

論文 / 著書情報
Article / Book Information

題目(和文)	ステレオ動画像を用いた車両の前方環境認識に関する研究
Title(English)	
著者(和文)	関晃仁
Author(English)	Akihito Seki
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第6808号, 授与年月日:2007年3月26日, 学位の種別:課程博士, 審査員:
Citation(English)	Degree:Doctor of Engineering, Conferring organization: Tokyo Institute of Technology, Report number:甲第6808号, Conferred date:2007/3/26, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

ステレオ動画像を用いた
車両の前方環境認識に関する研究

平成18年度 博士論文

東京工業大学
大学院 理工学研究科
機械制御システム専攻

関 晃仁

論文要旨

近年、自動車のインテリジェント化を実現するために、さまざまな技術が世界中で研究開発されている。既に車線維持支援機能、歩行者検出機能、駐車支援機能、運転者状態検知機能などを持つ車両が市販され始めている。これらのシステムは、センシングデバイスとしてカメラを用いており、車載カメラから得られる画像を処理することによって、危険検知や車線維持制御等に必要な情報を抽出している。今後は、より広範囲な環境で、安定かつ正確に画像から情報を抽出するための高度な画像処理技術の実現が求められている。

本論文では、車載ステレオカメラを利用した、一般道路環境でも利用できる車両の前方環境認識手法を提案する。提案手法では、車載カメラの映像から空間中の道路領域を推定し、それらを活用して危険領域や障害物を検出し、その空間位置を特定する。さらに、自車両の運動を推定する。加えて、シーン中の各点の空間位置とモーションを同時に推定し、それを利用して障害物の運動を推定する。また、一般的なカメラを利用した非同期ステレオに対しても、3次元計測やモーション推定が行える方法を提案する。本論文は以下の全8章から構成される。

まず、第1章「緒言」において、本研究の背景や目的、そして本論文の構成について述べる。

第2章「道路領域抽出と道路面の姿勢推定」では、ステレオ画像中の道路領域を抽出する手法について述べる。本手法では、空間中の平面とステレオ画像の幾何学的な性質を利用して、道路領域を抽出する。続いて、車載カメラと道路面間の相対的な姿勢を推定する方法について述べる。これらは、道路環境を認識する上で重要な先見情報となる。

第3章「仮想投影面画像の生成と危険領域検出」では、第2章で抽出された道路領域と、推定された道路面の姿勢を用いた危険領域検出手法を提案する。本手法では、障害物を陽に検出することなく、道路領域端の位置とその変化具合から、自車両にとって危険となる領域を検出する。本手法では、道路面に対して仮想的に垂直上方から観測した画像(VPP画像)を生成し、実空間における道路領域を求める。本手法は、ステレオ対応点探索をせずに、危険領域までの距離を簡単に求めることができ、かつ領域の危険性評価にトラッキング処理が必要ないといった利点がある。

第4章「道路面を基準とした障害物検出」では、抽出された道路領域と道路面の姿勢を

利用して、第3章の手法とは異なり、障害物を陽に検出する手法を提案する。本手法では、道路面の姿勢とステレオ計測を利用して、自車両の走行する可能性のある空間に含まれる立体物を検出し、さらにその位置を計測する。ステレオ計測で必要とされる対応点探索の計算コストを、道路領域と道路面の姿勢を用いることで減少させ、対応点探索の誤りなどによって発生する誤検出を、道路環境の障害物の性質を利用した処理を施すことで防止する。

第5章「自車両の運動推定」では、自車両の運動を推定する手法を提案する。本手法では、車両が道路面上を移動する性質を利用し、車両の道路面上における回転と並進運動の3つの未知パラメータを、時系列のVPP画像間のマッチングによって推定する。ここでは、静止座標系である道路領域と、車載カメラと道路間の関係を利用した上で、第一段階で車両の運動に基づくモデル化を行ない、簡単なマッチングだけで安定に運動推定を行なう。さらに、次の段階として、前段階で推定された運動パラメータを、勾配法の考え方を利用して更新することで、より高精度に推定する。本手法は、特定の注目点のトラッキングが必要ないため、頑健でかつ高精度な運動推定が行える。

第6章「空間内の位置とモーシヨンの同時推定」では、時系列ステレオ画像を利用して、シーンに含まれる各点の3次元位置と、カメラに対する3次元モーシヨンを同時に最適化して推定する。これを実現するため、ステレオ画像間の対応点探索と時系列画像間のトラッキングに相当する作業を同時に行う手法を提案し、それらを別々に行うより精度よく推定できることを確認する。本手法を利用することで、例えば、第4章の手法によって検出された障害物のカメラに対する相対モーシヨンを推定することが出来、さらに、第5章の自車両のモーシヨン推定結果を利用して、障害物の3次元モーシヨンを静止座標系を基準とした絶対モーシヨンに変換することもできる。それにより、障害物が移動物であるのか、静止物であるのかが判別できる。

第7章の「非同期ステレオカメラを利用した試み」では、第6章の手法を拡張し、非同期時系列ステレオ画像に対して、2フレーム以上の対応づけを同時に行なうことで、ステレオ画像間の時間オフセットと共に、3次元位置とモーシヨンが推定できる方法を提案する。

最後に、第8章「結言」において、本研究の総括と今後の課題についてまとめる。

目次

第1章	緒言	1
1.1	本研究の背景	1
1.2	本研究の目的	2
1.3	本論文の構成	3
第2章	道路領域抽出と道路面の姿勢推定	5
2.1	道路領域抽出	6
2.1.1	道路抽出処理の流れ	6
2.2	道路面の姿勢推定	9
2.2.1	ステレオカメラ間のベースライン長のみが既知の場合	9
2.2.2	外部パラメータが既知の場合	11
2.3	本章のまとめ	12
第3章	仮想投影面画像の生成と危険領域検出	13
3.1	仮想投影面画像	13
3.1.1	VPP 画像生成	14
3.1.2	仮想カメラの内部パラメータと実空間距離	16
3.2	危険領域検出	17
3.2.1	方向別距離測定	17
3.2.2	方向別相対速度測定	18
3.3	実験結果	18
3.3.1	原画像と VPP 画像に情報表示	18
3.3.2	危険領域検出結果	19
3.4	本章のまとめ	21
第4章	道路面を基準とした障害物検出	22
4.1	関連研究	22
4.2	処理の全体図	24
4.3	ステレオ計測と障害物の検出	26

4.3.1	道路領域と姿勢を利用した対応点探索の限定	26
4.3.2	カメラ座標系から道路座標系への変換	27
4.3.3	道路面への投票を利用した障害物の検出	29
4.4	障害物の描写	31
4.4.1	道路領域の修正	32
4.5	実験結果	33
4.5.1	道路領域と障害物の検出結果	33
4.5.2	障害物の検出率	40
4.5.3	他手法との比較実験	41
4.5.4	誤検出について	43
4.6	本章のまとめ	45
第5章	自車両の運動推定	46
5.1	関連研究	46
5.2	運動推定手法の流れ	48
5.3	自車両の運動推定	49
5.3.1	車両の運動モデルを利用した運動パラメータの推定	49
5.3.2	VPP 画像を利用した移動ベクトル推定と車両の運動推定	51
5.3.3	運動パラメータの更新	53
5.4	フィルタリング	55
5.5	実験結果	56
5.5.1	合成画像による実験	56
5.5.2	実画像による実験	59
5.6	本章のまとめ	63
第6章	空間内の位置とモーシヨンの同時推定	66
6.1	関連研究	66
6.2	対象の奥行きとモーシヨンと画像位置の関係	68
6.2.1	時系列ステレオ画像間の対応位置	69
6.3	奥行きとモーシヨンの同時推定	70
6.3.1	推定アルゴリズム	71
6.3.2	パラメータ推定の過程と画像間の関係	74
6.3.3	Implementation	74
6.4	実験結果	76
6.4.1	合成画像実験 1	76
6.4.2	合成画像実験 2	81

6.4.3	実画像実験	83
6.5	障害物のモーション推定	84
6.5.1	絶対モーション推定結果	86
6.6	本章のまとめ	88
第7章	非同期ステレオカメラを利用した試み	90
7.1	関連論文	91
7.2	画像間の関係	92
7.2.1	同一カメラの画像間の関係	93
7.2.2	異なるカメラの画像間の関係	93
7.3	同時推定アルゴリズム	93
7.3.1	コスト関数	94
7.3.2	コスト関数の最小化	94
7.3.3	推定が不可能な場合	100
7.3.4	Implementation	101
7.4	実験結果	104
7.4.1	合成画像実験	104
7.4.2	実画像実験	106
7.5	非同期ステレオを利用した道路環境認識についての考察	108
7.6	本章のまとめ	110
第8章	結言	112
8.1	本論文の結論	112
8.2	今後の課題	114
	参考文献	115
	付録A 実験環境	124
	付録B Rectification	126
	付録C 運動推定のフィルタリング	130
C.1	“Roughness”と分散の関係	130

目次

1.1	各手法の関連とそれに対応する章の関係	3
2.1	カメラと道路面の姿勢変化	5
2.2	道路抽出処理の流れ	7
3.1	VPP 画像の幾何関係	14
3.2	道路平面領域抽出結果およびその VPP 画像生成例. (青い領域が抽出された道路平面領域)	16
3.3	VPP 画像のスケールと実空間距離	17
3.4	道路領域の境界までの方向別距離測定	18
3.5	危険領域検出結果の出力例	19
3.6	市街地における実験結果 1(10 フレームごとに表示)	20
3.7	市街地における実験結果 2(10 フレームごとに表示)	20
3.8	一般道路における危険領域検出結果 1	20
3.9	一般道路における危険領域検出結果 2	20
4.1	車載カメラから見た道路面の見え方.	23
4.2	障害物: 車両の通過する可能性のある空間に存在する物体	24
4.3	障害物検出手法の全体の概要	25
4.4	対応点探索と平面領域の関係	26
4.5	カメラ座標系と道路座標系	28
4.6	道路障害物の特徴	29
4.7	車両の通過空間と投票処理 (左は投票処理, 右は投票結果)	30
4.8	障害物描写方法	31
4.9	障害物描写例	31
4.10	道路領域の修正	32
4.11	検出範囲	33
4.12	高速道路における障害物検出結果 (青領域が抽出された道路領域, 赤領域は検出された障害物領域)	35

4.13	一般道路における結果 (青領域が抽出された道路領域, 赤領域は検出された障害物領域)	36
4.14	降雪, 積雪した環境における障害物検出結果	37
4.15	障害物の位置検出結果	38
4.16	ワイパーが写り込んだ場合の結果	38
4.17	HDR カメラによる様々な環境下における結果	39
4.18	検出率評価シーン	40
4.19	V-disparity ベース手法との比較結果, 赤線は推定された水平線を表す.	42
4.20	第3章の手法と V-disparity ベース法との比較実験結果.	42
4.21	障害物の位置検出結果, VPP 画像と投票グリッドのスケールを合わせて表示.	43
4.22	フレアによる誤検出例	44
4.23	複数平面による誤検出例	44
5.1	運動推定手法の全体図	48
5.2	2次元空間上の剛体のモーション	49
5.3	車両の運動モデル	50
5.4	旋回半径と限界旋回速度から得られるヨーレート	52
5.5	移動ベクトルの推定手順	52
5.6	類似度分布と “Roughness” の関係	56
5.7	合成画像1の基準画像列 (左から Frame No.0,10,30)	57
5.8	合成画像1:ヨーレート	57
5.9	合成画像1: 並進ベクトルの推定位置の誤差	58
5.10	車両の移動と誤差	58
5.11	合成画像1:移動位置推定結果	58
5.12	合成画像2の基準画像列 (左から Frame No.0,10,20)	59
5.13	合成画像2: 運動推定結果	59
5.14	シーン1の画像列 (左上から右へ, 左下から右へ続く80フレームごとの連続画像). ただし, 色の明るい領域は抽出された平面領域を表す.	60
5.15	シーン1: Y軸方向速度	60
5.16	シーン1: ヨーレート	60
5.17	シーン2の画像列 (左上から右へ, 左下から右へ続く70フレームごとの連続画像). ただし, 色の明るい領域は抽出された平面領域を表す.	61
5.18	シーン2: Y軸方向速度	62
5.19	シーン2: ヨーレート	62
5.20	シーン2: 車両の軌跡	62

5.21	さまざまな環境下での運動推定結果	63
5.22	運動推定結果 (10 フレームごとに左から右へ, 上から下へ表示)	65
6.1	対象の奥行きとモーションに対する画像座標位置の関係	68
6.2	時系列ステレオ画像の対応点の関係	69
6.3	パラメータ推定の過程と画像間関係	74
6.4	奥行きと3次元モーション推定アルゴリズムの計算手順	75
6.5	入力した時系列ステレオ画像	76
6.6	カメラやオブジェクトの配置とそのモーション	76
6.7	奥行き推定結果	77
6.8	モーション推定結果 (下段は, 上段の結果を真上から見た様子)	77
6.9	画像利用ペア数に対する X 軸方向のモーション推定結果	78
6.10	従来手法による X 軸方向のモーション推定結果	79
6.11	利用した時系列ステレオ画像のペア数に対する RMS エラー	80
6.12	入力画像例 (右カメラの画像)	81
6.13	想定環境の平面図	81
6.14	3次元位置推定結果	81
6.15	モーション推定結果	82
6.16	3次元位置にそのモーションを併せて表示した結果	82
6.17	実画像実験の入力画像例	83
6.18	実画像実験のモーション推定結果	84
6.19	さまざまな視点から見た3次元位置とモーション推定結果	85
6.20	3次元位置とモーション推定結果の拡大図	86
6.21	障害物の絶対モーション推定結果	87
6.22	さまざまな視点から見た3次元位置と絶対モーション結果	88
6.23	障害物の絶対モーション推定結果. 1フレームおきの結果を表示	89
7.1	同期ステレオと3次元位置計測	91
7.2	非同期ステレオと3次元位置計測	91
7.3	非同期時系列ステレオ画像の対応点の関係	92
7.4	推定不可能な対象のモーション	101
7.5	複数点を用いたときの処理手順	102
7.6	カメラとオブジェクトの配置とそのモーション	104
7.7	入力した非同期ステレオ画像の一部	104
7.8	3次元位置とモーション推定結果	105
7.9	従来手法との推定精度比較結果	106

7.10	非同期ステレオ画像ペア	106
7.11	従来手法による再構成結果をさまざまな方向から見た様子	107
7.12	提案手法による再構成結果をさまざまな方向から見た様子((C)の緑の点は、推定の際に同時に利用した特徴点である)	107
7.13	別々のカメラで撮影された非同期時系列ステレオ画像	109
7.14	(a)3次元推定結果をさまざまな視点から観察した様子. (b)3次元推定位置とモーション推定結果. モーションを2倍に拡大して表示.	110
7.15	非同期ステレオを用いた際の各手法の関連	111
A.1	実験車両	124
A.2	実験機材(左側:外からみた様子. 右側:車内の様子)	124
A.3	キャリブレーション風景	124
B.1	エピポーラ幾何と Rectification	126
B.2	画像面と焦点面	126
C.1	1次元マッチングとパラボラフィッティング	130
C.2	推定成功時の例	131
C.3	推定失敗時の例	131
C.4	2次元マッチング時のパラボラ方向の選択	131

表目次

4.1	障害物の検出率	41
4.2	誤検出と未検出の距離別内訳	41
4.3	計算時間の比較	43
5.1	合成画像1：運動推定結果のRMSエラー	58

第1章 緒言

1.1 本研究の背景

近年、自動車のインテリジェント化を実現するために、さまざまな技術が世界中で研究開発されている。それにより、ドライバーの運転の負担を軽減することを目的として、車両を道路白線内へキープさせる車線維持支援機能 [77, 19] や、ステアリングやアクセルを自動的に行なうことで駐車を支援する機能 [77] などの運転支援技術、交通事故の発生を防ぐことを目的として、夜間などの運転者から見えにくい歩行者を検出するナイトビジョン機能 [78, 19] や、わき見や居眠りなどの運転者の状態を検知する機能 [77] などの予防安全技術を搭載した車両の市販が始まっている。

これらのシステムは、センシングデバイスとしてカメラを利用している。カメラは、安価で小型なセンサにもかかわらず、大量の情報を取得することができる。加えて、他のセンサでは得られない標識のパターン認識などにも利用可能であることから、このような運転支援や予防安全技術に有効なセンシングデバイスであり、今後も自車両の前方環境認識において、中心的な役割を果たしていくものと考えられる。しかし、撮影された映像は、そのままでは制御などに利用することができない。そのため、例えば、車線の位置等の所望な情報を得るための画像処理が必要である。加えて、自動車に搭載する先進装置の研究により、車載画像処理用プロセッサである Visconti®[92] や IMAPCAR®[35] などの高速な演算装置が開発されたことで、車載コンピュータの性能も向上し、より広範囲な環境で、安定かつ正確に画像から情報を抽出するための高度な画像処理技術の実現が求められている。

さて、運転支援・予防安全技術のうち、前走車との車間距離を一定に保つクルーズコントロール機能や前走車の急停車に対処して、自動的にブレーキをかけるブレーキアシスト機能では、車両前方の障害物を検出し、その3次元的な位置とそのモーションを取得することが制御に必要である。市販されているシステムでは、障害物検出やモーション推定にミリ波レーダーを利用したものが大部分を占めている。レーダーセンサは、比較的複雑なシステムではないため、導入はしやすいが、障害物までの距離測定を行える間隔がまばらなため、空間的な解像度が低くなってしまう。また、電波の反射によって物体を検出する仕組みであるため、検出対象の材質によって反射性能が異なる。そのため、例えば、車両などの金属製物体は検出しやすく、人や植木など非金属製物体は検出しにくいといった短

所がある。これらの性質のため、利用できる範囲が高速道路環境下などに限定されてしまう。しかし、霧のように視界が不良な状況下でもカメラと違い、利用できる。カメラの映像を利用した画像処理による方法では、検出対象の材質に依存せず、センサの空間解像度が高いことや、画角が広いことから、より広範囲かつ詳細に多様な物体の情報を得られる可能性がある。それぞれのセンサを組み合わせ、短所を補い合うセンサフュージョンによって、これまでは高速道路などに限定されていた安全支援技術の想定環境が、市街地のような複雑な環境下へと適用範囲が広がるだけでなく、自動運転のような、より高度な技術への礎になっていく可能性がある。

センサフュージョンをするには、まずそれぞれのセンサ単体を使った研究をし、単体のセンサで実現できる機能や、適用可能な範囲を見極める必要がある。そこで、本論文ではカメラのみを利用した障害物検出や、そのモーション推定手法を提案する。これまでも画像処理によって、障害物を検出する手法が提案されてきており、障害物と路面のパターンとの分離方法や、障害物の抽出方法にそれぞれの手法で工夫が見られるものの、道路環境認識の要である、道路に関する情報をほとんど活用していない。

1.2 本研究の目的

本研究では、車載カメラによるステレオ動画像を処理して、自車両の前方環境を認識するための情報を画像から抽出することを目的としている。そのために、以下に挙げる項目を実現する手法をそれぞれ提案する。

- 危険領域の検出

走行する際に、自車両にとって危険となる可能性がある領域をステレオ動画像から検出し、その領域の空間的な位置を求める。

- 障害物の検出

ステレオ画像から自車両が走行する可能性のある空間に存在する立体物を検出し、その空間位置を求める。

- 自車両の運動推定

自車両の運動パラメータである、並進運動と回転運動を推定する。

- 対象のモーション推定

対象の3次元空間中のモーションを推定する。

これらを実現する手法は、道路環境が整備されていない一般道路環境下においても機能することが重要である。

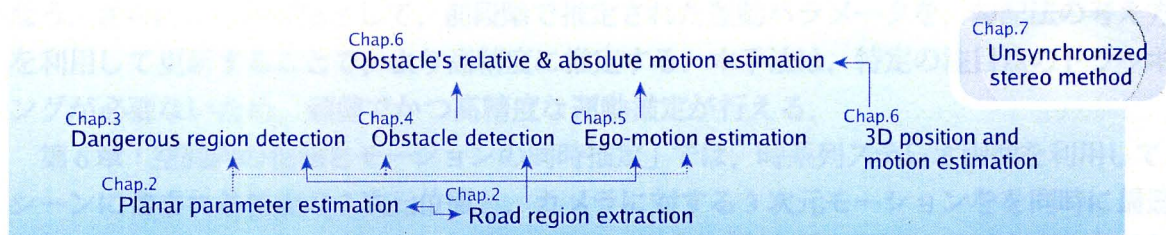


図 1.1: 各手法の関連とそれに対応する章の関係
 Fig. 1.1: Overview of proposed methods and chapters.

1.3 本論文の構成

本論文では、各章で関連のある研究を章ごとに述べるとともに、提案手法の位置付けをし、各章の最後にそれぞれの実験結果を示す。以下に、章ごとの内容を簡単に述べる。

まず、第2章「道路領域抽出と道路面の姿勢推定」では、ステレオ画像中の道路領域を抽出する手法について述べる。本手法では、空間中の平面とステレオ画像の幾何学的な性質を利用して抽出する。続いて、車載カメラと道路面間の相対的な姿勢を推定する方法について述べる。これらは、道路環境を認識する上で重要な先見情報となる。

第3章「仮想投影面画像の生成と危険領域検出」では、第2章で抽出された道路領域と、推定された道路面の姿勢を用いた危険領域検出手法を提案する。本手法では、障害物を陽に検出することなく、道路領域の位置とその変化具合から、自車両に危険となる領域を検出する。本手法では、道路面に対して仮想的に垂直上方から観測した画像(VPP画像)を生成し、実空間における道路領域を求める。本手法は、ステレオ対応点探索をせずに、危険領域までの距離を簡単に求めることができ、かつ領域の危険性評価にトラッキング処理が必要ないといった利点がある。

第4章「道路面を基準とした障害物検出」では、抽出された道路領域と道路面の姿勢を有効に使い、第3章の手法とは異なり、障害物を陽に検出する手法を提案する。本手法では、道路面の姿勢とステレオ計測を利用して、自車両の走行する可能性のある空間に含まれる立体物を検出し、さらにその位置を計測する。ステレオ計測で必要とされる対応点探索の計算コストを、道路領域と道路面の姿勢を用いることで減少させ、対応点探索の誤りになどによって発生する誤検出を、道路環境の障害物の性質を利用した処理を施すことで防止する。

第5章「自車両の運動推定」では、自車両の運動を推定する手法を提案する。本手法では、車両が道路面上を移動する性質を利用し、車両の道路面上における回転と並進運動の3つの未知パラメータを、時系列のVPP画像間のマッチングによって推定する。ここでは、静止座標系である道路領域と、車載カメラと道路間を利用した上で、第一段階で車両の運動に基づくモデル化を行ない、簡単なマッチングだけで安定に運動推定を行

なう。さらに、次の段階として、前段階で推定された運動パラメータを、勾配法の考え方を利用して更新することで、より高精度に推定する。本手法は、特定の注目点のトラッキングが必要ないため、頑健でかつ高精度な運動推定が行える。

第6章「空間内の位置とモーションの同時推定」では、時系列ステレオ画像を利用して、シーンに含まれる各点の3次元位置と、カメラに対する3次元モーションを同時に最適化して推定する。これを実現するため、ステレオ画像間の対応点探索と時系列画像間のトラッキングに相当する作業を同時に行う手法を提案し、それらを別々に行うより精度よく推定できることを確認する。本手法を利用することで、例えば、第4章の手法によって検出された障害物のカメラに対する相対モーションを推定することが出来、さらに、第5章の自車両のモーション推定結果を利用して、障害物の3次元モーションを静止座標系を基準とした絶対モーションに変換することもできる。それにより、障害物が移動物であるのか、静止物であるのかが判別できる。

これまで述べてきた手法では、ステレオ画像がそれぞれ同時刻に撮影されている同期ステレオカメラを想定していた。しかし、一般的な市販カメラを利用した場合には、両カメラで同一時刻に撮影ができない非同期ステレオカメラであることがほとんどである。また、形式の異なるカメラのように物理的に同期ができない場合も考えられる。このような非同期カメラでは、従来の方法を利用したステレオ計測やモーション推定ができず、障害物を検出することはできない。そこで、第7章の「非同期ステレオカメラを利用した試み」では、第6章の手法を拡張し、非同期時系列ステレオ画像に対して、2フレーム以上の対応づけを同時に行なうことで、ステレオ画像間の時間オフセットと共に、3次元位置とモーションが推定できる方法を提案する。

最後に、第8章「結言」において、本研究の総括と今後の課題についてまとめる。

以下に、本論文の内容と関連の深い研究発表の対応を記す。

第3章で述べる「仮想投影面画像の生成と危険領域検出」については、文献[50, 51, 43, 42]である。

第4章で述べる「道路面を基準とした障害物検出」については、文献[52, 59, 54, 58, 65, 64, 62, 61]である。

第5章で述べる「自車両の運動推定」については、文献[49, 53, 56, 63, 57]である。

第6章で述べる「空間内の位置とモーションの同時推定」については、文献[55]である。

第7章で述べる「非同期ステレオカメラを利用した試み」については、文献[60]である。

第2章 道路領域抽出と道路面の姿勢推定

車両の前方環境を認識するにあたって、道路面に関する情報を先に求めておくことは、障害物検出処理などの環境認識処理の高速化や、頑健性向上につながる重要な要素となる。

カメラと道路面間の姿勢は、走行中の車両の振動や出現する道路面によって変化する。図 2.1 は、走行中に撮影された画像の一部を取り出して、水平線を点線で書き入れたものである。このとき、カメラと自動車の配置はどちらも同じである。同図より、(B)では(A)に比べて、ピッチ角とロール角が変化していることが分かり、走行中に道路面とカメラ間の相対的な位置関係が変化している。

画像を利用して道路面とカメラ間の姿勢を推定するために、いくつかの方法が提案されており、例えば、道路白線の平行性などのモデルを利用した方法が提案されている [44, 13, 34, 33]。しかし、一般道路環境では必ずしも平行な白線があるわけではない。また、ステレオ画像間でマッチングを行ない、ステレオ計測をして道路面を推定する方法も提案されている [2, 27]。しかし、Williamsonら [83]が指摘しているように、道路面のようにフロントパラレルと大きく異なる面に対して、従来の変形を許さないウィンドウベースのマッチングを適用するのでは、左右画像のウィンドウ内での変形が大きく、対応点を正しく求めることが難しい。また、小ウィンドウを使うマッチングのため、道路面に十分なテクスチャがあることや、道路面以外の計測データを除去するために、画像に道路面が十分

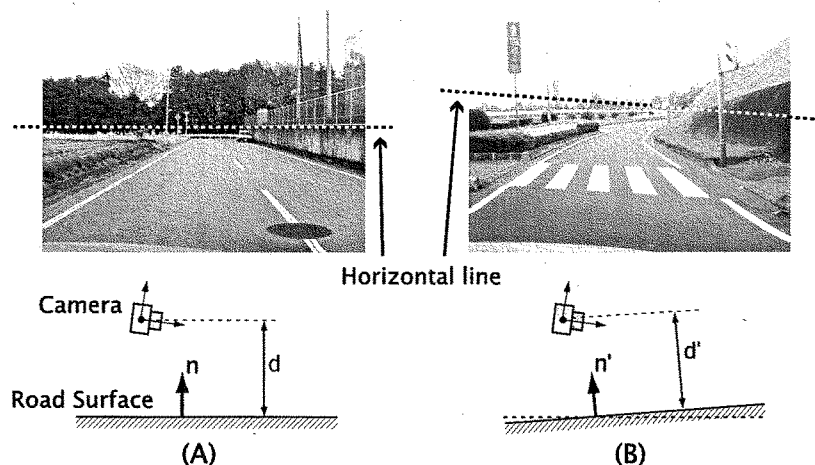


図 2.1: カメラと道路面の姿勢変化
Fig. 2.1: Example of planar pose change.

に写っている必要があると考えられる。

そこで本章では、道路環境認識において基盤となる道路領域抽出処理および道路面とカメラ間の相対的な姿勢を推定する処理について述べる。

2.1 道路領域抽出

本研究室では、連続するステレオ画像から平面領域を抽出する研究を行っていた [41, 40, 39, 38]。この方法では、空間中の平面がステレオ画像に写るときに生じる幾何学的関係を利用して、平面領域を抽出する。具体的には、空間中にある平面上の点が2枚のステレオ画像に投影されるとき、ステレオ画像における基準画像 I_b 上での同次座標を $\tilde{\mathbf{m}}_b$ 、参照画像 I_r 上での同次座標を $\tilde{\mathbf{m}}_r$ とすると、各々は式 (2.1) に示すように、2次元射影変換によって関係付けられる性質を利用する。

$$\tilde{\mathbf{m}}_r \simeq \mathbf{H}\tilde{\mathbf{m}}_b \quad (2.1)$$

ここで、 \mathbf{H} は3行3列の行列 (射影変換行列 [47]) であり、 \simeq は定数倍の不定性を許して等しいものとする。

道路等の走行可能な領域は、空間中でほぼ平面であると見なせるため、一方の画像に対して適当な射影変換を施すことにより、平面の部分に対しては一致するような画像が得られる。その一致した部分を判定することによって、道路領域を画像から抽出することができる。以下では、その処理内容について簡単に述べる。

2.1.1 道路抽出処理の流れ

図 2.2 は、道路抽出処理の全体図である。各々の部分の処理を以下で簡単に説明する。2台の車載カメラで撮影された画像をそれぞれ基準画像と参照画像とし、それらを入力画像とする。以降では、左側を参照画像、右側を基準画像とする。

1. 歪み除去

撮影された画像はレンズの歪みによって、透視投影モデルとは異なってしまっている。特に、レンズの歪みの影響を大きく受ける画像の端では、たとえ同一平面上に存在する点であっても、平面の射影変換行列を使って、正しく変換されない可能性がある。それに対処するため、レンズの歪みの影響を取り除く。

2. LOG フィルタ

式 (2.1) に基づいて、ステレオ画像同士をレジストレーションする際に、ステレオ画像間の明るさが異なるとうまく処理が行えない。そこで、画像間の明るさの差を取

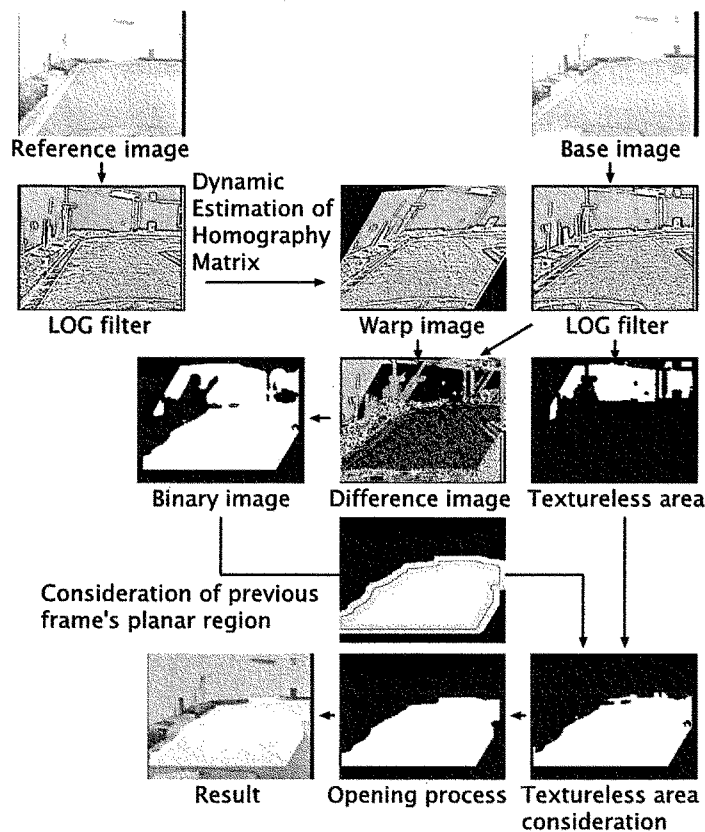


図 2.2: 道路抽出処理の流れ

Fig. 2.2: Flow chart of the road-region extraction method.

り除くために、入力画像に LOG (Laplacian Of Gaussian) フィルタを施す。LOG フィルタは、以下の式で表される。

$$\nabla^2 G(x, y) = \frac{-1}{\pi\sigma^4} \left(1 - \frac{r^2}{2\sigma^2}\right) e^{-\frac{r^2}{2\sigma^2}}. \quad (2.2)$$

また、LOG フィルタを施した画像は、コントラストが低いため、ヒストグラム平坦化によって、コントラストを上げる処理を行う。

3. 射影変換行列の推定

ステレオ画像間の道路面に対する射影変換行列を、フレームごとに動的に推定する。走行中の車両の振動などによって、カメラと道路面との相対的な位置関係が変化する場合においても、動的に射影変換行列を推定することで、安定に道路領域を抽出することができる。

射影変換行列の推定には、大きく分けて2つの方法がある。1つめは、射影変換行列をそのまま推定する方法である。この方法では、カメラの内部パラメータや外部

パラメータに関する事前情報を利用しないで、画像間の射影変換行列を推定することができる。ただし、道路面とカメラ間の相対的な姿勢を推定する場合には、2.2節で後述するようにそれぞれのカメラの内部パラメータが必要であり、さらに平面までの距離を出すには、カメラ間のベースライン長が必要である。

もう1つは、カメラの内部パラメータとステレオカメラ間の幾何学的関係(外部パラメータ)が既知であれば、式(2.1)の射影変換行列に含まれる、道路面によって変化するパラメータを直接推定(2.2.2節で述べる)することで、射影変換行列を求めることができる。推定するパラメータが少ない分、前者より安定に推定ができる。

4. 射影変換画像

3により推定された道路面に関する射影変換行列によって参照画像を射影変形する。変換された参照画像は、道路面に関しては基準画像に一致し、それ以外の領域に関しては、歪んで投影される。

5. 差分画像

道路平面上に存在する点は参照画像から基準画像へ正確に投影されるので、基準画像との輝度値の差分は小さい。逆に、平面上にない点の輝度値の差分は大きくなる。そこで、閾値処理を施して差分の小さい領域を平面領域とする。このとき、ピクセルごとの差分ではノイズによる影響が大きいため、SAD (Sum of Absolute Difference) によるマッチングを施す。具体的には、差分画像に平滑化フィルタをかけることで実現している。その後、閾値を使って二値化する。

6. 前時刻道路領域の利用

ステレオ動画像において、前時刻と現時刻に撮影された画像には高い相関がある。処理結果においても相関が高いことを利用して、前時刻の処理結果を利用することで安定した処理を行う。ここでは、現時刻に抽出される道路領域と非道路領域の境界が前時刻から予想される境界に対してある一定幅に収まると考え、それを二値画像に反映する。

7. テクスチャレス領域

5の差分画像だけを利用すると、画像上でテクスチャの少ない領域、つまりテクスチャレス領域では濃度値の変化が少ないため、道路領域以外の異なる位置同士が重なりあっていたとしても道路領域として抽出されてしまう可能性がある。それに対処するため、テクスチャレス領域をそのまま道路領域からすべて取り除いてしまうと、道路領域内にテクスチャレス領域が存在する場合には、本来は道路領域であるのに対して、誤って非道路領域としてしまう可能性もある。

そこで、まず、テクスチャレス領域を決定する。テクスチャレス領域は、画像に LOG フィルタを施すことで、空間周波数が低い場所ではその輝度値が低くなることから、輝度値に基づいてラベリングをすることで求める。

ラベリングしたテクスチャレス領域ごとに基準画像と参照画像のマッチング結果に完全に収まっている領域は道路領域とし、ずれが生じている領域は非道路領域とする処理を行なう。

8. Opening 処理

求めたい道路領域は、自動車の走行可能な道路領域である。そこで、自動車の走行不可能な小さな領域を道路領域抽出結果から取り除く。得られた道路領域抽出結果からその面積を求め、閾値以下の領域を除去し、道路領域全体に膨張収縮処理を施す。

2.2 道路面の姿勢推定

本節では、道路平面とカメラ間の相対的な姿勢を表すパラメータである、道路平面とカメラの光学中心間の距離 d と道路平面の法線ベクトル \mathbf{n} を推定する方法について述べる。

道路面の姿勢を求めるには、大きくわけて2つの方法がある。1つめは、カメラ間の外部パラメータは未知で、ステレオカメラ間のベースライン長(並進ベクトルの大きさ)だけが既知な場合である。2つめは、カメラ間の外部パラメータである、回転行列と並進ベクトルが既知な場合である。また、どちらの場合もカメラの内部パラメータは既知であることを前提にしている。次節において、それぞれの方法について述べる。

2.2.1 ステレオカメラ間のベースライン長のみが既知の場合

Faugeras らの手法 [10] によって、射影変換行列 \mathbf{H} を特異値分解をすることによって平面の姿勢を推定する。この方法では、カメラ間の拘束がゆるく、平面までの距離かベースライン長が既知でないとどちらかの絶対的な大きさを求めることができない。しかし、事前に外部パラメータのキャリブレーションの必要が無かったり、不定性は残るものの非同期ステレオ画像も扱うことができる。

文献 [10] に述べられている方法を元に、射影変換行列から平面の姿勢を表すパラメータである平面までの距離 d と平面の法線ベクトル \mathbf{n} を求める方法について簡単に説明する。まず、射影変換行列には以下のような関係がある。

$$\mathbf{H} = \mathbf{A}_2 \cdot k \left(\mathbf{R} + \frac{t\mathbf{n}^T}{d} \right) \mathbf{A}_1^{-1} \quad (2.3)$$

ここで、 \mathbf{R} はカメラ間の回転行列、 t はカメラ間の並進ベクトル、 d は基準カメラの光学中心と平面の距離、 \mathbf{n} は平面の法線ベクトル、 $\mathbf{A}_1, \mathbf{A}_2$ はそれぞれ基準カメラと参照カ

カメラの内部パラメータ、定数項 $k \neq 0$ を付けるのは、画像から得られた射影変換行列には定数倍の自由度が存在することを表す。

カメラの内部パラメータ A_1, A_2 が既知であると、式 (2.4) のように射影変換行列 H' を特異値分解できる。

$$H' = A_2^{-1} H A_1 = k \left(R + \frac{t \mathbf{n}^T}{d} \right) = U \Sigma V^T \quad (2.4)$$

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3) \quad (\sigma_1 \geq \sigma_2 \geq \sigma_3 > 0)$$

また、 k は、固有値 σ_2 に一致することが文献 [10] で示されている。さて、もとの $\frac{t}{d}, \mathbf{n}$ は以下のような関係式から求めることができる。

$$\begin{cases} \frac{t}{d} = U \frac{t'}{d} \\ \mathbf{n} = V \mathbf{n}' \end{cases} \quad (2.5)$$

$\sigma_1 \neq \sigma_2 \neq \sigma_3$ のとき、

$$\mathbf{n}' = \begin{bmatrix} n'_1 \\ n'_2 \\ n'_3 \end{bmatrix} = \begin{bmatrix} \epsilon_1 \sqrt{\frac{\sigma_1^2 - \sigma_2^2}{\sigma_1^2 - \sigma_3^2}} \\ 0 \\ \epsilon_3 \sqrt{\frac{\sigma_2^2 - \sigma_3^2}{\sigma_1^2 - \sigma_3^2}} \end{bmatrix} \quad (\text{ただし, } \epsilon_{1,3} = \pm 1) \quad (2.6)$$

$$\frac{t'}{d} = \frac{\sigma_1 - \sigma_3}{\sigma_2} \begin{bmatrix} n'_1 \\ 0 \\ -n'_3 \end{bmatrix} \quad (2.7)$$

$\epsilon_{1,3}$ を決定するには、2台のカメラから平面が見えているという条件を加える。すなわち、 \mathbf{n} と基準カメラの同次座標 \mathbf{m}_b との内積が正になる。これで、解が2つに減る。次に、それぞれの \mathbf{n} について、求まった t と、実際のカメラ配置のおおまかな並進ベクトルである \hat{t} (ここでは基準カメラが参照カメラの右側にあるので、 $\hat{t} = (1, 0, 0)$) との内積を計算し、1に近いほうの t を \mathbf{n} の解とすればよい。

d を求めるには、得られた $\frac{t}{d}$ の絶対値とカメラ間のベースライン長 $|t|$ から次式により、求めることができる。

$$d = \frac{|t|}{\left| \frac{t}{d} \right|} \quad (2.8)$$

2.2.2 外部パラメータが既知の場合

あらかじめ、カメラ間の外部パラメータである、カメラ間の回転行列と並進ベクトルをキャリブレーションをして求めておくことで、本節の手法が利用できる。本手法では、撮影中にカメラ間の関係は変化しないことを仮定しており、同期ステレオカメラを用いる際には妥当な設定である。さらに、本手法は2つの手法があり、それぞれで推定方法が異なる。1つめは、道路面に対する射影変換行列を推定する際に道路面の姿勢を含めて推定する方法(3パラメータ推定による方法)である。2つめは、射影変換行列を推定した後、外部パラメータを用いて代数的に分解し、道路面の姿勢を求める方法(代数的に分解する方法)である。それぞれの方法について以下で述べる。

3パラメータ推定による方法

勾配法によって道路面に対する射影変換行列を推定する際に、外部パラメータを含めた形で推定を行うことにより、道路面の姿勢パラメータである法線ベクトルと道路面までの距離を含めた \mathbf{n}/d を直接的に推定する。詳細については、杉本ら [73] の文献を参照されたい。

代数的に分解する方法

式(2.3)にカメラの内部パラメータ行列を両方から掛けることで、

$$\mathbf{H}' = \mathbf{A}_2^{-1} \mathbf{H} \mathbf{A}_1 = k \left(\mathbf{R} + \frac{\mathbf{t} \mathbf{n}^T}{d} \right) \quad (2.9)$$

を得る。 k は Faugeras らの手法 [10] により、 \mathbf{H} の2番目に大きい固有値とする。式(2.9)の両辺を k で割り、回転行列 \mathbf{R} を左辺に移すと、

$$\frac{1}{k} \mathbf{H}' - \mathbf{R} = \frac{\mathbf{t} \mathbf{n}^T}{d} \quad (2.10)$$

となる。次に、両辺に \mathbf{t}^T をかけて整理すると

$$\frac{\mathbf{n}^T}{d} = \frac{\mathbf{t}^T}{|\mathbf{t}|^2} \left(\frac{1}{k} \mathbf{H}' - \mathbf{R} \right) \quad (2.11)$$

となる。ここで、 $|\mathbf{n}| = 1$ であるので、式(2.11)の右辺のノルムが1であるベクトルが法線ベクトル \mathbf{n} である。最後に平面までの距離 d は、

$$\frac{1}{d} = \frac{\mathbf{n} \mathbf{t}^T}{|\mathbf{t}|^2} \left(\frac{1}{k} \mathbf{H}' - \mathbf{R} \right) \quad (2.12)$$

として与えられる。

2.3 本章のまとめ

本章では、空間中の平面とステレオ画像の幾何学的関係を利用して、画像中の平面領域を抽出する方法を説明した。道路等の走行可能な領域を平面とみなすことで、本手法を利用して、道路領域を抽出することができる。

次に、道路面とカメラ間の姿勢を推定する方法について述べた。この姿勢は、道路面に対するステレオ画像間の射影変換行列から求められる。その際、いくつかの推定方法があり、推定された射影変換行列を SVD 分解して求める方法や、射影変換行列を推定する際に、直接その姿勢を推定する方法などがあった。

第3章 仮想投影面画像の生成と危険領域検出

本章では、車載ステレオカメラを使った危険領域検出について述べる。危険領域とは、自車両にとって危険となる領域のことであり、例えば、対向車などのように自車両に近付いてくるものは危険性が高くなる。それらを検出することで、視覚誘導や安全走行のための運転支援などに利用できると考えられる。

提案手法では、自車両にとって危険であるのか否かを第2章で抽出された道路領域と非道路領域の境界の変化具合を時系列的に計測することで求める。例えば境界が近づいている領域は、自車両が障害物などの領域に相対的に近付いたということである。境界の変化を測定する際に、画像中で測定した場合には空間的に同じ変化をしたとしても境界が遠くにあった場合には、その変化が小さく観測され、近くでは大きく観測される。また、その境界変化の空間的な位置や大きさが容易には求まらない。そこで、提案手法では仮想的に道路面を垂直上方から見た画像 (VPP 画像) を生成することで、実空間における道路領域を求め、VPP 画像上での道路領域の境界の変化から、危険領域の検出を行なう。通常、危険領域までの距離を測定するためには、ステレオ画像間の対応を取る必要があるが、VPP 画像を生成したことで、ステレオ対応点探索を行うことなく、危険領域までの距離の測定が可能である。さらに、障害物の危険性を評価するには、従来手法では検出された障害物をトラッキングして、その動きを計測する必要があるが、提案手法ではカメラの光学中心から道路領域の方向毎の境界変化を計測するため、トラッキングなしに領域の危険性を評価することができる。

危険領域検出は、障害物検出の一種と考えられるため、関連研究については第4章の障害物検出において、まとめて述べることとする。本章では、まず、仮想投影面画像 (VPP 画像) の生成方法について述べる。次に、生成された VPP 画像を利用した、危険領域検出方法について述べる。最後に、実験結果を通じて、本手法の有効性を示す。

3.1 仮想投影面画像

本節では、画像を仮想的に道路面と平行に上方から観測した画像について述べる。この画像を本論文では、仮想投影面画像 (Virtual Projection Plane Image 略して VPP 画像) と呼

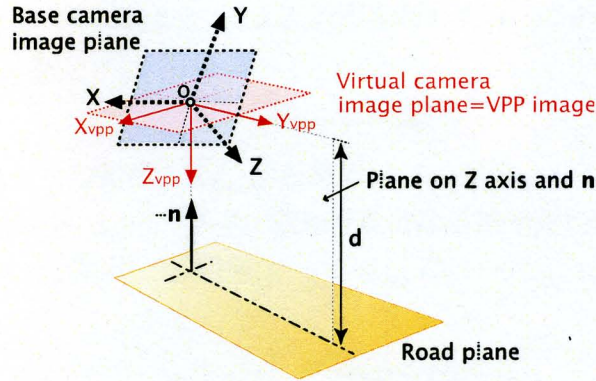


図 3.1: VPP 画像の幾何関係
Fig.3.1: VPP image geometry.

