

学位論文

リソース制約環境下における
探査機の自律地形相対画像航法に関する研究

狩谷 和季



総合研究大学院大学
物理科学研究科 宇宙科学専攻

2020年3月

Contents

1	序論	1
1.1	宇宙機における航法	1
1.2	地形相対航法	3
1.3	地形相対航法の必要性	4
1.4	地形相対航法の問題点	6
1.5	本論文の目的と構成	7
	Notes	8
2	宇宙機における自律画像処理を取り巻く環境	9
2.1	あらまし	9
2.2	宇宙機用の計算機環境	9
2.3	地形相対航法における画像処理	12
2.3.1	Mars Exploration Rover ミッション	14
2.3.2	MUSES-C ミッション	15
2.3.3	嫦娥3号ミッション	16
2.4	まとめ	18
	Notes	18
3	現状の低計算リソース環境における地形相対航法	21
3.1	あらまし	21
3.2	線分を用いた位置推定手法	22
3.2.1	地形相対航法における位置推定	22
3.2.2	点パターンマッチング	24
3.2.3	提案する位置推定手法の狙いと概要	26
	ねらい	26
	手法全体の概要	27
3.2.4	提案する線分情報を用いた位置推定手法	28

	線分の作成とデータベースの絞り込み	29
	探索範囲の決定	31
	マッチングの確認	32
3.2.5	アルゴリズムの検証	34
	検証内容と方法	34
	クレータ・線分データベースの作成	35
	足切りの閾値と最悪計算量の検証	38
	誤差に対する精度と計算量の検証	42
	検証のまとめと考察	48
3.3	動きブレ画像による速度推定手法	49
3.3.1	地形相対航法における速度推定	49
3.3.2	ブレによる画像の劣化	50
	パワーケプストラムによるブレ量の推定法	52
3.3.3	速度の算出法	54
3.3.4	提案する動きブレを含む単一画像を用いた位置推定手法	56
3.3.5	アルゴリズムの検証	57
	検証内容と方法	57
	動きブレとノイズを付与した月面画像の作成	59
	ノイズの存在する動きブレ画像に対する検証	74
	実画像を用いた検証	77
	検証のまとめと考察	80
3.4	宇宙機への搭載性評価	81
3.4.1	高位合成	81
3.4.2	設計法の評価	84
3.4.3	提案手法の宇宙機搭載性評価	86
	位置推定法の搭載性評価	86
	速度推定法の搭載性評価	87
3.5	まとめ	90
	Notes	91
4	将来の低リソース探査に向けた地形相対航法	93
4.1	あらまし	93
4.2	Neuromorphic Computing	97
4.2.1	スパイクングニューラルネットワーク	98
	神経細胞における処理の概要	98

	スパイキングニューロンのモデル	100
	シナプスのモデル	101
4.2.2	人工ニューラルネットワーク	102
	人工ニューロンのモデル	102
4.2.3	ネットワークのアーキテクチャ	104
	畳み込みニューラルネットワーク	105
	ネットワークの学習	106
4.2.4	SNN の実施	109
	ソフトウェアシミュレータ	109
	ニューロモーフィックハードウェア	110
	イベントカメラ	110
4.3	提案する ANN から SNN への変換手法	112
4.3.1	ANN の定式化	113
4.3.2	SNN の定式化	114
4.3.3	変換手法	116
4.4	シミュレーションによる検証実験	119
4.4.1	月面画像による検証	119
	検証方法	119
	検証結果	120
	消費電力の解析	124
4.4.2	イベントカメラを入力とした検証	124
	検証方法	124
	検証結果	128
	消費電力の解析	130
4.5	まとめと考察	131
	Notes	132
5	結論	135
	References	137
	List of Publications	147
	Acknowledgement	149

List of Figures

2.1	一般的な地形相対航法における画像処理	13
2.2	MER における地形相対航法のシーケンス	14
2.3	MER における地形相対航法アルゴリズム	15
2.4	GCP-NAV のプログラム画面	16
2.5	嫦娥 3 号での降下フェーズの概要	17
2.6	嫦娥 3 号での障害物検知法	18
3.1	点パターンマッチング概念図	24
3.2	提案手法の位置推定の流れ	28
3.3	月クレータ点パターンの例	29
3.4	線分リストの対応関係	30
3.5	重ね合わせ概念図	31
3.6	誤差範囲	32
3.7	位置推定アルゴリズムフロー	33
3.8	月面地形画像	36
3.9	作成したクレータデータベース, 赤点が記載クレータ	37
3.10	データベース中の線分分布と使用線分範囲	38
3.11	データベース画像範囲中の平均線分分布と使用線分範囲	38
3.12	各密度のデータベースにおける, $ M' $ に対して一致する箇所の数	40
3.13	模擬点パターンの一例	43
3.14	露光時間中に斜め方向に運動したブレ画像	50
3.15	動きブレによる劣化過程	51
3.16	各パワーケプストラムの概形	53
3.17	カメラモデル	54
3.18	平均化の有無によるケプストラムの差異	57
3.19	速度推定のアルゴリズムフロー	58

3.20	動きブレのない原画像	60
3.21	動きブレのない原画像のケプストラムと $L = 15$ の PSF の概形	61
3.22	ノイズのない動きブレ画像のケプストラム	62
3.23	明度変化を与えたブレ画像	64
3.24	明度変化を与えた動きブレ画像のケプストラム	65
3.25	コントラスト変化を与えたブレ画像	66
3.26	コントラスト変化を与えた動きブレ画像のケプストラム	67
3.27	焦点ぼけを与えたブレ画像	68
3.28	焦点ぼけを与えた動きブレ画像のケプストラム	69
3.29	輝度の揺らぎを与えたブレ画像	70
3.30	輝度の揺らぎを与えた動きブレ画像のケプストラム	71
3.31	放射線ノイズを与えたブレ画像	72
3.32	放射線ノイズを与えた動きブレ画像のケプストラム	73
3.33	各データセットにおける付与したブレ幅ごとの推定精度	75
3.34	各データセットにおける付与したノイズ量ごとの推定精度	76
3.35	ブレ幅が 7-20pix での各データセットにおける付与したノイズ量ごとの 推定精度	78
3.36	実験装置概観図	79
3.37	実験用画像	79
3.38	速度推定結果	80
3.39	設計手法の比較	82
3.40	高位合成を用いた設計フロー	83
3.41	C 言語による全方向微分フィルタリング記述	84
3.42	全方向微分フィルタリングの入力と出力例	85
3.43	速度推定法の搭載性検証に用いた入力画像	89
4.1	計算機の電力・性能とクロック数の傾向	95
4.2	ニューロンの模式図	99
4.3	RC 回路として動作する細胞膜の模式図	100
4.4	人工ニューロンとスパイクの模式図	103
4.5	LeNET-5 のアーキテクチャ	105
4.6	イベントカメラの動作原理	112
4.7	入力が定電流の時の LIF ニューロンの応答	115
4.8	入力がノイズのある電流の時の LIF ニューロンの応答	116
4.9	検証に使用した ANN	120

4.10	検証に用いた入力データ	121
4.11	検証の出力データ	122
4.12	各ステップにて分類に失敗した入力画像	123
4.13	イベントカメラによるデータセット内容	126
4.14	検証に使用した ANN	127
4.15	イベントカメラ入力の検証に用いた入力データ	129
4.16	イベントカメラ入力の検証の出力データ	129
4.17	$\tau_r = 0, \tau_m = 50$ の時の発火率応答	133

List of Tables

1.1	宇宙機における航法技術の分類	2
1.2	成功した着陸ミッションの一覧	5
2.1	主要な宇宙機用 CPU	11
2.2	主要な宇宙機用 FPGA	12
3.1	点パターンマッチングの手法概要	25
3.2	計算機実験環境	35
3.3	前提条件の諸元	35
3.4	データベース記載情報の諸元	36
3.5	各データベースにおける閾値 R	39
3.6	データベース切り出しの場合の位置推定精度と計算時間	41
3.7	未検出・誤検出が 10% の場合の位置推定精度と計算時間	45
3.8	未検出・誤検出が 20% の場合の位置推定精度と計算時間	46
3.9	未検出・誤検出が 30% の場合の位置推定精度と計算時間	47
3.10	ブレ画像に付与するノイズとその量	63
3.11	対象 FPGA	85
3.12	評価用ツール	85
3.13	高位合成系と Verilog の比較結果	86
3.14	処理時間	88
3.15	消費リソース	88
3.16	出力検証結果	90
3.17	消費リソース	90
3.18	処理時間	90
4.1	従来の計算機と神経模倣型計算機との間の特徴差	96
4.2	ANN を SNN へと変換するモデル	118

4.3	ニューロンのパラメータ	120
4.4	提案手法の各ステップにおける分類精度	121
4.5	クレータ識別のネットワークが使用する消費電力の概算	124
4.6	イベントデータを入力とした各ステップにおける分類精度	130
4.7	イベントデータを入力としたクレータ分類のネットワークが使用する消費電力の概算	131

Chapter 1

序論

1.1 宇宙機における航法

古代において、ある人類が初めてその住処から出て周辺を探索している時、彼は次のような疑問を抱いたと思われる。つまり、「目的地へと辿る (誘導; guidance) にあたり、いったい自分は今どこにいるのか (航法; navigation)」。地球に関する知識が何ら存在せず、測定手段を持ち得ない古代においては、彼は自らの視覚情報と記憶とを、脳でもって照らし合わせることで、その問題に対処する他なかつたはずである。その後、近代までに人類は、船舶や車両、航空機、宇宙機を用いて、地球上を隈なく探索し、それら移動体の航行を導くにあたって、地形や天体、電波による航法技術が発展した。現在では、地球上の個々の人間は、衛星測位システムによって数 m 程度の誤差で供される位置情報を、自らの航法手段として用することまで可能となっている。

地球外の探索においては、人類はその技術力をもって他の天体へと版図を広げ、宇宙機を用いて、着陸を伴う精密な探査を行う段階にまで至っている。しかしながら、地球上で培われた諸航法技術を活用することが難しい条件下において、宇宙機は、古代人類の彼と同様の問題を、同様に対処する必要がある。すなわち、「自分は今どこにいるのか (航法)」という問題に対して、視覚 (画像) 情報と記憶 (地図) を脳 (計算機) でもって照らし合わせる、という解決手段である。本論文では、宇宙機における航法技術の中でも、画像を用いた航法 (vision-based/visual navigation)、特に地形相対航法 (Terrain Relative Navigation; TRN) に関する研究について述べる。

航法とは、船舶や航空機などの移動体において時々刻々と変化する位置や速度、姿勢を計測・推定することによって、それらの航行を導く機能を指す。宇宙機において使用される航法技術を分類すると、表 1.1 のようになる。

地文航法 (pilotage navigation) は、搭乗員の目視によって地形の特徴を認識し、それをランドマークとして地図と照らし合わせることで位置を推定する手法である。航法技術

表 1.1 宇宙機における航法技術の分類

航法技術	記述
地文航法	目視での地形と地図との照合による測位
天測航法	既知の恒星による姿勢や相対位置の推定
自蔵航法	搭載機器のみによる速度や位置、姿勢の推定
電波航法	無線施設との交信による速度や位置の推定

としては最も初歩的なものであり、精度よく実行するためには、地形図などのランドマークが記された地図が必要となる。

天測航法 (celestial navigation) は、背景に見える恒星や至近の天体の縁を観測することによって、位置や姿勢を推定する手法であり、恒星追跡器 (star tracker) や光学航法 (optical navigation) などが含まれる。恒星追跡器では、撮像素子によって供される画像に映る恒星の位置を測定することで、姿勢を推定する。光学航法では、画像に映る恒星の位置と至近の天体 (小惑星や彗星など) や、その縁を観測することで、その天体との相対的な姿勢を推定する。宇宙機における天測航法は、主として姿勢情報を供する手法であり、位置決定のための慣性航法や電波航法と組み合わせて使用されることが多い。

自蔵航法 ^{*1}(dead reckoning navigation) は、装置に搭載されたセンサから取得される情報のみを使用して、位置や速度、姿勢を推定する手法であり、慣性航法やドップラレーダ (doppler radar) 航法などが含まれる ^{*2}。慣性航法では、慣性航法装置 (Inertial Measurement Unit; IMU) における、加速度計によって取得された加速度計測値を積分することで位置や速度を推定し、ジャイロによって取得された角速度計測値を積分することで姿勢を推定する。ドップラレーダ航法では、ドップラ効果による反射波の周波数の変移を計測することで、速度を推定する。自蔵航法は、装置外の機器によって供される情報を使用することなく航法値を推定するため、通信の環境などに性能が左右されることがないという長所を持つ一方で、始点からの移動距離に応じて誤差が累積されるという短所を持つ。

電波航法 (radio navigation) は、既知の地点にある無線施設からの電波を利用して、位置や速度を推定する手法であり、RARR (Range And Range Rate) 法や DDOR (Delta Differential One-way Range) 法などが含まれる。これらの手法では、地上局と宇宙機との間の電波交信によって取得される、距離や距離変化率などの計測値から位置や速度を推定する。衛星測位システムは、地球周回上に配置された衛星を無線施設として見做すことで測位を行う。電波航法は、宇宙機の軌道決定において主たる役割を果たしているが、地球と宇宙機との間の距離に応じて精度が悪化する点や、情報取得のスループットの悪さが問題点となる。

これらの航法技術は、いずれかを選択すれば全ての航法値を賄えるという種類のものではなく、ミッション毎に必要なとされる航法値の種別や精度を鑑み、時と場合に応じて複合的に使用することが通常である。例を挙げると、はやぶさ2ミッションにおいては地球から約3億 km 先の天体へと接近するために、電波航法による位置の誤差(数百から数 km)を、天測航法(特に光学航法)を用いて修正することで、約900 mの大きさである小惑星に到達せしめた。

1.2 地形相対航法

地形相対航法とは、いわば自動化された地文航法である。地形相対航法では、撮像素子によって取得された天体地表面の動画像に対して、地形特徴の抽出やマッチング、追跡などの画像情報処理を施すことによって位置や速度などの航法値を推定する。他の航法技術と比較したときの利点としては、地表面画像には、地形や障害物、相対的な移動量など多岐に渡る情報が含まれており、位置や速度のみならず物体の認識や領域識別といった、多様な応用が可能であることや、光学カメラと計算機という、搭載性が高く比較的低リソースな構成で動作することが挙げられる。また近年では、Flash LIDAR (Light Detection And Ranging) のような能動方式の撮像装置を用いて、より高度な地形識別などを行う研究例も存在する [1, 2]。

宇宙機において初めて地形相対航法が使用されたのは、2004年に火星への着陸を果たした、MER (Mars Exploration Rover) ミッションにおける、EDL (Entry, Descent, and Landing) 時である。搭載の経緯は次のようになる。開発最終段において、着陸安全性の確保のため、EDL時に地表面に対して水平方向の速度測定を実施する必要性が判明したが、ドップラレーダのような従来航法センサの搭載には電力・質量リソースが不足した。そこで、余剰のカメラインタフェースを使用し、画像処理によって速度検出が可能な手法の検討がなされた [3]。このMERにおける地形相対航法の手法はDIMES (Descent Image Motion Estimation System) と呼称される。DIMESでは、パラシュートによる降下中の撮影画像から、コントラストの高い地形を特徴として抽出し、画像フレーム間で追跡することで、水平方向速度を推定した。一方で、制限のある計算機による画像処理に時間が費やされることから、約15 sの降下期間中に3回の撮影と2回の速度推定に留まった。

その後、地形相対航法を用いた探査機として、MUSES-C (はやぶさ) が存在する。MUSES-Cにおいては、往復約40分の通信伝搬遅延が存在する中、限られた着陸地点への誘導のために、高精度で測位をする必要性があった。電波航法や自蔵航法の位置決定精度を増強するため、GCP-NAV (Ground Control Point NAVigation) と呼称される画像

処理ベースの航法が開発された。GCP-NAV では、事前の観測により取得した小惑星の形状モデルと、撮影画像中の特徴点とを照合する事で位置を推定した [4]。岩石やクレータといった特徴点の抽出と照合の処理は、小惑星地表面の画像を地球に送信し、手動にて実施された。この手法は人間の手を介して実施されるため、事前に想定し得なかった状況や誤差に対して柔軟に対応可能である。一方で、地球と探査機間における通信の伝搬遅延や位置の同定に要する時間の存在により、画像送信から位置情報の取得まで約 1 時間の時間遅れが生じた。

また、月面への軟着陸を果たした嫦娥 3 号においては、着陸最終段において事前観測の不可能な障害物を検出・回避するために、地形相対航法が使用された [5]。

1.3 地形相対航法の必要性

地球以外の天体に対する探査ミッションは、太陽系の起源や生命のルーツを探るといった理科学的関心のみならず、科学技術の発展のため、各国で勢力的に行われてきた。記念碑的な出来事として記憶されている例としては、米国と旧ソ連間での宇宙開発競争において、旧ソ連のルナ 9 号が史上初めて地球外天体への軟着陸を達成し、米国のアポロ 11 号では遂に人類を月面に送り込む事に成功した。その後も、米国は勢力的に着陸探査を行っており、MSL (Mars Science Laboratory) ミッションにおいては、スカイクレーンを用いて 900 kg の探査車を軟着陸させる事に成功している。また、我が国でも、MUSES-C がアポロ群の小惑星に着陸し、史上初めて地球重力圏外天体からのサンプルリターンを果たした事は記憶に新しい。表 1.2 に、着陸に成功したミッションと使用された航法技術の一覧を示す。

ここで示したように、2000 年代以前の全てのミッションにおいては、電波航法による軌道決定、天測航法による姿勢推定、自蔵航法による着陸寸前の高度や速度推定が主として使用されており、それらによる着陸誤差は数 km (月) から数十～百 km (火星) のオーダーである。これは偏に、着陸探査ミッションそのものが、天体の初期探査や着陸技術の獲得を狙ったものであるという面による。すなわち、ミッションの成功性に主眼がおかれ、広大な着陸誤差をも問題としない、障害物の少なく安全で広い領域へと着陸地点が設定されたために、その着陸誤差は大きく考慮されていなかった。実際、人による測定が可能なアポロ 11 号においても、着陸誤差楕円はダウンレンジ方向で 20 km、クロスレンジ方向でも 5 km の領域である [6]。この設計思想は 2000 年代に入っても基本的には変わっていない。

しかしながら、近年ではこれらのミッション成果の獲得、および周回機による全球的な天体表面情報の取得が進んだことにより、科学ミッションで要求される探査領域は、従来の着陸しやすい場所から、精密に観測したい特定の場所へと移行してきている。そのよう

表 1.2: 成功した着陸ミッションの一覧

国	ミッション名	対象天体	航法方式
旧ソ連	ルナ	月	電波航法 [7]
米国	サーベイヤー	月	自蔵航法 (高度・ドップラレーダ, IMU), 電波航法 [8, 9, 10, 11, 12]
米国	アポロ	月	地文航法, 天測航法 (人力), 自蔵航法 (高度・ドップラレーダ, IMU), 電波航法 [6, 13, 14, 15, 16, 17]
旧ソ連	ベネラ	金星	天測航法, 自蔵航法 (レーダ高度計), 電波航法 [18]
旧ソ連	マルス	火星	天測航法, 自蔵航法 (レーダ高度計), 電波航法 [18]
米国	バイキング	火星	天測航法, 自蔵航法 (高度・ドップラレーダ, IMU), 電波航法 [19]
旧ソ連	ベガ	金星	天測航法, 自蔵航法 (レーダ高度計), 電波航法 [20]
米国	パスマインダー	火星	天測航法, 自蔵航法 (レーダ高度計), 電波航法 [21]
2000 年以降↓			
米国	MER	火星	天測航法, 自蔵航法 (レーダ高度計, IMU), 電波航法, 地形相対航法 [22, 23]
欧州	ホイヘンス	タイタン	天測航法, 自蔵航法 (レーダ高度計), 電波航法 [24]
日本	MUSES-C	イトカワ	天測航法, 自蔵航法 (LIDAR, IMU, レーザ高度計), 電波航法, 地形相対航法 [?]
米国	フェニックス	火星	天測航法, 自蔵航法 (高度・ドップラレーダ, IMU), 電波航法 [25]
米国	MSL	火星	天測航法, 自蔵航法 (高度・ドップラレーダ, IMU), 電波航法 [26]
中国	嫦娥 3 号	月	天測航法, 自蔵航法 (高度・ドップラレーダ, LIDAR, IMU), 電波航法, 地形相対航法 (障害物検知) [5]

な科学探査要求の高度化や、探査地域により近い場所へ着陸するといったようなミッションリターンの効率化を考えると、従来よりも高精度な着陸を可能とする航法技術の獲得が期待される。

高精度な着陸の達成には、MUSES-C で実施されたように、天体近傍での着陸フェーズにおいて、自蔵航法、特に慣性航法の位置・速度推定誤差を修正する何らかの手法が必要となる。加えて、通信の伝搬遅延時間が長大である場合や、重力天体のように高速に降下する場合には、地球を介することのない自律的かつリアルタイムな航法処理が求められる。この観点において、地表画像を処理することによって航法値を得ることのできる、地形相対航法への期待は非常に大きい。従来の航法技術と比較した地形相対航法の利点は、次のように識別することができる。

- 探査対象天体近傍において、電波航法や自蔵航法より高精度な航法値の取得が可能である。
- 航法に必要な装置が光学カメラと計算機であり、レーダ等と比較して電力・質量リソースの面で有利である。
- 位置や速度といった情報のみならず、障害物検知のように画像処理特有の多様なタスクの実施が可能である。

これらの利点から、月面に対して 100 m 級の高精度で着陸を行う、SLIM (Smart Lander for Investigating Moon) ミッション [27] や、火星衛星へと着陸してサンプルリターンを行う、MMX (Martian Moons eXploration) ミッション [28] において、着陸精度や安全性の向上のために地形相対航法の搭載が計画されている。

1.4 地形相対航法の問題点

高精度な着陸の実施のための航法技術として、画像情報処理を使用する地形相対航法の、従来の航法技術に対する利点は既に述べた。近年の地球上における画像情報処理分野の発展はめざましく、地形相対航法に使用される、物体認識や識別、追跡、照合などのアルゴリズムは、高度化の一途を辿っている。近年では、人工知能 (Artificial Intelligence; AI) の分野、特に機械学習 (Machine Learning; ML) や深層学習 (Deep Learning; DL) に関する研究が画像処理分野においても流行の様相を見せている。

月や火星といった重力天体、あるいは地球から遠く離れた木星や土星圏などに存在する天体を探査対象とし、地形相対航法を実施しようとする場合には、画像の取得から航法値の算出までを、これら画像処理によって機上で自律的かつ実時間で行う必要がある。これは、前者では降下開始から着陸までの時間が限られること、後者では通信の往復伝搬遅

延時間の存在によって、地球を介した航法処理が困難となるためである。しかしながら、自律的な地形相対航法の宇宙機への搭載にあたっては、計算機の性能が非常に低いという、避けがたい問題点が存在する。

宇宙機で使用される計算機は、主には熱的制約により動作クロックに上限があることや、耐放射線性の獲得に起因して、地上の汎用計算機と比較すると計算処理性能が劣る。従来の機上データ処理において、姿勢制御やコマンドのデコード、システム管理のようなタスクについては、多くの処理能力を必要としないために性能の低い単純なマイクロコントローラでも実施可能である [29]。一方で、画像処理 (特徴抽出やマッチング、追跡) のような集中的なデジタル信号処理は、より高度なマイクロプロセッサでの実行が必要となる。近年では、FPGA (Field-Programmable Gate Array) のような、並列処理によって処理速度の向上が期待できる、非ノイマン型のプロセッサチップが宇宙機での画像処理において主流となりつつあるものの、地上における CPU (Central Processing Unit) の処理速度に対して、1-2桁程度処理速度が落ちる結果となるのが現状である [30]。

この宇宙用計算機の計算処理性能は、地上において 20 年程度前のものと同程度^{*3}であり、現在主流となる画像処理アルゴリズムを直接搭載するには難がある。さらに将来の探査ミッションについて考えると、現在主流となる月火星より太陽から遠く離れた、低エネルギーリソースの天体に対し、より高度な画像処理を用いた航法によって着陸を行うことが予想される。したがって、この実時間の計算処理要求と、低リソースな計算機能力との間の懸隔は、自律的な地形相対航法を実施するために埋めるべき重要な課題と認識される。

1.5 本論文の目的と構成

「リソース制約環境下における探査機の自律地形相対画像航法に関する研究」と題した本論文は、着陸探査機において、光学画像を用いた地形相対航法を自律的に実施するための二種類の研究をまとめたものである。リモートセンシングによる広域データが取得されている現在、より科学的に関心高い地点の直接的な探査を可能とする高精度着陸に対する関心が高まっている中で、その実現のためのキー技術である、地形相対航法の一つの成立解を示すことは重要である。

前述のように、自律的な地形相対航法の実施には、画像処理アルゴリズムの計算規模と、宇宙機における計算機の処理能力との間に存在する、ミスマッチを解消する必要がある。本論文では、およそ次の二つの方向性からこの問題に対処する。

- 画像処理アルゴリズムの計算規模を落とすことによる、低計算リソースである現状

の宇宙機用計算機に搭載可能な、地形相対航法手法の研究.

- 現状の計算機より低電力リソースで動作する革新的なプロセッサアーキテクチャに対する、地形相対航法手法の適用性の研究.

以下に、本論文を構成する各章の目的と概要を示す.

第2章「宇宙機における自律画像処理を取り巻く環境」では、本論文で問題とする、宇宙機における計算機環境について消費電力面から概説する. また、そのような計算機環境において実施されてきた地形相対航法の手法について説明する.

第3章「現状の低計算リソース環境における地形相対航法」では、月着陸機をターゲットとして、計算機環境が低リソースである状況においてもリアルタイムかつ自律的に位置・速度情報を推定することが可能な地形相対航法手法をそれぞれ提案する. また、それらの手法について、実際に宇宙機で使用される耐放射線性の計算機に実装を行うことで、その搭載性について評価する.

第4章「将来の低リソース探査に向けた地形相対航法」では、将来の探査においては低エネルギーリソース環境下にて、高度な地形相対航法の必要性があることを予期し、その解決手段として、ニューロモーフィックコンピューティングに着目した. 具体的には、ニューロモーフィックな計算アーキテクチャにおいて、地形相対航法に使用される画像処理アプリケーションを具体的に適用することによって、低消費電力性や性能においてメリットが得られることを確認・検証する.

第5章「結論」では、以上の内容について総括する.

Notes

*1 より最近では「自律航法」との呼称が一般的なようであるが、本論文では自律的 (autonomous) との混同を避けるためにこの表記とする.

*2 搭載機器としてのレーダから供される距離や速度の測定値による航法は、しばしば電波航法として分類されることがある (語感からすれば当然である). この種の技術分類によく見られるように、分類先の定義が互いに独立でないことに起因しての揺らぎであるが、本論文では搭載機器としての側面を重視するために自蔵航法に分類している.

*3 例えば、宇宙機において多く用いられる CPU である RAD750 は、PowerPC 750 をベースとしている. これは 1997 年に発表されたものである [31].

Chapter 2

宇宙機における自律画像処理を取り巻く環境

2.1 あらまし

本章では、地形相対航法の自律化に関して、その実施に際して必要となる計算機環境とその上で動作する航法アルゴリズムの現状について整理する。

本論文が、宇宙機における画像処理アルゴリズムと、計算機の間には存在するギャップを埋めることを目的とすることは既に述べた。

2.2 節では、現状の宇宙機における計算機を取り巻く環境と具体的な性能について整理する。続く 2.3 節では、2.2 節の状況下において、地形相対航法で使用される画像処理技術について述べる。

2.2 宇宙機用の計算機環境

自律的な地形相対航法のような、高度な組み込み画像処理を宇宙機にて実施する場合、その航法装置は宇宙特有の制約に囚われる。そのような電子システムに利用可能な宇宙機のリソースは制限されており、その主な要素は電力や処理能力、質量、体積である [29]。質量や体積は機器の構成に左右されるが、地形相対航法は光学カメラと計算機によって構成されるため、他の搭載機器と比較して大きな問題となることは少ない^{*1}。一方で、電力は最もシビアとなる要素であり、太陽光発電による少ない供給と熱的な制約によって、大きく制限される。CPU や FPGA といった、クロックによって稼働する現在の計算機において、処理能力はその電力リソースと耐放射線性の獲得によって制限を受ける。したがって、地形相対航法のアプリケーションを考えるにあたっては、宇宙機における計算機の環境を把握し、その性能に適う設計を行うことが肝要となる。

宇宙機における計算機プロセッサの多くは、対放射線性を備えた IBM PowerPC と Gaisler LEON CPU を使用している。

消費電力の観点において、LEON ベースのプロセッサは PowerPC ベースのものよりも低リソースであり、10 から 20 W に対して 1 から 2 W の電力消費量となる。

処理能力の観点においては、近年販売されている LEON ベースの GR712RC でおよそ 200 DMIPS (Dhrystone Million Instructions Per Second) ^{*2} であり、PowerPC ベースの RAD750 でおよそ 400 DMIPS となる。最新の高性能モデルとして、32bit アーキテクチャで 4 コアの CPU となる、LEON ベースの E698PM で 2000 DMIPS、64bit アーキテクチャで 4 コアとなる PowerPC ベースの RAD5545 で 5000 DMIPS という性能の計算機が存在している。

1999 年に発売された汎用 CPU である、Intel Pentium III 800 の処理性能が約 2000 DMIPS、2004 年に発売された、Intel Pentium 4 531 の処理性能が約 5000 DMIPS である [32]。最新となる対放射線性 CPU の性能はこれらと同程度であり、2014 年に発売された、地上用途の Intel Core i7-4790 の処理性能が約 130000 DMIPS であることを考えると、宇宙機で使用される対放射線性 CPU は、地上の計算機で使用される CPU と比較して、処理性能が 1 から 2 桁程度低く、15 から 20 年程度の遅れがあることを示している。表 2.1 に、主要な宇宙機用途の対放射線性 CPU および比較対象としての民生 CPU の諸元を示す ^{*3}。

画像処理の観点において、これら CPU ベースの計算機よりも高い処理性能を確保するために、GPU (Graphics Processing Unit) や FPGA のような並列演算が可能な計算機の使用が選択肢に入る。GPU は、多数のコアを使用することによって大きな命令やデータの並列処理を実行するため、消費電力が大きくなる傾向にある。したがって、現状の宇宙機にはあまり用いられていない。一方で、FPGA は小さな処理回路とメモリのネットワークで構成されており、それらを組み合わせて深いパイプライン処理や並列化を実行するために、電力を大きく消費することなく処理性能の向上が可能である。

宇宙機用の対放射線性 FPGA は、一般的な計算機である CPU に対して活用できる範囲が限られているために、多くの選択肢があるわけではない。表 2.2 に現在利用可能な宇宙機用 FPGA および比較のための民生 FPGA の諸元を示す ^{*4}。

FPGA での画像処理の実行においては、ロジック数とメモリ量が性能に影響を与える。この観点から、同じ企業から提供される製品の中では、Xilinx 社の Virtex-5QV と Microsemi 社の RTG4 が最も高性能となる。それでも、民生用途において近年使用されている Xilinx Zynq 7000 SOC と比較すると、1 桁程度ロジック数とメモリ量が少ない。近年発売された対放射線 FPGA では、ATMEL 社の ATFEE560 や NanoXplore 社の NG-MEDIUM が存在するが、これらは上記 Virtex-5QV などと比較してもロジック数が

表 2.1 主要な宇宙機用 CPU

	デバイス	動作周波数	DMIPS	消費電力
PowerPC	RAD750 (Rad-Tolerant)	110-130 MHz	260 @133 MHz	14 W @133 MHz
	RAD750 (Rad-Hard)	200 MHz	400 @200 MHz	14 W @200 MHz
	SCS750 SBC	400-800 MHz	1800 @800 MHz (200@400 MHz)	7-25 W
	RAD5545 64-bit 4-core	466 MHz	5600	20 W
	TSC695F SPARC V7	10-25 MHz	20 @25 MHz	1 W @25 MHz
LEON	AT697E SPARC V8	100 MHz	86 @100 MHz	1 W @100 MHz
	GR712RC Dual-Core SoC	100 MHz	140-200	1.5 W ×2
	OCE E608PM 32-bit 4-core	600 MHz	2100	2.8 W
	Pentium III 800	800 MHz	2000 @800 MHz	20.8 W
地上用途	Pentium 4 531	3000 MHz	5000 @3000 MHz	84 W
	Core i7-4790	3600 MHz	130000 @3600 MHz	84 W

少なく、現状画像処理に適しているとは言えない。

自律的な画像処理の需要が高まっていること、およびその需要を満たす宇宙機用の FPGA が少ないことから、FPGA を用いた新たな計算処理アーキテクチャの研究が存在する。例を挙げると、NASA のジェット推進研究所では、Xilinx Zynq 7000 をベースとして、放射線などの宇宙環境に耐えられる高性能な計算機システムを開発している [33]。このシステムでは、ウォッチドッグタイマなどの機能を、FPGA における処理の監視に用いることで、民生 FPGA の処理性能を保ちながらも耐放射線性を獲得している。また、

表 2.2 主要な宇宙機用 FPGA

デバイス	ロジック数	メモリ	消費電力
Xilinx Virtex-5QV	81920 LUT6	596 RAMB16	5-10 W
Microsemi RTG4	151824 LE	5.3 Mbit	1-4 W
Microsemi ProASIC3	35000 LE	0.5 Mbit	2 W
Microsemi RTAX	37000 LUT	0.5 Mbit	TBD
ATMEL ATFEE560	28800 LUT4	0.23 Mbit	TBD
NanoXplore NG-MEDIUM	34272 LUT4	56 RAMB48	TBD
↓放射線耐性なし (民生)			
Xilinx Zynq 7000 SOC	277400 LUT6	1510 RAMB18	3-6 W
Microsemi SmartFusion2 SOC	146124 LE	4.5Mbit	2 W

CHREC プロセッサ [34] も Zynq 7000 をベースとして開発されており，耐放射線性のフラッシュメモリを使用することで宇宙機での使用に対応している。

2.3 地形相対航法における画像処理

1.2 章で述べたように，地形相対航法とは自動化された地文航法である。地文航法とは，移動体に搭乗する人間が地形を認識し，地図あるいは記憶と照らし合わせることで，自機の位置を推定する方式であった。

これを自動的に行うためには，まず地形情報を画像として取得する必要がある。最も一般的に使用されるのは，撮像素子を用いた光学カメラである。他には，レーザ距離計や LIDAR を使用して距離画像を取得する場合も存在するが，本論文では，光学カメラを用いた手法に限定する。これは，機器がコンパクトとなることや，ハードウェアが単純であること，低消費電力性，他の用途にも使用可能な汎用性から，他の撮影装置と比較して有利なためである。

続いて，地形相対航法を実施するための画像処理として，次の方策が取られることが多い。

- 撮影された地表面画像から，何らかの地形特徴を抽出し，情報として保持する。
- 特徴情報を地図あるいは前フレームの画像と照合し，位置や速度情報を得る。

この一般的な地形相対航法における画像処理の概要を図 2.1 に示す。

前者では，周回機によって得られた画像と着陸機によって得られた画像という，時刻や日照・姿勢条件の異なる画像においても，同等に識別可能な特徴点を抽出する事が問題と

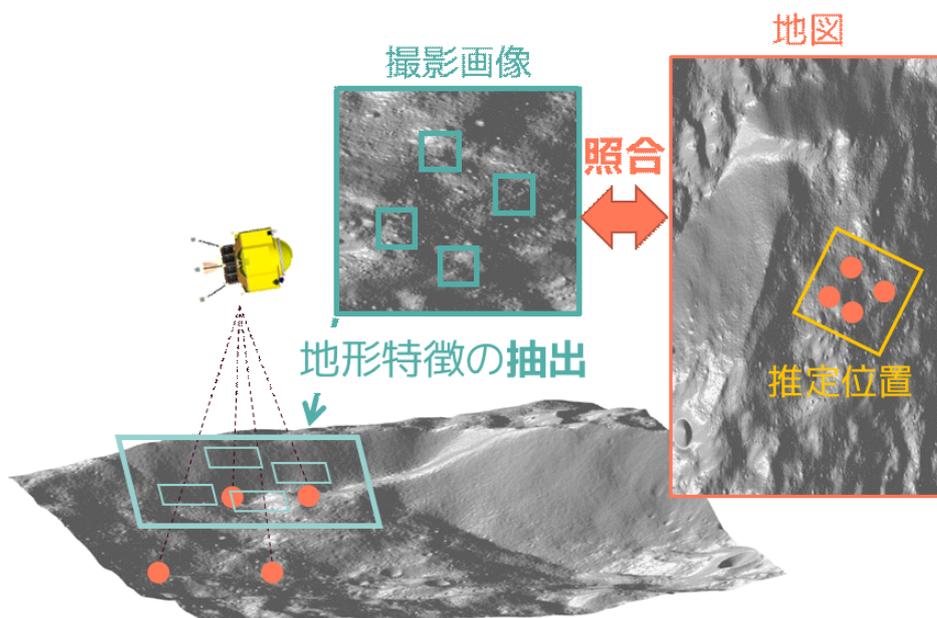


図 2.1 一般的な地形相対航法における画像処理

なる。ここで使用される特徴点には、画像内の輝度勾配によって発生する、エッジ (edge) やコーナー (corner) といった直接的なものや、画像の様相変化に対して、ロバスト性を高めるために設計された特徴量記述子によって得られるキーポイント (keypoint) などが含まれる。これらの特徴点を検出する一般的なアルゴリズムとして、Canny のエッジ検出 [35]、Harris のコーナー検出 [36] や、特徴量ベースの手法として、SIFT[37] や SURF[38] が多く使用される。加えて、より高級なランドマークとして、月などに特有のクレータ形状を抽出する手法も存在する [39, 40]。

後者では、地図 (データベース) と比較して、得られた特徴点に欠落や挿入、位置ずれが存在する中で、ロバストかつ高精度、リアルタイムな照合を行う事が問題となる。特徴点の抽出に、特徴量ベースの手法を使用している場合には、抽出した特徴点群と地図との間で、その特徴量を比較することで対応が得られるため比較的容易である。そうではない、画像内の位置情報として特徴点群を取得している場合には、RANSAC[41] や ICP[42] のような位置合わせ手法、あるいは単にパターンマッチングによって対応を求める。

これらの画像処理手法は、汎用の計算機で用いられることが想定されているため、リアルタイム処理が求められる自律的な地形相対航法における、処理性能の低い計算機に直接の搭載は難しい。以下では、過去に実施された地形相対航法の処理内容について概説する。

2.3.1 Mars Exploration Rover ミッション

2004年に火星に着陸を行った、NASAのMERミッションでは、そのEDL時においてエアブレーキとパラシュートによる減速を行った後、エアバッグを用いて着陸した。エアバッグによる着陸方式では転倒の心配は無いものの、大気中の風の影響によって減速が不十分となった場合、水平方向の速度が設計限界を超え、着陸時の衝撃によってエアバッグの破損が危惧された。質量リソース等の観点から、ドップラレーダなどによる速度測定ではなく、降下時の撮影画像を用いた水平方向速度の推定を行い、レトロモータで減速することで速度を許容値に収める方式を採用した。このMERにおける地形相対航法のシーケンスを図2.2に、速度推定手法のフローチャートを図2.3に示す^{*5}。

この地形相対航法システムであるDIMESでは、2枚の降下時画像と機体の姿勢、高度情報を入力とし、まず画像に対して重なり得る領域の計算を行う。その領域中でHarrisのコーナー検出法を用いて、特徴点中の輝度の高い上位2箇所を検出する。その後、輝度補正と姿勢・高度情報から算出される、ホモグラフィ (homography) 行列による射影変換を行う。最後にパターンマッチングにより特徴点を追跡し、その移動量から速度の推定を行っている。

このアルゴリズムはIBM社の宇宙機用耐放射線CPUであるRAD6000 @20 MHzに搭載され、計算量の削減のため1024×1024 pixの画像サイズに対して、使用画像領域を256×256 pixに限定し、注目領域を削減する等の工夫が行われている。しかしながら、処理時間は約15秒となっており、リアルタイム性に乏しく、降下時の速度推定回数は2回

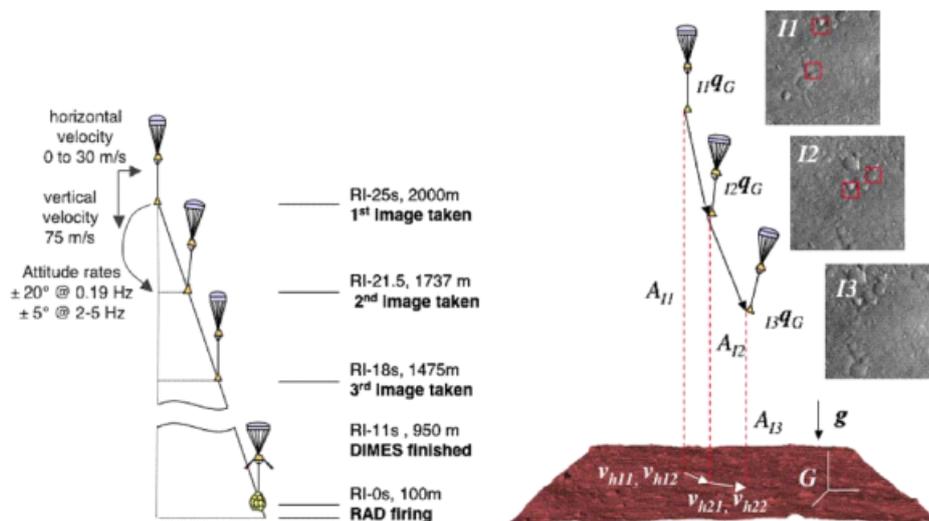


図 2.2 MER における地形相対航法のシーケンス

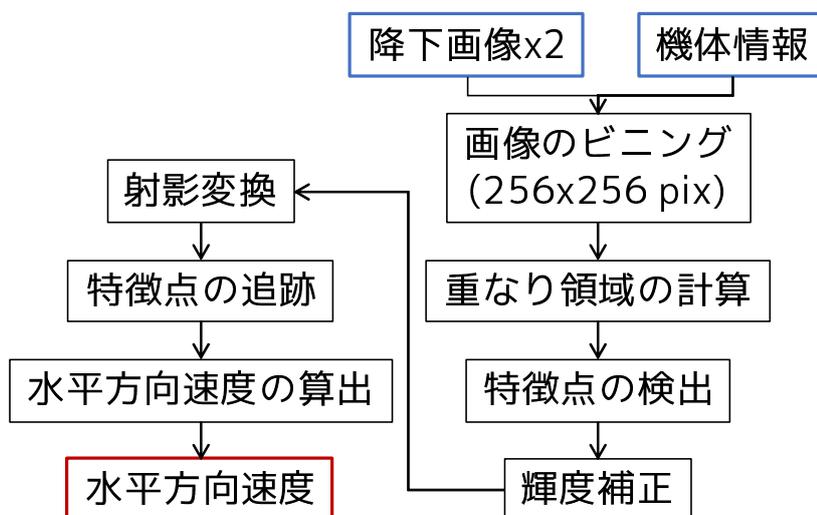


図 2.3 MER における地形相対航法アルゴリズム

に留まっている。

2.3.2 MUSES-C ミッション

2005 年に小惑星イトカワへとタッチダウンを行った、我が国の MUSES-C ミッションでは、往復約 40 分の通信伝搬遅延が存在する中、限られた着陸地点への誘導のために、高精度で測位をする必要性があった。そこで、光学複合航法と呼ばれる、画像を用いた着陸方式が用いられた。光学複合航法においては、高度 500 m までの降下フェーズにおいて GCP-NAV と呼ばれる撮影画像による位置同定が、その後の最終降下フェーズではターゲットマーカをランドマークとした、画像による相対水平方向速度の推定が行われた。

GCP-NAV は、事前の観測によって得られた小惑星のモデルと、撮影画像中の特徴点とを照合する事で探査機位置を同定する手法である。岩石やクレータといった特徴点の抽出と照合の処理は、小惑星地表面の画像を地球に送信し、手動にて実施された。地上における GCP-NAV の処理画面を図 2.3.2 に示す [43]。この手法は人間の手を介して実施されるため、事前に想定し得なかった状況や誤差に対して柔軟に対応可能である。

一方で、地球と探査機間における通信の伝搬遅延や位置の同定に要する時間の存在により、画像送信から位置情報の取得まで約 1 時間の時間遅れが生じた。重力天体を対象とした場合、降下速度が速くなるため、このような時間遅れの存在はクリティカルな問題であり、よりリアルタイム性のある手法が求められる。

最終着陸フェーズにおける、ターゲットマーカを用いた水平方向速度の推定手法では、ターゲットマーカに対して投光している画像としていない画像の差分を取ることで、マー

カを抽出した。その後、輝点の大きさと明るさを評価値として、その中心位置を推定することで、その移動量から水平方向速度の計算を行った。このアルゴリズムは、IBM 社の宇宙機用耐放射線 CPU である RAD6000 に搭載され、自律的な処理が行われた。

2.3.3 嫦娥 3 号ミッション

2013 年に月面へと軟着陸を成功させた、中国の嫦娥 3 号ミッションにおいては、垂直降下フェーズの最終段階にて画像による障害物の検知を自律的に行った。嫦娥 3 号の降下フェーズの概要を図 2.5 に示す [5]。

嫦娥 3 号においては、動力降下を行った後、高度約 3 km 地点で垂直降下を行った。その後、高度約 100 m 地点において 10 秒程度のホバリングを行い、その間得られた画像から広域での障害物を検知し、最終着地点を決定した。最後に、決定した着陸地点に向かう低速降下フェーズにおいて、LIDAR によって得られた三次元の数値標高モデルを用いた、局所的な障害物検知と傾斜の検知を行い、安全な着陸を図った。

ここでの画像を用いた障害物検知手法は、ホバリング中の比較的高速な処理が求められたため、軽快なアルゴリズムとなっている。具体的には、まず入力されたグレースケールの画像に対してヒストグラム解析を行う。そこで求められたある閾値を用いて画像を二値化することで、明な部分に障害物が存在するとした。その後、その二値画像に対してエッ

