, シーン中で円形をしており, その円弧上に全体のおよそ 50% 以上の点列が得られていると判断したものである. そして, 楕円パラメータの正解値は, 円弧上に得られたエッジに基づいて最尤推定法により算出した. その際, 検出したエッジ点のみでは正しい楕円形状が得られないものに関しては目視で円弧上の点を追加した上で楕円パラメータを推定した.

精度評価の結果を表 4.1, 表 4.3 に示し, 各手法で評価対象とした楕円の中で検出できた数を表 4.2 にそれぞれ示す. 表中の列の番号は図 4.4 に示した画像番号である. また, 各手法について表 4.1 は, 図 4.4(a) に示した評価対象の楕円に対する検出した楕円の平均評価値を表す. このとき, 同一楕円上に複数の楕円が検出されたものもすべて含めて楕円の評価値をそれぞれ求め平均値を取得し, 検出されなかった評価対象の楕円は評価値 0 として平均値の導出に加えた. 表 4.3 は, 評価値が 0 となったものを除いた平均値を表す. また, 画像例ごとに最高評価値を太文字で強調した.

表 4.1 では, 多くの画像で提案手法が精度よく楕円を検出できていることがわかる. 表 4.2 より, 提案手法は対象の楕円の検出漏れが少なく, 既存手法よりも多くの楕円を検出している場合が多く, 高い再現率によって表 4.1 の結果が得られているといえる. これに対し, 表 4.3 は, 検出できた評価対象の楕円のみについて評価するため, 対象楕円が検出できなかった結果でも評価値

平均が上がる．その中でも提案手法は，既存手法と比較して精度の高い結果が多く得られ，正しい楕円形状が得られていることも確認できた．表 4.1 で示した全ての画像の評価値を平均した値は，提案手法 0.889，Shao らの手法 0.717，Yu らの手法 0.376，Fornaciari らの手法 0.767 となり，この値からも提案手法の有効性が確認できる．

提案手法の結果では，楕円の検出数が多いが，画像中の楕円物体でない物体上でも楕円を検出してしまってるが，本手法は画像中より楕円をまれ無く検出するための基礎手法であるため，この問題を考慮しない．

計算時間

それぞれの手法について計算時間を計測した結果³を表 4.4 に示す．Fornaciari らの手法は非常に高速である．Yu らの手法と提案手法はほぼ同程度の計算時間である．Yu らの手法も我々の手法と同様に楕円当てはめを繰り返して楕円弧を選択する枠組みであるが，提案手法は楕円弧の統合処理が加わるため，提案手法の方が若干ではあるが計算時間が長い傾向にある．Shao らの手法だけは Matlab を用いて実行をしているため，単純には計算時間を比較できないが，入力点列の部分弧分割に計算時間がかかっているものと思われる．

また，提案手法について，楕円弧の統合対象を選別せずに検出した全ての楕円弧を統合対象にする場合（表 4.4 中では「効率化処理なし」と記述した）との計算時間の比較を行うと，検出した楕円弧の数が多いほど提案手法の効果が現れていることがわかる．ただし，統合対象の点列数に応じて計算時間が長くなる問題は完全には解決できていないため，今後の検討課題となる．

表 4.1: 検出されなかった楕円を含めて評価対象とした楕円の評価値平均の比較．

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
提案手法	0.979	0.772	0.931	0.986	0.651	0.970	0.857	0.964
Shao	0.410	0.812	0.719	0.622	0.647	0.714	0.655	0.970
Yu	0.532	0.631	0.376	0.310	0	0.414	0.334	0.413
Fornaciari	0.306	0.657	0.802	0.867	0.266	0.821	0.914	0.886

表 4.2: 評価対象楕円の検出数の比較．(検出した楕円数/評価対象の楕円数)．

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
提案手法	5/5	6/6	10/10	7/7	12/16	3/3	6/6	4/4
Shao	3/5	6/6	8/10	5/7	11/16	3/3	5/6	4/4
Yu	3/5	4/6	6/10	3/7	0/16	1/3	4/6	2/4
Fornaciari	1/5	4/6	9/10	7/7	3/16	3/3	6/6	4/4

³Shao らの手法のみ Matlab を用いて実験を行っている．

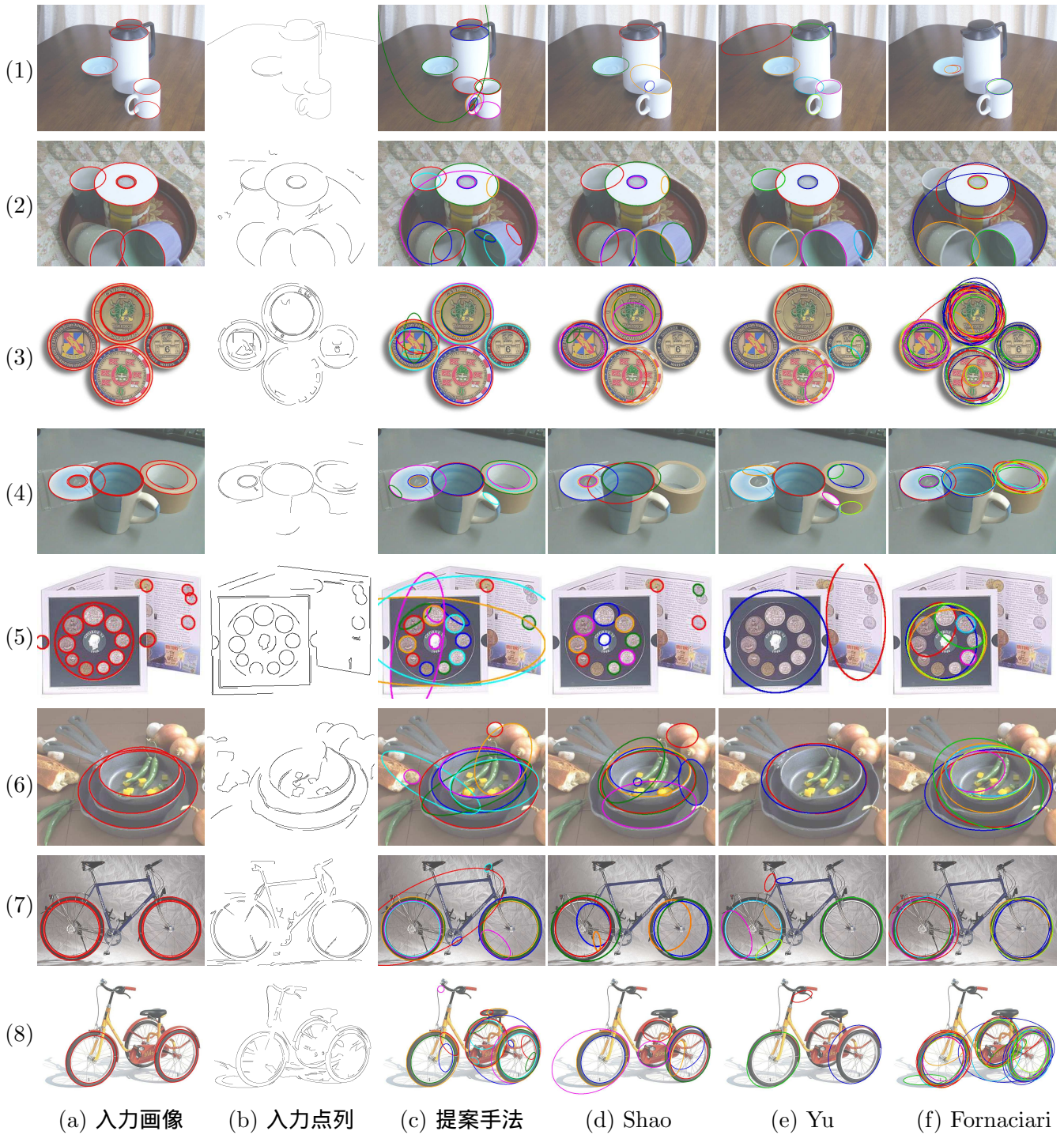


図 4.4: 楕円の検出結果の比較.

4.8 まとめ

本章では楕円当てはめに基づく楕円検出手法の高性能化の枠組みとして、連結したエッジ点列から楕円弧を選択する手法と、同一楕円上のエッジ点列を統合のための組み合わせ削減手法を合わせた新たな楕円検出法を提案した。その中で、従来手法の問題点を解決し、楕円弧抽出性能を向上させるとともに、楕円弧の統合効率の向上を実現し、高性能な楕円検出を可能とした。そして、シ

表 4.3: 検出できた楕円のみを用いた評価値平均の比較.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
提案手法	0.979	0.772	0.931	0.986	0.837	0.970	0.857	0.964
Shao	0.614	0.812	0.862	0.871	0.940	0.714	0.765	0.970
Yu	0.886	0.946	0.627	0.559	0	0.827	0.468	0.826
Fornaciari	0.919	0.876	0.824	0.867	0.761	0.821	0.914	0.886

表 4.4: 楕円検出時間の比較 (単位は秒)

	提案手法 (効率化処理なし)	Shao	Yu	Fornaciari
(1)	0.214(1.016)	336.560	0.292	0.022
(2)	0.236(1.844)	19.304	0.292	0.013
(3)	1.493(10.418)	22.907	0.624	0.111
(4)	0.193(0.981)	16.468	0.156	0.035
(5)	0.868(10.415)	86.830	0.136	0.011
(6)	0.406(5.092)	42.337	0.268	0.010
(7)	0.387(22.786)	174.236	2.228	0.035
(8)	1.342(26.163)	43.622	2.532	0.094

ミュレーションデータや実画像による実験を行い, 提案手法の有効性を示した.

提案手法によって画像中の楕円物体でない物体上にも楕円を検出した結果が得られたが, 本手法は画像中より楕円をまれ無く検出するための基礎手法であるため, この問題を考慮しない. 本手法によって画像中の楕円を可能な限り検出し, 応用手法を考慮したときに対象とする楕円物体特有の特徴を用いて検出した楕円の中から必要な楕円を選別していくべきである.

提案手法の計算時間については, 検出した楕円弧の数が多いほど高性能化による計算時間の削減効果が得られる. ただし, 統合対象の点列数に応じて計算時間が長くなる問題は完全には解決できていないため, 今後の検討課題である.