ぶこととする。VPP 画像は，図 3.1 に示す幾何関係にあり，以下に述べるような二つの性質を満たすものとする。

- VPP 画像の座標系は，実空間の座標系に一致する。そのため，VPP 画像上の一点は，3次元実空間中の一点に対応する。
- VPP 画像の縦軸は，基準カメラの光軸を道路平面に投影した方向と一致する。

さて，VPP 画像の生成には，道路平面の姿勢パラメータが必要である。本研究では，2.2 節で述べたように道路領域を抽出する際に，道路面に対する射影変換行列を推定しており，そこから姿勢パラメータを求めている。

3.1.1 VPP 画像生成

VPP 画像の生成方法について述べる。先に述べた VPP 画像の性質は，以下のように言い替えることもできる。

1. 仮想カメラの光軸 Z_{vpp} は，平面の法線ベクトル \mathbf{n} に一致する。
2. 仮想カメラの Y_{vpp} 軸は，法線ベクトル \mathbf{n} と基準カメラの光軸である Z 軸で張る平面上に存在する。

この2つの性質に基づいて，基準カメラから仮想カメラへの座標変換を考える。この座標変換は，それらの光学中心が変化しないことから，回転行列 \mathbf{R} を使って表現することができる。回転行列 \mathbf{R} を以下のように表現する。

$$\mathbf{R} = \begin{bmatrix} c\theta c\phi c\psi - s\phi s\psi & c\theta s\phi c\psi + c\phi s\psi & -s\theta c\psi \\ -c\theta c\phi s\psi - s\theta c\psi & -c\theta s\phi s\psi + c\phi c\psi & s\theta s\psi \\ s\theta c\psi & s\theta s\phi & c\theta \end{bmatrix} \quad (3.1)$$

ここで、 $c\phi$ や $s\phi$ はそれぞれ $\cos \phi$ と $\sin \phi$ を示すものとする。この回転行列を使い、座標変換を以下のようにする。

$$\mathbf{X}_r = \mathbf{R}_{vpp \rightarrow r} \mathbf{X}_{vpp}, \quad (3.2)$$

ここで、 \mathbf{X}_r は基準カメラの座標、 \mathbf{X}_{vpp} は仮想カメラの座標である3行1列のベクトルである。ここで、未知数は (θ, ϕ, ψ) であり、既知なものは道路平面の法線ベクトル \mathbf{n} である。

まず、先に述べたVPP画像の性質1によって Z_{vpp} 軸は、 $\mathbf{X}_{vpp} = [0, 0, 1]^T$ として、表現できる。よって、

$$\mathbf{n} = [n_x, n_y, n_z]^T = \mathbf{R}_{vpp \rightarrow r} [0, 0, 1]^T. \quad (3.3)$$

が得られる。さらに成分ごとに計算することによって、

$$\begin{cases} \theta = \arccos(n_z) \\ \psi = \arctan\left(\frac{-n_y}{n_x}\right) \end{cases} \quad (3.4)$$

が得られる。これによって、未知数である (θ, ψ) を求めることができる。

次に、VPP画像の性質2によって、 $Y_{vpp} = [0, 1, 0]^T$ 軸は、 $Z = [0, 0, 1]^T$ 軸と法線ベクトル $Z_{vpp} = \mathbf{n}$ の張る面上に存在することを利用することにより、

$$\mathbf{R}_{vpp \rightarrow r} [0, 1, 0]^T = a[0, 0, 1]^T + (1 - a)\mathbf{n}, \quad (3.5)$$

を得る。ここで、 a は任意定数である。上式を解くことで、

$$\phi = \pm\pi/2 \quad (3.6)$$

が得られる。 Y_{vpp} 軸と Z 軸は同じ方向であるため、 ϕ の符号はそれらの内積が正であればよい。つまり、

$$[0, 0, 1] \cdot \mathbf{R}_{vpp \rightarrow r} [0, 1, 0]^T > 0 \quad (3.7)$$

である。ここまでの説明で未知数がすべて得られたことから、回転行列 $\mathbf{R}_{vpp \rightarrow r}$ を求めることができる。

さて、基準画像からVPP画像への変換は射影変換行列 \mathbf{H}_{vpp} を使って、以下のように表現することができる。

$$\tilde{\mathbf{m}}_{vpp} \sim \mathbf{H}_{vpp} \tilde{\mathbf{m}}_r = \mathbf{A}_{vpp} \mathbf{R}_{vpp \rightarrow r}^{-1} \mathbf{A}_r^{-1} \tilde{\mathbf{m}}_r, \quad (3.8)$$

ここで、 $\tilde{\mathbf{m}}_{vpp}$ は、VPP画像の同次座標であり、 $\tilde{\mathbf{m}}_r$ は、基準画像の同次座標である。また、 \mathbf{A}_{vpp} は仮想カメラの内部パラメータである。

図3.2は、基準画像とその基準画像に対するVPP画像を生成した結果である。青領域は、抽出された道路平面領域である。右側のVPP画像では、道路白線が平行になっており、正しくVPP画像が生成されていることが分かる。

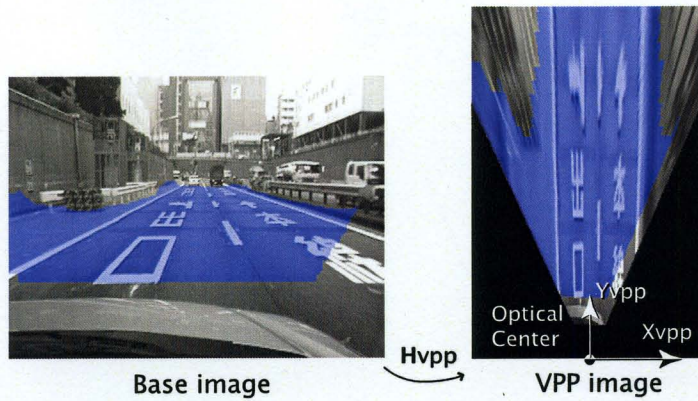


図 3.2: 道路平面領域抽出結果およびその VPP 画像生成例。(青い領域が抽出された道路平面領域)

Fig. 3.2: Composition of VPP image and road region. The blue areas in the images are the extracted road regions.

3.1.2 仮想カメラの内部パラメータと実空間距離

本節では、仮想カメラの内部パラメータと実空間距離の換算方法について述べる。仮想カメラの内部パラメータである A_{vpp} は、実空間に対する VPP 画像上のスケールや VPP 画像上でのカメラ位置などを自由に設計することができる。

仮想カメラの焦点距離を f ，有効画素間隔を δu として

$$\alpha = \frac{f[\text{m}]}{\delta u[\text{m}/\text{pix}]} = f/\delta u[\text{pix}] \quad (3.9)$$

とし、光学中心を VPP 画像座標上で (C_x, C_y) とすると、仮想カメラの内部パラメータは、

$$\mathbf{A}_{vpp} = \begin{bmatrix} \alpha & 0 & C_x \\ 0 & \alpha & C_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

と表すことができる。平面までの距離が $1[\text{m}]$ であるときは、実空間の平面において $1[\text{m}]$ の長さは VPP 画像中では、 $\alpha[\text{pixel}]$ である。平面までの距離を $d[\text{m}]$ とすると、図 3.3 のように相似関係から、 $\alpha/d[\text{pixel}]$ となる。

そこで、平面までの距離 d が走行中に変化したとしても、VPP 画像上でスケールを一定に保つためには、仮想カメラの内部パラメータを以下のようにすればよい。

$$\mathbf{A}_{vpp} = \begin{bmatrix} \alpha \times d & 0 & C_x \\ 0 & \alpha \times d & C_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

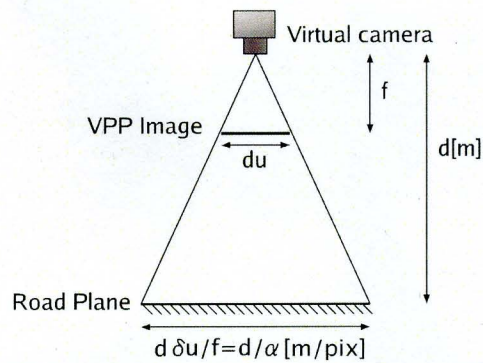


図 3.3: VPP 画像のスケールと実空間距離
Fig. 3.3: Scale on VPP image and real distance .

式(3.11)に基づいて仮想カメラの内部パラメータを設計することで，平面までの距離に依存することなく，実空間の平面上での 1[m] の長さは，VPP 画像中で常に α [pixel] 分となる。

3.2 危険領域検出

第 2 章で述べた方法により道路領域が抽出され，その過程で推定される射影変換行列から道路面とカメラ間の相対的な姿勢が求められ，それらを利用することで道路平面領域の VPP 画像を求めることができる。危険領域検出では，各時刻で生成される道路領域の VPP 画像を用いる。

VPP 画像上でカメラの光学中心を道路面に投影した位置から非平面領域までの距離並びに時系列画像を利用して相対速度を求めることで危険領域検出を行う。

3.2.1 方向別距離測定

道路領域の方向別距離を求めるには，図 3.4 のように道路領域を VPP 画像に変換した画像を用意する。同図の道路領域は図 3.2 の実際の抽出結果である。

次に，基準カメラの光学中心を VPP 画像に投影した点を VPP 画像の光学中心とすると，この光学中心から VPP 画像縦軸 (基準カメラの光軸を道路面へ投影した方向) に対して θ だけ傾いた直線を伸ばし，平面領域の端を計算する。この端が方向別の道路面の拡がりの端点である。また，VPP 画像上での長さは 3.1.2 節で述べたように，仮想カメラの内部パラメータと基準カメラの光学中心から平面までの距離を使うことで実際の距離に変換できる。これによって，光学中心から方向別の平面の端点までの距離，すなわち方向別距離が算出できる。

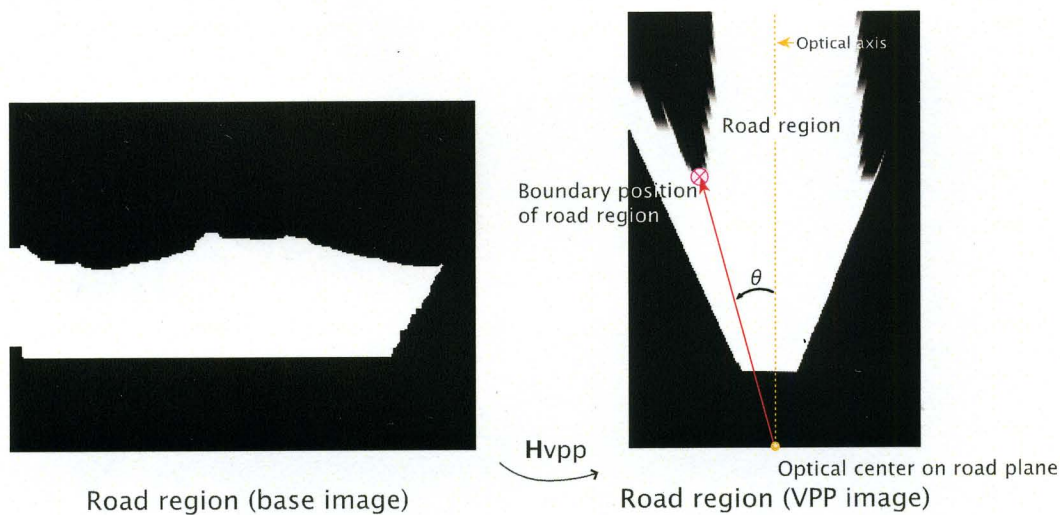


図 3.4: 道路領域の境界までの方向別距離測定
 Fig. 3.4: Distance of road region boundary for every direction.

3.2.2 方向別相対速度測定

次に道路領域の方向別相対速度を求める。画像は 1 フレーム 1/30[sec] の等間隔で撮影される。そこで、1 フレームごとの方向別距離を時系列にわたってフィルタリングすることで、その速度を算出する。この速度は基準カメラの光軸を道路面へ投影した方向からの各方向ごとの道路領域の拡大収縮速度であり、方向別相対速度と呼ぶことにする。

方向別相対速度がマイナスの場合には、道路領域が削減されていることに相当するため、その方向には対向車などの危険性の高い障害物が存在している可能性がある。何も変化がない領域では、方向別相対速度が 0 である。

3.3 実験結果

本節では、付録 A で述べた実験装置で車載カメラのステレオ動画像を取得し、実験に利用した。

3.3.1 原画像と VPP 画像に情報表示

図 3.5 は出力結果の例である。同図の左側は、推定された各種情報を基準画像中表示している。A は方向別の道路領域の境界を表しており、その色は方向別速度に応じて、C に示す色に従う。速度が 0 の場合には、緑色であり、危険性が高い (近づく方向) 場合には、暖色系の色となり、危険性の低い (遠ざかる方向) 場合には、寒色系の色となる。同図の実験結果では、左側のガードレールによって道路の狭まっている領域で危険性が高く表

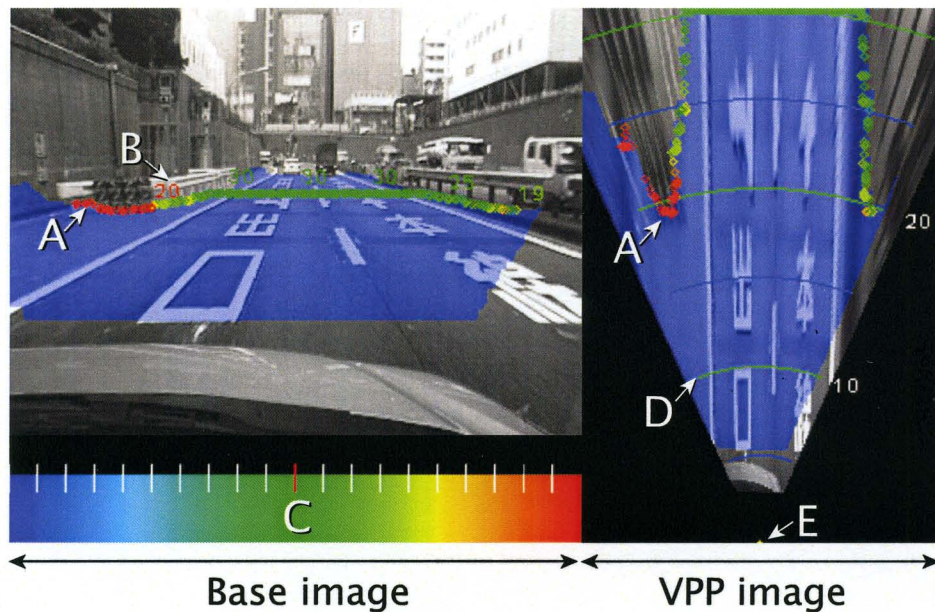


図 3.5: 危険領域検出結果の出力例
 Fig. 3.5: Output example of dangerous region detection.

示されている。Bは、一定角度ごとの方向別距離をメートル単位で表示している。また、車載カメラの解像度の関係で、VPP 画像は前方 30[m] まで生成したため、表示される範囲もその範囲に限られる。次に同図右側は、推定された情報を VPP 画像に表示している。Dは、基準カメラの光学中心を道路面に投影した位置 E からの距離を 5[m] ごとに描いたものである。

3.3.2 危険領域検出結果

図 3.6 は、市街地を走行するシーンの実験結果を 10 フレームおきに表示している。このシーンでは、黒い対向車が右側を通りすぎる。同図の一番上の結果では、ガードレールによって道路領域が縮むため、ガードレールと道路の接している場所で、危険性が高いことを示す赤色に表示されている。次に、対向車が近付いて来るにつれ、それによって道路領域が削られていくため、対向車の前方で方向別速度が赤く表示されている。

図 3.7 も市街地を走行するシーンの実験結果を 10 フレームおきに表示している。対向車線をバイクが走行して来るため、道路領域が削られ、危険性が高い赤色となっている。また、同様にガードレールなども赤色となっている。

図 3.8 と図 3.9 は、一般道路を走行するシーンである。対向車両の部分が赤く、危険性が高く推定されていることがわかる。また、前方は特に道路領域の変化がないため、方向別速度変化のないことを示す緑色となっている。

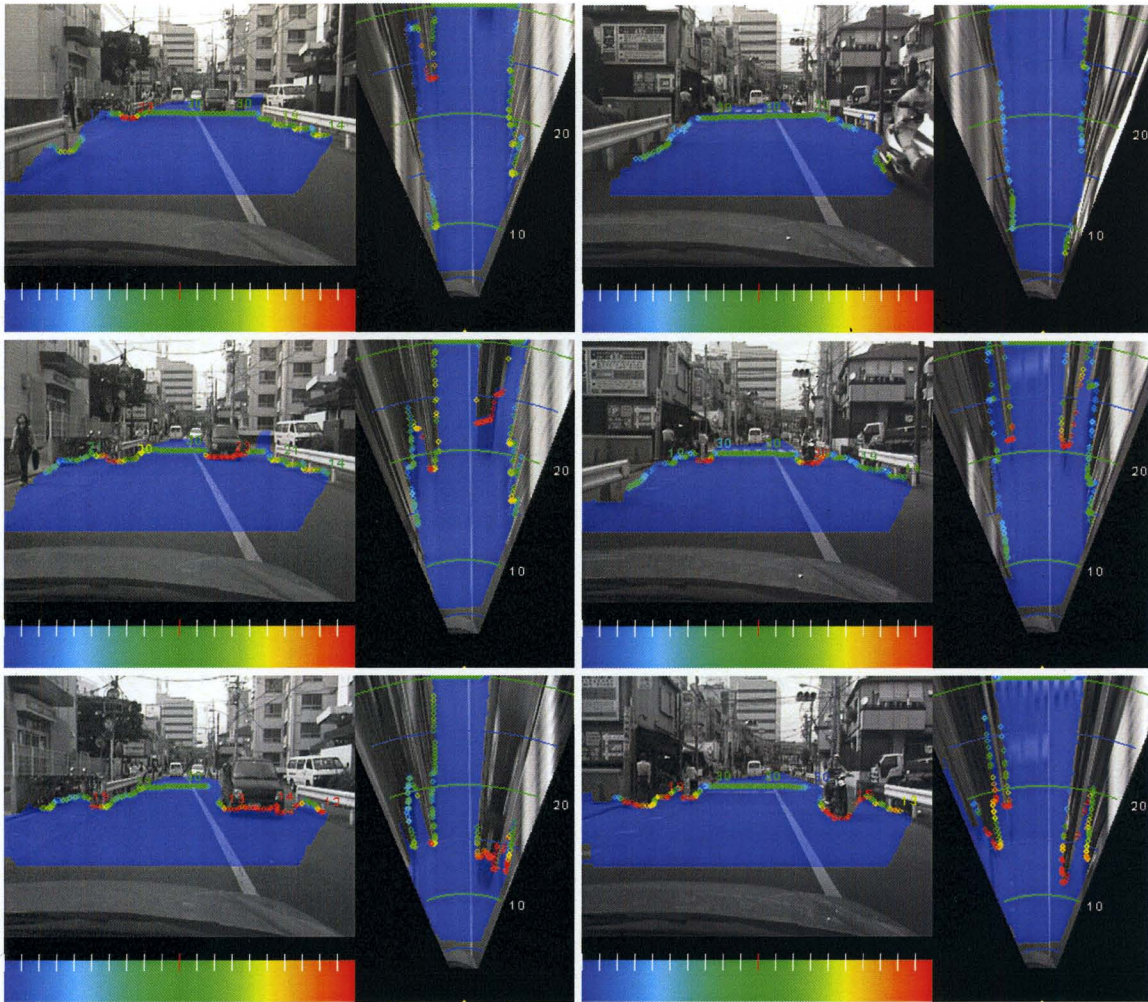


図 3.6: 市街地における実験結果 1(10 フレームごとに表示)

Fig. 3.6: Dangerous region detection results in urban environment.

図 3.7: 市街地における実験結果 2(10 フレームごとに表示)

Fig. 3.7: Dangerous region detection results in urban environment.

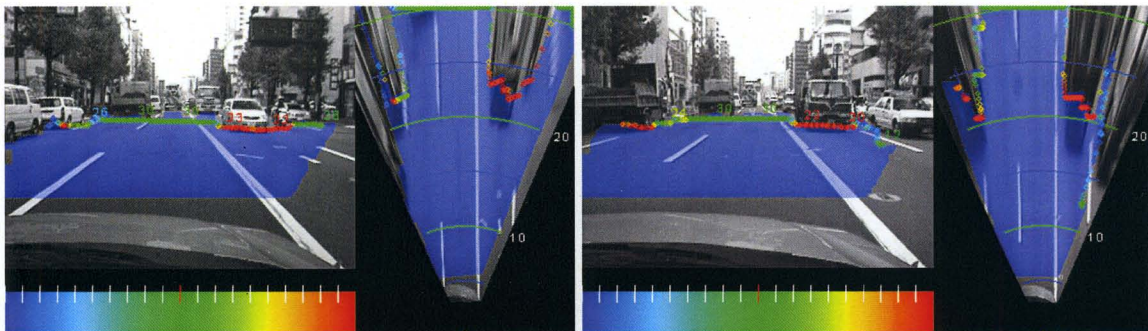


図 3.8: 一般道路における危険領域検出結果 1

Fig. 3.8: Dangerous region detection results in general road.

図 3.9: 一般道路における危険領域検出結果 2

Fig. 3.9: Dangerous region detection results in general road.

3.4 本章のまとめ

本章では、ステレオ動画像のみを用いて危険領域を検出する手法について述べた。まず、ステレオ画像から道路平面領域を求め、その過程で推定される射影変換行列から道路面とカメラ間の相対的な姿勢を用いて、VPP 画像を生成した。危険領域は、抽出された道路領域を VPP 画像に変換し、VPP 画像上での道路領域の境界を方向別に距離を計測して、時系列間でその変化を測定することによって判断した。そのため、トラッキング作業などを行わずに危険性の判定が可能である。また、VPP 画像は道路面を垂直情報から観測した画像に相当するため、実座標に対応しており、実空間中での位置や距離をステレオ対応点探索なしに求めることが可能である。

本手法は、簡単な処理で危険領域の検出ができ、実験結果から対向車などの大きな物体以外にも、一般的に検出が難しいと考えられるガードレールの先端などの細かい部分も、危険な領域として検出できることが確認できた。

第4章 道路面を基準とした障害物検出

第3章で述べた方法では、障害物を陽に検出するのではなく、自車両に近付いてくる領域を簡易な方法で検出することで危険領域検出を行なった。本章では、障害物を陽に検出する手法を提案する。

まず、本章ではカメラを利用した障害物検出手法の関連研究について述べ、提案手法との違いを明確にするとともに、第3章で述べた手法の問題点を明かにし、提案手法によってそれが解決されたことを示す。次に、道路領域と道路面の姿勢を有効利用し、かつ道路障害物の性質を考慮した処理をすることによって、障害物を検出する提案手法について述べる。提案手法をさまざまな天候・道路環境下に適用し、その実験結果を示す。また、他手法との比較実験も示す。最後に、本章のまとめを述べる。

4.1 関連研究

単眼カメラを用いた障害物検出手法の従来手法として、画像内の2次元な動きをオプティカルフローなどを用いて推定し、道路面などの背景と動きの異なる領域を障害物領域として検出する方法 [87, 29, 85] が提案されている。しかし、小さい領域の動きを正確に求めることは難しいため、小さい物体を検出するのは困難であると考えられる。また、画像中の拡張焦点 (FOE) 付近では検出感度が落ちるという問題がある。他の方法として、道路平面上にある水平エッジと障害物上にある水平エッジの複比を比較することによって障害物を検出する方法 [36] が提案されている。これらの単眼による方法では、対象やカメラ配置などに何らかの拘束を設けないと障害物までの距離を出すことができない。

これらの問題は、複眼カメラを用いることで解決できる。複眼カメラを用いる方法として、第3章では、道路面を平面と仮定して道路平面領域の時間変化によって自車両に迫ってくる障害物が存在する領域を推定する手法を提案している。他には、ステレオ画像をそれぞれ平面に投影した2枚の画像間の差から障害物を求める方法 [25, 5, 7] が提案されている。これらの方法では3次元再構成を行わないため計算は速いが、図4.1に示すように障害物が平面と接しておらず、障害物の下の平面が見える場合には対象までの距離を正確に求めることができない。他にステレオ画像間の対応点探索を行うことで3次元再構成をし、障害物を求める手法 [27, 88, 48, 32, 20, 26, 81, 82] が提案されている。例えば Labayradeら [27] は、ステレオ計測された点群の3次元情報から道路面を推定し、車両の

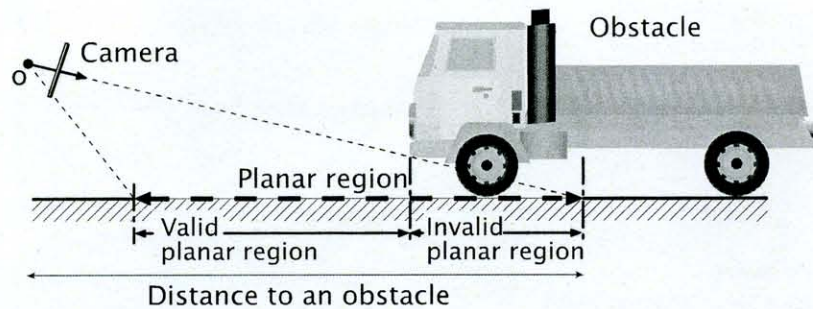


図 4.1: 車載カメラから見た道路面の見え方.
 Fig. 4.1: A view of planar region from vehicle-mounted cameras.

後面などをカメラに対して平行な平面と想定することで、障害物を検出する手法を提案している。さらに Hu ら [20] は、Labayrade らの枠組を拡張して道路構造物なども検出している。しかし、どちらも障害物を平面としてモデル化しているため、曲線物体 (例えば、カーブしているガードレール) を正確に検出することができない。また、窪田ら [26] は道路に垂直な方向に短冊状のウィンドウを設けて、ステレオ画像間の対応点探索を行ない、Dynamic Programming を使って物体を検出するため、曲線物体もそのままの形で検出可能である。しかし、短冊状のウィンドウでのマッチングは、建物などの垂直に長い物体には有効である [8] が、車高の低い自動車や踏切のバーなどのようにウィンドウ内における対象物体の占有率が低いと、誤った位置で対応づけられてしまい、正しく検出されない可能性がある。

さて、本論文では環境中に存在する数々の物体のうち、図 4.2 に示すように自車両にとって障害となる可能性のある立体体を障害物とする。例えば、標識や立体横断歩道のように道路上方の物体は自車両より高い位置に存在しており、自車両と衝突しない。また、路面上に描かれている道路標識も同様に自車両にとって安全である。道路面は自車両が走行する 2 次元平面に相当するため、道路面からの物体高さを元にすれば、障害物か否かを判定することができる。そのため、カメラと道路面の相対的な姿勢を推定することが重要である。

提案手法では、道路面に対するステレオ画像間の射影変換行列を動的に推定することで道路領域を抽出し、道路面の姿勢を得る。この処理では道路面に対する大きなウィンドウを変形してマッチングをするため、テクスチャの少ない路面でも安定に推定することができる。さらに、抽出された道路領域以外の領域に関して、ステレオ計測をすることで計算量を抑えつつ障害物を検出する。ステレオ計測を行なうことで、障害物が道路面と接していなくても距離を正しく求めることができる。さらに計測結果から障害物を抽出する際、特定の対象に限定したモデル化を行なわないため、さまざまな物体を元の形状に近い形で検出することができる。本論文では、道路環境の拘束を効果的にそれぞれの処理に取り入

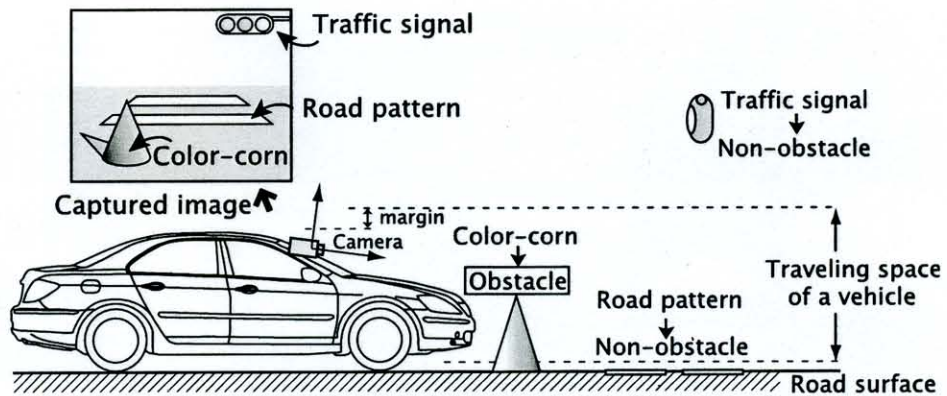


図 4.2: 障害物: 車両の通過する可能性のある空間に存在する物体
 Fig. 4.2: Obstacle: Obstacle exists between certain heights from the road surface.

れることで、画像の入力から障害物検出までの一貫したシステムを構築し、さまざまな環境下において、安定かつ頑健に障害物が検出されることを確認した。

4.2 処理の全体図

提案手法は、図 4.3 に示すように第 2 章で述べた道路面抽出や姿勢推定などの複数の処理を組み合わせて障害物検出を行なう。本節では、それぞれの処理を同図の流れに従い説明する。

まず、車載ステレオカメラで撮影された画像を入力し、道路領域を抽出する (図 4.3 ①)。前述した道路領域抽出アルゴリズムと姿勢推定アルゴリズムによって、それぞれを推定する。

次に、障害物の空間位置を求める (図 4.3 ③)。まず、ステレオ画像に対して Rectification (付録 B) を行ない、ステレオ画像の平行化を行なう。次に、障害物は画像中の平面領域以外に存在するため、平面領域以外のエッジ点に関して、ステレオ計測を行なう。抽出された平面領域を活用することでエッジ点の数を減らしたり、対応点探索時の視差探索範囲を限定したりすることで計算コストを抑える。

さて、ステレオ計測をして得られる点の 3 次元位置は、カメラを基準とする座標系である。カメラと路面の関係が分からないと、その点が障害物に属するものなのか、それ以外に属するものなのかを区別することが困難である。また、先にも述べたように走行中、道路面とカメラの位置関係は変化するため、その関係を固定することはできない。そこで、先ほど求めた姿勢パラメータを用いることで、ステレオ計測結果を道路面を基準とした座標系 (道路座標系) に変換する。道路座標系では道路面に属する点の高さは 0 となるため、容易に路面パターンであると判断することができる。また、自車両が通過する可能性のある、道路面から一定の高さを持つ空間も容易に特定することができる。

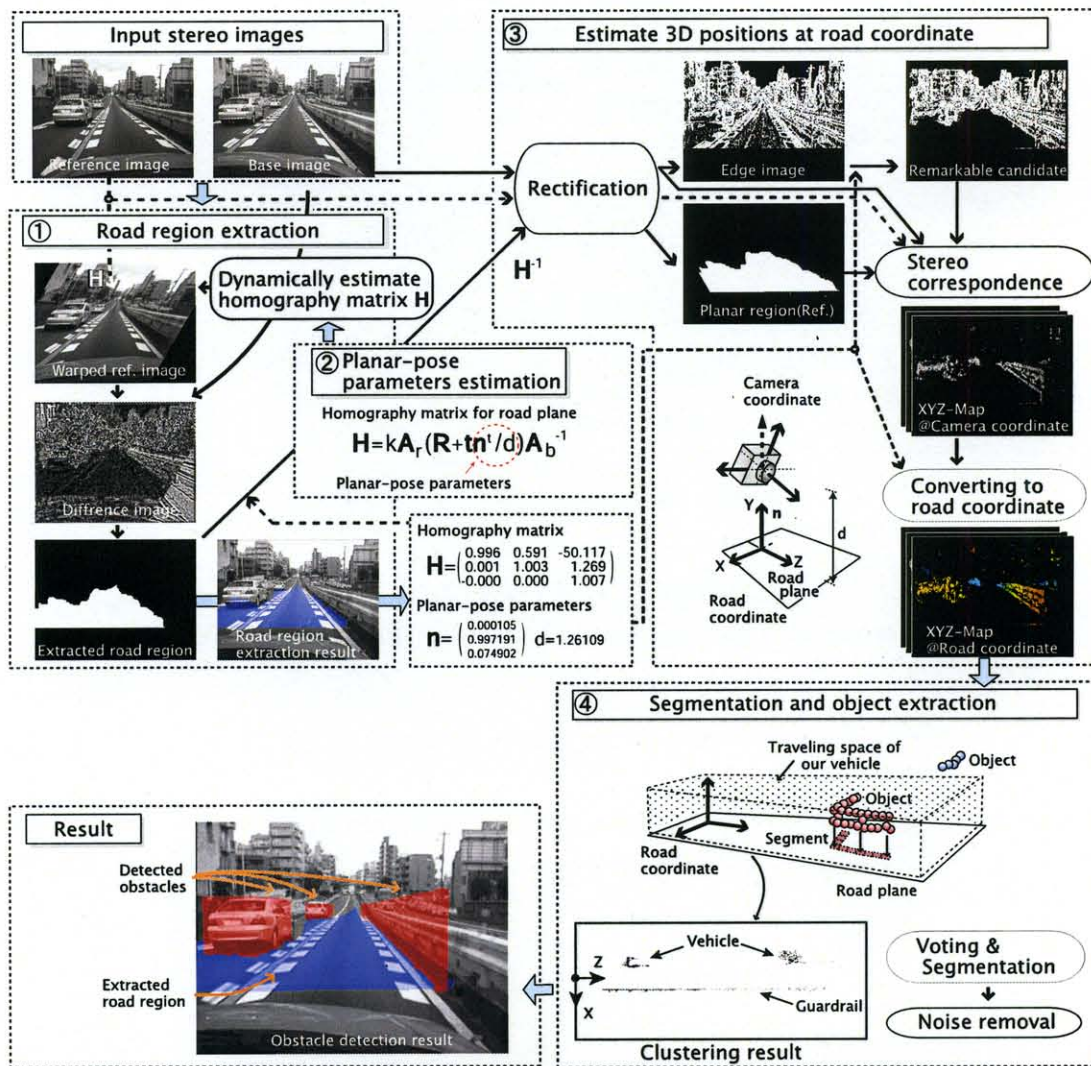


図 4.3: 障害物検出手法の全体の概要
 Fig. 4.3: Overview of our proposed obstacle detection method.

次に、その道路座標系に変換された点の位置を利用して障害物を抽出する(図 4.3 ④)。道路環境中に存在する自動車、歩行者やガードレールなどの障害物は、道路面に垂直であったり、水平方向に幅を持っている。そこで、自車両の通過する空間に含まれる点をそれぞれ 1 票として、道路面に対して高さ方向に垂直に投影し、道路面を投票グリッドとして投票処理を行なう。さらに投票グリッドを用いて、セグメンテーション処理とノイズ除去をし、障害物を検出する。結果として検出される障害物は、自車両が通過する可能性のある空間に含まれる物体に限られる。

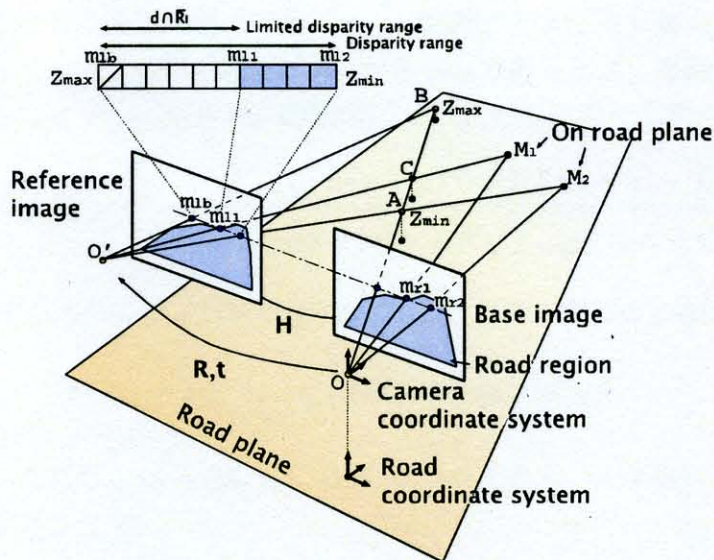


図 4.4: 対応点探索と平面領域の関係
Fig. 4.4: Stereo correspondence and homography matrix of road region.

4.3 ステレオ計測と障害物の検出

本節では抽出された道路領域と射影変換行列から推定された道路面の姿勢パラメータを利用して、障害物を検出する方法を図 4.3 中の③の流れに従って述べる。

4.3.1 道路領域と姿勢を利用した対応点探索の限定

物体の空間位置を計測するには、基準画像の注目点に対して、実空間の奥行方向に設定された範囲内で参照画像のエピポーラ線を探査し、注目点と最も類似性の高い位置を対応位置として求める。そこで対応点探索をする注目点の候補 (以下、注目点候補) の決定方法についてまず述べる。自動車のボディのようにテクスチャレスな部分は、対応点を直接求めることができないため、注目点候補はエッジ部分のみにする。さらに前処理によって求められた道路領域には障害物が存在しないことから、道路領域に含まれるエッジ点を注目点候補から除く。また、道路面の姿勢パラメータを使うことで、空間中の検出範囲が画像に射影される範囲を求められるため、範囲外は注目点候補から除く。最終的に残った注目点候補を図 4.3 の③中の'Remarkable Candidate'に示す。

次に、ステレオ画像間の対応を求める処理について説明する。本研究では、図 4.4 に示すように、道路平面上の点は射影変換行列で対応がついているため、その性質を考慮した対応点探索を行なう。同図で、基準カメラの光学中心 O から AB 方向への光線を考える。 A は奥行き Z_{min} の位置、 B は Z_{max} の位置とする。 A が参照画像に投影される点は m_{1r} であり、道路領域に含まれるため、基準画像の m_{1b} と対応がついている。すると、参照画像

のエピポーラ線上で道路領域の端点である m_{l1} , 空間中では C より遠方が対応点の探索候補である。そのため, 通常の探索範囲が m_{lb} から m_{l2} であるとき, 道路領域とその射影変換行列を利用することで m_{lb} から m_{l1} の範囲において正規化相互相関 (ZNCC) である,

$$\text{ZNCC}[I_1(\mathbf{u}), I_2(\mathbf{v})] = \frac{\sum_{\mathbf{x} \in \mathbf{R}} (I_1(\mathbf{x} + \mathbf{u}) - \bar{I}_1)(I_2(\mathbf{x} + \mathbf{v}) - \bar{I}_2)}{\sqrt{\sum_{\mathbf{x} \in \mathbf{R}} (I_1(\mathbf{x} + \mathbf{u}) - \bar{I}_1)^2} \sqrt{\sum_{\mathbf{x} \in \mathbf{R}} (I_2(\mathbf{x} + \mathbf{v}) - \bar{I}_2)^2}} \quad (4.1)$$

ここで, \mathbf{R} は注目点周りのウィンドウ領域である。上式が最大になる位置 $\hat{\mathbf{x}}_r$ を

$$E(\hat{\mathbf{x}}_r) = \arg \max_{\mathbf{x}_r \in (d \cap \mathbf{R}_l)} \text{ZNCC}[I_b(\mathbf{x}_b), I_r(\mathbf{x}_r)] \quad (4.2)$$

として推定する。ただし, $\mathbf{x}_b = (u_b, v_b)$, $\mathbf{d} = (d_r, v_b)$ であり, \mathbf{d} は奥行き推定範囲から視差の範囲を限定することで,

$$d_r = f(Z, \mathbf{x}_b) = \frac{P'_{11}X + P'_{12}Y + P'_{13}Z + P'_{14}}{P'_{31}X + P'_{32}Y + P'_{33}Z + P'_{34}} \quad (4.3)$$

である。ただし, d_r は $Z \in [Z_{\min}, Z_{\max}]$ の範囲であり, X, Y は以下の式で求められる。

$$\begin{aligned} [X \ Y]^T &= \mathbf{A}^{-1} \mathbf{b} \\ \mathbf{A} &= \begin{bmatrix} P_{11} - P_{13}u_b & P_{12} - P_{32}u_b \\ P_{21} - P_{13}v_b & P_{22} - P_{32}v_b \end{bmatrix} \\ \mathbf{b} &= \begin{bmatrix} (-P_{13} + P_{33}u_b)Z - P_{14} + P_{34}u_b \\ (-P_{23} + P_{33}v_b)Z - P_{24} + P_{34}v_b \end{bmatrix} \end{aligned} \quad (4.4)$$

式中の P_{ij} は Rectification 後の基準カメラの透視投影行列の i 行 j 列の要素である。同様に P'_{ij} は参照カメラについてである。また, \mathbf{R}_r は参照画像における道路領域であり, 基準画像の道路領域を射影変換行列 \mathbf{H}^{-1} で変換して求める¹。非道路領域のみ探索を行なうことで, 計算コストを抑えることができる²。さらに, マッチングミスを抑えるために Cross checking[11]を利用した。Cross checkは, 基準画像の注目点を参照画像の点へ対応づけた後, 対応づけられた参照画像の点を基準画像へ対応づけを行ない, 基準画像で対応づけられた点と注目点とが一致していれば, ステレオ画像間の対応が成功したと判断する方法である。

4.3.2 カメラ座標系から道路座標系への変換

三角測量の原理からステレオ画像間の対応を求めることで, カメラ座標系における3次元位置を求めることができる [17]。カメラ座標系では, 路面パターンや道路上方の立

¹抽出された道路領域は基準画像に対するものであるため。

²計算コストの比較については 4.5.3 節で示す。

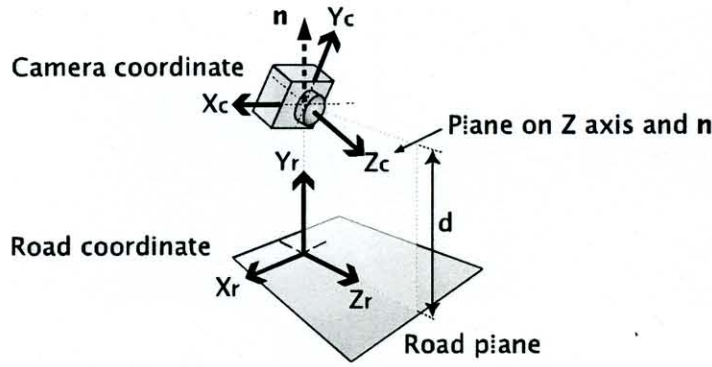


図 4.5: カメラ座標系と道路座標系
Fig. 4.5: Camera coordinate system and road coordinate system.