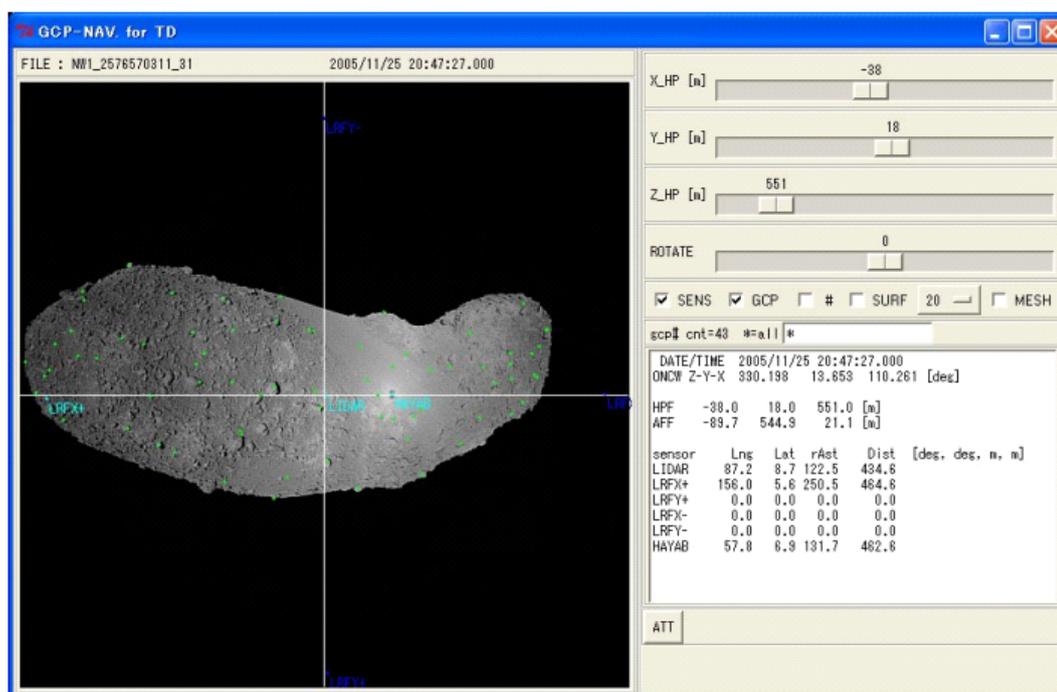


図 2.4 GCP-NAV のプログラム画面

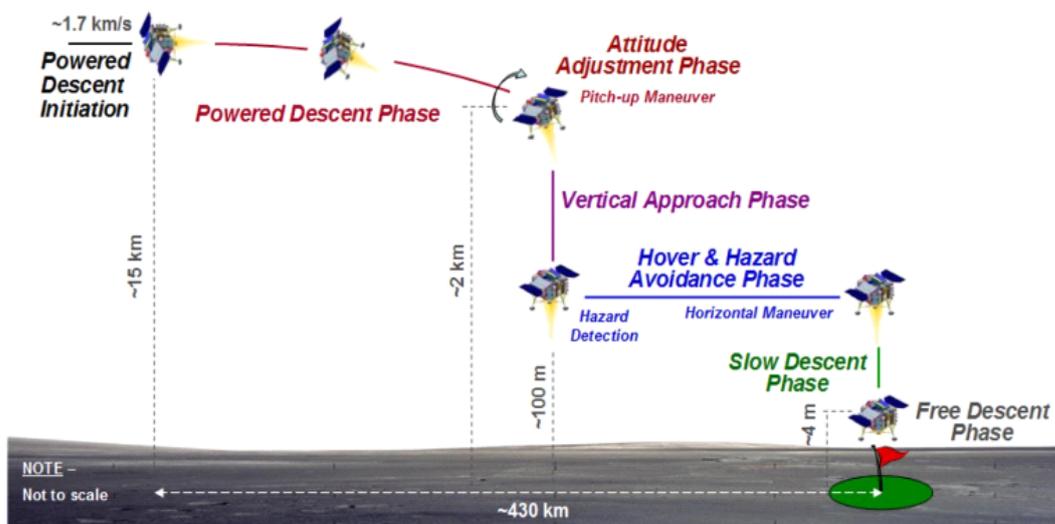


図 2.5 嫦娥 3 号での降下フェーズの概要

ジの検出を行い、障害物の分布を求め、安全領域と非安全領域に分割する。この画像に対して、着陸範囲を表す窓を設け、中心から渦を描くように走査することで、最も画像中心から近い安全着陸領域を同定している。(図 2.6)。

この手法は軽快なものであり、リアルタイム性に優れる。一方で、このような画像の明な部分のみを用いた障害物検知は、細やかな障害物を見落としやすく、また日照が浅い場合に影の影響が大きく出てしまい、障害物の同定が不安定なものになってしまう恐れがある。この問題に対処すべく、嫦娥 3 号ではホバリング後の低速降下フェーズにおいて、ライダーによって観測された三次元数値標高モデルから細かな障害物の検知と傾斜の算出を行っているが、この時点で着陸地点は殆ど決定してしまっているため、ホバリング時の着陸地点決定が間違いであった場合、安全領域がそもそも存在しない事態となる可能性が存在した。

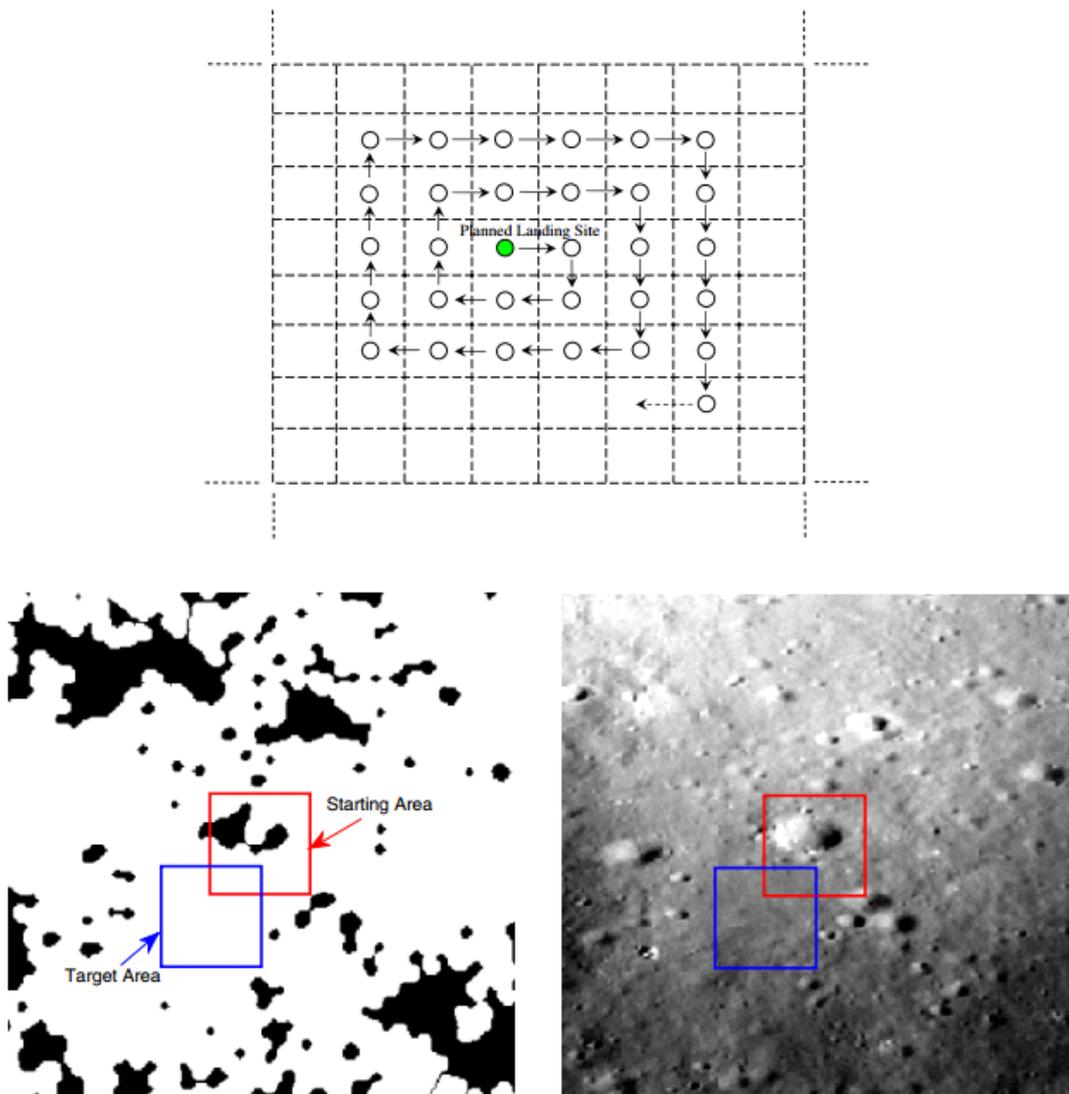


図 2.6 嫦娥 3 号での障害物検知法

2.4 まとめ

本論文で取り扱う事柄として、宇宙機における計算機環境の現状と用いられる画像処理手法について概説した。

Notes

*¹ 実際、前述のように MER における DIMES はその観点から急遽搭載されるに至った。

*² Dhrystone ベンチマーク [44] によって測定された VAX 11-780 計算機の性能を 1 DMIPS とした指標値。1 MIPS は、ベンチマークテストにおいて、浮動小数点演算を含まない命令を 1 秒間に 100 万回実行したことに等しい。1 MIPS の性能を持つ VAX 11-780 の Dhrystone 測定値が 1757 であるので、各プロセッサにお

る Dhrystone 測定値を 1757 で除したものとなる。

*³ ベンチマークテストによる性能測定は環境によって多少上下する。表中では、開発元の資料 [31]、CPU の網羅的なベンチマーク評価 [32]、また [30] 中のまとめを参考にした。

*⁴ FPGA におけるロジック数やメモリの単位は、提供する企業によって統一されていないために、数字を直接比較することは難しい。(Xilinx 社ではロジックは LUT (Look Up Table), Microsemi 社では LE (Logic Element) で表記される。) 同一の企業が提供したものに関しては比較が可能である。

*⁵ シーケンス図とフローチャートは共に [23] を参照した。

Chapter 3

現状の低計算リソース環境における 地形相対航法

3.1 あらまし

1.4 節で述べたように，着陸機が重力天体表面に着陸を行うとき，従来のような慣性航法のみを用いた場合では，その着陸精度に限界が存在する．これは，慣性航法装置による航法値の計測時において，その起点からの航行距離に応じて誤差が蓄積するためである．したがって，高精度な着陸を達成するためには，その誤差を何らかの方法によって修正する必要がある．加えて，その誤差修正は，降下速度が非常に速いことや通信時の往復伝搬遅延から，地上を介すことのない自律的な実施が必須となる．

ここで得る航法値として，着陸機が動力降下を行う誘導パス中の位置情報と速度情報は特に重要であり，ロバストな測定を高頻度に行うことができれば，より高精度な着陸が可能となる．しかしながら，2.2 節で述べたように，宇宙機における計算機リソースは，熱的制約や耐放射線性の獲得のために，地上の計算機環境と比較して2桁程度性能面で劣る．

このように計算機環境が低リソースである状況において，画像を用いた自律的な地形相対航法を実施するためには，現状入手可能な宇宙機用計算機においても動作するよう，航法アルゴリズムの計算規模を下げつつも性能を確保する必要がある．この観点から，本章では月着陸機をターゲットとして，上記のような制限下においてもリアルタイムかつ自律的に位置・速度情報を推定することが可能な地形相対航法手法をそれぞれ提案する．また，それらの手法について，実際に宇宙機で使用される耐放射線性の計算機に実装を行うことで，その搭載性について評価する．

3.2 節では，クレータの位置情報を活用することで，ロバストかつ高精度に機体の位置情報を推定可能な，新たな地形相対航法手法について提案し，性能評価を実施する．

3.3 節では、単一画像中に含まれる動きブレの成分を活用することで、従来の手法よりも低計算リソースで速度情報を推定可能な、新たな地形相対航法手法について提案し、性能評価を実施する。

3.4 節では、それらの手法について実際に宇宙機用計算機に実装し、性能を測ることで宇宙機への搭載性を評価する。

3.2 線分を用いた位置推定手法

3.2.1 地形相対航法における位置推定

地形相対航法において探査機の位置を推定するにあたっては、予め取得された惑星表面の地形情報により得られた地図と、降下中に探査機が撮影した地形画像とを照合することにより、地図中の画像位置を算出する画像レジストレーション (image registration) 技術が用いられる。画像レジストレーションの手法は、テンプレートマッチングや位相限定相関 [45] のような画像そのものの輝度値やその空間周波数を使用して、相関を取ることで位置合わせを行うものと、画像中における輝度のエッジやその方向、あるいはランドマークとなる物体などの特徴的な点を抽出し、その点群の情報を使用することにより位置合わせを行うものに分けられる。

相関ベースの手法は、地図と取得画像間の様相変化や幾何学変換にロバストでない一方で、画像情報全体を用いるためにサブピクセル単位の精密な位置合わせが可能となることから、時系列画像によるモザイクングや超解像のような目的で使用される。特徴点ベースの手法は、画像中の特徴となる情報について適切に選択することによって、地図と取得画像間の様相変化や幾何学変換にロバストな位置合わせが可能である。また地図も画像そのものではなく、疎な特徴点の集合となるため、そのサイズも相関ベースの手法と比較して小さく抑えられる。地形相対航法においては、地図となる地形画像が周回機による事前の取得に基づくことから、画像レジストレーション実施時の取得画像との間に様相変化や幾何学変換が生じる。したがって、特徴点ベースの手法が使用される [46]。

特徴点ベースの画像レジストレーションにおいては、まず画像中の輝度情報から特徴となる点を抽出する。一般的に使用される特徴として、エッジ [35] やコーナー [36] のように画像中の輝度構造を直接的に用いるものから、SIFT[37] や SURF[38] のように画像の変化に頑健となるよう輝度勾配からなる特徴量を設計することによって、特徴となる情報を記述するものが存在する。これらの手法は、輝度そのもの、あるいはその勾配情報によって特徴点を記述するため、照合させたい画像間においてテクスチャが同等であることが前提となる。したがって、明度やコントラストの変化には頑健な抽出が可能である一方

で、地形相対航法における地図と取得画像間に発生するような、照明方向・強度の変化に対応することは難しい。

そこで、照明の変動に依存しない特徴として、地形形状そのものを検出するように設計された、特徴抽出器を使用する方策がとられる。地形相対航法の場合、月惑星において普遍的に存在する地形としてクレータが主に利用され、照明環境の変動に対してロバストに検出する手法が提案されている [39, 40]。これらの手法では、入力された輝度画像から特徴点としてクレータの位置情報を返す。

続いて、画像同士の位置合わせを行うために、抽出された特徴点を地図と取得画像の間で重ね合わせる必要がある。このような問題では、地図と取得画像における特徴点情報は二次元ユークリッド空間上に与えられた二つの点集合となり、それらに幾何変換を与えて重ね合わせることで、地図上の取得画像の位置を算出することが目的となる。このような位置合わせ手法を点パターンマッチング (point pattern matching) と呼び、指紋照合や物体認識の分野で重要な役割を果たしている。

点パターンマッチング問題は組合せ問題であるため計算複雑度が高く、地図と取得画像における点群のサイズが大きくなるにつれ爆発的に計算量が大きくなる。特に、地形相対航法のように広大な地図範囲に対する画像の位置合わせを、FPGA のような組み込みの計算機にて処理する例はほぼ無い。よく似た問題設定として、スタートラッカに搭載される星の同定アルゴリズムが存在するものの、特徴量として星の等級情報を用いる [47] ほか、画像間のテクスチャ変化を想定しておらず、特徴点の誤検出や未検出に対してシビアであることから、着陸機の位置推定問題に直接適用することは難しい。

このように、地形相対航法特有の状況下にて従来の点パターンマッチング手法を使用することは計算量の観点にて難がある。しかしながら、地形相対航法を実施する状況について考えると、回転やスケール変化について事前の軌道計画により探索範囲を設けることが可能である。したがって、それらの制約を緩和する事により、宇宙機用の計算機環境においても自律的な位置推定を実施高速なマッチングを行う事が可能であると目される。

そこで本節では、地形相対航法における自律的な位置推定を限られた計算機環境において実施可能とするために、クレータを特徴点とした場合における地図と取得画像間のマッチング手法について提案する。また、その手法の性能について対応可能となる範囲を評価し、加えて実際の宇宙機用計算機に対する実装性を評価する。

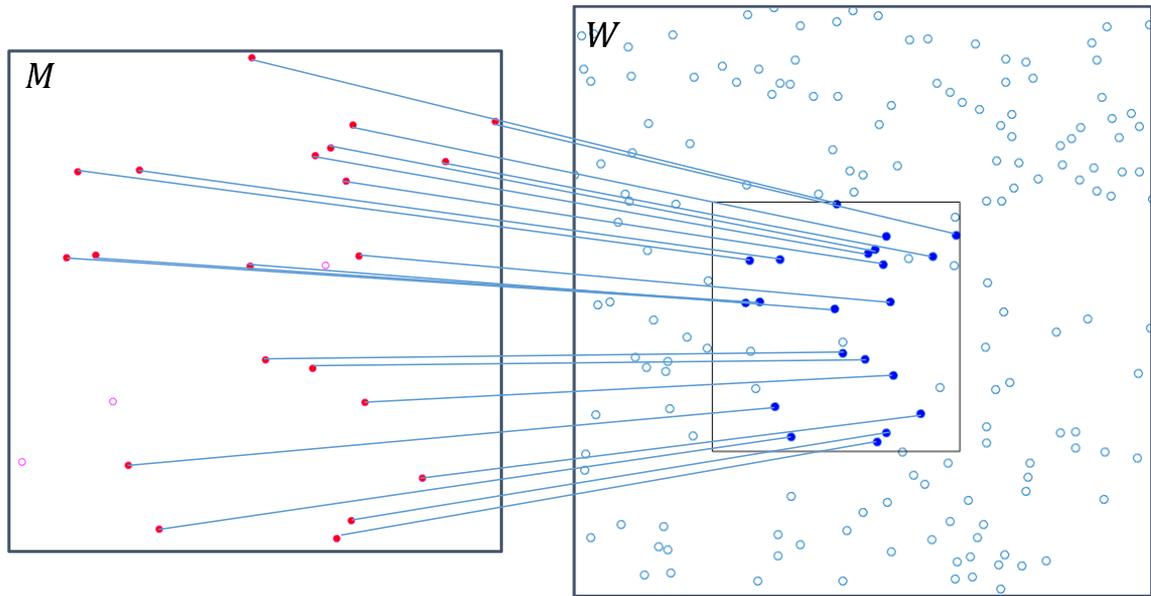


図 3.1 点パターンマッチング概念図

3.2.2 点パターンマッチング

点パターンマッチングとは、ある二つの点群が与えられたとき、それら各々の点群の部分集合が最もよく重なり合う幾何学変換を求めることにより、点群同士のレジストレーションを行う手法である。

二次元ユークリッド空間に与えられた、ある点の集合をモデル点パターン M 、その位置合わせ先として与えられる点の集合をワールド点パターン W とし、それぞれの要素数が $m = |M|, n = |W| (m < n)$ であるとき、これらの集合を次のように表す。

$$M = \{p_1, p_2, \dots, p_m\} \quad (3.1)$$

$$W = \{q_1, q_2, \dots, q_n\} \quad (3.2)$$

ここで、 $p_i, q_j \in \mathbb{R}^2$ である。このような M, W の点パターンから対応関係を発見し、最もよく重なる幾何変換を導出することによって、 W における M の位置把握を行うことが点パターンマッチングの目的である。この点パターンマッチングの概念図を図 3.1 に示す。

モデル点パターン M とワールド点パターン W のそれぞれの部分集合 $M' \subseteq M, W' \subseteq W$ ($|M'| = |W'| = N$) 間の対応関係がある程度判明している場合、この問題は次

のように定式化できる.

$$\text{minimize } \sum_{i=1}^N \left(f_t(M'_i, P_{trans}) - W'_i \right)^2 \quad (3.3)$$

f_t は幾何学変換であり, P_{trans} はそのパラメータである. 幾何学変換として相似変換に基づいた重ね合わせを考えると, この変換 f_t は次のように表される.

$$f_t \left(M'_i, s, \theta, T \right) = s \text{rot}(\theta) M'_i + T \quad (3.4)$$

s はスケールファクタ, $\text{rot}(\cdot)$ は二次元の回転行列, θ はその回転角, $T \in \mathbb{R}^2$ は平行移動である. モデル点パターン M とワールド点パターン W から2点ずつ選択することで, これらの変換パラメータは一意に決まる. したがって, それら全ての組合せについてパラメータを算出し, 式 (3.3) に基づき評価することにより, いずれ最適な解を発見することができる. しかしながら, モデル点パターン M とワールド点パターン W のサイズ m, n の値が増加するにつれ, これら点の組合せ数は爆発的に増加するため, 処理に多くの時間がかかってしまい現実的でない.

この解決のため, 緩和法 (relaxation) を用いた手法 [48][49] や, 点パターンのマッチングを順に評価する手法 [50][51], 点パターンをクラスタリングすることで高速化を図る手法 [52] などが提案されている. これらの手法では, 並行移動, 回転, スケールの変化を許したまま, いかにして位置合わせを行うかを問題としている. また, モデル点パターン M とワールド点パターン W のサイズ m, n が数十個程度で同等 ($m = n$) であることや, 点群間の誤差については考慮していないものが多い. これら手法の方策と対応する誤差, 最悪計算量を表 3.1 に示す.

表 3.1 点パターンマッチングの手法概要

手法	考慮される誤差	最悪計算量
Relaxation[48]	並行移動	$O(m^4)$
Fuzzy relaxation[49]	並行移動, 回転, スケール変化	$O(m^6)$
Alignment[50]	並行移動, 回転, スケール変化	$O(n^3 m^2 \log m)$
Randomized alignment[51]	並行移動, 回転, スケール変化	$O(m^2 n \log m)$
2-D Cluster[52]	並行移動, 回転, スケール変化, 誤・未検出	$O(m^4)$

3.2.3 提案する位置推定手法の狙いと概要

ねらい

着陸機の位置推定問題のアプローチとして、点パターンマッチングを考える場合、一般的な手法と比較して前提条件が次の点において異なる。

- **広大な探索範囲**： M と W の要素数と範囲に数十倍近い差が存在する。
- **誤・未検出点の存在**： W の検出元となる地形画像と、実観測によって得られる画像との観測条件の差異を起因とした点の付与と削除、位置ずれによって、 M 中に誤検出、未検出点が存在し、また完全な重ね合わせを行う事が出来ない。
- **計算コスト**：数十 MHz 程度のクロック周波数である宇宙機用計算機においても、リアルタイム推定が可能となるような計算コストが求められる。
- **誤差範囲**：他のセンサによる測定等によって、回転・スケール変化、自己位置が一定の誤差範囲で規定できる。

特に、探索すべき範囲が広大であることから、スケール・回転変化を許容しつつ局所的なマッチング関係を逐次的に評価してゆくような手法や、緩和法といった反復的手法の適用は計算コストの面で難がある。また、多角形の対応関係を用いて、一挙に範囲を絞り込むような手法においても、誤検出や未検出といった問題から、図形的に同等な多角形を一意に作成すること自体が難しい。

一方で、地形相対航法における位置推定を実施する際には、事前の軌道計画や IMU による姿勢のフィードバックが得られることから、地図に対する取得画像の回転やスケール変化については一定の範囲に収めることができる。したがって、式 (3.3) で示した点パターンマッチングの問題設定は、変換パラメータに制約がある中で、ワールド点パターン W の部分集合 $|W'|$ と最もよく重なり合う、未検出・誤検出点の存在しない最大のモデル点パターン M の部分集合 $|M'|$ を算出する問題となる。これは次式のように記述される。

$$\begin{aligned}
 & \text{minimize} \quad \sum_{i=1}^{|M'|} \left(f_t(M'_i, s, \theta, T) - W'_i \right)^2 \\
 & \text{maximize} \quad |M'| \\
 & \text{subject to} \quad s_{min} \leq s \leq s_{max}, \theta_{min} \leq \theta \leq \theta_{max}, \\
 & \quad \quad \quad |M'| = |W'| \leq 2
 \end{aligned} \tag{3.5}$$

ここで、 $s_{min}, s_{max}, \theta_{min}, \theta_{max}$ はそれぞれスケール変化と回転の誤差範囲である。

これらのことから、本手法では式 (3.4) における幾何学変換のパラメータを算出するた

めに必要な最小要素となる、モデル点パターン M とワールド点パターン W それぞれ 2 点からなる線分対に着目する。スケール変化や回転を完全に許す場合、これら M, W に存在する線分は、全組み合わせにおいて対応付けが可能となる。しかしながら、それらに制約が存在する場合、誤差範囲に基づいて対応付けが可能となる組み合わせが減少するために、十分高速に処理を行える期待が生じる。したがって、それら線分対を使用して、式 (3.5) に基づいて逐次的に評価することにより位置合わせを行う方策を採る。この時、地図としてワールド点パターン W からなる線分を予め計算し、データベースとして保持しておくことにより、ワールド点パターン W の線分作成にかかる処理を削減する。

この線分を用いた方策を採ることによる利点は次の通りである。

- より多角形を作成する手法よりも誤検出・未検出に関する影響を最小限に留めることができる。
- ワールド点パターンの線分を予め規定することにより、本来 $O(m^2)$ のオーダーである W 内の 2 点の選択を $O(m)$ のオーダーまで削減することができる。

手法全体の概要

本手法は図 3.2 に示すように、位置合わせ先となるデータベースを地上にて作成する前処理部と、着陸機に搭載し、実際に位置推定を行う実処理部の 2 つから構成される。

前処理部では、まず先行の周回機によって取得された数値標高モデル (Digital Elevation Model; DEM) を用いて、事前の軌道決定や着陸の日時・場所によって把握される機体姿勢や日照条件から、月表面の CG 画像を作成する。次に、CG 画像から日照変化や時間経過にロバストな特徴点であるクレータ位置を抽出する。最後に、抽出したクレータ位置から近傍点等のマッチングに不要な点を削除するなどの整備を行った上で、後述の線分情報等のリストを作成し、それらを合わせてデータベースとして作成しておく。

実処理部では、着陸機が降下中に撮影した画像からクレータ位置を抽出し、高度計や IMU といった他センサで得られる、誤差を含んだ回転・高度・位置情報を用いて、データベースと位置合わせを行うことで、月面における位置推定を行う。

また、当手法では、図 3.3 に示すような、月面におけるクレータ位置については他の検出法によって得られると仮定し、データベース (ワールド) 点パターンと取得クレータ (モデル) 点パターンを、点パターンマッチングを用いて位置合わせを行うことで、着陸機位置の推定を行う。

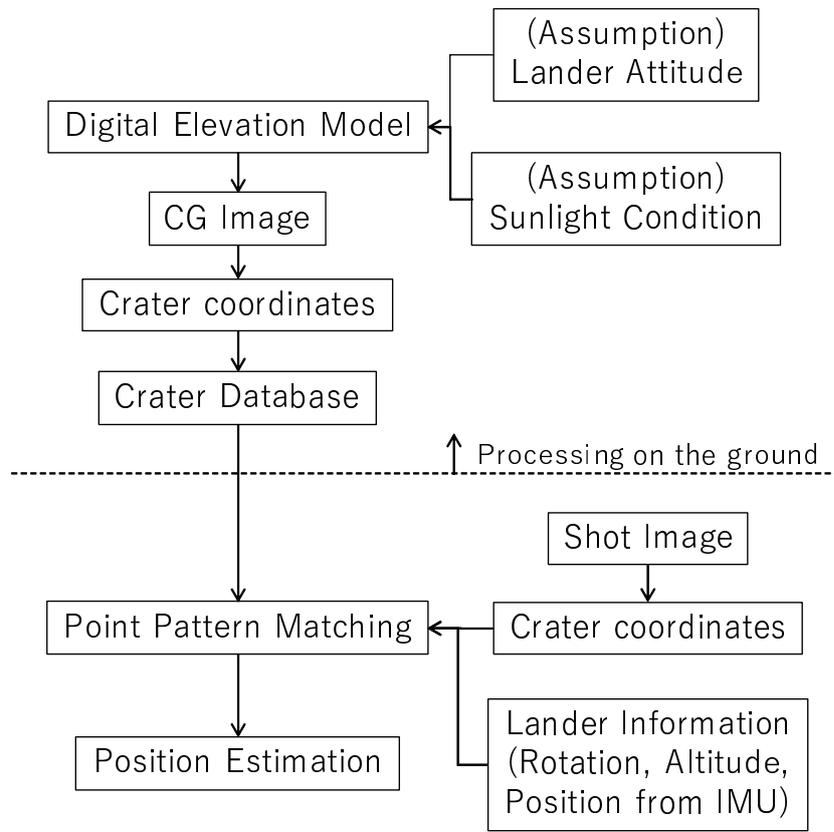


図 3.2 提案手法の位置推定の流れ

3.2.4 提案する線分情報を用いた位置推定手法

本手法では、点パターン同士の重ね合わせに必要な最小要素である線分に着目し、まず、線分情報のみを用いてデータベース範囲におけるマッチング位置の絞り込みを行い、その後、対応点の探索を行ってマッチングの確度を評価することで、上記のような条件の下でもロバストに位置推定を行う。本手法の大まかな流れは以下の通りである。

- Step 1 モデル点パターン M から線分を作成する。
- Step 2 ある線分と同等な長さ・傾きを持つデータベース上の線分リストを作成する。
- Step 3 線分同士の対応から、 M を W 上に重ね合わせ、任意の $p \in M$ に対応する q の探索を行う。
- Step 4 対応点の個数を評価値として、一定値以上のマッチングを成功とみなし、位置の計算を行う。

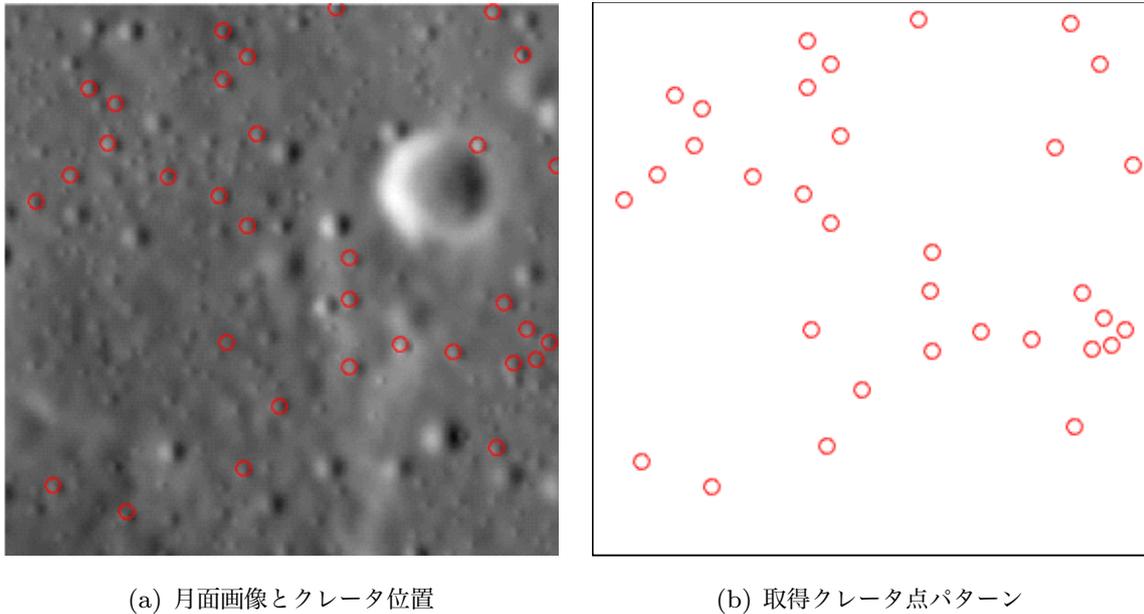


図 3.3 月クレータ点パターンの例

線分の作成とデータベースの絞り込み

まず、点パターン M から任意の 2 点を選択し、線分を作成する。2 点同士の対応によって変換 f_t が導出されるので、この線分は有向線分である。作成した線分を S としたとき、 M に含まれる全ての点の組合せからなる線分の集合 L_M は以下ようになる。

$$L_M = \{S_1, S_2, S_3, \dots, S_u\} \quad (3.6)$$

ここで、 $u = |L_M|$ である。また予め計算し、データベースとして保持している W 中の線分 S' の集合 L_W についても同様に、 $v = |L_W|$ として次のように表す。

$$L_W = \{S'_1, S'_2, S'_3, \dots, S'_v\} \quad (3.7)$$

任意の $S \in L_M$ と $S' \in L_W$ を選択したとき、これらを重ね合わせる変換 f_t は一意に求まるが、何らかの制約を与えない限りこれらは全て対応しうる。そこで、想定される誤差範囲によってこれらに制限を与える。

想定高度に対して、画像撮影時に起こり得るスケール変化の最大値、最小値を s_{max}, s_{min} 、回転の最大値を $\pm\theta_{max}$ ($|\theta_{max}| < \pi/2$) とする。ある S を選んだ際、これ

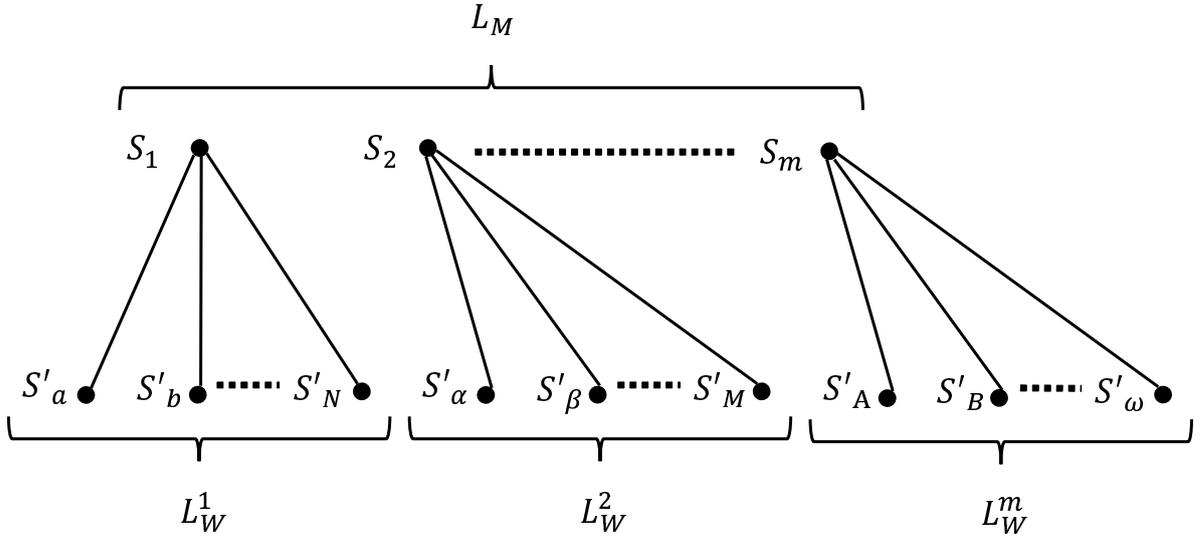


図 3.4 線分リストの対応関係

に対応しうる S' は次式によって決定される.

$$s_{min}l < l' < s_{max}l \quad (3.8)$$

$$\cos(\theta_{max}) < \cos(\phi) \quad (3.9)$$

ここで, l, l' はそれぞれ S, S' の長さ, ϕ は S, S' 間の角度を表す. ある $S_i \in L_M$ に対して, 上式 (3.8)(3.9) を満たすような全ての $S' \in L_W$ の集合を L_W^i とする. 本手法では, これらの線分の組に対して, マッチングを逐次的に評価してゆく.

L_M と L_W^i の関係を図 3.4 に示す. この図から, 本手法におけるマッチングの確認回数 N_m は次の式で表される.

$$N_m = \sum_{i=1}^m |L_W^i| \quad (3.10)$$

処理の高速化のために, 位置合わせに十分なマッチングを確認した時点で処理を打ち切る事を考えると, ある S_i と L_W^i 中の線分の重ね合わせは, 確からしいものから順に評価してゆく事が望ましい. そこで, L_W^i 中の線分に対して, IMU から得られる予想現在位置を用いて順序付けを行う. すなわち, 予想現在位置 $q_{IMU} \in \mathbb{R}^2$ から近いものから順に L_W^i 中の線分 S' を並び替え, 以降はこの順にしたがって S, S' の組を決定し, 逐次的に

マッチングの確認を行う。

探索範囲の決定

次に、線分同士の重ね合わせによって移された M と W との間でマッチングの確認を行う。ある $S_i = \{p_a, p_b\}$ と L_W^i 中のある $S'_i = \{q_{a'}, q_{b'}\}$ を式 (3.4) に代入することで、図 3.5 に示すように、有向線分同士の対応から、変換 f_t における変換のパラメータが一意に決定する。この重ね合わせを行った後に、 p に対応する q を探索する事で、マッチングを評価する。

一方で、 M には抽出時の影響として位置ずれが含まれているため、重ね合わせに用いる変換 f_t 自体に誤差が発生する。したがって、ある p_c に対応する $q_{c'}$ を調べるためには、探索範囲を設ける必要がある。

画像から取得された M 上の点 p_a, p_b には、 W 上の対応点と位相的にぴったりと重なる「真の」点 p'_a, p'_b から E だけ位置の誤差が存在すると仮定する。図 3.6 に示すように、誤差の含まれた p_a, p_b によって構成される S_{ab} を用いた変換 f_t によって、 $p_c \in M$ は、 W 上にて対応する $q_{c'}$ から離れた点 $f_t(p_c)$ へと移される。真の点 p'_a, p'_b は得られた点 p_a, p_b から E の範囲内に存在するため、 $f_t(p_c)$ を中心とした対応点 $q_{c'}$ の存在しうる範囲は、ラグランジュの未定乗数法を用いて次のように求まる。

与えられた点 p_a, p_b から距離 E に真の点 p'_a, p'_b が存在する事から、この束縛条件 g は以下のように表される。

$$g(p'_a, p'_b) \equiv d(p_a, p'_a) + d(p_b, p'_b) - 2E = 0 \quad (3.11)$$

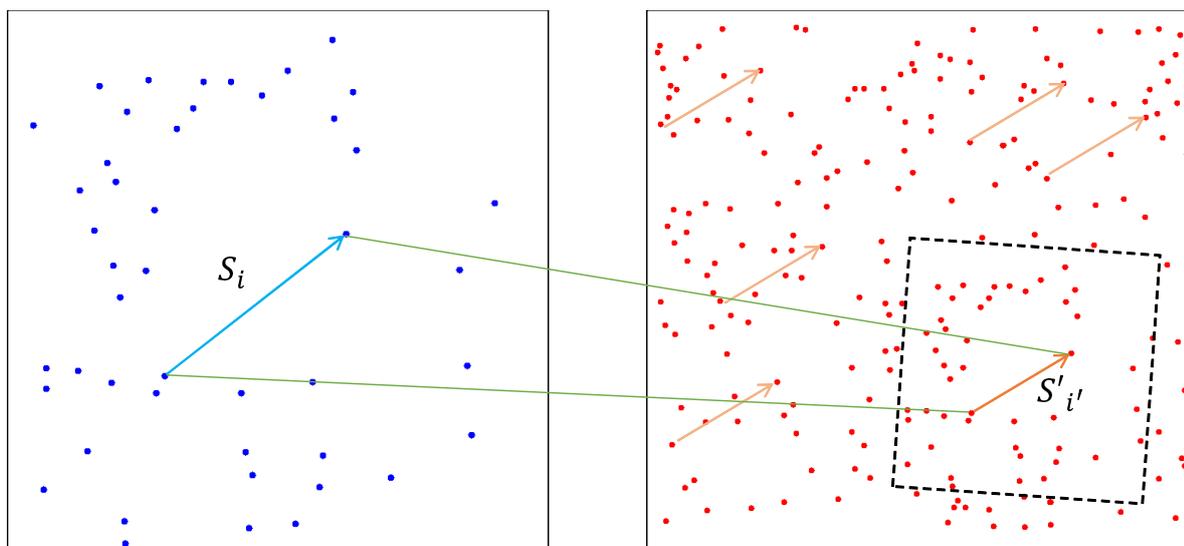


図 3.5 重ね合わせ概念図

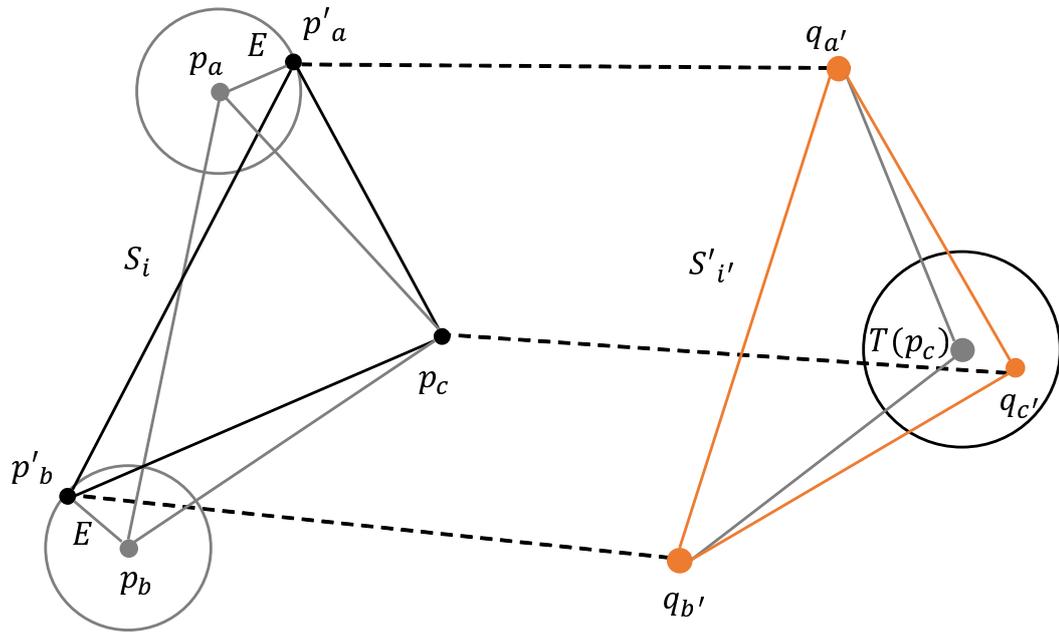


図 3.6 誤差範囲

ここで、 $d(A, B)$ は A, B 間のユークリッド距離である。これを用いて、ラグランジュ関数 \mathcal{L} は次のようになる。

$$\mathcal{L}(p'_a, p'_b, \lambda) = f'_t(p_c) + \lambda g(p'_a, p'_b) \quad (3.12)$$

ここで、 f'_t はある p'_a, p'_b と q'_a, q'_b との組で求まる変換、 λ はラグランジュの未定乗数である。

これらの式をラグランジュの未定乗数法を用いて解き、得られた p'_a, p'_b を用いて変換 f'_t を求める事で、 q'_c の存在しうる $f'_t(p_c)$ の範囲の最大・最小値を得る事ができる。実際には、点 p_c にも E だけ誤差が含まれるため、得られた範囲を E だけ広げたものが、マッチング時の探索範囲となる。

マッチングの確認

最後に、線分同士の重ね合わせ後に、 M 上の全ての点に関して、前述の探索範囲を設けて、対応する W 上の点を探索する。このとき、 f_t の誤差を減らす事を期待し、対応点を発見する毎に f_t の再計算を行う。各点には位置のずれが存在するため、3点以上の対応においてアフィン変換を用いた完全な重ね合わせを行う事は不可能である。したがって、対応点間のユークリッド距離の二乗を最小化させるような変換を最小二乗法により求める。このような再計算を繰り返す事で、バイアスの無い位置ずれに起因した変換 f_t の誤差を

低減することが可能である。

M における全ての点の対応点探索が終了した後、マッチングの評価を行う。評価は発見できた対応点の数 t によって行われ、マッチングしたと十分に判断できる、ある閾値 R を超えるか否かで判断される。本手法においては、最大の対応点を持つマッチングを発見する事は目的とせず、十分マッチングする部分を出来るだけ速く発見する事に主眼を置き、 $t \geq R$ であった場合にはマッチングを終了する。 $t < R$ であった場合には、3.2.4 節で求めた S' のリスト上にある次の線分とマッチングの確認を、 $t \geq R$ となるマッチングを発見するまで繰り返す。

マッチングが終了した後、最後に全ての対応点から変換 f_t の再計算を行い、画像中心座標をデータベース座標に移す事によって位置の推定を行う。

このようにして作成した、提案手法の実処理部におけるフローチャートを図 3.7 に示す。

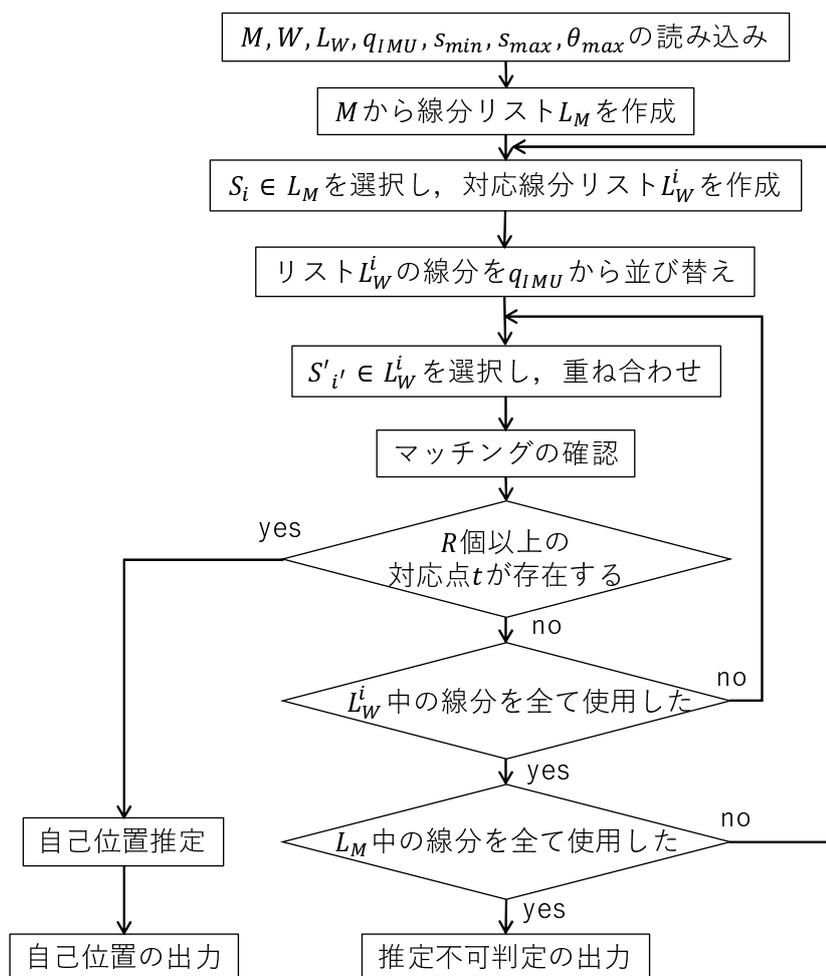


図 3.7 位置推定アルゴリズムフロー

3.2.5 アルゴリズムの検証

検証内容与方法

本項では、提案したクレータ位置によって構成される線分情報を用いた着陸機位置推定法の有効性を確認するため、月面地形画像からクレータ位置の抽出を行い、撮影画像で得られるクレータ位置を模擬することによる実験を行った。