第5章 移動ロボットへの応用

本章では、これまで研究した楕円検出法を、移動ロボットの壁認識システムへ応用した手法について述べる。また、前章で提案した楕円検出の効率化処理を移動ロボットのリアルタイム処理に適用した場合の動作を確認し、提案手法の有用性を示す。

5.1 ロボットの環境認識

自律移動ロボットにおいて、環境中の障害物の認識は重要な課題の一つである。ロボットに環境を認識させる方法として、レーザレンジファインダ (LRF) のようなセンサーを用いた方法 [36] と、画像処理によって推定する方法 [2, 31, 34] がある。

画像処理による環境認識をロボットを実時間で動作させる場合には、手法の計算時間が大きな課題となり、簡易で高速な手法が求められる。これまでの、画像処理による環境の推定には、ステレオカメラを用いた平面投影による地面や障害物を認識する方法 [31, 34] や、移動中に撮影した連続画像を用いて環境を認識する方法 [2] が提案されている。これらに撮影された複数枚の画像を用いることで環境認識を可能としている。これに対し本研究では、新たなアプローチとして、プロジェクタで楕円を投影し、その楕円の形状の変化を観測することで、一枚の画像から壁と床の境界を推定する手法を提案する。

5.2 提案手法

提案手法では、これまでに研究した楕円検出法を用いて、壁認識手法の新たなアプローチを提案する。本手法では、プロジェクタで投影した円錐状の光は、床に投影した場合と壁に投影した場合とは異なる楕円の形状で観測される。この現象より、壁と床の境界線上に一つの楕円を投影した場合に、壁上の楕円弧と床上の楕円弧にそれぞれ楕円を当てはめ、二つの楕円の交点より壁と床の境界を推定する。

提案手法は、プロジェクタを用いるため、実環境の表面上に直接描画でき、環境を汚さず、シーンによって当てはめやすい楕円に動的に操作することが可能である。提案手法で行う画像処理は、高速な楕円当てはめ処理と簡易な交点検出による壁との境界認識処理であるため、高速な動作を実現できる。

5.3 壁認識システムの概要

本研究では、図 5.1 (a) に示すような、プロジェクタとカメラを装備した移動ロボットより得られる画像を入力とする。プロジェクタより円錐状の光を床に向けて投影し、投影した楕円を撮影する。図 5.1 (1) の状態から図 5.1 (2) のように壁にロボットが近づいた場合、投影した楕円の一部は床と壁でそれぞれ異なる形状の楕円として観測される。提案手法では、この特性を用いて、抽出された楕円弧上のエッジ点列にそれぞれ楕円を当てはめ、その交点より壁と床の境界を推定す

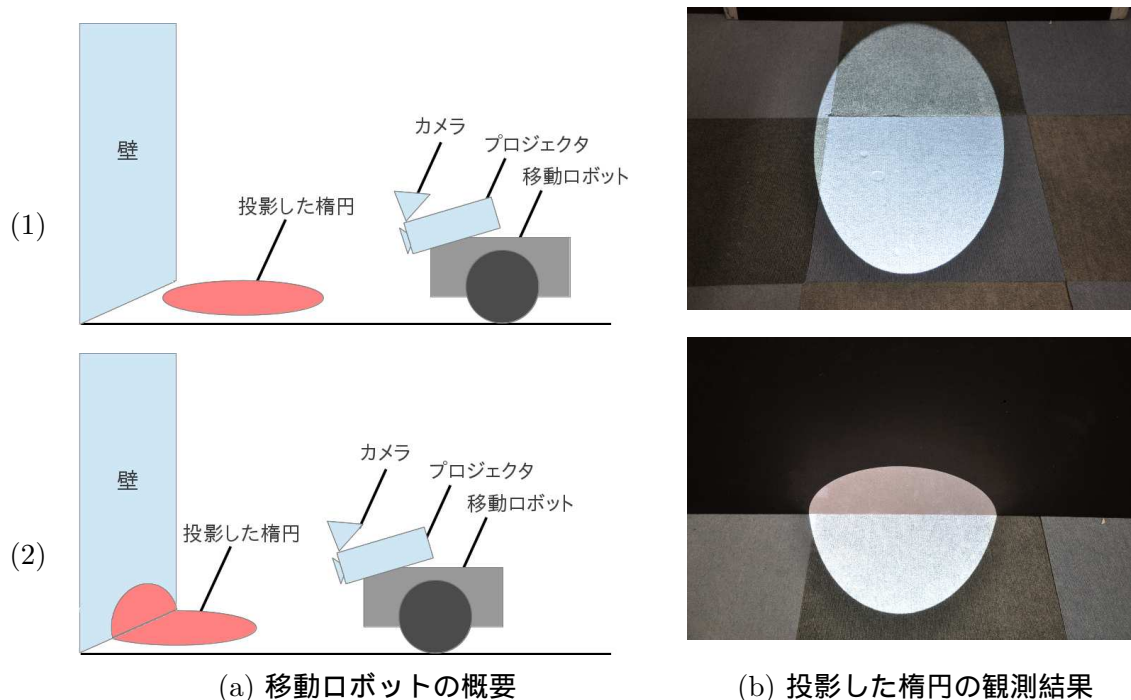


図 5.1: 提案手法の概要 . (1) 床に楕円を投影した状態. (2) 壁と床の境界へ楕円を投影した状態.

る. 本研究では床に投影された楕円から抽出された楕円弧点列に当てはめた楕円を「初期楕円」と呼ぶ. 壁と床の境界に投影した楕円から抽出される点列から, 初期楕円上の点列を除去した楕円弧点列に 2 つ目の楕円を当てはめる. 筆者らの提案する楕円当てはめでは楕円上の点列に楕円以外の点列が連結して含まれる場合でも, 楕円弧点列のみを選択して楕円当てはめが可能である. 提案手法の流れを以下にまとめる.

初期手順

1. 床に投影された楕円を撮影する.
2. 画像より得られた点列に楕円を当てはめ, これを初期楕円とする.

壁と床の境界推定

1. 投影された楕円を撮影する.
2. 画像より得られた点列を複数の楕円弧に分割し, その中から最も点数の多い楕円弧点列を選択する.
3. 選択した楕円弧点列より初期楕円上の点列を除去する.
4. 残った点列に楕円を当てはめる.
5. 初期楕円と当てはめた楕円から, 壁と床の境界線を推定する.

次節で詳しい交点の選択と境界の検出方法を説明する.

5.4 楕円の交点を用いた壁と床の境界の検出

プロジェクタで投影する1種類の楕円は壁面と床面で2種類の形状で観測される。楕円を壁と床の境界に投影した場合に、床上に投影した楕円と壁面に投影した楕円が壁と床の境界線上の2点で接するとすると、この2点を通る直線を壁と床の境界線として求める。

次に、境界の推定方法について楕円の定義、2つの楕円の交差線の導出、境界線の検出方法についてそれぞれ述べる。

5.4.1 楕円の定義

求める楕円の式は

$$Ax^2 + 2Bxy + Cy^2 + 2f_0(Dx + Ey) + f_0^2F = 0, \quad (5.1)$$

で表され、楕円以外に放物線、双曲線と退化した2直線を示す。ここで、 f_0 はスケール定数を示す。ここで、3次元のベクトル x と 3×3 の行列 Q を次のように定義する。

$$x = \begin{pmatrix} x \\ y \\ f_0 \end{pmatrix}, \quad Q = \begin{pmatrix} A & B & D \\ B & C & E \\ D & E & F \end{pmatrix} \quad (5.2)$$

式 (5.2) を用いると、式 (5.1) は次のように表すことができる。

$$(x, Qx) = 0 \quad (5.3)$$

ここで (a, b) はベクトル a とベクトル b の内積を表す。

5.4.2 2つの楕円の境界線

壁面と床面から得られる二つの楕円 $(x, Q_1x) = 0$ と $(x, Q_2x) = 0$ が得られた場合、

$$(x, (Q_1 + \lambda Q_2)x) = 0 \quad (5.4)$$

は2つの楕円の全ての交点を通る2次曲線またはその退化した直線を示す。ここで λ は任意の定数を示す。 λ の値を

$$|Q'| = 0, \quad Q' = Q_1 + \lambda Q_2 \quad (5.5)$$

のように求めると式 (5.4) は二つの実線か仮想線の直線に退化する。2つの楕円が4点で交わる場合に2つの直線は

$$A'x + (B' \pm \sqrt{B'^2 - A'C'})y + \left(D' - \frac{B'D' - A'E'}{\sqrt{B'^2 - A'C'}} \right) f_0 = 0 \quad (5.6)$$

で表される。ここで、 A', B', \dots は式 (5.2) の Q' の要素を示す。2つの楕円が2点で接している場合は2点の接点より直線が求められ、次の式で表すことができる。

$$A'x + B'y + D'f_0 = 0. \quad (5.7)$$

5.4.3 壁と床の境界検出

本研究では、壁と床の境界に投影された壁上の楕円と床上の楕円は、境界線上の2点で接するとして、その2点を通る直線を壁と床の境界線と判定する。しかし、実際に画像より楕円検出法によって得られる2つの楕円は必ずしも2点で接するわけではない。そこで、本研究では、式(5.5)中の λ を次の2つのルールによって定義する。

1. 式(5.5)が3つの実数解を持つ場合、2つの近い解の中間値を λ とする。
2. 式(5.5)が1つの実数解と2つの複素共役解を持つ場合、複素共役解の実数部を λ とする。

提案手法は、前述の条件で求めた λ を用いて Q' を計算し、式(5.7)より壁と床の境界線を導出できる。

5.5 実験

本実験では、提案手法による壁と床の境界検出に、筆者が提案したアウトライアを考慮した楕円限定当てはめを用いた場合の境界検出の結果を求め、提案手法の有効性を確認する。次に、移動ロボットを提案手法によってリアルタイムで制御した場合に、前章で述べた効率化処理を施した場合とそうでない場合の移動ロボットの動作の比較を示し、本研究の成果を示す。動作確認では、関らのロボット制御方法[37]により行う。提案した境界推定手法を用いて、移動ロボットは壁と床の境界が機体と一定距離以下の位置に来た場合に、機体を回転させ直進を繰り返す。この関らの距離推定と回転動作については、付録の7.4節にて詳細を説明する。

5.5.1 壁と床の境界検出結果

壁と床の境界に楕円を投影し、撮影した場合について、壁と床の境界推定結果を図5.2に示す。図5.2(a)は入力画像を示し、図5.2(1)で示す入力画像はシンプルな画像であるが、図5.2(2)の入力画像では境界付近に物体を置き、入力点列中に求めたい楕円以外の点列が含まれるような画像を用いることで、アウトライアを含んだ入力であっても境界を検出できることを確認する。図5.2(b)は入力として選択した点列を示す。

提案手法では、事前に床のみに楕円を投影した場合の楕円形状を求めておき、床上の楕円上の点列を図5.2(b)から除去する。床上の楕円弧点列を除去した、図5.2(c)に示される点列に楕円を当てはめ、境界を推定する。

図5.2(d)は実験結果を示しており、事前に求めた初期楕円、青い楕円が図5.2(c)の点列に当てはめた楕円を示しており、筆者の提案した楕円当てはめ法によりアウトライアを含んだ点列に対しても正しく楕円弧に楕円が当てはめられていることが確認できる。