体歩道などの非障害物を区別することが困難であるため、道路面を基準とした座標系である道路座標系に変換する。それは、道路面に対する射影変換行列 \mathbf{H} によって推定された平面とカメラの距離 d と基準カメラに対する平面の法線ベクトル \mathbf{n} を利用することで、図 4.5 に示すように、カメラ座標系から道路座標系へ座標変換する。この座標変換は、カメラの光軸を道路面に投影したときの方向が座標変換後も変わらず、さらに道路面上は $Y = 0$ となるように決める。カメラ座標系上の点 $\mathbf{X}_c = [X_c, Y_c, Z_c]^T$ から道路座標系上の点 $\mathbf{X}_r = [X_r, Y_r, Z_r]^T$ に変換するには、以下の式を利用する。

$$\mathbf{X}_r = \mathbf{R}_{rc}^{-1}(\mathbf{n})\mathbf{X}_c + \mathbf{t}_{rc}(d) \quad (4.5)$$

ここで、 $\mathbf{R}_{rc}^{-1}(\mathbf{n})$ は、カメラ座標系から道路座標系への回転行列であり、 $\mathbf{t}_{rc}(d)$ は道路座標系から見たカメラの並進ベクトルである。算出方法を以下に述べる。

座標変換行列算出

回転行列 \mathbf{R}_{rc} を一般的な回転行列と考えて、

$$\mathbf{R}_{rc}(\mathbf{n}) = \begin{bmatrix} c\alpha c\gamma - s\alpha s\beta s\gamma & -c\beta s\alpha & c\gamma s\alpha s\beta + c\alpha s\gamma \\ c\gamma s\alpha + c\alpha s\beta s\gamma & c\alpha c\beta & -c\alpha c\gamma s\beta + s\alpha s\gamma \\ -c\beta s\gamma & s\beta & c\beta c\gamma \end{bmatrix} \quad (4.6)$$

とする。ただし、 $c\alpha = \cos \alpha$ 、 $s\alpha = \sin \alpha$ を示し、他も同様である。

カメラ座標系と道路座標系の座標変換は、カメラの光軸を道路面に投影したときの方向が座標変換後も変わらず、さらに道路面上は $Y = 0$ となるように決める。

そこで、まず、 $\mathbf{X}_c = \mathbf{R}_{rc}\mathbf{X}_r$ を考える。 \mathbf{n} と Y_r 軸が一致していることと、 $\mathbf{n} = [n_x, n_y, n_z]^T$ と Z_c 軸で張る平面上に Z_r が存在することから α, β, γ を求める。一つめの条件より、

$$\mathbf{R}_{rc}\mathbf{e}_2 = \mathbf{n} \quad (4.7)$$

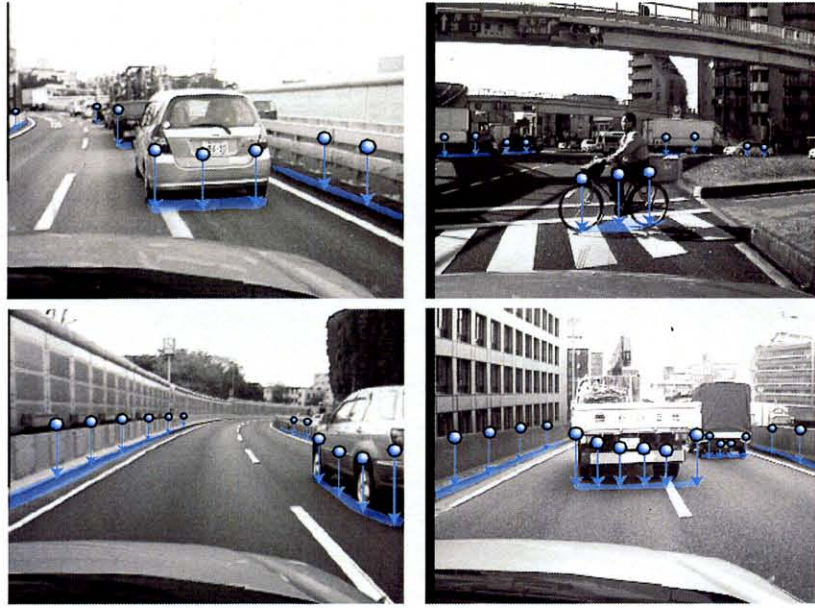


図 4.6: 道路障害物の特徴
Fig. 4.6: Features of road obstacles.

から, $\beta = \arcsin(n_z)$, $\alpha = \arccos(n_y / \cos \beta)$ を得る. 次に, Z'_r は \mathbf{n} と Z_c 軸で張る平面上に存在するので,

$$\mathbf{R}_{rc} \mathbf{e}_3 = a \mathbf{e}_3 + b \mathbf{n}. \quad (4.8)$$

a, b は任意の定数であるとして, γ を求めると $\gamma = 0$ となる.

この座標変換によって, カメラと平面の位置関係が変化しても, カメラ座標系の 3 次元位置を道路面を基準とした道路座標系に変換することができる.

4.3.3 道路面への投票を利用した障害物の検出

図 4.6 に示すように, 道路環境中に存在する自動車, 歩行者やガードレールなどの障害物は, 道路面に垂直であったり, 水平方向に幅を持っている. そこで, 道路座標系に変換した 3 次元計測結果を利用して障害物を抽出する. 障害物は自車両が通過する可能性のある空間に存在する物体であるため, その空間に相当する, 道路面からの高さが T_{\max} から T_{\min} ³ の範囲に存在する 3 次元計測結果をそれぞれ 1 票として道路面に対して垂直に射影し, 投票グリッドに投票する. 同じ奥行きと幅に計測結果が多く存在するほど, そのグリッドに多くの投票数を得る. その様子を図 4.7 に示す. 同図の右側は, 図 4.3 の入力画像を使用した際の投票した結果とその領域に相当する仮想投影面 (VPP) 画像を示している. 同図では, 右側のガードレールと左車線の前走車や対応点探索ミスによるノイズに

³本論文では, 実験車両の大きさを考えて $T_{\max} = 2.0[\text{m}]$, $T_{\min} = 0.2[\text{m}]$ とした.

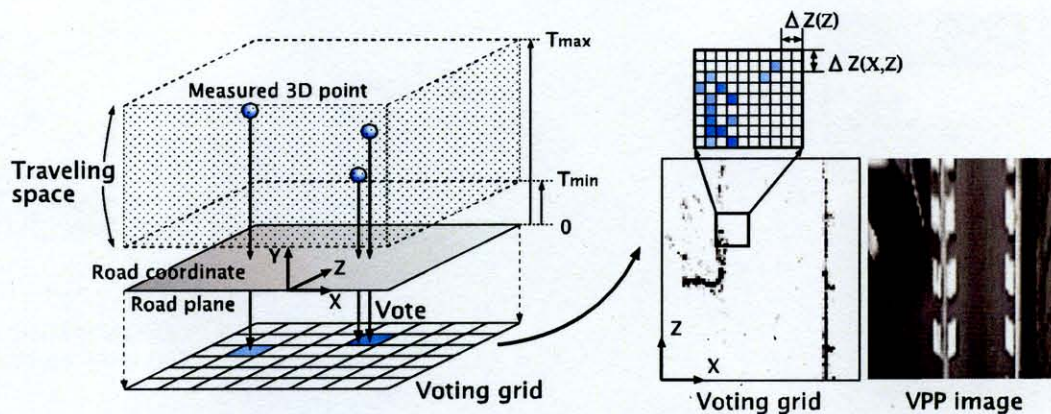


図 4.7: 車両の通過空間と投票処理 (左は投票処理, 右は投票結果)

Fig. 4.7: The traveling space of a vehicle and voting process. Left side is voting process and right side is VPP image.

投票値があることがわかる。道路環境に存在する構造物は高さ方向に大きさを持つことが多いため、高さ方向に射影することで点群が少なくても、後述するクラスタリング処理によって障害物を検出可能である。

次に、投票後の投票グリッドを利用してクラスタリングを行なう。クラスタリングは、空間中の幅と奥行きを特徴量とした最短距離法を用いる。ここで、ステレオ計測では遠方になるにつれて計測誤差が大きくなること、注目点候補はピクセル単位で設定するため、遠方の点の密度は近くの点の密度に比べて疎になることを考えると、クラスタの結合範囲を距離にかかわらず一定とすると遠方の物体が検出されなかったり、近接する物体が不必要に結合したりしてしまう。そこで、結合範囲を平行ステレオの誤差に従って、

$$\Delta Z(Z) = \frac{2\delta_u b f Z^2}{b^2 f^2 - \delta_u^2 Z^2}$$

$$\Delta X(X, Z) = \begin{cases} -\frac{\delta_u Z(2X-b)}{b f} & (X < 0) \\ \frac{\delta_u Z}{2f} & (0 \leq X \leq b) \\ \frac{\delta_u Z(2X-b)}{b f} & (X > b) \end{cases} \quad (4.9)$$

とする。ただし、 δ_u は有効画素間隔、 f は焦点距離、 b はベースライン距離を表す。さらに生成されたクラスタの合計投票数 N_c に対して、

$$N_c > T \quad (4.10)$$

として、一定の閾値 T で処理することでノイズを除去し、障害物を抽出する。障害物は、実空間中の位置とステレオ画像中での位置を求めることができるため、自動車の制御やその他の画像処理に利用し易いと考えられる。

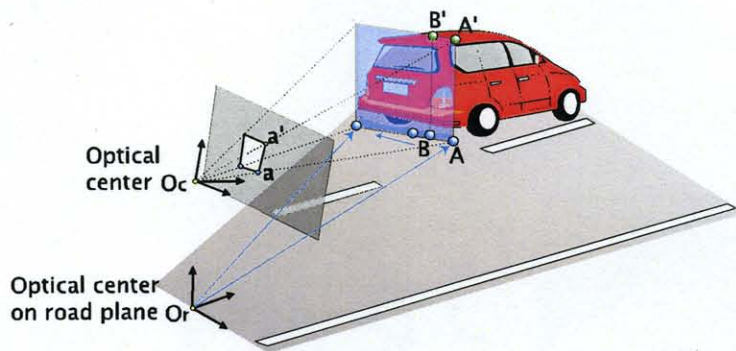


図 4.8: 障害物描写方法
Fig. 4.8: Obstacle's rendering mask.



図 4.9: 障害物描写例
Fig. 4.9: Obstacle's rendering example.

4.4 障害物の描写

検出された障害物を画像に描き入れることで、障害物の位置を人が視覚的に知ったり、その他の画像処理に利用することもできる。そこで、障害物は、道路との接地面と道路面からの高さ T (実験結果はすべて 1.5[m] で固定) までとして描いた。図 4.8 は、描写方法を簡略化したものである。描写をするための障害物マスクの生成手順について、以下に述べる。

1. 検出された障害物と道路面の接点(ここでは点 A とする)を探索する。接点は、基準カメラの光学中心 O_c を道路平面に投影した点 O_r を中心として、角度を変えながら放射状に伸ばすことで探索する。求められた接点の道路座標系における横と奥行きを求める。また、道路面に接していることから接点の高さは 0 である。
2. 接点 A を道路座標系からカメラ座標系へ式 (4.5) を利用して変換する。
3. 透視投影行列を利用して、カメラ座標系における接点 A を画像座標上の点 a に変換する。
4. 接点 A と横と奥行きが同じ位置にあり、高さ T である点 A' についても 1~3 を行なう。
5. 点 O_r からの角度を変え、接点 A と同様に接点 B を求め、1~4 を行なう。

以上の処理によって障害物を描写した結果を図 4.9 に示す。生成された同図上段の障害物マスクに対して、それを原画像へ書き込んだ結果が同図下段である。

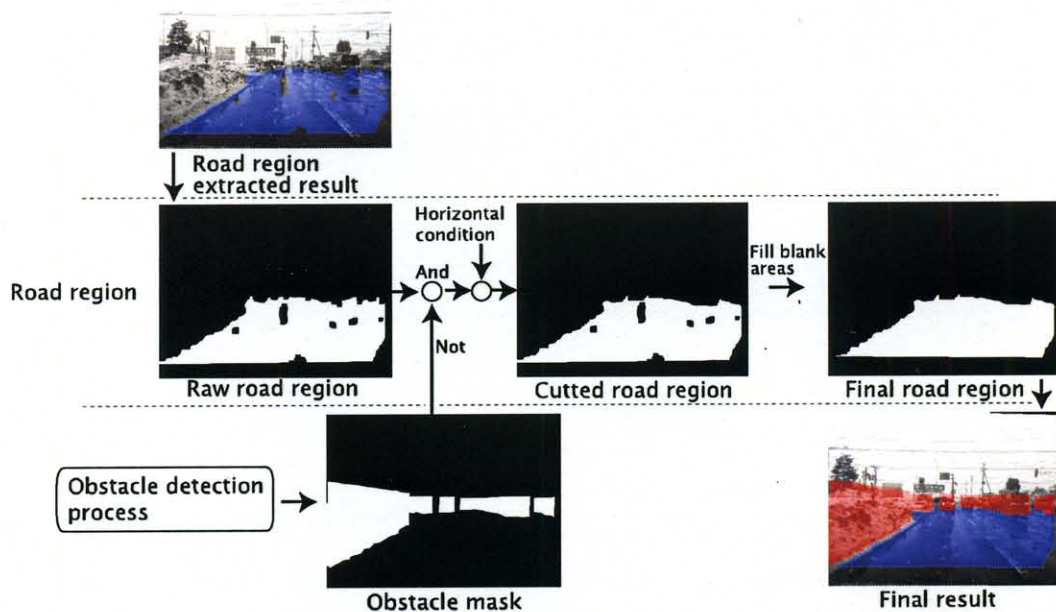


図 4.10: 道路領域の修正
Fig. 4.10: Revision of road regions.

4.4.1 道路領域の修正

検出された障害物情報を利用して、抽出された道路領域を修正する。

道路領域の乗り上げ修正

画像テクスチャのエッジがエピソード線と平行な領域やテクスチャの少ない領域 (テクスチャレス処理を行っても取り切れない領域) では、道路領域が非道路領域に乗り上げてしまうことがある。そのような場合には、障害物検出の結果を反映されることによって、道路領域の修正する。

また、道路領域は水平線以上まで乗り上げることはない。道路領域抽出処理によって、道路面の姿勢が推定されていることから、水平線以上に乗り上げた道路領域に関しても、修正を行なう。

道路領域の穴修正

雨などによって路面が濡れると鏡面反射が起こり易く、写り込みがあった場合にはステレオカメラでその見え方が異なってしまう。それにより、道路領域に穴が開くことがある。それらを障害物検出結果を利用して修正する。

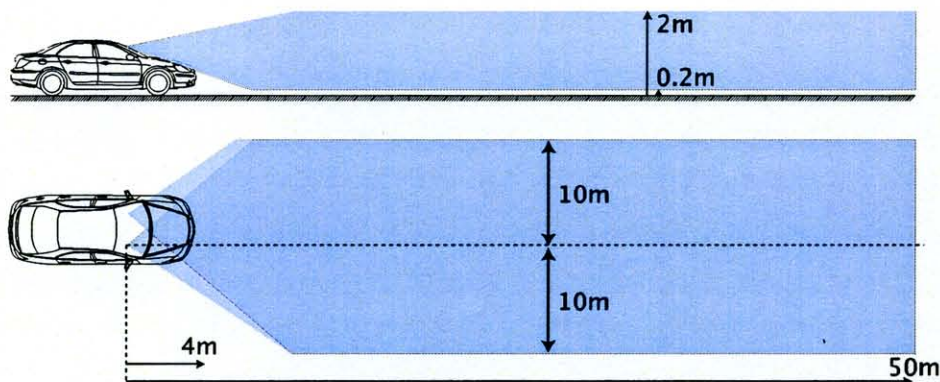


図 4.11: 検出範囲
Fig. 4.11: Detection range.

図 4.10 は、これらの 2 つの修正処理を表している。道路領域抽出処理によって抽出された道路領域を同図中の “Raw road region” とする。先の障害物描写処理によって得られた “Obstacle mask” を利用して、まず乗り上げ箇所を修正する。そのために “Obstacle mask” の NOT を計算し、“No revision” との AND をとる。また、水平線以上の乗り上げ領域に関しても、道路領域抽出処理で併せて推定された道路面の姿勢パラメータを利用することで除去する。その結果が “Cutted road region” であり、乗り上げ箇所などのドル尾領域が不要な領域が除去された結果となっている。次に、“Cutted road region” に対して穴埋め処理を行なう。穴埋め処理は、道路領域に内包された領域のみに対して行なう。その結果が “Final road region” である。仮に障害物が穴領域に存在した場合でも、先の不要な道路領域を除去する処理によって、その穴領域は内包された領域ではなくなり、穴埋め処理の対象外となる。同図では、当処理によって鏡面反射によって道路面に開いた穴が埋まっていることが分かる。

4.5 実験結果

本節では、提案手法を実画像に適用して、障害物を検出した結果を示す。障害物の検出範囲は、図 4.11 に示すように左右の幅方向に 10m ずつ、奥行きは 4m 以上 50m 以内とした。また、高さ方向に関しては、道路平面から 20cm 以上 2m 以下とした。

4.5.1 道路領域と障害物の検出結果

自動車はさまざまな時間、天候、道路環境を走行する。そのため、実際に利用可能な障害物検出機能であるには耐環境性が求められる。そこで、提案手法の有効性を確認するため、さまざまな環境下で撮影された画像に対して本手法を適用した。

Point Grey 社製カメラによる結果

図 4.12, 4.13, 4.14 は, Point Grey 社製 Flea カメラを利用した実験結果である。

まず, 図 4.12 は, 高速道路における結果である。図中の赤い領域は検出した障害物領域, 青い領域は道路領域をそれぞれ表している。赤線は推定された道路面の姿勢パラメータを利用して計算した水平線である。同図 (A) では, 両壁が正しく検出されている。(B) では, 両壁の他に遠方のバイクや車両が検出されていることが分かる。(C) は, 両壁や駐車車両と手前のカラーコーンが検出されている。(D) は, 首都高速のカーブであるため, 水平線が若干傾いて推定されていることが分かる。トラックや側壁が検出されている。次に, (E) と (F) では自動車の他に壁がカーブに沿って推定されていることが分かる。(E) では, 上方の看板があるものの, それを検出していないことも分かる。(G) は, 逆光であるため, 薄暗く, 加えて前方の車両は黒色のため, 見えにくい, 正しく検出できていることが分かる。(H) は, トンネル内を走行している。前方や側方の車両が検出できている。(I) は, 薄暗い道路であるが, 側壁や左側の車両が検出されていることが分かる。

図 4.13 は, 市街地における結果である。市街地は, 一般的に高速道路より様々な障害物が存在し, その大きさや形状が多様である。また, 道路も高速道路ほど整備されていないため, 白線が掠れていたり, 無かったりする。同図 (A) は, 両壁や前方の車両が検出されている。(B) は, 道路白線が描かれていないものの, 平面の姿勢が正しく推定されており, 対向車や樹木などが障害物として検出されている。(C) は, 交差点で多くの車両が存在しているが, それらすべてが障害物として検出されていることが分かる。(D) は, 電柱や人やバスなどが検出されている。また, 縁石は道路平面領域抽出で道路と判定されているため, 障害物としては検出されていない。(E) は, 車両や中央分離帯が検出されている。(F) では, 車両やガードレールがカーブのそのままに近い形状で検出されている。(G)-(I) は, 夕方の薄暗い中でカメラのゲインを上げて撮影したため, 画像にノイズが多く入っている。そのような場合においても, ガードレールや車両などが検出できている。

次に, 図 4.13 は悪天候下で撮影された画像に対して適用した結果である。(A)-(C) と (F) は, 降雪中のシーンである。画像中には雪粒が写りこんでいるため, 大きなノイズが発生している。(A) と (B) は, 路面で溶けた雪によって道路面が濡れており, 道路上の物体によって反射が起こっている。そのため, 若干道路領域に欠けが生じているものの, 障害物については車両や雪壁なども検出できている。(C)-(F) では, 路面の雪と雪壁の区別がつきにくい, 障害物検出処理によって, 推定された道路面から設定した高さ以上の部分が壁として検出されていることが分かる。(F) は暗い環境下であったのでカメラのゲインを上げて撮影されており, ノイズが目立っているものの, ガードレールやバスが検出できている。

図 4.15 は, 4.3.3 節で述べた投票グリッドに検出された障害物のみを残して表示した結果であり, 障害物の検出位置を表している。同図左は, 図 4.12 の (F) に相当する結果であ

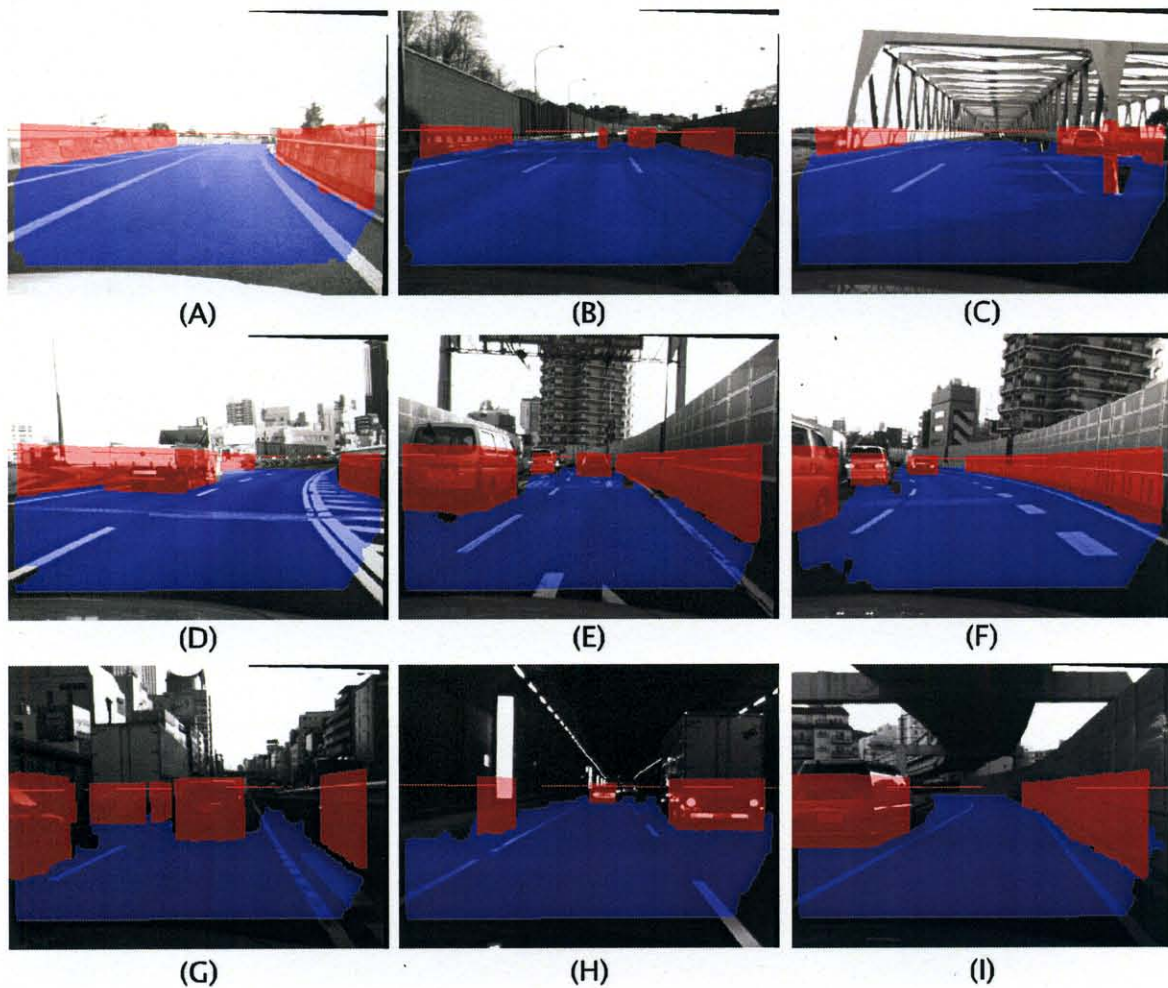


図 4.12: 高速道路における障害物検出結果 (青領域が抽出された道路領域. 赤領域は検出された障害物領域)

Fig. 4.12: Experimental results under highway environment (red regions are extracted road region, blue regions are detected obstacles.).

る。右側の道路側壁が左に曲がっているのと左側の前走車3台が観測できる。それぞれの色は、セグメンテーションをして得られたクラスごとに同じ色で表している。同図真中は、図 4.13 の (G) に相当する結果である。同図では、ガードレールが空間中で直線になっていることが分かり、他にも車両などのさまざまな障害物が検出されている。同図右は、図 4.14 の (A) に相当する結果である。右側の車両は、セグメンテーションによって、2つに分かれたクラスタになってしまっていることが色の違いにより分かる。これは、セグメンテーションで位置情報のみを使っているためである。

通常、降雨または降雪している場合には、フロントガラスの水滴・雪粒を落とすため、ワイパーを作動させる。図 4.16 は、そのような場合の実験結果を示している。フレーム 716

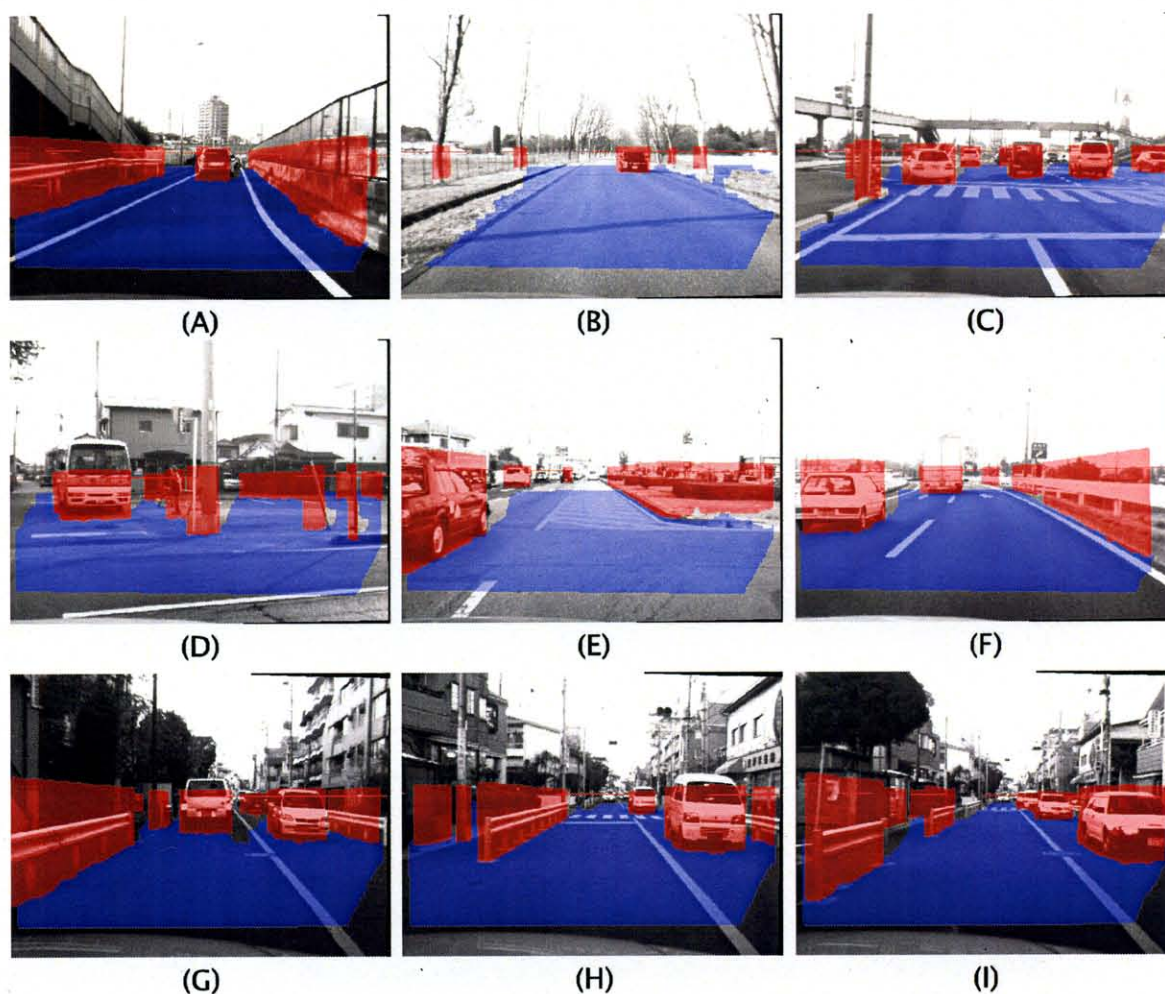


図 4.13: 一般道路における結果 (青領域が抽出された道路領域. 赤領域は検出された障害物領域)

Fig. 4.13: Experimental results under general road environment (red regions are extracted road region, blue regions are detected obstacles.).

番では、両画像にワイパーが写りこんでいないため、すべての車両や雪壁が検出されている。次に、フレーム 722 番では、参照画像にワイパーが写っているため、その部分に関してステレオ対応点探索が不可能なことから、遮蔽領域である前方の車両が検出できていない。同様にフレーム 728 番でも基準画像に写っていない右側の車両については検出できていない。本手法では、障害物検出に時系列的な情報を入れていないためであるが、ワイパーで遮蔽されたとしても、その後画像に写れば検出できており、多少の遮蔽が起きても道路領域抽出処理も含めて、処理が不安定になっていないことが分かる。

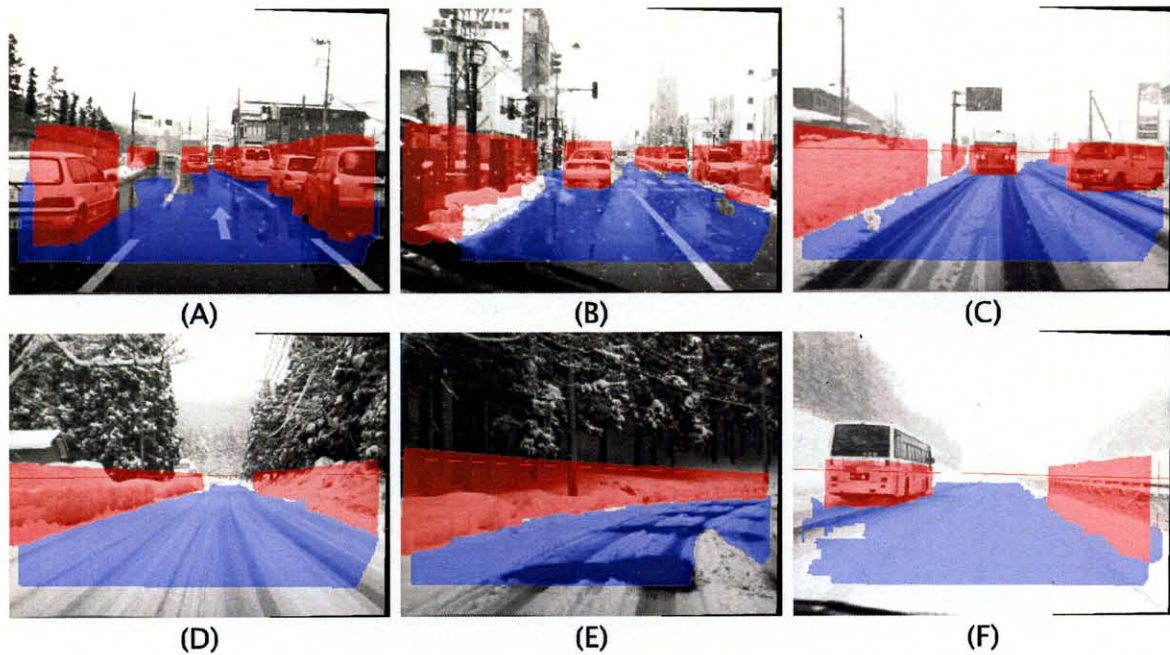


図 4.14: 降雪, 積雪した環境における障害物検出結果
 Fig. 4.14: Experimental results under wet road and snowy condition.

High Dynamic Range カメラによる結果

図 4.17 は, High Dynamic Range(以下 HDR) カメラで撮影された画像に対して適用した結果である. HDR カメラでは, 広い範囲の明るさに対応できるため, 例えば白い雪壁では輝度値が飽和してしまったり, 暗い環境下では輝度値が低くノイズと信号の区別が付きにくいような場合においては, 通常のカメラよりはっきり撮影ができる. 本節では, 通常のカメラと HDR カメラによる障害物検出結果を分けて述べたが, カメラと障害物検出の性能についての検討は特には行なっていない. 図 4.17 には, 一般道路や高速道路のさまざまな天候についての結果である.

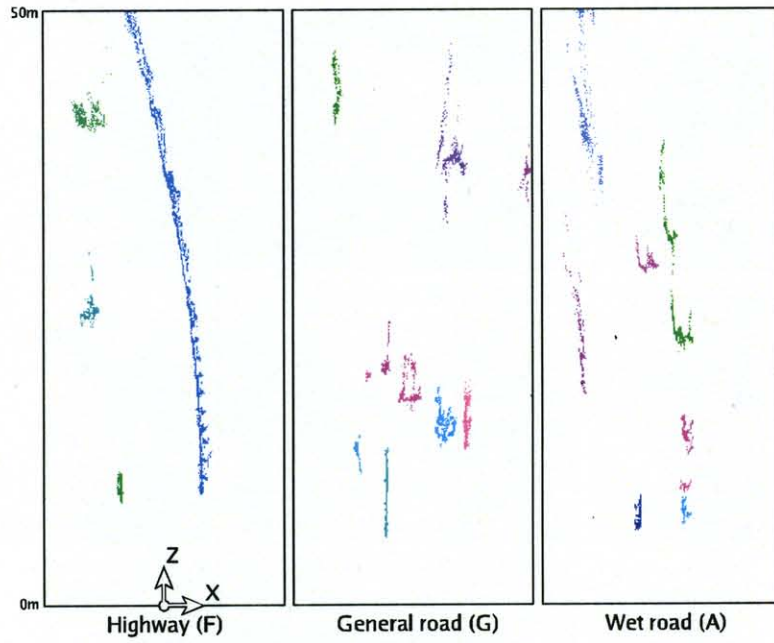


図 4.15: 障害物の位置検出結果
 Fig. 4.15: Location detection results.

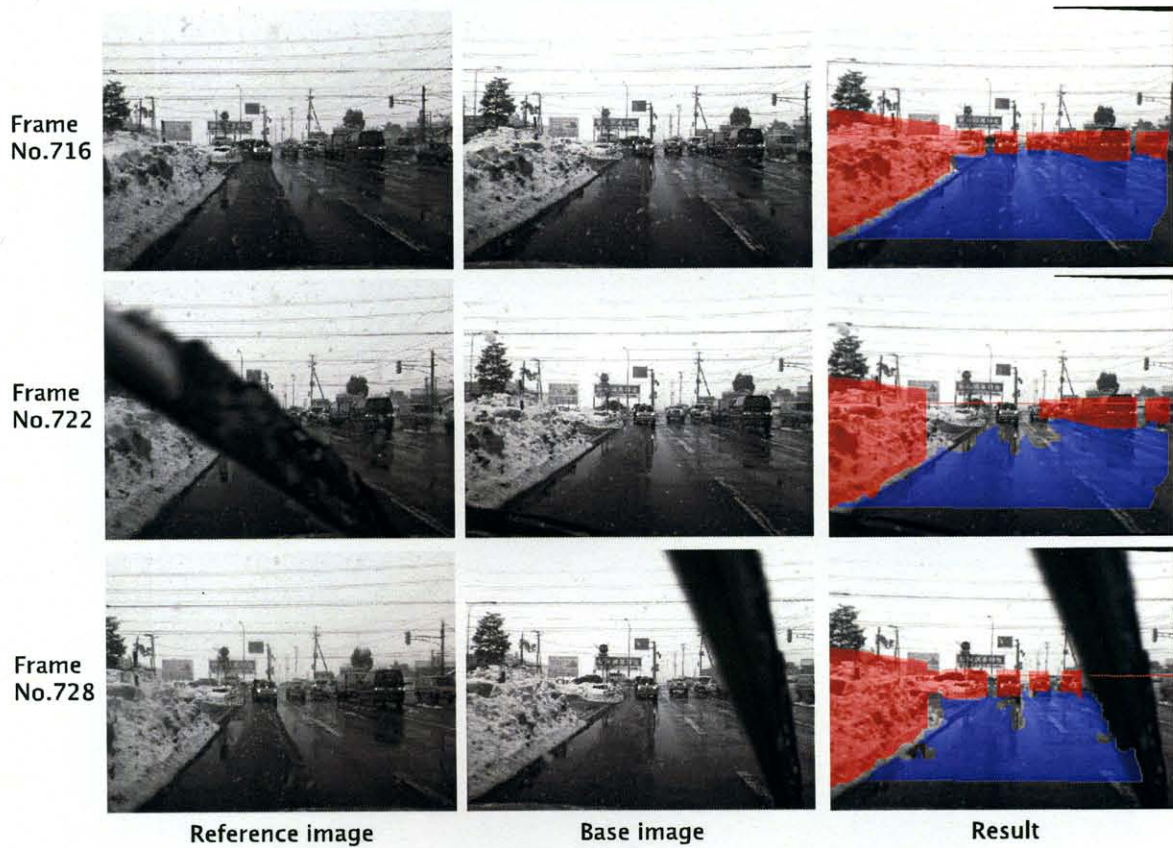


図 4.16: ワイパーが写り込んだ場合の結果
 Fig. 4.16: Experimental results with wiper.

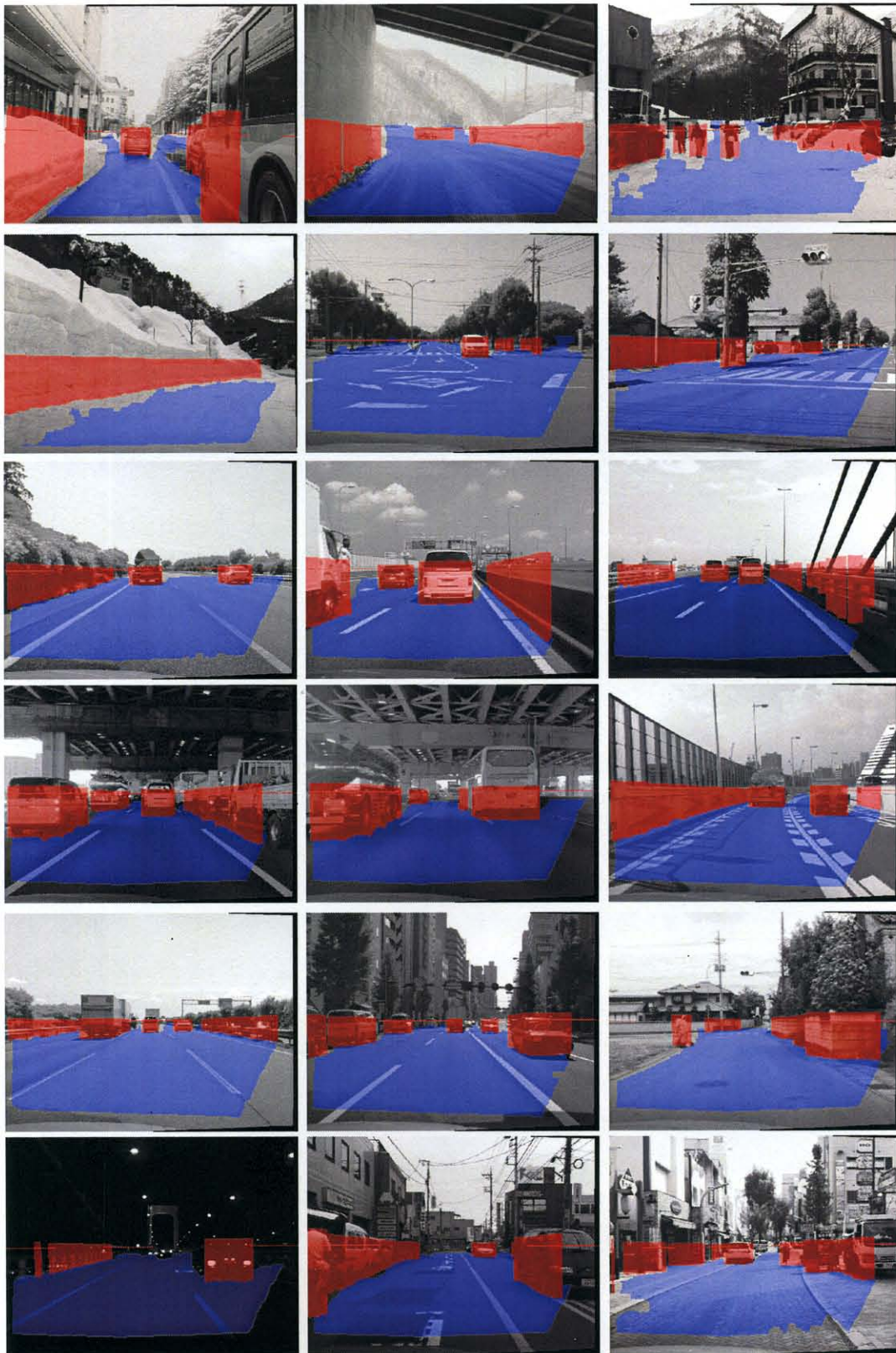


図 4.17: HDR カメラによる様々な環境下における結果
 Fig. 4.17: Experimental results under various environment using HDR camera.

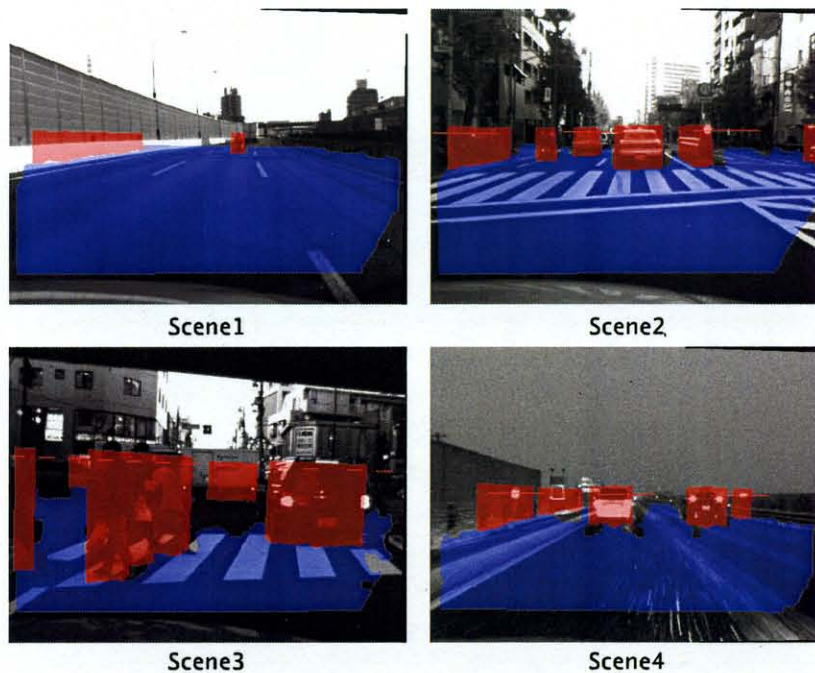


図 4.18: 検出率評価シーン
Fig. 4.18: Sequence used for detection rate evaluation.

4.5.2 障害物の検出率

表 4.1 は、図 4.18 のシーン 1 から 4 に対して、提案手法を適用した結果である。表中の「車両」「2 輪車」「歩行者」には、画像に写るそれぞれの個数を手動で数えて「種」にはその種類数を、「出現」は出現した個数を記した。「正検出」は、正しく検出された障害物の数である。検出率は、正検出の合計を出現した数の合計で割った比率である。また誤検出率は、誤検出数の合計を検出された数の合計で割った比率である。

次に表 4.2 は、表 4.1 で示したシーンにおける誤検出と未検出の内訳である。誤検出は、誤検出が生じた個数をカメラからの距離ごとに分けた。未検出についても同様である。

まず、シーン 1 では晴天の高速道路を走行している。表 4.1 より、出現した車両 184 個はすべて検出されているものの、2 輪車が検出されない場合があり、合計すると検出率が 97% である。未検出となった 2 輪車は、表 4.2 に示すようにカメラから 30~45[m] に存在していることがわかる。この未検出の原因は、2 輪車が遠方であるために画像に写る面積が小さく、かつ隣接車線の車両の陰によって画像中で明確に映らない場合であった。

シーン 2 は、晴天時の市街地を走行している。未検出は発生しなかったが誤検出が遠方で少し発生している。誤検出の原因については、4.5.4 節でまとめて述べる。

シーン 3 は、夕暮れの交差点で停止している車両の前を障害物が移動している。このシーンでは、未検出誤検出ともになく、安定して障害物検出ができています。

表 4.1: 障害物の検出率
Table 4.1: Obstacle detection rate.