提案した手法について、処理性能に関わるパラメータは次の通りである。

- モデル点パターン M とワールド点パターン W のサイズ
- 十分なマッチングを満たしたと判定する足切り閾値 R
- 入力となる特徴点の未検出と誤検出

当手法では入力としてクレータ位置を想定しているため、各々の点パターンのサイズは、対象となる地形に最大量は依存し、そのサイズによって処理時間は増減する。また足切りの閾値 R はワールド点パターン W における、部分集合のユニークさに依存する。未検出と誤検出は、点パターン間における対応点数を減少させる。したがって、入力となるモデル点パターン M のサイズに対して、マッチング成功を判断する対応点数の閾値 R より、低くなる量で付与された場合に位置推定は失敗する。加えて、誤検出の点は正解となるマッチングを発見するまでの処理を遅延させる。

そこで本検証では、以下の二つの検証を行う。

- 月面の地形画像から複数サイズのワールド点パターン W を作成し、その各々によるデータベースに対して誤差なしのワールド点パターン W の切り出しパターンを、入力 M として与えることにより、データベース中に存在するユニークな部分集合を構成する最小サイズ ($=R$) を決定する。この時の処理においては、足切り閾値 $R = \infty$ とすることでマッチングの全探索を実施し、最悪計算量も同時に見積もる。
- 上の検証によって得られた各データベースと足切りの閾値 R を使用して、回転、スケール変化、位置ずれ、誤検出、未検出を与えたデータセットにより、データベースのサイズに対する処理時間、精度を見積もる。

本検証において、提案アルゴリズムは C 言語にて実装し、表 3.2 に示した汎用 PC 上での実験とした。位置推定を行う誘導パス中に得られる、あるいは制限される情報を実際の地形相対航法の実施に関わる値を参考にし、それよりも大きな誤差範囲を持つよう仮定した。その諸元を表 3.3 に示す。

表 3.2 計算機実験環境

OS	Windows 10 Professional 64bit
CPU	Intel Core i9-7980XE @2.60 GHz
RAM	64.00 GB

表 3.3 前提条件の諸元

取得画像サイズ	512×512 pix
データベースサイズ	60×70 km, 2264×2641 pix
データベース画像の解像度	27.2 m/pix
s_{max}	1.15
s_{min}	0.85
θ_{max}	45 deg
E	2 pix

クレータ・線分データベースの作成

検証のため、月面地形画像から複数サイズのワールド点パターン W となる、クレータと線分のデータベースを作成する。使用した月面画像は、神酒の海周辺の 60×70 km, 2264×2641 pix となる範囲の画像である。データベース作成に用いた画像を図 3.8 に示す。

この地形画像に含まれるクレータについて、既存のクレータ検出手法 [40] を使用することにより、特徴点としてのクレータ位置を抽出した。抽出したクレータについて、地形画像範囲内において取得画像範囲 (512×512 pix) あたりの密度が、それぞれ 20,30,40,50 個となるような点パターンを 4 つ作成し、クレータデータベースとした。それぞれのデータベースについて、図 3.9 に示す。

続いて、線分の集合 L_W の作成を行った。密度 30 のデータベース点パターンから作成できる線分の分布は図 3.10 となる。この時、組合せ数の削減を考え、使用する線分の長さに制限を与える。実際には、撮影画像の範囲によって使用する線分の長さは制限されるため、データベース範囲内で 512 pix 四方の窓を設け、その窓の中に存在するクレータで作成される線分の平均分布を確認した。これを図 3.11 に示す。これを見ると、長さが 250 pix 付近でピークを迎えて両端に向かって減少している。

使用する線分の長さはマッチングの確認を実施する回数に影響し、アルゴリズム全体の計算量を決定する値となる。モデル点パターン M とワールド点パターン W 全ての線分の対応を確認する場合、その計算量オーダーは $O(m^2n^2)$ となる。本検証では、マッチング

に十分かつ計算量オーダを低減させるため、これらのデータベース密度において画像中の線分数が m のオーダ、データベース中の線分数が n のオーダとなるような値として、使用する線分の長さについて、75-125 pix となるように制限を与えた。

また、各線分に対して長さ、中点座標、水平方向成分、垂直方向成分を予め計算し、リスト化することで式 (3.8),(3.9) と順序付けの計算時間の短縮を図った。このようにして整備を行ったデータベースの諸元を表 3.4 に示す。ここで示されるように、各密度のデータベースに記載されている線分数は n のオーダに従っている。このことから、提案アルゴリズム全体の最悪計算量は、モデル点パターン M とワールド点パターン W で使用される全ての線分の組み合わせと、その組み合わせ毎に高々 m 回のマッチングの確認が行われるため、本検証のような月面での一般的な地形画像とクレータ密度の条件の下においては、 $O(m^2n)$ となる。

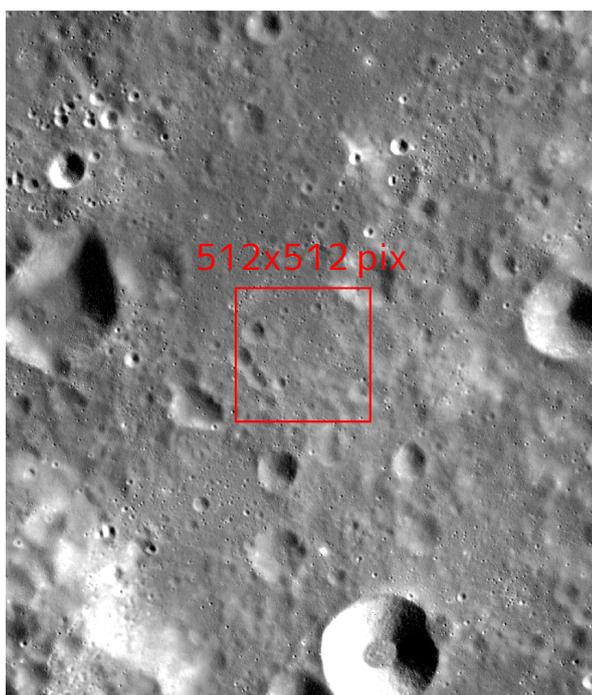
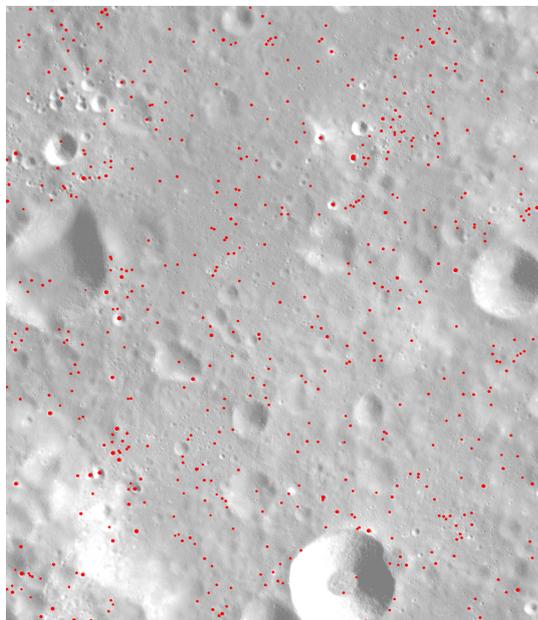


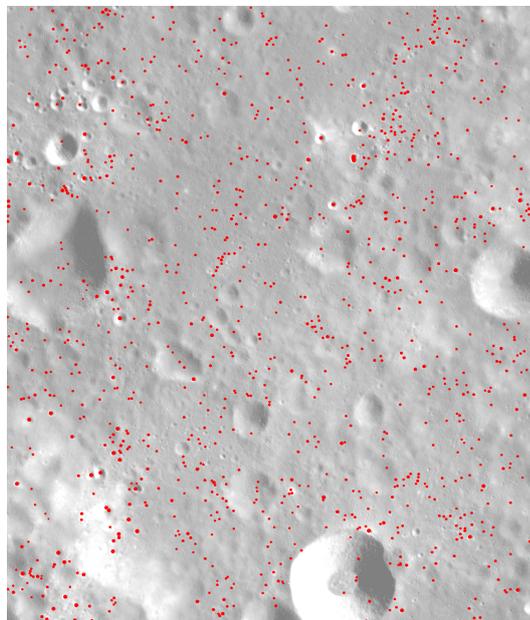
図 3.8 月面地形画像

表 3.4 データベース記載情報の諸元

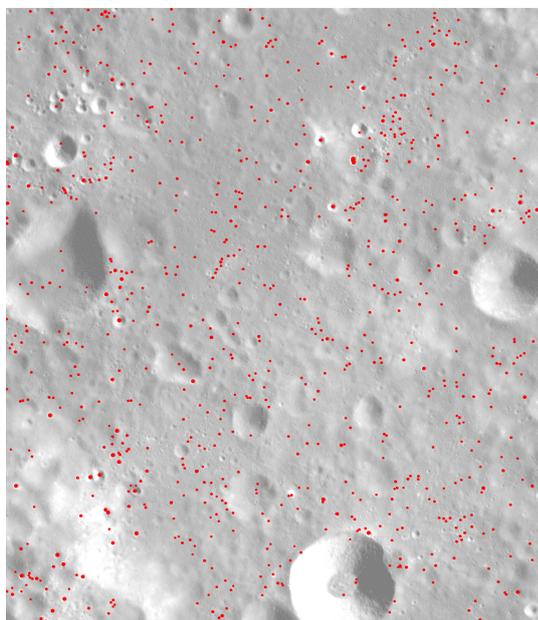
	密度 20	密度 30	密度 40	密度 50
ワールド点パターン数 (n)	536	786	1045	1292
総線分数 (p)	815	1702	3013	4610



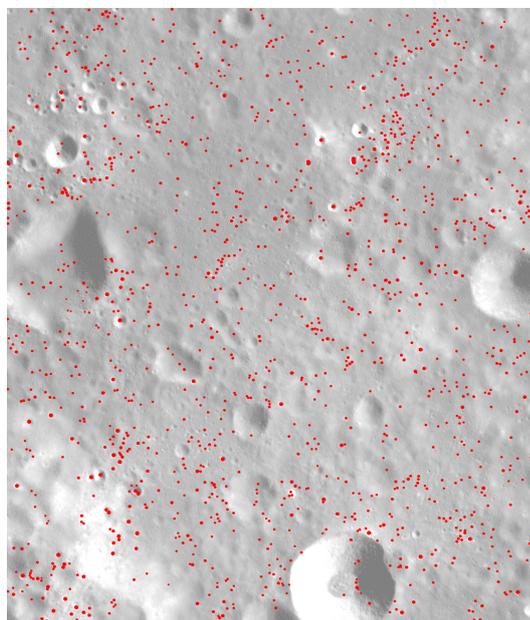
(a) 密度 20 のデータベース



(c) 密度 40 のデータベース



(b) 密度 30 のデータベース



(d) 密度 50 のデータベース

図 3.9 作成したクレータデータベース，赤点が記載クレータ

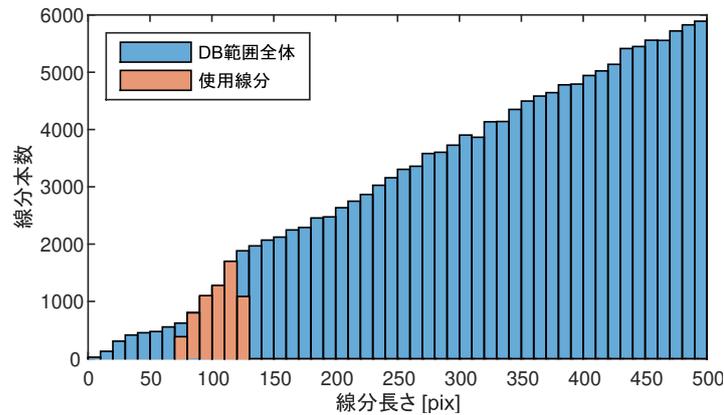


図 3.10 データベース中の線分分布と使用線分範囲

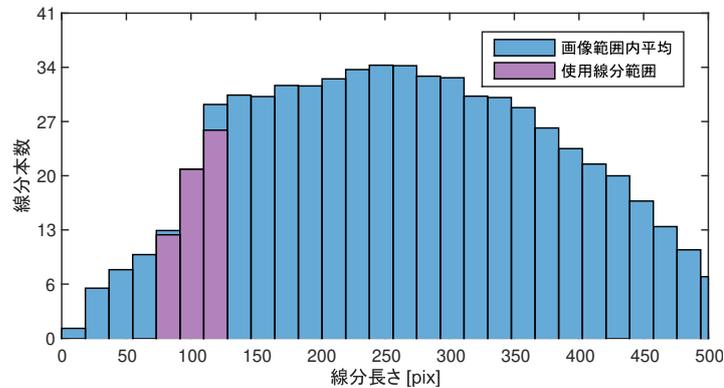


図 3.11 データベース画像範囲中の平均線分分布と使用線分範囲

足切りの閾値と最悪計算量の検証

本検証では、前項で作成した複数の密度を持つデータベースについて、データベースに記載されているクレータを切り出し、誤差を与えることなく、それを入力として提案手法にてマッチングを行うことにより、データベースに含まれる最小の $|M'| = |W'|$ を持つユニークな部分集合 $|W'|$ を同定する。それによって、データベース中でマッチングが一意に決定される最小の $|M'|$ となる、足切り閾値 R を決定する。

また、この検証時においては足切り閾値 $R = \infty$ とし、データベースから切り出したテストセットと、データベース内全ての線分の組み合わせについてマッチングを評価することにより、各クレータ密度における最悪計算量を測定する。

上述のように、異なるクレータ密度をもつデータベースそれぞれにおいて、ランダムな位置から画像範囲の大きさだけ切り出した、誤差の乗らないクレータパターンを 10000

ケース作成した。そのテストセットを入力としてマッチングを行ったとき、対応点数 $|M'|$ に対し、その対応点数をもつ部分集合 $|M'|$ が W においてマッチングした個数、すなわち、ある $|M'| = 2$ を選択したときに対応しうる W' の数を $|M'| = 3, 4, \dots$ と続けたときの、全てのケースにおける平均と分散を箱ひげ図として記したものを、図 3.12 に示す。

図 3.12 から読み取れるように、対応点数が 2 個のときはデータベース内のほぼ全ての線分と一致するため、対応しうる箇所が最も多い。対応点数が増加するにつれて、それら M' の空間上の位置関係はユニークなものとなるために、対応しうる W' の箇所が少なくなる。

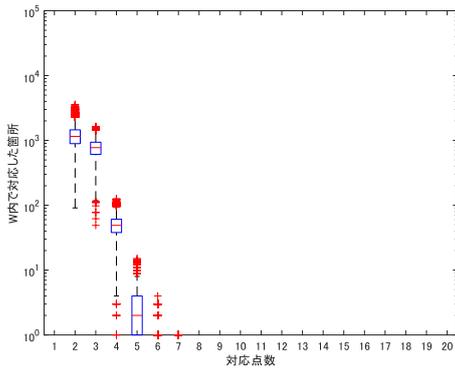
この検証により、各々の密度をもつデータベース中に唯一存在する部分集合の最小サイズは表 3.5 のようになった。これは、あくまで最小の閾値であるために以降の検証では誤検出等の誤差を鑑みて、それらを微増させた足切り閾値を使用する。

またこの時の位置合わせにおける精度と処理速度について、表 3.6 に示している。本検証では入力に対して誤差を与えていないため、それぞれのデータベースに対する入力 10000 ケース全てにおいて位置とスケール、回転のパラメータが推定できている。水平位置について最大で 1pix 程度の誤差が存在している理由としては、入力となる点群の切り出しにおいて、サブピクセル単位の位置を使用しており、その丸め誤差が存在することと、処理において高速化のために小数点の丸めを行っており、その誤差が蓄積したことによるものと考えられる。

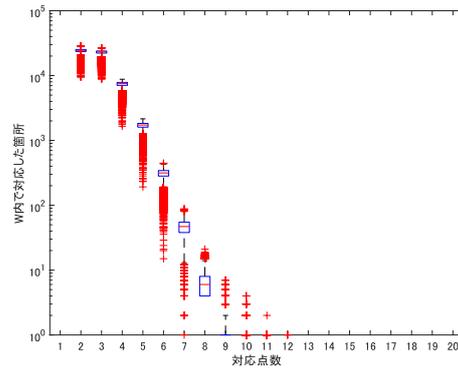
最悪計算時間については、クレータ密度によってマッチング時の組み合わせ数が増えるため、その傾向が顕著に現れている。

表 3.5 各データベースにおける閾値 R

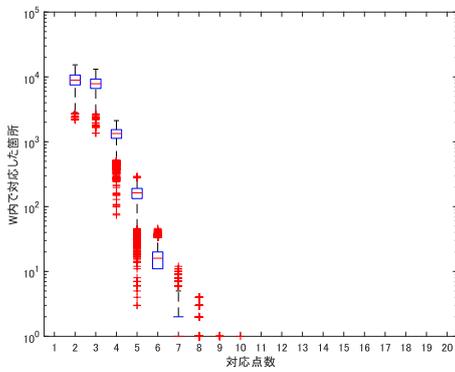
	密度 20	密度 30	密度 40	密度 50
最小の足切り閾値	8	11	13	15
検証で使用する足切り閾値 (R)	9	12	14	16



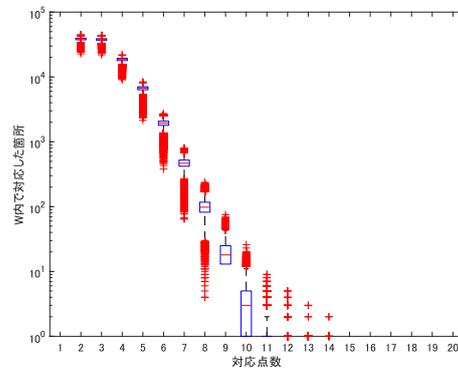
(a) 密度 20 のデータベースにおける，対応点数毎の一致数



(c) 密度 40 のデータベースにおける，対応点数毎の一致数



(b) 密度 30 のデータベースにおける，対応点数毎の一致数



(d) 密度 50 のデータベースにおける，対応点数毎の一致数

図 3.12 各密度のデータベースにおける， $|M'|$ に対して一致する箇所の数

表 3.6 データベース切り出しの場合の位置推定精度と計算時間

密度	x 方向誤差 [pix]		y 方向誤差 [pix]		スケール誤差 [%]		回転誤差 [deg]		処理時間 [ms]				
	平均	最大	平均	最大	平均	最大	平均	最大	平均	最大			
20	0.00	1.23	0.02	0.98	0.00	0.01	0.20	0.03	0.00	0.31	9.86	3.98	30.0
30	-0.02	1.12	0.04	1.00	0.00	0.01	0.11	0.03	0.00	0.31	116.12	33.17	221.0
40	-0.01	1.00	0.03	1.08	0.00	0.01	0.09	0.04	0.00	0.31	434.72	56.63	565.0
50	-0.01	1.08	0.05	1.17	0.00	0.01	0.13	0.06	0.00	0.31	977.16	75.50	1268.0

誤差に対する精度と計算量の検証

本検証では、想定しうる誤差に対する位置合わせの精度と計算量への影響について評価する。地形相対航法における位置推定を実施するにあたって、事前に知り得ない誤差として、入力となる点パターンに未検出となる点と誤検出を起こした点が発生することが挙げられる。未検出点はデータベースに存在している点を検出できないことにより、対応点が減少するために精度低下やマッチングの未了を引き起こす。誤検出点はデータベースに存在していない点を検出することにより、新たな対応点を増加させるために処理速度の増加や誤マッチングを引き起こす。

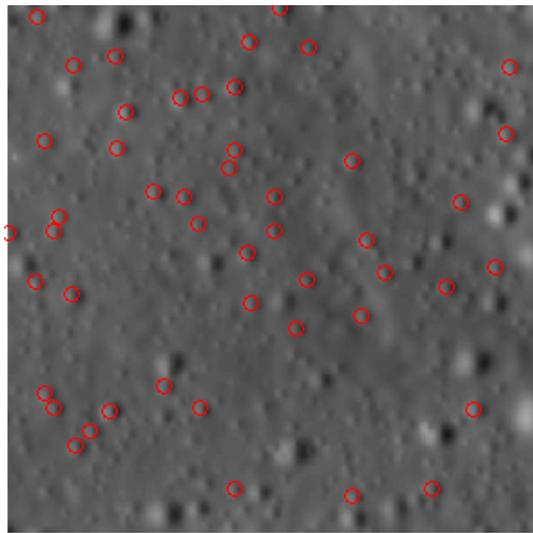
これら誤差による性能への影響を調べるため、入力となる点パターンに対して未検出点と誤検出点をそれぞれ 10,20,30% ずつ付与し、位置推定を行った。その他の誤差については、表 3.3 に示した条件の下ランダムに割り振り、また計算打ち切りの閾値 R については、前項で算出したものを各データベースにおいて使用している。また、探索範囲の初期値を決定する予想現在位置 q_{IMU} についても、位置の真値に関わらずランダムに割り振った。

具体的なテスト用の入力点パターンの生成方法は次の通りである。まず、データベース点パターンの任意の位置において 512×512 pix の窓を設け、回転、スケール変化の想定誤差を与える。その窓中に存在する点パターンに対して、位置ずれの付与をランダムに与えることにより、テスト用のクレータ点パターンを取得する。その誤差ありパターンについて、10,20,30% の点を削除することにより未検出を、10,20,30% の点を異なる位置の点に変更することにより誤検出を模擬する。それぞれ誤差の量が最大 30% である理由としては、データベースの密度に対して、合わせて最大 60% より多くの点を変更すると、データベースに存在し、対応する真の点パターンの数が閾値 R を割ってしまい、無条件でマッチングが未了となるからである。

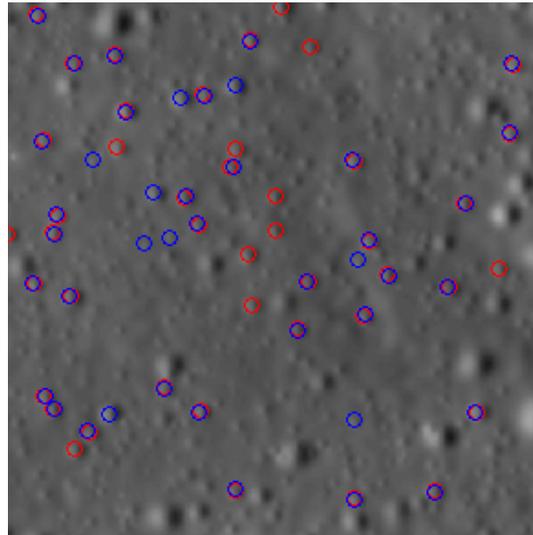
ここで模擬した点パターンの例を図 3.13 に示す。データベース点パターンに対して、模擬点パターンには誤検出、未検出、位置ずれが付与されている事がわかる。この模擬点パターンを、各密度のデータベース毎、3段階の未検出・誤検出量毎に 10000 ケース生成した。

未検出・誤検出が 10% 存在するときの位置推定結果を表 3.7 に、未検出・誤検出が 20% 存在するときの位置推定結果を表 3.8 に、未検出・誤検出が 30% 存在するときの位置推定結果を表 3.9 にそれぞれ示す。

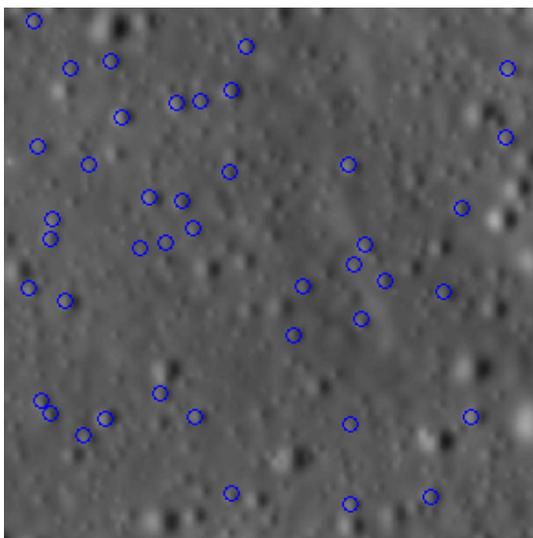
まず、それぞれ誤差が 10% のときの結果を見てみると、全てのデータベースかつ全てのテストセットで位置推定に成功している。精度については、表 3.6 に示した誤差がないものと比較して、画像の x, y 方向誤差の最大値が 1pix 程度増加している。これは前述した小数点の丸め誤差に加えて、点の位置ずれの誤差に起因するものである。各点において



(a) 模擬点パターン



(c) (a)(b) の重ね合わせ



(b) 同等の位置のデータベース点パターン

図 3.13 模擬点パターンの一例

2pix のずれを与えているため対応点全てにおいて同じ方向にずれが発生した場合、最大で 2pix の位置推定誤差が生じることになる。また、スケールの誤差について、密度 20 のデータベースのみが最大で 3% 程度の誤差が生じている。これはデータベースの密度が小さく対応点が少ないために、画像内で対応点が広範に分布せず固まっている場合にて、スケールのずれが過大に評価されることに起因するものである。他の密度においては、画像の x, y 方向誤差以外に大きな誤差が生じているものはない。また、足切り閾値を設けたことにより、最大でも前項で示した最悪計算量の半分程度の時間で処理が行えている。

続いて、それぞれ誤差が 20% のときの結果に関しても、誤差が 10% の結果と比べて変

わりない精度で推定が行えている。処理時間においては、未検出点や誤検出点の影響によって数割程度遅れる結果となっている。一方で、密度 20 のデータベースにおいて、3 ケース位置推定に失敗した。これらの失敗は全てデータベース内の他の領域と誤マッチングをしたものである。密度 20 のデータベースにおける閾値 R は 9 であり、図 3.12(d) で示したように、地図画像領域内には対応点数 9 で誤マッチングを起こす場所が一定数存在する。したがってデータベースに記載されていないクレータが誤検出されることにより、データベース内の他の位置でのマッチングが発生した。

未検出・誤検出が 30% 存在するときの結果では、密度 20 のデータベースは全てのケースで位置推定が行えていない。これは、入力点パターンの中で正解となる点が 4 割、つまり 8 個しか存在しないため、全てのケースにおいて足切り閾値を超えることがないためである。また、密度 30 のケースにおいても正解となる点が減ったために、未検出・誤検出が 10,20% 存在するときの密度 20 のものと同様に、スケール変化の推定誤差が大きくなっている。

表 3.7 未検出・誤検出が 10% の場合の位置推定精度と計算時間

密度	成功率 [%]	x 方向誤差 [pix]			y 方向誤差 [pix]			スケール誤差 [%]			回転誤差 [deg]			処理時間 [ms]		
		平均	3σ	最大	平均	3σ	最大	平均	3σ	最大	平均	3σ	最大	平均	3σ	最大
20	100	-0.48	0.59	2.93	0.56	0.58	2.26	1.61	1.70	3.32	0.00	0.14	1.00	1.72	1.67	14.0
30	100	-0.51	0.58	2.15	0.56	0.58	2.36	0.07	0.34	1.57	0.00	0.14	1.55	9.10	8.93	94.0
40	100	-0.49	0.59	2.61	0.56	0.58	2.56	-0.32	0.25	1.52	0.00	0.26	2.30	30.24	30.39	250.0
50	100	-0.49	0.59	2.80	0.57	0.59	2.85	-0.12	0.32	1.45	-0.01	0.24	2.22	77.24	76.45	807.0

表 3.8 未検出・誤検出が 20% の場合の位置推定精度と計算時間

密度	成功率 [%]	x 方向誤差 [pix]			y 方向誤差 [pix]			スケール誤差 [%]			回転誤差 [deg]			処理時間 [ms]		
		平均	3σ	最大	平均	3σ	最大	平均	3σ	最大	平均	3σ	最大	平均	3σ	最大
20	99.9	-0.48	0.60	2.65	0.56	0.59	2.62	2.46	2.46	4.22	0.00	0.15	0.45	2.10	1.92	17.0
30	100	-0.50	0.58	2.84	0.56	0.58	2.98	0.17	0.32	1.58	0.00	0.19	3.02	11.20	10.64	131.0
40	100	-0.49	0.60	2.77	0.56	0.59	2.97	-0.15	0.26	1.62	0.00	0.28	2.90	37.04	35.11	343.0
50	100	-0.49	0.60	2.88	0.57	0.59	2.96	-0.10	0.30	1.44	0.00	0.26	2.82	93.38	87.78	828.0

表 3.9 未検出・誤検出が 30% の場合の位置推定精度と計算時間

密度	成功率 [%]	x 方向誤差 [pix]			y 方向誤差 [pix]			スケール誤差 [%]			回転誤差 [deg]			処理時間 [ms]		
		平均	3σ	最大	平均	3σ	最大	平均	3σ	最大	平均	3σ	最大	平均	3σ	最大
20	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30	100	-0.50	0.59	2.92	0.56	0.58	2.36	0.59	0.96	2.53	0.00	0.19	2.90	14.12	12.66	116.0
40	100	-0.49	0.59	2.82	0.55	0.59	2.98	0.01	0.26	1.52	0.00	0.20	4.20	46.13	42.21	373.0
50	100	-0.49	0.60	2.93	0.56	0.59	2.91	0.03	0.32	1.68	0.00	0.28	3.62	115.61	103.37	1076.0

検証のまとめと考察

本検証では、提案手法の性能を評価するために、実際の月面画像を用いて、異なる密度をもつデータベースを作成した。この時、データベースとして使用する線分の長さを絞ることにより、入力点パターンとデータベース内の線分間の対応付けにおける計算量の削減を図った。提案手法における最悪計算量は、入力となるモデル点パターンとデータベースであるワールド点パターンのサイズによって決定される。当手法ではクレータを特徴点として使用するために、これらのサイズは対象となる地形に存在する、クレータの密度に依存する。本検証で想定した月面地形のように、取得画像範囲内のクレータ密度が高々 50 個程度であるとき、線分間の対応付けにおける計算量は、 $O(mn)$ まで抑えられる。

このように作成したデータベースについて、データベース内のクレータを切り出し、誤差を与えずに手法の入力として与えることで、データベースに存在するユニークな部分集合の最低サイズを求め、線分の対応を取る際の足切り値を同定した。また、その際に線分の対応について全探索を実施することで、密度別の各データベースにおける最悪処理時間を算出した。

これら作成したデータベースと、同定した足切りパラメータを使用することにより、想定しうる誤差を付与した入力点パターンに対する各データベース、誤差率別の性能評価を実施した。結果としては、クレータ密度が 30,40,50 の場合において、精度に問題なく提案手法が位置推定を実施可能であることを示した。クレータの密度においては、小さいほど高速に処理が可能である一方で、大きいほど誤差にロバストとなる。この選択については、対象となる地形やクレータの検出性能を鑑みて、誤差範囲を規定することで決定されるべきである。検証においては未検出・誤検出が最大 30% であるとしたが、より誤差が大きい場合には、対象地形のクレータの密度がより高い地形である必要がある。

また、処理時間に関しては、足切りを設けることにより正解が存在すれば最悪処理時間のおよそ半分程度の時間で位置推定が可能となる。これは、探索範囲の初期位置を決定する予想現在位置 q_{IMU} によって高速化が可能である。なぜならば、信頼性の高い初期位置を与えることは、データベースの探索範囲を狭めることになるからである。

これらのことから、クレータが画像範囲内で数十個程度の密度であり、また誤差が一般的な地形相対航法で考えられる範囲であるとき、当手法は精度を損なうことなく高速に位置推定が可能であるといえる。その最悪計算量は $O(m^2n)$ であり、表 3.1 に示した一般的な手法と比較しても、十分に高速である。

3.3 動きブレ画像による速度推定手法

3.3.1 地形相対航法における速度推定

着陸機の速度推定手法において最も一般的な方法は、レーダによるものである。レーダを用いた手法においては、ドップラ周波数から速度を推定する方式が一般的であり、NASA の MSL ミッション等で採用されている [53]、JAXA の SLIM ミッションにおいても、着陸レーダの搭載検討が行われている [54, 55]。また、レーザ光の往復時間を測る LIDAR を用いた手法は、分解能の高さから、自動車等の産業分野での研究が行われている。

これらアクティブセンサによる手法は、質量・電力リソース面におけるコストが高く、また着陸直前の低高度（～10 m）では、測定精度に制約がある場合が多い。一方で、そのような手法と比較して、リソース面で低コストであるイメージセンサを用い、移動物体を追跡しその速度を推定する手法は、民生分野において多数提案されている。そのような画像による速度推定手法においては、画像フレーム間に存在する共通した特徴を抽出し、位置合わせを行うことによって撮影間の時間に生じた移動量を推定することで、速度を推定する。

このように画像を複数枚使用する手法では、特徴抽出を複数回行う必要があり、また宇宙機においては画像取得そのものに時間がかかるという難点がある。地形相対航法における速度推定の適用例としては、MER の DIMES [23] が存在する。DIMES では、複数フレーム間において重複する特徴点を追跡することによって速度の推定を行っている。しかしながら、その処理時間は約 15 秒となっており、リアルタイム性に乏しい。

そこで、単一画像から速度情報を得る事について考える。着陸機が地表面に対して一方方向に高速で移動することを鑑みたとき、短い露光時間においても画像中に着陸機の動きによるブレ（motion blur）が発生し、劣化を引き起こすといった問題が存在する。このような劣化が生じると、画像に含まれる情報が不確かなものとなるため、リモートセンシング等の画像を扱う分野において動きブレは避けるべき現象である。一方で、ブレ画像にはその劣化を引き起こした移動情報が含まれており、その情報は劣化関数に保存されている。露光時間がごく短い場合、劣化画像内のブレの量はカメラと物体間の相対的な移動量と線形である事から、ブレ量とカメラと被写物の位置関係を推定する事で、幾何的に撮影時の速度を求める事が可能である。

したがって、着陸機が月惑星表面への降下パスにある中で撮影した、地表面に対する一次元ブレを含む単一の画像から、露光時間中の移動量を算出することによって、速度を推定することが可能であると目される。一般的なシーンにおいては複数枚の画像から速度推

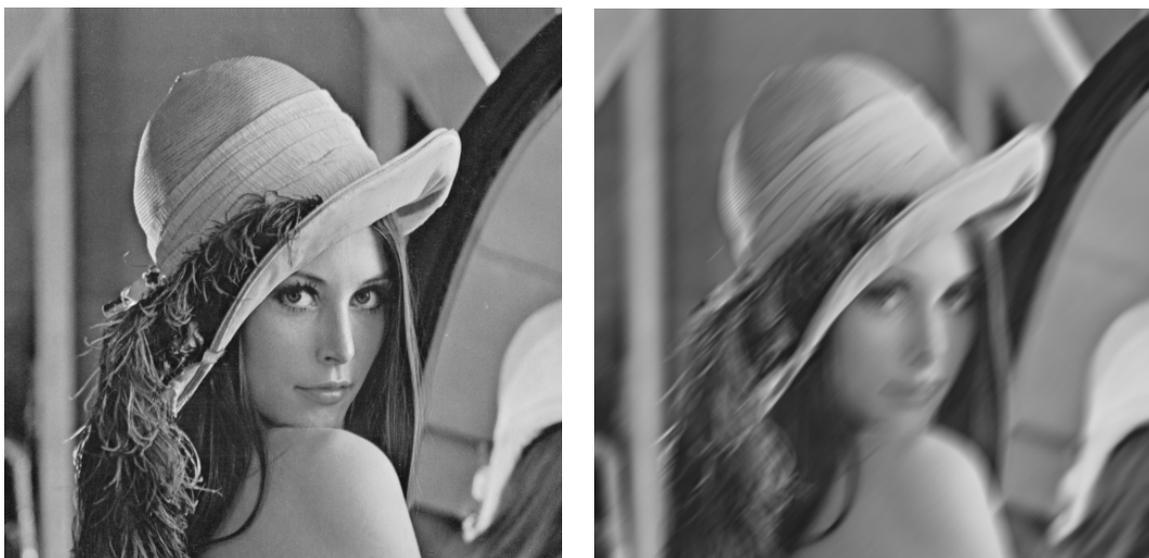
定を実施することが主流であり、特に自然画像の画面全体に渡る動きブレから移動体の速度を推定する手法はほぼ無く、またそのアルゴリズムを FPGA のような組み込みの計算機で処理する例は調べた限りでは存在しない。そこで本節では、地形相対航法における自律的な速度推定を限られた計算機環境において実施するために、月惑星表面を撮影した一次元の動きブレを含む単一画像から、露光時間中の速度を推定する手法を提案する。また、その手法について適用可能な動きブレの範囲を評価し、加えて実際の宇宙機用計算機に対する実装性を評価する。

3.3.2 ブレによる画像の劣化

動きブレとは、図 3.14 に示すような、イメージセンサの露光時間中に、被写物とカメラとの相対的な運動によって引き起こされる空間的な画像劣化の事をいう。

このような劣化のプロセスにおいて、画像内の一部分のみが移動する場合、劣化はその位置に依存して発生する（被写体ブレ）。一方、画像全体が一様に移動するような場合には、像は平行移動を除いて位置に依存しない。その劣化画像 $g(x, y)$ は、平行運動以外にセンサ上に投影される風景の像が時間的に不変であり、ノイズを加法的であると仮定すると以下のように表される。

$$g(x, y) = f(x, y) \otimes h(x, y) + n(x, y) \quad (3.13)$$



(a) ブレの無い原画像

(b) 動きブレのある画像

図 3.14 露光時間中に斜め方向に運動したブレ画像

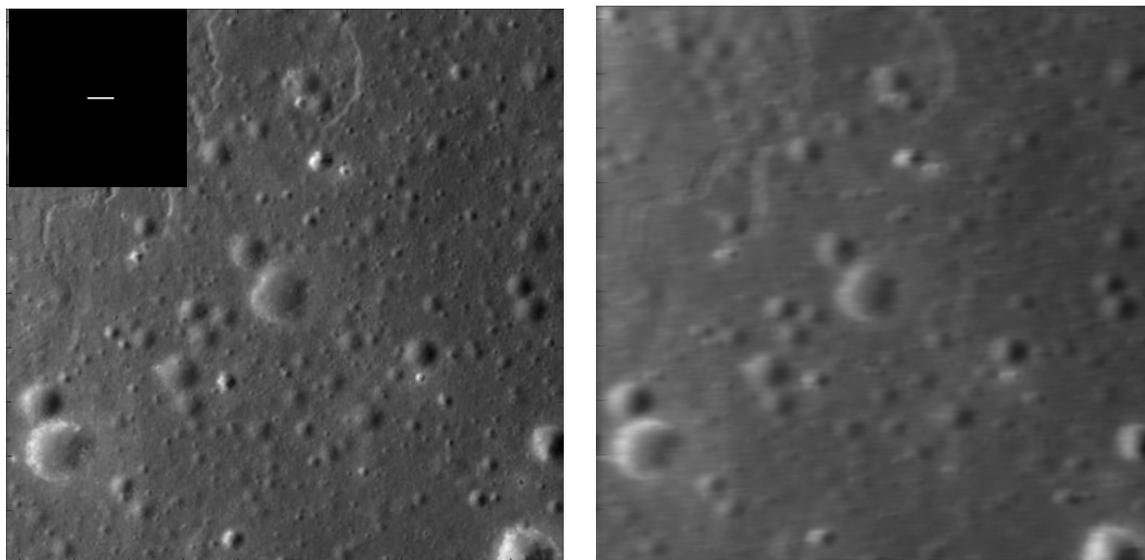
ここで $f(x, y)$ はブレを含まない理想的な原画像分布， \otimes は畳み込み積分を表す演算子， $h(x, y)$ は劣化関数である PSF，また $n(x, y)$ は光学系に起因するノイズやバックグラウンドノイズを含むノイズ項である。

この時，PSF の様相は露光時間中の被写物の移動の軌跡によって決定され，長時間露光時などには二次元となる場合もある。しかし，一般的な着陸機を想定すると，その降下フェーズにおいては数十ミリ秒といった短い露光時間中に地表面に対して一方向に直線的に動いているため，このブレの軌跡は等速直線運動で近似する事ができる。そのような場合， $h(x, y)$ は次のようにモデル化される。

$$h(x, y) = \begin{cases} \frac{1}{L} & (|x| \leq \frac{L}{2} \cos \vartheta, y = x \tan \vartheta) \\ 0 & (\text{otherwise}) \end{cases} \quad (3.14)$$

ここで L と ϑ はそれぞれ像面におけるブレの長さや方向である。すなわち，PSF は像面における被写体の移動量の長さをもつ矩形関数となる。

この劣化過程を図 3.15 に示す。このような一次元的なブレの長さ L は，被写物の画像上における移動量と比例の関係にある。したがって，このブレ量 L ，カメラと被写物との位置関係が把握できれば，カメラ，すなわち宇宙機と被写物である地表との相対的な速度を幾何的に求める事が可能である。



(a) PSF(左上) と原画像

(b) 劣化画像

図 3.15 動きブレによる劣化過程

パワーケプストラムによるブレ量の推定法

ブレ画像から PSF を推定し、ブレを含まない原画像を復元する手法はブラインド・デコンボリューションと呼ばれ、汎用カメラにおける手ぶれ画像の補正技術等として広く研究されている。本研究では宇宙機においては撮影時に複雑なブレが発生せず、直線的でかつ方向も既知であるとの仮定をおき、故にオンボード計算が可能なシンプルな手法として、パワーケプストラムを用いた PSF の推定法を検討する。

画像におけるパワーケプストラムとは、パワースペクトルの対数を取ったものに逆フーリエ変換を行う事で得られる、スペクトルのスペクトルともいえる領域の量を指す。パワーケプストラムには、畳み込みで表現されていた式 (3.13) の関係が、この変換を経る事により、和の形式で表すことができる利点がある [56]。本項では、その領域で現れる特徴を利用する事による、PSF に含まれるブレ量の推定法を導入する。

式 (3.13) に示した劣化画像 $g(x, y)$ のパワースペクトル $\mathcal{S}_g(u, v)$ は、 $h(x, y)$ が決定的関数かつ $f(x, y)$ と $n(x, y)$ がそれぞれ独立であるとする、以下のように表す事が出来る。

$$\mathcal{S}_g(u, v) = \mathcal{S}_f(u, v)|H(u, v)|^2 + \mathcal{S}_n(u, v) \quad (3.15)$$

ここで $\mathcal{S}_f(u, v)$, $\mathcal{S}_n(u, v)$ はそれぞれ $f(x, y)$ と $n(x, y)$ のパワースペクトル、 $H(u, v)$ は $h(x, y)$ の二次元フーリエ変換で、 $f(x, y)$ を $g(x, y)$ に変換するシステムの伝達関数である。

この伝達関数 $H(u, v)$ は、 $\omega = \sqrt{x \cos \vartheta + y \sin \vartheta}$ と置くと次のように与えられる。

$$\begin{aligned} H(u, v) &= \mathcal{F}[h(x, y)] \\ &= \frac{\sin \pi L \omega}{\pi L \omega} \\ &= \text{sinc}(L \omega) \end{aligned} \quad (3.16)$$

ここで、 \mathcal{F} はフーリエ変換を表す。式 (3.16) から分かるように、この関数は $\omega = \pm \frac{1}{L}, \frac{2}{L}, \dots$ において周期的な零点をもつ sinc 関数となる。

続いて、ある関数 $img(x, y)$ のパワーケプストラム $\mathcal{P}_{img}(i, j)$ は、そのパワースペクトル $\mathcal{S}_{img}(u, v)$ を用いると次式のように定義される。

$$\mathcal{P}_{img}(i, j) = \mathcal{F}^{-1}[\log |\mathcal{S}_{img}(u, v)|] \quad (3.17)$$

これより、劣化画像におけるパワーケプストラム $\mathcal{P}_g(i, j)$ は、ノイズがごく小さいとす

ると次式のように表される.

$$\begin{aligned}
 \mathcal{P}_g(i, j) &= \mathcal{F}^{-1}[\log |\mathcal{S}_f(u, v)| H(u, v)|^2 + \mathcal{S}_n(u, v)] \\
 &\approx \mathcal{F}^{-1}[\log |\mathcal{S}_f(u, v)| + 2 \log |H(u, v)|] \\
 &\approx \mathcal{P}_f(i, j) + 2\mathcal{C}_h(i, j)
 \end{aligned} \tag{3.18}$$

ここで $\mathcal{P}_f(i, j)$ は $f(x, y)$ のパワーケプストラム, 第二項は $h(x, y)$ のケプストラムである.

式 (3.18) で示したように, 劣化画像はパワーケプストラムを取った際, $\mathcal{P}_f(i, j)$ と $\mathcal{C}_h(i, j)$ との総和で表現できる. 一般的に, $\mathcal{P}_f(i, j)$ は強い DC 成分を持った後, 高ケプレンシ^{*1} へ小さな値を持つ. 一方, $\mathcal{C}_h(i, j)$ は, $H(u, v)$ で表れた周期的な零値の影響で, 大きな負の極小値がブレ量に応じた周期で生じる.

図 3.14 で示した女性の画像を原画像とし, ブレ幅 $L = 16$ である PSF, 原画像とその PSF を畳み込むことで作成した劣化画像のそれぞれについて, ケプストラム領域に変換し, ブレ方向におけるそれぞれのパワーケプストラムを重ね合わせたものを図 3.16 に示す.