図5.2(1)-(d)の二つの楕円のパラメータを用いて、式(5.4)より求めた λ は、 -0.331839 、 -0.337805 、 -3.15419 である。全て実数であるため、値の近い -0.331839 と -0.337805 の平均値を用いて式(5.7)の直線を求めた。図5.2(2)-(d)では、 λ は、 -0.285334 、 -0.296089 、 -2.35997 の3つの実数であるため、値の近い -0.285334 と -0.296089 の平均値を用いて式(5.7)の直線を求めた。図5.2(d)中の緑の線が求めた直線を示している。結果より、およそ正しく境界を求めることが出来たと言える。

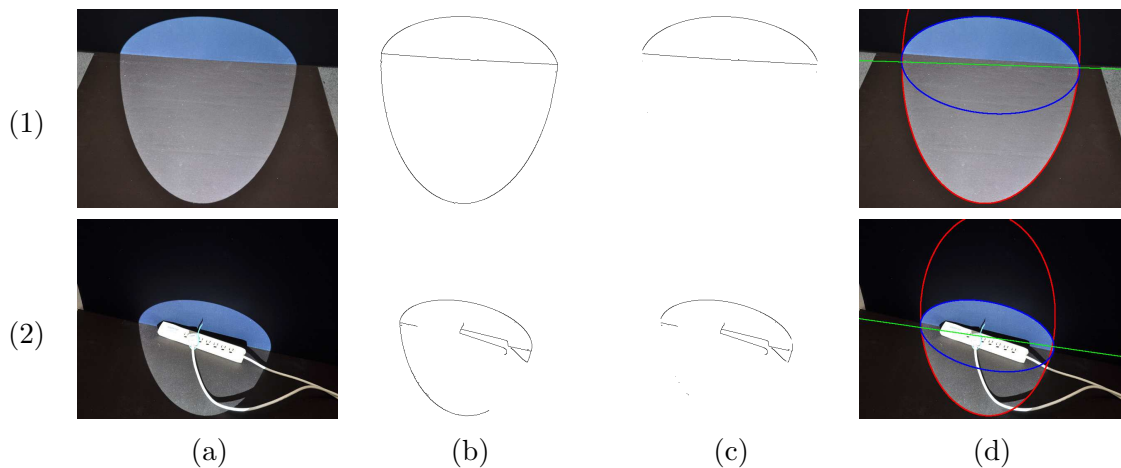


図 5.2: 壁と床の境界検出結果. (a) 入力画像. (b) 入力点列. (c) 初期楕円上の点列を除いた入力点列. (d) 当てはめた楕円と推定した境界線. 赤:初期楕円, 青:(c) の点列へ当てはめた楕円, 緑:推定した境界線.

5.5.2 移動ロボットの動作

実際に移動ロボットへ本システムを実装し, 本研究で提案した効率化を施した楕円検出法と菅谷らの手法 [40] を用いた場合の動作速度を比較し, リアルタイム処理における提案手法の有効性を確認した.

本実験では, 移動ロボットとして Yujin Robotics 社の Turtle Bot2 を採用し, Web カメラとプロジェクタ, ノート PC を搭載した. Web カメラは, 床から高さ 320mm に設置し, プロジェクタは高さ 200mm の位置に設置する. また, システムは OS:ubuntu12.04, CPU:Intel Core i5-5200U(2.20GHz, 3MB), メモリ:4GB の環境で動作させる. web カメラはピクセルサイズ $37.5\mu\text{m}/\text{pixel}$ のものを用い, 焦点距離 30mm, $640 \times 480\text{pixel}$ のサイズで画像を取得した.

図 5.3 に動作を比較した状況を示す. 図 5.3 (a) 入力画像, 図 5.3 (b) に入力画像より取得した入力点列, 図 5.3 (c) に初期楕円を赤, 当てはめた楕円を青, 推定した境界線を緑で示す. 図の状況より境界線を 10 回求める処理より平均処理速度を求めた結果, 提案手法を用いた境界推定システムは 0.954fps, 菅谷らの手法を用いた境界推定システムは 0.726fps で動作し提案手法の方が高速に動作することを確認した.

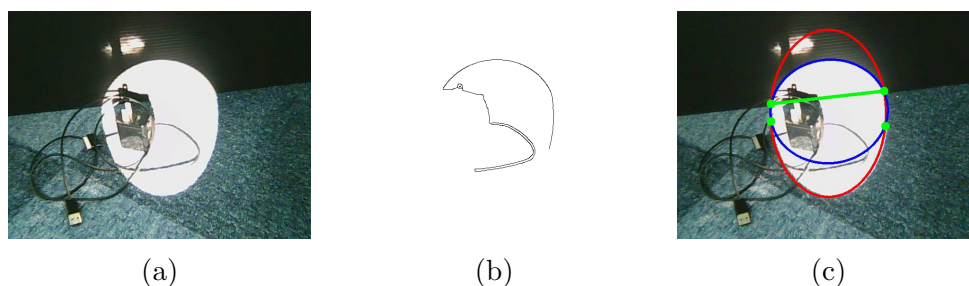


図 5.3: 動作速度の確認画像. (a) 入力画像. (b) 入力点列. (c) 当てはめた楕円と推定した境界線. 赤:初期楕円, 青:当てはめた楕円, 緑線:推定した境界線.

実際のロボットの走行状況を図 5.4 に示す. 図 5.4 (1) に実際の移動ロボットの外観を示す. 図

5.4 (2) にロボットが観測し、楕円を当てはめた画像を示す。図 5.4 (a), (b), (c) はそれぞれ、楕円が床のみに投影され直進している状態、楕円が壁と床に投影され、境界を推定した状態、境界までの距離が一定値以下になり、回転をした状態を示す。

図 5.4 より、提案した壁と床の境界線の推定手法は正しく動作していることが確認した。

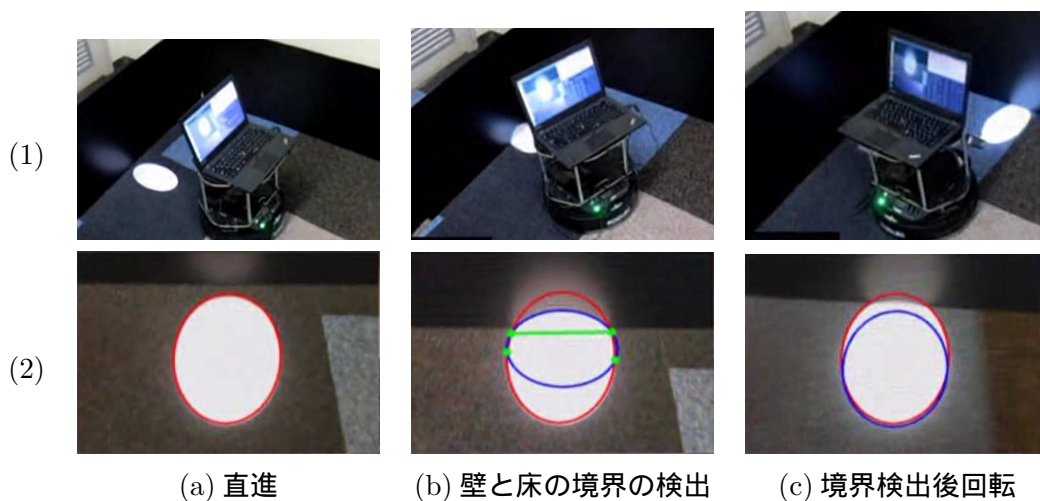


図 5.4: ロボットの動作実験. (1) ロボットの外観, (2) 観測した楕円と境界線. 赤:初期楕円, 青:当てはめた楕円, 緑:推定した境界線.

5.6 まとめ

本章では、これまで研究した楕円検出法を移動ロボットのための壁認識システムへ応用した新たな手法を提案した。本手法では、移動ロボットに装備したプロジェクタより円錐状の光を投影し、画像より投影された楕円を検出する。投影された楕円は壁と床の境界上では2つの形状の楕円として観測され、本手法ではこの2つの楕円の境界を床と壁の境界として推定する。実験により、提案手法は正しく壁と床の境界を推定できることが確認し、アウトライアを含んだ入力に対しても有効であることが確認した。また、本システムを用いて前章で提案した楕円検出の効率化処理を適用して動作の高速化を確認し、本研究の有用性を示した。楕円検出手法の高速化は実現したが、それ以外の処理によって実時間で有効とは言えないフレームレートになってしまっている。今後の課題として、楕円検出による境界推定システムを運用する場合は、壁検出システム全体の動作速度の高速化が必要である。

第6章 まとめ

本研究では、楕円検出法の高性能化を検討した。まず、一つのノイズを含んだ楕円弧点列に楕円を高精度に当てはめる楕円限定当てはめ法を提案した。次に、楕円限定当てはめの発展として、楕円弧以外の点列を含んだ入力点列から楕円弧点列を効率的に抽出する方法について提案した。この楕円弧選択法を用いて、入力画像中の楕円弧を抽出・統合することで高い再現率を持った高性能な楕円検出法を提案した。最後に、提案した楕円検出法を移動ロボットの壁認識システムへの応用し、リアルタイムで動作するシステムにおいて、提案手法の有効性を示した。次に、提案した各手法について、詳しくまとめる。

まず本研究では、金谷らの超精度くりこみ法に点の分布を考慮したランダムサンプリングを組み合わせた新しい楕円限定当てはめ法を提案した。従来の楕円当てはめ法の中でも優れた楕円当てはめ法だと言われる超精度くりこみ法であるが、ノイズが大きいと双曲線が当てはまることがある。提案手法はそのような場合に入力点列のランダムサンプリングによって点列に最も近い楕円を当てはめる方法である。実験により、入力点列に加わるノイズが大きくなり、超精度くりこみ法が双曲線を当てはめる割合が増加した場合に提案手法により精度が向上することを確認した。

これまでに楕円のみを当てはめる方法を提案したのは Fitzgibbon ら [5] および Szpak ら [41] であるが、シミュレーション実験によって提案手法はそれらより精度が高いことを確認した。また、提案手法は、より細かい楕円上の点列に対して精度が高い結果が得られる傾向があった。そして、Fitzgibbon らの方法では扁平な楕円が当てはまりやすく、Szpak らの方法では双曲線に近い楕円が当てはまりやすいのに対して、提案方法はその中間で、より真の形状に近い楕円が当てはまることを観察した。本研究の提案手法は非常にシンプルな方法であるが最新の手法である Szpak らの方法と同等かそれ以上の精度であり、楕円限定当てはめにおいて有効な方法であると言える。