Scene (画像枚数)	車両			二輪車			歩行者			検出率 [%]	誤検出率 [%]
	種	出現	正検出	種	出現	正検出	種	出現	正検出		
1 (301)	2	184	184	1	301	286	-	0	0	96.9	0.00
2 (501)	13	1123	1123	1	294	295	-	0	0	100	1.53
3 (101)	4	156	156	1	65	65	3	129	129	100	0.00
4 (101)	2	174	174	-	0	0	-	0	0	100	0.57

表 4.2: 誤検出と未検出の距離別内訳
Table 4.2: False detection and non-detection rate.

Scene	誤検出の距離と個数			未検出の距離と個数		
	0-15[m]	15-30[m]	30-45[m]	0-15[m]	15-30[m]	30-45[m]
1	0	0	0	0	0	15
2	1	4	17	0	0	0
3	0	0	0	0	0	0
4	1	0	0	0	0	0

シーン 4 は、降雪中の夜間である。障害物が比較的近いところに存在することもあって、障害物は完全に検出されている。

また、図 4.18 の連続シーンとは別に、多くのシーンの中から画像を取り出し、車両の種類を増やした実験を行なったところ、186 種類の車両に対して検出率は 98.9%、誤検出率は 3.2% であった。これらの結果から提案手法が有効であることが確認できた。

4.5.3 他手法との比較実験

本節では、V-disparity[27] を姿勢推定に利用したものと第 3 章によるものとの比較結果を示す。道路面の姿勢推定部分に V-disparity を利用したものを以降では、V-disparity ベース手法と呼ぶこととする。V-disparity ベース手法では画像全体について 3 次元再構成を行ない、得られた視差を利用して Hough 変換をし、それによって道路平面に相当する直線を求め、その直線の傾きと切片から道路面の姿勢を求める。検出する道路面の許容範囲は、俯角が 0 から 10[degree] まで、平面の高さを 1.0 から 1.5[m] までの間とした。

検出結果の比較

図 4.19 は、雪道で障害物検出をした結果である。図中の赤線は推定された平面の姿勢を利用して描いた水平線である。雪道ではテクスチャが少なく、小ウィンドウを使った対

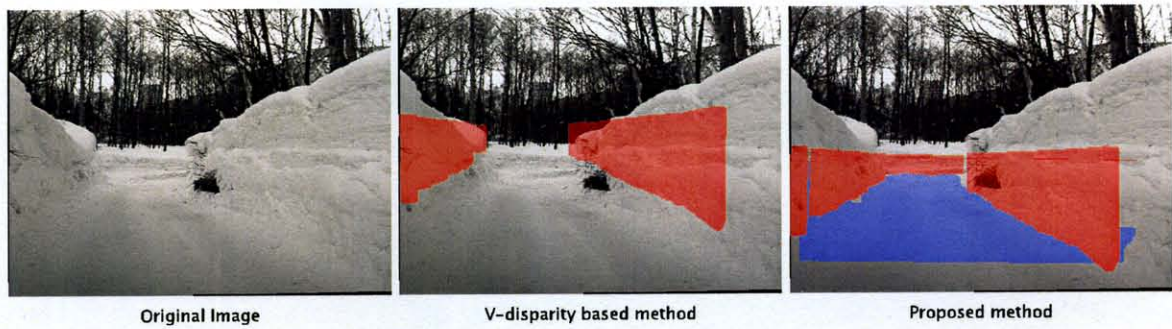


図 4.19: V-disparity ベース手法との比較結果. 赤線は推定された水平線を表す.
 Fig. 4.19: Compared results with V-disparity based method.

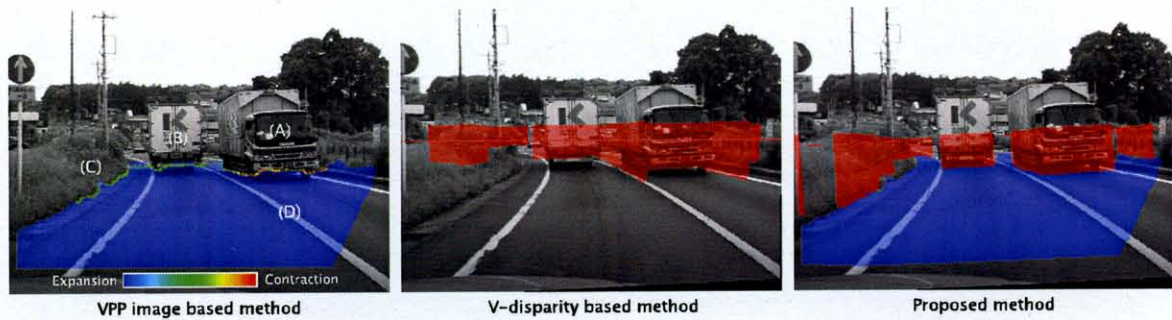


図 4.20: 第 3 章の手法と V-disparity ベース法との比較実験結果.
 Fig. 4.20: Compared results with chapter 3 and V-disparity based method.

応点探索では対応がとりづらいため、V-disparity ベース手法では道路面の推定が正確に行なわれていない。そのため、前方の雪壁が正しく検出されていない。一方、提案手法では道路面を1つの変形可能なウィンドウとして対応をとるため、正しく姿勢が推定され、雪壁がすべて検出できている。

図 4.20 は、3 章の手法 (VPP 画像ベース手法と表記) と V-disparity ベース手法と提案手法の結果である。このシーンではロール角方向に傾いた路面を走行している。このときの障害物の位置関係を示したものが図 4.21 である。同図は、VPP 画像と投票グリッドであり、すべての画像でスケールを合わせてある。図 4.20(A) は対向車であるため、平面端の点が自車に向かう方向で危険を示す赤となっている。しかし、その障害物までの距離は図 4.21 を見ると提案手法と比べて 3.5[m] ほど後方と推定されている。これは、図 4.1 で説明したように車両の下の路面がカメラから見えるためである。一方で提案手法は障害物までの距離を計測しているため、その影響がない。次に V-disparity ベースの手法では、図 4.20(A)-(C) は検出されているものの、ロール角の推定をしていないため同図 (D) のような路面パターンが誤って障害物として検出されている。提案手法では正しく障害物が検出され、誤検出もないことがわかる。

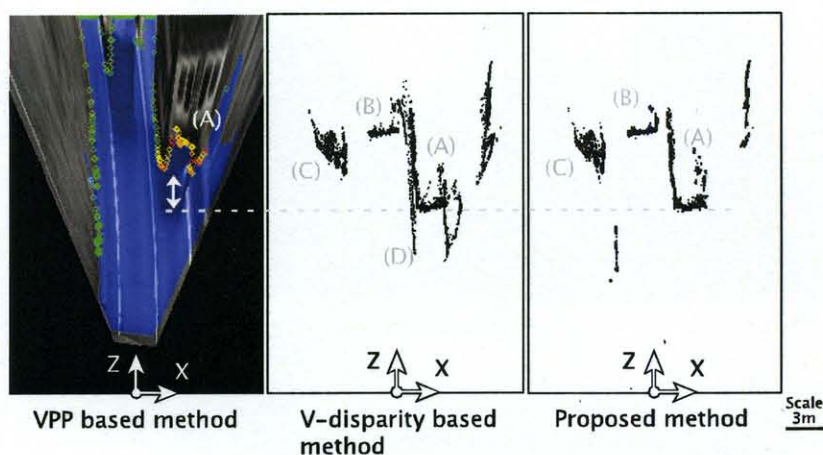


図 4.21: 障害物の位置検出結果. VPP 画像と投票グリッドのスケールを合わせて表示.
 Fig. 4.21: Results of detected position.

表 4.3: 計算時間の比較
 Table 4.3: Comparison about calculation cost.

処理 \ 手法	第 3 章の手法	V-disparity ベース	提案手法
道路領域抽出	5.85e-2	—	5.85e-2
障害物検出	1.12e-2	1.39	9.41e-1
合計	6.97e-2	1.39	1.00

※ 7 シーンで実際に処理した計算時間の平均.

計算時間の比較

表 4.3 は, 提案手法全体の計算時間を 1.00 とした場合のそれぞれの手法の計算時間を表している. 第 3 章の手法は, ステレオ対応点探索を行わないため, 計算時間が際だって短い. しかし, 前述したように障害物が道路面に接していないと障害物までの正確な距離を求めることができない. 一方, 提案手法ではステレオ対応点探索を行なうため, 計算時間はかかるものの障害物までの距離を常に正しく求めることができる. また, 提案手法では道路領域抽出処理によって対応点探索で限定ができることから, V-disparity ベース手法に比べて約 40% 高速であることがわかる.

4.5.4 誤検出について

さまざまな環境下で提案手法を適用した結果, 誤検出の原因は大きく 2 つに分けられることが分かった.

- 入力画像によるもの



図 4.22: フレアによる誤検出例
 Fig. 4.22: False detection example for flare.

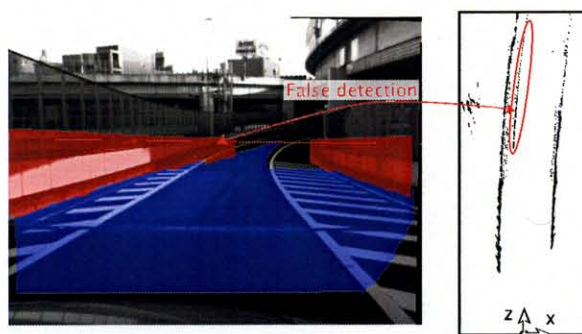


図 4.23: 複数平面による誤検出例
 Fig. 4.23: False detection example for multi plane.

太陽光がカメラに入り込むことによって、スミアやフレアとしてステレオ画像に現れると、それを障害物として検出してしまう場合がある。その例が図 4.22 である。ステレオ画像にフレアが写り込んでおり、誤検出をしている。

また、本研究では二眼ステレオカメラを利用しているため、繰返しパターンが存在する場合には、ステレオ対応誤りが発生することがある。そのため、異なった位置に物体が計測されたり、誤って検出されたりする場合がある。繰返しパターンによる影響は、カメラの配置と繰返しパターンの周期によって決まるが、自動車の場合は走行して対象までの距離が変化することでそのパターンの周期が変わるため、シーケンス中の一部のフレームだけ誤検出が発生するといった現象が起こる。また、ステレオ対応誤りはオクルージョンの場合にも発生することがある。

- 想定モデルとの差異によるもの

道路面を単一平面として仮定しているため、図 4.23 のように道路面が勾配の異なる複数の面から形成されている場合には、障害物の検出を誤ることがある。同図では、前方で道路面が持ち上がっているため、道路白線を誤って障害物として検出してい

る。通常の走行時において路面形状によって誤検出が発生する場合はほとんどの場合が遠方であった。

4.6 本章のまとめ

本章では、障害物を自車両が通過する可能性のある空間に属する立体物とし、道路面を基準としたロバストな障害物検出手法について述べた。提案手法は、ステレオ計測を利用して、画像中の各点の空間中の3次元位置を測定し、それらの点の集合から、道路上に存在する立体物を、道路面の姿勢を利用して、分離し、さらにセグメンテーション処理を行なって、障害物を検出した。その際、ステレオ画像から道路領域を検出する処理を用いることで、画像中に障害物が写っている領域である、非道路領域だけに限定して、対応点探索を行なうことや、動的に推定されたカメラと道路面間の姿勢パラメータを利用することで車両の振動に耐えられること、さらに投票グリッドを用いることで自動車、バイク、歩行者などの道路環境に存在する物体をそのままの形状に近い形で、安定に検出できることなど、道路環境の拘束条件をそれぞれの処理に巧みに取り入れることで、実環境で実際に機能するシステムを構築した。

今後の課題として4.5.4節で述べたように、稀に発生する誤検出の原因の1つとして挙げられる繰り返しパターンによるステレオ対応誤りに対して、時系列情報を取り入れて誤検出低減を図ることが必要であると考えている。

また、計算時間に関しては、特に高速化処理を加えていないプログラムを利用しているため、Intel Pentium 4[®] 2.8GHzの汎用計算機を使って、すべての処理にVGA画像(640×480ピクセル)では約13秒、QVGA画像(320×240ピクセル)では約1.9秒ほどかかっている。その計算時間の大部分が対応点探索であるため、文献[20]や主に顔検出で利用されているIntegral image[80]のように工夫したり、文献[14, 84, 28]のようにGPU(Graphics Processing Unit)などのハードウェアを利用して並列計算させたりすることで計算時間を大幅に短縮できると考えている。

第5章 自車両の運動推定

本章では、車載ステレオカメラを利用した自車両の運動推定手法について述べる。通常、車両の運動を推定するには、車速センサー、舵角センサーやヨーレートセンサーからの情報を使う場合がほとんどである。しかし、車輪の回転を利用した車速センサーなどからの情報では、路面の動きを直接計っていないため、車輪に滑りが発生した場合などに正確な速度を求めることができない。一方、カメラを利用することで直接路面の動きを計測することもできる。車載カメラからの映像を利用して自車両の運動推定を行うことにより、その推定結果から、車両の走行状況をモニタリングしたり、ナビゲーションや安全走行のための運転支援などに利用できる。本論文では、第6章で推定される障害物のカメラに対する相対モーションを静止座標系を基準とした絶対速度へ変換するために、自車両の運動推定結果を利用した。

さて、自車両の運動推定は、その車両に搭載されたカメラのエゴモーション推定を行なうことと同等であるため、以降ではカメラのエゴモーション推定を自車両の運動推定と同じものとして扱う。

本章では、まず自車両の運動推定手法に関する関連研究について述べ、提案手法の位置付けを明確にする。続いて、提案手法の推定方法について述べる。提案手法の推定精度を合成画像を利用して考察し、実画像を利用してさまざまな環境下で適用可能であることを示す。最後に、本章のまとめを述べる。

5.1 関連研究

時系列画像から、カメラの3次元モーションを推定するには、シーンに含まれるモーションからカメラのモーションを取り出せばよい。シーンのモーションとその奥行きを推定する問題は相互に結びついており、どちらかが既知の場合には、残りの未知パラメータを比較的容易に推定することができる[79]。カメラのエゴモーション推定では、シーンに含まれるモーションからカメラのモーションを取り出す方法と、そのモーションの推定方法が重要な要素となる。

画像に写っている3次元モーションが単一の場合には、単眼カメラで時系列画像を撮影し、時系列画像間での注目点のトラッキングによって得られた軌跡を因子分解することで、3次元モーションと各注目点の奥行きを推定する手法が提案されている[76]。しかし、

この方法ではカメラの絶対的なモーシヨンの大きさを特定することができない。次に複眼を利用する方法として、画像中の複数の注目点についてステレオ画像間の対応から3次元位置を計測し、それら注目点の3次元位置の時間的変化により運動を推定する方法 [31] が提案されている。この方法では注目点のトラッキングに成功し、かつその点に関してステレオ3次元計測が成功することが必要である。また、これらの手法では注目点が静止領域に含まれていなくてはならず、もし移動領域に含まれている場合には、カメラのエゴモーシヨンと移動領域のモーシヨンとの相対モーシヨンが推定されてしまう。そのため、市街地環境のように画像中に多くの移動物体が存在する場合には安定性に欠けると考えられる。

モーシヨン推定と奥行き推定は相互に結び付いていると先に述べたが、その推定精度についても同様の関係がある。注目点の空間位置がカメラから遠い場合には、奥行きの推定精度が低下すると同時に、その点のモーシヨンの推定精度も低下する。そのため、精度よくエゴモーシヨンを推定するには、できる限りカメラに近い注目点を設定した方がよい。自動車の運動推定においては、カメラに最も近い静止領域が道路領域であるため、その情報を積極的に利用した方がよい。また、道路面のモーシヨンを推定するために、道路面の奥行きに相当する、道路面のカメラに対する3次元的な姿勢を推定する必要がある。

そのようなものとして、高速道路の白線の長さのように規格化されている情報を利用することで、車両速度を算出する方法 [33] が提案されている。また、スリットカメラを利用して特別な模様を読み取ることで、速度とともに自車両の位置を測定する方法 [23] も提案されている。しかし、規格化された路面パターンは限定された環境下でしか利用できない。

特定の路面パターンの情報を用いない方法として、単眼カメラから得られる画像中のフローから、自車両の運動を求める方法 [74, 71] が提案されている。しかし、道路領域内に路面模様がないとフローを正しく求めることは難しい。さらに、画像中の道路領域に属するフローを安定に特定することは容易ではない。それ以外にも、道路面によって定義される時系列画像間の射影変換行列を勾配法を利用して求めることで、運動を推定する方法 [24, 3] も提案されている。しかし、単眼カメラを用いた方法では、道路面とカメラの間の距離を求めることができないため、推定される並進運動は定数倍の不定性を持ち、その絶対的な大きさを知ることはできない。また、勾配法に必要な初期値に関して言及されていない。

提案手法では、まず、第2章の方法によって、ステレオ画像から道路平面領域を抽出することで静止領域を得る。さらに道路面を抽出する際に動的に推定されるカメラと道路間の姿勢を用いて、道路面を仮想的に上方から見た画像 (Virtual Projection Plane image: VPP 画像) に変換する。VPP 画像を導入することで、車両の運動を2次元平面内の並進と回転の3自由度に限定できる。自車両の運動は、抽出した道路面領域のパターン全体を使い、

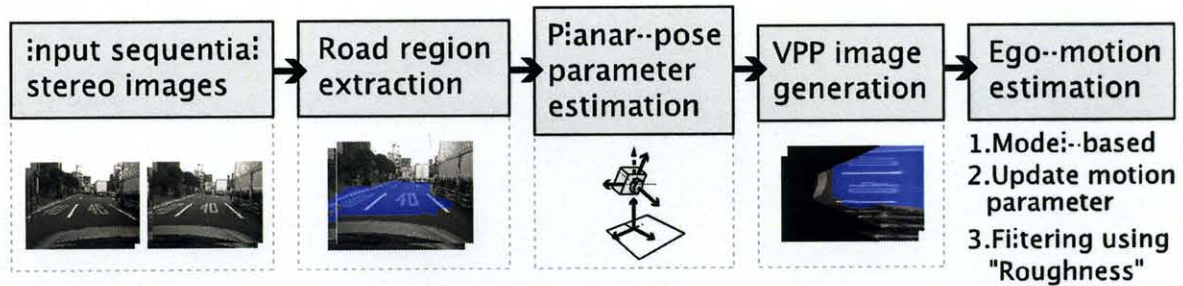


図 5.1: 運動推定手法の全体図
 Fig. 5.1: Overview of ego-motion estimation process.

時系列にわたって VPP 画像をマッチングすることで直接的に推定する。その際、まず車両の運動モデルを導入することにより、VPP 画像の道路平面領域全体に対する平行移動だけのマッチングとその重心座標から車両の運動パラメータ (並進移動と回転角) を推定する。この方法によって、計算量を軽減しながら安定に運動パラメータを推定することができる。さらにその結果を初期値とし、勾配法を用いて更新することで、より高精度に自車両の運動を推定する。

5.2 運動推定手法の流れ

運動手法の流れは図 5.1 である。まずステレオ画像間の射影変換行列を動的に推定しながら道路平面を抽出する。さらに、カメラと平面間の姿勢を表すパラメータを推定し、平面の法線ベクトルとカメラから平面までの距離を求める。平面の姿勢パラメータは、動的に推定された射影変換行列から求めるため、走行時の振動等によるカメラに対する平面の相対的な姿勢変化に対応できる。次にその姿勢パラメータを使って、道路平面を仮想的に上方から見た画像 (VPP 画像) を生成する。VPP 画像については、3.1 節で述べている。

次いで、隣接したフレームの VPP 画像を使い、道路平面領域内に対してマッチングをすることで自車両の運動を推定する。その際、まず車両の運動モデルを用いて計算量を軽減しながら安定に運動パラメータを求めた後、その運動パラメータを勾配法により更新することで、より高精度に求める。最後に、類似度分布によって推定される“Roughness”に基づいて観測誤差分散を変化させるフィルタリングを行なう。

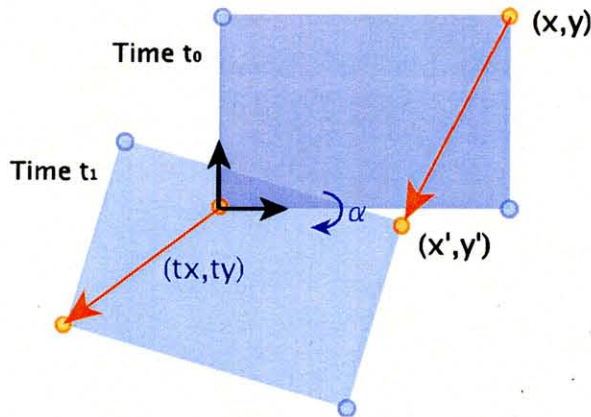


図 5.2: 2次元空間上の剛体のモーショ
Fig. 5.2: Motion on 2D plane.

5.3 自車両の運動推定

5.3.1 車両の運動モデルを利用した運動パラメータの推定

図 5.2 のように、ある剛体の平面内の回転と並進運動を考える。まず、一般的な場合において、回転と並進による 3 つの未知数 (運動パラメータ) を推定するために必要な注目点と移動ベクトルの個数について考察する。

ある点 (x, y) が原点のまわりに α だけ回転し、 (t_x, t_y) だけ並進運動したとき、運動後の点 (x', y') は

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (5.1)$$

となり、移動ベクトル $(\Delta x = x' - x, \Delta y = y' - y)$ は以下のように表される。

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} x(-1 + \cos \alpha) - y \sin \alpha + t_x \\ x \sin \alpha + y(-1 + \cos \alpha) + t_y \end{bmatrix}. \quad (5.2)$$

よって、1 つの注目点の座標 (x, y) と移動ベクトル $(\Delta x, \Delta y)$ が分かると式が 2 つ得られる。一方、運動を表す未知数が 3 つであるので、運動している物体 (ここでは道路平面) 上の 2 つ以上の注目点とその移動ベクトルを利用すれば、運動パラメータを推定できる。しかし、各点の移動ベクトルを求めるために、例えば小ウィンドウを使う必要があったり、それらの点の中から正しく推定された移動ベクトルを選定したりしなくてはならず、容易ではない。

次に車両の運動モデルを利用することで、1 つの注目点とその移動ベクトルから運動パラメータが推定できることを示す。

まずアッカーマンステアリングジオメトリ [1] を想定し、自動車を 2 車輪モデルに当てはめる。4 車輪モデルを 2 車輪モデルに当てはめるために、カメラの設置位置を通る車両

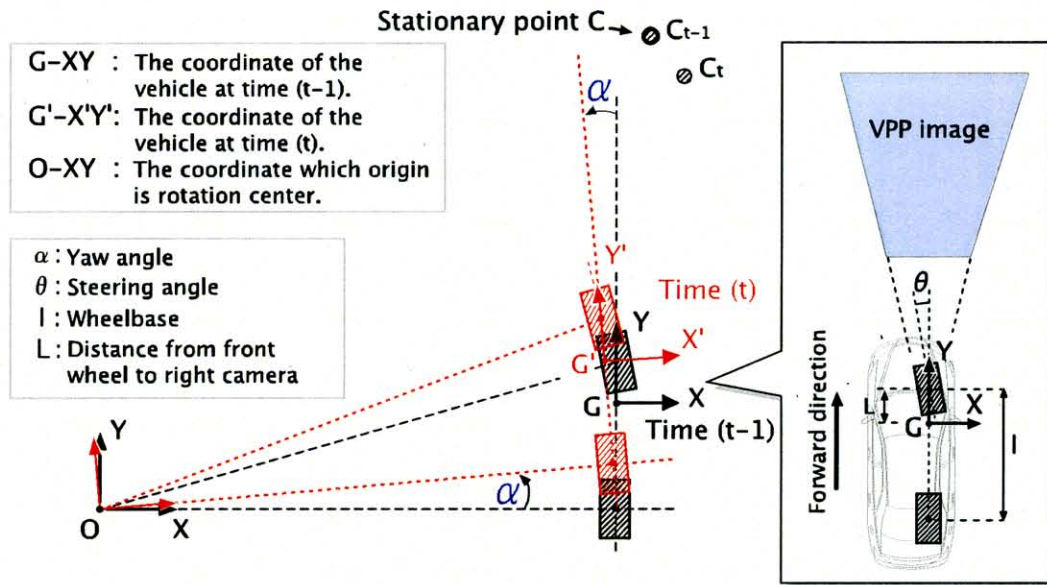


図 5.3: 車両の運動モデル
Fig. 5.3: Vehicle motion model.

の中心軸を考える。その様子を図 5.3 に示す。同図中でカメラは車両上の G に設置されている。ここで、 l はホイールベース、 L は前輪からカメラの設置位置までの距離、 θ は操舵角、 α は車両の回転角であるヨー角を示している。

$\Delta x = 0$ すなわち直進している場合 ($\alpha = \theta = 0$) はモデルに当てはめる必要がないため、車両の並進ベクトルは、

$$(t_x, t_y) = (\Delta x = 0, \Delta y) \quad (5.3)$$

である。以下では、 $\Delta x \neq 0$ の場合について述べる。

時刻 $t-1$ の車両上の点 G を原点とした $G-XY$ 座標系において、静止点 C を $C_{t-1,G} = (x_G, y_G)$ とする。車両の回転中心 O は $G-XY$ 座標系において、

$$O_G = \left(-\frac{l}{\tan \theta}, -l + L \right) \quad (5.4)$$

と表される。そこで、静止点 C を $O-XY$ 座標系に変換すると、

$$C_{t-1,O} = \left(\frac{l}{\tan \theta} + x_G, l - L + y_G \right) \quad (5.5)$$

となる。

次に時刻 t では、車両が点 O を中心として α だけ回転したとする。静止点 C は $O-XY$ 座標系上では移動しないが、各時刻の車両上の座標系から観測すると、その位置が変化する。そこで、時刻 t の車両上の $G'-X'Y'$ 座標系で観測される静止点 C の位置を $O-XY$ 座標系に変換する。変換された点を図 5.3 で点 C_t とし、その位置は以下のように表される。

$$\begin{aligned}
\mathbf{C}_{t,O} &= \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \frac{l}{\tan \theta} + x_G \\ l - L + y_G \end{bmatrix} \\
&= \begin{bmatrix} \left(\frac{l}{\tan \theta} + x_G\right) \cos \alpha + (l - L + y_G) \sin \alpha \\ -\left(\frac{l}{\tan \theta} + x_G\right) \sin \alpha + (l - L + y_G) \cos \alpha \end{bmatrix}.
\end{aligned} \tag{5.6}$$

車両の運動によって引き起こされる静止点 C の移動ベクトル $(\Delta x, \Delta y)$ は $\mathbf{C}_{t,O} - \mathbf{C}_{t-1,O}$ である。ここで、 α が十分に小さいものと仮定すると、 $\cos \alpha \cong 1$ 、 $\sin \alpha \cong \alpha$ が成り立つので、

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \mathbf{C}_{t,O} - \mathbf{C}_{t-1,O} \cong \begin{bmatrix} (l - L + y_G) \alpha \\ -\left(\frac{l}{\tan \theta} + x_G\right) \alpha \end{bmatrix} \tag{5.7}$$

である。これより θ と α を求めると、

$$\begin{cases} \alpha = \frac{\Delta x}{l - L + y_G} \\ \theta = \arctan \left(\frac{l}{-\frac{\Delta y}{\Delta x} (l - L + y_G) - x_G} \right) \end{cases} \tag{5.8}$$

となる。これによって、注目点 (x, y) と移動ベクトル $(\Delta x, \Delta y)$ から操舵角 θ とヨー角 α が求められる。さらに車両の並進ベクトル (t_x, t_y) は、以下の式により求められる。

$$\begin{cases} t_x = \frac{l}{\tan \theta} (\cos \alpha - 1) - (l - L) \sin \alpha \\ t_y = \frac{l}{\tan \theta} \sin \alpha + (l - L) (\cos \alpha - 1) \end{cases} \tag{5.9}$$

5.3.2 VPP 画像を利用した移動ベクトル推定と車両の運動推定

自車両の運動を求めるには、静止領域を基準としてカメラの相対的な運動を求める必要がある。2.1 節で抽出された道路平面領域は、道路面に相当するため、静止領域である。また、VPP 画像は道路平面を垂直上方から観察した画像に相当するため、VPP 画像上での車両の運動は 2次元平面内の運動として扱うことができる。そこで、隣接するフレームの VPP 画像同士を道路平面領域内に対してのみ領域ベースマッチングをし、移動ベクトルを推定する。そして、その移動ベクトルから車両の運動パラメータを求めることができる。

実際の車両では、回転半径に応じた限界の旋回速度が存在する [18]。その限界旋回速度 V_{\max} [m/s] は経験則によって、

$$V_{\max} = \sqrt{\mu R_{\text{ad}} g} \tag{5.10}$$

と表される。ただし、 μ は路面とタイヤの横滑り摩擦係数、 g は重力加速度、 R_{ad} は旋回半径である。限界旋回速度から算出される、33[ms](30[fps] の動画像の間隔) 間のヨー角 (ヨーレート) α_{\max} [rad/frame] は、

$$\alpha_{\max} = \frac{V_{\max}}{30 R_{\text{ad}}} \tag{5.11}$$

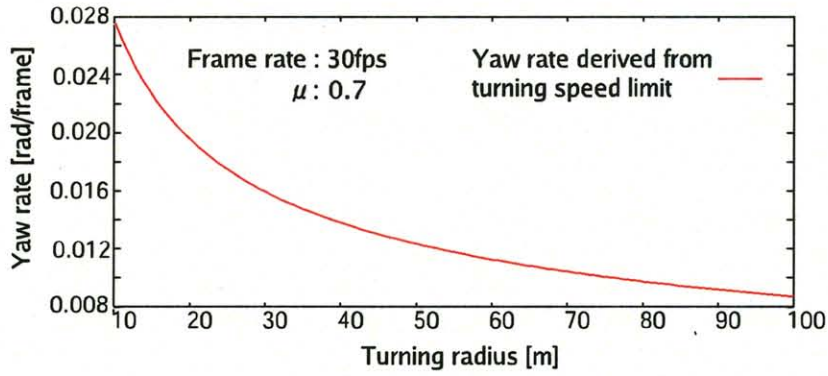


図 5.4: 旋回半径と限界旋回速度から得られるヨーレート
 Fig. 5.4: Yaw rate derived from turning speed limit.

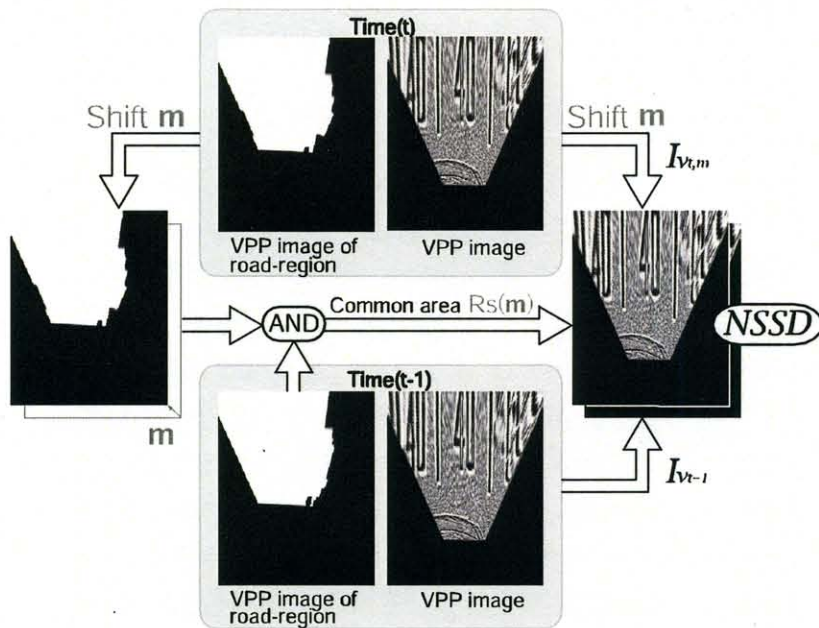


図 5.5: 移動ベクトルの推定手順
 Fig. 5.5: Shift vector estimation flow.

である。図 5.4 に、ヨーレート α_{\max} と旋回半径の関係を示す。同図から、例えば旋回半径が 10[m] でのヨーレート α_{\max} は 0.028[rad](≈ 1.6 [degree]) と読み取れ、ヨーレート α_{\max} は小さいことがわかる。旋回半径に対する最大のヨー角であっても十分に小さいため、VPP 画像に映る自車両の運動によって生じるフローを一様な平行移動と近似して、移動ベクトル $\mathbf{m} = (\Delta x, \Delta y)$ を求める。

次に、移動ベクトルの推定方法を説明する。この処理の全体図を図 5.5 に示す。まず、時刻 $t-1$ の平面領域を VPP 画像に変換して R_{t-1} とする。同様に時刻 t の平面領域も変換し、その画像を \mathbf{m} だけ平行移動させて $R_{t,m}$ とする。次いで、 $R_{t,m}$ と R_{t-1} との共通領

域 $R_s(\mathbf{m})$ を式 (5.12) に従って算出する。この共通領域は両時刻における平面領域に相当する。

$$R_s(\mathbf{m}) = R_{t,m} \cap R_{t-1}. \quad (5.12)$$

次に、時刻 $t-1$ の基準画像に LOG フィルタを施し、ヒストグラム平坦化を行なった後、VPP 画像に変換する。この VPP 画像を $I_{V_{t-1}}$ とする。時刻 t の画像に対しても同様の処理をして VPP 画像に変換した後、 \mathbf{m} だけ平行移動させて $I_{V_{t,m}}$ とする。この 2 枚の VPP 画像の \mathbf{m} に対する相違度を式 (5.13) に従って、共通領域 $R_s(\mathbf{m})$ 内について計算する。

$$NSSD(\mathbf{m}) = \frac{1}{N} \sum_{\mathbf{x} \in R_s(\mathbf{m})} [I_{V_{t,m}}(\mathbf{x}) - I_{V_{t-1}}(\mathbf{x})]^2. \quad (5.13)$$

ただし、 N は共通領域 $R_s(\mathbf{m})$ の面積を示す。この相違度が最も小さくなる (類似度が最も高くなる) 移動ベクトル \mathbf{m} を以下のように探索する。

$$\hat{\mathbf{m}} = \arg \min_{\mathbf{m} \in M_0} \{NSSD(\mathbf{m})\}. \quad (5.14)$$

ただし、 M_0 はフレーム間の移動ベクトルの探索範囲である。

最後に、求められた移動ベクトルから車両の運動を算出する。まず、3.1.2 節で述べたように平面とカメラ間距離および仮想カメラの内部パラメータから VPP 画像の 1 画素に相当する実距離を算出し、移動ベクトルの成分である $(\Delta x, \Delta y)$ の単位系を実座標系に一致させる。次に $\Delta x = 0$ なら車両は直進運動であるので式 (5.3) に、それ以外は式 (5.8) に代入する。ここで注目点 (x_G, y_G) は、共通領域の重心位置を用いる¹。

フレーム間の回転角度が十分に小さいことから、隣接するフレーム間では VPP 画像中の平面領域全体の運動が純粋な平行移動であると仮定してマッチングを行ない、移動ベクトルを推定する。そして車両の運動モデルを利用し、その移動ベクトルと重心座標から回転運動を含めた自車両の運動を推定できる。そのため画像中から特定の注目点を選んだり、そのトラッキングを行なったりする必要がない。

5.3.3 運動パラメータの更新

本節では、前節で求められた運動パラメータを初期値として、より高精度に運動を推定する方法を述べる。

車両の運動は VPP 画像上では回転運動と並進運動であるため、運動推定は 2 枚の VPP 画像間のユークリッド変換行列を求めることに相当する。ここでは文献 [70] に述べられている勾配法の考え方を利用して、本目的のために新たに定式化を行う。勾配法では適切な初期値が必要であるため、前節で求められた運動パラメータを初期値として用いる。

¹ある画像同士のマッチングをとる場合に、マッチしたときのその領域を代表する運動は、領域の重心であると考えるのが自然である。さらに 5.5.1 節の合成画像実験でその妥当性を検証している。

まず2枚の画像間の関係は、ユークリッド変換行列 M を用いて、

$$\mathbf{x}' \simeq M\mathbf{x} \quad (5.15)$$

と表される。ただし、

$$M = \begin{bmatrix} \cos \alpha & \sin \alpha & t_x \\ -\sin \alpha & \cos \alpha & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (5.16)$$

である。このとき、

$$M \leftarrow M(\mathbf{I} + \mathbf{D}) \quad (5.17)$$

として、ユークリッド変換行列 M を更新することで最適なユークリッド変換行列 M を求めるとする。ただし、 \mathbf{D} は更新量を表す行列である (後に式 (5.23) でその内容が示される)。最適なユークリッド変換行列 M は、時刻 t における画像 I_{V_t} をユークリッド変換行列 M で変換した画像 \tilde{I}_{V_t} と時刻 $t-1$ における画像 $I_{V_{t-1}}$ の相違度 E が最小になるように求められる。

まず更新量 $(\mathbf{I} + \mathbf{D})$ の求め方を説明する。 $\mathbf{x}'' = (\mathbf{I} + \mathbf{D})\mathbf{x}$ と置き、相違度 E を

$$E(\mathbf{d}) = \frac{1}{N} \sum_{i \in R_s} \left[\tilde{I}_{V_t}(\mathbf{x}''_i) - I_{V_{t-1}}(\mathbf{x}_i) \right]^2 \quad (5.18)$$

と定義する。ただし、 \mathbf{d} は \mathbf{D} の各要素を並べた9次元ベクトルである。式 (5.18) をテーラー展開することで、

$$E(\mathbf{d}) \approx \frac{1}{N} \sum_{i \in R_s} \left[\mathbf{g}_i^T \mathbf{J}_i^T \mathbf{d} + e_i \right]^2 \quad (5.19)$$

を得る。ただし、 $\mathbf{g}_i^T = \nabla \tilde{I}_{V_t}(\mathbf{x}_i)$ 、 $e_i = \tilde{I}_{V_t}(\mathbf{x}_i) - I_{V_{t-1}}(\mathbf{x}_i)$ 、 $\mathbf{J}_i = \frac{\partial \mathbf{x}''}{\partial \mathbf{d}}$ である。ユークリッド変換では、回転成分 ω と並進移動成分 (dt_x, dt_y) の更新量を考える必要がある。最初に回転成分については、

$$\mathbf{R} \leftarrow \mathbf{R}\hat{\mathbf{R}} \quad (5.20)$$

として更新することを考える。ここでRodoriguezの公式を利用することで、

$$\hat{\mathbf{R}} \approx \mathbf{I} + \mathbf{X}(\hat{n}) \sin \theta' \quad (5.21)$$

とする。ただし、

$$\theta' = \|\omega\|, \quad \hat{n} = \frac{\omega}{\theta'}, \quad \mathbf{X}(\omega) = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \quad (5.22)$$

である。次に、並進移動も考慮することで更新量 $(\mathbf{I} + \mathbf{D})$ は、

$$\mathbf{I} + \mathbf{D} = \begin{bmatrix} \mathbf{I} + \mathbf{X}(\omega) & dt \\ \mathbf{0} & 1 \end{bmatrix} \quad (5.23)$$

となる。ただし、 $dt = [dt_x, dt_y]^T$ である。次いで、式(5.19)の d を Θ に置き換えると、

$$E(\Theta) \approx \frac{1}{N} \sum_{i \in R_s} [\mathbf{g}_i^T \mathbf{J}_{\Theta_i}^T \Theta + e_i]^2 \quad (5.24)$$

となる。ただし、

$$\begin{aligned} \Theta &= [\omega, dt_x, dt_y]^T \\ \mathbf{J}_{\Theta_i} &= \frac{\partial \mathbf{x}''}{\partial \Theta} = \frac{\partial \mathbf{x}''}{\partial d} \frac{\partial d}{\partial \Theta} \\ &= \begin{bmatrix} -y & 1 & 0 \\ x & 0 & 1 \end{bmatrix}^T \end{aligned} \quad (5.25)$$

である。相違度 E が最小になるには、変数 Θ に関して式(5.24)が最小になればよい。つまり式(5.24)を Θ について偏微分したものが0と等しい。その条件から以下の式を得ることが出来る。

$$\mathbf{A}\Theta = -\mathbf{b}. \quad (5.26)$$

ただし、

$$\begin{aligned} \mathbf{A} &= \sum_{i \in R_s} \mathbf{J}_{\Theta_i} \mathbf{g}_i \mathbf{g}_i^T \mathbf{J}_{\Theta_i}^T \\ \mathbf{b} &= \sum_{i \in R_s} e_i \mathbf{J}_{\Theta_i} \mathbf{g}_i \end{aligned} \quad (5.27)$$

である。式(5.26)から Θ を求め、ユークリッド変換行列 \mathbf{M} を繰り返し計算で更新することで、最適なユークリッド変換行列 \mathbf{M} を精度よく求める。求められたユークリッド変換行列 \mathbf{M} から自車両の運動パラメータ(並進ベクトルと回転角)を算出する。

5.4 フィルタリング

先に述べた手法を利用して、より安定に推定結果を取得するために、フィルタリング処理を行なう。本研究ではカルマンフィルタ[22]を導入し、観測誤差共分散に対して運動推定によって得られた評価値を反映させる処理を行なうことで、信頼性の低い推定値を推定結果に反映しにくくする工夫を行なった。

まず、状態方程式を以下のように立てる。

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \text{noise}(\mathbf{Q}), \quad y_t = \mathbf{H}\mathbf{x}_t + \text{noise}(R), \quad (5.28)$$

ここで、

$$\mathbf{x} = [x \quad \dot{x}] \quad , \quad \mathbf{H} = [1 \quad 0] \quad , \quad \mathbf{F} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}$$

である。観測誤差共分散 R を以下のように、

$$R = \frac{K}{a} \quad (5.29)$$

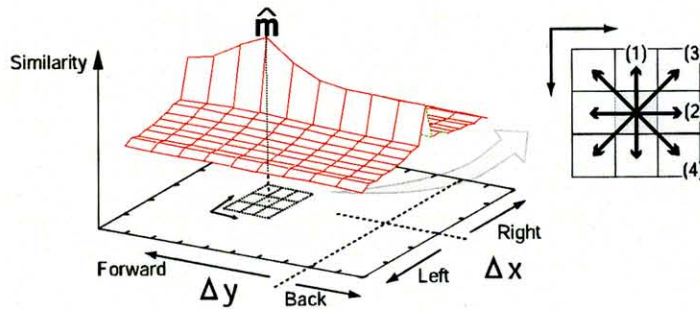


図 5.6: 類似度分布と“Roughness”の関係
 Fig. 5.6: Relation between similarity surface and “Roughness”.