この図に示すように, 劣化画像 $g(x, y)$ のパワーケプストラムにおいて, $\mathcal{C}_h(i, j)$ の概形が強く残っているため, ケプストラム領域においてこの極小値を探索する事で, ブレ幅 L

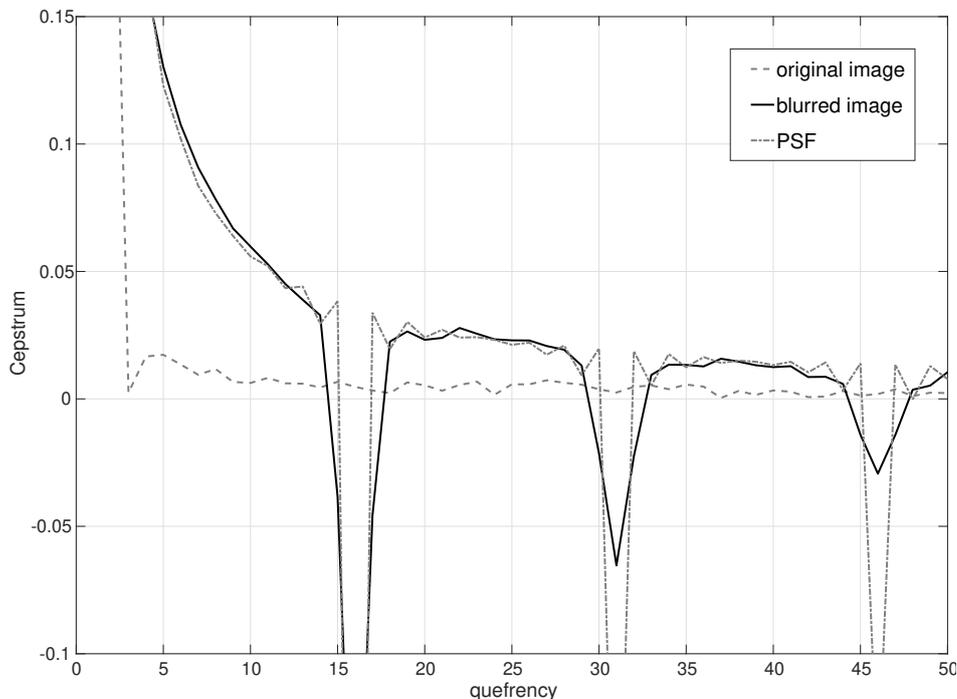


図 3.16 各パワーケプストラムの概形

3.3 動きブレ画像による速度推定手法 3 現状の低計算リソース環境における地形相対航法

の推定を行う事が可能である。本手法では、極小値を探索するために、ケプストラム領域において原点からブレ方向に対して勾配をとり、勾配がゼロかつケプストラムが負である点を極小値とする。

3.3.3 速度の算出法

画像中におけるブレ量から速度を推定するために、ピンホールカメラモデルについて考える。露光時間中の被写物の変位と画像面との関係は図 3.17 のようになる。ここで、被写物の移動量は d 、カメラと被写物との角度は ϕ であり、また画像面上での L は前項で述べたように像面での移動量である。従ってこれらの幾何学的関係から、以下の 2 式を得ることができる。

$$\frac{c}{bS_p} = \frac{d \sin \phi}{f} \quad (3.19)$$

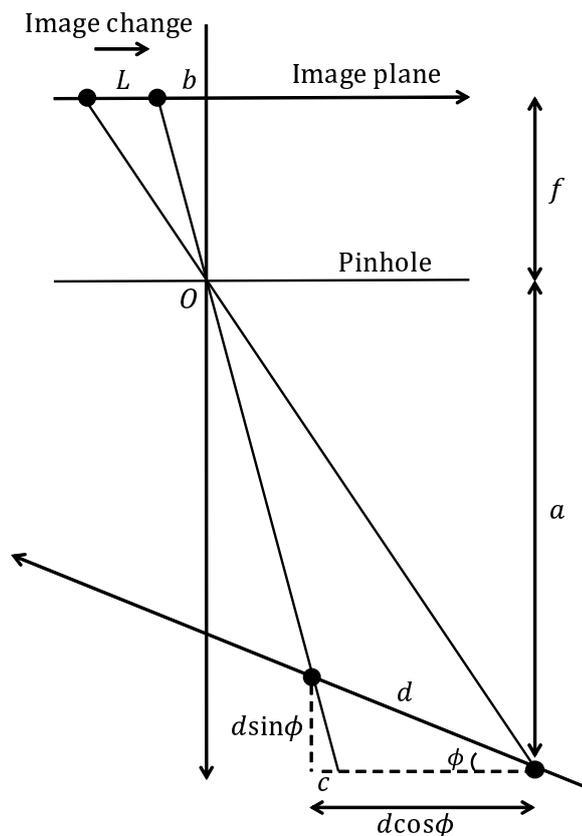


図 3.17 カメラモデル

$$\frac{f}{a} = \frac{LS_p}{d \cos \phi - c} \quad (3.20)$$

上式において、 a はカメラと被写物間との距離、 b は画像面上での被写物の終点、 f はカメラの焦点距離、 S_p はイメージセンサのピクセルサイズである。

式 (3.19),(3.20) から c を除くと、 d は次式のように導かれる。

$$d = \frac{aLS_p}{f \cos \phi - bS_p \sin \phi} \quad (3.21)$$

これより露光時間 T 中の被写体の速度 v は次のようになる。

$$\begin{aligned} v &= \frac{d}{T} \\ &= \frac{aS_pL}{T(f \cos \phi - S_p b \sin \phi)} \\ &= \frac{aS_pL}{Tf \cos \phi (1 - \frac{S_p}{f} b \tan \phi)} \\ &\approx \frac{aS_pL}{Tf \cos \phi} \end{aligned} \quad (3.22)$$

この近似は $S_p \ll f$ で、 $\phi < 45^\circ$ の時成り立つ。特に、 $\phi = 0^\circ$ の場合には上式 (3.22) は更に簡単となり、

$$v = \frac{aS_pL}{Tf} \quad (3.23)$$

と表す事ができる。ここで、 S_p, f 及び T はカメラ搭載時あるいは撮影時に把握できるパラメータである。

従って、画像上の移動量 L とカメラと被写体との距離、つまり宇宙機の高度 a を取得する事で、宇宙機の速度を推定する事が可能である。本手法では、高度は他のセンサによる測定等で得られるものとし、画像上での移動量 L を推定し、(3.23) 式から速度の算出を行う。

3.3.4 提案する動きブレを含む単一画像を用いた位置推定手法

前述したように、ブレ画像のケフレンシ領域において、極小値を探索する事で速度を算出する事が可能となる。しかし、実画像においては図 3.18 の (a) に示すように、原画像のケプストラムが PSF のケプストラムと重なる事やノイズ等の影響から極小値は他の成分に紛れてしまい、極小値がシフトする事で誤差が大きくなってしまふことが分かっている [57]。従って、何らかの工夫を行って PSF に由来するケプストラムを発見しやすくする必要がある。

この問題に対して、Ji ら [58] はブレの方向に画像の勾配を取る事で、PSF のケプストラムを強調して解決した。また、Asai ら [59] は全方向微分フィルタとケプストラム上での平均化を併用する事で、極小値の埋没を防ぎ二次元 PSF の推定を行っている。Shimamoto ら [60] は更にケプストラム上において、平滑化フィルタを用いて極小値の強調を行っている。

一方で、このようにフィルタを用いる事やケプストラム上での平均を取る事は、一画素毎に複数回の乗算を行う事や、分割画像を複数回フーリエ変換する事から、アルゴリズムの計算時間に大きく影響し、リアルタイム性を損なう懸念がある。そこで、本手法では画像を複数枚に分割してパワースペクトル上の平均を取る事で、計算回数を極力増やさないようにし、問題の解決を図る。

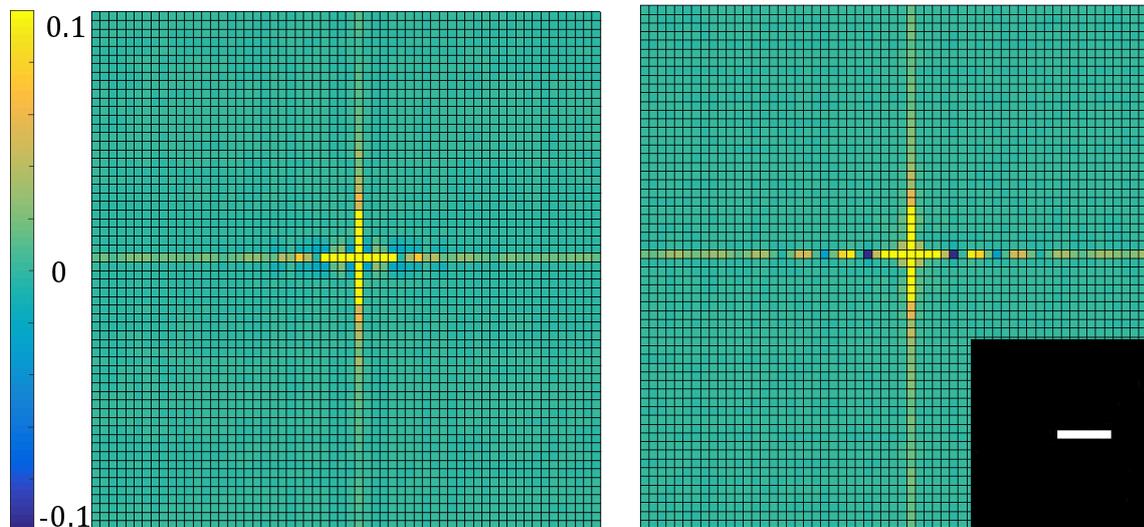
まず、複数枚に分割した画像毎に、フーリエ変換時のエッジ効果を小さくするために窓関数を乗じる。本手法では以下に示すような 2 次元ハン窓を用いる。

$$W_h(x, y) = \frac{1 + \cos\left(\frac{\pi(x-N)}{N}\right)}{2} \frac{1 + \cos\left(\frac{\pi(y-N)}{N}\right)}{2} \quad (3.24)$$

ここで、 N は、分割画像のピクセルサイズの半分の値である。次に、各画像をフーリエ変換しパワースペクトルを求めた後、それらの平均をとる。すなわち、画像の分割数を M とすると、

$$S_g = \frac{1}{M} \sum_{i=1}^M |\mathcal{F}[g_i]|^2 \quad (3.25)$$

のような操作を行う。ここで、 S_g は劣化画像のパワースペクトル、 g_i は分割された劣化画像である。



(a) 平均化を行わない場合

(b) 平均化を行った場合と PSF(左下)

図 3.18 平均化の有無によるケプストラムの差異

このように平均を取る事で、 S_g において原画像成分 f とノイズ成分 n は分割画像毎に異なるために寄与が小さくなり、その一方で PSF 成分 h は全ての分割画像に一様に存在しているために寄与が大きくなる。このような操作を行い、埋没した極小値を発見しやすくしたケプストラムと、真の PSF の概形を図 3.18 の (b) に示す。

この図に示すように、パワースペクトル上で平均化を行う事で、PSF のケプストラムによる負の極小値がはっきりと表れている事が見て取れる。また、ケプストラム上で平均をとった場合は、画像の分割数と同数回の逆フーリエ変換を行う必要があることに対し、本手法においては逆フーリエ変換を一度行えばよいというメリットも存在する。以上に述べたアルゴリズムの流れを図 3.19 に示す。

3.3.5 アルゴリズムの検証

検証内容と方法

本項では、提案した動きブレを含む単一画像による着陸機速度推定法の有効性を確認するため、月面地形画像に対して動きブレとノイズの付与を行い、撮影画像として得られる動きブレを含む画像を模擬することにより実験を行った。

提案した手法について、その性能に関わる事項は次の通りである。

- 原画像が自然画像であること
- 宇宙環境特有のノイズが画像に対して乗ること
- 地形の変化による視差変化が存在すること

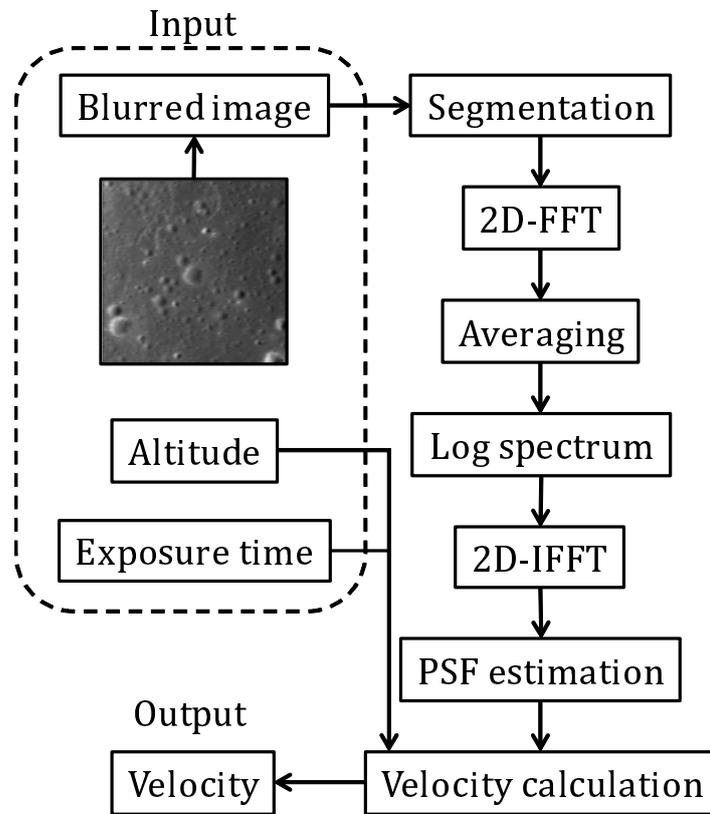


図 3.19 速度推定のアルゴリズムフロー

当手法では、入力として動きブレを含む月面の地形画像を想定しているため、その原画像としては自然画像となる。図 3.16 で示したように、人物や人工物の画像においては低周波成分から高周波成分まで広く分布するために、低ケプレンシ領域における原画像の影響は少なくなる。一方で、月面地形のような自然画像においては低周波成分が強くなる傾向にあるため、低ケプレンシ領域における原画像の影響が強くなる可能性がある。原画像の影響が強い場合には、PSF に由来する負の極小値は埋没してしまい、結果としてブレ幅の推定精度に影響を与える。

また、式 (3.18) で示したように、原画像のケプストラムと PSF のケプストラムが加法的に分離できるのは、ノイズがごく小さい場合である。一方で、地形相対航法として速度推定を行う場合には、宇宙環境特有のノイズが多く乗る可能性がある。動きブレを含む画像に、大きな加法的ノイズが乗った場合には原画像のケプストラムと PSF のケプストラムは加法的に分離できず、そのノイズの影響で極小値が埋没する。焦点ぼけのように畳み込みで表されるノイズに関しても、ケプストラム領域で PSF の概形を埋没させる。

式 (3.13) で示した動きブレを含む画像のモデルは、露光時間中に対象となる風景が平行移動を除いて時間的に不変であることを仮定している。したがって、はっきりとしたブ

レを取得するために露光時間を長く取るなどして風景内に視差による変化などが発生した場合には、その原画像の形状変化によって PSF の概形もケプストラム領域上で変動する。そこで本検証では、以下の二つの検証を行う。

- 月面の地形画像から様々な長さの動きブレやノイズを付与し、提案手法を適用することによって正しいブレ幅を推定可能な性能について見積もる。
- 月面の地形ジオラマを対象として動きブレを発生させた実画像を作成し、速度推定の性能について見積もる。

本検証において、提案アルゴリズムは前節における検証と同様に C 言語にて実装し、表 3.2 に示した汎用 PC 上での実験とした。

動きブレとノイズを付与した月面画像の作成

検証のため、月面地形画像から複数の幅を持つ動きブレと地形相対航法の実施時に存在する可能性のあるノイズを付与することで、動きブレを含む画像を模擬する。使用した月面画像は、前節における検証と同様の神酒の海周辺の 60×70 km, 2264×2641 pix となる範囲の画像である。動きブレ画像作成に用いた画像は図 3.8 である。この地形画像について、任意の位置で 512×512 pix の範囲を持つ 1000 枚の画像を作成し、以降のノイズ付与処理の原画像とした。

この原画像の例を図 3.20 に、原画像をケプストラムに変換した時の低ケフレンシ領域の概形について図 3.21 に示す。ここで、赤実線は 1000 枚の原画像におけるケプストラムの平均で、ピンク色の領域はその標準偏差の 3 倍 (3σ) を表しており、青実線は $L = 15$ のときの PSF である。図 3.16 と比較してみると、自然画像のケプストラムは人物のケプストラムよりも低ケフレンシ領域において概形を色濃く残していることがわかる。

まず、式 (3.13), (3.14) にしたがって、原画像に対して PSF を畳み込むことにより、ノイズのない理想的な動きブレを付与する。このとき、ブレ幅 L は 3-25pix の範囲を 1pix づつ刻み、各 1000 ケース作成する。図 3.22 に、ノイズのない動きブレ画像のケプストラムを示す。以降同様の図においても、赤実線は 1000 枚の原画像におけるケプストラムの平均で、ピンク色の領域はその標準偏差の 3 倍 (3σ) の範囲を表している。

これを見ると、 $L = 3$ のように小さいブレ幅のケプストラムは、ノイズが無いにもかかわらず原画像の成分に埋没し、周期的な負の極小値が表現されていない。ブレ幅が大きくなるにつれ高ケフレンシ、領域に極小値が移動するため、原画像の成分より PSF の成分が強くなることではっきりとした極小値の概形が表現されている。

この原画像に対して、ノイズを与える。付与するノイズは次の通りである。

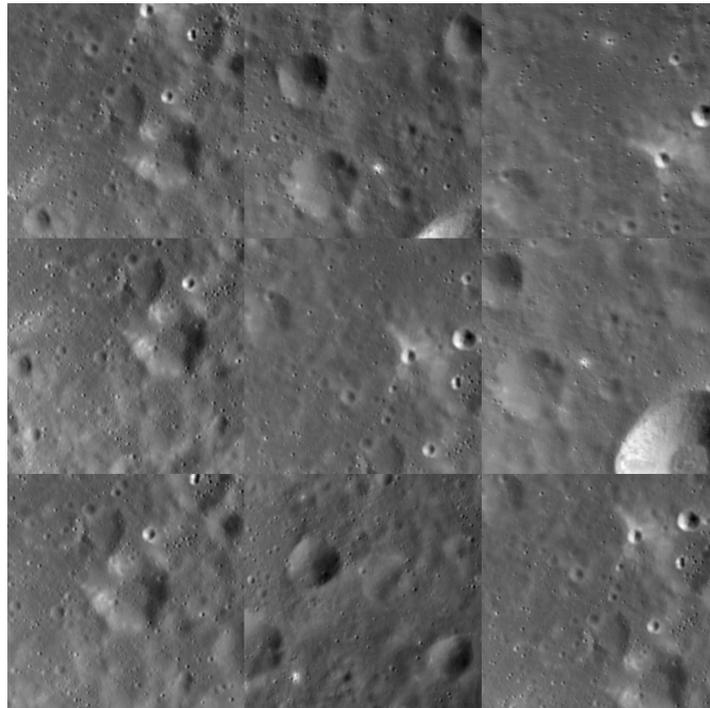


図 3.20 動きブレのない原画像

明度 画像の明度に増減を与える

コントラスト 画像のコントラストに増減を与える

焦点ぼけ 画像に対してガウシアンに基づくぼけを与える

輝度の揺らぎ 画像の各画素においてガウシアンノイズを与える

放射線ノイズ 白と黒の確率が半分のごま塩ノイズを与える

これらのノイズは、原画像そのものに付与されるもの、ブレ画像に加法的に付与されるものに分けられる。ノイズ付与の方法や、付与したノイズの大きさについて、表 3.10 に示す。ここで、 img は輝度を 0-1 に正規化した原画像であり n_b はノイズの大きさである。また検証のため、ノイズの最小・最大値については通常想定されるよりも過分に与えている。この表の方策に従い、原画像に対してランダムなノイズを付与した。

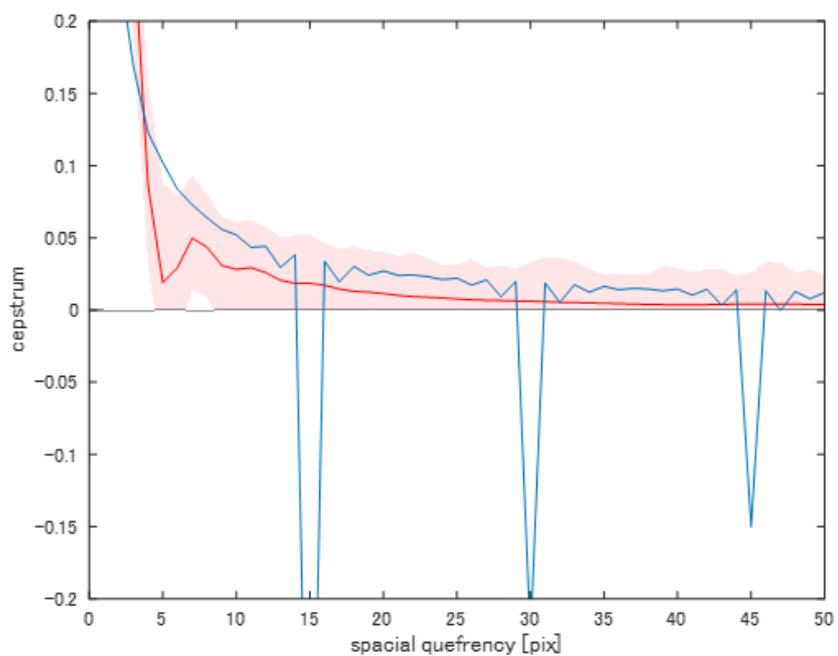


図 3.21 動きブレのない原画像のケプストラムと $L = 15$ の PSF の概形

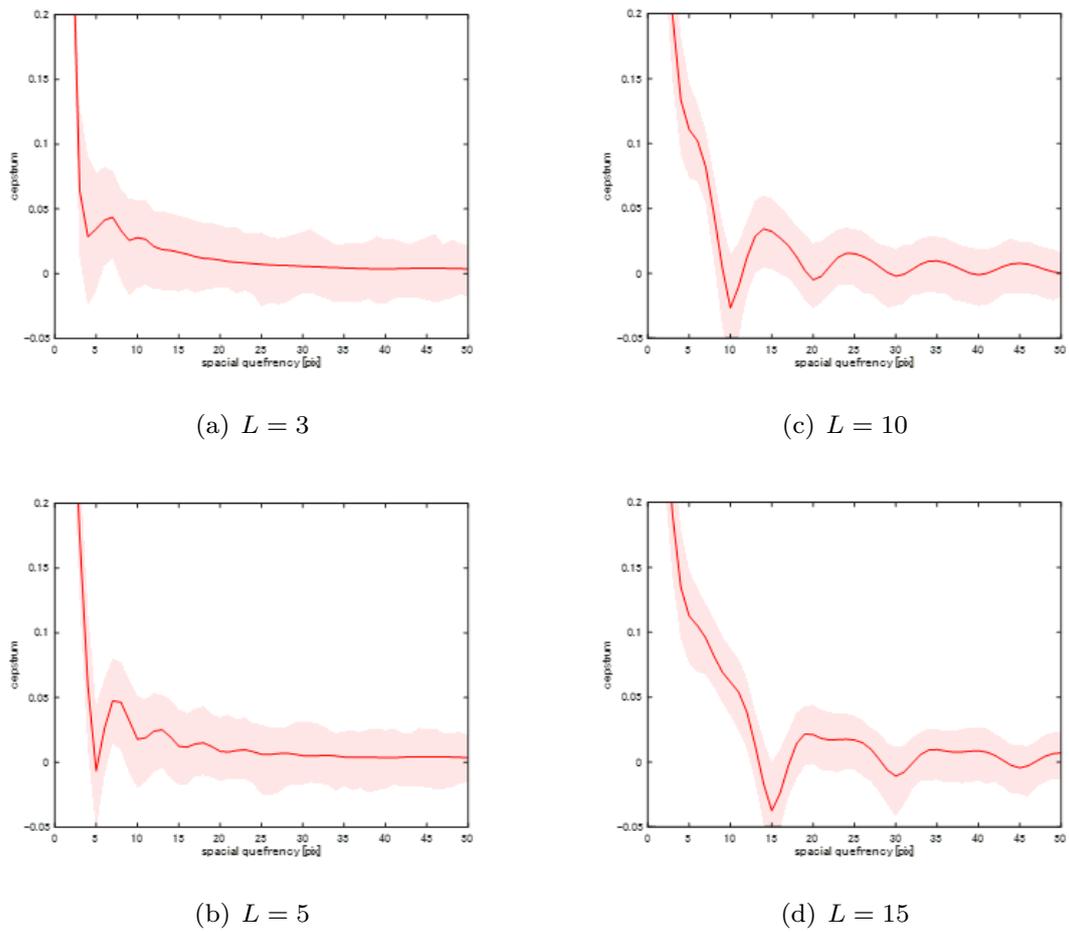


図 3.22 ノイズのない動きブレ画像のケプストラム

表 3.10 ブレ画像に付与するノイズとその量

付与するノイズ	付与の方法	最大値	最小値
明度	原画像, $img + n_b$	-0.5	0.5
コントラスト	原画像, $(img - 0.5) \times 10^{n_b} + 0.5$	-1	-1
焦点ぼけ	原画像に対して標準偏差 n_b のガウシアンフィルタをかける	0	5
輝度の揺らぎ	加法的, 各画素を分散 n_b で変化させる	0	1e-4
放射線ノイズ	加法的, $n_b \times 100\%$ の画素にランダムでごま塩ノイズを生成する	0	1e-4

3.3 動きブレ画像による速度推定手法 3 現状の低計算リソース環境における地形相対航法

明度変化を与えたブレ画像の例を図 3.23 に示す。また、そのケプストラム形状について図 3.24 に示す。ここで赤実線とピンクの領域は図 3.22 と同様に 1000 ケースの平均と 3σ である。また、比較対象としてノイズのないブレ画像のケプストラムである、図 3.22 における平均値を青実線で表している。明度変化においては、赤実線が青実線よりも強く極小値を表しており、原画像の影響が低減されている。これは、明暗方向に輝度値が圧縮されることによりコントラストが上がり低周波成分が少なくなったため、結果として低ケプレンシ領域の原画像の信号が小さくなったと考えられる。

コントラスト変化を与えたブレ画像の例を図 3.25 に示す。また、そのケプストラム形状について図 3.26 に示す。コントラスト変化については、明度変化よりはっきりと極小値が現れており、より原画像の影響が低減されている。これは当然明度変化よりもコントラストが上がったために、低周波成分が少なくなったからだと考えられる。

焦点ぼけを与えたブレ画像の例を図 3.27 に示す。また、そのケプストラム形状について図 3.28 に示す。焦点ぼけについては、輝度値が鈍り低周波成分が増えるために極小値は見えづらくなっており、原点に近い部分でピークを持つ $L = 3, 5$ の動きブレにおける負の極小値は消失してしまっている。

輝度の揺らぎを与えたブレ画像の例を図 3.29 に示す。また、そのケプストラム形状について図 3.30 に示す。輝度のゆらぎについては加法的なノイズであるため、原画像の成分も PSF の成分も完全に概形は失われている。ただ、図では平均しているため見づらい

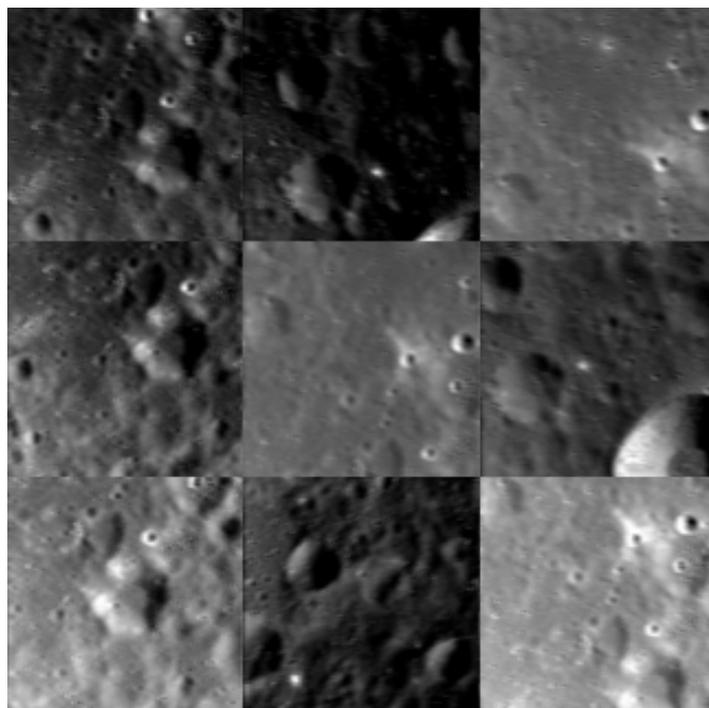


図 3.23 明度変化を与えたブレ画像

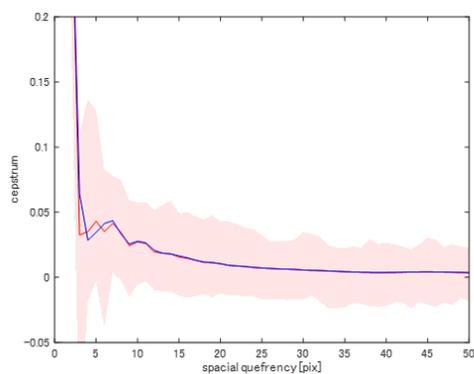
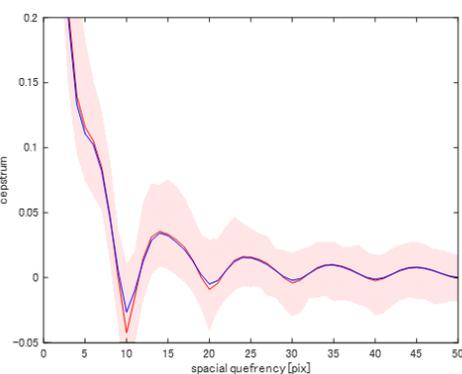
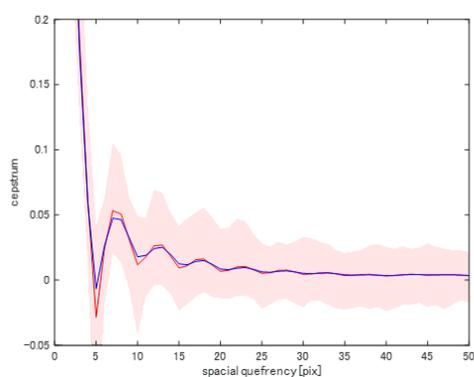
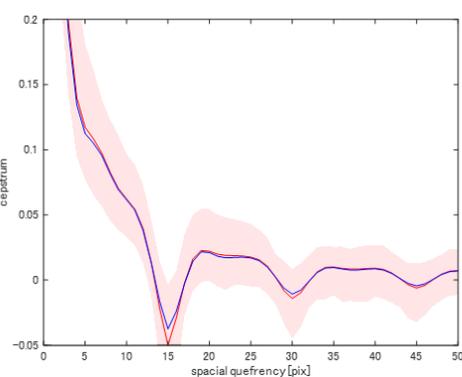
(a) $L = 3$ (c) $L = 10$ (b) $L = 5$ (d) $L = 15$

図 3.24 明度変化を与えた動きブレ画像のケプストラム

ものの、ブレ幅にしたがった極小値は小さくはあるものの存在している。

放射線ノイズを与えたブレ画像の例を図 3.31 に示す。また、そのケプストラム形状について図 3.32 に示す。放射線ノイズについても加法的なノイズであるため、輝度のゆらぎと同様の理由から原画像の成分も PSF の成分も完全に概形は失われている。また、極小値は小さくはあるものの存在している。

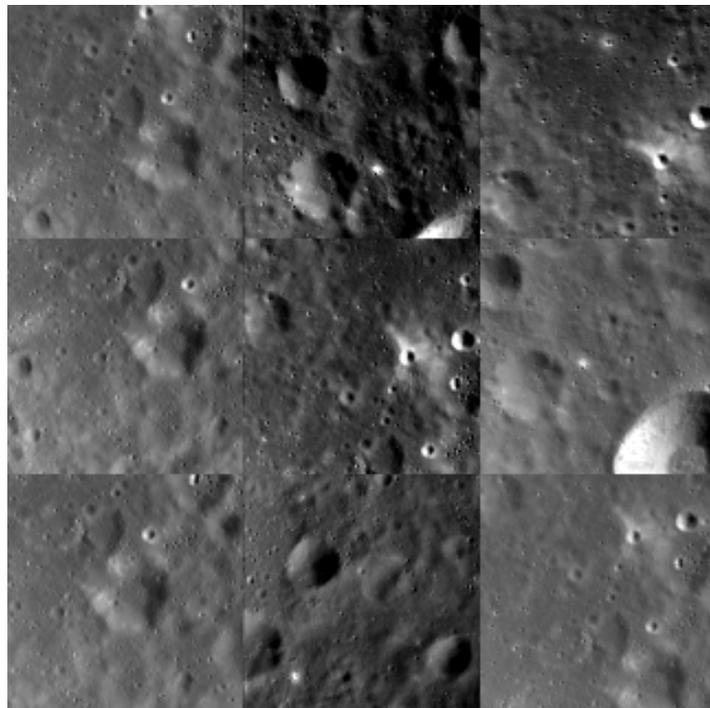
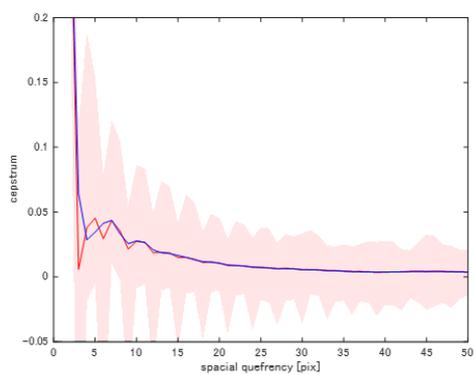
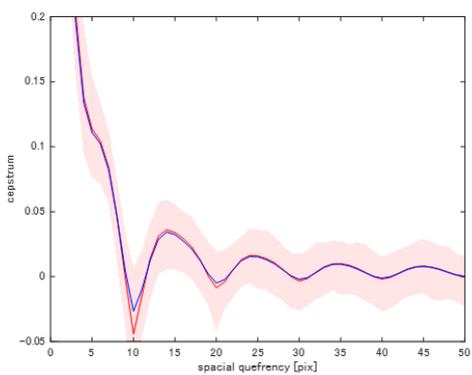


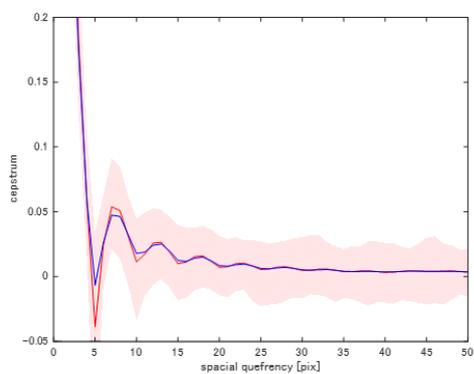
図 3.25 コントラスト変化を与えたブレ画像



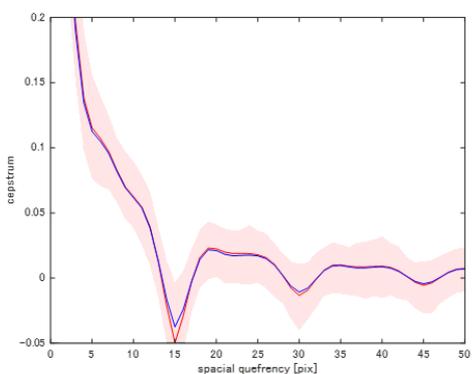
(a) $L = 3$



(c) $L = 10$



(b) $L = 5$



(d) $L = 15$

図 3.26 コントラスト変化を与えた動きブレ画像のケプストラム

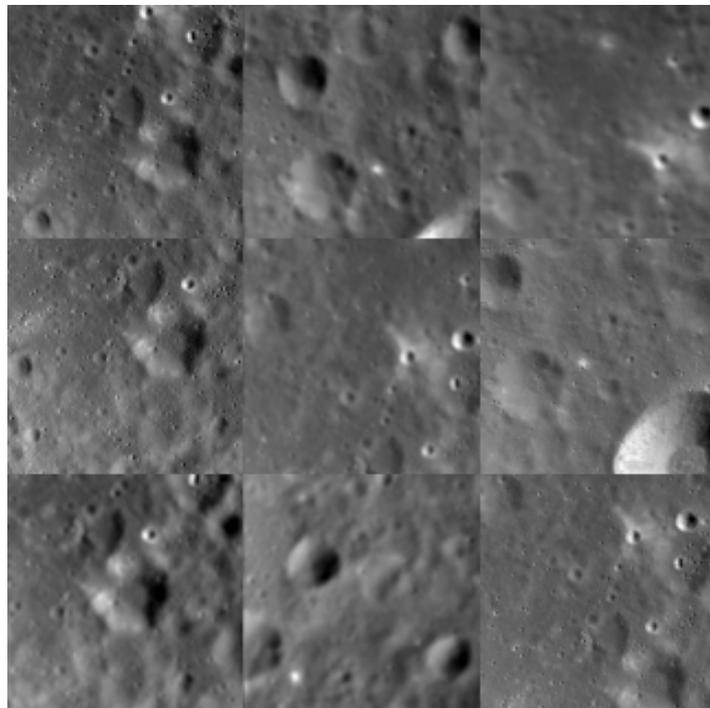


図 3.27 焦点ぼけを与えたブレ画像

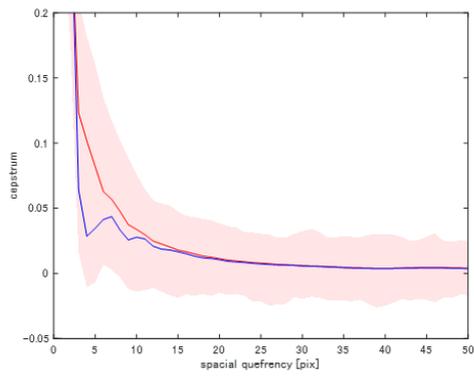
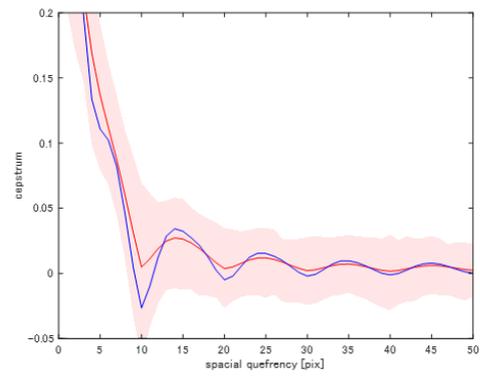
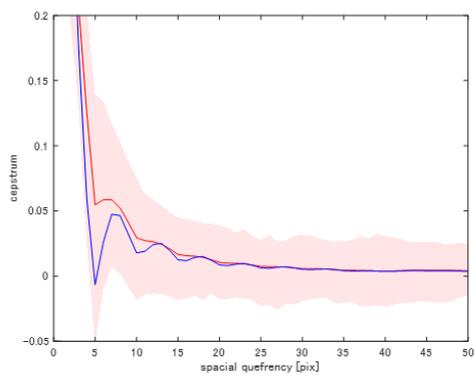
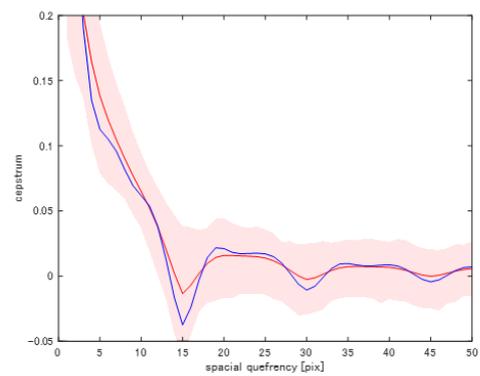
(a) $L = 3$ (c) $L = 10$ (b) $L = 5$ (d) $L = 15$

図 3.28 焦点ぼけを与えた動きブレ画像のケプストラム

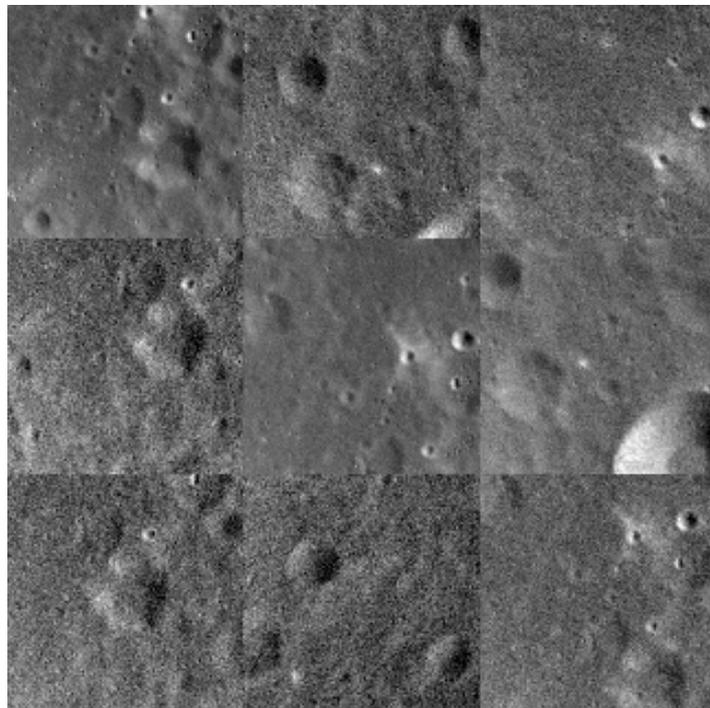


図 3.29 輝度の揺らぎを与えたブレ画像

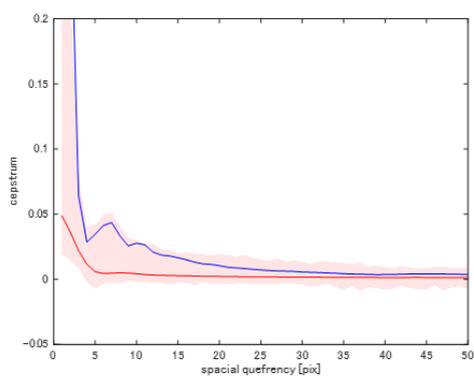
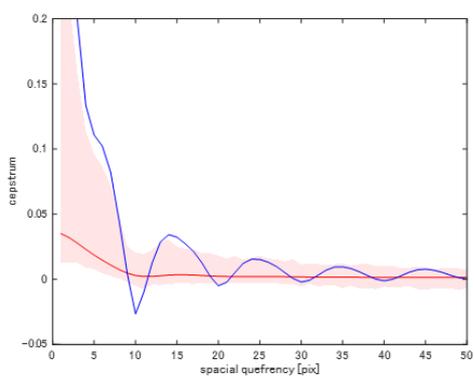
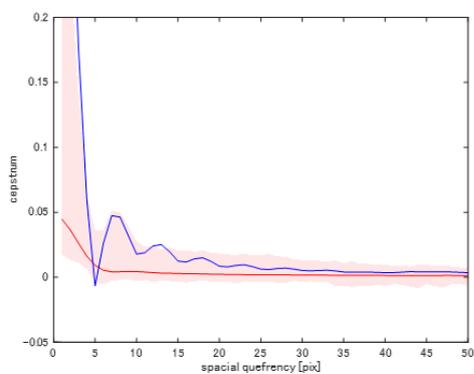
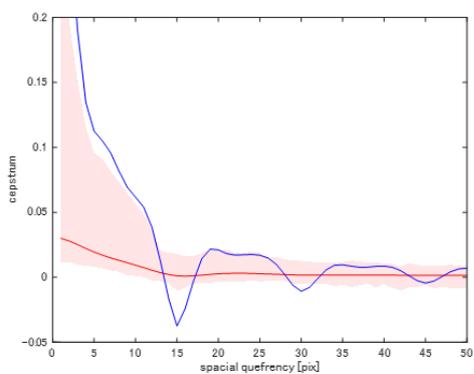
(a) $L = 3$ (c) $L = 10$ (b) $L = 5$ (d) $L = 15$

図 3.30 輝度の揺らぎを与えた動きブレ画像のケプストラム

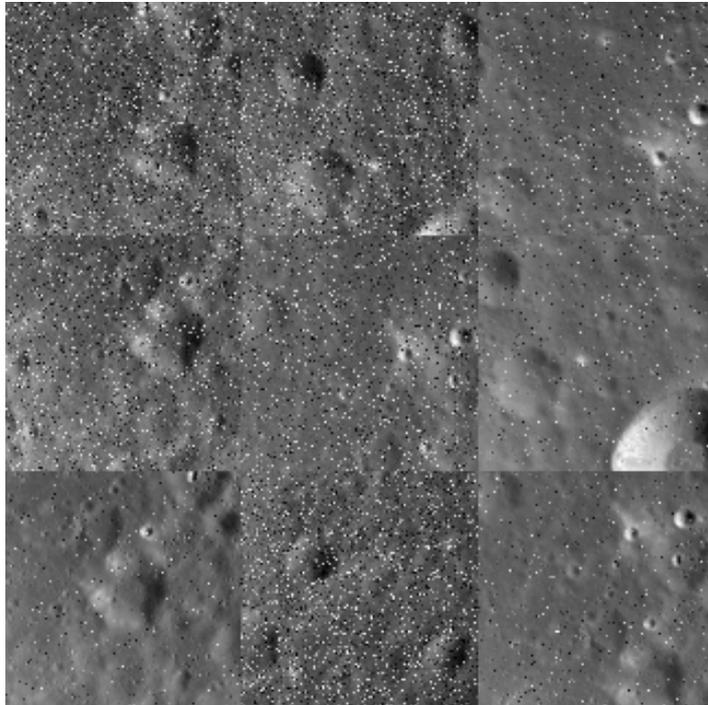
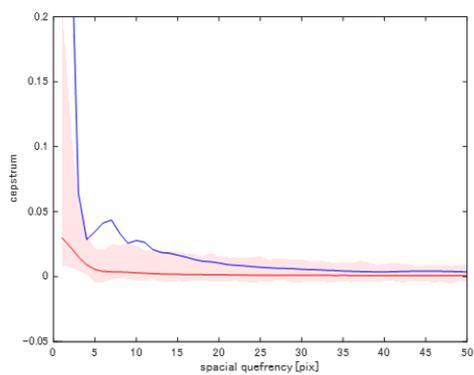
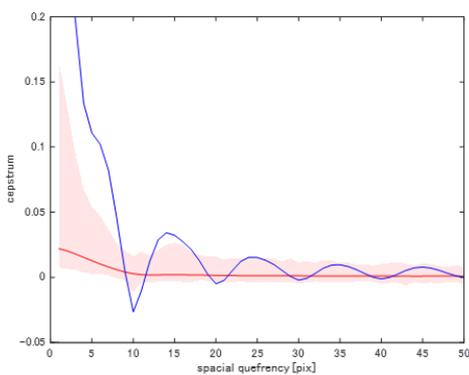


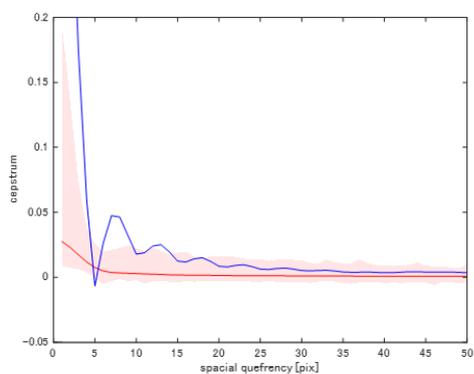
図 3.31 放射線ノイズを与えたブレ画像



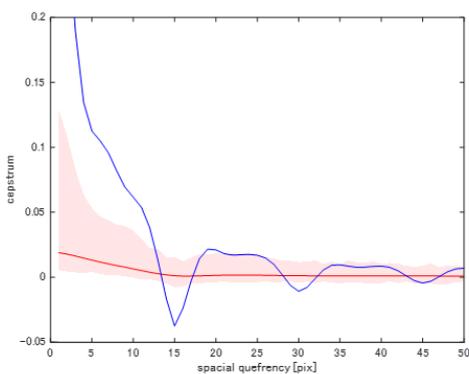
(a) $L = 3$



(c) $L = 10$



(b) $L = 5$



(d) $L = 15$

図 3.32 放射線ノイズを与えた動きブレ画像のケプストラム

ノイズの存在する動きブレ画像に対する検証

本検証では、前項で作成した複数のノイズを付与した動きブレ画像について、提案手法を適用することにより、PSF の推定が難しい自然画像やノイズを含んだ画像において、ブレ幅の推定がどの程度正しく行えるかを評価する。提案手法では、前述したようにブレ幅を推定した後に、外部から与えられた高度情報と露光時間をが正しいとして速度推定を行っているため、本検証においては、誤差原となる画像中のブレ幅の推定精度について考える。