また、ランダムサンプリングを用いた楕円当てはめでは、従来 RANSAC が容易な方法として用いられるが、サンプリングする点の分布を考慮しないためサンプリング回数が多くなる。提案手法のランダムサンプリング法では、サンプリングする点の位置が偏らないように入力点の分布を考慮してサンプリングを行い楕円を当てはめることで、解の収束回数を削減した。提案手法のサンプリング方法について、正規乱数を用いたサンプリング法と、入力点列を5分割したサンプリング法と解の収束回数を比較し、提案手法が最も収束回数が少ないことを確認した。これは、提案手法には、サンプリングした点同士の距離が近ければサンプリングし直す処理を加えているため、正規乱数を用いた場合よりも入力点列から点の位置に偏りが少ないサンプリング結果が得られ易いためだと考えられる。これより今後は、サンプリングした点同士の距離を考慮して正規乱数を用いたサンプリングを行うことにより、提案手法に近い収束回数が実現できると考えられる。

超精度くりこみ法で双曲線が当てはまる場合の入力点列に対し、RANSAC, PROSAC, 提案手法の解をそれぞれ比較し、提案手法は高精度かつ PROSAC と同等またはそれ以上に収束回数であることを確認した。また、提案手法の収束回数の削減効果は、曲率が低い入力点列で楕円が細くなるほど効果が上がることを確認した。

前述した提案手法は入力点列は楕円上の点列を入力として想定しており、アウトライア (楕円上の点とは関係無い点) が含まれることを考慮していない。そこで、我々は、楕円限定当てはめによって当てはめた楕円と入力点列の交点で点列を分割し、当てはめ誤差の点列間での変化の傾向

をもとに、楕円当てはめに使用する楕円弧点列を効率的に選択する手法を提案した。この手法では、入力点列を当てはめた楕円との交点によって自動的に部分弧に分割するため、経験的なしきい値などを必要せず、選択した楕円弧に対して当てはめた楕円を用いた入力点列の再分割を繰り返すことで、楕円弧上の点列選択を実現した。

本研究では、楕円弧選択のための評価値として、部分弧ごとの楕円当てはめの誤差グラフの変化をもとにした楕円弧選択法を提案し、その計算効率と有効性について示した。しかし、誤差グラフの変化のみを評価するだけでは楕円弧を正しく選択出来ない例外が存在することも確認した。

そこで、本研究では部分弧ごとへ楕円を当てはめ、それぞれの楕円の当てはめ誤差平均より楕円弧点列を選択する手法を提案し、問題を解決した。実験により正しく楕円弧を選択できることを示した。

しかし、入力画像中の楕円上の全ての点列が一つに連結した入力点列として得られる状況は少なく、楕円弧点列が短い場合は、楕円の検出精度は低下する。この問題を解決するため、同一楕円上の楕円弧点列を統合して楕円を当てはめる菅谷らの手法を用いることで、より正しい形状の楕円を当てはめる方法を提案した。菅谷らの手法は、入力楕円弧の数が多いと楕円弧の統合判定の組合せ数が多くなり、計算コストが大きくなる問題があった。本研究はこの問題を解決するための効率化処理として、まず楕円弧点列を当てはめた楕円の4象限の領域に分類する。分類した点列は、3領域に点が存在する点列と長軸上の2領域に点が存在する点列は楕円当てはめに十分な点列が得られたとして組合せから除外する。次に、同一領域に存在する楕円弧と対象の楕円弧に当てはめた楕円を拡大した領域外の楕円弧を組合せから除外する。上記の4つの条件により統合判定のために作成する楕円弧の組合せを大きく削減し、処理を効率化した。実験にて既存手法と比較し、高性能な楕円検出であることを示した。

最後に、提案した楕円検出法を移動ロボットへ応用した壁と床の境界認識システムを提案した。この手法では、プロジェクタで投影した楕円が壁面と床面で異なる形状で観測される現象を用いて、壁面の投影光のエッジに当てはめた楕円と床面に投影した楕円の交点より、壁と床の境界線を推定する新たなアプローチを示した。実験により、アウトライアを含んだ実画像に対しても正しく境界線を推定できることを確認し、実際に提案した楕円検出法の効率化によってシステムが高速化されることも示した。

本研究では、入力画像より得られるアウトライアを含んだ連結点列を入力とした、高性能な楕円検出手法を提案し、実験によりその有効性を示した。しかし、本研究の楕円の検出能力は、入力画像より楕円物体上のエッジ点列がどの程度抽出できるかに依存するが、本論文ではその点について考慮しておらず、今後の検討課題といえる。また、入力画像からエッジ点列の抽出する場合に、楕円の交点を用いた部分弧の分割を実現するため、連結点列を端点から整列して取得しなければならないため、エッジ点列の抽出処理と合わせて、連結点列の効率的な抽出法の検討も必要であると考えられる。

謝辞

本研究の機会を与えて下さり，ご指導，ご教示を頂きました菅谷保之准教授に深く感謝致します．

また，本研究を進めるにあたり多岐にわたるご助言をいただきました金谷健一教授，審査に携わっていただきました三浦純教授ならびに栗山繁教授に深く感謝致します．

最後に，研究データの作成やロボット制御の実験等で，大きく協力頂いた豊橋技術科学大学 画像情報メディア研究室の関正貴くんをはじめ，さまざまなことで切磋琢磨し合った研究室の皆様方に深く感謝致します．

発表論文

査読付き学術論文

- 益崎智成, 関正貴, 菅谷保之, “ 効率的な楕円弧選択と幾何学的 MDL による楕円弧統合を組み合わせた楕円検出 ”, 情報処理学会論文誌 59 巻 3 号, pp. 951-960, 2018.
- 益崎智成, 菅谷保之, “ Effective elliptic arc selection from connected edge points ”, IPSJ Transactions on Computer Vision and Applications, Vol.8, No.6, 11pages, 2016.

国際学会口頭発表

- Tomonari Masuzaki, Yasuyuki Sugaya, Kenichi Kanatani, ”Floor-wall boundary estimation by ellipse fitting”, International Conference on Cybernetics and Intelligent Systems(CIS) and the Robotics, Automation and Mechatronics(RAM), pp. 30-35, Apsara Palace Resort (Angkor Wat, Cambodia), 2015.
- Tomonari Masuzaki, Yasuyuki Sugaya, Kenichi Kanatani, ”High Accuracy Ellipse-Specific Fitting.”, Proceedings of the 6th Pacific-Rim Symposium on Image and Video Technology(PSIVT 2013), University of Guanajuato (Guanajuato, Mexico), pp. 314-324, 2013.

国際学会ポスター

- Tomonari Masuzaki, Yasuyuki Sugaya, ”Connectivity-based error evaluation for ellipse fitting”, IAPR International Conference on Machine Vision Applications(MVA 2015), pp. 118-121, 日本科学未来館 (東京), 2015.

その他 (国内発表)

- Tomonari Masuzaki, Masataka Seki, Yasuyuki Sugaya, ”Ellipse detection by effective elliptic arc extraction and integration using geometric MDL”, 画像の認識理解シンポジウム MIRU2017, PS1-32, 広島国際会議場 (広島), 2017.
- 益崎智成, 菅谷保之, ”効率的な楕円弧点列の選別と同一楕円上点列の統合”, 電子情報通信学科研究報告. パターン認識・メディア理解 PRMU2016-53, pp. 105-110, NTT 武蔵野研究開発センター (東京都), 2016.
- 益崎智成, 菅谷保之, ”誤差グラフの形状を考慮した点列選択による楕円当てはめ”, 情報処理学会研究報告書. CVIM[コンピュータビジョンとイメージメディア], 2015-CVIM-199(27), pp. 1-8, 神戸大学 六甲台第 2 キャンパス (兵庫県), 2015.

- 益智成, 菅谷保之, ”点列の連結性を考慮したアウトラシアを含むデータへの楕円当てはめ”, 情報処理学会研究報告. CVIM[コンピュータビジョンとイメージメディア], 2014-CVIM-194(6), pp. 1-8, 早稲田大学 西早稲田キャンパス (東京都), 2014.
- Tomonari Masuzaki, Yasuyuki Sugaya, Kenichi Kanatani, ”High Accuracy Ellipse-Specific Fitting.”, 画像の認識理解シンポジウム MIRU2013, OS4-4, 国立情報学研究所 (東京都), 2013.
- 益崎智成, 菅谷保之, 金谷健一, ”精度の高い楕円限定当てはめ.” 情報処理学会研究報告. CVIM[コンピュータビジョンとイメージメディア], 2013-CVIM-186(14), pp. 1-8, 大阪大学 吹田キャンパス (大阪府), 2013.

参考文献

- [1] 有馬利洋, 菅谷保之, エッジ点列の分割とモデル選択を用いた統合による楕円検出, 情報処理学会研究報告, 2009-CVIM-166-5 (2009-3), 33–40.
- [2] M. Barth, H. Ishiguro and S. Tsuji, Determining Robot Egomotion from Motion Parallax Observed by an Active Camera, Proc. 12th Int. Joint Conf. Artificial Intell., pp. 1247–1253, 1991.
- [3] W. Chojnacki, M. J. Brooks, A. van den Hengel and D. Gawley, On the fitting of surfaces to data with covariances, *IEEE Trans. Patt. Anal. Mach. Intell.*, **22**-11 (2000), 1294–1303.
- [4] M. Fornaciari, A. Prati, R. Cucchiara, A fast and effective ellipse detector for embedded vision applications, *Pattern Recognition* **47**-11, pp. 3683–3708, 2014.
- [5] A. Fitzgibbon, M. Pilu, and R. B. Fisher, Direct least squares fitting of ellipses, *IEEE Trans. Patt. Anal. Mach. Intell.*, **21**-5 (1999-5), 476–480.
- [6] G. Griffin, A. Holub, and P. Perona, Caltech-256 object category dataset. Technical Report UCB/CSD-04-1366, California Institute of Technology, 2007.
- [7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, Cambridge, U.K., 2004.
- [8] 今泉一崇, 小林大起, 菅谷保之, 複数の円形マーカーを用いた複合現実感システムの構築, 第16回画像センシングシンポジウム (SSII10) 講演論文集, pp. IS219, 2010.
- [9] 岩元祐輝, プラサンナ・ランガラヤン, 金谷 健一, 楕円当てはめの超精度最小二乗法, 情報処理学会研究報告, 2009-CVIM-168-14 (2009-8/9), 1–8.
- [10] O. Chum and J. Matas, “ Matching PROSAC - Progressive Sample Consensus, ”*Proc. of CVPR, Vol.1. 2005*, pp.220 – 226 .
- [11] K. Kanatani, *Geometric Computation for Machine Vision*, Oxford University Press, Oxford, U.K., 1993.
- [12] K. Kanatani, Renormalization for unbiased estimation, *Pro. 4th Int. Conf. Comput. Vis. (ICCV'93)*, May 1993, Berlin, Germany, pp. 599–606.
- [13] 金谷健一, コンピュータビジョンのためのくりこみ法, 情報処理学会論文誌, **35**-2 (1994-2), 201–209.
- [14] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice* Elsevier, Amsterdam, the Netherlands, 1996; reprinted, Dover, York, NY, U.S.A., 2005.