とした。ここで、 a は画像の“Roughness”であり、 K は定数である。“Roughness”は、式(5.13)によって算出された類似度空間におけるパラボラの滑らかさである。分散が“Roughness”を使って式(5.29)として表せることは付録C.1で述べる。

図 5.6には、類似度空間を実際に計算した結果を示す。同図で“Roughness”は、(1)の方向のパラボラの滑らかさである。

5.5 実験結果

5.5.1 合成画像による実験

合成画像の作成には OpenGL を利用し、道路面のテクスチャは実画像から切り出して利用した。画像の解像度は 320×240 [pixel]、撮影速度は 30 [fps] と仮定した。また車両はホイールベースが 2.7 [m]、前輪軸上にカメラが搭載されており、カメラと平面間の距離は 1.2 [m] とした。そして、車両は限界旋回速度に近い一定速度で回転運動をしているとした。以下では、(1)カメラと平面の姿勢が変化する場合と(2)非平面領域が存在する場合の運動推定結果をそれぞれ示す。

シーケンス 1

カメラと平面間の姿勢が変化する場合の運動推定結果を示す。カメラはピッチ方向に 85° を中心として振幅 3° の正弦波で変動している。図 5.7 は、その運動状況における連続画像を 10 フレームおきに示している (Frame No.20 は Frame No.0 と同じなので省略)。

図 5.8 は、推定されたヨーレートを示している。5.3.1 節で述べた車両の運動モデルを使って運動推定を行なう手法を、以下ではモデルベース推定と呼ぶことにする。図中にはモデルベース推定の結果を青線で示し、5.3.3 節で述べた運動パラメータを更新した結果を赤線で示す。また、真値を緑線で示す。



図 5.7: 合成画像 1 の基準画像列 (左から Frame No.0,10,30)

Fig. 5.7: Synthetic image 1: Image sequence of right camera (left to right: Frame No. 0, 10, 30).

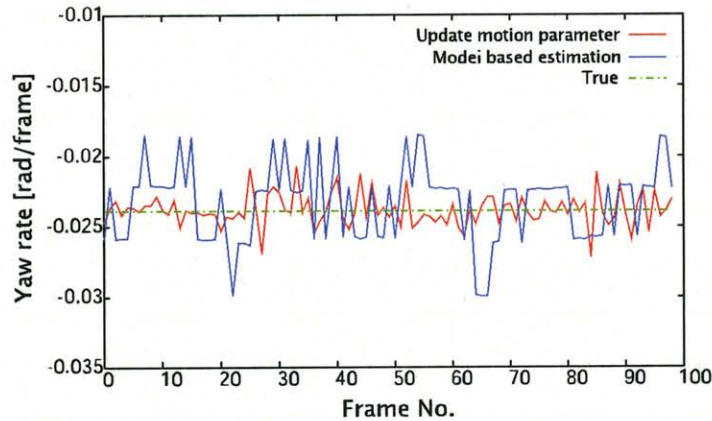


図 5.8: 合成画像 1:ヨーレート

Fig. 5.8: Synthetic image 1: Yaw rate.

図 5.9 は、図 5.10 中の r に相当する真値との誤差を示している²。0 に近いほど正確に並進ベクトルが推定されている。図 5.9 の線の色は、図 5.8 と同じである。

次に運動パラメータを利用し、移動後の車両の位置を求めた結果を図 5.11 に示す。同図は $(X, Y) = (0, 0)$ が真値であり、図 5.10 中の領域 A に示すように、真の位置からの隔たりを表している。モデルベース推定による結果では、生成した VPP 画像の解像度によって離散化された値となっている³。

図 5.8, 5.9, 5.11 からモデルベース推定でも自車両の運動が推定できているが、運動パラメータを更新することでより真値に近づき、精度が向上していることがわかる。

表 5.1 は、推定結果の RMS エラーを示している。同表から、モデルベース推定でもかなりの精度で推定できているが、運動パラメータを更新することにより、さらに RMS エラーは小さくなり、より良いことがわかる。また同表中の「平面の姿勢パラメータ」が「固定値」の場合は、カメラと道路平面間の姿勢 (平面の法線ベクトルとカメラと平面までの距離) を動的に推定せず、ピッチ角を 85° で固定した場合の結果を表している。平面

²この運動で想定した並進ベクトルは、 $|t_y|$ が 0.588[m]、 $|t_x|$ が 0.0576[m] である。

³マッチングに用いる VPP 画像は 20[pixel/m] とした。

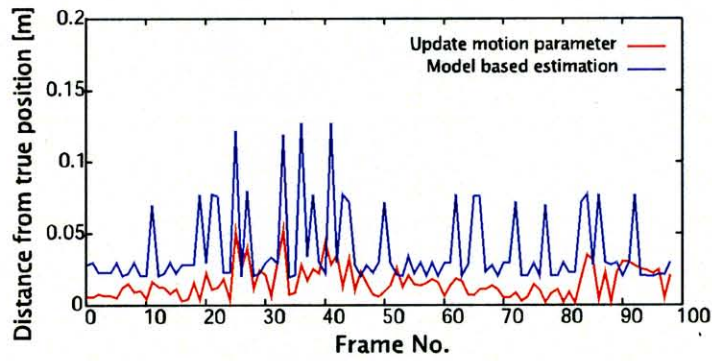


図 5.9: 合成画像 1: 並進ベクトルの推定位置の誤差
 Fig. 5.9: Synthetic image 1: Translation error.

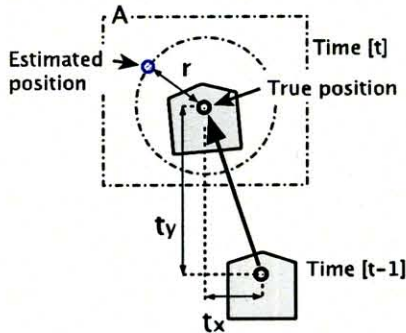


図 5.10: 車両の移動と誤差
 Fig. 5.10: Vehicle motion and error.

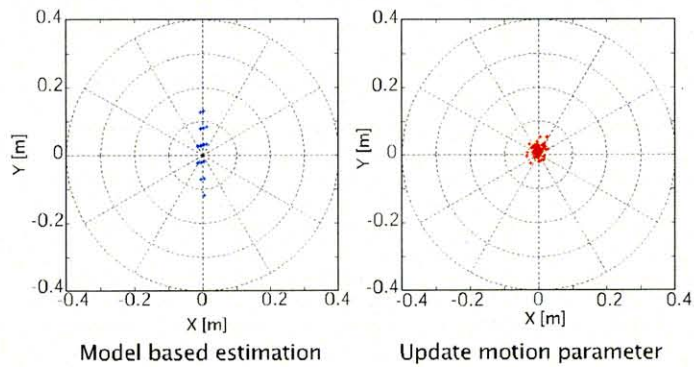


図 5.11: 合成画像 1: 移動位置推定結果
 Fig. 5.11: Synthetic image 1: Position estimation result.

の姿勢パラメータを固定すると RMS エラーが 1 桁大きくなっている。

表 5.1: 合成画像 1: 運動推定結果の RMS エラー
 Table 5.1: Synthetic image 1: RMS error of motion estimation.

平面的姿勢パラメータ	運動推定法	RMS error	
		移動速度 [m/frame]	ヨーレート [rad/frame]
推定値 ※ ₁	モデルベース	4.672e-2	2.803e-3
	パラメータ更新	1.902e-2	1.141e-3
固定値 ※ ₂	モデルベース	6.185e-1	1.492e-2

※₁ 推定値で動的に変更する

※₂ 平面の法線ベクトルを 85° に固定する

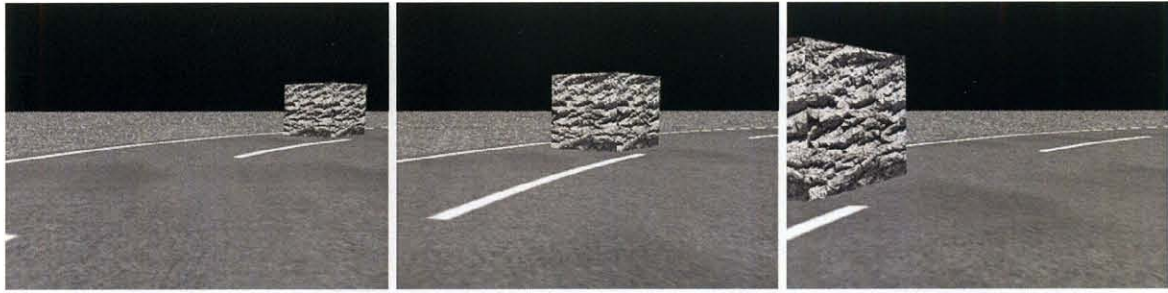


図 5.12: 合成画像 2 の基準画像列 (左から Frame No.0,10,20)

Fig. 5.12: Synthetic images 2: Image sequence of right camera (left to right: Frame No. 0, 10, 20).

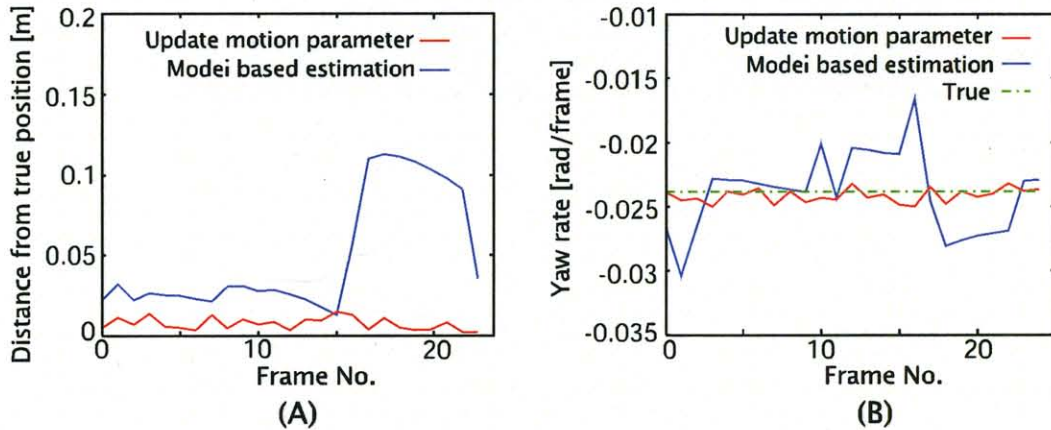


図 5.13: 合成画像 2 : 運動推定結果

Fig. 5.13: Synthetic image 2: Ego-motion estimation result.

シーケンス 2

平面上に非平面領域が存在する場合の運動推定結果を示す。図 5.12 に基準画像列の一部を示している。この実験では、カメラと道路平面間の姿勢は変動させず、平面上に静止物体を配置した。

図 5.13(A) は、並進ベクトルの推定位置の誤差を、同図 (B) はヨーレートの推定結果を示している。モデルベース推定では非平面領域が近い場合に精度が若干落ちるが、運動パラメータを更新することで真値に近い値が推定されている。これらの結果から、平面上に物体が存在する場合でも精度よく運動を推定できることがわかる。

5.5.2 実画像による実験

本節では、提案手法を実画像に対して適用した結果を示す。

実験車両にステレオカメラ (SONY XC003) をほぼ平行に配置し、撮影速度を 30 [fps]、解

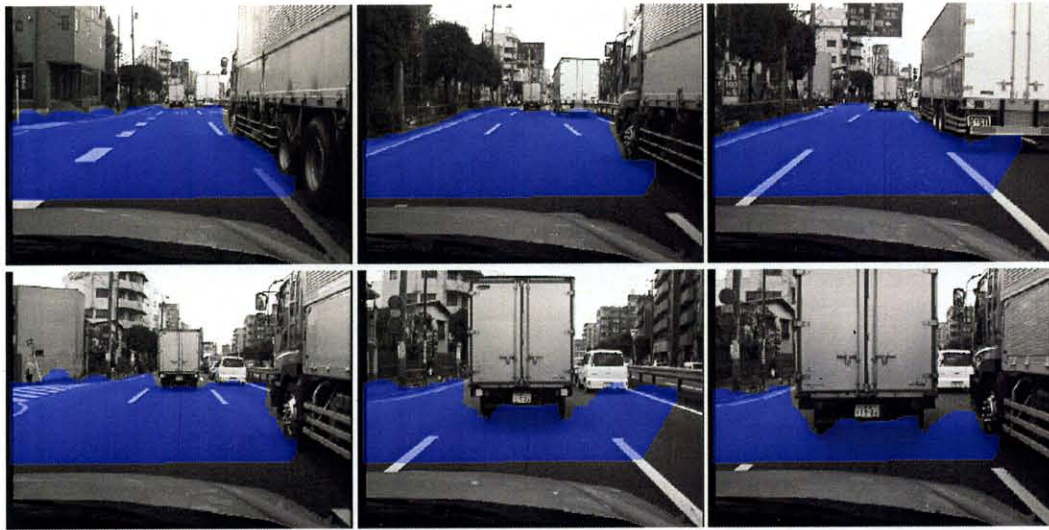


図 5.14: シーン 1 の画像列 (左上から右へ, 左下から右へ続く 80 フレームごとの連続画像). ただし, 色の明るい領域は抽出された平面領域を表す.

Fig. 5.14: Scene 1: Image sequence of right camera.

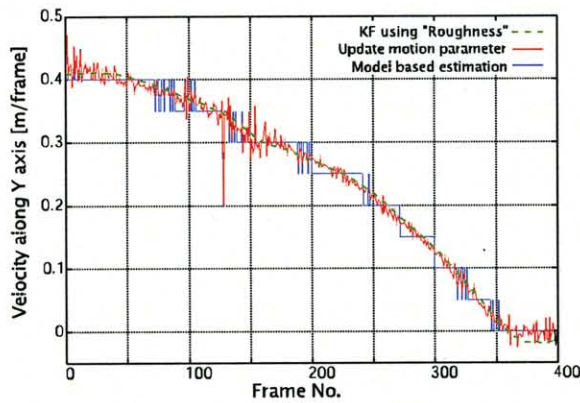


図 5.15: シーン 1: Y 軸方向速度
Fig. 5.15: Scene 1: Velocity along Y axis.

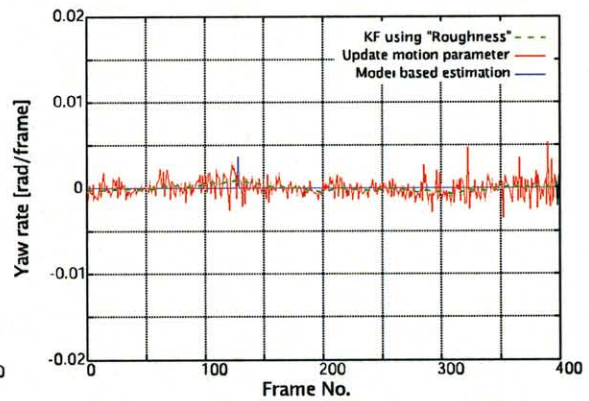


図 5.16: シーン 1: ヨーレート
Fig. 5.16: Scene 1: Yaw rate.

像度を 320×240 [pixel] として撮影した. 実験車両はホイールベースが 2.7 [m] であり, カメラは前輪から 0.85 [m] 後方に設置されている. 車両が静止した状態で, カメラと道路面との距離は約 1.2 [m], カメラの道路面に対する俯角は約 5° であり, それらは走行中に車両の振動などで変動する.

シーン 1

図 5.14 は, 実験に利用した基準画像を 80 フレームおきに示している. このシーンは, 併走する車両が自車両の周りに存在するため, 画像中に移動物体が含まれる. 自車両の運

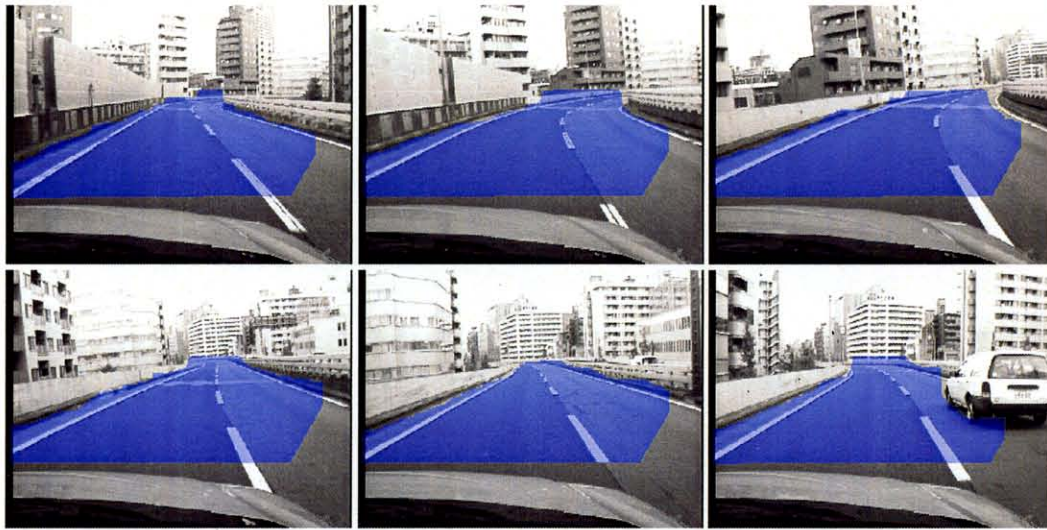


図 5.17: シーン 2 の画像列 (左上から右へ, 左下から右へ続く 70 フレームごとの連続画像)。ただし, 色の明るい領域は抽出された平面領域を表す。

Fig. 5.17: Scene 2: Image sequence of right camera.

動は前走車の停止に合わせて徐々に減速し, シーケンスの最後で停止する。

推定した移動速度 (並進ベクトルはフレーム間の移動なので速度に相当する) およびヨーレートを図 5.15, 5.16 に示す。

図 5.15 は, Y 軸 (直進) 方向の移動速度を示している。図中の青線はモデルベース推定による結果を示し, 赤線は運動パラメータを更新した結果を示している。どちらの推定結果もシーンが進むに従って, 移動速度が遅くなっている。モデルベース推定で求めた結果では移動速度が階段状になっているが, これは生成した VPP 画像の解像度が低いためである。運動パラメータを更新することで移動速度の変化がなめらかになり, 実際の現象により近いことが確認できる。また緑点線は, “Roughness” を考慮したフィルタリング結果を示している。図 5.16 は, ヨーレートを示している。この結果から, 自車両は直進していることが読みとれる。

シーン 2

図 5.17 は, 実験に利用した基準画像を 70 フレームおきに示している。このシーンでは, 自車両が緩いカーブを走行している。

図 5.18 は, Y 軸方向の移動速度の推定結果を示している。同図より, 移動速度がほぼ等速であることから, 自車両は一定速度で走行していることがわかる。

図 5.19 は, ヨーレートを示している。モデルベース推定も運動パラメータ更新による推定結果もノイズは交じるものの, ハンドルを切り始めてから直進に戻すまでの変化が読

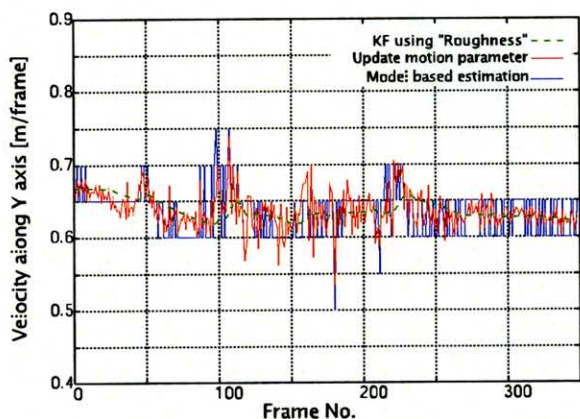


図 5.18: シーン 2: Y 軸方向速度
Fig. 5.18: Scene 2: Velocity along Y axis.

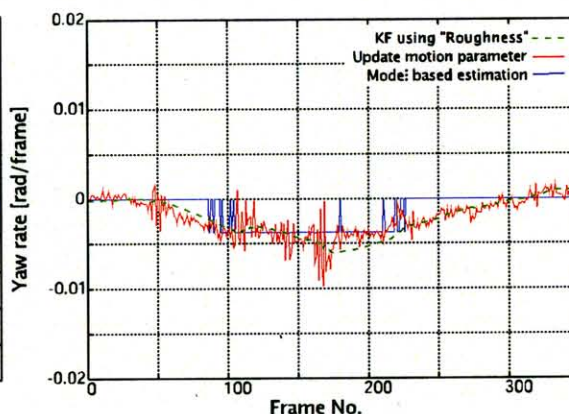


図 5.19: シーン 2: ヨーレート
Fig. 5.19: Scene 2: Yaw rate.

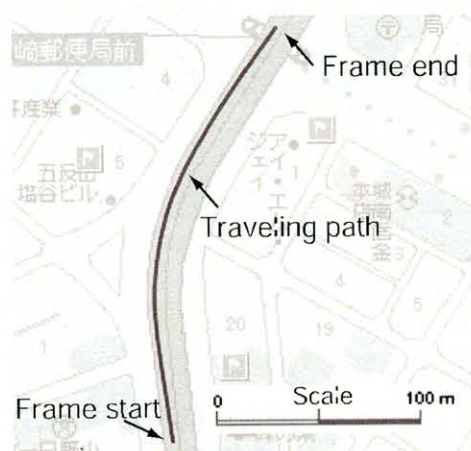


図 5.20: シーン 2: 車両の軌跡
Fig. 5.20: Scene 2: Trajectory of the vehicle.

み取れる。

図 5.20 は、各時刻で推定された運動推定結果を積算することで車両の軌跡を計算し、その軌跡を地図のスケールと位置を合わせて表示した結果である。同図から、自車両の軌跡は実際の道路の形状に沿って推定されていることがわかる。

さまざまな環境下における推定結果

図 5.21 は、さまざまな環境下で撮影された画像に対し、提案手法を適用した結果である。各画像は、その画像が撮影された時刻より後の各時刻の画像を利用して推定した運動推定結果を利用して、軌跡を描いており、左側の点は実験車両の左端、右側は右端、中の点はカメラの位置を示している。また画像上部には、推定された現在の時速とヨーレートを示している。本研究ではオフライン処理で計算を行なっているため、このような表示が

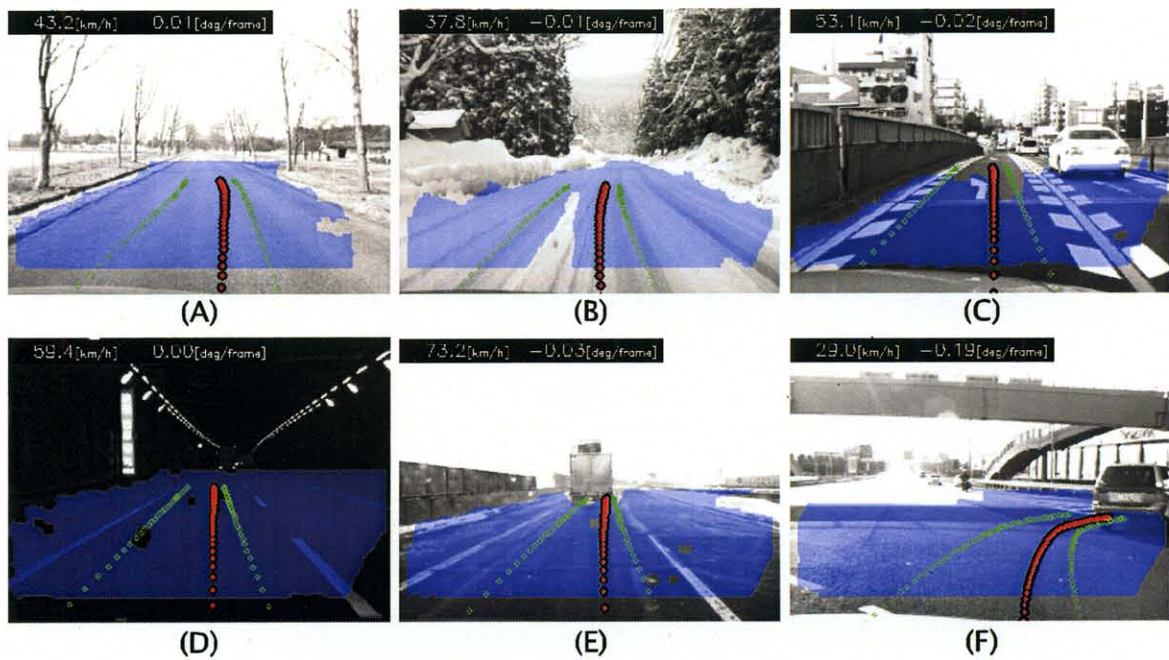


図 5.21: さまざまな環境下での運動推定結果
 Fig. 5.21: Ego-motion estimation results in various conditions.

可能である。

同図 (A) は道路白線のない道路であり, (B) は積雪した道路を走行している。これらのシーンは路面テクスチャが少なく, 運動推定が難しいと考えられるが, どちらも推定できている。(C) は下り坂を走行している。路面の湾曲のために, 道路の先方部分が検出されていないが, 手前側の平面として検出された部分のみを利用することで運動が推定されている。(D) はトンネル内を走行している。暗いシーンにおいても, 提案手法が有効であることが確かめられる。(E) は降雪中の高速道路を走行している。雪粒や前走車の水飛沫が映り込んでいても, 安定に推定されている。(F) は交差点を右折しているシーンである。推定された軌跡が交差点の形状と合っており, 正しく推定されていることがわかる。

図 5.22 は, 雪道シーンにおける連続推定結果を表している。同図より, 車両が進んでいった位置が雪溝の位置にほぼ沿っており, 推定がうまくいっていることが分かる。

5.6 本章のまとめ

本章では, 車載ステレオカメラのみを利用した自車両の運動推定手法について述べた。提案手法では, 抽出された道路平面領域によって静止領域を特定し, 推定されたカメラに対する道路面の姿勢を利用して, VPP 画像を生成した。VPP 画像は道路面を垂直上方から観察した画像に相当するため, 車両の運動を 2 次元平面内の並進と回転の運動に限定で

きた。まず、VPP 画像の道路平面内の移動ベクトルと車両の運動モデルを用いることで、特定の注目点のトラッキングを行なうことなく、簡便かつ安定に運動を推定した。次に、その結果を初期値として並進と回転だけに限定して勾配法を用いることで運動パラメータを更新し、より高精度に運動を推定した。

合成画像に対して提案手法を適用し、自車両に大きな振動が発生している場合や非平面領域が存在する場合にも精度よく運動が推定できることを確認した。また、さまざまな環境下で撮影された実画像に対しても適用し、運動が推定できることを示した。

計算時間は、Intel Pentium 4 ® 2.8GHz の汎用計算機を使い、ステレオ画像の入力から運動推定までで 0.6 [sec/frame] である。本研究では、プログラムの最適化を行っていないため、高速化を目指したプログラムを実装することで、計算時間の短縮が見込まれる。

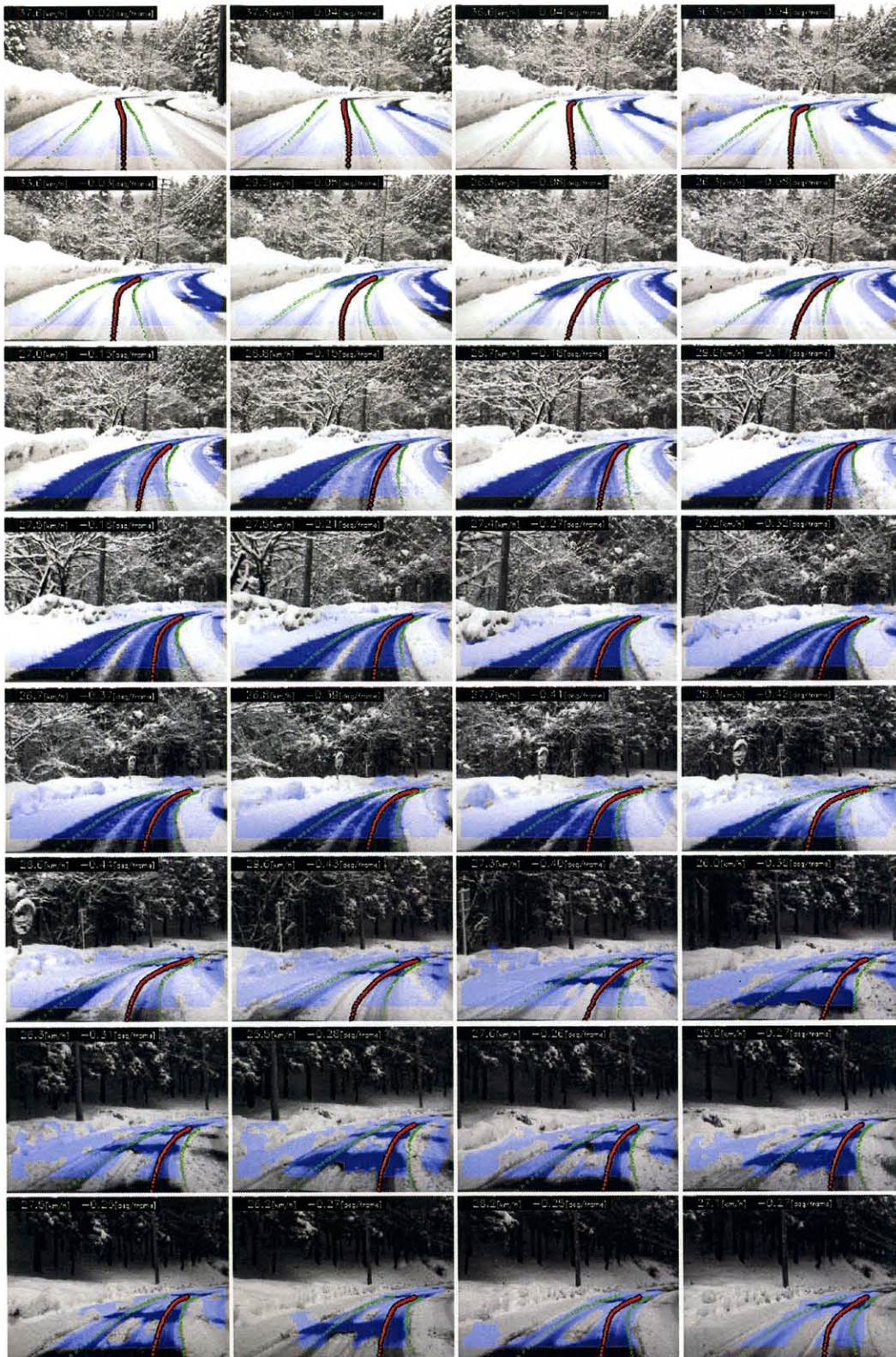


図 5.22: 運動推定結果 (10 フレームごとに左から右へ, 上から下へ表示)
 Fig. 5.22: Ego-motion estimation results. (Every 10 frame. Left to right and top to bottom.)

第6章 空間内の位置とモーションの同時推定

移動カメラの映像から、シーン中の3次元位置とその3次元モーションを取得することで、空間位置とモーションの情報を利用したセグメンテーション、エゴモーション推定、自己位置推定や視覚誘導、3次元地図の生成、さらに障害物の回避などに利用できる。

通常、ステレオ動画像を利用して3次元位置とその3次元モーションを推定するには、ステレオ対応点探索とトラッキング作業を別々に行なう必要があった。そのため、モーションによる物体の見え方の変化を考慮して処理することは困難であった。本章では、時系列ステレオ画像間のステレオ対応点探索とトラッキングに相当する作業を同時に行なうことで、画像中の各点ごとの3次元位置とその3次元モーションを同時に最適化して推定する手法を提案する。提案手法では、画像内で複数の異なるモーションが存在するようなシーンにおいても前処理なしに、各点ごとにモーションと奥行きを推定することが可能である。また、第5章で述べた自車両の運動推定と本章で述べる方法の違いは、第5章の方法は車両の運動推定に特化して精度よく頑健に推定する手法であるのに対して、本章で述べる手法はシーン中の各点の3次元位置と3次元モーションを精度よく求める方法である。

本章では、まず関連手法を述べ、提案手法の位置付けを明確にする。続いて、時系列ステレオ画像間の対応関係を述べた後、コスト関数を設計してその解法について述べる。提案手法の推定精度を合成画像を使って検証し、従来手法との比較を行なう。次に、実画像に適用した結果を示す。そして、第4章の障害物結果と第5章の自車両の運動推定結果を活用して、加えて本章の手法を用いた障害物の絶対モーション推定結果を示す。最後に、本章のまとめを述べる。

6.1 関連研究

シーンの各点の3次元位置とそのモーションを推定する問題は相互に結びついており、どちらかが既知(例えば3次元位置が既知でモーションを求める)の場合には、残りの未知パラメータを比較的容易に推定することができる[79]。しかし、一般的にどちらかが求まっていることは稀であるため、3次元位置とそのモーションの両方を推定する必要がある。

3次元位置とそのモーションの推定問題は、一般的な環境を考えると2つの場合に分けることができる。1つめは、カメラに写るモーションが単一の場合である。例えば、静止しているシーン中をカメラが移動する場合がこれに相当する。この場合には、カメラによる全体のモーションと画像内の各点ごとの3次元位置を推定する [72, 30, 15, 21, 45]。しかし、実環境では画像中に複数の異なるモーションを含むことが多いため、利用範囲が限定されてしまう。

2つめは、動的なシーン中をカメラが移動する場合であり、本論文で想定する状況である。この場合には、シーン中に様々なモーションが存在する。

Tomasiらは、剛体上の特徴点のトラッキング結果を因子分解することで、構造と3次元モーションを推定する手法(画像に写る対象のモーションから構造を推定する方法を総称して Structure from Motion:SFM という)を提案した [76]。この手法を拡張して Costeiraらは、動きの異なる複数の剛体が存在するシーン中で各物体のモーションと構造を推定する手法を提案した [9]。しかし、特徴点を物体ごとに分離する処理が必要であるため、トラッキングのノイズに弱いといった問題がある。Yuanら [89]は、平面上を運動する動物体と背景をモーションでセグメンテーションを行なって物体ごとに分離し、それぞれに関して Tomasiらの手法と同様に SFMにより、3次元モーションと構造を推定する手法を提案している。これらの手法は、単眼カメラを利用しているため、絶対的な3次元位置やモーションをすることができない。他には、Haoらは物体を領域ごとに平面と近似し、オプティカルフローとカラー情報を利用してモーションと奥行き変化に拘束を設け、奥行きとモーションを推定している [16]。この方法では、色ごとにセグメンテーションされた領域内では、モーションが同一であると仮定している。

次に、画像の各点ごとに3次元位置と3次元モーションを求める方法について考える。最も単純な方法は、ある時刻で従来のステレオ対応点探索を行なうことで3次元位置を得、次にその点を次時刻の画像上でトラッキングして時系列間で対応づける。さらに、対応づけられた次時刻の点のステレオ対応点探索を行なえば、各時刻の奥行きとモーションを求めることができる。しかし、時系列画像のトラッキングとステレオ画像間の対応を別々に行なうため、誤対応や対応誤差が生じやすく、また奥行き変化によって生じる時系列画像間のスケール倍の変化をトラッキングをする際に考慮することができない。他に、Yiらは、画像内での対応点探索をするのではなく、注目点のモーションと奥行きのパラメータを全探索し、各対応位置に対する画像間の相違度を最小にするパラメータを求めることで推定している [86]。

提案手法では、画像の各点ごとに3次元位置と3次元モーションを直接的に同時に推定する。そのため、あらかじめ同一運動する物体ごとにセグメンテーションをする必要がない。また、モーションと奥行きを同時に推定するため、奥行き変化によるスケール倍変化をトラッキングする際に考慮することが可能である。推定に用いる時系列画像は2フレー

ムから推定することができる。3フレーム以上利用する場合は、対象をカメラに対して等速直線運動と仮定することで同様に推定でき、より推定精度を上げることができる。

6.2 対象の奥行きとモーションと画像位置の関係

本章では、3次元モーションをカメラと物体の間の相対モーションとして推定する。本章の最後に、5章で述べた自車両の運動推定結果を利用して、相対モーションを静止座標を基準とした絶対モーションへ変換した結果も示す。

さて、図 6.1 に示すように奥行き Z の点 P が画像に投影される位置を $\mathbf{x} = [x, y]^T$ とし、モーション $\mathbf{T}_m = [T_x, T_y, T_z]^T$ によって変化した後には、画像に投影された位置を $\mathbf{x}' = [x', y']^T$ とすると、カメラの内部パラメータが両カメラで同じであれば、

$$\mathbf{x}' = \mathbf{x} + \frac{\mathbf{A}_x \mathbf{T}_m}{Z - T_{mz}} \quad (6.1)$$

で表される。ここで、

$$\mathbf{A}_x = \begin{bmatrix} -f & 0 & \hat{x} \\ 0 & -af & \hat{y} \end{bmatrix}, \quad \hat{x} = x - u_0, \quad \hat{y} = y - v_0 \quad (6.2)$$

である。また、カメラの内部パラメータ \mathbf{A} は、

$$\mathbf{A} = \begin{bmatrix} f & 0 & u_0 \\ 0 & af & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.3)$$

である。

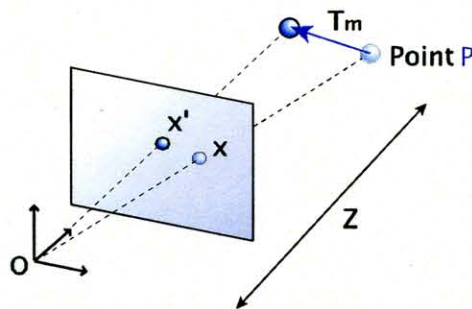


図 6.1: 対象の奥行きとモーションに対する画像座標位置の関係
Fig. 6.1: Relation between depth and motion of target and positions of image coordinate.

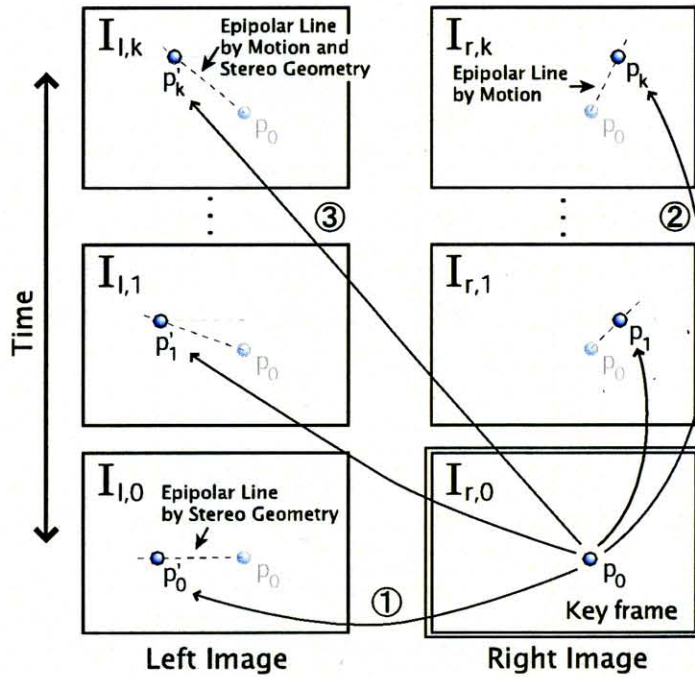


図 6.2: 時系列ステレオ画像の対応点の関係
 Fig. 6.2: Correspondence between sequential stereo images.

6.2.1 時系列ステレオ画像間の対応位置

本節では時系列ステレオ画像間の対応を考える。図 6.2 は、時系列画像の対応点の関係を示している。k 番目の左画像を $I_{l,k}$ で示し、それと同時刻の右画像を $I_{r,k}$ で示す。本章では、キーフレームを右画像の 0 番目とする (左画像をキーフレームとしても同様の議論ができる)。以降ではキーフレームの点 $p_0 = [p_{x0}, p_{y0}]^T$ を基準として、各画像との関係について述べる。

同一時刻のステレオ画像間関係

図 6.2 の①の関係である。奥行きが Z であるときの、点 p_0 が左画像に写る位置が点 p'_0 である。また、 T_s はステレオカメラ間の並進ベクトルである。この 2 点は、ステレオカメラ間のエピポーラ拘束が成り立っており、

$$p'_0 = p_0 + \frac{A_x T_s}{Z} \quad (6.4)$$

と表される。

同一カメラの画像間の関係

図 6.2 の②の関係である。k 番目の右画像中の点 \mathbf{p}_k とすると、

$$\mathbf{p}_k = \mathbf{p}_0 + \frac{k\mathbf{A}_x\mathbf{T}_m}{Z - kT_{mz}} \quad (6.5)$$

で表すことができる。右画像には、モーションに依存するエピポーラ拘束が存在する。このエピポーラ拘束は、モーションが未知であるので一意に決まっていなかったが、モーションが決まると 1~k 番目のすべての右画像のエピポーラ拘束が一意に決定する。

異なる時刻のステレオ画像間の関係

図 6.2 の③の関係である。k 番目の左画像上の点 \mathbf{p}'_k とキーフレーム上の点 \mathbf{p}_0 との関係は、

$$\mathbf{p}'_k = \mathbf{p}_0 + \frac{\mathbf{A}_x(k\mathbf{T}_m + \mathbf{T}_s)}{Z - kT_{mz}} \quad (6.6)$$

である。左画像にも、モーションとステレオカメラ間の関係で決まるエピポーラ拘束が存在し、モーションが決まると 1~k 番目のすべての左画像のエピポーラ拘束が一意に決定する。

また、それぞれの時刻のステレオ画像間にはエピポーラ拘束が成り立っているため、各時刻のステレオ画像間には、

$$\mathbf{p}'_k = \mathbf{p}_k + \frac{\mathbf{A}_x\mathbf{T}_s}{Z - kT_{mz}} \quad (6.7)$$

が常に成り立っている。

以上説明したように、モーション \mathbf{T}_m と奥行き Z が決まるとすべての画像の位置を一意に決定できる。

6.3 奥行きとモーションの同時推定

本節では、画像内の点ごとのモーションと奥行きを同時に求める方法について述べる。各画像におけるキーフレーム中の点 \mathbf{p}_0 の対応位置は、点 \mathbf{p}_0 の回りにウィンドウ領域 W を考え、全画像でその領域と最も合う位置とする。そこで、コスト関数 E を以下のように設計し、最小になるモーションと奥行きのパラメータ \mathbf{m} を求める。

$$E(\mathbf{m}) = \sum_{k=0}^n \sum_{\mathbf{x} \in W} [\mathbf{I}_{l,k}(\mathbf{x} + \mathbf{u}_k) - \mathbf{I}_{r,0}(\mathbf{x})]^2 + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [\mathbf{I}_{r,k}(\mathbf{x} + \mathbf{u}'_k) - \mathbf{I}_{r,0}(\mathbf{x})]^2 \quad (6.8)$$

ここで、 $\mathbf{m} = [T_{mx}, T_{my}, T_{mz}, Z]^T$ であり、 $\mathbf{I}(\mathbf{x})$ は $\mathbf{x} = [x, y]^T$ における画像の輝度を表す。また、 \mathbf{u}_k と \mathbf{u}'_k は式 (7.1), 式 (7.5) より、

$$\mathbf{u}_k = \frac{\mathbf{A}_x(k\mathbf{T}_m + \mathbf{T}_s)}{Z - kT_{mz}}, \quad \mathbf{u}'_k = \frac{k\mathbf{A}_x\mathbf{T}_m}{Z - kT_{mz}} \quad (6.9)$$

である。式(6.8)の前半は時系列の左右画像間の SSSD (sum of SSDs)を表し、後半は時系列の右画像間の SSSDを意味している。

6.3.1 推定アルゴリズム

本節では、式(6.8)を最小にする m の求め方を説明する。まず u_k について Δm の変化分を考えると、

$$u_k^\Delta = \frac{A_x \{k(T_m + \Delta T) + T_s\}}{Z + \Delta Z - (kT_{mz} + \Delta T_z)} \quad (6.10)$$

さて、この式の分母は以下のように表せる。

$$\frac{1}{Z + \Delta Z - (kT_{mz} + \Delta T_z)} = \frac{1}{\dot{Z}_k} + \xi \quad (6.11)$$

ここで、

$$\begin{aligned} \dot{Z}_k &= Z - kT_{mz} \\ \xi &= \frac{-(\Delta Z - k\Delta T_z)}{\dot{Z}_k^2 + \dot{Z}_k(\Delta Z - k\Delta T_z)} \end{aligned} \quad (6.12)$$

である。次に、変化量は真値に対して小さい、つまり $(Z - kT_{mz}) \gg (\Delta Z - k\Delta T_z)$ と仮定すると、

$$\xi \approx \xi' = \frac{-\Delta Z + k\Delta T_z}{\dot{Z}_k^2} \quad (6.13)$$

となる。 ξ' を使い、式(7.8)を書くと

$$u_k^\Delta = \left(\frac{1}{\dot{Z}_k} + \xi' \right) A_x \{k(T_m + \Delta T) + T_s\} \quad (6.14)$$

となる。さらに Δ が 2 乗の項は十分に小さい為無視し、平行ステレオを想定しているため、 $T_s = [T_{sx}, 0, 0]^T$ とし、式(7.12)を変形すると、

$$u_k^\Delta \approx \left(\frac{1}{\dot{Z}_k} + \frac{-\Delta Z + k\Delta T_z}{\dot{Z}_k^2} \right) \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \frac{1}{\dot{Z}_k} \begin{bmatrix} -kf\Delta T_x + k\hat{x}_0\Delta T_z \\ -ka f\Delta T_y + k\hat{y}_0\Delta T_z \end{bmatrix} \quad (6.15)$$

ここで、

$$\begin{aligned} \alpha &= -kfT_{mx} + k\hat{x}_0T_{mz} - fT_{sx} \\ \beta &= -ka fT_{my} + k\hat{y}_0T_{mz} \end{aligned} \quad (6.16)$$

である。上式をさらに変形すると、式(7.15)を得ることができる。

$$\mathbf{x} + \mathbf{u}_k^\Delta = \begin{bmatrix} \gamma & 0 & \frac{\alpha'}{\dot{Z}_k} \\ 0 & \gamma & \frac{\beta'}{\dot{Z}_k} \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \frac{\kappa}{\gamma} & 0 & \frac{\zeta}{\gamma} \\ 0 & \frac{\kappa}{\gamma} & \frac{\eta}{\gamma} \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{I} + \mathbf{D}} \tilde{\mathbf{x}} \quad (6.17)$$

ここで,

$$\begin{aligned}
\tilde{\mathbf{x}} &= (x, y, 1)^T \\
\alpha' &= -kfT_{mx} - ku_0T_{mz} - fT_{sx} \\
\beta' &= -kafT_{my} - kv_0T_{mz} \\
\gamma &= 1 + \frac{kT_{mz}}{Z_k} \\
\kappa &= 1 + \frac{k}{Z_k}(T_{mz} + \Delta T_z) + \xi'kT_{mz} \\
\zeta &= -\frac{k}{Z_k}(f\Delta T_x + u_0\Delta T_z) + \xi'\alpha' \\
\eta &= -\frac{k}{Z_k}(af\Delta T_y + v_0\Delta T_z) + \xi'\beta'
\end{aligned} \tag{6.18}$$

で表される. 次に微小な要素を持つ行列 \mathbf{D} を用い, 座標変換 $\tilde{\mathbf{x}}'' \sim (\mathbf{I} + \mathbf{D})\tilde{\mathbf{x}}$ を考える. ここで, $\tilde{\mathbf{x}}$ は \mathbf{x} の同次座標表現とし, \mathbf{I} は 3 次の単位ベクトルである. $\tilde{\mathbf{I}}_{l,k}(\mathbf{x}'')$ ¹ について座標 \mathbf{x} の周りで 1 次のテイラー展開をすると,

$$\mathbf{I}_{l,k}(\mathbf{x} + \mathbf{u}_k^\Delta) - \mathbf{I}_{r,0}(\mathbf{x}) = \tilde{\mathbf{I}}_{l,k}(\mathbf{x}'') - \mathbf{I}_{r,0}(\mathbf{x}) \approx \mathbf{g}_{l,k}^T \mathbf{J}_{l,k}^T \Delta \mathbf{m} + e_{lr,k} \tag{6.19}$$

が得られる. ここで,

$$\begin{aligned}
\mathbf{g}_{l,k}^T &= \nabla \tilde{\mathbf{I}}_{l,k}(\mathbf{x}) \\
\mathbf{J}_{l,k}^T &= \frac{\partial \mathbf{x}''}{\partial \Delta \mathbf{m}} \\
e_{lr,k} &= \tilde{\mathbf{I}}_{l,k}(\mathbf{x}) - \mathbf{I}_{r,0}(\mathbf{x})
\end{aligned} \tag{6.20}$$

である. さて, Jacobian を構成する行列は

$$\frac{\partial \mathbf{x}''}{\partial \Delta \mathbf{m}} = \frac{\partial \mathbf{d}}{\partial \Delta \mathbf{m}} \frac{\partial \mathbf{x}''}{\partial \mathbf{d}} \tag{6.21}$$

と変形できる. \mathbf{d} は, 行列 \mathbf{D} を行ごとに並べたベクトルである. 右辺の後半は,

$$\frac{\partial \mathbf{x}''}{\partial \mathbf{d}} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x^2 & -xy & -x \\ 0 & 0 & 0 & x & y & 1 & -xy & -y^2 & -y \end{bmatrix}^T \tag{6.22}$$

である [70]. 右辺の前半は, 式(7.15)の \mathbf{D} に相当する行列を $\Delta \mathbf{m}$ で偏微分することで

$$\frac{\partial \mathbf{d}}{\partial \Delta \mathbf{m}} = \begin{bmatrix} 0 & 0 & \frac{k}{Z_k} + \frac{k^2 T_{mz}}{Z_k^2} & -\frac{k T_{mz}}{Z_k^2} \\ -\frac{kf}{Z_k} & 0 & -\frac{ku_0}{Z_k} + \frac{k\alpha'}{Z_k^2} & -\frac{\alpha'}{Z_k^2} \\ 0 & 0 & \frac{k}{Z_k} + \frac{k^2 T_{mz}}{Z_k^2} & -\frac{k T_{mz}}{Z_k^2} \\ 0 & -\frac{kaf}{Z_k} & -\frac{kv_0}{Z_k} + \frac{k\beta'}{Z_k^2} & -\frac{\beta'}{Z_k^2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{6.23}$$

¹ $\tilde{\mathbf{I}}_{l,k}(\mathbf{x})$ は $\mathbf{I}_{l,k}$ を座標 $(\mathbf{x} + \mathbf{u}_k)$ でサブサンプリングして生成した画像である.