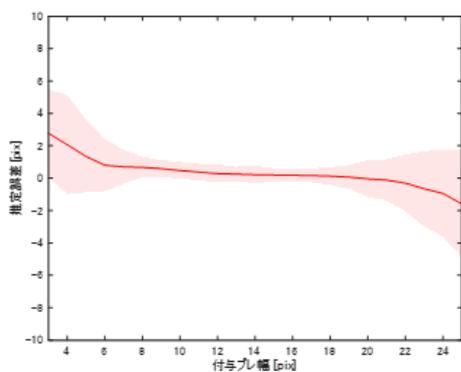
前項にて、明度、コントラスト、焦点ぼけ、輝度の揺らぎ、放射線ノイズの5種類のノイズに関して、それぞれ3-25pixの動きブレを付与した画像を作成した。これらノイズありの画像と、ノイズのない理想的な動きブレ画像の6種類のデータセットに対して提案手法によるブレ幅の推定を行った。ここで、平均化を行う際の画像分割数 M は25pixのブレ幅を推定可能な最大の値である16に設定した。

各ノイズのデータセットについて、付与したブレ幅ごとの推定精度を表したものを図3.33に示す。ここで、赤実線は1000枚の原画像における推定誤差の平均で、ピンク色の領域はその標準偏差の3倍 (3σ) を表している。まずこの結果から、全体の傾向として付与したブレ幅がおおよそ7pixを下回るあたりから、正の方向にバイアス誤差がかかっており、また20pixを上回るあたりから負の方向にバイアス誤差がかかっている。前者は、低ケフレンシ領域において強い成分を持つ原画像の影響によって、極小値が高ケフレンシ方向に移動することによって生じる。後者は、高ケフレンシ領域になるにつれPSFに由来する極小値は小さくなることから、原画像の成分と混ざることによってPSFと関係のない成分との区別が付きづらくなることによって生じると考えられる。それ以外の領域では、 3σ でおおよそ2pix以内の推定誤差となっている。

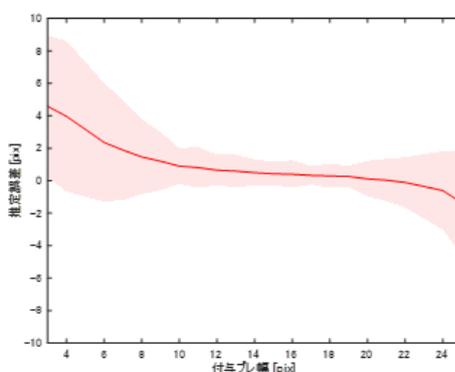
各ノイズ毎には、前項で確認したようにノイズのないものよりも明度、コントラストに変化があるものの方が精度がよい。またケプストラム上の極小値が埋没してしまっていた、輝度の揺らぎと放射線ノイズに関しては、平均化によって加法的なノイズの影響が低減したことにより、他のケースと比較しても遜色無く推定が行えている。最も精度が悪いものは焦点ぼけによるケースである。これは、焦点ボケは動きブレのPSFと同様にブレ画像に畳み込む形で付与されるため、平均化によるノイズ成分の低減ができないことに起因している。

続いて、ノイズを付与した量ごとの推定精度について、図3.34に示す。全体の傾向としては、ノイズ量によって大きく推定誤差が変動するということない。輝度の揺らぎ、放射線ノイズのケースではノイズが無い場合に 3σ で2pix程度の推定誤差だったものがノイズ量が増加するにあたって徐々に4pix程度まで増大する傾向にある。

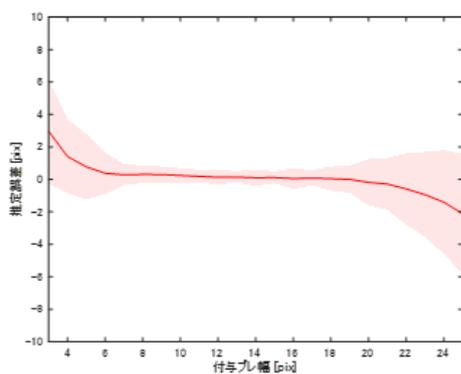
この結果は、図3.33に示した全ての結果を含んだものである。そこで誤差の大きい7



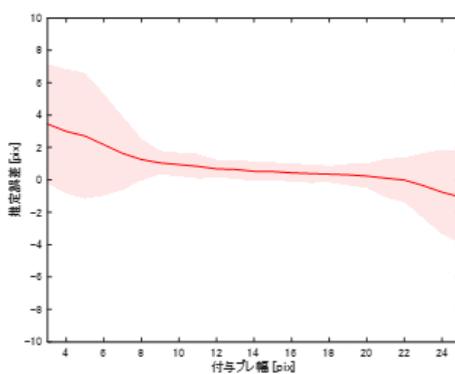
(a) ノイズなし



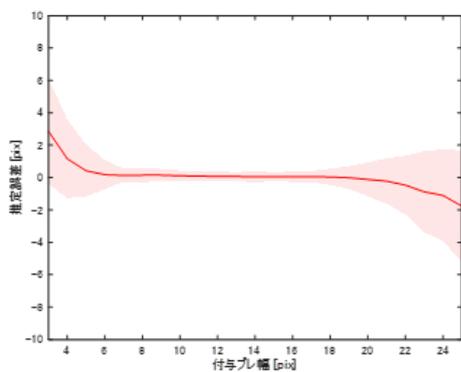
(d) 焦点ぼけ



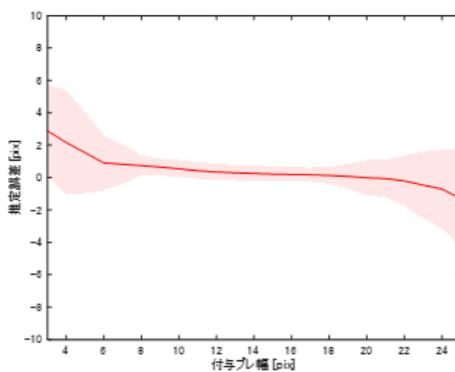
(b) 明度変化



(e) 輝度の揺らぎ



(c) コントラスト変化



(f) 放射線ノイズ

図 3.33 各データセットにおける付与したブレ幅ごとの推定精度

3.3 動きブレ画像による速度推定手法 3 現状の低計算リソース環境における地形相対航法

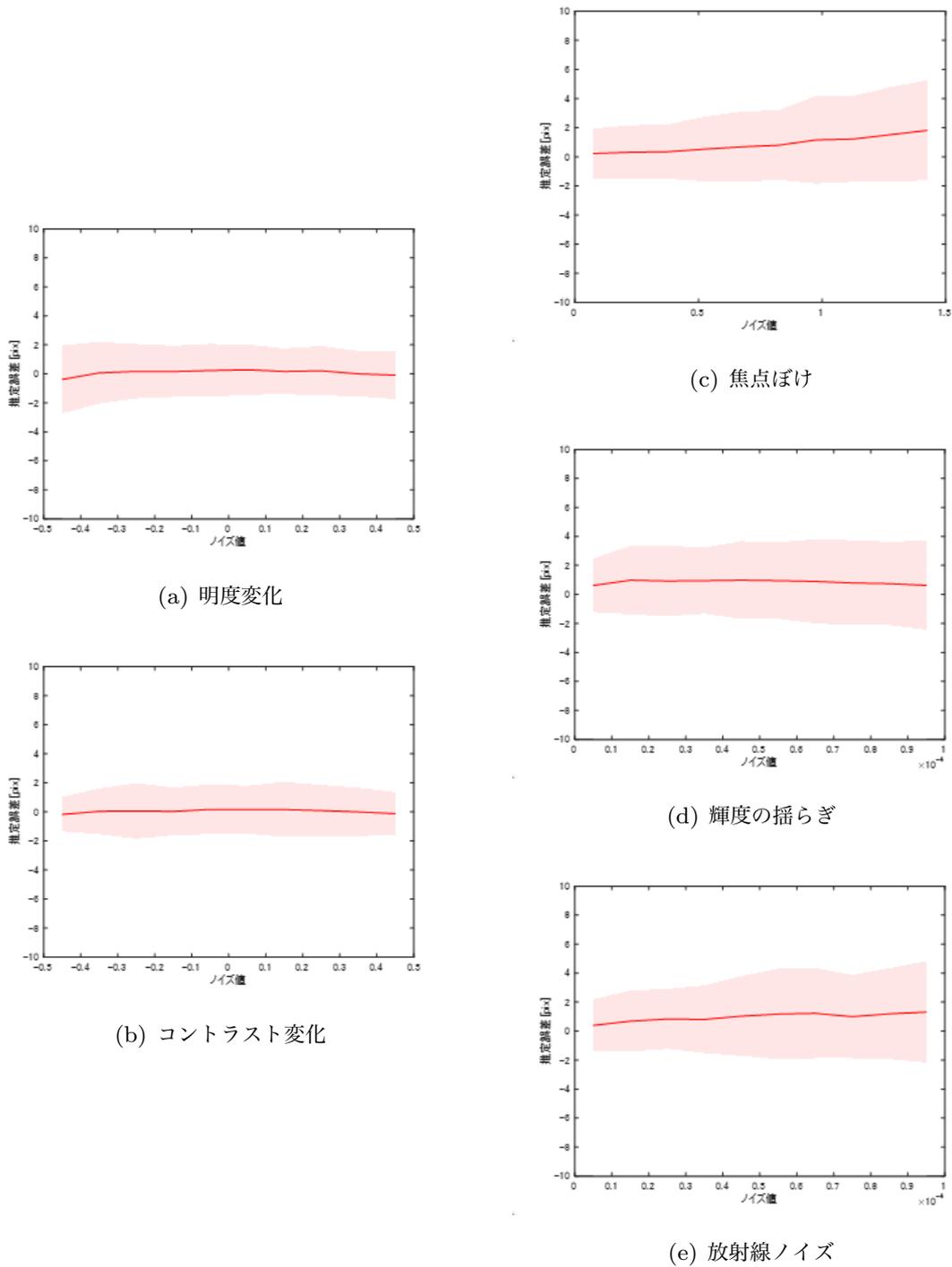


図 3.34 各データセットにおける付与したノイズ量ごとの推定精度

pix 以下, 20 pix 以上のブレ幅について除き, 再度ノイズ量ごとの推定精度について見てみると, 図 3.35 のようになる. これは, 全ての結果において 3σ で 2.5 pix 以内の推定誤差となる.

したがって, 本手法は表 3.10 に示したノイズの種類と量が付与されている動きブレ画像について, そのブレ幅が 7-20 pix 程度であれば, 3σ で 2.5 pix 程度の誤差で一次元の動きブレにおけるブレ幅を推定することが可能である.

実画像を用いた検証

前項においてはノイズを含んだ動きブレ画像を模擬することで, どの程度の精度でブレ幅を推定可能かを検証した. 一方で, 実画像においてはノイズの他に, 露光時間中に風景が変動することにより式 (3.13) で表したモデルから外れる可能性が存在する.

そこで本検証においては, カメラを地表面に正対している状態で水平移動をさせ, 得られる画像を用いて速度の推定を行った. 動きブレ画像の撮影にあたって, 実験装置は直動ステージ (シグマ光機 SGSP 46-800(X)) に取り付けられた産業用カメラと, 月地表面の 1/400000 スケールのジオラマで構成される. 直動ステージによってカメラが等速直線運動を行う間に, ジオラマを撮影する事によって動きブレ画像を取得出来るようになっている. この実験装置の概観図を図 3.36 に示す.

本実験における画像取得時の速度 v は 5 cm/s, カメラとジオラマとの距離 a は 114 mm である. また画像中に動きブレが確実に発生するよう, 露光時間 T を 100 ms とした. この時のブレ幅の理論値は 47.0 pix である. また, 照明はジオラマ表面に対しておよそ 45 度上方から当てられている. このような条件の下, 取得された画像 35 枚に対して提案手法による速度推定を行った.

得られた劣化画像とケプストラムの例を図 3.37 に, 速度の真値と各画像における推定速度をプロットしたものを図 3.38 に示す. この時推定されたブレ幅の最大値は 51.0 pix, 最小値は 43.0 pix, 平均値は 47.4 pix となった. 前項では, ブレ幅が 7-20 pix 程度であればノイズを含んでいても 3σ で 2.5 pix 程度の誤差でブレ幅を推定することが可能であると分かった. 本検証のブレ幅の推定においては, それらよりも倍程度長い動きブレが発生していてもノイズがない場合では 3 pix 程度の誤差で推定が可能であることを示している.

速度の推定結果を見ると, 平均値は 50.1 mm/s で標準偏差が 2.1 mm/s となった. また, 推定値は段階的な誤差が生じている. 誤差の原因としては, 実験系に起因するものと速度推定の分解能に起因するものの 2つが考えられる. 前者においては, 影の存在による情報欠落に加え, カメラ-被写物間の距離に対して画面内の地形の高低差が 1 割程度存在し, その影響で入力として与えたスケールに若干の誤差が乗ったと考えられる. 後者は,

3.3 動きブレ画像による速度推定手法 3 現状の低計算リソース環境における地形相対航法

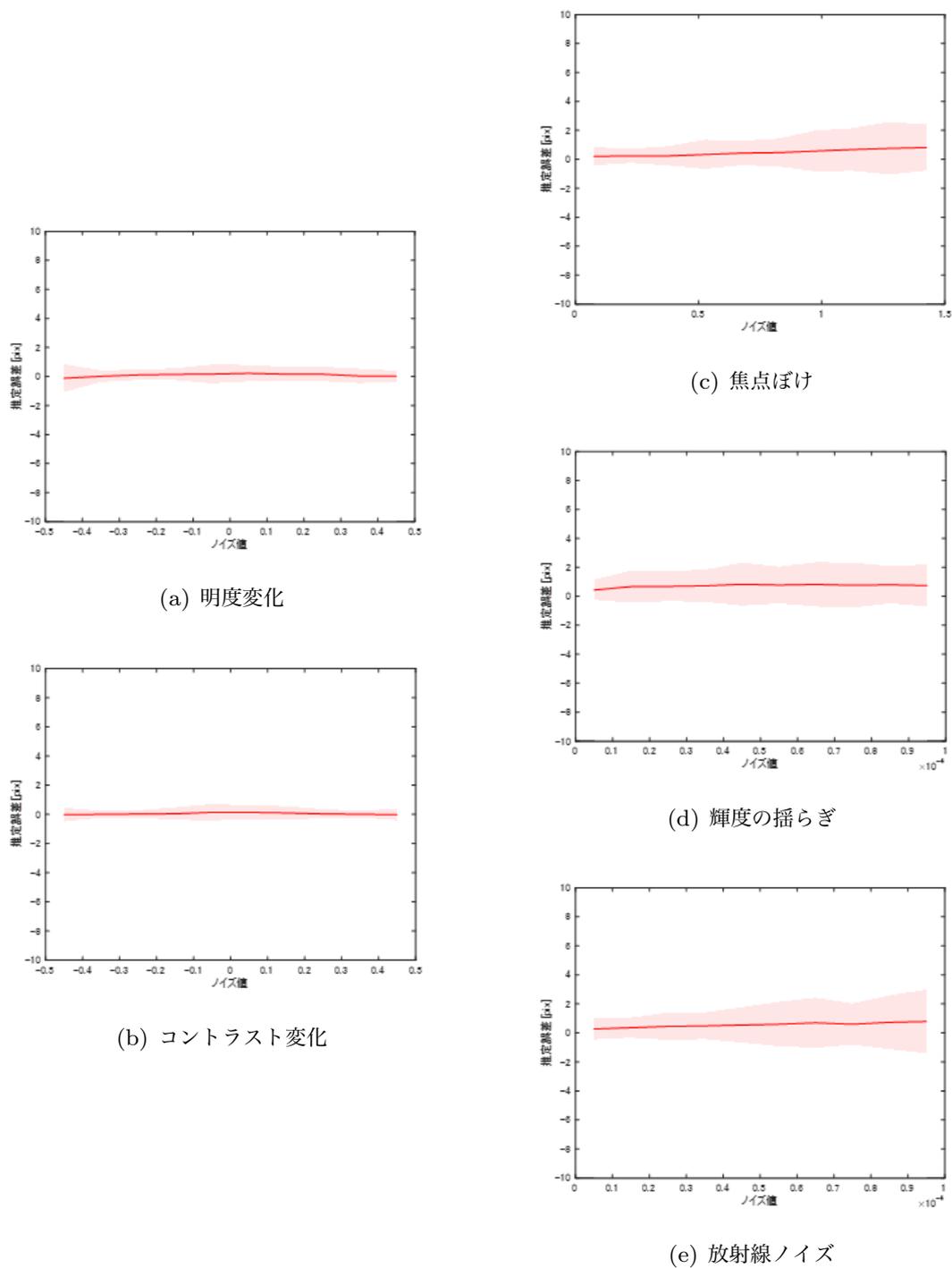


図 3.35 ブレ幅が 7-20pix での各データセットにおける付与したノイズ量ごとの推定精度

実際の劣化画像においては、ブレ幅 L がサブピクセル単位まで存在する事に対して、提案手法は1ピクセル単位までの推定に留まることに起因し、段階的に誤差を発生させる原因となっている。それに加えて、図 3.37 の (b) で見られるように、ケプストラムの極小値が複数ピクセルにわたって算出される事で、真の極小値がピクセル間に存在した場合に、推定値が1ピクセル程度変動してしまう場合がある。

前者の問題に対して、この結果にはバイアス誤差が存在しないため、着陸機の誘導パス中に速度推定を複数回実施し、適当なフィルタを通す事で実用に足ると考えられる。後者に対しては、イメージセンサのピクセル数を増やす、レンズの画角を狭める等の分解能を

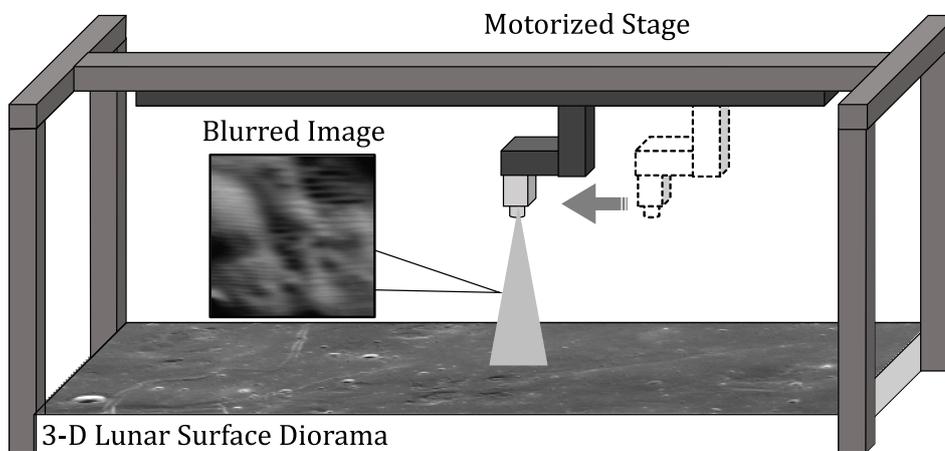
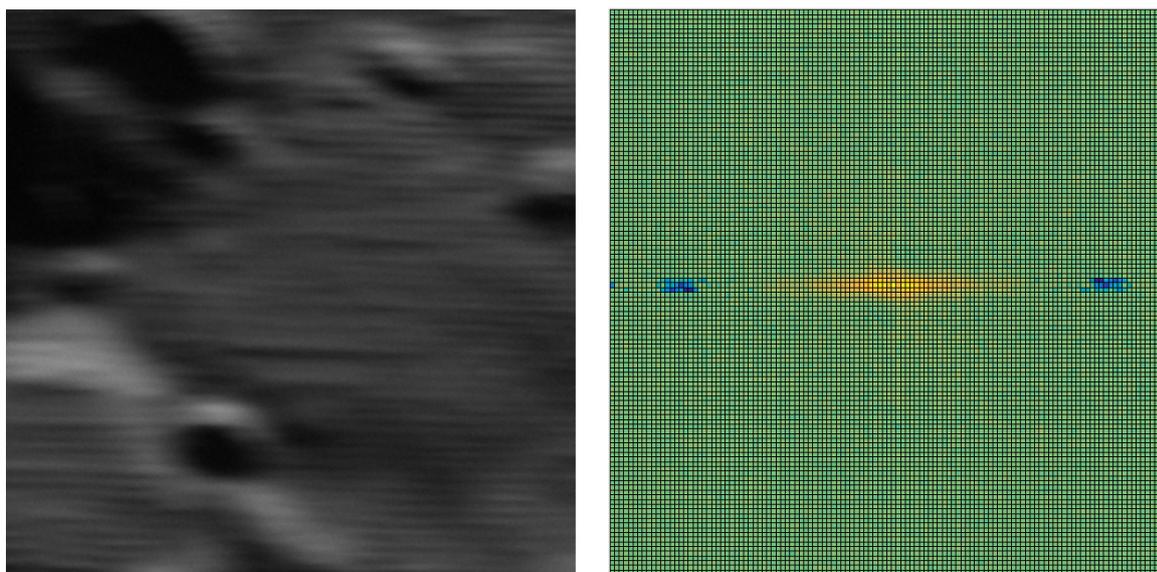


図 3.36 実験装置概観図



(a) 取得劣化画像

(b) 劣化画像のケプストラム

図 3.37 実験用画像

3.3 動きブレ画像による速度推定手法 3 現状の低計算リソース環境における地形相対航法

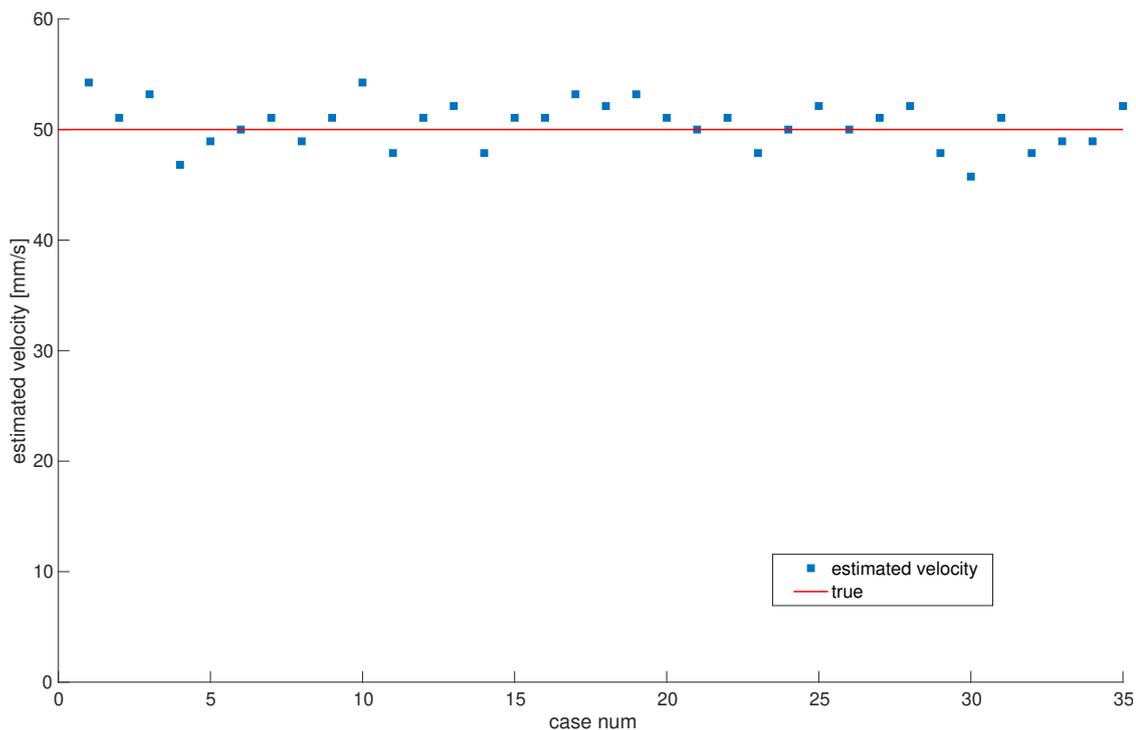


図 3.38 速度推定結果

向上を図る他、極小値前後のピクセル情報からブレ量のサブピクセル補間を行う事で、より高精度な推定が可能であると考ええる。

検証のまとめと考察

本検証では提案手法の性能を評価するために、実際の月面画像を用いて想定されるノイズについて5種類を動きブレ画像に対して付与した。これらのノイズの量については、実際に起きうる範囲よりも広く取ることにより、ノイズに対する提案手法の限界を図ることを狙った。また、自然画像のケプストラムは低ケフレンシ領域にて影響が強いことを確認した。

このノイズを含む動きブレ画像に対して提案手法を適用することにより、露光時間中の移動量となるブレの幅について推定した。結果としては、多くノイズが乗っている場合に、数 pix 程度以下の小さいブレと二十数 pix 以上のブレについては推定結果にバイアス誤差が乗る。これは、自然画像のケプストラムの影響が強いことにより、小さいブレと大きいブレの極小値が埋没してしまうことに起因する。それ以外の範囲においては、ノイズが多く乗っていても 3σ で 2.5 pix 程度の精度で移動量の推定が可能であることを確認した。殆どのノイズについては提案手法によってその影響を低減できたものの、一方で焦点ぼけ画像については、画像に畳み込まれる形でノイズが乗るために提案手法によって原理的に影響を減らす事ができない。しかしながら今回付与した量のぼけ量においては、加法

的なノイズ下における推定結果とほぼ同等の推定が可能であることを確認した。

このような模擬画像では、原画像についての変動の影響を図ることは出来ないため、ジオラマと直動ステージによる動きブレを含む実画像の取得を行い、速度推定を実施した。結果としては、真値が 47 pix と大きな幅を持つにもかかわらず、3 pix 程度の推定誤差でブレ幅の推定が可能であることを確認した。

これらのことから、動きブレ画像に多くのノイズが乗っていても、画像上において 7-20 pix の移動量であり、被写体までの高度情報が適切に与えられているとき、当手法は高精度で速度推定が可能であるといえる。また、ノイズが存在せずハイコントラストな地形を対象とする場合には、50 pix 程度の移動量まで検出が可能である。

3.4 宇宙機への搭載性評価

本章では、宇宙機用の FPGA に対して提案アルゴリズムが消費するリソース量の規模感、処理速度を確認するための評価を行った。FPGA 回路の設計においては、従来的にハードウェア記述言語 (hardware description language; HDL) によるレジスタ転送レベル (register transfer level; RTL) 設計が用いられる。RTL 設計では、記述段階から処理をクロック単位で記述する必要がある、動作周波数や信号のタイミングを意識する必要がある等の理由から、試行錯誤による最適化試行や対象機器の変更に伴う設計変更が安易とは言い難く、また動作評価に多くの時間がかかってしまう。したがって、そのような手法によって複雑な画像処理を開発するのは難しく、開発の長期化を招く。このため、本評価では高位合成を用いた設計手法を適用することで、画像航法アルゴリズムの評価と実装を行った。

3.4.1 高位合成

高位合成とは、ハードウェアの設計対象となる処理の動作記述と、それに協調するソフトウェア記述を、拡張された C 言語や C++ 言語といった抽象度の高いプログラミング言語を用いて記述し、高位合成ツールに入力する事で、RTL 記述の生成を自動的に行う手法である。高位合成系を用いた開発においては、プログラミング言語を用いるために、システムレベルでの検証やデバッグが容易であり、かつ、その検証によってハードウェア設計、全体検証の殆どを一貫して行う事が可能であることから、全体的な設計工数も短縮できる。従来設計手法と高位合成系を用いた設計手法の比較を図 3.39 に示す。

このような高位合成は、近年の FPGA の大規模化に伴って今後必須となる事が予想されるため、ザイリンクス社やアルテラ社といった各 FPGA メーカー等でもツールの提供が始まっている。メーカー提供の高位合成ツールは、主に各メーカーの FPGA に搭載す

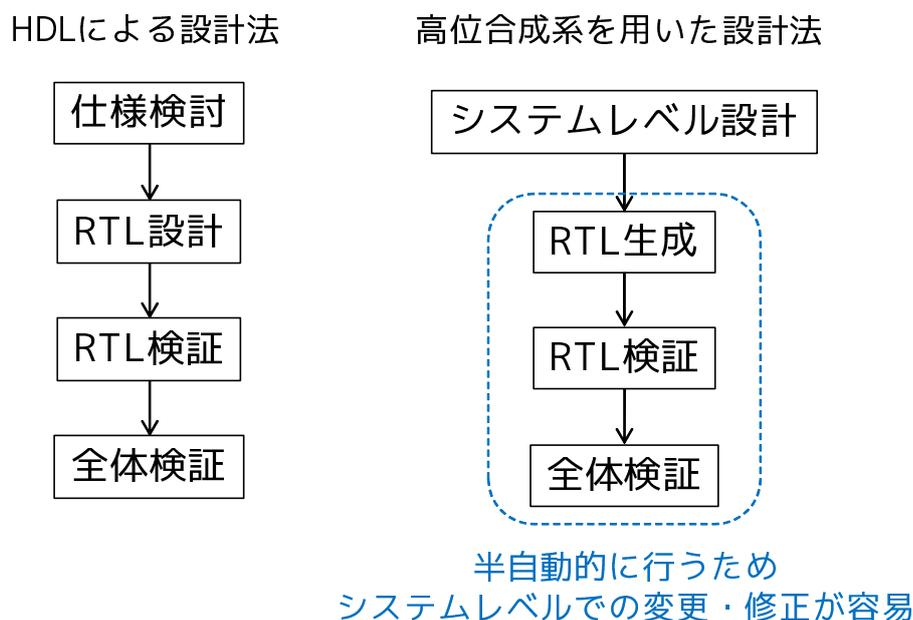


図 3.39 設計手法の比較

ることを目的として提供されており、通常、その FPGA 向けに最適化された RTL 記述が出力される。その記述を他社の FPGA に搭載することも原理上不可能ではないが、ライセンス等の問題があるため、対象とする FPGA が複数存在する場合には高位合成ツールを使い分ける必要があり、あまり汎用的とは言えない。

FPGA メーカー以外が提供する、C 言語ベースの高位合成ツールで代表的なものとして、Impulse Accelerated Technologies 社の Impulse C[61] や、NEC 社の CyberWorkBench[62]、ソリトンシステムズ社の eXCite[63] 等が存在する。これらの高位合成ツールはどれも、システムレベル設計に対応しており、ソフトウェア・ハードウェア間の協調検証が可能なツールである。中でも、CyberWorkBench と eXCite については、国内メーカーによる開発が行われており、迅速なサポートが期待できる。CyberWorkBench は、SPRINT-A やはやぶさ 2 の機器設計に適用した実績が存在し[64]、ASIC 開発に加え、殆どの FPGA を対象デバイスとして開発を行うことが可能であるが、その分高価であり、研究用途のみで使用するには少々オーバスペックである。一方 eXCite は、あらゆる FPGA を対象機器としているわけではないものの、ザイリンクス社の Virtex-4qv やマイクロセミ社の RTG4 といった、宇宙機用 FPGA の開発に対応しており、本評価にて必要とする条件を満たしている。この観点から、本評価では高位合成ツールとして eXCite を使用し、高位合成系を用いたハードウェアの評価を行った。eXCite を用いた高位合成系のハードウェア設計フローを図 3.40 に示す。

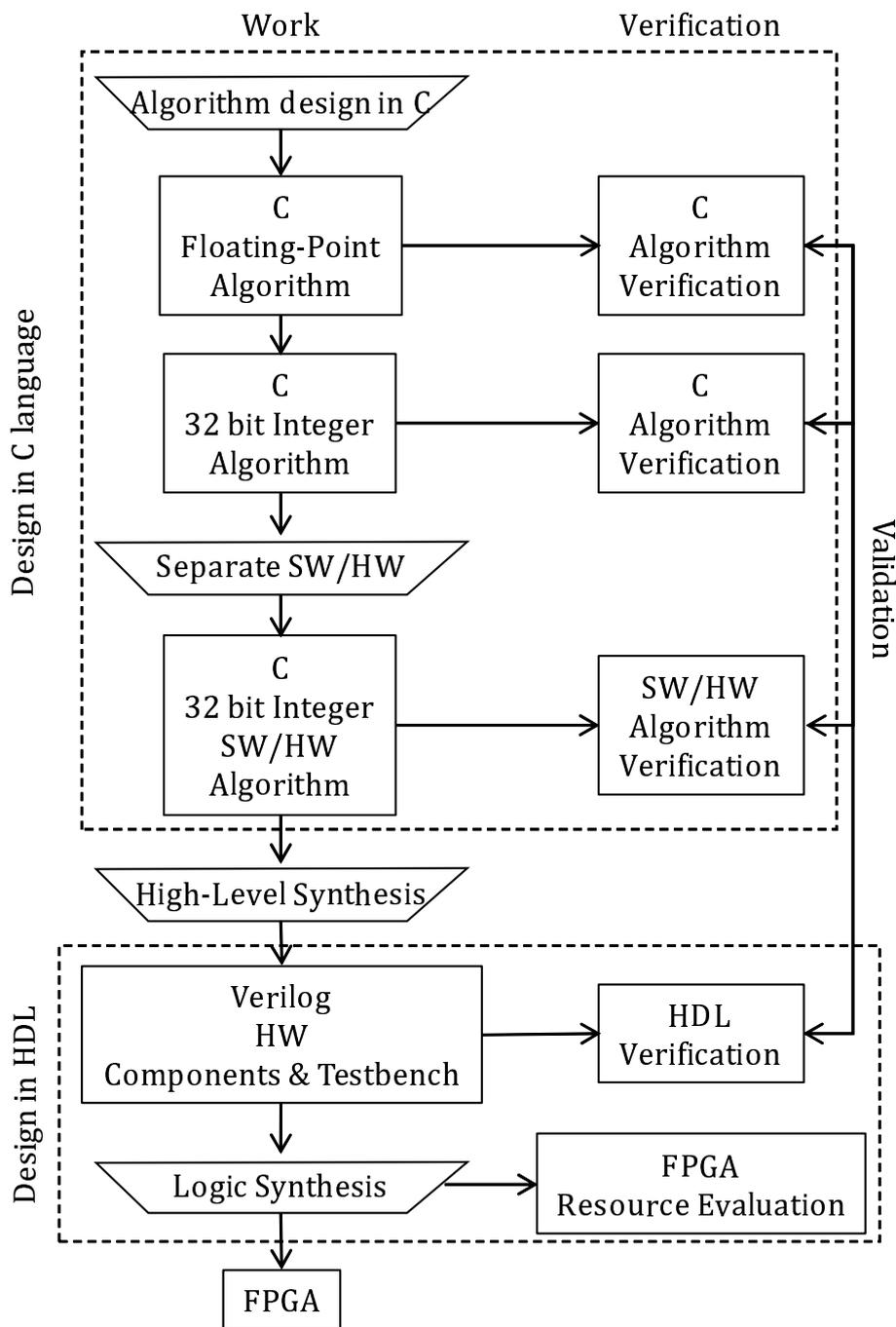


図 3.40 高位合成を用いた設計フロー

3.4.2 設計法の評価

本節では、図 3.40 に示した設計法が、HDL による設計法に比べ、性能面を保ちつつ、設計期間の短縮が有効であるかの検証を行う。今回、比較的手設計が容易な画像処理アルゴリズムとして、8 bit グレースケール、64×64 pix の画像データを入力とし、それに対して 3×3 の全方向微分フィルタを施し、結果となる画像データを出力するアルゴリズムを検証対象とした。このフィルタリングを行うアルゴリズムを C 言語で記述したものを図 3.41 に、このアルゴリズムの入力とその出力例を図 3.42 に示す。

このアルゴリズムについて、高位合成を行い RTL 検証を行ったものと、Verilog-HDL を用いて RTL 設計・検証を行ったものについて、回路性能と生産性について比較する。C 言語記述のフィルタ処理は、並列化を考えず逐次処理で設計を行ったため、Verilog-HDL 記述においても同等の動作となるよう設計した。

ここで、対象 FPGA、評価に用いたツールをそれぞれ表 3.11、表 3.12 に示す。また、評価環境は表 3.2 に示した、汎用 PC 上で行った。

高位合成系を用いた場合と、Verilog-HDL でそれぞれ設計を行ったときの、実際に記述したソースコードの行数（空行、コメント行を含まず）と、論理合成を行った際の回路面積（LUTs）、最大動作周波数を表 3.13 に示す。荒い評価ではあるものの、Verilog-HDL

```
#include <stdio.h>

#define IMAGE_SIZE 3966

void filtering(
    unsigned short *image,
    short *image2)
{
    unsigned short i,j;
    unsigned short k,l,m;

    for (i = 0; i < IMAGE_SIZE; i++){
        k = i + 1;
        l = i + 65;
        m = i + 129;
        image2[l] = (short)(image[k-1] + image[k] + image[k+1] +
            image[l-1] - 8*image[l] + image[l+1] +
            image[m-1] + image[m] + image[m+1]);
    }
}
```

図 3.41 C 言語による全方向微分フィルタリング記述



(a) 入力画像

(b) 出力画像

図 3.42 全方向微分フィルタリングの入力と出力例

表 3.11 対象 FPGA

提供元	Microsemi
Family	RTG4
Device	RT4G150_ES-CG1657M
Package	1657 CG
Speed	STD

表 3.12 評価用ツール

C 設計	Microsoft Visual Studio
高位設計	Y Explorations eXCite
論理合成	Microsemi LiberoSoC
HDL 検証	Mentor Graphics ModelSim

での設計に比べて、高位合成系での記述量はおよそ 1/2 程度であり、記述時間が同等であれば、高位合成は RTL 設計の手法に比べて倍の速度で設計が可能であるといえる。実際には、動作のみを記述すればよい高位合成系に比べて、RTL 設計においてはアルゴリズムの複雑さが記述内容に影響するために、処理内容の煩雑な画像処理においては、更に時間が掛かるものと予想できる。

また、回路面積も高位合成系では 2 割程度下回っており、これは読みやすさ・書きやす

表 3.13 高位合成系と Verilog の比較結果

	高位合成系	Verilog-HDL
記述量 (行/文字数)	63/1404	110/2197
回路面積 (LUTs)	252	296
最大動作周波数 (MHz)	178.3	211.4

さ重視の人手による RTL 記述よりも、高位合成時の最適化が行われた事で回路面積が小さくなったと考えられる。一方、最大動作周波数においては 1 割程度下回る結果となったが、実用においては大きな問題ではないと考える。

これらの結果から、高位合成を用いた FPGA の設計手法は、RTL 記述での設計手法に比べて遜色ない性能を保ちつつ、設計期間の短縮が可能であるといえる。

3.4.3 提案手法の宇宙機搭載性評価

本節では、前節までに述べた高位合成によるハードウェア設計法を、提案した二種の地形相対航法手法に適用し、表 3.11 に示した宇宙機用の耐放射線 FPGA に対して、提案アルゴリズムが消費するリソース量の規模感、処理速度を確認するための評価を行った。評価は、計算機上でのハードウェア評価と、シミュレーションから成る。具体的には、図 3.40 に示した設計フローを用いて、各検証段階の結果に差異がないか確認を行い、差異があった場合には問題点を確認し、修正を行ってゆく。それらに差異がないことを確認した後は、ハードウェア部の論理合成を行い、リソースの規模感を把握するといった手順をとることで、C 言語によるアルゴリズム設計の結果とハードウェア記述言語での動作シミュレーション結果との検証・修正を迅速に行い、評価結果の確実性の向上を狙った。

位置推定法の搭載性評価

3.2 節で提案した位置推定法について、宇宙機用 FPGA への搭載性を評価した。処理アルゴリズムについては、3.2 節の検証で使用した C 言語による記述と同様のものを使用し、高位合成により生成された HDL から、回路規模とその処理時間をシミュレーションにより取得した。処理のシミュレーションに際し、入力パターンによる処理時間の差異を確認するため、画像あたりのクレータ密度が 30 個の入力を前提として、マッチング探索の初期位置と誤検出率の異なる 4 ケースの点パターンと、マッチングが成功しないワーストケースとして作成した点パターンを用い、それぞれに対して図 3.40 で示した評価フローを適用した。この 5 ケースはそれぞれ、探索の初期位置となる IMU 推定位置 q_{IMU} が真値と同等なもの (1)、位置は (1) と同等だが、IMU 予想位置が真値から 2000 pix ず

れているもの (2), 更に誤検出率が高く, これらと同様なもの (3),(4), 全探索を行うワーストケースとして, データベース上に存在しない点パターン (5) となっている。

提案アルゴリズムの検証を行った結果を, 表 3.14, 表 3.15 に示す。このとき, 設計中の各段階での出力結果は, 表 3.14 に示した, 推定位置・回転角・スケールの全ての出力について同等な結果を得ており, C 言語によるアルゴリズム記述から, RTL による回路記述まで一貫した処理を行えることを確認している。

まず, 消費リソースについて, 表 3.15 を見ると, RAM64x18 以外のリソース使用量は全体の 3 割程度, 動作周波数は 44.8 MHz であり, 抽出部分を除いた位置推定部分のみの値としては十分なものとなっている。一方, 内部メモリは 59 % 消費しており, その原因としては, データベース配列を FPGA 内部に搭載する事により, リソースを大きく圧迫していると考えられる。これは, FPGA 外部のメモリにデータベースを格納することで解決可能だが, 外部メモリにアクセスすることで処理速度が遅くなる事が予想されるため, 最適化にあたっては注意が必要と考える。

続いて, 処理時間については, 表 3.14 を見ると, ワーストケース以外の全ケースでは, 500 ms 以内の速度で処理が行えている。IMU 予想位置をずらしたものについても, ずらしていないものと比べて, 数十から数百 ms の処理時間の伸びはあるものの, 十分リアルタイム性があるといえる。また, ケース (1) と (3) においては, どちらも予想値が真値であるにも関わらず, 処理速度には 20 倍近い差がある。この原因としては, ケース (3) においては未検出率が 23 %, 誤検出率が 28 % とどちらも高く, 当アルゴリズムでは点パターン中の線分を逐次処理するため, 正解となる線分の発見が遅れ, 処理速度が遅くなってしまったと考えられる。

ワーストケースにおいては全ての線分の組み合わせにおいてマッチングの確認を行うため, 最も遅く, 24 秒近い処理時間となってしまっている。この問題に対しては, 時間や線分の組み合わせ数で処理を打ち切るなどの処理を行い, 新たな地点で再度撮影し, 位置推定を行うことで, 実用に際しても十分対処が可能であると考えられる。

また, これらの結果は, 回路面積と計算時間のバランスを考え最適化した結果であり, 回路面積を犠牲にすれば, 並列化等の処理から処理時間は速くなり, またその逆によって面積を削減可能である。したがって, 実際の搭載時にはミッションの要求や対象 FPGA の使用可能な回路規模に合わせて, 調整を行うことが可能である。

速度推定法の搭載性評価

続いて, 3.3 節の速度推定法について, 宇宙機への搭載性評価を行った。今回の評価はワーストケースの検討と位置付け, 評価に際してビット最適化やメモリ最適化等の特別な効率化を図る事は行っていない。特に, 近年高速フーリエ変換 (fast Fourier transform; FFT) の計算法の高速化・効率化が図られており, これらの採用次第で結果が大きく変わ

表 3.14 処理時間

ケース番号	1	2	3	4	5
真の位置	(2191,2236)	(2191,2236)	(191,2630)	(191,2630)	無し
予想値	(2191,2236)	(191,236)	(191,2630)	(2191,4630)	(0,0)
回転	8.12 deg	8.12 deg	5.27 deg	5.27 deg	無し
スケール	+3.80%	+3.80%	+1.28%	+1.28%	無し
未検出率	25%	25%	23%	23%	100%
誤検出率	7.8%	7.8%	28%	28%	100%
推定位置	(2191,2236)	(2191,2236)	(191,2630)	(191,2630)	推定不可
推定回転角	8.19 deg	8.19 deg	5.32 deg	5.32 deg	推定不可
推定スケール	+3.70%	+3.70%	+1.11%	+1,11%	推定不可
実行時間@50MHz	21 ms	103 ms	431 ms	689 ms	24026 ms

表 3.15 消費リソース

Logic	Used	Available	Utilization
LUT	50658	151824	33.37 %
SLE	34714	151824	22.86 %
RAM64x18	125	210	59.52 %
RAM1K18	10	209	4.78 %
MACC	61	462	13.20 %
動作周波数	44.8MHz		

ると予想されるが、今回はごく標準的な Cooley-Tukey 型の FFT アルゴリズムを用いて検証を行った。

入力画像は図 3.43 に示す、3.3.5 項で使用した 3 つの画像を用いた。また各段階での出力、消費リソース、処理時間を検証した結果を表 3.16、表 3.17、表 3.18 に示す。

まず、設計中の各段階での出力結果について、表 3.16 を見ると、全てのケースにおいて、浮動小数点型の C 言語で設計したオリジナル記述における処理結果から、高位合成後の HDL 検証結果まで差異なく出力が一致している。このことから、高位合成によって作成された RTL 記述は、オリジナル記述と一致した処理を行っているといえる。

次に、消費リソースについて、表 3.17 を見ると、内部メモリ (RAM1K18) 以外のリソース使用量は全体の 1 割強であり、また動作周波数は 50.6 MHz と十分に満足できる値