- [15] 金谷健一, 「形状CADと図形の数学」, 共立出版, 1998.
- [16] 金谷健一, 幾何学的当てはめにおけるモデル選択, 電子情報通信学科論文誌 A, J84-A-11, pp. 1385–1393, 2001.
- [17] 金谷健一, 「これなら分かる最適化数学—基礎原理から計算手法まで—」, 共立出版, 2005.
- [18] K. Kanatani, Ellipse fitting with hyperaccuracy, *IEICE Trans. Inf. & Syst.*, **E89-D-10** (2006-10), 2653–2660.
- [19] K. Kanatani, Statistical optimization for geometric fitting: Theoretical accuracy analysis and high order error analysis, *Int. J. Comput. Vis.*, **80-2** (2008-11), 167–188.
- [20] 金谷健一, アリ・アルシャラドカー, ニコライ・チェルノフ, 菅谷保之, 超精度くりこみ法, 情報処理学会研究報告, 2012-CVIM-180-23 (2012-1), 1–8.
- [21] K. Kanatani, A. Al-Sharadqah, N. Chernov, and Y. Sugaya, Renormalization Returns: Hyper-renormalization and Its Applications, *Proc. 12th Euro. Conf. Comput. Vis.*, October 2012, Florence, Italy, Vol. 3, pp. 385–398.
- [22] K. Kanatani and P. Rangarajan, Hyper least squares fitting of circles and ellipses, *Comput. Stat. Data Anal.*, **55-6** (2011-6), 2197–2208.
- [23] K. Kanatani, P. Rangarajan, Y. Sugaya and H. Niitsuma, HyperLS and its applications, *IPSJ Trans. Comput. Vis. Appl.*, **3** (2011-10), 80–94.
- [24] K. Kanatani and Y. Sugaya, Performance evaluation of iterative geometric fitting algorithms, *Comput. Stat. Data Anal.*, **52-2** (2007-10), 1208–1222.
- [25] 金谷健一, 菅谷保之, 幾何学的当てはめの厳密な最尤推定の統一的計算法, 情報処理学会論文誌: CVIM, **2-1** (2009-3), 53–62.
- [26] K. Kanatani and Y. Sugaya, Unified computation of strict maximum likelihood for geometric fitting, *J. Math. Imaging Vis.*, **38-1** (2010-9), 1–13.
- [27] Y. Leedan and P. Meer, Heteroscedastic regression in computer vision: Problems with bilinear constraint, *Int. J. Comput. Vis.*, **37-2** (2000-6), 127–150.
- [28] 益崎智成, 点の分布を考慮したランダムサンプリングによる楕円限定当てはめ, 平成 25 年度 豊橋技術科学大学 修士論文, 2013.
- [29] J. Matei and P. Meer, Estimation of nonlinear errors-in-variables models for computer vision applications, *IEEE Trans. Patt. Anal. Mach. Intell.*, **28-10** (2006-10), 1537–1552.
- [30] 森本正志, 興水大和, Hough 変換の諸課題と新しいパターン計測 基礎編, 計測と制御 **35**(11), pp. 869–877, 1996.
- [31] D. Murray, and James J. Little, Using real-time stereo vision for mobile robot navigation, *Autonomous Robots* **8-2**, pp. 161–171, 2000.
- [32] 中川裕介, 金谷健一, 菅谷保之, 高ノイズレベルにおける最尤楕円当てはめ, 情報処理学会研究報告, 2008-CVIM-162-10 (2008-3), 53–60.

- [33] 岡部光生, 金谷健一, 太田直哉, 楢円成長法による円形物体の自動検出, 電子情報通信学会論文誌 D-II, **J85-D-II-12** (2002-12), 1823–1831.
- [34] K. Onoguchi, N. Takeda and M. Watanabe, Planar projection stereopsis method for road extraction, *IEICE TRANSACTIONS on Information and System* 81.9, pp. 1006–1018, 1998.
- [35] D. K. Prasad, M. K. H. Leung, and S.-Y. Cho, Edge curvature and convexity based ellipse detection method, *Pattern Recognition*, Vol. 45, Issue 9, pp. 3204–3221, 2012.
- [36] E. Prassler and J. Scholz, Tracking Multiple Moving Objects for Real-Time Navigation, *Autonomous Robots*, **8-2**, pp. 105–116, 2000.
- [37] 関正貴, 菅谷保之, 楢円の投影像を用いた床と壁の境界検出によるロボットの走行制御, 第 22 回画像センシングシンポジウム, IS1-25, 2016.
- [38] M. Shao, Y. Ijiri, and K. Hattori, Grouped outlier removal for robust ellipse fitting, 14th IAPR International Conference on Machine Vision Applications(MVA2015), pp. 138–141, 2015.
- [39] 菅谷保之, 金谷健一, [講座] 画像の三次元理解のための最適化計算 [I]–[IV] 電子情報通信学会誌, **92-3**, 4, 6, 7 (2009-3, 4, 6, 7), 229–233, 301–306, 463–468, 573–578.
- [40] Y. Sugaya, Ellipse Detection by combining division and model selection based integration of edge points, 4th Pacific-Rim Symposium on Image and Video Technology(PSIVT2010), pp. 64–69, 2010.
- [41] Z. L. Szapak, W. Chojnacki, and A. van den Hengel, Guaranteed ellipse fitting with Sampson distance, *Proc. 12th Euro. Conf. Comput. Vis.*, October 2012, Florence, Italy, Vol. 5, pp. 87–100.
- [42] G. Taubin, Estimation of planar curves, surfaces, and non-planar space curves defined by implicit equations with applications to edge and range image segmentation, *IEEE Trans. Patt. Anal. Mach. Intell.*, **13-11** (1991-11), 1115–1138.
- [43] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM* **24.6**, (1981-6), 381–395.
- [44] 山田純平, 金谷健一, 超精度の楢円当てはめ, 情報処理学会研究報告, 2005-CVIM-151-15 (2005-11), 197–114.
- [45] 山田 純平, 金谷 健一, 菅谷 保之, 楢円当てはめの高精度計算法とその性能比較, 情報処理学会研究報告, 2006-CVIM-154-36 (2006-5), 339–346.
- [46] 横田健太, 村田和洋, 菅谷保之, 金谷健一, 楢円当てはめの精度比較: 最小二乗法から超精度くりこみ法まで, 情報処理学会研究報告, 2012-CVIM-180-26 (2012-1), 1–8.
- [47] Yu Qiao, S. H. Ong, Arc-based evaluation and detection of ellipses, *Journal of Pattern recognition*, **40-7**, pp. 1990–2003, 2007.

- [48] Y. Wang, Z. He, X. Liu, Z. Tang and L. Li, A fast and robust ellipse detector based on top-down least-square fitting, *Computers and Electrical Engineering* 40.4, pp. 1415–1428, 2014.

付録 A 既存の楕円当てはめ

現在知られている楕円当てはめ法について説明する。従来の楕円当てはめ方は、当てはめる楕円と点列との距離の二乗和 (再投影誤差) を最小になるよう計算する方法と、0 になるべき式 (A.1) の円錐曲線の式の二乗和 (代数的距離) を最小になるように計算する方法がある。また、再投影誤差や代数的距離を最小化するのではなく、誤差解析によって精度がなるべく高くなるように計算方式を定める方法がある。

$$Ax_\alpha^2 + 2Bx_\alpha y_\alpha + Cy_\alpha^2 + 2f_0(Dx_\alpha + Ey_\alpha) + f_0^2 F \approx 0 \quad (\text{A.1})$$

式 (A.1) を簡単化するため、6 次元ベクトル

$$\begin{aligned} \xi_\alpha &= (x_\alpha^2, 2x_\alpha y_\alpha, y_\alpha^2, 2f_0 x_\alpha, 2f_0 y_\alpha, f_0^2)^\top, \\ \theta &= (A, B, C, D, E, F)^\top \end{aligned} \quad (\text{A.2})$$

と定義し、ベクトル a, b の内積を (a, b) としたとき、式 (A.1) を、

$$(\xi_\alpha, \theta) \approx 0, \quad \alpha = 1, \dots, N \quad (\text{A.3})$$

と表す。

それぞれの楕円当てはめ法には次のような手法がある。ここで各手法の入力には、楕円上の入力点列 (x_α, y_α) , $\alpha = 1, \dots, N$ を式 (A.2) で定義した ξ_α とし、当てはめ結果の解を θ として説明する。

- 再投影誤差最小化に基づく方法
FNS 法 [3], HEIV 法 [29], 射影ガウスニュートン法 [24, 45]
- 代数適距離最小化に基づく方法
最小二乗法, Taubin 法 [42], 超精度最小二乗法 [9, 22, 23]
- 最小化によらない方法
重み反復法, くりこみ法 [12, 13], 超精度くりこみ法 [20, 21]

これらの手法の中では、超精度くりこみ法が最も高精度に楕円が当てはめられることが確認されている。提案手法も主にこの方法を用いる。

本節ではこれらの楕円当てはめ法について説明し、楕円当てはめ法の問題点を挙げる。

A.1 FNS 法

FNS 法とは、 Sampson 誤差を最小化するための固有値問題の反復解法である。式 (2.9) を最小化するには、 θ で微分した式

$$\nabla_{\boldsymbol{\theta}} J = \frac{1}{N} \sum_{\alpha=1}^N \frac{(\boldsymbol{\theta}, \boldsymbol{\xi}_{\alpha}) \boldsymbol{\xi}_{\alpha}}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_{\alpha}] \boldsymbol{\theta})} - \frac{1}{N} \sum_{\alpha=1}^N \frac{(\boldsymbol{\theta}, \boldsymbol{\xi}_{\alpha})^2 V_0[\boldsymbol{\xi}_{\alpha}] \boldsymbol{\theta}}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_{\alpha}] \boldsymbol{\theta})^2} \quad (\text{A.4})$$

を 0 と置いた式を解く．このとき各項を行列 M , L で次のように置く．

$$\begin{aligned} M &= \frac{1}{N} \sum_{\alpha=1}^N W_{\alpha} \boldsymbol{\xi}_{\alpha} \boldsymbol{\xi}_{\alpha}^{\top}, \\ L &= \frac{1}{N} \sum_{\alpha=1}^N W_{\alpha}^2 (\boldsymbol{\theta}, \boldsymbol{\xi}_{\alpha})^2 V_0[\boldsymbol{\xi}_{\alpha}] \\ W_{\alpha} &= \frac{1}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_{\alpha}] \boldsymbol{\theta})} \end{aligned} \quad (\text{A.5})$$