が得られる。ここで、 $\mathbf{0}$ は4次の零ベクトルを示している。これより Jacobian は

$$\frac{\partial \mathbf{x}''}{\partial \Delta \mathbf{m}} = \frac{1}{\gamma} \begin{bmatrix} -\frac{kf}{Z_k} & 0 \\ 0 & -\frac{kaf}{Z_k} \\ \mu & \nu \\ -\frac{\alpha' + kxT_{mz}}{Z_k^2} & -\frac{\beta' + kyT_{mz}}{Z_k^2} \end{bmatrix} \quad (6.24)$$

となる。ここで、

$$\begin{aligned} \mu &= \frac{k}{Z_k}(x - u_0) + \frac{k}{Z_k^2}(\alpha' + kT_{mz}x) \\ \nu &= \frac{k}{Z_k}(y - v_0) + \frac{k}{Z_k^2}(\beta' + kT_{mz}y) \end{aligned} \quad (6.25)$$

である。

\mathbf{u}'_k についても、 \mathbf{u}_k と同様に式展開をすることができる。こちらはステレオ画像間の \mathbf{T}_s を $\mathbf{0}$ と考えればよい。よって、

$$\begin{aligned} \mathbf{g}_{r,k}^T &= \nabla \tilde{\mathbf{I}}_{r,k}(\mathbf{x}) \\ \mathbf{J}_{r,k}^T &= \left. \frac{\partial \mathbf{x}''}{\partial \Delta \mathbf{m}} \right|_{T_{sz}=0} \\ e_{rr,k} &= \tilde{\mathbf{I}}_{r,k}(\mathbf{x}) - \mathbf{I}_{r,0}(\mathbf{x}) \end{aligned} \quad (6.26)$$

として、まとめると求めたいコスト関数 E は、以下のように表すことができる。

$$E(\mathbf{m} + \Delta \mathbf{m}) = \sum_{k=0}^n \sum_{\mathbf{x} \in W} [\mathbf{g}_{l,k}^T \mathbf{J}_{l,k}^T \Delta \mathbf{m} + e_{lr,k}]^2 + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [\mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \Delta \mathbf{m} + e_{rr,k}]^2 \quad (6.27)$$

さて、この式を最小化するために、コスト関数 E を $\Delta \mathbf{m}$ で偏微分して、

$$\begin{aligned} \frac{\partial E}{\partial \Delta \mathbf{m}} &= \sum_{k=0}^n \sum_{\mathbf{x} \in W} [\mathbf{J}_{l,k} \mathbf{g}_{l,k} \mathbf{g}_{l,k}^T \mathbf{J}_{l,k}^T \Delta \mathbf{m} + e_{lr,k} \mathbf{J}_{l,k} \mathbf{g}_{l,k}] \\ &+ \sum_{k=1}^n \sum_{\mathbf{x} \in W} [\mathbf{J}_{r,k} \mathbf{g}_{r,k} \mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \Delta \mathbf{m} + e_{rr,k} \mathbf{J}_{r,k} \mathbf{g}_{r,k}] = 0. \end{aligned} \quad (6.28)$$

この式を整理すると、

$$\mathbf{A} \Delta \mathbf{m} = -\mathbf{b} \quad (6.29)$$

となる。ここで、

$$\begin{aligned} \mathbf{A} &= \sum_{k=0}^n \sum_{\mathbf{x} \in W} \mathbf{J}_{l,k} \mathbf{g}_{l,k} \mathbf{g}_{l,k}^T \mathbf{J}_{l,k}^T + \sum_{k=1}^n \sum_{\mathbf{x} \in W} \mathbf{J}_{r,k} \mathbf{g}_{r,k} \mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \\ \mathbf{b} &= \sum_{k=0}^n \sum_{\mathbf{x} \in W} e_{lr,k} \mathbf{J}_{l,k} \mathbf{g}_{l,k} + \sum_{k=1}^n \sum_{\mathbf{x} \in W} e_{rr,k} \mathbf{J}_{r,k} \mathbf{g}_{r,k} \end{aligned} \quad (6.30)$$

と表される。コスト関数を最小にするためには式(6.29)を繰り返し計算する。その際のパラメータの更新は

$$\mathbf{m} \leftarrow \mathbf{m} + \Delta \mathbf{m} \quad (6.31)$$

である。繰り返し計算後に得られたパラメータ \mathbf{m} が、推定されたモーシヨンと奥行きである。また、奥行き Z が求められたとき、3次元位置は、

$$[X, Y, Z] = \left[\frac{(x - x_0)Z}{f}, \frac{(y - y_0)Z}{af}, Z \right] \quad (6.32)$$

で求めることができる。

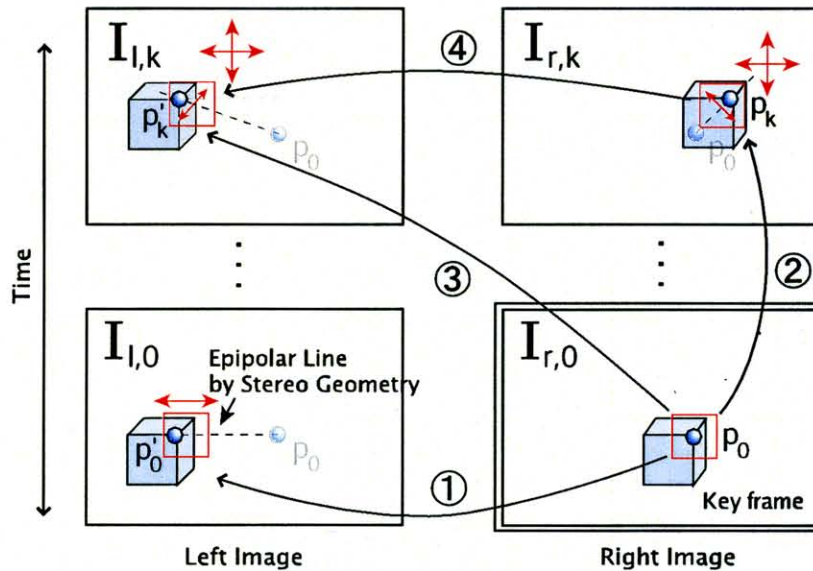


図 6.3: パラメータ推定の過程と画像間の関係

Fig. 6.3: Parameters estimation process and correspondence between sequential stereo images.

6.3.2 パラメータ推定の過程と画像間の関係

式(6.8)のコスト関数を先に述べた方法によって、奥行きとモーションを推定していく。その過程で図 6.3 に示すように、同時刻のステレオ画像間①では、ステレオカメラ間のエピポーラ線上を平行移動している。さらに、時刻の異なる同一カメラの画像間②と時刻の異なるステレオ画像間③では、モーションと奥行きで決まる拘束直線(モーションと奥行きな未決定なため、未定なエピポーラ線)上を移動し、時刻間での奥行き変化に応じてウィンドウがスケール倍変化し、かつ④の関係によって、それらはステレオカメラ間のエピポーラ線上にある。これらの条件を満たしつつ、コスト関数の小さくなる方向を求めることで、式(6.8)を最小化している。時系列ステレオ画像を 3 ペア以上利用して推定する場合も、これらの関係がすべてのペア間で成り立っており、幾何学的な矛盾が発生しない。

6.3.3 Implementation

図 6.4 は、これまでに述べた推定アルゴリズムを簡単にまとめたものである。ある点を選び、同図の流れに従って未知パラメータである奥行きと 3 次元モーションを同時に最適化する。また、複数点の推定したい場合は、この手順を点の数だけ行えば良い。

以下は、実験を行なうに当たって、参考となるとと思われる事項である。

```

Set target point.
for (iter=0 to iter max or convergence condition)
  if iter is 0
    Set initial value to m.
  end if.
  for (k=0 to max point)
    if k is not 0
      Warp image  $I_{r,k}$  and get  $\tilde{I}_{r,k}$  using m.
      Calculate  $g_{r,k}$ ,  $J_{r,k}$ , and  $e_{rr,k}$  in eq.(6.26).
    end if.
    Warp image  $I_{l,k}$  and get  $\tilde{I}_{l,k}$  using m.
    Calculate  $g_{l,k}$ ,  $J_{l,k}$ , and  $e_{lr,k}$  in eq.(6.20).
  end for loop.
  Calculate A and b in eq.(6.30).
  Calculate  $\Delta m$  using eq.(6.29).
  Update m in eq.(6.31).
end for loop.
3D MOTION and DEPTH parameters are given by m.

```

図 6.4: 奥行きと 3 次元モーション推定アルゴリズムの計算手順
 Fig. 6.4: Algorithm flow chart of estimation of depth and 3D motion.

ガウシアンピラミッドの利用

式 (6.29) の繰り返し計算回数の削減と初期値に対してロバストにするため、ガウシアンピラミッドを利用した。原画像より 1 つ上層の画像は画像サイズが 1/2 であり、原画像のカメラの内部パラメータを式 (7.3) とすると、

$$A_{pyr} = \begin{bmatrix} -\frac{f}{2} & 0 & \frac{u_0}{2} \\ 0 & -\frac{af}{2} & \frac{v_0}{2} \\ 0 & 0 & 1 \end{bmatrix} \quad (6.33)$$

となる。モーションや奥行きは空間座標であるため、内部パラメータを式 (6.33) のように変換することで、画像サイズによらず、推定パラメータを一致させることができる。

ステレオカメラの関係

提案手法では、式展開を簡単にするために平行ステレオを想定しているが、カメラ間の外部パラメータがキャリブレーションされていれば、実際のカメラが平行配置でなくても、Rectificationを行なうことで仮想的に平行ステレオにすることができる。また、カメラの内部パラメータもキャリブレーションされていれば、Rectificationによって両カメラ

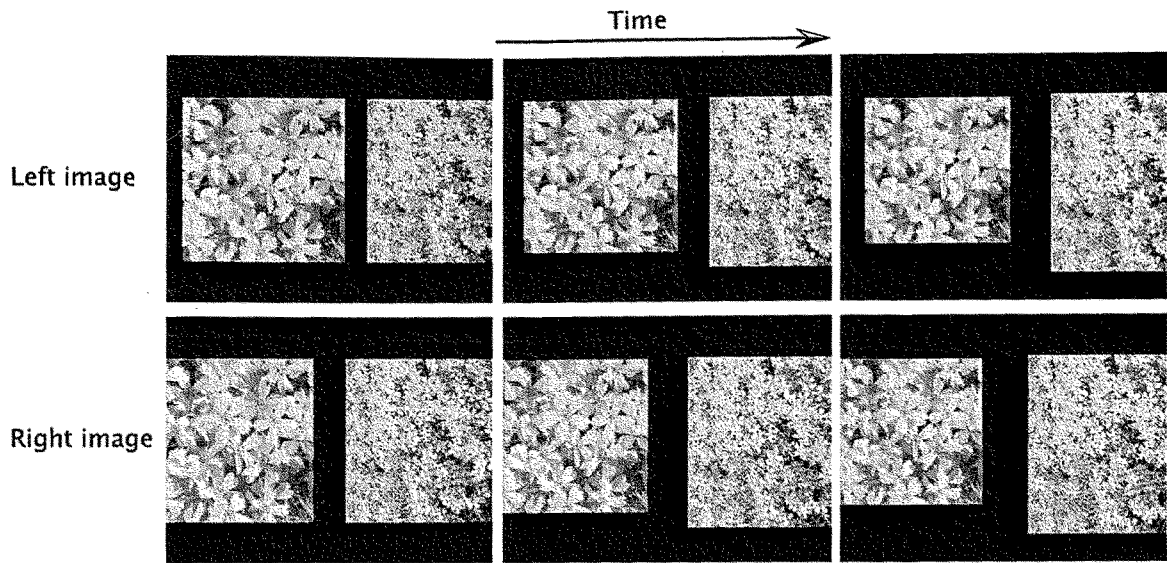


図 6.5: 入力した時系列ステレオ画像
Fig. 6.5: Input sequential images.

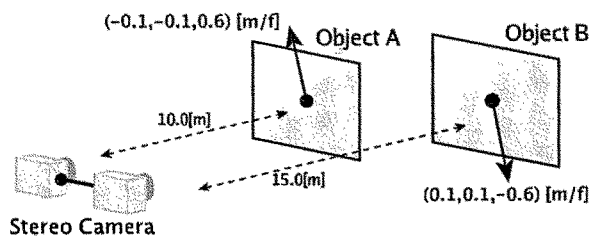


図 6.6: カメラやオブジェクトの配置とそのモーショ
Fig. 6.6: Layout of cameras, objects, and its 3D motion.

で一致させることができるため、両カメラで内部パラメータを物理的に一致させる必要はない。よってステレオカメラが平行ステレオでない場合や内部パラメータが一致していないときは、前処理として Rectification を行なえばよい。Rectification については、付録 B で述べる。

6.4 実験結果

6.4.1 合成画像実験 1

図 6.5 は、合成画像実験 1 で用いた時系列ステレオ画像を示している。本実験では、同図に示す 3 組の時系列ステレオ画像 (計 6 枚の画像) を利用した。画像には図 6.6 に示すように、奥行きも運動も異なる 2 枚の平面オブジェクトが写っている。ステレオカメラは、解像度が 320×240 [pixel] のモノクロ画像とし、ステレオカメラのベースライン長は 0.77 [m]

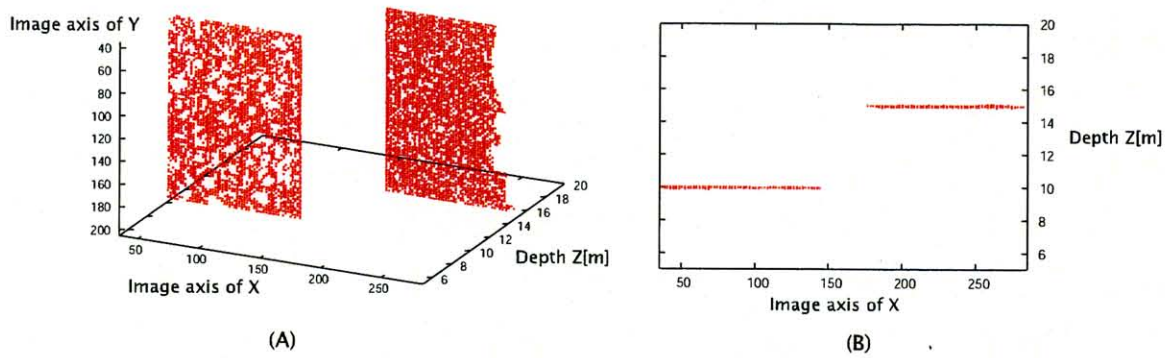


図 6.7: 奥行き推定結果
Fig. 6.7: Estimated depth.

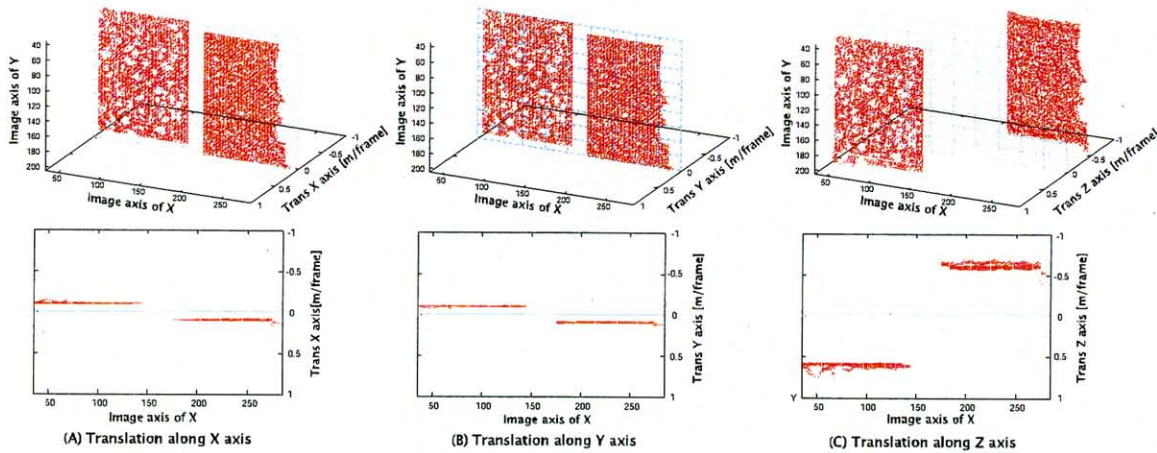


図 6.8: モーション推定結果 (下段は、上段の結果を真上から見た様子)
Fig. 6.8: Estimated motion.

とした。

計算に利用した点は、キーフレームにソーベルフィルタを施し、一定の強度を持ったピクセルとした。初期値は、画像内で一様に奥行きを 12[m] とし、モーションを XYZ 軸すべてを 0 とした。

図 6.7(A) は、奥行き推定結果を斜め方向から見た様子であり、同図 (B) は画像の Y 軸に沿って見た様子である。同図よりそれぞれの平面オブジェクトの奥行きが 10[m] と 15[m] の位置に推定されており、2つの平面オブジェクトにそれぞれ対応して正しく推定されていることが分かる。

次に図 6.8 は、推定された 3次元モーションを示している。同図 (A) は、X 軸方向 (横方向) のモーション推定結果である。下段は、推定結果を画像の Y 軸に沿って見た様子である。同図 (B) は Y 軸方向 (高さ方向) のモーション、同図 (C) は Z 軸方向 (奥行き方向) のモーションをそれぞれ示している。これらの結果より、それぞれの平面オブジェクトごと

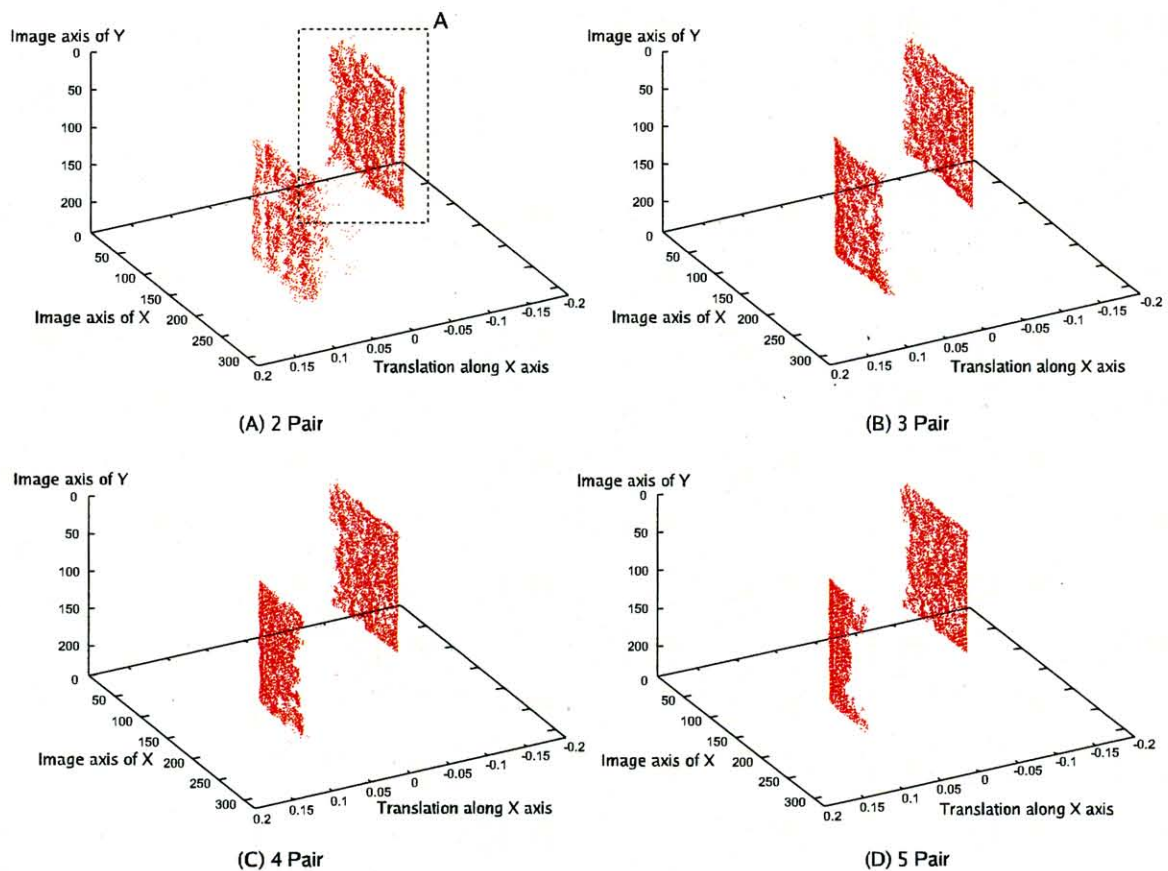


図 6.9: 画像利用ペア数に対する X 軸方向のモーション推定結果
 Fig. 6.9: Translation along X axis on image pair number by proposed method.

に推定値が一定となっており、精度よく求められていることがわかる。例えば、X 軸方向の運動は、真値と同じ ± 0.1 [m/frame] 付近に推定されている。

利用した時系列ステレオ画像のペア数と推定精度について

処理に利用する時系列ステレオ画像のペア数と推定精度の関係について、実験を通じて検証した結果を述べる。実験には、図 6.5 の画像を利用し、それぞれの画像に標準偏差 5 のガウスノイズを加えた。オブジェクトのモーションや位置関係は先の実験と全く同じである。

図 6.9 は、利用する時系列ステレオ画像のペア数を変えた時の X 軸方向のモーションの推定結果である。それぞれの平面オブジェクトは、並進運動であるのでオブジェクトごとにモーションは一定値である。そのため、精度よく推定されるほど、推定値が凹凸の少ない平面としてグラフに現れる。同図より、2 ペア (計 4 枚の時系列ステレオ画像) の推定結

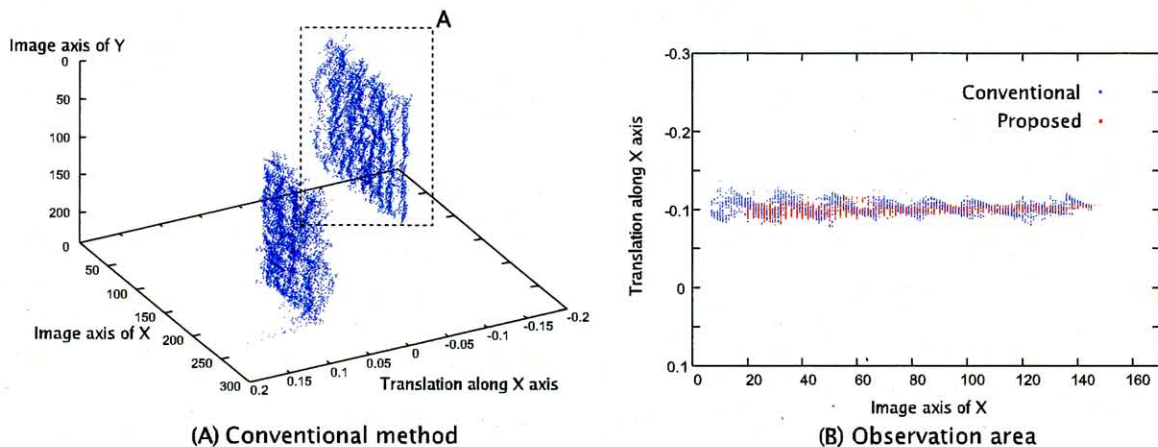


図 6.10: 従来手法による X 軸方向のモーション推定結果
 Fig. 6.10: Translation along X axis by conventional method.

果より、3ペア(計6枚の時系列ステレオ画像)の推定結果の方が推定値の集まる面の凹凸が少なくなり、推定精度が良いことが分かる。5ペアにすると、より精度が良くなっていることも分かる。また、ペア数が多い結果では、手前のオブジェクトの推定結果が少なくなっているが、これはオブジェクトの右側がモーションによって画像外へ出てしまったためである。

図 6.10(A)は、従来法によって X 軸方向のモーションを推定した結果であり、図 6.9 と同じ方向から観察している。従来法は、画像間のマッチングをすべて別々に取ることで推定した。すなわち、ステレオ画像間で対応点探索にウィンドウベースのマッチングを行い、次にその点に関して同じカメラの時系列画像間でトラッキングを行い、最後に、トラッキングされた点について、その時刻のステレオ画像間で対応点を探索することで、各時刻の同一点に関する 3 次元位置同士の差分から、モーションと奥行きを求めた。次に、同図(B)は同図(A)の領域 A(図 6.9 も同じ)について、画像の縦軸と平行になるように観察した様子である。従来手法を青点で、提案手法を赤点で示している。従来手法では、2ペア4枚の時系列ステレオ画像で推定しているため、提案手法についても2ペアのものとした。従来手法では、ピクセルロッキング現象[67]が起こっており、一定となるはずの推定結果が波状になってしまっている。それに対して、提案手法では従来法より推定のばらつきが小さく、真値である -0.1 に近いことがわかる。同じ枚数のステレオ画像ペアを利用しても提案手法の方が精度がよく、さらに提案手法では、ステレオ画像ペア数を増やすことが容易であり、それによってさらに精度を向上させることができる。

次に、図 6.11 は、推定に利用した時系列ステレオ画像ペア数に対する推定結果の RMS エラーである。RMS エラーを計算した領域は、図 6.6 に示すオブジェクト A と B についてである。これらのオブジェクトは、奥行きとモーションがそれぞれ異なっている。提案手法は、ペア数を変えても同じ処理の枠組で推定できるのに対して、従来手法では、2ペア

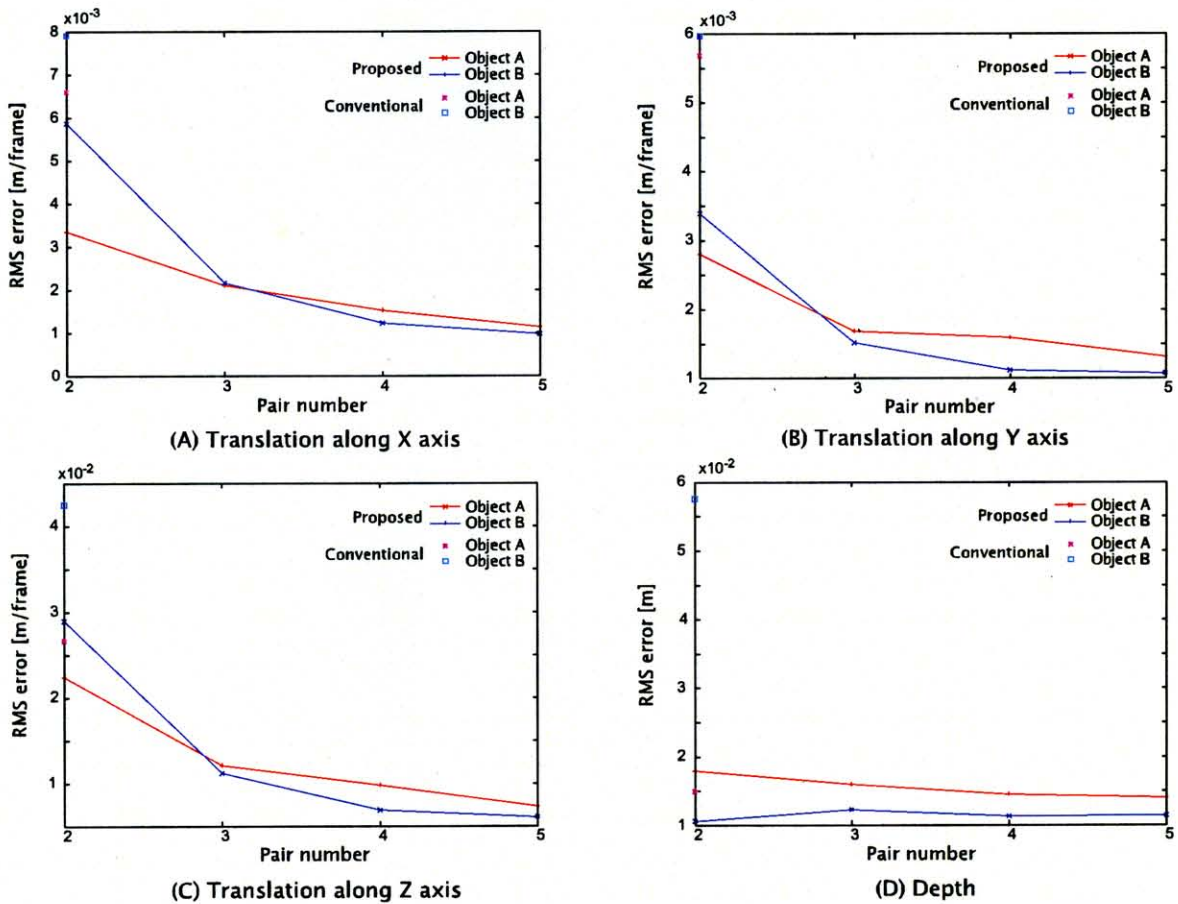


図 6.11: 利用した時系列ステレオ画像のペア数に対する RMS エラー
 Fig. 6.11: RMS error on stereo image pair number.

以上に対しては別の枠組が必要になるため、2ペアについてのみの結果を示す。同図(A)は、X軸方向のモーショントのRMSエラーであり、(B)はY軸、(C)はZ軸をそれぞれ表している。(D)は、奥行き推定結果のRMSエラーである。まず、提案手法のモーショントの推定エラーに関しては、赤青領域ともにペア数を増やすとRMSエラーが減少していることが分かる。つまり、時系列ステレオ画像ペア数を多く利用すれば、より精度良い結果が得られることが分かる。次に、提案手法の奥行き推定エラーに関しては、赤領域ではペア数の増加に伴って減少し、青領域では増減している。しかし、奥行きについては増減幅が小さいため、画像ペア数と推定精度についての関係性はそれほどないことが分かる。全体的に従来手法に比べて提案手法の方が推定エラーが大幅に小さいことが確認された。

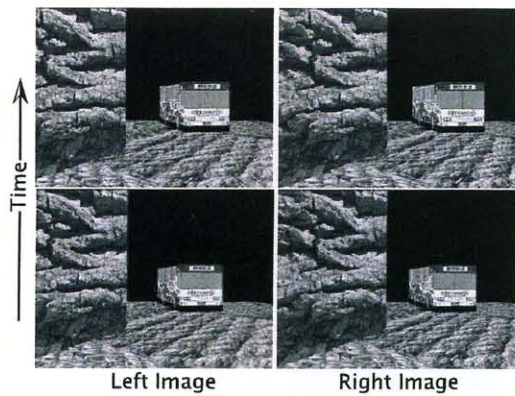


図 6.12: 入力画像例 (右カメラの画像)
Fig. 6.12: Example of input sequential images.

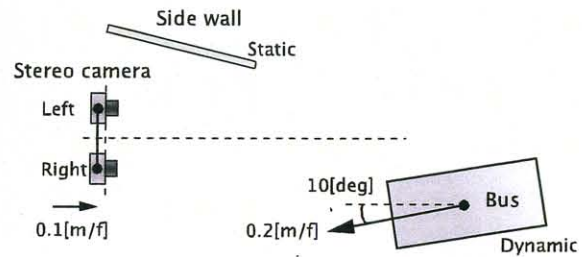
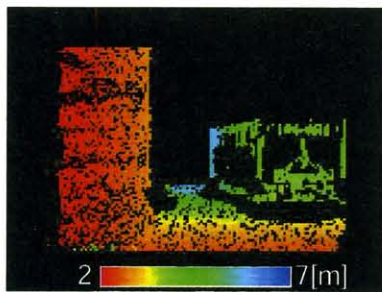
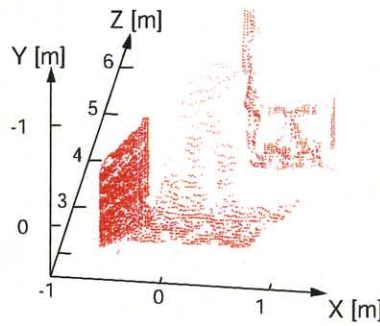


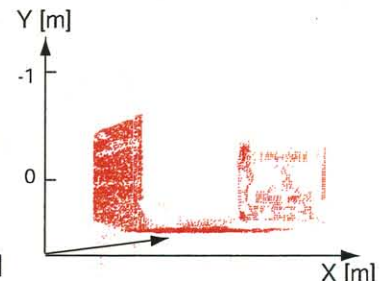
図 6.13: 想定環境の平面図
Fig. 6.13: Plane view of assumption-environment.



(A)



(B)



(C)

図 6.14: 3次元位置推定結果
Fig. 6.14: Estimated depth.

6.4.2 合成画像実験 2

図 6.12 は、本実験で利用したステレオ動画の一部であり、図 6.13 は想定環境の平面図である。シーン中には床面と側壁が静止しており、バスが移動している。ステレオカメラは、平面に対してベースライン (ベースライン長は $0.15[\text{m}]$) を水平にし、カメラ縦軸を平面に対して垂直に配置した。このシーン中でのモーションは 2 つあり、1 つはカメラが平面上を平行に奥行き方向に $0.1[\text{m}/\text{frame}]$ で移動している。もう 1 つはバスがカメラと正対から $10[\text{degree}]$ 傾いて $0.2[\text{m}/\text{frame}]$ で移動している。つまり画像中に存在するモーションは、カメラのモーションが加わったバスの相対モーションとカメラモーションによる側壁と床面のモーションである。よって、静止している側壁と床面は相対速度として、X 軸 (横) 方向と Y 軸 (高さ) 方向は $0.0[\text{m}/\text{frame}]$ で Z 軸 (奥行き) 方向に $-0.1[\text{m}/\text{frame}]$ の速度を持つ。バスは、相対速度として X 軸方向に $0.2 \times \sin(10[\text{deg}]) \approx 0.035[\text{m}/\text{frame}]$ 、Y 軸方向に $0[\text{m}/\text{frame}]$ 、Z 軸方向に $-0.2 \times \cos(10[\text{deg}]) - 0.1 \approx -0.297[\text{m}/\text{frame}]$ である。本実験では、画像サイズが $320 \times 240[\text{pixel}]$ である、ステレオ動画を 4 ペア利用した。計

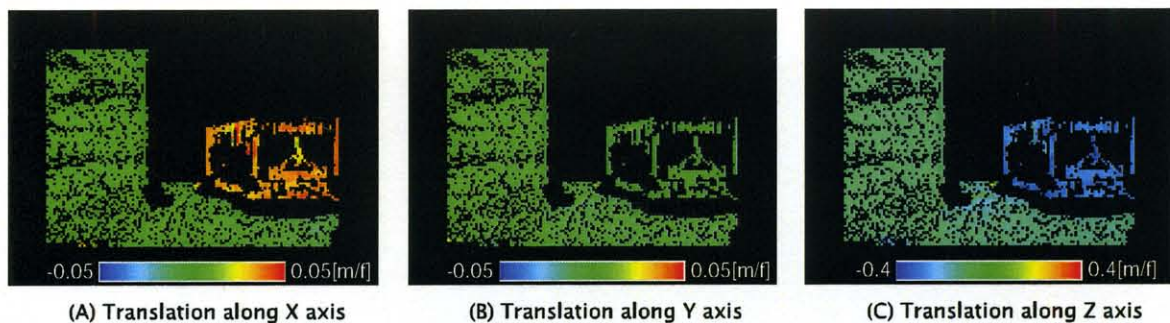


図 6.15: モーション推定結果
Fig. 6.15: Estimated motion.

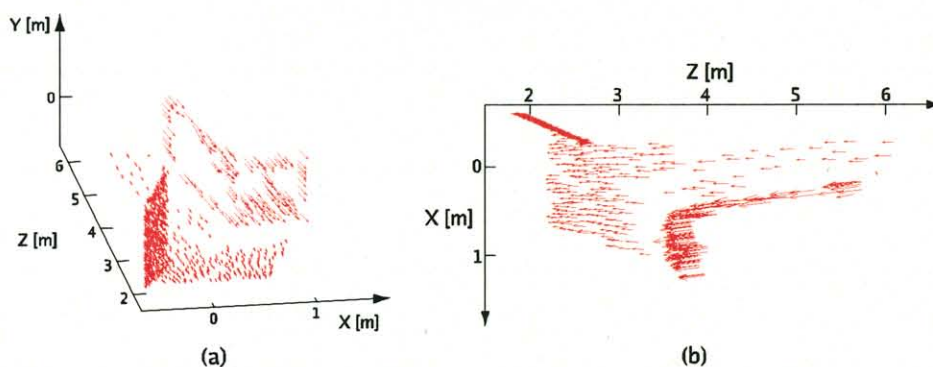


図 6.16: 3次元位置にそのモーションを併せて表示した結果
Fig. 6.16: Estimated structure and motion.

算に用いた初期値は、奥行きを一様に 3.7[m] とモーションは XYZ 軸ともに 0 とした。

図 6.14(A) は、推定された奥行きに応じて色を使って表現した結果であり、カメラから近い方を赤く、遠い方を青くなるようにしている。同図 (B) は、推定された 3 次元復元結果を斜め上から観測した結果である。同図 (C) は、ほぼ正面から 3 次元復元結果を観測した結果である。左側の側壁や床面がほぼ平面上に推定されていることや、バスが直方体で推定されていることから、正しく 3 次元復元されていることが分かる。

図 6.15 は、推定されたモーションを色を使って表現している。まず同図 (A) は、X 軸のモーション推定結果であり、赤色は 0.05、青色は -0.05 [m/frame] のモーションである。側壁と床面は中間の緑色であり、X 軸に関しては移動していないことが分かる。また、バスの領域だけ橙色となっており、正方向のモーションがあることが分かる。次に、同図 (B) の Y 軸方向のモーションに関しても同じように見ると、側壁、床面とバスともに緑色となり、 0 [m/frame] 付近である。また同図 (C) より Z 軸は静止座標系に存在する側壁と床面は -0.1 [m/frame] 付近で、バスが -0.3 [m/frame] 付近に推定されている。

続いて、図 6.16 は、推定された 3 次元位置に推定された 3 次元モーションを併せて表示した結果である。矢印の方向は、モーションの方向を、長さはモーションの大きさをそれ

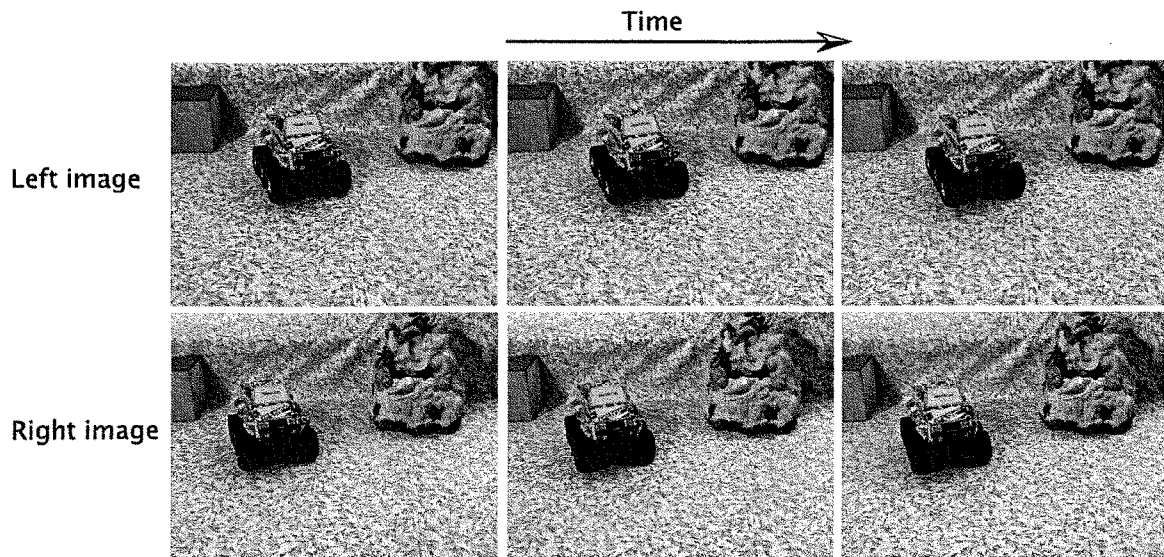


図 6.17: 実画像実験の入力画像例
Fig. 6.17: Input stereo images.