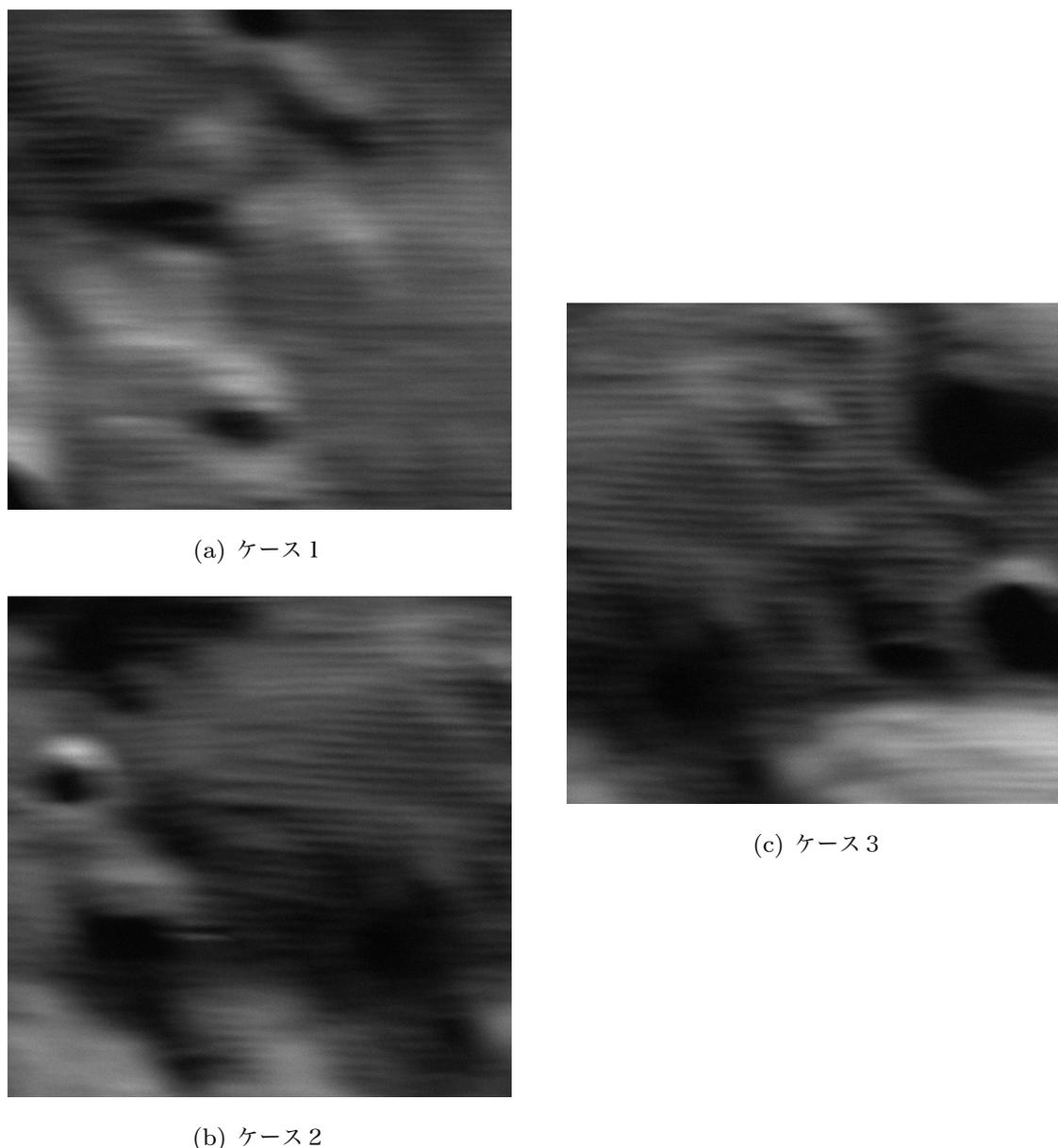


図 3.43 速度推定法の搭載性検証に用いた入力画像

に収まっている。一方、内部メモリは 64 % 消費しており、その原因としては、FFT の中間データの配列が分割画像のピクセルサイズ分存在する（今回の場合は 65536 個）事により、リソースを大きく圧迫していると考えられる。この問題に対しては、中間データを外部メモリに格納するか、画像分割数を増やす事で解決する事が出来るが、前者では処理速度が遅くなる事、後者では推定出来るブレ幅の上限が低くなる事が予想されるため、最適化にあたっては注意が必要と考える。

続いて、処理時間については、表 3.18 を見ると、50 MHz での動作換算において、約 2.2 秒で処理が実行できている。本アルゴリズムは入力に依存する処理が殆ど存在しない

表 3.16 出力検証結果

	Case1	Case2	Case3
浮動小数 C 検証結果 [mm/s]	47.87	50.01	52.15
固定少数 C 検証結果 [mm/s]	47.87	50.01	52.15
分割 C 検証結果 [mm/s]	47.87	50.01	52.15
HDL 検証結果 [mm/s]	47.87	50.01	52.15

表 3.17 消費リソース

Logic	Used	Available	Utilization
LUT	16897	151824	11.13 %
SLE	11659	151824	7.68 %
RAM1K18	134	209	64.11 %
MACC	55	462	11.90 %
動作周波数	50.6MHz		

表 3.18 処理時間

	Case1	Case2	Case3
サイクル数	113,181,312	112,995,225	113,021,838
実行時間 [ms] (50 MHz)	2264	2260	2260

ため、各ケースに殆ど処理時間に差異がない結果となっている。2.2 秒という処理時間に関しては、分割した画像を逐次処理でフーリエ変換するなど、アルゴリズムに対して並列化等の最適化を行わずに実行した結果であり、今後の最適化次第では 1 秒を下回る値となる事も十分に期待できる。一方で、並列化を行う事で処理時間は速くなるが、回路面積は増大するため、バランスの取れた最適化を行う必要があると考える。

3.5 まとめ

現状入手可能な宇宙機用計算機にて、自律的かつリアルタイムに実行可能な地形相対航法の手法として、位置と速度を推定するアルゴリズムについて新たに提案した。

位置推定手法については、面を規定する最小要素である線分と、事前の誤差情報によって画像のアフィン変換の範囲を制限可能な点に着目し、性能を担保しつつ計算規模の削減

を図った。その適用範囲と性能について、月面のクレータ地形を対象に、DEMによって生成された画像を使用することで評価を行った。結果としては、クレータ密度が 30,40,50 の場合において、十分な精度で位置推定を実施可能であることを示した。また、処理時間に関しては、対応点数の足切りを設けることにより、正解が存在すれば最悪処理時間のおよそ半分程度の時間で位置推定が可能となることを示した。これらのことから、クレータが画像範囲内で数十個程度の密度であり、また誤差が一般的な地形相対航法で考えられる範囲であるとき、当手法は精度を損なうことなく高速に位置推定が可能である。その最悪計算量は $O(m^2n)$ であり、一般的な点マッチング手法と比較しても、十分に高速である事を示した。

速度推定手法については、単一画像中に発生する動きブレに含まれる移動量に着目し、従来の複数枚画像を使用する手法より計算速度と規模の削減を図った。その適用範囲と性能について、月面画像にブレとノイズを付与したシミュレーション画像と、ジオラマを用いて実際に撮影した動きブレ画像を使用することで評価を行った。結果としては、動きブレ画像に多くのノイズが乗っていても、画像上において 7-20 pix の移動量であり、被写体までの高度情報が適切に与えられているとき、当手法は高精度で速度推定が可能であることを示した。また、ノイズが存在せずハイコントラストな地形を対象とする場合には、50 pix 程度の移動量まで検出が可能である。

また、実際の宇宙機用 FPGA への搭載性についても、高位合成技術を使用することで評価された。結果として、それぞれについて性能を損なわない範囲で実際に搭載可能であることを示し、また FPGA においてもリアルタイム性を保ったまま処理が可能であることを示した。

これらの結果により、現在の低計算処理リソース環境下において、提案したアルゴリズムが現実的なある種の適用範囲内であれば、実際の宇宙機用 FPGA 上においても地形相対航法タスクが自律的かつリアルタイムに実施可能であることを示した。一方で、地形相対航法タスクに対して、現状より高度な認識機能やより低密度なエネルギー消費が要求される、土星圏などの遠方天体を目指すような将来ミッションについて考えると、各アルゴリズムの性能について本検討の適用範囲から外れる可能性が高い。その解決のためには、より処理リソースを消費する、高度な機能を持つアルゴリズムの搭載を考える必要があるが、太陽から遠く離れるために計算処理リソースも更に低くなることと予想される。したがって、現在の計算機アーキテクチャとは異なる発想による処理系における、地形相対航法の実施が研究の発展先として示唆される。

Notes

*1 スペクトルに対するフリークエンシと同様，ケプストラムの変数の次元

Chapter 4

将来の低リソース探査に向けた 地形相対航法

4.1 あらまし

自律的な地形相対航法の実現にあたっては、地上よりも電力の制約が厳しい宇宙機における計算機環境の下、物体認識・識別や追跡、照合といった高度なコンピュータビジョンタスクの実施が求められる。この状況下で、現在入手可能である低計算リソースな宇宙機用計算機において、搭載アルゴリズムの計算規模を削減する方向から、自律的な地形相対航法を実現することを企図したのが3章の内容であった。ここで提案した二種類のアルゴリズムについては、確かに宇宙機用計算機に実装可能であり、また一定の性能を持つことは既に示した。しかしながらその結果は、計算処理リソースが制限されている現状の計算機環境に対して、より高度な画像処理技術を実装することは難しいということも同時に示している。

将来の無人機による着陸探査について考えると、現在の月火星を中心とする探査より遠く、木星圏、土星圏やその衛星といった、太陽から遠く離れることで現在より低エネルギー環境下での活動を行うものや、崖や溪谷のような険しい地形にもロバストに着陸を行うものなどが予想される。したがって、宇宙機の計算機環境に対する電力リソースはより苛烈に、地形相対航法のアプリケーションはより複雑になる。

このような、計算機の低消費電要求と航法アプリケーションの高度複雑化という課題について、計算機アーキテクチャの面から考える。

我々が日常的に使用する計算機において、CPUが主たる演算コアとして使用されている。そのようなノイマン型 (von-Neumann) の計算アーキテクチャにおいては、クロック駆動による逐次的な命令実行をもって処理が実施され、その性能 (単位時間あたりの処理

命令数) は次式で表される。

$$Performance = Frequency \times IPC \quad (4.1)$$

すなわち、性能向上にあたっては動作周波数 (*Frequency*) あるいは、サイクル毎命令数 (instructions per cycle; *IPC*) の改善が必要となる。IPC の向上には命令セット (instruction set) の面から向上を図らなければならないため、近年の性能向上の主たる要因は、プロセッサの動作周波数の増加が担っている^{*1}。

この性能向上のための動作周波数の増加に伴い、プロセッサの消費電力も同様に増加する傾向にある。プロセッサの消費電力は次式で示される。

$$Power = \alpha N_{active} C_{node} V_{node}^2 \times Frequency \quad (4.2)$$

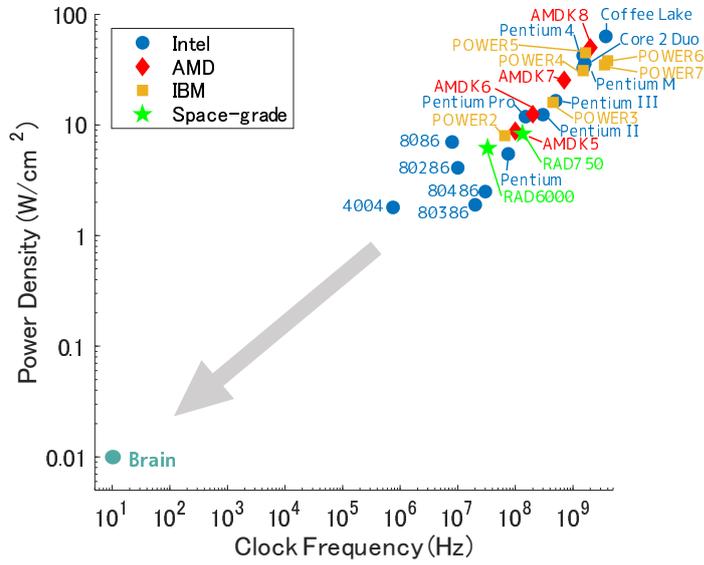
ここで、 α は動作率、 N_{active} は稼働するノードの数、 C_{node}, V_{node} は各ノードの負荷容量と電圧である。したがって、性能向上のために動作周波数を増加させた時、消費電力の増加に対して直接影響を与えることとなる。

これまでに発表された、主たる民生/宇宙用 CPU の電力密度とクロック数の傾向を図 4.1 (a) に示す^{*2}。各メーカーのプロセッサについて、世代を経るごとに、すなわち性能の向上のために、動作周波数が増加し、それに伴って電力密度も上昇する傾向が存在していることがわかる。また、宇宙機用のプロセッサにおいても同様に、世代が上がるにつれ動作周波数と共に電力消費も大きくなっている。

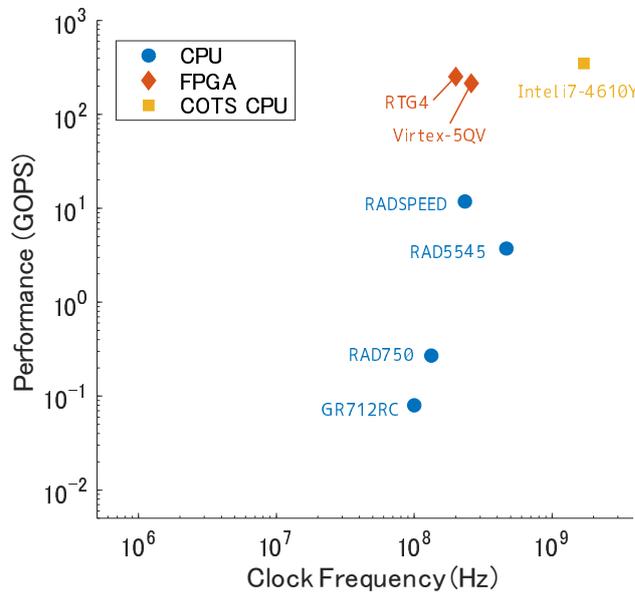
宇宙機用計算機における性能とクロック数の傾向を図 4.1 (b) に示す^{*3}。民生の CPU に比べて、最新の CPU (RADSPEED, RAD5545) でも動作周波数、性能共に 1 桁程度低い値に留まっている。また FPGA はベンチマーク上では民生 CPU と同等のスペックを発揮しているように見えるが、これはベンチマークにおいて、加積算の演算のみを並列化して行っていることに起因するものであり、実際のアプリケーションを実装するにあたっては、より低い性能となることが前の章における検証から分かっている。

このように、計算機の性能向上において消費電力の増加は避け得るものではなく、宇宙機用計算機においても同等のアーキテクチャを使用するにあたっては同様である。したがって、高度な自律地形相対航法のため、低消費電力要求の存在下でより複雑な画像処理タスクを実施するには、既存より異なる方向性の計算機アーキテクチャにおける処理が、解決の方策として浮上する。

図 4.1 (a) の左下に示したように、生物の脳神経はノイマン型計算機と比較して、電力密度と動作周波数が著しく低い。これは、脳神経の処理アーキテクチャが非同期で並列に



(a) 脳と CPU の電力密度とクロック数の傾向



(b) 宇宙機用計算機の性能とクロック数の傾向

図 4.1 計算機の電力・性能とクロック数の傾向

動作することに起因する。そのような脳神経機能を模倣する、ニューロモーフィックコンピューティング (neuromorphic computing) と呼ばれる新たな計算アーキテクチャが存在する。従来の計算機アーキテクチャとニューロモーフィックな計算機アーキテクチャとの間の処理における特徴の違いを表 4.1 に示す。

ニューラルネットワークを集積回路にて表現することは、1990年代にホップフィールド型ネットワークモデルをLSI化することから始まり、ボルツマンマシンや多層パーセプトロンの集積回路化が進められてきた。2010年代に入り、大規模なスパイクングニューラルネットワーク (spiking neural network; SNN) のシミュレーション [65] が、脳機能の理解に有用であると認識されたことにより、SNNを回路的に構築する方式の、ニューロモーフィックコンピューティングへの期待が、高まってきている。

このような観点から、計算機上で実施されるSNNのシミュレーションの効率を上げるため、演算回路としてSNNを実装するプロジェクトが多く立ち上げられている [66, 67]。これらニューロモーフィックプロセッサの開発においては、アナログ/デジタル回路上でスパイクングニューロンの通信を実施することで、電力効率や処理効率を向上させた処理システムの構築を目的としている。

一方で、このようなプロセッサチップの開発理由は、SNNのシミュレーションに用いることを目的としたものである。現状では脳における神経回路の挙動やダイナミクスを再現し、神経科学的な脳機能の理解を目的とした使用例が殆どとなる。したがって、神経細胞網の挙動を確認するための実験装置としてではなく、情報処理を行うための低消費電力な処理プロセッサとして、そのアーキテクチャ上でアプリケーションを動作させる観点からの研究例は未だ多くない。

表 4.1 従来の計算機と神経模倣型計算機との間の特徴差

従来の計算機	高クロックでの動作 (> GHz) 同期的処理 (global clock) CPU とメモリ間の通信ボトルネック 電力消費量大 論理回路として動作
神経模倣型計算機	低クロックでの動作 (< KHz) 非同期的処理 (clock free) 統合された処理回路とメモリ 電力消費量少 Neural network としての動作

ニューロモーフィックな計算機は、従来の計算機とは異なり論理演算には向いていないが、画像認識処理などの AI 的動作に適性があり、超低消費電力での動作についても期待されている。これは、低エネルギー環境下で高度な画像処理を行う、地形相対航法の条件に合致する。一方で、ニューロモーフィックコンピューティングのアーキテクチャ上において、どのようにアプリケーションを構成するかについては、黎明期の段階であり、宇宙分野においては前例がない [68]。

そこで本章では、ニューロモーフィックな計算機アーキテクチャにおいて、地形相対航法に使用される画像処理アプリケーションとして、クレータ識別タスクを採用し、これを具体的に適用することによって、低消費電力性や性能においてメリットが得られることを確認・検証する。

4.2 ニューロモーフィックコンピューティング

近年の計算機環境と機械学習技術の発展は目覚ましく、画像処理分野における物体認識等のタスクの性能は、向上の一途を辿っている。一方で、我々のような生物について考えると、視覚による認識においては、エネルギーや処理効率の点で格段に優れている。生物の脳神経における電気的な活動は、計算機における処理プロセッサの動作と比較して非常に疎なものであり、したがって消費電力量も劇的に少ない。機能面において、人間は日常的に目にする物体や、耳に入った他人の声を容易に識別することが可能である。現在の計算機と機械学習アルゴリズムにおいても、人間と同等以上の識別タスクを実施可能である一方で、タスク実行時に必要とされる計算能力や消費電力量は膨大となる。具体的には、Google は動画像から猫の顔を識別するために、16000 個のプロセッサと約 100 kW の電力を必要としている。一方で、人間の脳は数十 W で動作する [69]。また、一部のハエの視神経は約 200 Hz 頻度での視覚情報を得ることが可能である。

生物の脳神経網における、認知機能の達成原理は未だ詳細に解明されていないものの、そのような神経生物学的な計算システムと現代の計算機との間に存在する差は、その処理方法の違いに起因すると目されている。典型的な計算機では、同期クロックに基づく論理演算が用いられる。脳神経網では、神経細胞 (ニューロン; neuron) のような、非同期的かつ並列で分散されたイベント駆動な構成により機能している [70]。

ニューロモーフィックコンピューティングとは、このような神経生物学的な原理を計算機に適用することで、脳神経網のような処理を可能とする機械を構築する技術分野である。脳内では、ニューロン間の通信は、活動電位 (スパイク; spike) の列を下流のニューロンに伝達することによって行われる。これらの個々のスパイクは時間的に疎であるので、各スパイクは高い情報量を持つ。ニューロモーフィックコンピューティングにおいて

は、このニューロンの振る舞いを模倣することで、低消費電力かつ高速に計算を行うことを可能とする。

近年のニューロモーフィックコンピューティングに使用される演算コアは、生物学的な単一ニューロンのダイナミクスを、数学的にモデル化したスパイクングニューロン (spiking neuron) と、ニューロン間の接続によるアーキテクチャと、その間の情報伝達について記述するネットワークモデル (SNN) を回路的に実装したものである。

本節では、ニューロモーフィックコンピューティングにおける処理方法である SNN について、特に本研究で使用されるモデルとそのアーキテクチャ上で動作させる機能の学習法、またその実施方法について概説する。

4.2.1 スパイクングニューラルネットワーク

SNN は、生物学的な神経のダイナミクスを記述する、スパイクングニューロンを基本の処理単位として構成されている。ここでは、ニューロモーフィックプロセッサにおいて実装されるスパイクングニューロンのモデルと、接続に用いるシナプスのモデル、またそれらを接続したネットワークのモデルについて示す。

神経細胞における処理の概要

ニューロンは脳神経網における認知や思考、行動などの機能を表現するための基本処理要素であり、人間の脳においては、約 1 億のニューロンとその 4 桁上の神経接続が存在している。このように巨大で複雑なシステムであるにもかかわらず、脳内のニューロンはその構造によって、神経接続を介して他の細胞に迅速かつ正確に信号を送ることが可能である。

生物学的なニューロンの模式図を図 4.2 に示す。典型的なニューロンは、樹状突起 (dendrite) や細胞体 (soma)、および軸索の三つの部位で構成される。樹状突起は、他のニューロンからの刺激を受け、神経信号を細胞体に伝達する。細胞体は核が位置するニューロンの本体であり、蓄積された総入力がある閾値を超えたときに、信号を出力する非線形の処理装置として機能する。この出力信号は、軸索を介して他のニューロンに伝播する。殆どのニューロンは軸索を一つしか持たないが、軸索の末端を分岐させることにより、他の多くのニューロンへと接続する事が可能である。また、あるニューロンから別のニューロンへの信号伝達は、これら二つのニューロンの間の接合部であるシナプス (synapse) にて発生する。シナプスは、信号を送るシナプス前ニューロン (presynaptic neuron) およびそれを受け取るシナプス後ニューロン (postsynaptic neuron) からなる。

ニューロン間で伝播する信号は、短い電気パルスである。この活動電位はスパイクと呼

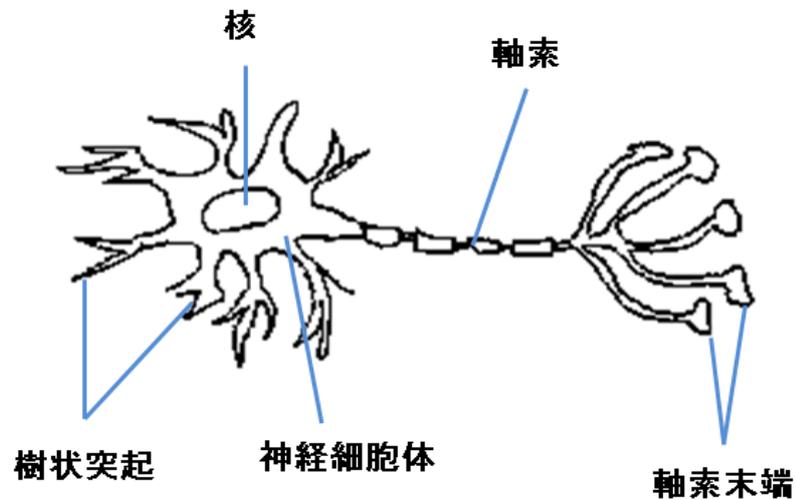


図 4.2 ニューロンの模式図

ばれ、約 100 mV の振幅を持ち、1 から 2 ms の間持続する。処理信号となるスパイクの大きさと持続時間は生物種間で大きく異なるものの、形状としては同様である。したがって、スパイクの形態そのものは殆ど情報を持たないとされる。この場合、処理情報を表現するのはスパイクの頻度とタイミングである。単一のニューロンによって生成された一連の活動電位はスパイク列 (spike train) と呼ばれ、離散時間で起きるバイナリのイベントとして見なすことができる。このバイナリのイベントにおける頻度とタイミングによって、処理情報がエンコードされる。

スパイクがニューロンへと到達した際、膜電位と呼ばれる細胞体の内部と外部との間の電位差の変化を引き起こす。ニューロンの膜電位は、入力がない場合は静止電位に留まり、スパイクが到達すると、直ちにシナプスの種類に応じて膜電位が増加、あるいは減少し、その後静止電位へと戻る。単一のスパイクによって引き起こされる膜電位変化の状態は、シナプス後電位 (postsynaptic potential; PSP) と呼ばれる。同じニューロンの、異なるシナプスに到達するスパイクは、シナプスの接続の強度である、シナプス効力 (synaptic efficacy) に応じて異なる振幅の PSP を有する。また、スパイクの到達直後には、ニューロンがそれ以上の刺激に対して応答しない期間が存在する。ニューロンにおける電位が変動しないこの時間は、スパイクを発生させることのできない絶対不応期 (absolute refractory period) と呼ばれる。

これら膜電位や PSP、スパイク列の神経ダイナミクスは、すべて時間依存性をもつ。一方で、機械学習や深層学習に代表される、人工ニューラルネットワーク (artificial neural network; ANN) のニューロンモデルは、スパイクニューロンのダイナミクスにおい

て、時間情報を抜いた発火頻度を表現している。入出力の表現と計算におけるこの差異が、ANN の機能を SNN に変換する事を困難にしている。

スパイクングニューロンのモデル

神経細胞のダイナミクスをモデル化したスパイクングニューロンは、単純なものからより生物学的に確からしいものまで多くの種類が存在する。本研究では、ニューロモフィックコンピューティングに対するアプリケーションの構築を目的とするため、ニューロモフィックプロセッサの実装において最も使用されるモデルである、漏れ積分発火 (leaky integrate-and-fire; LIF) ニューロンモデルを扱う。他のニューロンモデルについては Izhikevich[71] に詳しい。

実際のニューロンにおける、種々のイオン (ion) の流れを考慮せず、膜電位が閾値に達するまで、入力電流を時間的に積分するモデルが積分発火 (integrate-and-fire; IF) ニューロンモデルである。IF モデルにおいては、入力が発生しない場合にも膜電位が一定に保たれ、ニューロンの挙動から乖離するために、膜電位の漏れ^{*4}を考慮したモデルが LIF ニューロンモデルである。

ニューロンにおける膜電位 V の発生においては、膜容量 C_m と膜抵抗 R_m からなる RC 回路に単純化することができ、これは入力電流 I によって動作する。この RC 回路として動作する細胞膜の模式図を図 4.3 に示す [72]。入力がない休止状態では、膜電位 V は静止膜電位 V_{rest} と同じ電位に留まっており、電流がニューロンに入力されると、電流 $I_C(t)$ でコンデンサを充電し、電流 $I_R(t)$ で抵抗を介して放電する。入力電流が停止すると、コンデンサの電荷は抵抗を介して漏れ、 V_{rest} まで減衰する。このニューロンのダイナミクスは次式として表される。

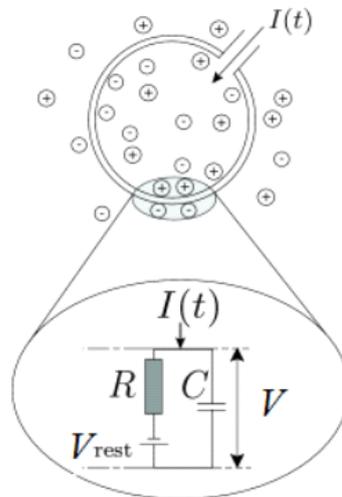


図 4.3 RC 回路として動作する細胞膜の模式図

$$\begin{aligned}
 I(t) &= I_R(t) + I_C(t) \\
 &= \frac{V - V_{rest}}{R_m} + C_m \frac{dV}{dt}
 \end{aligned}
 \tag{4.3}$$

標準的な LIF ニューロンモデルでは、両辺に R_m を乗することで次式のように変形される。

$$\tau_m \frac{dV}{dt} = -(V - V_{rest}) + R_m I(t)
 \tag{4.4}$$

ここで、 $\tau_m = C_m R_m$ は膜電位の時定数である膜時定数 (membrane time constant) である。膜電位は閾値 V_{th} に達すると発火し、接続された後続のニューロンにスパイクを出力する。その後、直ちに膜電位は静止膜電位 V_{rest} に設定される。また、前述のように発火後はニューロンがそれ以上の刺激に対して応答しない期間である、不応期 τ_r が存在する。

シナプスのモデル

発火したニューロンで生じたスパイクは、後続のニューロンへとシナプスを通して伝搬する。生理学的にはこの伝搬の過程において、シナプス前ニューロンの末端から放出された神経伝達物質が、接続されたシナプス後ニューロンに存在する受容体にて結合することにより、後続のニューロンにおける入力電流が発生する。更にその間にもより詳細なダイナミクスが存在するものの、それら全てのモデル化と処理にあたっては計算量の増大が危惧されるため、生理学的に妥当かつ詳細なシミュレーションの実施を除いては、簡易的なモデルが使用される。これらのシナプスにおける複雑な情報伝達の過程の存在から、ニューロンで発生したスパイク列は、後続のニューロンに直接伝搬されるわけではなく、シナプスフィルタ (synaptic filter) と呼ばれる時間的なフィルタを介して伝搬される。

電流ベースのシナプスにおいて、シナプスに対する入力電流は次式のように表される。

$$I(t) = w s_{syn}(t)
 \tag{4.5}$$

ここで、 w はシナプス強度^{*5}、 s_{syn} は神経伝達物質の放出量あるいは結合量などをモデル化したシナプスモデルである。

シナプスモデルは生理学的に妥当なものから簡略化されたものまで複数存在するものの、中でもシナプス後電流の挙動をよく近似できており計算が簡素である、指数関数型のモデルが SNN においてよく使用される。シナプス前ニューロンの出力となるスパイクは、シナプス後ニューロンにおける電流パルスの延長を誘発する。これは、シナプス前

ニューロンのスパイクにローパスフィルタをかけたものとみなすことができ、そのインパルス応答は一次の指数関数でよく表現できる。この指数関数型のシナプスモデルは、シナプスの時定数 τ_s を用いて次式で表される。

$$f(t) = \frac{1}{\tau_s} \exp\left(-\frac{t}{\tau_s}\right) \quad (4.6)$$

時刻 $t = t_0, t_1, \dots, t_n$ にスパイクが供給されるとすると、この関数を時間的なフィルタとして、全てのスパイクに適用することでシナプスモデルが表現できる。

$$s_{syn}(t) = \sum_{t_k < t} f(t - t_k) \quad (4.7)$$

このモデルは、ニューロン間の接続が一つ対一つに対応している場合の表現である。一般的な SNN では、シナプス後ニューロンは多数のシナプス前ニューロンと接続する形でネットワークを構成しており、また複数のニューロンから出力されたスパイクの、後ニューロンに対する各々の入力は同等ではなく、異なるシナプス強度をもつ。この時、シナプス後ニューロンへの入力電流 I_{syn} は、接続された各シナプス前ニューロンにおけるシナプス強度と、入力スパイクに対してシナプスモデルを適用したものの総和で表現される。

このようにスパイクニューロンとシナプスモデルを用いてネットワークを構成することにより、神経回路のモデルを作成することができる。ニューロモーフィックプロセッサでは、これらの神経ダイナミクスをハードウェアとして実装することにより、計算処理を実行している。

4.2.2 人工ニューラルネットワーク

人工ニューラルネットワーク (ANN) は、近年の画像処理分野において最も成功した領域の一つである。ネットワークの層を深く構築したものは深層学習 (deep learning) としてよく知られている。ANN は、SNN と同様に神経細胞から着想を得た、人工ニューロン (artificial neuron) を計算の基本ユニットとして利用する。^{*6} 本項では、この ANN について概説する。

人工ニューロンのモデル

人工ニューロンは、SNN の文脈上ではニューロンの発火率 (firing rate) モデルとして知られている。これは、スパイクニューロンにおけるスパイク駆動による情報伝達の

過程を、時間要素を除いたその発火率で表現したものであるからである。

人工ニューロンとスパイクニューロンの模式図を図 4.4 に示す。前述のように、スパイクニューロンでは入力となるスパイク列がシナプス強度との線形和をとる形で PSP を誘起し、膜電位が閾値に達すると出力スパイクを生成する。一方で、人工ニューロンでは入力として発火率を受け取り、シナプス強度との総和を取った後に非線形な関数に渡すことで、出力となる数値を得る。

人工ニューロンのモデルは、神経細胞のダイナミクスを簡略化し、状態がバイナリで表現される形式ニューロン [73] の発想が元となっている。その後、この形式ニューロンを複数組み合わせることにより、学習能力を持ちパターン認識が可能な、パーセプトロン (perceptron) [74] が提案されることによってニューラルネットワーク分野の流行を齎した。現在では、一般的に人工ニューラルネットワークと呼ぶ場合には、この発火率モデルを多層にわたって構成したネットワークのことを表す。

発火率とは、単位時間におけるニューロンの発火頻度のことであり、着目する時間窓を T 、出力スパイクの発生時刻を t_k とすると、ニューロンの発火回数 n_λ は次式で表され

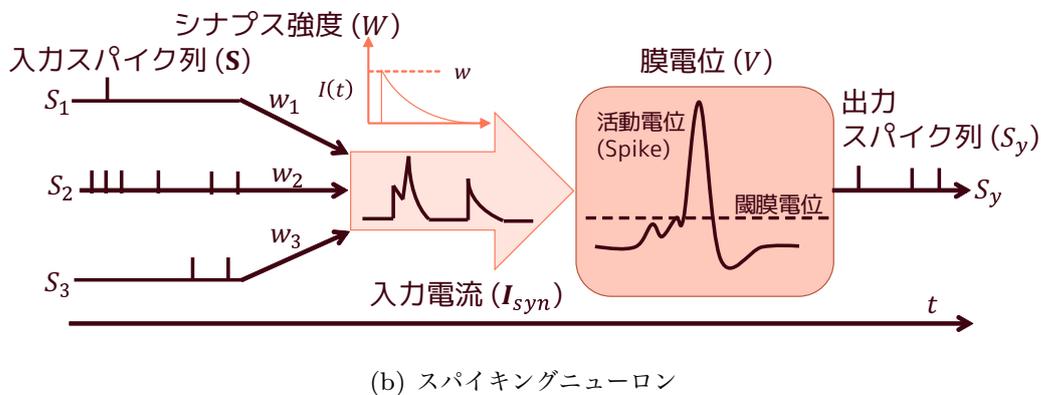
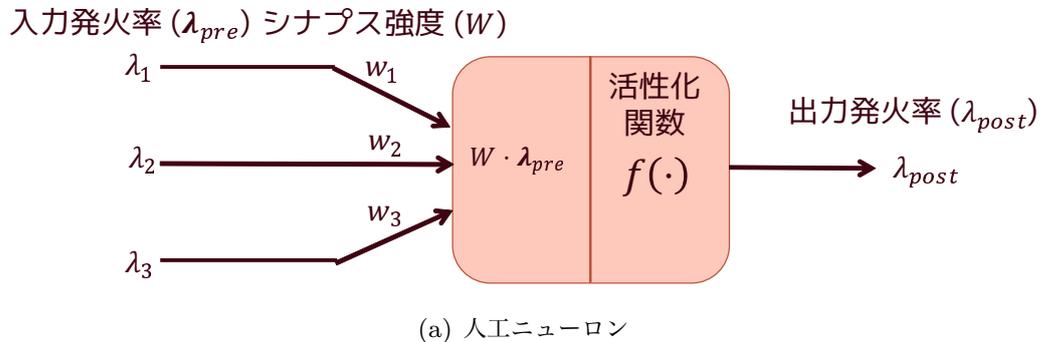


図 4.4 人工ニューロンとスパイクの模式図

る.

$$n_\lambda = \int_0^T dt \delta(t - t_k) \quad (4.8)$$

ここで δ はディラックのデルタである. これにより, 発火率 λ は

$$\lambda = n/T \quad (4.9)$$

と表される.

この発火率を元に, 人工ニューロンは次式で示される.

$$\lambda_{post} = f(W\lambda_{pre} + \mathbf{b}) \quad (4.10)$$

ここで, λ_{post} はシナプス前ニューロンの発火率, λ_{pre} はシナプス後ニューロンの発火率, W はシナプス強度, \mathbf{b} はバイアス項である. また, $f(\cdot)$ は活性化関数と呼ばれ, 任意の非線形関数を適用する. この活性化関数はスパイクングニューロンにおける入力電流に対する発火率の関係である, 応答関数 (response function), すなわち frequency-current 曲線 (F-I curve) を表現している.

従来, 活性化関数として次式のようなシグモイド (sigmoid) 関数が使用されてきた.

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \quad (4.11)$$

これは不応期をもつニューロンは, 入力を大きくすることにより発火率が飽和するという発火特性に類似していることに起因する. また, 近年では正規化線形関数 (rectified linear unit; ReLU) が, シグモイド関数に代わって精度の高い活性化関数として使用される.

$$\text{ReLU}(x) = \max(0, x) \quad (4.12)$$

ReLU 関数は IF モデルのニューロンの応答関数とほぼ一致する.

4.2.3 ネットワークのアーキテクチャ

4.2.1 項では生理学的なニューロンのダイナミクスと, それをモデル化したスパイクングニューロン, シナプス接続のモデルについて, 4.2.2 項ではスパイクングニューロンの文脈での発火率モデル, 一般的には人工ニューロンとして知られているニューロンのモデルについて示した. 本項では, これらのモデルを処理の基本単位として, ネットワークを

構成することによって神経回路を構築する方法について概説する。

畳み込みニューラルネットワーク

現在のニューラルネットワークにおけるネットワーク構造は、単純型細胞による勾配選択性と複雑型細胞によるプーリング (pooling) の機能が、一次視覚野の基本機能を表現しているという発見 [75] が基礎となっている。これを受け、多層で構成される初期のネットワークであるネオコグニトロン (NeoCognitron)[76] が提案された。この多層ニューラルネットワークに対して教師あり逆伝搬法を適用することで、高い認識性能を持たせた畳み込みニューラルネットワーク (convolutional neural network; CNN)[77, 78] が開発された。

CNN のモデルである LeNET のアーキテクチャを図 4.5 に示す [78]。このネットワークは、入力層 (input layer)、畳み込み層 (convolutional layer)、プーリング層 (pooling layer)、および全結合層 (fully connected layer) の四種類の層で構成されている。図 4.5 の左から右にかけて、CNN のメカニズムを詳細に示す。

まず入力画像は、最初の畳み込み層に送られる。畳み込み層に存在する全てのニューロンは、入力と畳み込みカーネルによって決定された各入力ピクセルの重みを受け取る。このとき、人工ニューロンは図 4.4 (a) で示したように機能し、入力と重みとの線形和を取った後に非線形関数である活性化関数に入力される。この結果、入力画像全体をカーネルによって畳み込むと、畳み込み層の出力として特徴マップ (feature map) が構成される。したがってこの例では、最初の畳み込み層に六つの畳み込みカーネルを持ち、結果として六つの特徴マップが出力される。

畳み込み層の特性は、局所的な接続と重みの共有にある。すなわち、畳み込み層のニューロンは、畳み込みカーネルで指定された、前の層の局所的なニューロンとのみ接続しており、また畳み込みカーネルの持つ重みのパラメータは全てのニューロンで共有される。結果として、同じ大きさを持つ全結合のネットワークと比較して、学習によって変動

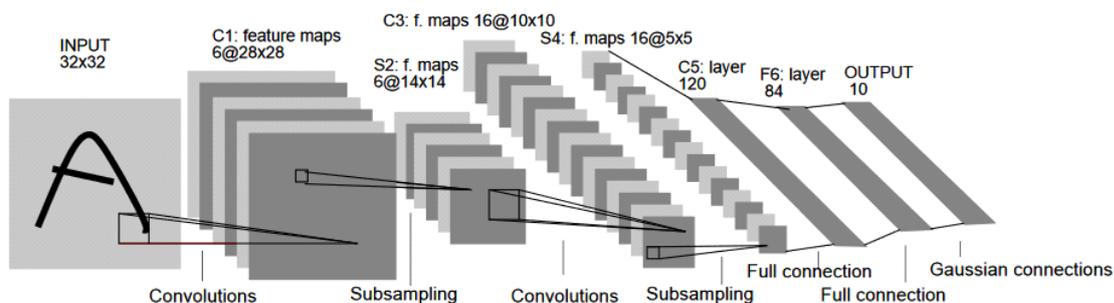


図 4.5 LeNET-5 のアーキテクチャ

する重みの数が大幅に減少する。

続いて、畳み込み層の出力である特徴マップはプーリング層に送られる。プーリング層のニューロンは、重複しない領域の入力に対して、最大/平均フィルタを適用することによって、入力の最大あるいは平均を出力する。この処理によって、プーリング層は入力される特徴マップの空間方向のサイズを圧縮する。

その後、畳み込み層とプーリング層での処理が繰り返される。この畳み込み層とプーリング層の繰り返しによって、ネットワークは深くなり獲得できる表現が増大する。加えて、畳み込み層の学習パラメータの削減とプーリング層のネットワークサイズの削減効果によって、過学習や学習の失敗のような悪影響を抑えることが可能である。

典型的な CNN においては、最後に全結合層が配置され、一つ前のプーリング層の出力となる特徴マップが入力される。全結合層のニューロンは、入力を全て接続し、重みとの線形和を取った後に活性化関数を通して出力される。出力ニューロンにおいて最も強い応答は、入力画像が出力のどのクラスに属するかを示す。図 4.5 の場合には、入力画像のアルファベットがどの文字であるのかを識別している。

ネットワークの学習

本章の序文で述べたように、通常の計算機で SNN をシミュレーションするには ANN よりも多分な計算処理リソースを消費する。ニューロモーフィックプロセッサは SNN に特化しているため、より低消費電力で SNN をシミュレーション可能である。そのようなプラットフォームが提供され始めたことに加えて、神経科学的な見地から SNN は ANN にはない利点があるため、SNN を高効率に学習させるアルゴリズムが研究されている。率直に考えた時、ANN は情報の表現として発火率のみしか使えないことに対して、SNN は発火率に加えて時間方向の情報表現も可能であるために、多様な認知機能の実現という面では有利に思える。しかしながら、実際には SNN の学習は難しく、誤差逆伝搬法という高効率な学習アルゴリズムを使用可能な、ANN を超える性能というところまでは現状至っていない。

現在のところ、SNN を訓練する方針は大きく分けて3つ存在する。

ANN の重みの変換による学習

従来の ANN を学習させたのち、学習した重み (シナプス強度) を SNN に変換・転送することにより、SNN に機能を持たせる手法

ANN 特有の学習則による教師あり学習

ANN の学習則として有効である誤差逆伝搬法などを、スパイクニューロンと時系列入力に適した形に変更することにより、SNN で教師あり学習を可能とする

手法

SNN 特有な学習規則による教師なし学習

ヘッブの学習則 (Hebbian learning) やスパイクタイミング依存性可塑性 (spike-timing-dependent plasticity; STDP) のような, SNN に特有の学習規則により教師なし学習を行う手法

ANN によるアプリケーションが豊富である昨今, SNN にそのような機能を持たせるための最も直接的な手法は, ANN による学習の結果を SNN にそのまま適用させることである. そのような発想によるものが, 1 つ目の訓練手法である ANN の重みの変換による学習である. 初期の手法 [79, 80] では, 入力スパイク列に対して手動で設定された畳み込みカーネルを適用し, CNN を学習させた後にその重みを SNN に渡すことで SNN に機能を持たせた. 人工ニューロンの入出力は, スパイクニューロンにおけるスパイク列の発火率を使用していることは既に述べた. この訓練手法では, この発火率モデルにしたがって, 学習した重みはスパイクニューロンの不応期などのパラメータに対応したスケールリングを施すことによって SNN のシナプス強度へと変換される.

この手法の利点として挙げられるのは, ANN 分野において開発されている多種のアプリケーションを, 直接 SNN のアプリケーションとして使用可能である点である. つまり, 最先端の物体検出, 識別や分類といったタスクについて, 直接 SNN で実行可能となる.

一方で, 短点となるのは, 変換可能な ANN のネットワーク構造に制約が追加される点である. 問題のひとつは, 人工ニューロンの出力は正負どちらの値も取りうるが, スパイクニューロンの発火率は常に正の値を取ることである. これは, ANN における活性化関数として前述した ReLU を使用することで, 常に人工ニューロンの値は正となることで解決可能である [81]. もうひとつは, ANN のプーリング層において, 入力最大の値のみを選択する最大プーリングが主として使用されることに対して, SNN において発火率の最大の値のみを得る機能を持たせることが難しいことである. これは, 最大プーリングに対して入力の平均値を出力する, 平均プーリングを使用することで解決可能である [82].

このような学習済みの ANN の重みを SNN のシナプス強度として変換する手法より直接的な手法として, 上記のような制約やスパイクニューロンの特性を人工ニューロンの枠組みに適用させた, スパイクニューロンと人工ニューロンの中間の動作をするニューロンを用いて学習を行い, それに伴って学習済みの重みを変換することなく直接的に SNN に適用させるもの [83, 84] が存在する. 主な違いとしては, 重みパラメータを変換する手法では, 元となる ANN の学習を行い, その重みをスパイクニューロンのパラメータに応じて変換することにより, SNN に転送した. ANN と SNN の中間のニューロンを使用する手法では, スパイクニューロンのパラメータにしたがって調整された活性化関数を用いる. その活性化関数を使用した人工ニューロンによって, ANN と同様

に学習を行うことで、重みを変換することなく直接 SNN に転送することが可能となる。この手法では、学習の後に変換を行うことなく、スパイクニューロンに適した状態で学習を終えるために、実行時の精度損失が少ないという利点がある。

1つ目の手法では、ANN を変換することによってそのアプリケーションリソースを活用する形で SNN を実行することが可能である。一方で、変換による精度の損失や、ネットワーク構造に制限があるという短点が存在した。そこで、ANN を使用することなく SNN 単体で教師ありの学習を行うことにより、SNN でのアプリケーション構築を試みるのが2つ目の手法である。この手法では、後述するヘップの学習則や STDP のような SNN 特有の学習則によるアプリケーション構築法が黎明期であるため、代わりに、ANN において成功している学習則である誤差逆伝搬法などを、SNN に適用することで学習を試みる。

層を増やす毎に、そのネットワーク構造が複雑になるため、単一の層のみを使用して学習を行う手法 [85, 86] が存在する。これら精度よく学習を実施可能であるが、より深い層構造を持つネットワークに適用することができない。複数の層をもつ SNN を学習するために、出力層から得られるスパイク列に着目して、その到達時間の逆伝搬規則を用いることで学習する手法 [87] が提案された。ここでは、3層の SNN において学習を成功させているが、より深い層を持つネットワークにおいては、計算コストの問題から適用がなされていない。近年では、膜電位の値からスパイクを確認するもの [88] や、長い時間窓によってニューロンの発火率を確認して逆伝搬則を適用し、それを個々のスパイクに対して更新することによるもの [89] などが提案されてきている。

これらの手法は、SNN 単体で動作するために、時系列データを扱う SNN 特有のアプリケーションの構築や、1つ目の手法より精度の高い学習を行うことが可能になると期待されている。一方で、学習における計算コストが大きいことや、より深いネットワークでの動作、また精度面においても、具体的なアプリケーションを SNN に実装するという点で、現状では1つ目の手法より優れているとは言えない。

SNN において、ヘップの学習則 [90] や STDP [91] のような特有の学習則を使用することは、生物学的な妥当性や本来の脳機能模倣という点のみならず、ニューロンに備わる機能であるためハードウェア実装した際の計算効率がよいという点でも利点が存在する。また、その学習においては、スパイクの時空間パターンを特徴として捉えていると目されるため、時系列データを使用した新たなアプリケーションの構築が可能であると期待される。このような3つ目の訓練手法においては、殆どが研究の初期段階にある。理由としては、出力層における誤差となるスパイク情報の、上位ニューロンからの伝搬メカニズムが不明であるという事が挙げられる。したがって、現在では下位の層において再帰的に学習

を実施するという手法が採られている [92, 93]. この訓練手法は, 現在 SNN の分野において盛んな研究対象であるものの, 具体的なアプリケーション構築にあたっては未だ初期段階であり, その学習精度も 2 つ目の手法で学習されたものより劣るのが現状である.