FNS 法を用いた楕円当てはめの手順を次に示す．

1. $W_{\alpha} = 1$, $\alpha = 1, \dots, N$, $\boldsymbol{\theta}_0 = \mathbf{0}$ と初期値を設定する．
2. 行列 M , L を計算する．
3. 固有値問題

$$(M - L)\boldsymbol{\theta} = \lambda \boldsymbol{\theta} \quad (\text{A.6})$$

を解き，最小固有値 λ に対する単位固有ベクトル $\boldsymbol{\theta}$ を計算する．

4. 符号を除いて $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$ なら $\boldsymbol{\theta}$ を返して終了する．そうでなければ次のように更新して手順 2 に戻る．

$$W_{\alpha} \leftarrow \frac{1}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_{\alpha}] \boldsymbol{\theta})}, \quad \boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta} \quad (\text{A.7})$$

このとき $(\boldsymbol{\theta}, \boldsymbol{\theta}_0) < 0$ であれば $\boldsymbol{\theta} \leftarrow -\boldsymbol{\theta}$ と符号を変える．

A.2 HEIV 法

HEIV 法も FNS 法と同様にサンプソン誤差を最小化するための固有値問題の反復解法であるが，式 (A.6) の行列 M , L を用いて次の一般固有値問題

$$M\boldsymbol{\theta} = \lambda L\boldsymbol{\theta} \quad (\text{A.8})$$

を解くことにより，FNS 法と同等の解が得られる．

A.3 射影ガウスニュートン法

射影ガウスニュートン法も FNS 法，HEIVE 法と同様にサンプソン誤差を最小化する反復解法である．

射影ガウスニュートン法を用いた楕円当てはめの手順を次に示す．

1. $\theta_0 = 0$ の初期値を与える．
2. 次の行列 M ， L を計算する．

$$\begin{aligned} M &= \sum_{\alpha=1}^N \frac{\xi_{\alpha} \xi_{\alpha}^{\top}}{(\theta_0, V_0[\xi_{\alpha}] \theta_0)}, \\ L &= \sum_{\alpha=1}^N \frac{(\theta_0, \xi_{\alpha})^2 V_0[\xi_{\alpha}]}{(\theta_0, V_0[\xi_{\alpha}] \theta_0)^2} \end{aligned} \quad (\text{A.9})$$

3. θ を次のように計算する．

$$\theta = N[\theta_0 - (P_{\theta} M P_{\theta})^{-1} (M - L) \theta_0] \quad (\text{A.10})$$

ここで， P_{θ} は式 (A.11) で示す射影行列である．また $N[\cdot]$ は単位ベクトルへの正規化を表す．

$$P_{\theta} = I - \theta_0 \theta_0^{\top} \quad (\text{A.11})$$

I は単位行列を表す．

4. $\theta \approx \theta_0$ なら θ を返して終了する．そうでなければ $\theta_0 \leftarrow \theta$ のように更新して手順 2 に戻る．このとき $(\theta, \theta_0) < 0$ であれば $\theta \leftarrow -\theta$ と符号を変える．

A.4 最小二乗法と重み反復法

最小二乗法を用いた最小化によらない方法として，重み反復法がある．重み反復法を用いた楕円当てはめの手順を次に示す．

1. $W_{\alpha} = 1, \alpha = 1, \dots, N, \theta_0 = 0$ と置く．
2. 次の行列 M を計算する．

$$M = \frac{1}{N} \sum_{\alpha=1}^N W_{\alpha} \xi_{\alpha} \xi_{\alpha}^{\top} \quad (\text{A.12})$$

3. 固有値問題

$$M \theta = \lambda \theta \quad (\text{A.13})$$

を解いて，絶対値最小の固有値に対する単位固有ベクトル θ を計算する．ここで，初期状態 $W_{\alpha} = 1$ のもとで θ を計算することが最小二乗法に一致する．

4. 符号を除いて $\theta \approx \theta_0$ なら θ を返して終了する．そうでなければ次のように更新して手順 2 に戻る．

$$W_{\alpha} \leftarrow \frac{1}{(\theta, V_0[\xi_{\alpha}] \theta)}, \quad \theta_0 \leftarrow \theta \quad (\text{A.14})$$

このとき $(\theta, \theta_0) < 0$ であれば $\theta \leftarrow -\theta$ と符号を変える．

A.5 Taubin 法とくりこみ法

Taubin 法を用いた最小化によらない方法として、くりこみ法がある。くりこみ法を用いた楕円当てはめの手順を次に示す。

1. $W_\alpha = 1, \alpha = 1, \dots, N, \theta_0 = \mathbf{0}$ と置く。
2. 次の行列 M, N を計算する。

$$\begin{aligned} M &= \frac{1}{N} \sum_{\alpha=1}^N W_\alpha \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^\top, \\ N &= \frac{1}{N} \sum_{\alpha=1}^N W_\alpha V_0[\boldsymbol{\xi}_\alpha] \end{aligned} \quad (\text{A.15})$$

3. 一般固有値問題

$$M\boldsymbol{\theta} = \lambda N\boldsymbol{\theta} \quad (\text{A.16})$$

を解いて、絶対値最小の一般固有値 λ に対する単位一般固有ベクトル $\boldsymbol{\theta}$ を計算する。行列 N が正値で無い場合は、一般固有値

$$N\boldsymbol{\theta} = \frac{1}{\lambda} M\boldsymbol{\theta} \quad (\text{A.17})$$

の絶対値が最大の一般固有値 $1/\lambda$ に対する単位一般固有 $\boldsymbol{\theta}$ を計算する。ただし、入力点列に誤差が含まれない場合、 $\lambda = 0$ になるので使えない。ここで、初期状態 $W_\alpha = 1$ のもとで $\boldsymbol{\theta}$ を計算することが「Taubin 法」に一致する。

4. 符号を除いて $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$ なら $\boldsymbol{\theta}$ を返して終了する。そうでなければ次のように更新して手順 2 に戻る。

$$W_\alpha \leftarrow \frac{1}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, \quad \boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta} \quad (\text{A.18})$$

このとき $(\boldsymbol{\theta}, \boldsymbol{\theta}_0) < 0$ であれば $\boldsymbol{\theta} \leftarrow -\boldsymbol{\theta}$ と符号を変える。

A.6 超精度くりこみ法と超精度最小二乗法

超精度最小二乗法は高次の誤差項を除いて偏差が存在しない楕円当てはめ法である。超精度くりこみ法は、超精度最小二乗法の解のサンプルソン誤差を評価し、反復計算することにより精度を向上させる。

超精度くりこみ法を用いた楕円当てはめの手順を次に示す。

1. $W_\alpha = 1, \alpha = 1, \dots, N, \boldsymbol{\theta}_0 = \mathbf{0}$ と置く。

2. 次の行列 M, N を計算する .

$$\begin{aligned}
 M &= \frac{1}{N} \sum_{\alpha=1}^N W_{\alpha} \xi_{\alpha} \xi_{\alpha}^{\top}, \\
 N &= \frac{1}{N} \sum_{\alpha=1}^N W_{\alpha} \left(V_0[\xi_{\alpha}] + 2S[\xi_{\alpha} e^{\top}] \right) \\
 &\quad - \frac{1}{N^2} \sum_{\alpha=1}^N W_{\alpha}^2 \left((\xi_{\alpha}, M_5^{-} \xi_{\alpha}) V_0[\xi_{\alpha}] \right. \\
 &\quad \left. + 2S[V_0[\xi_{\alpha}] M_5^{-} \xi_{\alpha} \xi_{\alpha}^{\top}] \right)
 \end{aligned} \tag{A.19}$$

ただし e は次のベクトルである .

$$e = (1, 0, 1, 0, 0, 0)^{\top} \tag{A.20}$$

また $S[\cdot]$ は対称化作用素であり ($S[A] = (A + A^{\top})/2$) , M_5^{-} は行列 M のランク 5 の一般逆行列 , すなわち M の最小固有値を 0 に置き換えた一般逆行列である .

3. 一般固有値問題

$$N\theta = \frac{1}{\lambda} M\theta \tag{A.21}$$

を解いて , 絶対値が最大の一般固有値 $1/\lambda$ に対する単位一般固有ベクトル θ を計算する .
ここで , 初期状態 $W_{\alpha} = 1$ のもとで θ を計算することが「超精度最小二乗法」に一致する .

4. 符号を除いて $\theta \approx \theta_0$ なら θ を返して終了する . そうでなければ次のように更新して手順 2 に戻る .

$$W_{\alpha} \leftarrow \frac{1}{(\theta, V_0[\xi_{\alpha}]\theta)}, \quad \theta_0 \leftarrow \theta \tag{A.22}$$

このとき $(\theta, \theta_0) < 0$ であれば $\theta \leftarrow -\theta$ と符号を変える .

超精度くりこみ法は重みを更新し , 当てはまり誤差が大きい点の影響度を下げることでノイズにロバストな楕円当てはめを実現している .

付録B 既存の楕円限定当てはめ

ここでは、楕円限定当てはめの既存手法である Fitzgibbon らの方法について説明し、筆者らが提案する楕円限定当てはめについて説明する。

B.1 Fitzgibbon らの方法

円錐曲線の式 (A.1) について、式 (B.1) が必ず満たされる楕円当てはめ法を初めて示したのは Fitzgibbon らの方法である。

$$AC - B^2 > 0 \quad (\text{B.1})$$

これは式 (A.1) の左辺の二乗和（「代数的距離」）

$$J_{\text{LS}} = \sum_{\alpha=1}^N (\xi_{\alpha}, \theta)^2 \quad (\text{B.2})$$

を $\|\theta\| = 1$ ではなく、次の正規化条件のもとで最小にするものである。

$$AC - B^2 = 1 \quad (\text{B.3})$$

具体的な手順は以下ようになる。

1. 次の行列 M_{LS} , N_{F} を計算する。

$$M_{\text{LS}} = \frac{1}{N} \sum_{\alpha=1}^N \xi_{\alpha} \xi_{\alpha}^{\top}, \quad N_{\text{F}} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (\text{B.4})$$

2. 一般固有値問題

$$N_{\text{F}} \theta = \mu M_{\text{LS}} \theta, \quad (\text{B.5})$$

の最大一般固有値 μ に対する単位一般固有ベクトル θ を計算する。

これにより必ず楕円が得られるが、精度は低い。それは各データ ξ_{α} の誤差特性を表す式 (B.6) の共分散行列

$$V[\xi_{\alpha}] \equiv E[\Delta \xi_{\alpha} \Delta \xi_{\alpha}^{\top}] = \sigma^2 V_0[\xi_{\alpha}] \quad (\text{B.6})$$

を考慮せず、一律に二乗和した式 (B.2) の代数的距離を最小にしているためである。

B.2 Szpak らの方法

最近 Szpak らは式 (2.7) の共分散行列を考慮して楕円のみを計算する方法を考案した .
式 (2.2) の楕円の条件を式 (B.4) の行列 N_F を用いて次のように書く .

$$(\theta, N_F \theta) > 0 \quad (\text{B.7})$$

式 (B.7) を考慮し , Szpak らは次の関数を最小にする θ を計算した .

$$J = \frac{1}{N} \sum_{\alpha=1}^N \frac{(\xi_{\alpha}, \theta)^2}{(\theta, V_0[\xi_{\alpha}]\theta)} + \frac{\lambda \|\theta\|^4}{(\theta, N_F \theta)^2} \quad (\text{B.8})$$

右辺第 1 項は「サンプルン誤差」を示す . θ の空間は楕円から成る領域 $(\theta, N_F \theta) > 0$ と双曲線から成る領域 $(\theta, N_F \theta) < 0$ とに分離され , 放物線から成る $(\theta, N_F \theta) = 0$ が境界面となる . 式 (B.8) の右辺第 2 項は境界面で ∞ になるので , 領域 $(\theta, N_F \theta) > 0$ 内の初期値から探索を行えば境界 $(\theta, N_F \theta) = 0$ を超えないというのが基本的な考え方である . これは一種の正則化であり , 正則化定数 λ は楕円が求まる限り微小にとる . Szpak ら [41] は式 (B.8) の最小化にレーベンバーグ・マーカート法 [17] を用いている .