ぞれ表している。同図 (a) は、斜め上から観察した様子であり、(b) は真上からの様子である。これらの結果から、バス上の点と壁や床のモーシオンが矢印の大きさや方向が異なっていることが分かる。以上の結果より、3次元位置とモーシオンが正確に推定されていることが分かる。

6.4.3 実画像実験

実画像実験では、ステレオカメラ (ベースライン長は約 7.6[cm]) を平行ステージに設置し、ステージを上方向に動かしつつ、模型自動車が遠方から向かってくる状況とした。図 6.17 は、実験に利用した 3 組の時系列画像である。このシーンにおいて、静止系である背景領域はカメラの運動による相対モーシオンとして観測され、模型自動車領域ではカメラと模型自動車の相対モーシオンとして観測される。実験に用いた初期値は、奥行きを 0.7[m]、3次元モーシオンを XYZ 軸すべてについて、 $1.0e^{-3}$ [m/frame] として一様とした。

図 6.18 は、3次元モーシオンをそれぞれ色を付けて原画像に表示した結果である。各軸のモーシオン推定結果において、模型自動車の部分が静止領域である背景と異なって推定されていることが分かる。また、(B) の Y 軸方向のモーシオンに関しては、下段に横軸を画像の横軸、縦軸をモーシオンとして画像内のすべてのモーシオンをプロットしたグラフである。このグラフを見ると、カメラが上下運動をしているため、静止領域に関してもモーシオンが 0 でないことが分かる。また、本実験では XYZ 軸を Rectification 後の軸としたため、Y 軸が完全には上下方向の軸と一致しておらず、静止領域のモーシオンが一定

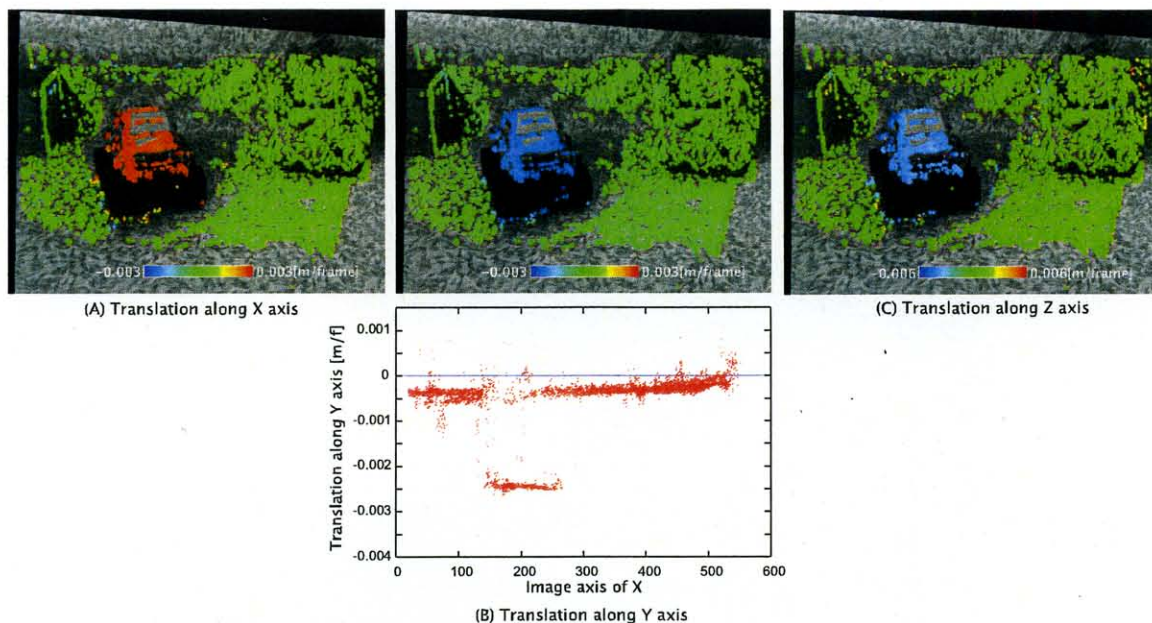


図 6.18: 実画像実験のモーシヨン推定結果
 Fig. 6.18: Estimated depth and 3D motion.

値とはならない。

次に図 6.19 は、推定された奥行きに基づいて 3 次元位置を求め、テクスチャとその位置における推定された 3 次元モーシヨンを矢印の方向と大きさを使って表している。推定された床面が平面になっていること、模型自動車や右側の置物が周りより盛り上がっており、正しく推定できていることが分かる。また、モーシヨンは黄色の矢印で表現されているが、模型自動車の部分で長い矢印が密集していることが分かる。

続いて図 6.20 では、同図の左側に示す A と B の領域の拡大図を中央と右側に示している。中央の図では、模型自動車部分である。模型自動車は、空間中をほとんど等速直線運動をしているため、モーシヨンは各点において向きもその大きさもほとんど等しい。さらに、同図で表示されている矢印の長さや向きが揃っていることから、推定値は実際の現象に一致していることが分かる。次に、同図右側は背景である床面と置物部分を拡大している。この部分に関しては、カメラの移動による微小な相対モーシヨンしか含まれていないため、表示されている矢印も小さい。

6.5 障害物のモーシヨン推定

第 4 章で述べた障害物検出手法を利用して検出された障害物に対して、本章で述べた提案手法を適用し、そのモーシヨン推定を行う。それによって、カメラに対する障害物の相対的な 3 次元モーシヨンを推定することができる。しかし、障害物の相対モーシヨンだけ

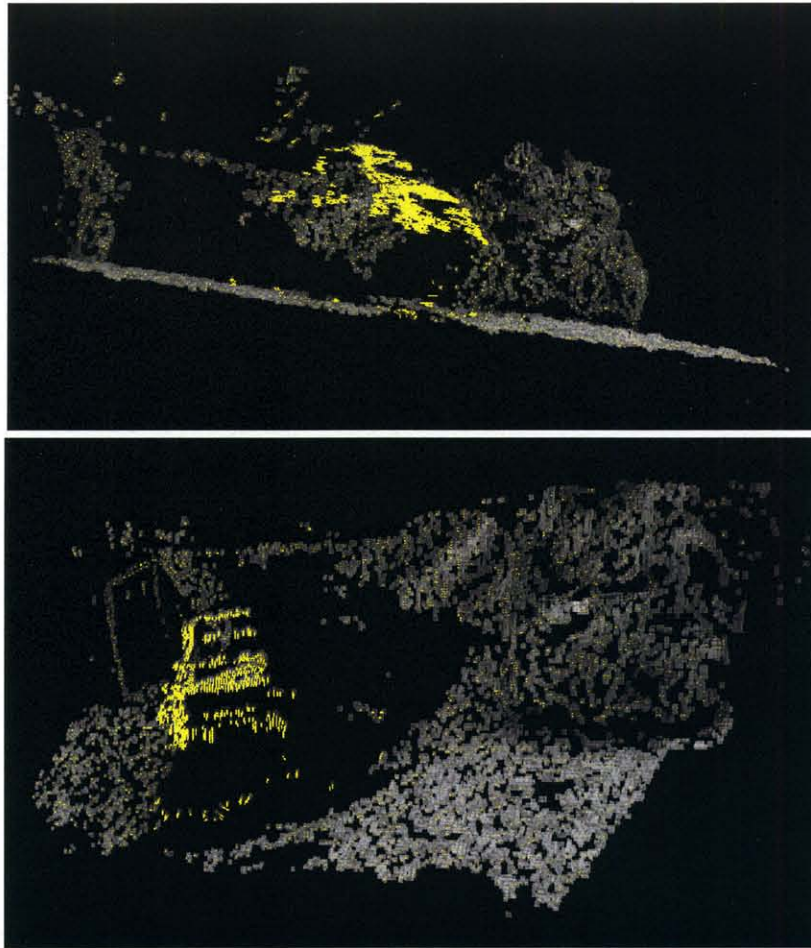


図 6.19: さまざまな視点から見た 3次元位置とモーシオン推定結果
 Fig. 6.19: Estimated 3D position and 3D motion viewed from multiple viewpoints.

では、その障害物が移動体なのかといった運動属性が分からない。そこで、第5章で述べた自車両のモーシオン推定を利用して、相対モーシオンを静止座標系を基準にした絶対モーシオンに変換する。障害物の絶対モーシオンを推定することにより、障害物が動物体なのか静止物体なのか判断することが可能となる。それによって、衝突回避や安全性の評価に利用ができると考えられる。

また、第4章で述べた障害物検出手法は、障害物を分離する方法として、障害物上の点の空間位置を特徴量としてセグメンテーションを行っており、空間位置が互いに近接していると複数の物体を誤って単一物体としてしまう可能性がある。そこで、障害物上の各点の3次元モーシオンを利用して、セグメンテーションの際にその情報も入れることで、モーシオンと位置を特徴量とすることができる。本研究では、空間とモーシオンを利用したセグメンテーションのアルゴリズムの提案および実装を施していないため、今後の課題として挙げておきたい。

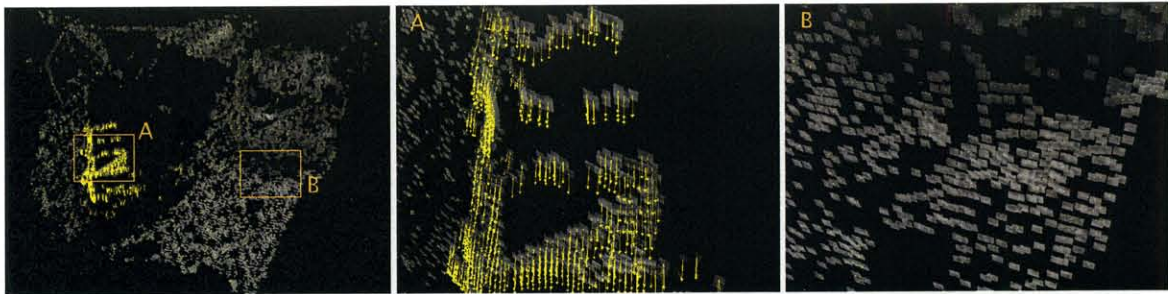


図 6.20: 3次元位置とモーシヨンの推定結果の拡大図
 Fig. 6.20: Estimated 3D position and 3D motion.

本節では、障害物の絶対モーシヨンの推定結果について述べる。

6.5.1 絶対モーシヨンの推定結果

図 6.21 は、障害物の絶対モーシヨンの推定結果である。同図の左側から横方向の運動である X 軸モーシヨン、高さ方向の運動である Y 軸のモーシヨン、奥行き方向の運動である Z 軸のモーシヨンの推定結果である。それぞれの図の上部には、推定された自車両の速度に換算した速度とフレーム毎のヨー角変化を表示している。モーシヨンが推定された点については、その色で絶対モーシヨンの大きさを表し、円形の大きさはその点の空間位置によって、遠くでは小さく、近くでは大きく表示している。

まず、上段の X 軸方向では [B] の前走車部分が若干黄色がかっている。Y 軸方向については、ほとんど変化のない緑色である。Z 軸方向のモーシヨンは、他車両の運動によって [B] の前走車は、自車に対して離れて行く方向に進んでおり、[C] や [D] の対向車は自車に近付いてくる方向に推定されている。[A] は、静止領域であるため、動きのほとんどない緑色となっている。図 6.22 は、図 6.21 の上段の結果をさまざまな視点から観察した様子を表示した結果である。同図 (a) は、前方を観察した様子であり、(b) はカメラの遠方からカメラの方向を見ている。同図で、青色の面は推定された道路面位置である。また、各点の 3次元推定位置には \times をつけ、そこから矢印を大きさと長さ (推定値の 5 倍に拡大) を推定されたモーシヨンに従って描写した。[D] の矢印がほとんど同じ大きさで同じ方向であることや、[A] の道路側の草はほとんどモーシヨンがないことが分かる。同図下段は、カメラ先方からカメラ方向を見た様子であり、手前に [B] が写っている。[B] はカメラから離れる方向の運動が推定されていることが表示されている矢印の向きより分かる。

次に図 6.21 の中段は、上段と同じような障害物の配置と運動である。対向車線を走るトラックも正しく絶対モーシヨンが推定されている。同図下段は、左車線に駐車している車両と右側に対向車が写っている。推定された絶対モーシヨンは、X 軸と Y 軸については運動のない緑色であり、Z 軸については、駐車車両は静止しているために緑色と、対向

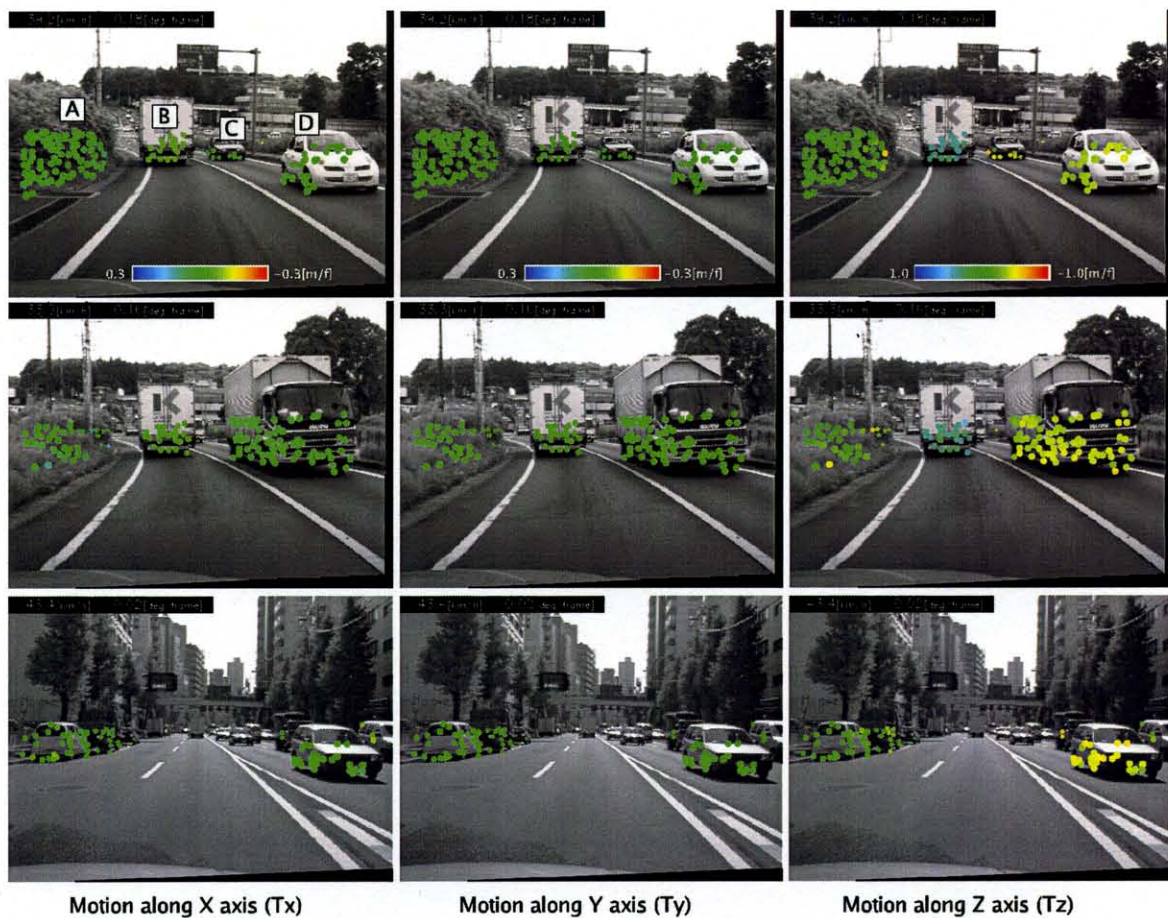


図 6.21: 障害物の絶対モーション推定結果
 Fig. 6.21: Estimated obstacle's absolute motion.

車は自車両に向かうため、黄色に表示されている。

図 6.23 は、絶対モーション推定結果を 1 フレームおきに示している。同図のシーンでは、左側に静止している道路側壁と前走車 2 台が右カーブを走行している。推定結果を確認すると、X 軸については前走車が緑より若干青みがかっているため、右方向に移動していることが分かる。手前の道路側壁は静止状態である緑色であるが、遠方では若干青みがかってしまっている。Y 軸については、高さ方向には運動していないため、静止状態として推定されている。Z 軸については、前走車が自車から遠ざかる絶対運動であるため、青色になっている。また、右側の前走車の方が速度が速いため、左の車両より深い青色となっている。

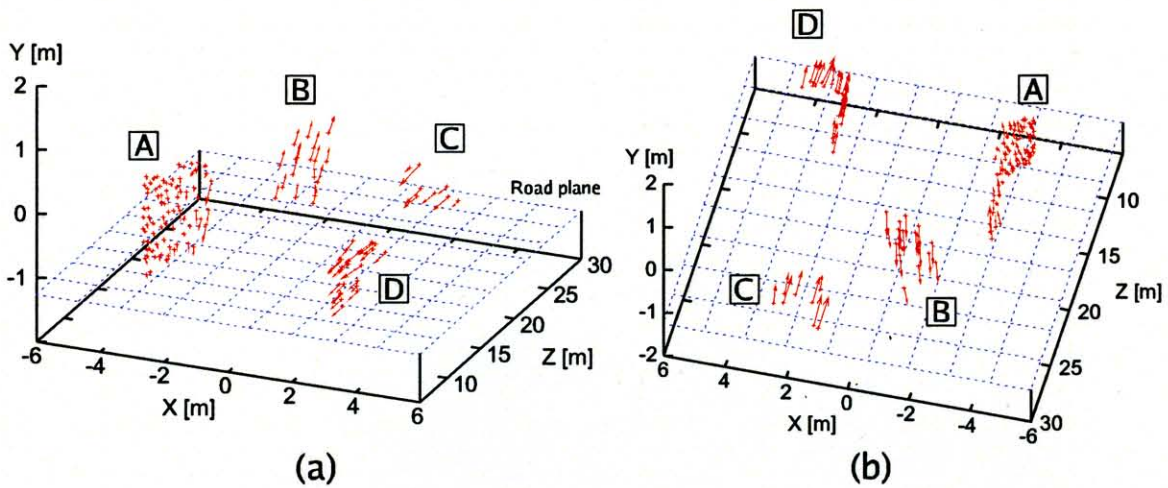


図 6.22: さまざまな視点から見た 3 次元位置と絶対モーション結果
 Fig. 6.22: Estimated 3D position and absolute 3D motion viewed from multiple viewpoints.

6.6 本章のまとめ

本章では、ステレオ動画像から画像中の点ごとにモーションと 3 次元位置を同時に最適化して推定する手法を提案した。提案手法では画像の点ごとにそれらを推定できるため、シーン中に複数のモーションが存在する場合においても、モーションセグメンテーションなどの前処理の必要がない。そして 3 次元位置とモーションを同時に推定するため、奥行き変化によるスケール倍変化をトラッキングの際に考慮することが可能である。また、利用する画像枚数を増やすことで推定精度を向上することが実験を通じて確かめられた。

今後の課題としては、ステレオ画像間や時系列画像間に発生するオクルージョンに対する処理が必要であると考えている。また、初期値として画像内で一様としているが、他の方法で初期値を推定しておけば、収束回数を早めることができ、計算時間を抑えられたり、ローカルミニマムに陥りにくくしたりできると考えられる。他に計算時間の短縮方法として、提案手法では Forward compositional の考え方に基づく方法であるため、式 (6.26) や式 (6.20) において、画像勾配である g を繰返しごとに計算し直しているが、Inverse compositional[4] の考え方を使ったアルゴリズムに拡張することでその必要がなくなり、計算の高速化ができると考えている。

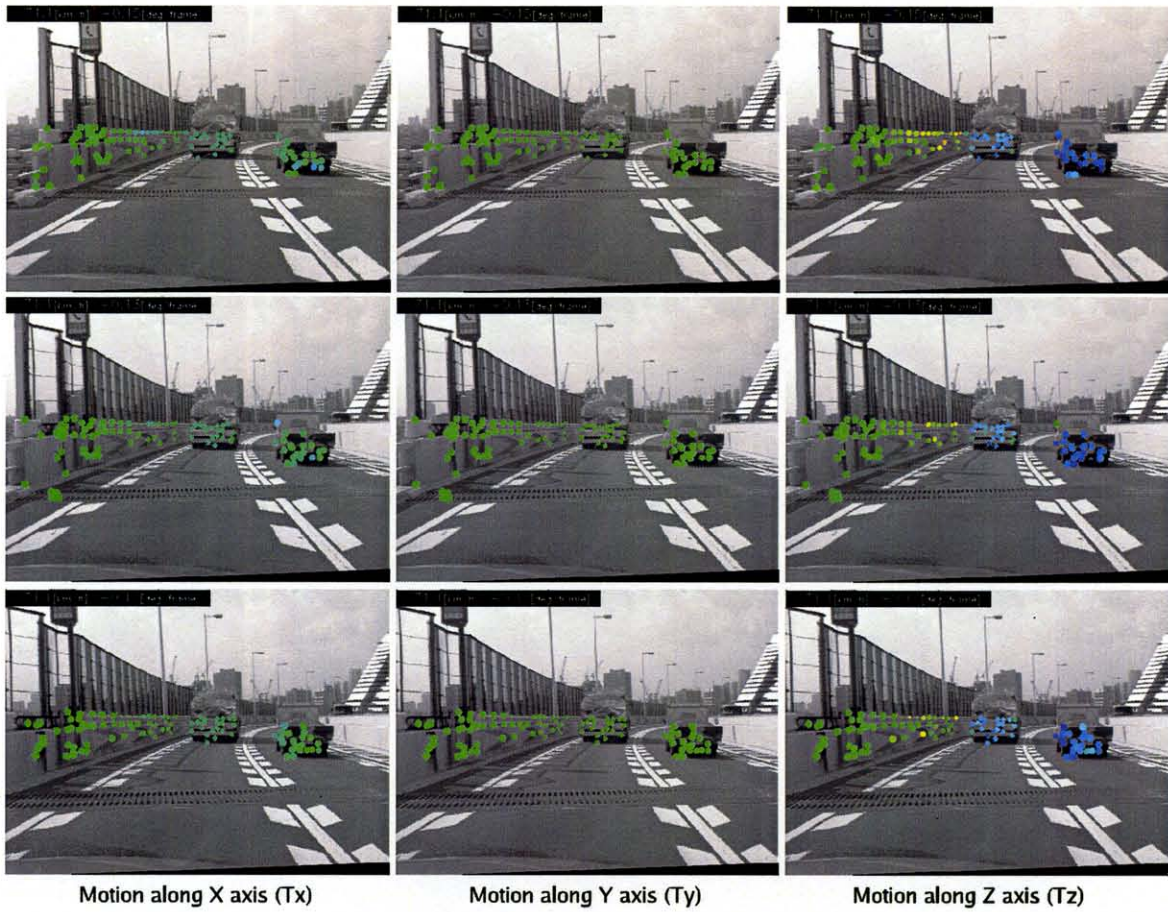


図 6.23: 障害物の絶対モーション推定結果. 1 フレームおきの結果を表示
 Fig. 6.23: Estimated obstacle's absolute motion. These results are taken every 2 frame.

第7章 非同期ステレオカメラを利用した 試み

ロボットや車両などの移動体にステレオカメラを設置して、ステレオ計測、モーション推定などを行なうことで、その情報を自己位置推定、視覚誘導、3次元地図の生成、障害物検出や回避などの用途に利用することができる。通常、これらを実現するために用いるステレオカメラは同期しており、同時刻に撮影されたステレオ画像であることが一般的である。このようなステレオ画像間には、図7.1に示すように、基礎行列によって対応位置に限定をすることができる。つまり、基準画像上のある点は参照画像上ではエピポーラ線上に対応位置が存在している。また、3次元計測によって推定される空間位置は同図で示すように、対応点の視線の交点として求めることができる。

一方、Webカメラや家庭用ビデオカメラのように、安価な市販のカメラを利用した場合には、両カメラで同一時刻に撮影ができない、非同期ステレオカメラであることがほとんどである。また形式の異なるカメラのように、物理的に同期ができない場合も考えられる。このようなカメラを用いて、同期ステレオカメラと同様にステレオ計測やモーション推定ができれば、安価で同期をとるための特別な装置などが不要な、カメラを選ばないシステムが実現できる。それによって、例えば家庭用ビデオカメラやネットワークカメラを用いることもできる。しかし、非同期ステレオでは、カメラごとに撮影されるタイミングが異なるため、移動物体が画像中に写っていたり、カメラが移動していたりすると各カメラで撮影されるタイミングによって、対象の3次元位置が異なってしまう。

図7.2には、非同期カメラで撮影された状況を示している。右カメラでは時刻 t で撮影されて、右画像に x_r の位置に投影され、左カメラでは Δt だけずれた時刻(本論文では時間オフセットと呼ぶこととする)で撮影されて、左画像に x_l' の位置に投影されている。また、 x_l は時刻 t と時刻 $t + \Delta t$ で仮にそれぞれ撮影された場合の左画像への投影位置を表している。同期ステレオの場合のように、基準画像の x_r の参照画像での対応点 x_l' は、撮影された時刻が異なるため、エピポーラ線上に存在しない。さらに、 x_r と x_l' の対応が仮に得られたとしても、それらの光線が交点から、正確な3次元位置を推定することが困難である。同図では、視線が交わらない場合である。

提案手法では、これらの問題を解決し、非同期ステレオカメラの映像だけを利用して、動的シーン中に含まれる対象の正しい3次元位置と、モーションを推定する手法を提案す

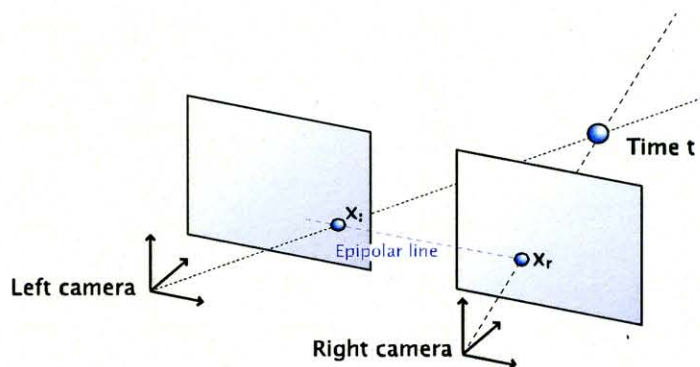


図 7.1: 同期ステレオと 3 次元位置計測
 Fig. 7.1: Synchronized stereo images and 3D measurement.

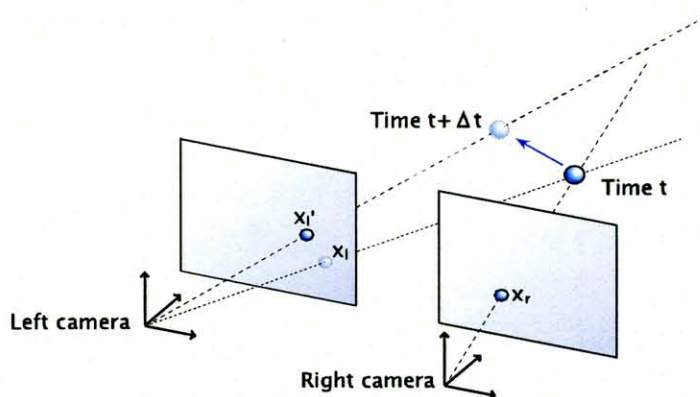


図 7.2: 非同期ステレオと 3 次元位置計測
 Fig. 7.2: Unsynchronized stereo images and 3D measurement.

る。また、ステレオ画像間のずれ時間である時間オフセットも併せて推定する。

そこで、まず関連論文について述べ、続いて非同期画像間のモーションと奥行きと時間オフセットに対する画像上の位置関係を定式化し、それを利用して、コスト関数を定義する。コスト関数は、1つの点で解く方法と複数点を同時に解く方法を説明し、合成画像と実画像実験を通じて、提案手法の有効性を示す。本章の最後に、本論文で提案した道路環境認識の処理に関して、非同期カメラを利用した場合の考察をする。

7.1 関連論文

非同期カメラを利用した研究は、同期カメラを利用したものに比べ、多くは提案されていない。Svedmanら [75]は非同期カメラを使って、特徴点の奥行きを推定する手法を提案している。また、清水ら [68, 69]は非同期カメラを使って、1台のカメラのフレームレート以上の頻度で奥行きを推定する手法を提案している。しかし、これらの方法では、

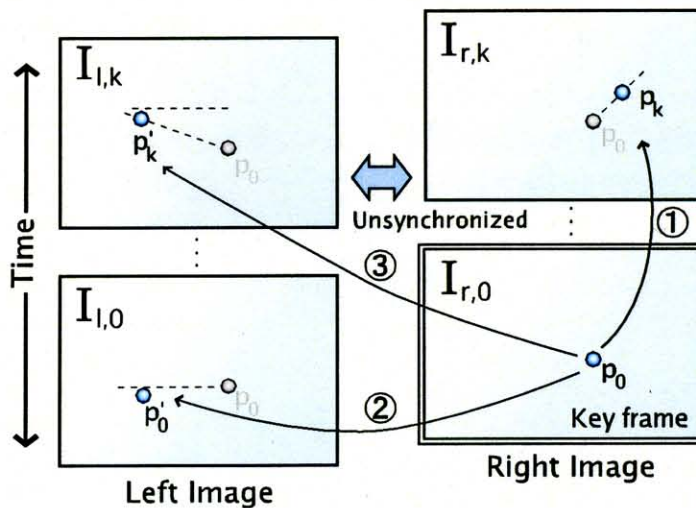


図 7.3: 非同期時系列ステレオ画像の対応点の関係
 Fig. 7.3: Unsynchronized stereo image sequence and correspondence.

非同期カメラの時間オフセットは既知としている。

Zhouら [91]は、あらかじめ非同期画像間で対応のとれた4点の位置とエピポーラ幾何によって、時間オフセットを推定した後、改めて時系列画像間でオプティカルフローを求め、推定されたフローを時間オフセットに応じて、同期ステレオ画像になるように画素ごとの位置と輝度をブレンディングして仮想的に同期画像を生成する。生成された同期ステレオ画像を用いて、従来のステレオ対応点探索を行なうことで、奥行きを推定している。

一方、提案手法では、複数の移動物体の含まれる動的なシーンに対して非同期ステレオカメラで撮影された画像をそのまま利用して、時間オフセットとそれぞれの点の奥行きと3次元モーションをすべて同時に最適化を行なうことができる。

7.2 画像間の関係

図 7.3は、キーフレーム(本章では右画像に設定) $I_{r,0}$ と時系列ステレオ画像間の関係を表している。

本章では、撮影間隔の等しい非同期ステレオカメラで動的シーンを撮影しており、カメラの外部パラメータおよび内部パラメータはあらかじめ求まっていることを想定している。また、推定するフレーム間の各点のモーションは、3次元空間中で等速直線運動と仮定する。

7.2.1 同一カメラの画像間の関係

同一カメラで異なる時刻に撮影された画像間の関係について述べる。この関係は、図7.3中の①である。空間中の奥行きが Z の点 $\mathbf{p}_0 = (x, y)^T$ が画像に投影される位置を $\mathbf{p}_0 = (x, y)^T$ とし、その点が 3D モーション $\mathbf{T}_m = [T_{mx}, T_{my}, T_{mz}]^T$ によって変化し、 k フレーム後(前)の画像 $I_{r,k}$ 中に写る点を \mathbf{p}_k とすると、

$$\mathbf{p}_k = \mathbf{p}_0 + \frac{k\mathbf{A}_x\mathbf{T}_m}{Z - kT_{mz}} \quad (7.1)$$

と表される。ここで、

$$\mathbf{A}_x = \begin{bmatrix} -f & 0 & \hat{x} \\ 0 & -af & \hat{y} \end{bmatrix}, \quad \hat{x} = x - u_0, \quad \hat{y} = y - v_0 \quad (7.2)$$

である。また、カメラの内部パラメータは、

$$\mathbf{A} = \begin{bmatrix} f & 0 & u_0 \\ 0 & af & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.3)$$

で表すことができる。右画像には、モーションに依存するエピポーラ拘束が存在する。このエピポーラ拘束は、モーションが未知であるので決まっていなかったが、モーションが決まると $1 \sim k$ 番目のすべての右画像のエピポーラ拘束が一意に決定される。

7.2.2 異なるカメラの画像間の関係

次に、異なるカメラの画像間の関係について述べる。まず、同図②の関係について説明する。ステレオカメラ間の時間オフセットを λ と表したとき、 \mathbf{p}_0 が左画像 $I_{l,0}$ に写る位置を \mathbf{p}'_0 とすると、

$$\mathbf{p}'_0 = \mathbf{p}_0 + \frac{\mathbf{A}_x\{\lambda\mathbf{T}_m + \mathbf{T}_s\}}{Z - \lambda T_{mz}} \quad (7.4)$$

で表される。ここで、 \mathbf{T}_s はステレオカメラ間の並進ベクトルである。

次に、同図③の関係については、 k フレーム後(前)の画像 $I_{l,k}$ に対応する位置は、

$$\mathbf{p}'_k = \mathbf{p}_0 + \frac{\mathbf{A}_x\{(k + \lambda)\mathbf{T}_m + \mathbf{T}_s\}}{Z - (k + \lambda)T_{mz}} \quad (7.5)$$

として表現できる。

7.3 同時推定アルゴリズム

本節では、前節で述べた非同期画像間の関係を利用して、まずコスト関数を定義し、次にそのコスト関数の解法について述べる。

7.3.1 コスト関数

各画像におけるキーフレームの点 p_0 の対応位置は、点 p_0 の回りに小領域 W を考え、全画像でその小領域内の各画素が最も合う位置とする。そこで、コスト関数 E を以下のように定義し、コスト関数が最小になるモーションと奥行きと時間オフセットを含むパラメータ \mathbf{m} を求める。

$$E(\mathbf{m}) = \sum_{k=0}^n \sum_{\mathbf{x} \in W} [\mathbf{I}_{l,k}(\mathbf{x} + \mathbf{u}_k) - \mathbf{I}_{r,0}(\mathbf{x})]^2 + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [\mathbf{I}_{r,k}(\mathbf{x} + \mathbf{u}'_k) - \mathbf{I}_{r,0}(\mathbf{x})]^2 \quad (7.6)$$

ここで、 $\mathbf{m} = [T_{mx}, T_{my}, T_{mz}, Z, \lambda]^T$ が未知数であり、 $\mathbf{I}_*(\mathbf{x})$ は画像 \mathbf{I}_* の位置 $\mathbf{x} = [x, y]^T$ における画像の輝度を表し、

$$\mathbf{u}_k = \frac{\mathbf{A}_x \{ (k + \lambda) \mathbf{T}_m + \mathbf{T}_s \}}{Z - (k + \lambda) T_{mz}}, \quad \mathbf{u}'_k = \frac{k \mathbf{A}_x \mathbf{T}_m}{Z - k T_{mz}} \quad (7.7)$$

である。式(7.6)の前半は異なるカメラの時系列画像間の SSSD (sum of SSDs) を表し、後半は同一カメラの時系列画像間の SSSD を意味している。つまり、式(7.6)は、マルチベースラインステレオ [37] の考え方を非同期ステレオ画像間に当てはめたものとも考えることもできる。

7.3.2 コスト関数の最小化

コスト関数を最小化するために、式(7.6)の近似をする。最初に、式(7.6)のコスト関数のうち、前半部分について述べる。

まず、式(7.7)の \mathbf{u}_k について、 $\Delta \mathbf{m}$ の変化分を考えると、

$$\mathbf{u}_k^\Delta = \frac{\mathbf{A}_x \{ (k + \lambda + \Delta \lambda) (\mathbf{T}_m + \Delta \mathbf{T}) + \mathbf{T}_s \}}{Z + \Delta Z - \{ (k + \lambda + \Delta \lambda) T_{mz} + \Delta T_z \}} \quad (7.8)$$

となる。さらに、上式の分母は以下のように表せる。

$$\frac{1}{Z + \Delta Z - \{ (k + \lambda + \Delta \lambda) T_{mz} + \Delta T_z \}} = \frac{1}{\dot{Z}_k} + \xi. \quad (7.9)$$

ここで、

$$\begin{aligned} \dot{Z}_k &= Z - (k + \lambda) T_{mz} \\ \xi &= \frac{-\{ \Delta Z - (k + \lambda) \Delta T_z - \Delta \lambda T_{mz} \}}{\dot{Z}_k^2 + \dot{Z}_k \{ \Delta Z - (k + \lambda) \Delta T_z - \Delta \lambda T_{mz} \}} \end{aligned} \quad (7.10)$$

である。次に、変化量は真値に対して小さい、つまり

$$Z - (k + \lambda) T_{mz} \gg \Delta Z - (k + \lambda) \Delta T_z - \Delta \lambda T_{mz}$$

と仮定すると,

$$\xi \approx \xi' = \frac{-\Delta Z + (k + \lambda)\Delta T_z + \Delta\lambda T_{mz}}{\dot{Z}_k^2} \quad (7.11)$$

となる. ξ' を使い, 式(7.8)を書くと

$$\mathbf{u}_k^\Delta = \left(\frac{1}{\dot{Z}_k} + \xi' \right) \mathbf{A}_x \{ (k + \lambda + \Delta\lambda)(\mathbf{T}_m + \Delta\mathbf{T}) + \mathbf{T}_s \} \quad (7.12)$$

となる. また, Δ が 2 乗の項は十分に小さい為無視し, 平行ステレオを想定しているため, $\mathbf{T}_s = [T_{sz}, 0, 0]^T$ として, 式(7.12)を変形すると,

$$\mathbf{u}_k^\Delta \approx \left(\frac{1}{\dot{Z}_k} + \frac{-\Delta Z + (k + \lambda)\Delta T_z + \Delta\lambda T_{mz}}{\dot{Z}_k^2} \right) \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \frac{1}{\dot{Z}_k} \begin{bmatrix} -(k + \lambda)f\Delta T_x + (k + \lambda)\hat{x}_0\Delta T_z + (-fT_{mx} + \hat{x}_0T_{mz})\Delta\lambda \\ -(k + \lambda)af\Delta T_y + (k + \lambda)\hat{y}_0\Delta T_z + (-afT_{my} + \hat{y}_0T_{mz})\Delta\lambda \end{bmatrix} \quad (7.13)$$

ここで,

$$\begin{aligned} \alpha &= (k + \lambda)(-fT_{mx} + \hat{x}_0T_{mz}) - fT_{sz} \\ \beta &= (k + \lambda)(-afT_{my} + \hat{y}_0T_{mz}) \end{aligned} \quad (7.14)$$

である. さらに変形すると, 式(7.15)を得ることができる.

$$\mathbf{x} + \mathbf{u}_k^\Delta = \begin{bmatrix} \gamma & 0 & \frac{\alpha'}{\dot{Z}_k} \\ 0 & \gamma & \frac{\beta'}{\dot{Z}_k} \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \frac{\kappa}{\gamma} & 0 & \frac{\zeta}{\gamma} \\ 0 & \frac{\kappa}{\gamma} & \frac{\eta}{\gamma} \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{I} + \mathbf{D}} \tilde{\mathbf{x}} \quad (7.15)$$

ここで,

$$\begin{aligned} \tilde{\mathbf{x}} &= [x, y, 1]^T \\ \alpha' &= -(k + \lambda)(fT_{mx} + u_0T_{mz}) - fT_{sz} \\ \beta' &= -(k + \lambda)(afT_{my} + v_0T_{mz}) \\ \gamma &= 1 + \frac{(k + \lambda)T_{mz}}{\dot{Z}_k} \\ \kappa &= 1 + \frac{1}{\dot{Z}_k} [(k + \lambda)(T_{mz} + \Delta T_z) + T_{mz}\Delta\lambda] + \xi'(k + \lambda)T_{mz} \\ \zeta &= -\frac{1}{\dot{Z}_k} [(k + \lambda)(f\Delta T_x + u_0\Delta T_z) + (fT_{mx} + u_0T_{mz})\Delta\lambda] + \xi'\alpha' \\ \eta &= -\frac{1}{\dot{Z}_k} [(k + \lambda)(af\Delta T_y + v_0\Delta T_z) + (afT_{my} + v_0T_{mz})\Delta\lambda] + \xi'\beta' \end{aligned} \quad (7.16)$$

で表される.

さて, 次に微小な要素を持つ行列 \mathbf{D} を用い, 座標変換 $\tilde{\mathbf{x}}'' \sim (\mathbf{I} + \mathbf{D})\tilde{\mathbf{x}}$ を考える. ここで, $\tilde{\mathbf{x}}$ は \mathbf{x} の同次座標表現とし, \mathbf{I} は 3 次の単位ベクトルである. $\tilde{\mathbf{I}}_{l,k}(\mathbf{x}'')$ ¹ について座標 \mathbf{x} の周りで 1 次のテイラー展開をすると,

$$\mathbf{I}_{l,k}(\mathbf{x} + \mathbf{u}_k^\Delta) - \mathbf{I}_{r,0}(\mathbf{x}) = \tilde{\mathbf{I}}_{l,k}(\mathbf{x}'') - \mathbf{I}_{r,0}(\mathbf{x}) \approx \mathbf{g}_{l,k}^T \mathbf{J}_{l,k}^T \Delta \mathbf{m} + e_{lr,k} \quad (7.17)$$

¹ $\tilde{\mathbf{I}}_{l,k}(\mathbf{x})$ は $\mathbf{I}_{l,k}$ を座標 $(\mathbf{x} + \mathbf{u}_k)$ でサブサンプリングして生成した画像である.