本研究では, 具体的なアプリケーションを実施することにより, SNN の適用可能性を検証することを目的とするため, 既存の ANN による地形相対航法アプリケーションの活用が可能である, 1 つ目の手法に焦点を当てる. したがって, 制約付きの ANN を学習し, そのシナプス強度を SNN に直接転送することによって, SNN におけるアプリケーションの構築を試みる.

4.2.4 SNN の実施

ソフトウェアシミュレータ

前項までに, スパイクニューラルネットワークに関わる数学的モデルを述べた. 実際到大規模な SNN のシミュレーションを実施するにあたっては, スパイクニューロンの動作レートが非常に高く^{*7}, またネットワーク内の神経接続の量が膨大であるため, LIF ニューロンのように簡易なモデルを用いたとしても, 解析的に行うことは難しい. したがって, ソフトウェアやハードウェアを用いたシミュレーションが, SNN 実行の解決策として挙げられる.

ソフトウェアシミュレーションにおける既存のアプローチとしては, ニューロンの状態が一定の時間分解能で更新される Clock-driven なものと, スパイクが到達した時のみ膜電位が更新される Event-driven なものが存在する. Clock-driven なモデルでは, 時間に対する膜電位の関数を記述する, 常微分方程式を解くための数値積分を使用する. しかしながら, 時間クロック (通常は 1ms 分解能) での更新のみでは, 非線形微分方程式を近似することができず, 離散時間ステップに拘束されるために正確さが失われる. これに対して, Event-driven なシミュレータでは, 常微分方程式の明示的な解法を使用し, スパイクの到着時間が丸められないために正確となる. しかしながら, LIF ニューロンを除けば, ほとんどのニューロンモデルは解析的に解くことはできない. また, イベントの頻度が高いため, このような非同期的アプローチによる計算効率の向上は望めない.

したがって, NEST[94] や Brian[95] などの, 多くの SNN の実行を行うソフトウェアシミュレータにおいては, ニューロンの状態が Clock-driven に更新されるが, シナプスは Event-driven に更新されるという複合的なアプローチをとっている.

ニューロモーフィックハードウェア

ニューロモーフィックコンピューティングにおける演算コアである、ニューロモーフィックプロセッサは、ニューロンやシナプス、スパイクの伝達モデルの実装方法に応じて、アナログ、デジタル、またはアナログ/デジタルの混合回路に分類できる [70].

一部のアナログ回路実装においては、ニューロンとシナプスによる処理を、直接ハードウェアでエミュレートする。この方法はエネルギー効率がよく、またデジタル回路よりも少ない面積で実装可能である。一方で、トランジスタの不整合によって、アナログ回路の動作を制御することは困難である。

High Input Count Analog Neural Network (HI-CANN)[96], Neurogrid[67] などの、アナログ/デジタル混合回路によって実装されたニューロモーフィックプロセッサの多くは、アナログ回路を使用してニューロンをエミュレートし、デジタルパケット通信を用いることによってニューロン間におけるスパイクの伝搬を表現する。これにより、ニューロン間の接続が再構成可能となり、低消費電力性を保ちつつリアルタイム処理の実施が可能となる。

TrueNorth[66] などの、デジタル回路によって実装されたニューロモーフィックプロセッサは、有限精度のデジタル回路を使用することにより、Event-driven な方式でニューロンをシミュレートし、電力消費を最小限に抑えている。このようなプロセッサでは、モデルの柔軟性が限られているため、ニューロンとシナプスは規模に制限があり、またネットワーク構成は変更できず直接ハードウェアとして製造される。

SpiNNaker[97] は、低消費電力の計算コアと、FPGA のようなスケラブルな通信ハードウェアを使用して、ニューロンとシナプスのモデルをソフトウェア上で実装している。ソフトウェアベースのプロセッサは、ニューロンモデルの選択や、ネットワークの構成においては優れた柔軟性を持っている。

一方で、これらデジタル回路による実装は、アナログ回路による実装に比べて、離散化の影響によって精度が低下し、またエネルギー消費が高くなる傾向にある。

イベントカメラ

上記まで、生物の神経回路を模倣することによって、その計算処理機能を再現する試みである、ニューロモーフィックプロセッサについて述べた。一方で、同様に視神経回路を模倣することによって、シリコンの網膜 (silicon retina) を作成する研究が存在する。そのような、イベントカメラ (event camera) と呼ばれるセンサは、生理学的な視覚経路のうち、動的な視覚情報の処理 (動体視力的) を行う機能の再現を行っている。

イベントカメラにおいては、従来のフレーム撮影を行うカメラと異なり、視神経におけ

るスパイクニューロンの動作を電子回路として実装している。したがって、従来の輝度画像のようにセンサ上に結像する風景の輝度値を同期的に取得するようなものではなく、イベントカメラにおいては、風景内の輝度変化に基づき、各画素で非同期的なバイナリイベントとして情報を取得する。

そのような非同期的で疎かつバイナリの情報取得は、同様に疎なスパイクによって非同期的に動作する SNN の入力として適している。また、従来のカメラにはないイベントカメラ固有の特徴が存在する。

イベントカメラでは、従来のカメラのように一定の時間間隔でセンサ上に蓄積された輝度値から画像を取得するのではなく、画素毎の輝度の変化に応じたイベント（スパイク）の列を非同期的に出力する。イベントカメラの動作における模式図を図 4.6 に示す。

各画素は、それらに入射する対数輝度 $L = \log(I)$ （ただし I は輝度値）に対して反応する [98]。特にノイズが無い場合における、ある時刻 t_k でのイベントの発生は、前回イベントが発生した時刻からの対数輝度の差分

$$\Delta L(t_k) = L(t_k) - L(t_k - \Delta t_k) \quad (4.13)$$

が、任意の閾値 $\pm C_{evt}$ ($C_{evt} > 0$) を超えた場合に発生する。

$$\Delta L(t_k) = p_k C_{evt} \quad (4.14)$$

ここで、 Δt_k は同じ画素において前回イベントが発生した時刻からの差分、 $p_k \in \{+1, -1\}$ は輝度変化の方向（明暗）を表す極性である。

図 4.6 の左上に示しているのが対数輝度変化、左下に示しているのが、その変化に対応するイベント発生時の模式図である。このように、センサに入力された輝度の対数値の変化が、任意の閾値を正負どちらかに超えた場合、マイクロ秒オーダのタイムスタンプと、イベントが発生した画素の座標、明暗のどちらの変化があったかの極性（ON, OFF）の三種類の情報を出力する。図 4.6 の右に示しているのが、実際に取得されたイベントのデータであり、緑色と赤色の点がそれぞれの極性とイベントの位置を示している。

輝度の変化は対数スケールで取得されるため、従来の可視光カメラでは対応の難しい高コントラストの変化であっても、イベントデータの取得が可能であり、そのダイナミックレンジは従来カメラの 50dB を上回る 130dB となる。また、輝度の変化が生じた時のみ、イベント駆動的にデータを取得するため、低レイテンシでの出力が可能である。この出力されるイベントはマイクロ秒オーダであり、数十 fps の画像出力となる従来カメラよりも高頻度の情報取得が可能である。これらのことから、フレーム間での情報の欠落が発生す

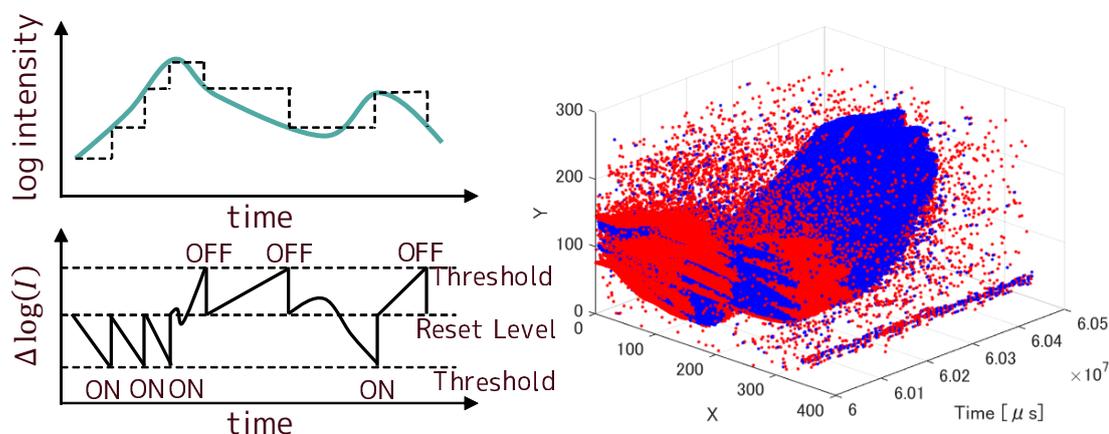


図 4.6 イベントカメラの動作原理

ることなく短い時間感覚でデータを取得することができ、より被写体の動きを捉えやすくなっている。

図 4.6 の右は、ヒトの手の移動を撮影することで取得された、イベントデータを時間軸で並べたものである。図中で赤色の点が ON イベント、青色の点が OFF イベントである。この時間軸の 1 目盛は 100ms であり、その間にも多くのイベントが短い間隔で取得されていることがわかる。このように、イベントカメラは高時間分解能やハイダイナミックレンジ、低ビットレートといった、従来のカメラの不得意とする条件に対して優れた特性をもつ。またその動作原理が視神経におけるニューロンのダイナミクスに着想を得ており、非同期かつ疎なバイナリのイベントを出力することから、SNN の入力として通常の輝度画像よりも適している。

4.3 提案する ANN から SNN への変換手法

スパイクニューロンは疎なスパイク電流によってイベント駆動で動作することに対して、人工ニューロンはスパイクニューロンの動作におけるタイミング情報を抜き、発火頻度のみを用いた数値計算である。このタイミングベースのスパイクニューロンと、レートベースの人工ニューロンとの間のギャップを埋めることが、ANN の機能を SNN に持たせるために必要である。

本節では、現在の ANN で一般的である、ReLU 活性化関数を用いた CNN を SNN に変換する方法について述べる。

4.3.1 ANN の定式化

ANN の機能を定式化する。まず、最も基本的なふるまいを持つ ANN である、全結合ネットワークについて説明する。全結合ネットワークは、全ての層が全結合層で構成されており、前の層の全てのニューロンが層内接続することなく次の層に完全に接続されている。また人工ニューロンは、接続された前の層のニューロンからの入力とその結合の重みとで線形和を取ったものを、非線形の関数である活性化関数に適用されることで出力を得るモデルであった。したがって、全結合ネットワークにおけるニューロンのふるまいは次式のように記述される。

$$x_i = f \left(\sum_j w_{ij} x_j + b_i \right) \quad (4.15)$$

ここで、 x_i は層内における i 番目のニューロンの出力、 w_{ij} は一つ前の層における j 番目のニューロンと今の層における i 番目のニューロンとの間の接続で発生する重み、 x_j は一つ前の層における j 番目のニューロンの出力であり、 b_i は i 番目のニューロンにおけるバイアス項である。また、 $f(\cdot)$ は活性化関数を表している。

活性化関数の一つの種類として ReLU が存在し、これは入力が負のときはゼロ値、それ以外の場合では入力を返す非線形の関数である。したがって、活性化関数として ReLU を用いると、式 (4.15) は次のようになる。

$$\begin{aligned} x_i &= \text{ReLU} \left(\sum_j w_{ij} x_j + b_i \right) \\ &= \max \left(0, \sum_j w_{ij} x_j + b_i \right) \end{aligned} \quad (4.16)$$

全結合ネットワークでは、現在の層の全ての出力を連続的に更新し次の層に伝播することによって、最終的な出力ニューロンを活性化する。

CNN は特徴検出器が畳み込みフィルターとなる、フィードフォワードなネットワークアーキテクチャである。このネットワークモデルは交互の畳み込み層とプーリング層から構成され、最後に全結合層を通して出力ニューロンを活性化する。畳み込み層では、前の層を任意の大きさのカーネルで畳み込むことによって、特徴マップを生成する。この畳み

込み層の動作は次式となる。

$$x_i^k = \max(W^k \otimes x^l + b^k), \quad (4.17)$$

ここで、 x_i^k は k 番目の畳み込みカーネルによって出力される i 番目の特徴マップ、 W^k は畳み込みカーネル、 \otimes は畳み込み演算を表す演算子であり、 x^l は前の層におけるニューロンの出力の l 番目のパッチを表し、また b^k は k 番目の畳み込みカーネルにおけるバイアス項である。

畳み込み層に続くプーリング層では、前の層から入力される特徴マップをサブサンプリングする機能を持つ。通常、この層の計算には次式のような最大プーリングが用いられる。

$$x_i = \max(x^l). \quad (4.18)$$

しばしば、タスクによっては平均プーリングが用いられるが、最大プーリングと比較して精度が落ちる [99]。平均プーリングは次のように表される。

$$x_i = \overline{x^l}. \quad (4.19)$$

CNN では、上記の畳み込み層とプーリング層を交互に実行することでデータの次元を減らしており、これは全結合ネットワークと比較して重みパラメータの量が少なくなるため、過学習が発生しづらくなる。

4.3.2 SNN の定式化

タイミングベースのスパイクングニューロンと、レートベースの人工ニューロンとの間のギャップを埋めるためには、SNN における発火頻度を、ANN の活性化関数と可能な限り一致させなければならない。

式 (4.4) を時間について解くことによって、スパイクングニューロンにおける発火頻度 (応答関数) λ を導出することができる。LIF ニューロンのモデルにおいて入力電流 I が一定だとすると、この発火頻度は次式となる。

$$\lambda(J) = \begin{cases} \left[\tau_r - \tau_m \log \left(1 - \frac{V_{th} - V_{rest}}{IR_m} \right) \right]^{-1} & (IR_m > V_{th} - V_{rest}) \\ 0 & (\text{otherwise}) \end{cases}. \quad (4.20)$$

ここで τ_r はニューロンの絶対不応期間である。

式 (4.20) に対して定電流 I を入力した時の応答を図 4.7 に示す。このように、入力となるスパイク列が定電流であるなら、図 4.7 のような応答を活性化関数の代わりに用いて ANN を訓練することで、SNN のネットワークにおける発火率について十分な精度で近似する事が可能である。

しかしながら実際には、入力電流はそのスパイク列のランダムさによりノイズが発生するため、入力電流は定電流ではない。ノイズの発生する電流 I_n は、加法的な白色雑音によって表現され、次式で表される。

$$I_n(t) = I_{mean} + I_{std}\xi(t) \quad (4.21)$$

ここで、 I_{mean}, I_{std} は入力電流の平均と標準偏差、 ξ は平均ゼロ、分散 1 の白色雑音である。式 (4.20) に対してこの電流 I_n を入力した時の応答を図 4.8 に示す。このように、ノイズの存在により関数形が変わるため、一意に活性化関数を決定することが難しい。加えて、式 (4.20) は原点付近で微分不可能であるため、誤差逆伝搬法のような ANN 特有の学習則を用いることに難がある。

この問題に対して、ノイズを含む LIF ニューロンの発火率の近似を行って、Contrastive Divergence アルゴリズムによる Deep belief network の学習を行う研究がある [83, 100]。また、[84] では LIF ニューロンの発火率に微分可能な形に修正し、ノイズを擬似的に与えることで、バックプロパゲーションによる CNN の学習を行っている。

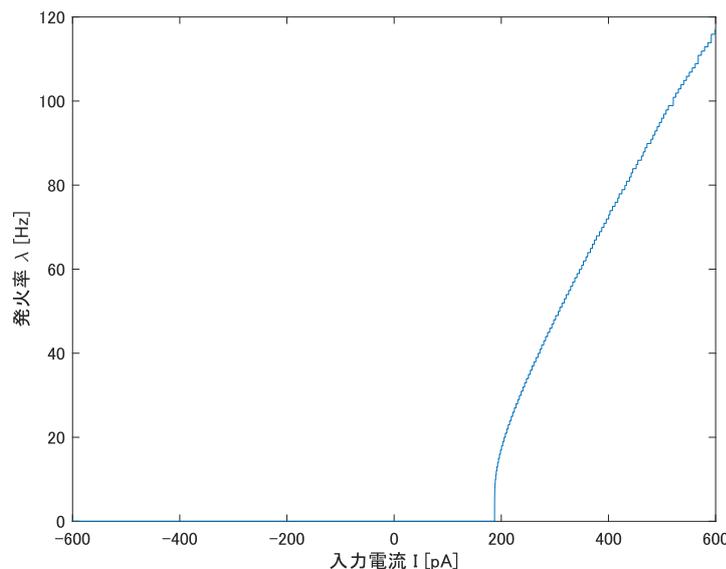


図 4.7 入力が定電流の時の LIF ニューロンの応答

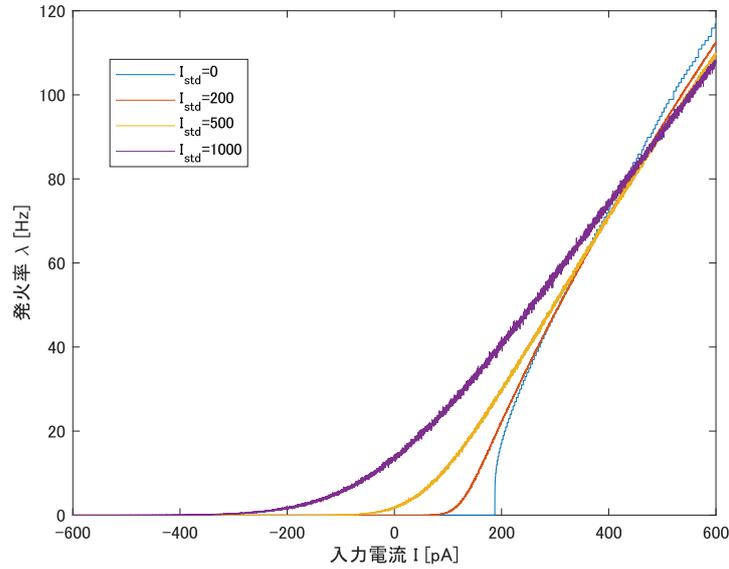


図 4.8 入力にノイズのある電流の時の LIF ニューロンの応答

この修正された LIF ニューロンは次のように記述される。

$$\lambda(I) = \left[\tau_r + \tau_m \log \left(1 + \frac{V_{th} - V_{rest}}{\gamma \log \left(1 + e^{\frac{IR_m - V_{th} + V_{rest}}{\gamma}} \right)} \right) \right]^{-1} + \eta(I), \quad (4.22)$$

ここで、

$$\eta(I) = \begin{cases} \text{Gauss}(0, \sigma) & (I > V_{th} - V_{rest}) \\ 0 & (\text{otherwise}) \end{cases}, \quad (4.23)$$

ここで γ は原点付近のスムージングの程度を決定する変更可能なパラメータであり、ゼロに近づくほど LIF ニューロンの応答関数に近くなる。

この近似された LIF ニューロンの応答関数を ReLU の代わりに用いて ANN を学習することで、ネットワークのパラメータを SNN に近づけることが可能となる。

4.3.3 変換手法

画像におけるパターン認識などのニューラルネットワークの能力は、幾つかの処理段階と学習メカニズムの結果である。ANN と比較して、SNN の学習手法は未だ黎明期である。そのような SNN を訓練するのに最も簡単なアプローチは、予め訓練した ANN のパラメータを同等の構造の SNN に変換することである。この手法を offline-learning[81, 82] と呼び、スパイクニューロンの入力電流と出力発火頻度との間の関係を表す応答関数

が、整流線形ユニット (ReLU) の活性化と関係していることから、重みパラメータの変換と正規化を行うことで SNN にマッピングを行う。この手法は、ANN の変換にあたってエラーレートが良いパフォーマンスを発揮するが、活性化関数やプーリングレイヤーにおいて制約が加わる。

次のアプローチは、活性化関数をスパイクニューロンに適した形式に修正した ANN を訓練することで、直接重みを SNN に渡すことである [83, 100, 84]。この手法は、SNN の推論時に多少精度は落ちるものの、入力データや重みパラメータを事後処理しなくとも良いという点で一般性がある。これらの手法は MNIST で良い結果を達成しているが、実際の画像認識タスクについては論じられていない。

表 4.2 に ANN を SNN に変換するための最近のモデルを示す。この表は変換モデルにおける適用可能なネットワークのタイプ、スパイクニューロンのタイプとその学習方法を示している。

上述のように、有効な学習則の欠如のため、SNN で ANN のアプリケーションを実行するには ANN の学習結果を SNN に変換する必要がある。本研究では、地形相対航法タスクのニューロモーフィックプロセッサへの適用性を評価するため、CNN の活性化関数として LIF ニューロンを用いたネットワークを用いる。

ここで提案する具体的な変換手順は次のとおりである。

1. 変換元となる ANN を通常どおり学習させる。
2. ニューロンのモデルと構成を SNN に近づけて修正した ANN を、step 1 の学習結果と合わせて転移学習する。
3. Step 2 で学習した重みを、元となる ANN と同等の層構成をもつ SNN に転送する。

ステップ 1 では変換したいオリジナルの ANN を通常通り訓練する。この研究では、後述する、クレータ識別を行う 1c-1p-1f の Deep ANN[101] について変換を行う。ステップ 2 では、ニューロンの活性化関数を式 (4.22) に置き換え、またプーリング層に修正を施したものを再訓練する。このとき、Step1 の重みを用いて転移学習を行うことで、学習の収束性と精度向上を図る。最後に Step3 では、通常の LIF ニューロンを用いた、Step1 と同構造の SNN に対して Step 2 で学習された重みを直接転送することで、推論を行う。

表 4.2 ANN を SNN へと変換するモデル

Model	Network type	Neuron type	Training method
Cao[81]	Spiking CNN	Integrate-and-fire	Offline learning
Diehl[82]	Spiking CNN	Integrate-and-fire	Offline learning and weight normalization
O’connor[83]	Spiking deep belief network	Leaky integrate-and-fire	Contrastive divergence
Stromatias[100]	Spiking deep belief network	Leaky integrate-and-fire	Contrastive divergence, hardware
Hunsberger[84]	Spiking CNN	Leaky integrate-and-fire	Direct conversion
This paper	Spiking CNN	Leaky integrate-and-fire	Direct conversion with transfer learning

4.4 シミュレーションによる検証実験

4.4.1 月面画像による検証

検証方法

提案した手法の評価のため、画像航法タスクとしてクレータ識別を行う ANN の SNN への変換を行った。ここで使用した ANN の構造について図 4.9 (a) に示す。このネットワークは 1 チャンネルの 28x28 の画像を入力とし、続くレイヤで 5 x 5 のカーネルを 20 個用いて畳み込みを行う。その後、2x2 でストライドが 2 の最大プーリングによりサブサンプリングする。最後にその特徴マップを全結合してクレータか否かの識別結果を出力する。

変換手順の Step 1 ではこのオリジナルのネットワークについて学習する。図 4.9 (b) に、Step 2 で用いる修正されたネットワークのアーキテクチャを示す。この修正されたネットワークでは、オリジナルのネットワークにおける畳み込み層と全結合層において、活性化関数として用いられた ReLU 関数を、式 (4.22) で示した修正 LIF ニューロンの応答関数に置き換える。加えて、最大プーリングを平均プーリングに変更することで SNN の複雑性が増加しないようにしている。

この変更によって、SNN の入力となるスパイク列の発火率を正しくニューロンにマッピングすることができる。これらのネットワークはどちらも誤差逆伝搬法と Conjugate gradient method を用いて学習した。

検証を行うにあたって、LIF ニューロンのパラメータである τ_r , τ_m , γ と σ は学習の成功に大きな影響を与える。これらを設定するとき重要なことは、ANN で使用される ReLU と、SNN で使用される LIF との間の乖離を少なくすることである。すなわち、入力の活性化関数の導関数が ReLU 同様に 1 付近であることと、原点付近のスミージングが LIF に近くなることが重要となる。

表 4.3 に、シミュレーションで使用した LIF ニューロンのパラメータを示す。

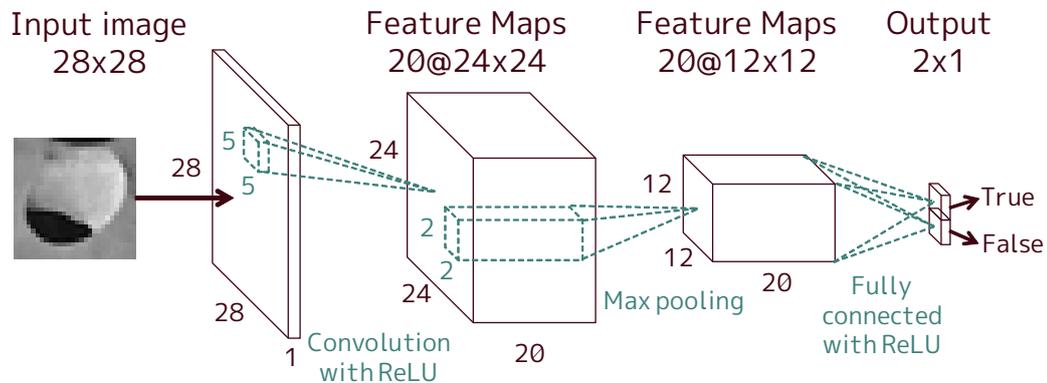
教師あり学習のためのデータセットもオリジナルのネットワークで使用されているものと同様のものを用いた。このデータセットは月面のクレータとクレータでない地形画像のコレクションである。これらの画像は SELENE の high definition camera system で撮影されたものから切り出したものであり、約 30000 枚の訓練画像が含まれている。

表 4.3 ニューロンのパラメータ

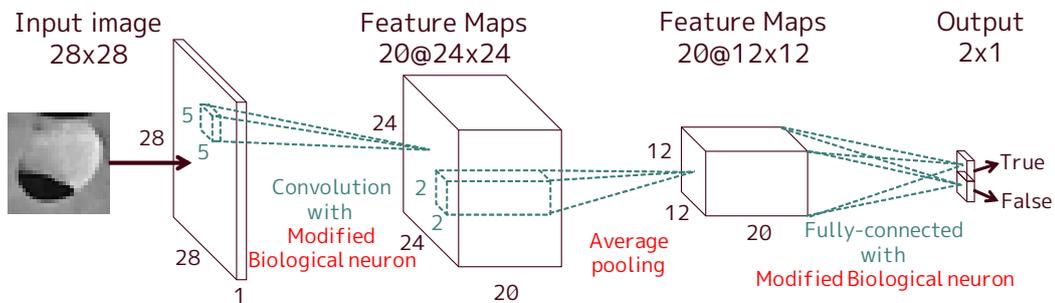
Parameters	Values
τ_r	1 ms
τ_m	20 ms
V_{th}	-50 mV
V_{rest}	-65 mV
γ	0.02
σ	15

検証結果

提案した変換手法を用いて、実際にクレータ識別の ANN を SNN へと変換し、PC 上のシミュレーションにて実行した。ANN の訓練については MATLAB を、SNN のシミュレーションには pyNN[102] と NEST シミュレータを用いた。シミュレーションにおいて



(a) オリジナルのクレータ識別のネットワーク



(b) 修正したクレータ識別のネットワーク

図 4.9 検証に使用した ANN

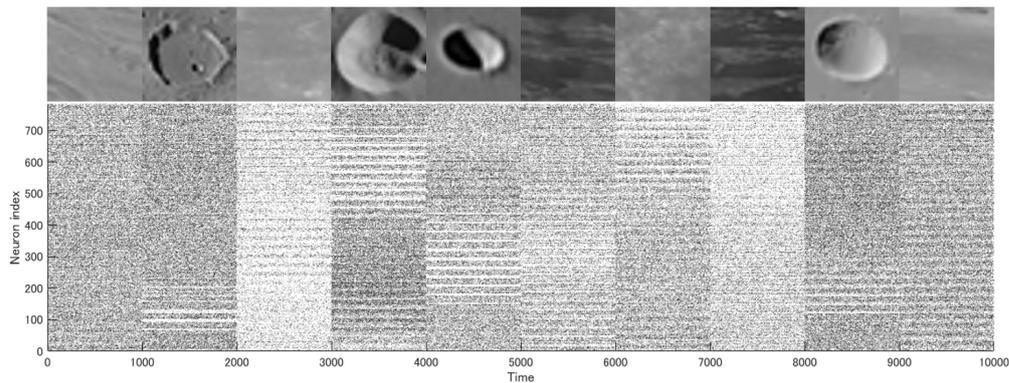


図 4.10 検証に用いた入力データ

表 4.4 提案手法の各ステップにおける分類精度

Step	Classification accuracy
Original network	99.4 %
Modified network	98.6 %
Spiking network	96.2 %

は、8000 ケースの入力画像について各変換ステップで推論を行った。また step 1, 2 における学習では、バッチサイズ 128, エポック数 20 で実施した。

図 4.10 にシミュレーションに用いた入力データを表す。ここで、上段に ANN の訓練に用いた画像が示されており、下段はその画像によって生成された SNN の入力となるポアソンスパイク列である。このスパイク列は、スパイクニューロンのダイナミクスによって続くレイヤに伝搬されて最後のレイヤで出力される。この入力によって出力された発火率とスパイク列を図 4.11 に示す。ここで、下段が入力画像を SNN で推論することで得られた出力スパイク列であり、中段はその出力スパイク列を発火率にしたものである。各ニューロンインデックスは図 4.9 で示したネットワークアーキテクチャにおける 2×1 の出力ニューロンに対応しており、発火率の高さは予測確率の高いことを表現している。入力画像について正しくクレータであるもの（左から 2,4,5,9 番目の画像）の結果を確認すると、クレータであるとする出力ニューロンの発火率が高いことから正しく識別が行えている。

各ステップにおける識別精度を表 4.4 に示す。識別精度はそれぞれ平均で、Step1 のオリジナルの ANN が 99.4%, Step2 の修正された ANN で 98.6%, 最後の SNN における推論が 96.2% となった。この結果は提案手法が、精度を大きく損なうことなくオリジナルの ANN を SNN に変換可能であることを示している。

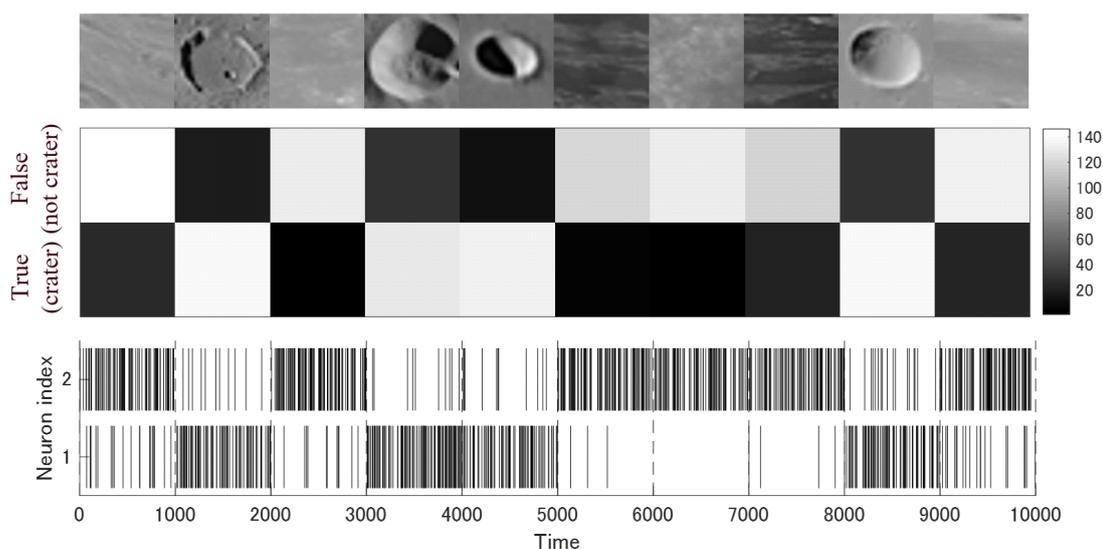
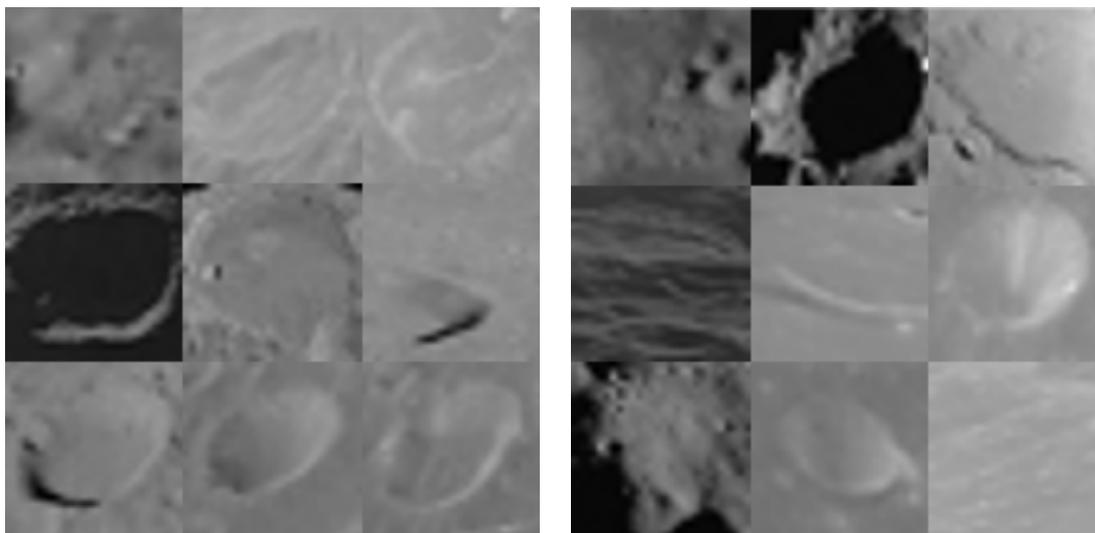


図 4.11 検証の出力データ

一方で、この識別精度は各ステップ毎にそれぞれ 1.2%, 2.4% づつ損失が発生している。この精度低下のひとつの原因は、修正された活性化関数と実際のスパイクニューロンの応答関数との間のミスマッチにあり、これは SNN で使用される LIF ニューロンの応答関数の形を、Step2 における ANN の活性化関数が完全に模擬できないことに起因する、変換手法共通の損失である。

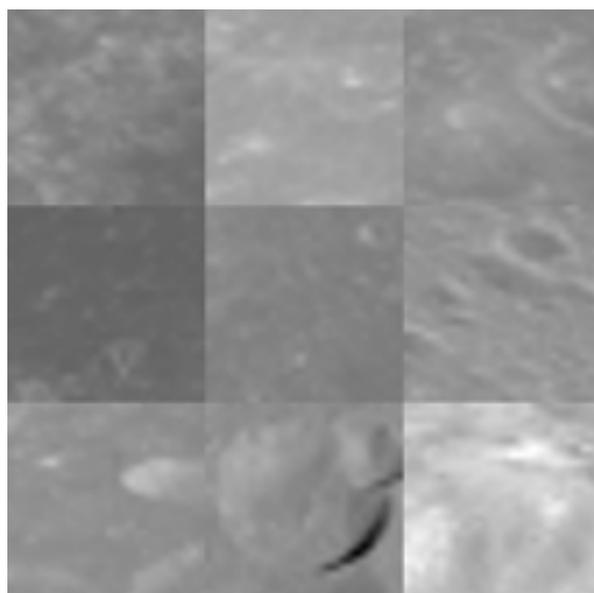
他の精度低下要因の発見のため、各ステップにおいてクレータの分類に失敗した入力画像を確認した。この各シミュレーションステップにおいて、分類を間違えたケースの例を 4.12 に示す。図 4.12(a) ではオリジナルのネットワークにて分類不可能だったケースが、図 4.12(b) ではオリジナルのネットワークにて分類可能かつ修正されたネットワークで不可能だったケースが、そして図 4.12(c) ではオリジナルのネットワークと修正されたネットワークで分類可能かつ SNN で識別不可能だったケースがそれぞれ 9 枚示されている。ここで、ステップを遡って間違えるケース、つまりオリジナルのネットワークでは間違えたが、修正したネットワークでは正解したケースは存在しなかった。加えて、上流のステップで間違えたケースは全て下流のケースにおいて間違えていた。

これらの図を見ると、(a) と (b) は殆どがクレータの輪郭が消えかけている画像や、影によって情報の少ない画像によって構成されている。この結果は、ANN における学習の実施において、そのような入力に対する特徴表現の獲得を仕損じていることを示唆する。したがって、より多くの特徴表現を得るためには訓練データの量を増やすか、ネットワークをより深くすることによって精度を向上させることができる可能性が存在する。また (c) のほとんどのケースは、高輝度かつ低いコントラストの画像で構成されている。この結果は、SNN において高頻度かつ一様なスパイク列の入力が、分類精度に影響を与えて



(a) Step1 で分類に失敗した入力画像の例

(b) Step1 で分類に成功したが、Step2 で失敗した入力画像の例



(c) Step1,2 で分類に成功したが、Step3 で失敗した入力画像の例

図 4.12 各ステップにて分類に失敗した入力画像

いることが示唆される。これは、ネットワーク内の各スパイクニューロンにおいて、連続するスパイクの到達は発火の飽和を招き、したがって後続のニューロンへと情報の伝達が行われなかったことに起因すると考えられる。この解決のためには、より SNN の動作に適した、より疎な形での入力を与えることが必要となる。

表 4.5 クレータ識別のネットワークが使用する消費電力の概算

Neuromorphic processors	α , pJ	E , μW
Merolla et al. (TrueNorth) [66]	45	10,350
Cruz-Albrecht et al.[103]	0.37	85.1

消費電力の解析

この研究の目的の一つは、SNN によるクレータ識別タスクがニューロモーフィックプロセッサに実装された時の低消費電力性の確認である。開発されている幾つかのニューロモーフィックプロセッサについては、スパイクあたりのエネルギー消費が示されている [66, 103]。ここで示されている値に基づいて、これらのハードウェアに実装されたクレータ識別タスクの消費電力の初期解析を行う。

[81] はスパイクがシナプス前かシナプス後かに関係なく、スパイクは同じエネルギーを消費するという単純化をして電力消費を解析した。我々もその仮定に基づいてニューロモーフィックプロセッサの電力消費を解析する。

解析のためにネットワークのニューロンにおける全てのスパイクの数をシミュレーション時に記録した。従って、このネットワークで 1 枚の画像をニューロモーフィックプロセッサで実行するのに必要なエネルギーは次式で記述される。

$$\begin{aligned}
 E &= 230 \times 10^6 \text{ spike/image} \times 1 \text{ image/s} \times \alpha \text{ pJ/spike} \\
 &= 230 \times 10^6 \alpha \text{ pJ/s} \\
 &= 230\alpha \text{ } \mu\text{W},
 \end{aligned} \tag{4.24}$$

ここで E は総エネルギーを表し、 α は各ニューロモーフィックプロセッサにおける消費電力を表す。表 4.5 に消費電力が発表されているニューロモーフィックプロセッサについて、式 (4.24) を用いて本稿で述べたクレータ識別 SNN の消費電力の推定を示す。この結果は既存の計算機よりも低消費電力で SNN を動作させられることを示している。

4.4.2 イベントカメラを入力とした検証

検証方法

前項では、オリジナルの ANN と同様の入力として、月面の輝度画像を入力とした。一方で、単なる数値である輝度画像は、非同期の時系列イベントであるスパイク列によって駆動する SNN に対する入力としては、必ずしも適しているとは言えないことも判明した。

そこで本項では、視神経におけるニューロンのダイナミクスを模倣することにより、非

同期かつ疎な時系列イベントを取得することが可能な、イベントカメラを用いて地形情報を SNN に入力する。

イベントカメラによって取得される (画像様の) データと輝度画像の違いは次のとおりである。

輝度画像

画素全体において密な、8bit で 1ch の数値データ。ANN の学習にあたってはそのまま使用し、SNN に入力する場合、数値をニューロンの発火率と読み替えることにより、ポアソン過程からスパイク列を生成し、入力層へ伝達する。

イベントデータ

画素空間内と時間方向に対して、疎なバイナリのイベント列。ANN の学習にあたってはイベントを積算することによって発火率としての画像を作成する。SNN に入力する場合、イベントをスパイクとして直接入力する。

このイベントデータの入力により、輝度画像での入力と比較して幾つかの利点が期待される。まず、入力が時空間方向に疎となることによりネットワーク内のニューロンの発火が抑えられ、全体として消費電力が減少する点。また、SNN の入力時にデータを変換する必要がなく、ANN で学習した内容に対して損失が少なくなることによって、変換によって生じる精度の低下を抑えられる可能性がある点である。

イベントデータ入力に対する影響を確認するため、データセットの作成を行った。前項の内容との整合性を取るため、同等の輝度画像によって構成されるクレータデータセットについて、イベントカメラによる撮影を実施することでイベントデータを取得した。

データ取得にあたっては、ディスプレイ上に表示したクレータ/非クレータ画像を一方に一定速度で移動させ、その画面を 1 秒間イベントカメラにて撮影した。このイベントデータ取得を、訓練画像 30000 毎、テスト用画像 8000 毎に対して実施し、イベントカメラのデータセットとした。また、ここで取得したイベントは時間軸方向に拡がりを持つバイナリデータであるため、発火率を扱い数値計算によって処理を実施する、ANN の入力には適していない。したがって、イベントデータを時間軸方向に積算することにより、各データにおける発火率を算出した。

このように作成されたイベントカメラデータセットの内容を図 4.13 に示す。図 4.13(a) では、イベントデータ作成に使用された、クレータ/非クレータを表す輝度画像を示している。

この輝度画像を撮影することによって得たイベントが、図 4.13(b) となる。上段のクレータ画像によるイベントデータでは、クレータ内部形状に起因する、明暗のコントラストにより、中央付近にイベント列が多く生成されていることが見て取れる。下段の非

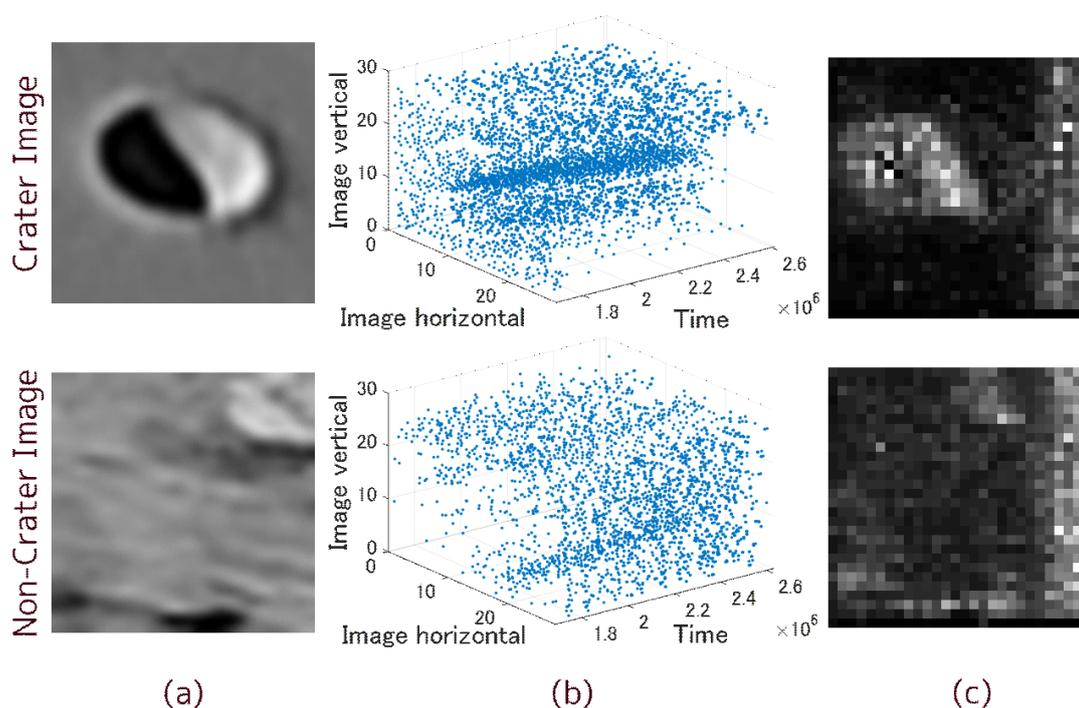


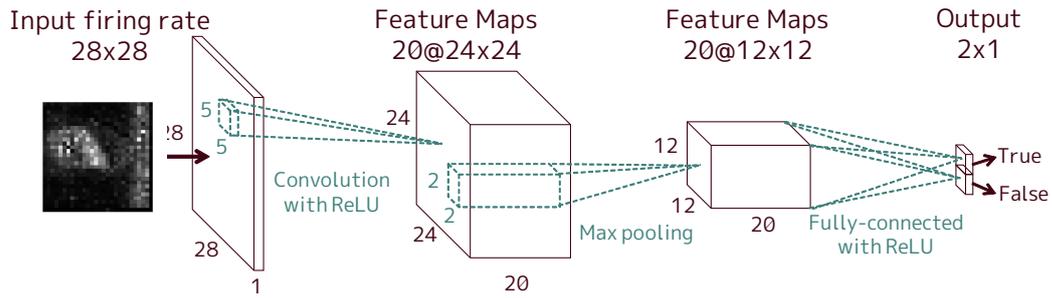
図 4.13 イベントカメラによるデータセット内容

クレータ画像によるイベントデータでは、コントラストの強い地形特徴が少ないために、データ全体においてイベントがまばらである。

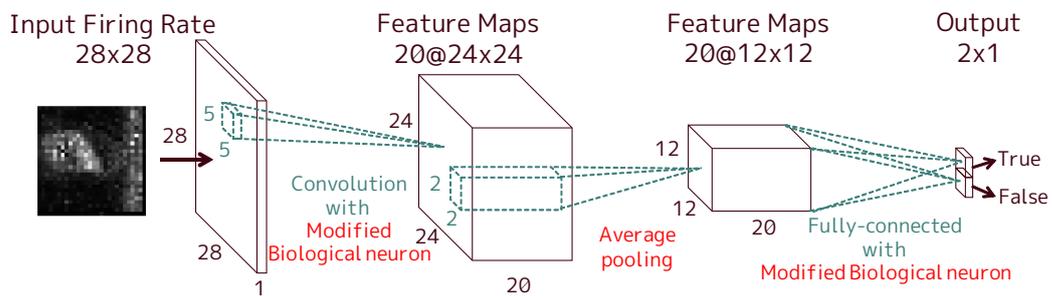
これらイベントデータの発火率を算出したものが、図 4.13(c) である。イベントが多い画素は発火率が高いため白色に、少ない画素は黒くなっている。上段では、前述のようにクレータによる輝度変化が高い空間位置において、発火率が高くなっていることがわかる。また、上下どちらの画像も右端の発火率が高くなっているのは、オリジナルの輝度画像を移動させることによってデータを取得することにより発生した、画像端の輝度変化によるイベントである。