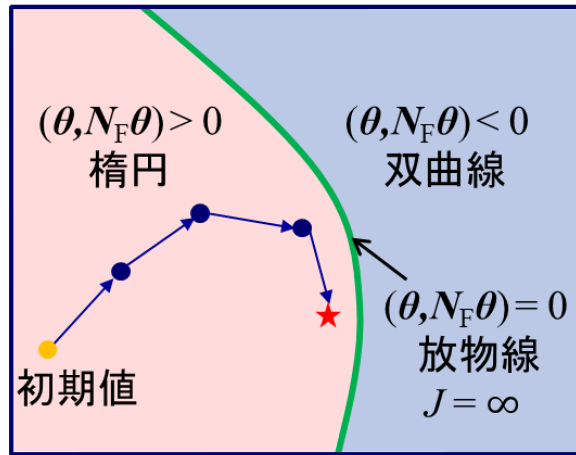


図 B.1: Szpak らの最小化 .

付録C 楕円弧選択のための放物線当てはめによるアウトライア除去

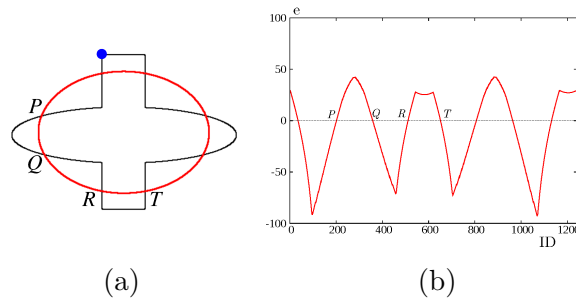


図 C.1: アウトライアを含む点列への楕円当てはめと誤差のグラフ. (a) 入力点列 (黒点) と当てはめた楕円 (赤). 入力点列は青丸を始点とし, 半時計回りに連結性を保って並んでいる. (b) 当てはめた楕円に対する入力点列との誤差. 横軸は (a) の青丸を始点とする点の番号. 縦軸は符号づけした楕円の当てはめ誤差.

本手法では, 誤差の変化を基にした楕円弧選択手法では除去できないアウトライアを除去するアルゴリズムを述べる.

図 C.1 の例では, 当てはめた楕円により入力点列が 8 つの部分弧に分割される. そして, インライアのみを含む部分弧 PQ のグラフでは, 誤差の変化が滑らかで凸形状になるのに対して, アウトライアを含む部分弧 QR のグラフでは, 誤差がピークの周辺で急激に変化しているため既存手法によってアウトライアと判定できる. しかし, アウトライアを含む部分弧 RT の誤差グラフでは誤差グラフのピーク値付近が凹形状になっており, 誤差グラフの曲率が小さいため, 前述の手法ではアウトライアとして除去できない.

そこで本手法では, 誤差グラフの曲率によって点列を選択する前に, 部分弧ごとの誤差グラフに放物線を当てはめ, 当てはめた放物線のピーク値が誤差グラフを上回る部分弧は誤差の曲率で判定できないアウトライアを含む部分弧として除去する.

放物線当てはめを用いた楕円弧選択手法の流れを以下にまとめる.

1. 分割した部分弧ごとに, 放物線を当てはめる.
2. 放物線のピーク値とその点での当てはめ誤差を比較する.
3. 当てはめ誤差より放物線のピーク値の方が大きい部分弧を選択する.
4. 選択した部分弧の放物線のピーク値付近の放物線と誤差グラフの当てはまり具合を判定し, 当てはまり具合が悪いものを除去する.

1 つの楕円弧からなる部分弧の誤差グラフは入力に類似した凸形状であれば, 当てはめた放物線のピーク値は, 図 C.2 (a), (c) のようにその点での誤差グラフの値と同じ値または誤差グラフよ

りも小さな値となると予想できる。これに対し、図 C.2 (b) のように放物線のピーク値付近のみ誤差の変化が緩やかになり凹形状になる部分弧は、当てはめた放物線のピーク値がその点での誤差グラフよりも大きい値を示すため、誤差グラフの曲率では除去できないアウトライアとして除去する。この手法で除去できないアウトライアを含んだ凸形状の誤差グラフを持つ部分弧もあるが、凸形状のグラフであれば誤差の曲率により判定できるため問題ない。

図 C.2 (c) のように放物線が当てはまる場合は、放物線が誤差グラフに沿うように当てはまっており、誤差グラフの形状が 2 次曲線であるならば、実際の入力点列も曲線と予想でき、インライアである可能性が高い。しかし、このような当てはめ結果では放物線のピーク値と誤差グラフの差の符号が入力に含まれるノイズの影響を受けてしまうため、インライアと正しく判定できない場合がある。この問題に対して、ピーク値の差よりアウトライアと判定された部分弧は、放物線のピーク値付近の点を取得し、それらの点の中で楕円の当てはめ誤差と当てはめた放物線の差が小さい点数を求め、取得した点数で割った値より放物線の当てはまり具合を調査し、当てはまり具合が良い部分弧は楕圆弧だと推定し、インライアの可能性が高いので除去せず、誤差グラフの曲率による判定を行なう。

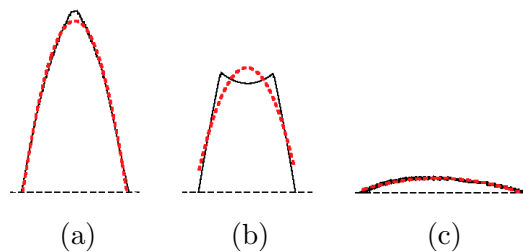


図 C.2: 部分弧の誤差グラフと放物線の当てはめ結果。黒線で誤差グラフ、赤の破線で当てはめた放物線を示す。(a) 楕圆弧点列からなる誤差グラフ。(b) 矩形のアウトライア点列からなる誤差グラフ。(c) 放物線がうまく当てはまる誤差グラフ。

分割した部分弧ごとの誤差グラフに放物線を最小二乗法を用いて当てはめ、アウトライア除去を行った後に、部分弧ごとの最大誤差を持つ点とその周辺の 2 点のなす角 ϕ によって、楕円を当てはめに使用する点列を選択する。この角度 ϕ は誤差のグラフの横軸のスケールに依存する。そこで、グラフの横軸値を

$$i_\alpha = \frac{\lambda e_{max} \alpha}{N} \quad (C.1)$$

のようにすべての入力点中の最大誤差 e_{max} を基準に正規化した値を表す。ここで、 λ は正規化のための定数¹、 e_α は式 (C.2) で計算される点 α の当てはめた楕円に対する誤差である。

$$e_\alpha = \sqrt{\frac{(\xi_\alpha, \theta)}{(\theta, V_0[\xi_\alpha] \theta)}} \quad (C.2)$$

¹実験では $\lambda = 2.0$ とした。

ここで，正規化共分散行列 $V_0[\xi_\alpha]$ を次のように置く．

$$V_0[\xi_\alpha] = 4 \begin{pmatrix} x_\alpha^2 & x_\alpha y_\alpha & 0 & f_0 x_\alpha & 0 & 0 \\ x_\alpha y_\alpha & x_\alpha^2 + y_\alpha^2 & x_\alpha y_\alpha & f_0 y_\alpha & f_0 x_\alpha & 0 \\ 0 & x_\alpha y_\alpha & y_\alpha^2 & 0 & f_0 y_\alpha & 0 \\ f_0 x_\alpha & f_0 y_\alpha & 0 & f_0^2 & 0 & 0 \\ 0 & f_0 x_\alpha & f_0 y_\alpha & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (\text{C.3})$$

正規化した部分弧に当てはめる放物線を $e_\alpha = ai_\alpha^2 + bi_\alpha + c$ とすると，最小二乗法で式 (C.4) の E を最小にする各係数 a, b, c を求める．

$$E = \sum_{\alpha=1}^N (e_\alpha - ai_\alpha^2 - bi_\alpha - c)^2 \quad (\text{C.4})$$

求めた係数より放物線のピーク値

$$(i_{\max}, h_{\max}) = \left(-\frac{b}{2a}, -\frac{b^2 - 4ac}{4a} \right) \quad (\text{C.5})$$

を求め， $i_{\hat{\alpha}} \leq i_{\max} < i_{\hat{\alpha}+1}$ となる $\hat{\alpha}$ 番目の点の誤差 $e_{\hat{\alpha}}$ を求める．当てはめた放物線のピーク値 h_{\max} と誤差グラフの値 $e_{\hat{\alpha}}$ の差

$$S = e_{\hat{\alpha}} - h_{\max} \quad (\text{C.6})$$

を求め， S が負の値を示し，当てはめた放物線のピーク値がその点の誤差グラフよりも大きい場合，その部分弧を除去する．

しかし，図 C.2 (c) のように誤差グラフに放物線が沿うように当てはまる場合，入力に含まれるノイズによって S が負を示しインライアとならない場合がある．このような場合に図 C.2 (b) は除去され，図 C.2 (c) の当てはめ結果は，点列の選択候補に残るように新たな指標を評価する．本研究では，放物線と誤差グラフの差がしきい値以下の点数の部分弧中の割合を評価する．しかし，図 C.2 (b) と図 C.2 (c) を評価する場合，部分弧の両端は図 C.2 (b) の部分弧でも放物線がよく当てはまっているため，この区間を含めると部分弧中で放物線とよく当てはまる点数が多くなってしまい，図 C.2 (c) との差が小さく，安定した動作が見込めない．放物線当てはめで図 C.2 (b) と図 C.2 (c) で差が現れるのは，放物線のピーク値付近である．放物線のピーク値付近では，誤差グラフがアウトライアを含む図 C.2 (b) のような場合，誤差グラフと放物線の差が大きくなり，図 C.2 (c) のような誤差グラフの変化が緩やかな形状の場合は，放物線との差は小さくなる．本手法では，放物線の当てはまる点数の差が大きく得られるピーク値付近のみに注目する．

本手法は，当てはめた放物線のピーク値から左右に (部分弧点数/4) 点で，各点 i_α ごとの放物線の値 h_α と楕円の当てはめ誤差 e_α の差の絶対値がしきい値 S_{\min}^2 よりも小さい点数を求め，(部分弧点数/4) で割ることで放物線のピーク値付近で差の平均を求める．当てはめた放物線と誤差グラフがピーク値付近の範囲で接する差の平均がしきい値 R^3 以上であれば除去せず，誤差グラフの曲率を計算し当てはめに使用する点列を選択する．

²実験では $S_{\min} = 2.0$ とした．

³実験では $R = 0.5$ とした．

付録D 壁とロボット間の距離推定

本研究で提案した境界認識システムを本節で説明する関らの手法 [37] の距離推定に適用し, 移動ロボットを制御する. 壁と床の境界線推定および壁とロボット間の距離推定の動作の流れを次に示す.