イベントデータ入力による手法の評価のため、画像航法タスクとしてクレータ識別を行う ANN の SNN への変換を行った。ここで使用した各変換ステップにおけるネットワークの構造は、前節と同様であるが、学習を行うにあたって入力イベントデータの発火率となる。

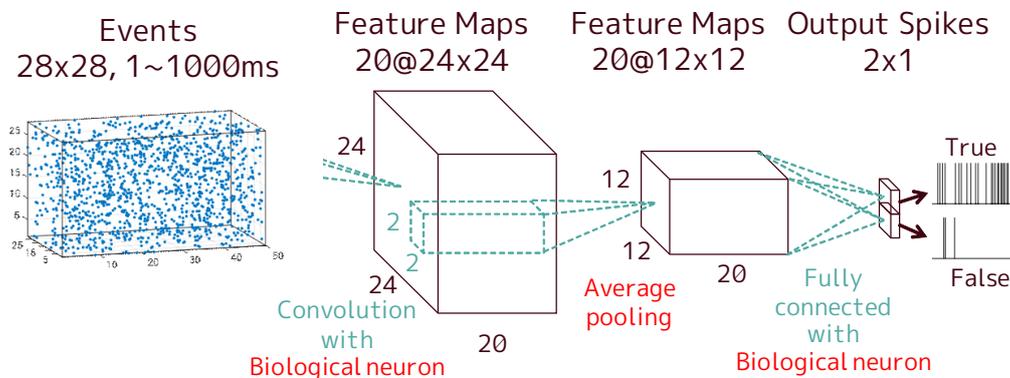
図 4.14 (a), (b) に、Step1, 2 で用いる修正されたネットワークのアーキテクチャを示す。Step1, 2 では先と同様に、各イベントデータの発火率に対して、オリジナルのネットワークと同様に学習が実施される。続いてその学習結果となるシナプスの重みを、活性化関数がスパイクニューロンの応答関数に変更された修正済みネットワークにおいて転



(a) オリジナルのクレタ識別のネットワーク



(b) 修正したクレタ識別のネットワーク



(c) SNN によるクレタ識別のネットワーク

図 4.14 検証に使用した ANN

移学習を用いて学習が実施される。

この変更によって、SNN の入力となるイベントデータの発火率を正しくニューロンにマッピングすることができる。これらのネットワークはどちらも誤差逆伝搬法と Conjugate gradient method を用いて学習した。

検証結果

提案した変換手法を用いて、実際にクレータ識別の ANN を SNN へと変換し、PC 上のシミュレーションにて実行した。ANN の訓練については MATLAB を、SNN のシミュレーションには pyNN[102] と NEST シミュレータを用いた。シミュレーションにおいては、8000 ケースの入力画像について各変換ステップで推論を行った。また step 1, 2 における学習では、バッチサイズ 128, エポック数 20 で実施した。

図 4.15 にシミュレーションに用いた入力データを表す。ここで、上段のクレータ/非クレータ画像は、オリジナルのデータセットに含まれる輝度画像を表しており、下段はその画像から作成したイベントデータについて、縦軸がその画素 (ニューロン)、横軸がイベントの発生した時刻を表すラスタプロットである。また中段は、イベントデータにおける発火率に関して、高頻度のものを白に、低頻度のものを黒になるよう画像にしたものである。

Step1, 2 における ANN の学習については、この中段の発火率画像を用いており、それら学習結果をもって推論を実施する SNN への入力、下段のイベントデータを用いている。この入力によって出力された発火率とスパイク列を図 4.16 に示す。ここで、下段がイベントを SNN で推論することで実際に得られた出力スパイク列であり、中段は出力スパイクの頻度から得られた、クレータ分類の予測確率である。中・下段の出力における、各ニューロンインデックスは図 4.14 で示したネットワークアーキテクチャにおける 2 x 1 の出力ニューロンに対応しており、中段の発火率の高さは予測確率の高いことを表現している。入力画像について正しくクレータであるもの (左から 2,4,5,9 番目の画像) の結果を確認すると、クレータであるとする出力ニューロンの発火率が高いことから正しく識別が行えている。

イベントデータを入力とした、提案手法における各ステップの識別精度を表 4.6 に示す。識別精度はそれぞれ平均で、Step1 のオリジナルの ANN が 94.8%, Step2 の修正された ANN で 91.5%, 最後の SNN における推論が 90.6% となった。

この結果を前項の結果と比較してみると、オリジナルの ANN の時点で 5% 弱の精度低下が存在している。この理由として挙げられる明確な差は、入力の違いにある。図 4.15 の上段に示したように、前項では月面のクレータ/非クレータ地形における輝度画像を入力として使用した。本実験では、学習用、テスト用のデータはそれと全く同じデータセットを元としてイベントデータを作成したため、地形の差によって精度低下が生じることは考えにくい。したがって、図 4.15 の中段に示したような、地形をイベントデータとして取得し、その発火率を入力として用いたことによって生じたと考える。イベントデータ

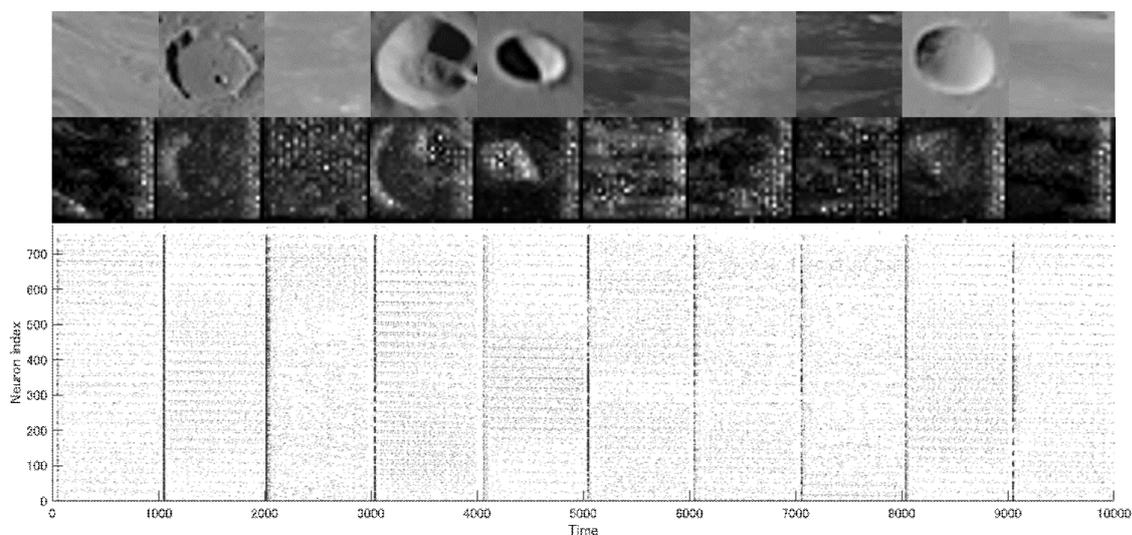


図 4.15 イベントカメラ入力の検証に用いた入力データ

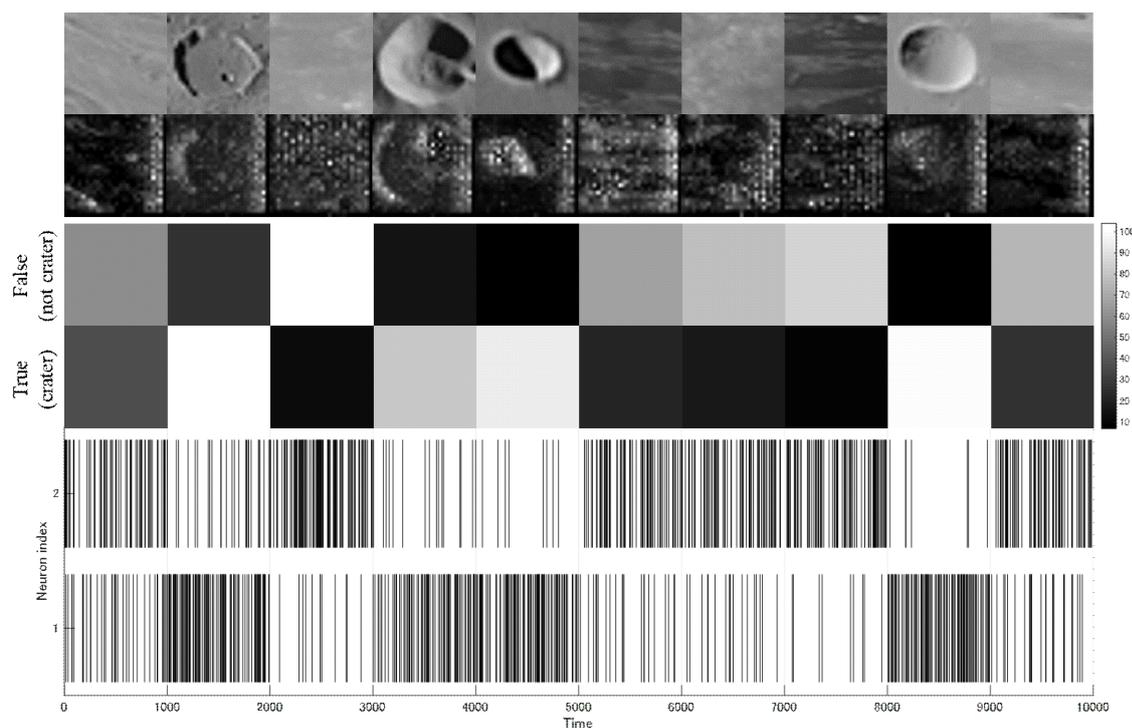


図 4.16 イベントカメラ入力の検証の出力データ

は、その取得元であるイベントカメラの原理により、コントラストの強い、エッジのような画素に強く反応することで生じる。よって輝度画像で表現されていた細かな地形情報は失われ、より明暗の差の大きい特徴のみが残る。このイベントデータ化の際に発生する情報欠落により、クレータ/非クレータの地形分類が困難となっている。

一方で、オリジナルのネットワークによる分類精度から、SNN での推論による精度の

表 4.6 イベントデータを入力としたの各ステップにおける分類精度

Step	Classification accuracy	
	In Sec. 4.4.1	Event data
Original network	99.4%	94.8%
Modified network	98.6%	91.5%
Spiking network	96.2%	90.6%

差によって見られる，変換手法に起因する精度低下は 4.2% 程度である．これは前項の 3.2% から大きく損なっているわけではない．この結果は，提案手法がイベントデータの入力に対しても精度を大きく損なうことなく，オリジナルの ANN を SNN に変換可能であることを示している．輝度画像よりも情報が疎かつノイズが乗る，イベントデータを入力としても変換に関わる精度低下が少ない理由としては，スパイクニューロンの不応期による連続したスパイクの影響の低減や，学習時にノイズを意図的に付与したことにより，その耐性の獲得に繋がっていることが考えられる．一方で，イベントカメラにおいて風景中の明暗の連続や瞬間の変化を捉えていた場合を考えると，連続したスパイクやイベントデータの取得原理によるノイズに，必ずしも情報が無いわけではない．したがって，変換時におけるノイズ表現の学習，あるいはノイズから情報を抽出する方向性の検討の余地が，対象となる風景とアプリケーション内容によっては存在する．

消費電力の解析

イベントカメラを SNN の入力ソースとして使用した理由は，そのデータが輝度画像から入力スパイク列を発生させることに比べて，疎であるために SNN の実行時消費電力が抑えられることを期待したものであった．

本実験においても前項と同様に，SNN 内で発火したスパイクを数え上げることによって，研究室レベルでは開発されている幾つかのニューロモーフィックプロセッサ [66, 103] について，これらのハードウェアに実装可能であることを仮定して，クレータ分類タスク実行時における消費電力の初期解析を行う．

[81] はスパイクがシナプス前かシナプス後かに関係なく，スパイクは同じエネルギーを消費するという単純化をして電力消費を解析した．我々もその仮定に基づいてニューロモーフィックプロセッサの電力消費を解析する．

この解析のためにネットワークのニューロンにおける全てのスパイクの数をシミュレーション時に記録した．入力としては，精度検証に用いたイベントデータのテストセット 8000 ケースを用い，ここで記録されたスパイクの発火数は平均して，約 21×10^6 回であった．したがって，このネットワークで 1 枚の画像をニューロモーフィックプロセッサ

表 4.7 イベントデータを入力としたクレータ分類のネットワークが使用する消費電力の概算

Neuromorphic processors	α , pJ	E , μW
Merolla et al. (TrueNorth) [66]	45	945
Cruz-Albrecht et al.[103]	0.37	7.77

で実行するのに必要なエネルギーは次式で記述される.

$$\begin{aligned}
 E &= 21 \times 10^6 \text{ spike/image} \times 1 \text{ image/s} \times \alpha \text{ pJ/spike} \\
 &= 21 \times 10^6 \alpha \text{ pJ/s} \\
 &= 21\alpha \text{ } \mu\text{W},
 \end{aligned} \tag{4.25}$$

ここで E は総エネルギーを表し, α は各ニューロモーフィックプロセッサにおける消費電力を表す.

表 4.7 に消費電力が発表されているニューロモーフィックプロセッサについて, 式 (4.25) を用いて本稿で述べたクレータ識別 SNN の消費電力の推定を示す.

この結果を, 入力を輝度画像とした場合の消費電力概算結果である表 4.5 と比較すると, IBM 社の TrueNorth プロセッサにてクレータ分類タスクを動作させたとき, 入力として画像を用いた場合は約 $10 \mu\text{W}$ の消費電力であった. 今回, 入力としてイベントデータを用いた場合は, $0.95 \mu\text{W}$ の消費電力である. この結果は, 入力ソースをイベントベースにしたとき同様のタスクが約 $1/10$ 倍の低消費電力での実行の可能性があることを示唆している.

4.5 まとめと考察

本研究では ANN で記述された高度な画像航法タスクを, 低消費電力で動作するニューロモーフィックプロセッサで実行するために, ANN を SNN に変換する手法を提案した. この手法ではオリジナルの ANN は SNN のニューロンの活動に適した活性化関数を用いて修正され, そのアーキテクチャは転移学習を用いて再学習される. その後, 学習された重みパラメータを直接転送された SNN は, オリジナルの ANN と同じタスクを行う事が可能である.

検証のために月面のクレータを識別する ANN に本手法を適用した. その結果は, 当手法がオリジナルの ANN で発揮される精度を大きく損なうことなく変換を行うことができることを示した. 加えて, ニューロモーフィックプロセッサを用いてクレータ識別を処理

するための消費電力の初期解析を行った。結果として、そのネットワークがニューロモーフィックプロセッサを用いて1画像あたりわずか10mW程度で処理できることを示した。

本研究の変換手法は、活性化関数がReLUであることと平均プーリングを用いることを除けば、オリジナルとなるANNの構成について、殆どのフィードフォワードなネットワークに用いる事が可能である。したがってここではクレータの識別タスクを例に検討を進めたが、実装可能なプロセッサの規模に制約がなければ、End-to-EndなTRNのANNアプリケーションも、従来のFPGAベースの計算機を用いた場合より2桁程度低い電力消費で動作する可能性があることを示唆している。よって当手法とその処理システムは、太陽から遠く離れた天体で行われるような将来の深宇宙探査において、従来型計算機では実施不可能な電力制約の下でも、高度な地形相対航法タスクが実行できる一つの方向性を示している。

より処理性能を向上させる発展の方向性として、本研究では図4.8に示すようなLIFニューロンの発火率に対する応答を模擬し、ANNにおける活性化関数として扱うために式(4.22)による近似を用いた。一方で、ニューロンの応答はそのパラメータによっても変動がある。ニューロンのパラメータは生理学的には妥当な値が存在するものの、電子回路で構成されるニューロモーフィックプロセッサのような計算機においては、その値は範囲があっても可変である。その範囲内でパラメータを変動させることにより、ニューロンの発火率応答をより単純にさせる方向性が考えられる。すなわち、本研究ではスパイクングニューロンに合わせる形でANNの活性化関数に変更を加えたが、ReLUなどの普遍的なANNの活性化関数形に近づけたスパイクングニューロンの応答により、ニューロモーフィックプロセッサにおけるSNNとANNをシームレスに接続させる方向性や発火率を抑える特性にすることによる、低消費電力処理などが研究の発展として考えられる。例として、本研究で使用した表4.3に示したパラメータから、 $\tau_r = 0, \tau_m = 50$ としたときのノイズを含む電流による発火率の応答を図4.17に示す。これは、本研究で使用したパラメータでの構成より、ノイズによる形の変動が少なくなっている。このように、消費電力を抑える方向や精度を上げる方向でのニューロンパラメータの探索や適用範囲の確認は将来研究とする。

Notes

- *1 実際、Intel社初期の80486プロセッサに対して現在のCore-i7プロセッサにおいては、IPCは約6倍向上していることに比べて、動作周波数は33倍の向上となっている。
- *2 図の作成にあたり、各CPUのスペックについては[66]をベースに、[104, 105, 106]で示された値を参照した。
- *3 図の作成にあたり、動作周波数と性能(10億命令毎秒, giga operations per second; GOPS)は、[105, 30]を参考にした。またここでは16bitの整数演算によるベンチマーク結果を載せている。
- *4 生理学的にはイオンの拡散によるもの。

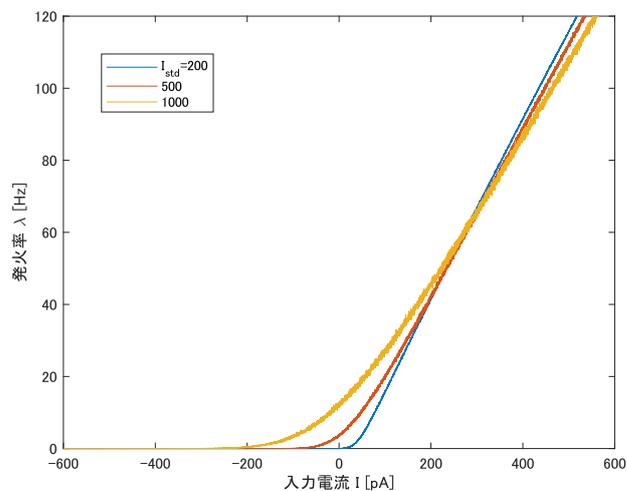


図 4.17 $\tau_r = 0, \tau_m = 50$ の時の発火率応答

- *5 シナプス強度にはシナプス効力やその他のシナプス結合に関する重みの要因が含まれているものとした.
- *6 実際にはスパイクニューロンも人工的なモデルであるが、本研究では区別と表記の簡便さのため、ニューロンの発火率モデルを人工ニューロンと呼ぶ.
- *7 1つのニューロンに対して 10^4 オーダの神経的イベントが1秒間に発生する

Chapter 5

結論

本論文では、画像を用いて自律的に実施される地形相対航法に関わる研究について述べた。全体としては、宇宙機における地形相対航法の実施を検討する際に存在する、実時間で計算処理要求と、低リソースな計算機能力との間の問題を解決すべく、二種の方向性から提案と評価を行った。一つは、現在の限られた計算処理リソース環境下で、地形相対航法に必要なタスクを自律的に実施するために重要となる、処理の削減を図った手法について提案し、その適用範囲と性能評価、具体的な実装を通して、現在実施可能な地形相対航法の一つの解と、処理性能の限界について論じた。もう一つは、現在の計算機環境の制約によるアプリケーション性能の限界を受け、より高度な地形相対航法タスクを実施するために、超低消費電力での計算処理が可能であると目される、ニューロモーフィックコンピューティングについて着目し、汎用的な地形相対航法アプリケーションの構築法とその適用について生じるメリットについて論じた。以下に、それぞれについての主な成果を述べる。

第3章「現状の低計算リソース環境における地形相対航法」では、月着陸機をターゲットとして、計算機環境が低リソースである状況においてもリアルタイムかつ自律的に位置・速度情報を推定することが可能な地形相対航法手法をそれぞれ提案した。

位置推定手法では、抽出された特徴点をマッチングする問題に対して組み合わせ問題であり計算複雑度の高い点パターンマッチングの手法を改善した。具体的には、地図とモデル点とを重ねあわせる際の最小要素としての線分に着目し、線分による大域的なマッチングの後に、局所的に点パターンの対応を探索することによって計算複雑度を下げながらも精度とロバスト性の確保をねらった。

速度推定では、画像処理による移動量の抽出では従来複数枚の画像間における差分を見る手法が主流であったが、計算規模と速度の面に不安があった。そこで、単一の画像に含まれる劣化過程としての動きブレに着目し、ブレの成分に移動量が保存されていることか

ら速度の推定を行う手法を提案した。

どちらの手法についても、月面を模擬した画像によってその推定精度が確認された。また、それらの手法について、実際に宇宙機で使用される耐放射線性の計算機への実装評価を行うことで、その搭載性について確認した。

これらのことにより、現状の宇宙機用計算機環境においても、自律的かつリアルタイムに処理が可能な地形相対航法の手法について示した。

第4章「将来の低リソース探査に向けた地形相対航法」では、将来の探査においては低エネルギーリソース環境下にて、高度な地形相対航法の必要性があることを予期し、その解決手段として、従来の計算機アーキテクチャとは異なる方式で処理を行う、ニューロモーフィックコンピューティングに着目した。ニューロモーフィックコンピューティングで使用されるニューロモーフィックプロセッサは脳神経のふるまいを模倣しており、その動作はニューラルネットワークとして実行される。一方で、超低消費電力性が期待される中で具体的なアプリケーションの構築については研究例が少なく、宇宙分野では例が見られない。

そこで、地形相対航法のアプリケーションとしてクレータ識別のニューラルネットワークを採用し、それをニューロモーフィックプロセッサで実行可能な形へと変換する手法を提案した。ANNの活性化関数とSNNの応答関数が同様の機能をネットワーク内で持つことに着目し、操作可能な活性化関数のふるまいを、実際の応答関数の形へと似せることにより、ANNでの学習結果をSNNに直接転送することで、SNNの学習を完了させるものである。

この手法はシミュレーションによって実際の月画像を用いて評価された。また、ニューロンの発火を記録することにより、発表されている複数のニューロモーフィックプロセッサにおいて実行されたときの、消費電力を概算した。

結果として、ニューラルネットワークを用いた地形相対航法のアプリケーションについて、従来の計算機よりも超低消費電力で実行可能であることを示した。

本研究では、現在の宇宙機における計算機環境で自律的に実施可能な、地形相対航法のタスクについて一つの解法を与えたのみならず、より将来のミッション制約を見据えた地形相対航法処理の発展についても論じている。その内容は、具体的なアプリケーションの適用範囲と設計法のような実用性の高いものから、航法タスクに関わらない汎用的なアプリケーション構築にも使用可能な汎用性の高いものまでを範囲としている。より将来の着陸探査においては、軌道上で非可視となる地域や溪谷などのチャレンジングな地形を対象とするものが予想される。そのような高機能性を必要とする着陸において、本論文がその一助となれば幸いである。

References

- [1] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, “Recognizing objects in range data using regional point descriptors,” in *Computer Vision - ECCV 2004*, T. Pajdla and J. Matas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 224–237.
- [2] A. E. Johnson and M. Hebert, “Using spin images for efficient object recognition in cluttered 3d scenes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, May 1999.
- [3] Y. Cheng, A. Johnson, and L. Matthies, “Mer-dimes: a planetary landing application of computer vision,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, June 2005, pp. 806–813 vol. 1.
- [4] T. Kubota, T. Hashimoto, J. Kawaguchi, M. Uo, and K. Shirakawa, “Guidance and navigation of hayabusa spacecraft for asteroid exploration and sample return mission,” in *2006 SICE-ICASE International Joint Conference*, Oct 2006, pp. 2793–2796.
- [5] S. Li, X. Jiang, and T. Tao, “Guidance Summary and Assessment of the Chang’e-3 Powered Descent and Landing,” *Journal of Spacecraft and Rockets*, vol. 53, pp. 258–277, Mar. 2016.
- [6] “Apollo 11: Mission report,” NASA, Tech. Rep. SP-238, 1971.
- [7] B. Harvey, *Soviet and Russian lunar exploration*, ser. Springer-Praxis books in space exploration. Springer, published in association with Praxis Publishing, 2007.
- [8] “Surveyor 1 a preliminary report,” NASA, Tech. Rep. SP-126, 1966.
- [9] “Surveyor 3 mission report,” JPL, Tech. Rep. TR 32-1177, June 1967.
- [10] “Surveyor 5 preliminary report,” NASA, Tech. Rep. SP-163, Dec 1967.
- [11] “Surveyor 6 mission report,” JPL, Tech. Rep. TR 32-1262, June 1968.

-
- [12] “Surveyor 7 mission report. part 1 - mission description and performance,” JPL, Tech. Rep. TR 32-1264 PT-1, Feb 1969.
- [13] “Apollo 12 mission report,” NASA, Tech. Rep. MSC-01855, Mar 1970.
- [14] “Apollo 14 mission report,” NASA, Tech. Rep. MSC-04112, Apr 1971.
- [15] “Apollo 15 mission report,” NASA, Tech. Rep. MSC-05161, Dec 1971.
- [16] “Apollo 16 mission report,” NASA, Tech. Rep. MSC-07230, Aug 1972.
- [17] “Apollo 17 mission report,” NASA, Tech. Rep. JSC-07904, Mar 1973.
- [18] N. L. Johnson, “Handbook of Soviet lunar and planetary exploration,” *NASA STI/Recon Technical Report A*, vol. 80, 1979.
- [19] G. A. SOFFEN, “The viking project,” *Journal of Geophysical Research*, vol. 82, pp. 3959–3970, 1977.
- [20] M. Harvey, B. *The new Russian space programme : from competition to collaboration*, ser. Wiley-Praxis series in space science and technology. John Wiley, 1996.
- [21] M. P. Golombek, “The mars pathfinder mission,” *Journal of Geophysical Research: Planets*, vol. 102, no. E2, pp. 3953–3965, 1997.
- [22] J. A. Crisp, M. Adler, J. R. Matijevic, S. W. Squyres, R. E. Arvidson, and D. M. Kass, “Mars exploration rover mission,” *Journal of Geophysical Research: Planets*, vol. 108, no. E12, pp. n/a–n/a, 2003, 8061.
- [23] A. Johnson, R. Willson, Y. Cheng, J. Goguen, C. Leger, M. Sanmartin, and L. Matthies, “Design through operation of an image-based velocity estimation system for mars landing,” *International Journal of Computer Vision*, vol. 74, no. 3, pp. 319–341, 09 2007.
- [24] J.-P. Lebreton, O. Witasse, C. Sollazzo, T. Blancquaert, and e. al, “An overview of the descent and landing of the Huygens probe on Titan,” *Nature*, vol. 438, no. 7069, pp. 758–64, Dec 08 2005.
- [25] P. N. Desai, J. L. Prince, E. M. Queen, M. Schoenenberger, J. R. Cruz, and M. R. Grover, “Entry, descent, and landing performance of the Mars Phoenix lander,” *Journal of Spacecraft and Rockets*, vol. 48, no. 5, pp. 798–808, Sep 2011.
- [26] R. Prakash, P. D. Burkhart, A. Chen, K. A. Comeaux, C. S. Guernsey, D. M. Kipp, L. V. Lorenzoni, G. F. Mendeck, R. W. Powell, T. P. Rivellini, A. M. S. Martin, S. W. Sell, A. D. Steltzner, and D. W. Way, “Mars science laboratory entry, descent, and landing system overview,” in *2008 IEEE Aerospace Conference*, March 2008, pp. 1–18.

-
- [27] 澤井 秀次郎, 福田 盛介, 坂井 真一郎, 櫛木 賢一, 荒川 哲人, 佐藤 英一, 冨木 淳史, 道上 啓亮, 河野 太郎, 岡崎 峻, 久木田 明夫, 宮澤 優, 植田 聡史, 戸部裕史, 丸 祐介, 下地 治彦, 清水 康弘, 芝崎 裕介, 島田 貞則, 横井 貴弘, 藪下剛, 佐藤 賢一郎, 中村 和行, 久原 隆博, 高見 剛史, 田中 伸彦, 古川克己, “小型月着陸実証機 slim のシステム設計”, *航空宇宙技術*, vol. 17, pp. 35–43, 2018.
- [28] K. Kuramoto, Y. Kawakatsu, M. Fujimoto, H. Genda, T. Imamura, S. Kameda, K. Matsumoto, H. Miyamoto, T. Morota, H. Nagaoka, T. Nakamura, K. Ogawa, H. Otake, M. Ozaki, S. Sasaki, H. Senshu, S. Tachibana, N. Terada, T. Usui, K. Wada, S. Watababe, and MMX Study Team, “Martian Moons Exploration (MMX) Conceptual Study Update,” in *Lunar and Planetary Science Conference*, ser. Lunar and Planetary Science Conference, vol. 49, Mar. 2018, p. 2143.
- [29] P. Fortescue, G. Swinerd, and J. Stark, *Spacecraft systems engineering*. John Wiley & Sons, 2011.
- [30] G. Lentaris, K. Maragos, I. Stratakos, L. Papadopoulos, O. Papanikolaou, D. Soudris, M. Lourakis, X. Zabulis, D. Gonzalez-Arjona, and G. Furano, “High-performance embedded computing in space: Evaluation of platforms for vision-based navigation,” *Journal of Aerospace Information Systems*, vol. 15, no. 4, pp. 178–192, 2018.
- [31] A. Cady and D. Geer, “Rad750 system flight computer,” 2002.
- [32] “CPU-World,” <http://www.cpu-world.com/benchmarks/>.
- [33] X. Iturbe, D. Keymeulen, E. Ozer, P. Yiu, D. Berisford, K. Hand, and R. Carlson, “An integrated soc for science data processing in next-generation space flight instruments avionics,” in *2015 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, Oct 2015, pp. 134–141.
- [34] D. Rudolph, C. Wilson, J. Stewart, P. Gauvin, A. D. George, H. Lam, G. Crum, M. J. Wirthlin, A. Wilson, and A. Stoddard, “Csp: A multifaceted hybrid architecture for space computing,” 2014.
- [35] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986.
- [36] C. Harris and M. Stephens, “A combined corner and edge detector,” in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.
- [37] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov 2004.
- [38] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,”

- in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.
- [39] Y. Cheng, A. E. Johnson, L. H. Matthies, and C. F. Olson, “Optical landmark detection for spacecraft navigation,” 2003.
- [40] 岡田 怜史, 中浜 優佳, 森部 美沙子, 鎌田 弘之, 狩谷 和季, 高玉 圭樹, 石田 貴行, 福田 盛介, 澤井 秀次郎, 坂井 真一郎, “主成分分析によるクレータ座標・サイズの検出とその評価”, *航空宇宙技術*, vol. 17, pp. 61–67, 2018.
- [41] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. Frahm, “Usac: A universal framework for random sample consensus,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 2022–2038, Aug 2013.
- [42] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, Feb 1992.
- [43] H. Morita, K. Shirakawa, T. Kubota, T. Hashimoto, and J. Kawaguchi, “Hayabusa’s real-time landmark tracking navigation for descents and touchings-downs.” Reston: American Institute of Aeronautics and Astronautics, 2006.
- [44] R. P. Weicker, “Dhrystone: A synthetic systems programming benchmark,” *Commun. ACM*, vol. 27, no. 10, pp. 1013–1030, Oct. 1984.
- [45] C. D. KUGLIN, “The phase correlation image alignment method,” *Proc. Int. Conf. on Cybernetics and Society, 1975*, pp. 163–165, 1975.
- [46] A. E. Johnson and J. F. Montgomery, “Overview of terrain relative navigation approaches for precise lunar landing,” in *2008 IEEE Aerospace Conference*, March 2008, pp. 1–10.
- [47] V. Tabur, “Fast algorithms for matching CCD images to a stellar catalogue,” *Publ. Astron. Soc. Austral.*, vol. 24, p. 189, 2007.
- [48] S. Ranade and A. Rosenfeld, “Point pattern matching by relaxation,” *Pattern Recognition*, vol. 12, no. 4, pp. 269 – 275, 1980.
- [49] H. Ogawa, “Labeled point pattern matching by fuzzy relaxation,” *Pattern Recognition*, vol. 17, no. 5, pp. 569 – 573, 1984.
- [50] D. P. Huttenlocher and S. Ullman, “Recognizing solid objects by alignment with an image,” *International Journal of Computer Vision*, vol. 5, no. 2, pp. 195–212, Nov 1990.
- [51] S. Irani and P. Raghavan, “Combinatorial and experimental results for randomized point matching algorithms,” *Computational Geometry*, vol. 12, no. 1, pp. 17 – 31, 1999.

-
- [52] S.-H. Chang, F.-H. Cheng, W.-H. Hsu, and G.-Z. Wu, "Fast algorithm for point pattern matching: Invariant to translations, rotations and scale changes," *Pattern Recognition*, vol. 30, no. 2, pp. 311 – 320, 1997.
- [53] B. D. Pollard and C. W. Chen, "A radar terminal descent sensor for the mars science laboratory mission," in *2009 IEEE Aerospace conference*, March 2009, pp. 1–10.
- [54] S. FUKUDA, T. SAKAI, and T. MIZUNO, "Landing radar simulation with digital terrain models," *TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES, AEROSPACE TECHNOLOGY JAPAN*, vol. 10, no. 28, pp. Pd.61–Pd.66, 2012.
- [55] T. MIZUNO, S. FUKUDA, T. SAKAI, Y. OKADA, T. SATO, and K. KISHIMOTO, "Dynamic performance of a landing radar bbm in field experiments," *TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES, SPACE TECHNOLOGY JAPAN*, vol. 7, no. ists26, pp. Td.1–Td.5, 2009.
- [56] M. Cannon, "Blind deconvolution of spatially invariant image blurs with phase," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 1, pp. 58–63, Feb 1976.
- [57] 狩谷和季, 福田盛介, "月惑星着陸機のための単一画像を用いた速度推定の試み", 2016 年電子情報通信学会総合大会, 2016.
- [58] H. Ji and C. Liu, "Motion blur identification from image gradients," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.
- [59] H. Asai, Y. Oyamada, J. Pilet, and H. Saito, "Cepstral analysis based blind deconvolution for motion blur," in *2010 IEEE International Conference on Image Processing*, Sept 2010, pp. 1153–1156.
- [60] Y. Shimamoto, Q. Chen, H. Wu, X. Ruan, and H. Matsumoto, "Efficient cepstrum analysis based unlm psf estimation in single blurred image," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Nov 2015, pp. 291–295.
- [61] I. Impulse Accelerated Technologies, "Impulse accelerated technologies, inc," <http://www.impulseaccelerated.com/>.
- [62] K. Wakabayashi, "Cyberworkbench: integrated design environment based on c-based behavior synthesis and verification," in *2005 IEEE VLSI-TSA International Symposium on VLSI Design, Automation and Test, 2005. (VLSI-TSA-*

- DAT*)., April 2005, pp. 173–176.
- [63] I. Y Explorations, “Yxi,” <http://www.yxi.com/>.
- [64] 日本電気株式会社, “Cyberworkbench のメリットとユーザー様導入事例,” Apr 2016.
- [65] H. Markram, “The blue brain project,” *Nature Reviews Neuroscience*, vol. 7, no. 2, pp. 153–160, 2006.
- [66] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, “A million spiking-neuron integrated circuit with a scalable communication network and interface,” *Science*, vol. 345, no. 6197, pp. 668–673, 2014.
- [67] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, “Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations,” *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699–716, May 2014.
- [68] G. Bersuker, M. Mason, and K. L. Jones, “Neuromorphic computing: The potential for high-performance processing in space,” Tech. rep., The Aerospace Corporation, Tech. Rep., 2018.
- [69] R. F. Service, “The brain chip,” *Science*, vol. 345, no. 6197, pp. 614–616, 2014.
- [70] G. Indiveri, E. Chicca, and R. J. Douglas, “Artificial cognitive systems: From vlsi networks of spiking neurons to neuromorphic cognition,” *Cognitive Computation*, vol. 1, no. 2, pp. 119–127, Jun 2009.
- [71] E. M. Izhikevich, “Which model to use for cortical spiking neurons?” *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1063–1070, Sep. 2004.
- [72] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.
- [73] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, Dec 1943.
- [74] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.
- [75] D. HUBEL, “Receptive fields of single neurons in the cat’s striate cortex,” *J*

-
- Physiol (Lond)*, vol. 148, pp. 574–591, 1959.
- [76] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr 1980.
- [77] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec 1989.
- [78] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, *Object Recognition with Gradient-Based Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 319–345.
- [79] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gomez-Rodriguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, S. Liu, R. Douglas, P. Hafliger, G. Jimenez-Moreno, A. Civit Ballcels, T. Serrano-Gotarredona, A. J. Acosta-Jimenez, and B. Linares-Barranco, “Caviar: A 45k neuron, 5m synapse, 12g connects/s aer hardware sensoryprocessing learningactuating system for high-speed visual object recognition and tracking,” *IEEE Transactions on Neural Networks*, vol. 20, no. 9, pp. 1417–1438, Sep. 2009.
- [80] J. A. Pérez-Carrasco, B. Zhao, C. Serrano, B. Acha, T. Serrano-Gotarredona, S. Chen, and B. Linares-Barranco, “Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing—application to feedforward convnets,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2706–2719, Nov 2013.
- [81] Y. Cao, Y. Chen, and D. Khosla, “Spiking deep convolutional neural networks for energy-efficient object recognition,” *International Journal of Computer Vision*, vol. 113, no. 1, pp. 54–66, May 2015.
- [82] P. U. Diehl, D. Neil, J. Binas, M. Cook, S. Liu, and M. Pfeiffer, “Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing,” in *2015 International Joint Conference on Neural Networks (IJCNN)*, July 2015, pp. 1–8.
- [83] P. O’Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, “Real-time classification and sensor fusion with a spiking deep belief network,” *Frontiers in neuroscience*, vol. 7, pp. 178–178, Oct 2013, 24115919[pmid].
- [84] E. Hunsberger and C. Eliasmith, “Spiking deep networks with lif neurons,” 2015.

-
- [85] R. Gütig and H. Sompolinsky, “The tempotron: a neuron that learns spike timing–based decisions,” *Nature neuroscience*, vol. 9, no. 3, p. 420, 2006.
- [86] F. Ponulak and A. Kasiński, “Supervised learning in spiking neural networks with resume: Sequence learning, classification, and spike shifting,” *Neural Comput.*, vol. 22, no. 2, pp. 467–510, Feb. 2010.
- [87] S. M. Bohte, J. N. Kok, and H. L. Poutré, “Error-backpropagation in temporally encoded networks of spiking neurons,” *Neurocomputing*, vol. 48, no. 1, pp. 17 – 37, 2002.
- [88] J. H. Lee, T. Delbruck, and M. Pfeiffer, “Training deep spiking neural networks using backpropagation,” *Frontiers in Neuroscience*, vol. 10, p. 508, 2016.
- [89] Y. Jin, W. Zhang, and P. Li, “Hybrid macro/micro level backpropagation for training deep spiking neural networks,” 2018.
- [90] D. O. Hebb, *The organization of behavior: a neuropsychological theory*. Science Editions, 1962.
- [91] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, “Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps,” *Science*, vol. 275, no. 5297, pp. 213–215, 1997.
- [92] Y. Bengio, T. Mesnard, A. Fischer, S. Zhang, and Y. Wu, “Stdp as presynaptic activity times rate of change of postsynaptic activity,” 2015.
- [93] S. Bartunov, A. Santoro, B. A. Richards, L. Marris, G. E. Hinton, and T. P. Lillicrap, “Assessing the scalability of biologically-motivated deep learning algorithms and architectures,” in *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, ser. NIPS’18. USA: Curran Associates Inc., 2018, pp. 9390–9400.
- [94] M.-O. Gewaltig and M. Diesmann, “Nest (neural simulation tool),” [http://www.scholarpedia.org/article/NEST_\(NEural_Simulation_Tool\)](http://www.scholarpedia.org/article/NEST_(NEural_Simulation_Tool)).
- [95] D. Goodman and R. Brette, “Brian: a simulator for spiking neural networks in python,” *Frontiers in Neuroinformatics*, vol. 2, p. 5, 2008.
- [96] J. Schemmel, D. Brüderle, A. Gribbl, M. Hock, K. Meier, and S. Millner, “A wafer-scale neuromorphic hardware system for large-scale neural modeling,” in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 1947–1950.
- [97] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, “The spinnaker project,” *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, May 2014.
- [98] P. Lichtsteiner, C. Posch, and T. Delbruck, “A 128×128 120 db 15 μ s latency

-
- asynchronous temporal contrast vision sensor,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb 2008.
- [99] Y. Zhang and B. Wallace, “A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification,” in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 253–263.
- [100] E. Stamatias, D. Neil, M. Pfeiffer, F. Galluppi, S. B. Furber, and S.-C. Liu, “Robustness of spiking deep belief networks to noise and reduced bit precision of neuro-inspired hardware platforms,” *Frontiers in Neuroscience*, vol. 9, p. 222, 2015.
- [101] T. Ishida, M. Takahashi, and S. Fukuda, “Crater detection robust to illumination and shape changes using convolutional neural network,” in *32nd International Symposium on Space Technology and Science*, 2019, pp. 2019–d–073.
- [102] A. P. Davison, D. Brüderle, J. Eppler, J. Kremkow, E. Müller, D. Pecevski, L. Perrinet, and P. Yger, “Pynn: A common interface for neuronal network simulators,” *Frontiers in neuroinformatics*, vol. 2, pp. 11–11, Jan 2009, 19194529[pmid].
- [103] J. M. Cruz-Albrecht, M. W. Yung, and N. Srinivasa, “Energy-efficient neuron, synapse and stdp integrated circuits,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, pp. 246–256, 2012.
- [104] C. Isci, “Workload adaptive power management with live phase monitoring and prediction,” Ph.D. dissertation, Princeton University, 2007.
- [105] T. M. Lovelley and A. D. George, “Comparative analysis of present and future space-grade processors with device metrics,” *Journal of Aerospace Information Systems*, vol. 14, no. 3, pp. 184–197, 2017.
- [106] J. Marshall, D. Rickard, D. Sova, H. Miller, R. Lapihuska, A. Dennis, and M. Graziano, “Heterogeneous high performance computing modules for next generation onboard processing,” in *2017 IEEE Aerospace Conference*, March 2017, pp. 1–10.

List of Publications

査読付論文

- (1) Kazuki Kariya and Seisuke Fukuda: Initial Study of Neuromorphic Application for Vision-based Navigation, Aerospace Technology Japan (in print)
- (2) 狩谷和季, 石田貴行, 澤井秀次郎, 木下智雄, 梶原邦弘, 岩佐修, 福田盛介: 高精度月着陸のためのクレータで構成する線分情報を用いた位置推定手法とその評価, 航空宇宙技術, 17, pp. 79-87, 2018.
- (3) 岡田 怜史, 鎌田 弘之, 中浜 優佳, 森部 美沙子, 狩谷 和季, 高玉 圭樹, 石田貴行, 福田 盛介, 澤井 秀次郎, 坂井 真一郎: 主成分分析によるクレータ座標・サイズの検出とその評価, 航空宇宙技術, 17, pp. 61-67, 2018.
- (4) 狩谷和季, 福田盛介. 月惑星着陸機における単一画像のブレを用いた速度推定法の検討: 日本航空宇宙学会論文集, 65, 5, pp. 184-192, 2017.

口頭発表

- (1) Kazuki Kariya and Seisuke Fukuda: Neuromorphic Computing for Spacecraft's Terrain Relative Navigation: A Case of Event-Based Crater Classification Task, Neuro-inspired Computational Elements Workshop, Heidelberg (Germany), March, 2020.
- (2) 狩谷和季, 福田盛介: クレータ分類を例としたニューロモーフィックコンピューティングの宇宙機航法に対する適用, ニューロコンピューティング研究会, 2020-01-NLP-NC-24, 沖縄, 1月, 2020.
- (3) Kazuki Kariya and Seisuke Fukuda: Study of Neuromorphic Application using Spiking Neural Network for Terrain Relative Navigation, International Planetary Probe Workshop 2019, Oxford (UK), July, 2019.
- (4) Kazuki Kariya and Seisuke Fukuda: Module Study of Neuromorphic Appli-

- cation for Vision-based Navigation, 32nd International Symposium on Space Technology and Science, Fukui (Japan), June, 2019.
- (5) 石田貴行, 狩谷和季, 福田盛介, 鎌田弘之: クレータを特徴点とした画像照合航法の実装に向けた最適化と精度評価, 第 61 回 宇宙科学技術連合講演会, 1C11, 新潟, 10 月, 2017.
 - (6) Kazuki Kariya, Takayuki Ishida, Shujiro Sawai and Seisuke Fukuda: Position and Displacement Estimation Using Crater-Based Line Segments for Pinpoint Lunar Landing, 26 th International Symposium on Space Flight Dynamics, Ehime (Japan), June, 2017.
 - (7) 狩谷和季, 石田貴行, 澤井秀次郎, 福田盛介: 月高精度着陸のためのクレータ点を用いた位置推定法における探索手法の高度化, 2017 年電子情報通信学会総合大会, 電子情報通信学会総合大会講演論文集 2017, B-2-41, 愛知, 3 月, 2017.
 - (8) 狩谷和季, 福田盛介: 高精度月着陸のためのクレータで構成する線分情報を用いた位置推定手法とその評価, 第 60 回宇宙科学技術連合講演会, 宇宙科学技術連合講演会講演集 60, P58, 北海道, 9 月, 2016.
 - (9) 石田貴行, 狩谷和季, 福田盛介, 木下智雄, 梶原邦弘: 高位合成による探査機搭載画像処理アルゴリズムの FPGA スキーム, 宇宙・航行エレクトロニクス研究会, 信学技報 116(115), pp. 31-36, 茨城, 6 月, 2016.
 - (10) 狩谷和季, 福田盛介: 月着陸機のための単一画像を用いた速度推定の試み, 2016 年電子情報通信学会総合大会, 電子情報通信学会総合大会講演論文集 2016, B-2-5, 福岡, 3 月, 2016.

Acknowledgement

本研究並びに研究生活を進めるにあたり，終始懇切なる御指導と御鞭撻を賜りました福田盛介准教授と津田雄一准教授に心から感謝致します。

本論文でまとめた研究の一部は，宇宙航空研究開発機構職員の石田貴行氏をはじめSLIM プロジェクトにおける画像分科会の皆様に御協力頂きました。皆様に感謝致します。

加えて，学生生活において公私共に関わらせていただいた，宇宙科学研究所の福田研究室の方々や津田研究室の方々，また学生の皆様に御礼申し上げます。

また，様々な面でサポートして頂いた両親と弟に心から感謝致します。