1. 入力画像から最長のエッジ点列を取得する.
2. 取得したエッジ点列から初期楕円上にあるエッジ点列を削除する.
3. 残った最長のエッジ点列に対して楕円を当てはめ, 壁楕円の楕円パラメータを取得する. もし, 残った点列がなければ手順 1 に戻る.
4. 当てはめた 2 次曲線が楕円か判定し, 楕円でなければ手順 1 に戻る.
5. 初期楕円との交点を 2 つ検出する. 交点が検出したどちらかの楕円上になれば手順 1 に戻る.
6. 2 つの交点から壁と床の境界を推定する.
7. カメラ座標の原点と境界の画像座標を結ぶ直線を求める.
8. あらかじめ求めた平面との交点から壁と移動ロボット間の距離を求める.

ロボットに装備したカメラの位置姿勢検出と壁との距離の推定について次節より詳しく説明する.

D.1 円形マーカによるカメラの位置姿勢推定

プロジェクタから投影した楕円の中心にマーカを置き, カメラキャリブレーションを行うことでカメラと投影した楕円の位置関係を推定する. 図 D.1 にカメラと円の位置関係を示す. ここで, d はカメラから支持平面までの距離, n は支持平面の法線ベクトル, r はマーカの半径, t_c はカメラから見た円の中心位置を表すベクトルとする. ここで, 支持平面とはマーカが乗っている平面を表す.

カメラと円の位置関係を知るために, 次の手順によりカメラから支持平面までの距離 d とカメラ座標系から見た円の中心位置 t_c を計算する.

1. マーカに当てはめた楕円から得られる楕円パラメータ $u = (A, B, C, D, E, F)^T$ を式 (D.1) に示す行列 Q で表現する.

$$Q = \begin{pmatrix} A & B & D \\ B & C & E \\ D & E & F \end{pmatrix} \quad (\text{D.1})$$

2. 楕円 Q を $\det Q = -1$ となるように正規化する. これは, ニュートン法を用いて, $x^3 = -\det Q$ の解を求め, x で Q の各要素を割ることで求まる.
3. 行列 Q の固有値を $\lambda_1, \lambda_2, \lambda_3$ とし, 対応する単位固有ベクトルを u_1, u_2, u_3 とする. また, $\lambda_1 \geq \lambda_2 > 0 > \lambda_3$ とする.
4. 支持平面の単位法線ベクトル n を次の式で求める.

$$n = N \left[\sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}} u_1 \pm \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}} u_3 \right] \quad (D.2)$$

ここで, $N[a]$ はベクトル a のノルム 1 とする正規化作用素である. また, n_i は, n の第 i 成分を表す.

5. 式 (D.2) では解が 2 つ得られる. これは, 円を見上げても見下ろしても画像として同じであるためである. 見下ろす解を得るために $n_1 n_3 \leq 0$ となる解を選択する.
6. カメラのレンズ中心から支持平面までの距離 d を次の式より求める.

$$d = \sqrt{\lambda_2^3 r} \quad (D.3)$$

7. 画像中の円の中心位置を指すベクトル x_{c1} を次の式より求める.

$$x_{c1} = Z[Q^{-1}n] \quad (D.4)$$

ここで, $Z[a]$ はベクトル a の最後の要素を 1 に正規化する正規化作用素である.

8. カメラ座標系から見た円の中心位置を表すベクトル t_c を次の式より求める.

$$t_c = \frac{dx_{c1}}{(-n, x_{c1})} \quad (D.5)$$

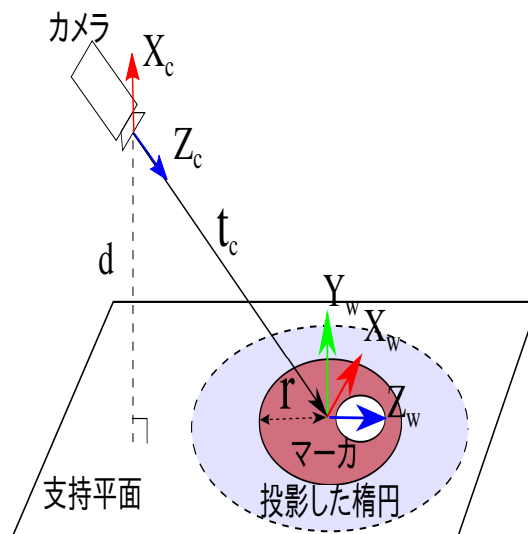


図 D.1: カメラと円の位置関係

D.2 距離推定

前述したカメラキャリブレーション結果と境界の画素座標から壁とロボット間の距離推定を行う。図 D.2(a) に示した画像面に写った楕円内の境界を 2 等分する画像座標 P_c を (x, y) とする。図 D.2(b) は、カメラと投影した楕円との位置関係を示す。また、図 D.3 に壁とロボットの実寸での位置関係を示す。

次に壁とロボット間の距離推定の流れを示す。

1. カメラ座標系から見た支持平面を表す平面の方程式 π は次式となる。

$$n_1(X_c - t_{c1}) + n_2(Y_c - t_{c2}) + n_3(Z_c - t_{c3}) = 0 \quad (\text{D.6})$$

n_i は、 \mathbf{n} の第 i 成分を表す。また、 t_{ci} は t_c の第 i 成分を表す。

2. カメラ座標系における 3 次元位置を表す直線の方程式 l は次の式となる。

$$\frac{X_c}{x} = \frac{Y_c}{y} = \frac{Z_c}{f} \quad (\text{D.7})$$

3. 境界の中間の画像座標 $P_c = (x, y)$ を取得する。

4. 平面 π と直線 l の連立方程式を解き、床と壁の境界の 3 次元位置座標 $P_h = (P_{hx}, P_{hy}, P_{hz})$ を求める。式 (D.7) と式 (D.6) から P_{hz} は次のように表せる。

$$P_{hz} = Z_c = \frac{n_1 t_{c1} + n_2 t_{c2} + n_3 t_{c3}}{n_1 \frac{x}{f} + n_2 \frac{y}{f} + n_3} \quad (\text{D.8})$$

式 (D.8) と式 (D.7) から P_{hx} , P_{hy} を次式でそれぞれ計算できる。

$$P_{hx} = \frac{P_{hz} x}{f} \quad (\text{D.9})$$

$$P_{hy} = \frac{P_{hz} y}{f} \quad (\text{D.10})$$

5. カメラ座標の原点から境界までの距離 d_1 を次の式から求める。

$$d_1 = \sqrt{P_{hx}^2 + P_{hy}^2 + P_{hz}^2} \quad (\text{D.11})$$

6. 壁とロボット間の距離 d_2 を次の式で求める。ただし、 $p[\text{mm}/\text{pixel}]$ はカメラのピクセルサイズとする。

$$d_2 = \sqrt{d_1^2 - d^2} \times p \quad (\text{D.12})$$

このようにして、壁とロボット間の距離を推定する。

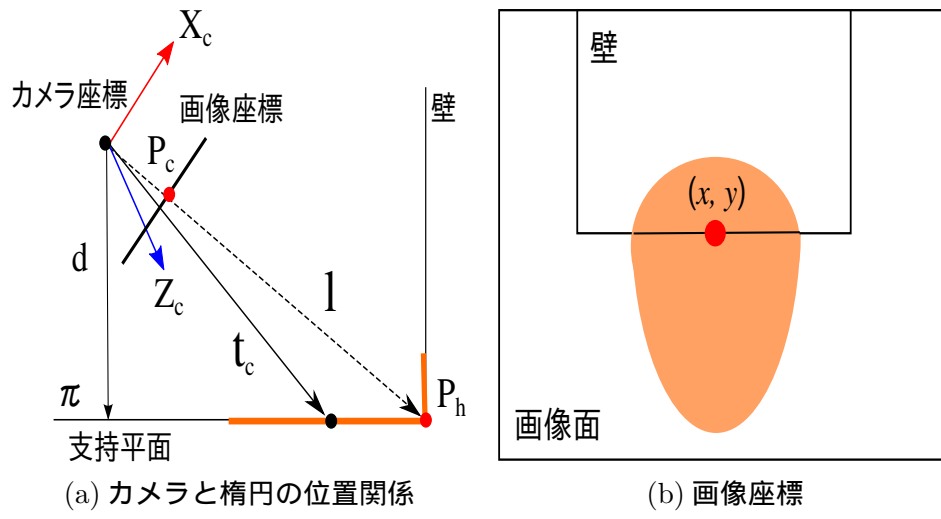


図 D.2: カメラと境界線との位置関係

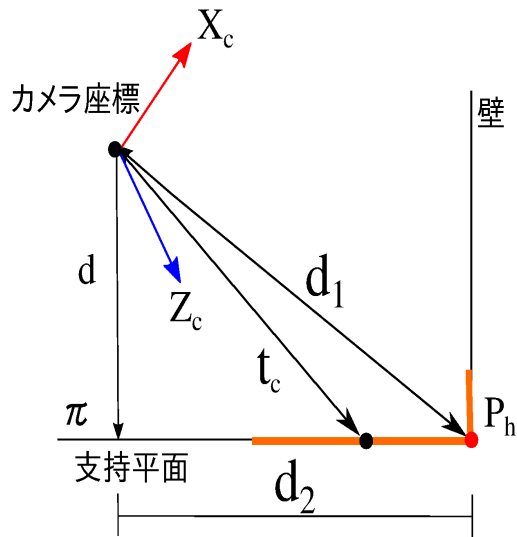


図 D.3: 壁とロボットの位置関係