が得られる。ここで、

$$\begin{aligned} \mathbf{g}_{l,k}^T &= \nabla \tilde{I}_{l,k}(\mathbf{x}) \\ \mathbf{J}_{l,k}^T &= \frac{\partial \mathbf{x}''}{\partial \Delta \mathbf{m}} \\ e_{lr,k} &= \tilde{I}_{l,k}(\mathbf{x}) - I_{r,0}(\mathbf{x}) \end{aligned} \quad (7.18)$$

$\mathbf{g}_{l,k}^T$ は $\tilde{I}_{l,k}$ の画像勾配, $\mathbf{J}_{l,k}^T$ は Jacobian, $e_{lr,k}$ は $\tilde{I}_{l,k}$ と $I_{r,0}$ の輝度差分である。

さて, Jacobian を構成する行列は

$$\frac{\partial \mathbf{x}''}{\partial \Delta \mathbf{m}} = \frac{\partial \mathbf{d}}{\partial \Delta \mathbf{m}} \frac{\partial \mathbf{x}''}{\partial \mathbf{d}} \quad (7.19)$$

と変形できる。d は, 行列 \mathbf{D} を行ごとに並べたベクトルである。右辺の後半は,

$$\frac{\partial \mathbf{x}''}{\partial \mathbf{d}} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x^2 & -xy & -x \\ 0 & 0 & 0 & x & y & 1 & -xy & -y^2 & -y \end{bmatrix}^T \quad (7.20)$$

である [70]。右辺の前半は, 式 (7.15) の \mathbf{D} に相当する部分を $\Delta \mathbf{m}$ で偏微分することで

$$\frac{\partial \mathbf{d}}{\partial \Delta \mathbf{m}} = \begin{bmatrix} 0 & 0 & \frac{k+\lambda}{Z_k} + \frac{(k+\lambda)^2 T_{mz}}{Z_k^2} & -\frac{k T_{mz}}{Z_k^2} & \frac{T_{mz}}{Z_k} + \frac{(k+\lambda) T_{mz}^2}{Z_k^2} \\ -\frac{(k+\lambda)f}{Z_k} & 0 & -\frac{(k+\lambda)u_0}{Z_k} + \frac{(k+\lambda)\alpha'}{Z_k^2} & -\frac{\alpha'}{Z_k^2} & -\frac{f T_{mz} + u_0 T_{mz}}{Z_k} + \frac{\alpha' T_{mz}}{Z_k^2} \\ 0 & 0 & \frac{k+\lambda}{Z_k} + \frac{(k+\lambda)^2 T_{mz}}{Z_k^2} & -\frac{(k+\lambda) T_{mz}}{Z_k^2} & \frac{T_{mz}}{Z_k} + \frac{(k+\lambda) T_{mz}^2}{Z_k^2} \\ 0 & -\frac{(k+\lambda)af}{Z_k} & -\frac{(k+\lambda)v_0}{Z_k} + \frac{(k+\lambda)\beta'}{Z_k^2} & -\frac{\beta'}{Z_k^2} & -\frac{af T_{mz} + v_0 T_{mz}}{Z_k} + \frac{\beta' T_{mz}}{Z_k^2} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad (7.21)$$

が得られる。ここで, $\mathbf{0}$ は 5 次の零ベクトルを示している。これより Jacobian は,

$$\mathbf{J}_{l,k}^T = \frac{\partial \mathbf{x}''}{\partial \Delta \mathbf{m}} = \frac{1}{\gamma} \begin{bmatrix} j_{11} & 0 \\ 0 & j_{22} \\ \mu & \nu \\ j_{41} & j_{42} \\ j_{51} & j_{52} \end{bmatrix} \quad (7.22)$$

となる。ここで、

$$\begin{aligned}
j_{11} &= -\frac{(k+\lambda)f}{Z_k} \\
j_{22} &= -\frac{(k+\lambda)af}{Z_k} \\
j_{41} &= -\frac{\alpha' + (k+\lambda)xT_{mz}}{Z_k^2} \\
j_{42} &= -\frac{\beta' + (k+\lambda)yT_{mz}}{Z_k^2} \\
j_{51} &= -\frac{fT_{mz} - (x-u_0)T_{mz}}{Z_k} + \frac{\alpha'T_{mz} + (k+\lambda)xT_{mz}^2}{Z_k^2} \\
j_{52} &= -\frac{afT_{my} - (y-v_0)T_{mz}}{Z_k} + \frac{\beta'T_{mz} + (k+\lambda)yT_{mz}^2}{Z_k^2} \\
\mu &= \frac{1}{Z_k}(k+\lambda)(x-u_0) + \frac{1}{Z_k^2}(k+\lambda)\{\alpha' + (k+\lambda)T_{mz}x\} \\
\nu &= \frac{1}{Z_k}(k+\lambda)(y-v_0) + \frac{1}{Z_k^2}(k+\lambda)\{\beta' + (k+\lambda)T_{mz}y\}
\end{aligned}$$

である。

次に、式(7.6)のコスト関数の後半についても、前述した手順と同様に1次テイラー展開をすると、

$$I_{r,k}(\mathbf{x} + \mathbf{u}_k^{\Delta}) - I_{r,0}(\mathbf{x}) = \tilde{I}_{r,k}(\mathbf{x}'') - I_{r,0}(\mathbf{x}) \approx \mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \Delta \mathbf{m}' + e_{rr,k} \quad (7.23)$$

と表すことができる。ここで、

$$\begin{aligned}
\mathbf{g}_{r,k}^T &= \nabla \tilde{I}_{r,k}(\mathbf{x}) \\
\mathbf{J}_{r,k}^T &= \frac{\partial \mathbf{x}''}{\partial \Delta \mathbf{m}'} \\
\Delta \mathbf{m}' &= [\Delta T_{mx}, \Delta T_{my}, \Delta T_{mz}, \Delta Z]^T \\
e_{rr,k} &= \tilde{I}_{r,k}(\mathbf{x}) - I_{r,0}(\mathbf{x})
\end{aligned} \quad (7.24)$$

である。 $\tilde{I}_{r,k}(\mathbf{x})$, $\mathbf{g}_{r,k}^T$, $\mathbf{J}_{r,k}^T$, $e_{rr,k}$ は式(7.17)の画像 $I_{l,k}$ が画像 $I_{r,k}$ に置き換わったものである。Jacobianについても、同様に計算することで、

$$\mathbf{J}_{r,k}^T = \frac{\partial \mathbf{x}''}{\partial \Delta \mathbf{m}'} = \frac{1}{\gamma''} \begin{bmatrix} -\frac{kf}{Z_k} & 0 \\ 0 & -\frac{kaf}{Z_k} \\ \mu'' & \nu'' \\ -\frac{\alpha'' + kxT_{mz}}{Z_k^2} & -\frac{\beta'' + kyT_{mz}}{Z_k^2} \end{bmatrix} \quad (7.25)$$

が得られる。ここで、

$$\begin{aligned}
\ddot{Z}_k &= Z - kT_{mz} \\
\alpha'' &= -k(fT_{mx} + u_0T_{mz}) \\
\beta'' &= -k(afT_{my} + v_0T_{mz}) \\
\gamma'' &= 1 + \frac{kT_{mz}}{Z_k} \\
\mu'' &= \frac{1}{Z_k}k(x-u_0) + \frac{1}{Z_k^2}k(\alpha'' + kT_{mz}x) \\
\nu'' &= \frac{1}{Z_k}k(y-v_0) + \frac{1}{Z_k^2}k(\beta'' + kT_{mz}y)
\end{aligned} \quad (7.26)$$

で表される。よって、

$$\Delta \mathbf{m}' = \mathbf{T}^T \Delta \mathbf{m}, \quad \mathbf{T}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (7.27)$$

として、式(7.27)を用いて式(7.23)の $\Delta \mathbf{m}'$ を除去すると

$$\mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \Delta \mathbf{m}' + e_{rr,k} = \mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \mathbf{T}^T \Delta \mathbf{m} + e_{rr,k} \quad (7.28)$$

が得られる。

非同期ステレオ画像から奥行きとモーシヨンと時間オフセットを求めるには、最低2ペア計4枚の時系列ステレオ画像と推定したい点が1つ必要である。以下では、まず1点を利用したコスト関数の最小化方法について述べ、続いてより一般的な場合である複数点を同時に使ってコスト関数を最小化する方法を述べる。

1つの点についての解法

まず、奥行きとモーシヨンを推定したい画像上の点 \mathbf{p}_0 を1つだけ考える。式(7.17), (7.28)を考慮すると、コスト関数 E は以下ようになる。

$$E(\mathbf{m} + \Delta \mathbf{m}) = \sum_{k=0}^n \sum_{\mathbf{x} \in W} [\mathbf{g}_{l,k}^T \mathbf{J}_{l,k}^T \Delta \mathbf{m} + e_{lr,k}]^2 + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [\mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \mathbf{T}^T \Delta \mathbf{m} + e_{rr,k}]^2. \quad (7.29)$$

さて、上式を最小化するために、コスト関数 E を $\Delta \mathbf{m}$ で偏微分して、

$$\begin{aligned} \frac{\partial E}{\partial \Delta \mathbf{m}} &= \sum_{k=0}^n \sum_{\mathbf{x} \in W} [\mathbf{J}_{l,k} \mathbf{g}_{l,k} \mathbf{g}_{l,k}^T \mathbf{J}_{l,k}^T \Delta \mathbf{m} + e_{lr,k} \mathbf{J}_{l,k} \mathbf{g}_{l,k}] \\ &+ \sum_{k=1}^n \sum_{\mathbf{x} \in W} [\mathbf{T} \mathbf{J}_{r,k} \mathbf{g}_{r,k} \mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \mathbf{T}^T \Delta \mathbf{m} + e_{rr,k} \mathbf{T} \mathbf{J}_{r,k} \mathbf{g}_{r,k}] = 0 \end{aligned} \quad (7.30)$$

この式を整理すると、

$$\mathbf{A} \Delta \mathbf{m} = -\mathbf{b} \quad (7.31)$$

となる。ここで、

$$\begin{aligned} \mathbf{A} &= \sum_{k=0}^n \sum_{\mathbf{x} \in W} \mathbf{J}_{l,k} \mathbf{g}_{l,k} \mathbf{g}_{l,k}^T \mathbf{J}_{l,k}^T + \sum_{k=1}^n \sum_{\mathbf{x} \in W} \mathbf{T} \mathbf{J}_{r,k} \mathbf{g}_{r,k} \mathbf{g}_{r,k}^T \mathbf{J}_{r,k}^T \mathbf{T}^T \\ \mathbf{b} &= \sum_{k=0}^n \sum_{\mathbf{x} \in W} e_{lr,k} \mathbf{J}_{l,k} \mathbf{g}_{l,k} + \sum_{k=1}^n \sum_{\mathbf{x} \in W} e_{rr,k} \mathbf{T} \mathbf{J}_{r,k} \mathbf{g}_{r,k} \end{aligned} \quad (7.32)$$

と表される。式(7.31)より $\Delta \mathbf{m}$ を求め、さらにコスト関数を最小にするために $\Delta \mathbf{m}$ を繰り返し更新する。その際のパラメータの更新は

$$\mathbf{m} \leftarrow \mathbf{m} + \Delta \mathbf{m} \quad (7.33)$$

この式を ΔM で偏微分して 0 になる位置を求めると,

$$\begin{aligned} \frac{\partial E}{\partial \Delta M} &= \sum_{i=0}^q \mathbf{K}_i \left\{ \sum_{k=0}^n \sum_{\mathbf{x} \in W} [\mathbf{J}_{l,k,i} \mathbf{g}_{l,k,i} \mathbf{g}_{l,k,i}^T \mathbf{J}_{l,k,i}^T] \right. \\ &\quad + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [\mathbf{T} \mathbf{J}_{r,k,i} \mathbf{g}_{r,k,i} \mathbf{g}_{r,k,i}^T \mathbf{J}_{r,k,i}^T \mathbf{T}^T] \left. \right\} \mathbf{K}_i^T \Delta M \\ &\quad + \sum_{i=0}^q \mathbf{K}_i \left\{ \sum_{k=0}^n \sum_{\mathbf{x} \in W} [e_{lr,k,i} \mathbf{J}_{l,k,i} \mathbf{g}_{l,k,i}] \right. \\ &\quad \left. + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [e_{rr,k,i} \mathbf{T} \mathbf{J}_{r,k,i} \mathbf{g}_{r,k,i}] \right\} = 0 \end{aligned}$$

となる。よって、この式を整理すると,

$$\mathbf{A} \Delta M = -\mathbf{b} \quad (7.38)$$

となる。ここで,

$$\begin{aligned} \mathbf{A} &= \sum_{i=0}^q \mathbf{K}_i \left\{ \sum_{k=0}^n \sum_{\mathbf{x} \in W} [\mathbf{J}_{l,k,i} \mathbf{g}_{l,k,i} \mathbf{g}_{l,k,i}^T \mathbf{J}_{l,k,i}^T] \right. \\ &\quad \left. + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [\mathbf{T} \mathbf{J}_{r,k,i} \mathbf{g}_{r,k,i} \mathbf{g}_{r,k,i}^T \mathbf{J}_{r,k,i}^T \mathbf{T}^T] \right\} \mathbf{K}_i^T \\ \mathbf{b} &= \sum_{i=0}^q \mathbf{K}_i \left\{ \sum_{k=0}^n \sum_{\mathbf{x} \in W} [e_{lr,k,i} \mathbf{J}_{l,k,i} \mathbf{g}_{l,k,i}] \right. \\ &\quad \left. + \sum_{k=1}^n \sum_{\mathbf{x} \in W} [e_{rr,k,i} \mathbf{T} \mathbf{J}_{r,k,i} \mathbf{g}_{r,k,i}] \right\} \end{aligned} \quad (7.39)$$

である。上式を使って求められた ΔM を使って、パラメータの更新を

$$\mathbf{M} \leftarrow \mathbf{M} + \Delta M \quad (7.40)$$

として行なう。更新を繰り返すことで \mathbf{M} を推定する。これにより、複数点すべての奥行きとモーションとステレオ画像間の時間オフセットを同時に最適化しており、各点について複数枚の画像間の対応を同時に取ることに相当している。

7.3.3 推定が不可能な場合

本節では、非同期ステレオカメラに写る映像のうち、3次元位置、モーション、時間オフセットが求まらない場合について述べる。まず、キーフレーム中の点 \mathbf{p}_0 と、各画像に投影される位置 \mathbf{p}_1 , \mathbf{p}'_0 , \mathbf{p}'_1 ² との位置ずれを考える。まず、同一カメラについては,

$$\mathbf{p}_1 - \mathbf{p}_0 = \frac{1}{Z - T_z} \begin{bmatrix} -fT_x + \hat{x}T_z \\ -afT_y + \hat{y}T_z \end{bmatrix} \quad (7.41)$$

が成り立ち、ステレオカメラ間では,

$$\mathbf{p}'_0 - \mathbf{p}_0 = \frac{1}{Z - \lambda T_z} \begin{bmatrix} \lambda(-fT_x + \hat{x}T_z) - fT_{sx} \\ \lambda(-afT_y + \hat{y}T_z) \end{bmatrix} \quad (7.42)$$

²記号は 7.2 節を参照。

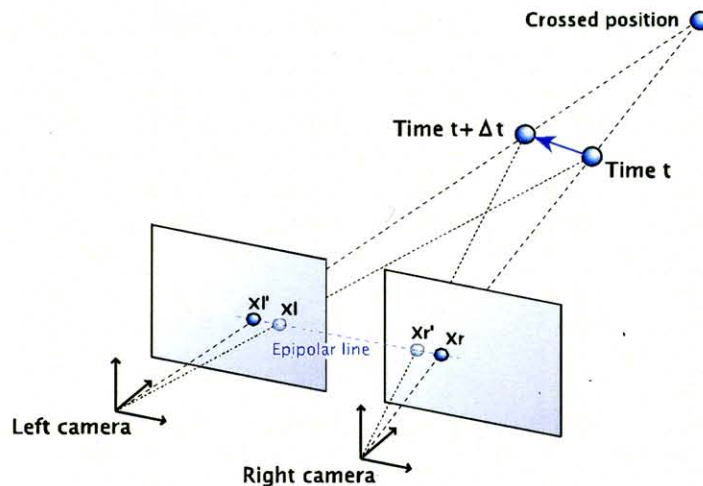


図 7.4: 推定不可能な対象のモーション
Fig. 7.4: Specific motion case.

と,

$$p_1' - p_0 = \frac{1}{Z - (1 + \lambda)T_z} \begin{bmatrix} (1 + \lambda)(-fT_x + \hat{x}T_z) - fT_{sx} \\ (1 + \lambda)(-afT_y + \hat{y}T_z) \end{bmatrix} \quad (7.43)$$

の関係が成り立っている。これらの式は、未知数が5つに対して式が6つあるので、2ペア4枚のステレオ画像と1点で未知パラメータの推定が可能である。ここで、これらの式の右辺の行列の2行目に注目すると、 $(-afT_y + \hat{y}T_z)$ が積の形だけであることがわかる。すなわち、

$$-afT_y + \hat{y}T_z = 0 \quad (7.44)$$

であるときは、式の数が4つになってしまい、未知数の数より少なくなってしまうため、未知パラメータを推定することができない。平行ステレオ配置では、画像縦軸に移動しない場合であり、例えば、 $T_y = 0$ かつ $T_z = 0$ のときには、X軸のみに運動する時である。このような場合には、文献[91]の手法においても推定することができない。幾何学的に考えると、図7.4に示すように、移動物体がエピポーラ線に一致した運動をする場合であり、視線が誤った空間位置で交差してしまう。

7.3.4 Implementation

7.3.2節で述べた複数点を同時に最適化するアルゴリズムの推定手順をまとめたものが、図7.5である。この推定手順に従うことで、モーションと奥行きと時間オフセットを推定する。その際、キーフレームから推定に利用する点を複数個選ぶ必要がある。本実験では

```

for (iter=0 to iter max or convergence condition)
  if iter is 0
    Set initial value to M.
  end if.
  for (k=0 to max image)
    for (i=0 to max point)
      if k is not 0
        Warp image  $I_{r,k,i}$  and get  $\tilde{I}_{r,k,i}$  using  $m_i$ .
        Calculate  $g_{r,k,i}$ ,  $J_{r,k,i}$ ,  $e_{rr,k,i}$  in eq.(7.24).
      end if
      Warp image  $I_{l,k,i}$  and get  $\tilde{I}_{l,k,i}$  using  $m_i$ .
      Calculate  $g_{l,k,i}$ ,  $J_{l,k,i}$ ,  $e_{lr,k,i}$  in eq.(7.18)
    end for loop.
  end for loop.
  Calculate A and b in eq.(7.39).
  Calculate  $\Delta M$  using eq.(7.38).
  Update  $M \leftarrow M + \Delta M$  in eq.(7.40).
end for loop.
3D MOTION, DEPTH for each point, and TIME OFFSET parameters are given.

```

図 7.5: 複数点を用いたときの処理手順

Fig. 7.5: Optimization step of cost function in case of multiple points.

Shiらの手法[66]を用いて特徴点を抽出し、利用した。また、その他に提案手法を実装し、処理を行う上で参考となる事項を以下に挙げる。

最適化のための繰り返し回数の削減

図 7.5 の iter に関する繰り返し計算の回数を削減し、初期値にロバストにするために、縮小した画像でおおまかに最適化し、それを原画像のスケールへ戻して最適化する、ガウシアンピラミッドを利用した。

カメラ配置

本章では、定式化を簡単にするためにステレオカメラは平行に配置され、かつカメラの内部パラメータが両方のカメラで同じと仮定している。そのため、ステレオカメラが平行ステレオでない場合や内部パラメータが一致していないときは、前処理として付録 B で述べている Rectification を行なうことで仮想的に平行カメラにし、かつカメラの内部パラ

メータをステレオカメラ間で一致させる。

逆行列の計算方法の工夫

式(7.38)に示すように未知パラメータ M を最適化するための繰返し計算中で、 ΔM を求める。その際、 A の逆行列を計算する必要がある。 A は、計算に利用する点の数が n 個であった場合には、 $(4n+1) \times (4n+1)$ と大きなものになってしまう。逆行列計算は計算コストが大きく、 $N \times N$ の行列では $O(N^3)$ である。一方、以下に述べる計算の工夫をすることで、そのコストを $O(N)$ に下げることができる。

A を要素ごとに書くと、以下のように 0 を多く持つ粗な行列であることが分かる。

$$A = \begin{bmatrix} a_{1,1} & \dots & a_{1,4} & 0 & \dots & 0 & a_{1,4n+1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & & \\ a_{4,1} & \dots & a_{4,4} & 0 & \dots & \vdots & \\ 0 & \dots & 0 & a_{5,5} & \dots & a_{5,8} & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \\ 0 & \dots & 0 & a_{5,5} & \dots & a_{5,8} & 0 \\ 0 & & & \dots & & 0 & \ddots \\ a_{4n+1,1} & & & \dots & & & a_{4n+1,4n+1} \end{bmatrix} \quad (7.45)$$

ここで、ある行列が正方行列 E, H と行列 F, G で構成されているとき、その逆行列は以下のようなになる。

$$\begin{bmatrix} E & F \\ G & H \end{bmatrix}^{-1} = \begin{bmatrix} E^{-1} + E^{-1}FS^{-1}GE^{-1} & -E^{-1}FS^{-1} \\ -S^{-1}GE^{-1} & S^{-1} \end{bmatrix} \quad (7.46)$$

ただし、 $S = H - GE^{-1}F$ である。そこで、

$$E = \begin{bmatrix} a_{1,1} & \dots & a_{1,4} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{4,1} & \dots & a_{4,4} & 0 & \dots & 0 \\ 0 & \dots & 0 & \ddots & 0 & \dots & 0 \\ \vdots & & \ddots & 0 & a_{4n-3,4n-3} & \dots & a_{4n-3,4n} \\ \vdots & & & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & a_{4n,4n-3} & \dots & a_{4n,4n} \end{bmatrix}, \quad F = \begin{bmatrix} a_{1,4n+1} \\ \vdots \\ a_{4n,4n+1} \end{bmatrix} \quad (7.47)$$

$$G = \begin{bmatrix} a_{4n+1,1} & \dots & a_{4n+1,4n} \end{bmatrix}, \quad H = a_{4n+1,4n+1} \quad (7.48)$$

とする。 S の逆行列は 1 行 1 列であるので、逆数として簡単に求まる。 E の逆行列は、 $4n$

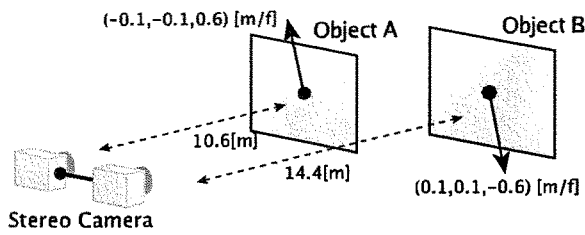


図 7.6: カメラとオブジェクトの配置とそのモーシオン
Fig. 7.6: Layout of cameras and objects, and 3D motion.

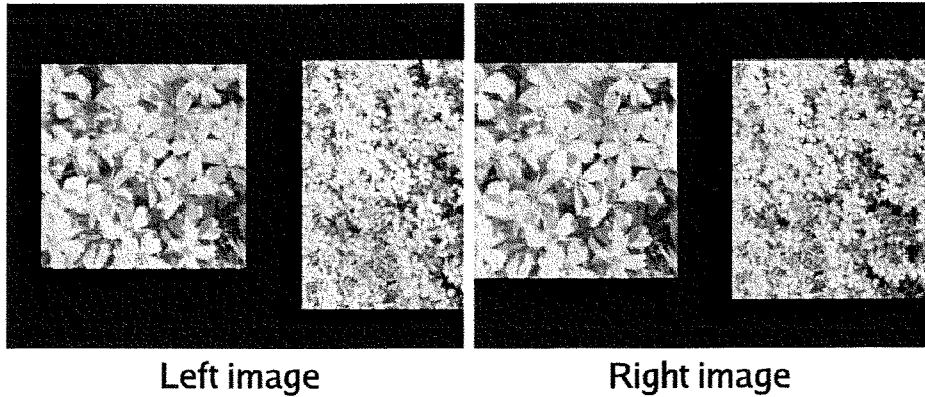


図 7.7: 入力した非同期ステレオ画像の一部
Fig. 7.7: Example of input unsynchronized stereo image.

行 $4n$ 列であるので計算コストが大きいが、粗な行列であるため、

$$\mathbf{E}^{-1} = \begin{bmatrix} \mathbf{e}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{e}_n \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{e}_1^{-1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{e}_n^{-1} \end{bmatrix} \quad (7.49)$$

とすることができ、実際には 4 行 4 列の逆行列を n 回計算を行えばよい。これによって、逆行列の計算コストを抑えることができる。

7.4 実験結果

7.4.1 合成画像実験

合成画像実験では、図 7.6 に示すようにカメラに対して、奥行き異なる 2 枚の平行な板を配置した。これらの板は、それぞれ別々の運動をしている。

実験には、図 7.7 に示すようにステレオ画像間で 1 フレームずれた非同期ステレオ画像 2 フレームを用いた。画像のサイズは、 320×240 [pixel] である。初期値は、奥行きを 12[m]、モーシオンを $\mathbf{T}_m = (1e^{-5}, 1e^{-2}, 1e^{-1})^T$ とし、時間オフセット λ は 0 とした。

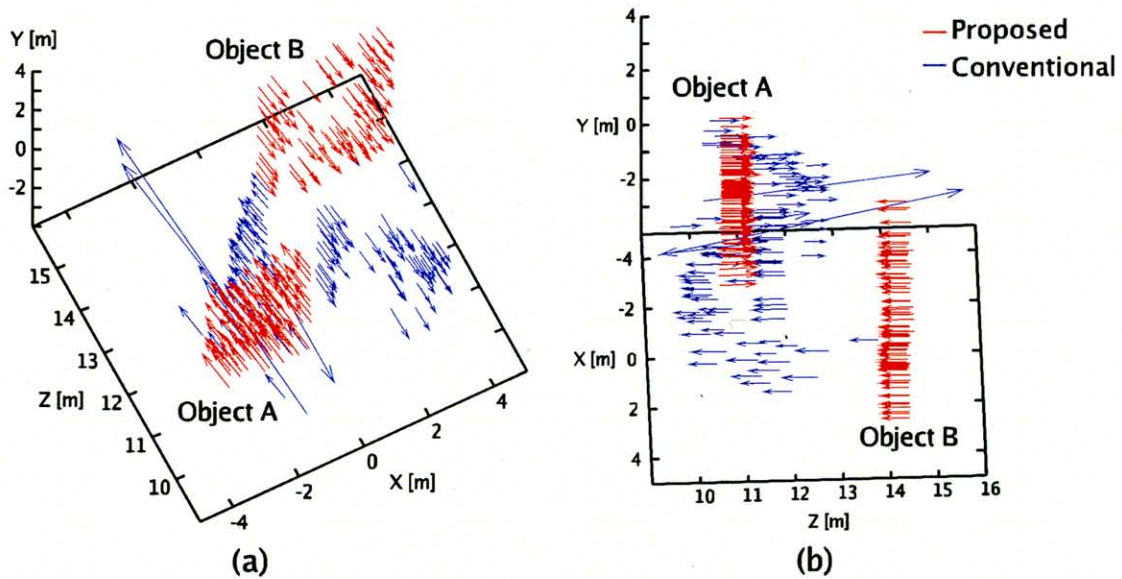


図 7.8: 3次元位置とモーシヨンの推定結果
 Fig. 7.8: Estimated 3D position and 3D motion.

提案手法によって推定された時間オフセットは、 -1.006 (真値 -1.0)となり、正確に推定されていた。また、奥行きとモーシヨンについては従来法と比較を行った。従来法は、非同期ステレオ画像間に対応点探索にウィンドウベースのマッチングを行い、次にその点に関して同じカメラの時系列画像間でトラッキングを行い、最後にトラッキングされた点について非同期ステレオ画像間に対応点を探索することで、各時刻の同一点に関する3次元位置同士の差分からモーシヨンと奥行きを求めた。

図 7.8は、提案手法と従来手法による3次元位置とモーシヨンの推定結果を表している。モーシヨンは、推定された空間位置を原点として、方向と長さはモーシヨンの方向と大きさに対応している。同図の赤矢印は提案手法であり、青矢印は従来手法である。同図 (b)を見ると、提案手法による推定結果は、矢印の開始位置が物体ごとに同じ面上にあり、かつその長さが等しいことから、正しく推定されていることが分かる。一方、従来手法では、推定位置やモーシヨンが散らばっており、正しく推定されていない。

図 7.9は、提案手法と従来手法の推定結果のRMSエラーを表している。RMSエラーは、提案手法と従来手法で違いが大きかったため、縦軸に対数をとっており、オブジェクトごとに計算している。従来法では、非同期ステレオのため、対応点探索がうまくいかないことや対応がとれたとしても撮影時間ずれによる物体の位置変化から、正しく奥行きとモーシヨンが推定できておらず、エラーが大きいたことが分かる。それに対して、提案手法では正確に推定されていることが分かる。

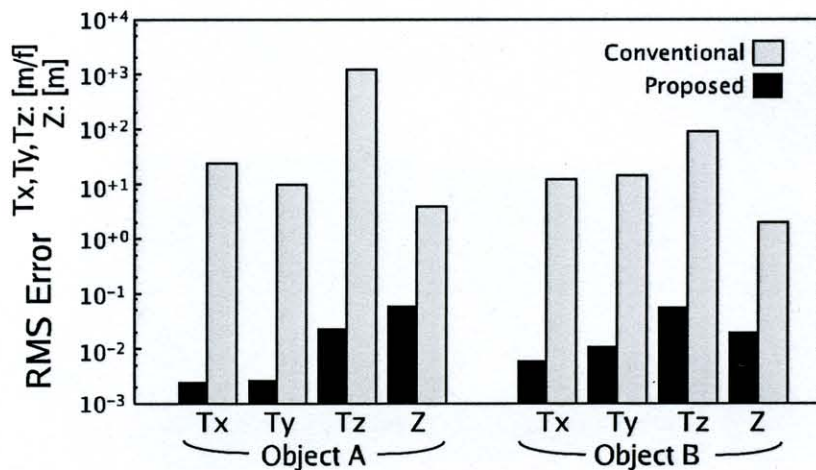


図 7.9: 従来手法との推定精度比較結果

Fig. 7.9: Comparison results of conventional and proposed method.



図 7.10: 非同期ステレオ画像ペア

Fig. 7.10: Input unsynchronized stereo image pair.

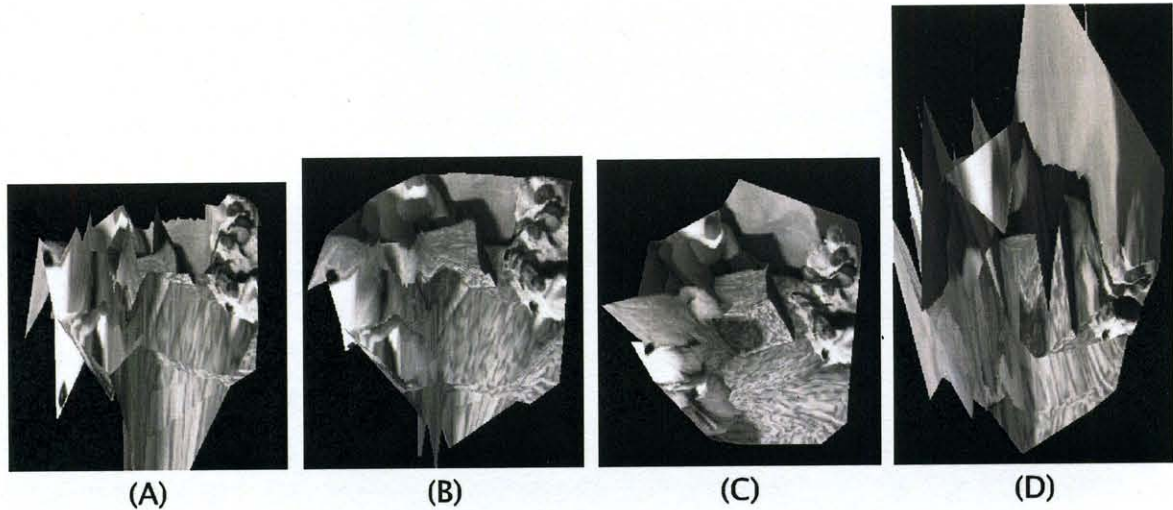
7.4.2 実画像実験

3次元復元結果

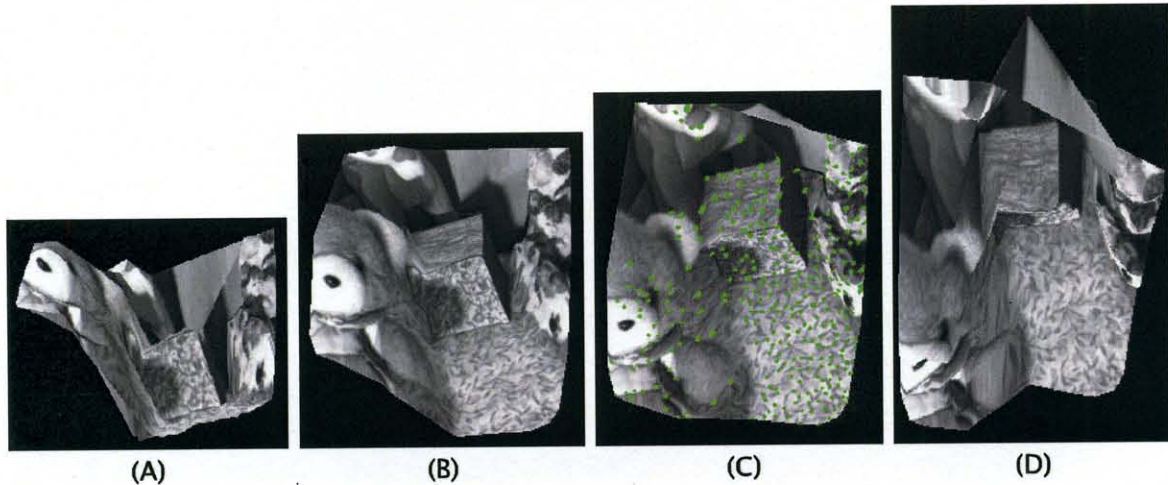
図 7.10 は、手持ちステレオカメラで撮影された同期カメラの画像を 1 フレームずらすことで生成した非同期ステレオ画像の一部である。

提案手法では、2 ペアの非同期時系列ステレオ画像を用いた。計算に用いる初期値として、奥行きは 0.7[m]、モーションはすべて 0.001[m/f]、時間オフセット λ は 0.1 とし、特徴点は 342 個利用した。

図 7.11 は、従来手法を利用して 3 次元復元した結果にテクスチャを貼り付けて表示している。次に、図 7.12 は、提案手法によって推定された結果を示している。従来手法の (A)-(D) と提案手法 (A)-(D) は、それぞれ同じ視点からほぼ同じ方向を観察した結果であ



(A) (B) (C) (D)
 図 7.11: 従来手法による再構成結果をさまざまな方向から見た様子
 Fig. 7.11: Reconstructed results with texture mapping by conventional method.



(A) (B) (C) (D)
 図 7.12: 提案手法による再構成結果をさまざまな方向から見た様子 ((C) の緑の点は、推定の際に同時に利用した特徴点である)

Fig. 7.12: Reconstructed results with texture mapping by proposed method.

る。従来手法では、画像間で対応が正しくとれないため、復元結果が大幅に悪化していることが分かる。一方で、提案手法では、立方体の物体、手前に写っている熊のぬいぐるみや床面などが正しく復元されていることが確認できる。

2台の異なるカメラによる結果

本実験は、Point Grey 社製 Dragonfly と Sony 社製 VX-2000 のホームビデオカメラを用いた。これらのカメラ同士は、同期できる仕組みがないため、各カメラは別々のタイミン

グで撮影され、真の時間オフセットを知ることができない。入力した非同期ステレオ画像は、図 7.13 に示す 2 フレーム、計 4 枚である。シーンには、模型自動車 **A** (同図中の記号に一致させている) と手で動かした熊のぬいぐるみ **C** が動物体として含まれている。ステレオカメラは三脚に固定したため、背景領域や置物 **B** は、静止している。

初期値は、奥行きを 0.7[m]、時間オフセットを 0.0、モーションは XYZ 軸ともに 0.001[m/f] とした。提案手法によって推定された時間オフセットは、0.719 であった。図 7.14(a) は、推定された 3 次元位置にテクスチャを張り付け、それぞれ異なる視点から観察した結果である。**A** の模型自動車や、**B** の置物、**C** の熊のぬいぐるみなどが正しい位置に推定されていることが分かる。

同図 (b) は、推定された 3 次元位置にモーションを併せて表示した結果である。矢印の方向と長さは推定されたモーションの方向と大きさに相当している。ここでは推定されたモーションを見易くするため、2 倍に拡大して表示している。また、同図 (b) は同図 (a) とそれぞれほぼ同じ方向から観察した様子に相当している。**A** の模型自動車領域でモーションが大きく、それらのモーションを示す矢印は平行でかつ同じ長さとなっており、模型自動車のモーションが正しく推定されていることが分かる。静止領域である **B** には、矢印がなく、+ マークだけが表示されていることから、動いていないことがわかる。**C** は、**A** と **B** とも違うモーションであることがわかる。

7.5 非同期ステレオを利用した道路環境認識についての考察

車載ステレオカメラが非同期ステレオであるときには、車両の運動速度や方向などによって、各時刻のステレオカメラにおける、幾何学的関係がそれぞれ異なってしまう。それによって、先にも述べたように通常のステレオ対応点探索を行なっても、対象までの距離やモーションを正しく求めることができない。

そこで本節では、本論文で述べてきた各手法が非同期ステレオカメラを利用した場合に適用可能であるのか否かについてまとめる。図 7.15 は、以下で述べる非同期ステレオカメラを利用した際の各手法の関連性を簡単に表している。

まず、本論文の冒頭で述べた研究の基盤になっている、道路領域抽出と道路面とカメラ間の姿勢推定について述べる。道路領域を抽出するには、ステレオ画像間の射影変換行列が推定できればよい。射影変換行列の推定方法は 2.1 節で述べたように、カメラ間の関係が既知でなくても推定可能である。よって、非同期カメラを使っても道路領域の抽出が可能である。次に、道路面とカメラ間の姿勢については、道路面に対する射影変換行列が正しく求められていれば、2.2.1 節の方法によって道路面の姿勢パラメータである道路面の法線ベクトルを求めることができる。しかし、カメラから道路面までの距離は、カメラ間のベースライン長が分からないため、原理的には推定ができない。しかし、道路面の法線

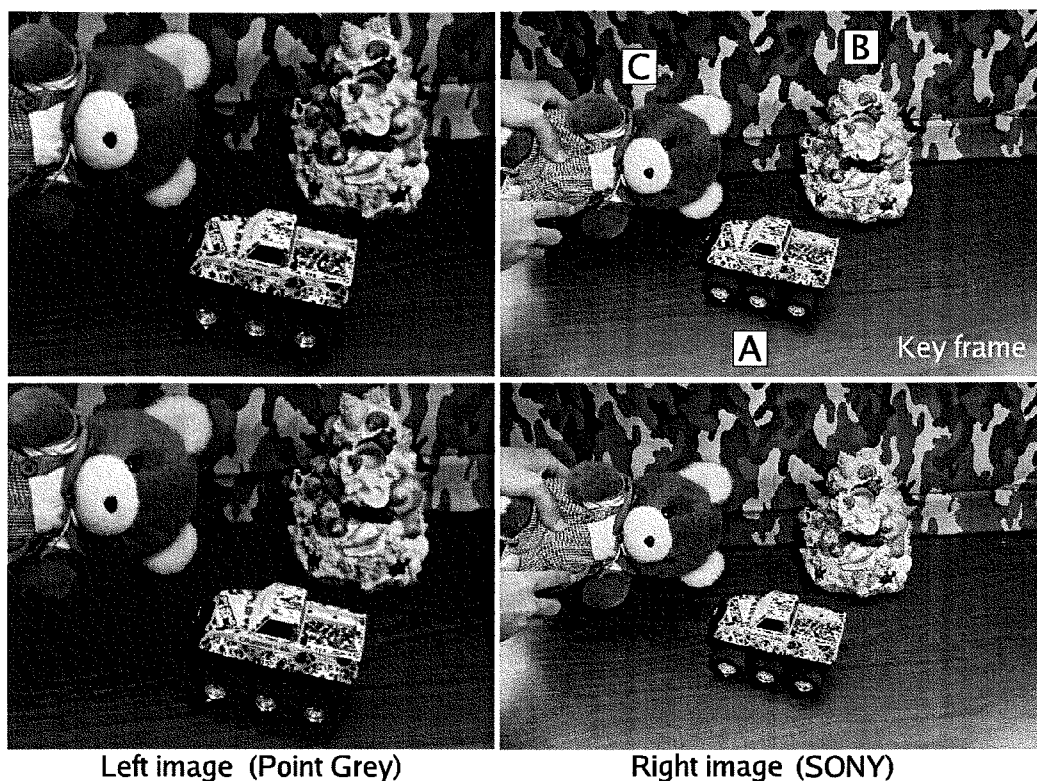


図 7.13: 別々のカメラで撮影された非同期時系列ステレオ画像
 Fig. 7.13: Unsynchronized sequential stereo images taken by cameras of different types.

ベクトルと違い道路面までの距離は大きく変化することは少ないと考えれば、車両が静止しているときに算出した値を利用できる。よって、近似的にはあるが、道路面とカメラ間の姿勢が得られる。

次に、第3章の危険領域検出については、本手法の処理には道路領域と道路面とカメラ間の姿勢のみを利用しており、ステレオ画像間のマッチングなどを行わないため、非同期ステレオカメラでも適用が可能である。

さて、次に第4章の障害物検出について述べる前に、第5章の自車両の運動推定について述べる。自車の運動を推定するには、道路領域と道路面の姿勢パラメータが必要であり、先に述べた通り、これらは非同期ステレオカメラでも利用可能である。カメラ間のフレームレートが既知であれば、これらを利用して自車の運動を推定することができる。続いて、第4章の障害物検出については、ステレオカメラ間の関係がステレオカメラ間の撮影時間ずれによって、見かけ上変化してしまう。そのため、第4章で述べたステレオ計測部分から正しい3次元位置データを得ることができない。そこで、ステレオ計測部分だけを、本章で述べた非同期ステレオカメラによる3次元位置と、モーションの同時推定法に置き換えることで、対象の3次元位置が分かり、以降の処理は第4章と同じように、その

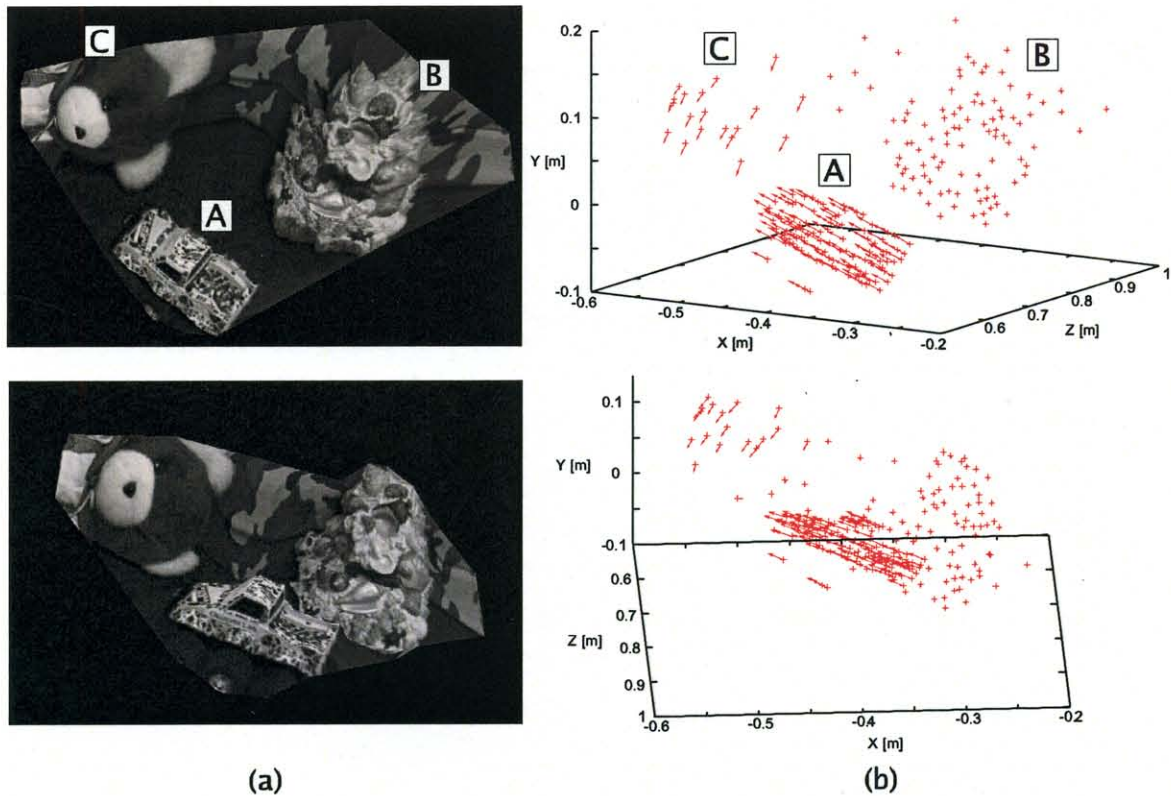


図 7.14: (a)3次元推定結果をさまざまな視点から観察した様子. (b)3次元推定位置とモーション推定結果. モーションを2倍に拡大して表示.
 Fig. 7.14: (a)Estimated 3D position viewd from multiple viewpoints.(b)Estimated 3D position and 3D motion.

3次元位置情報を利用して投票処理を行ない、クラスタリングによって障害物検出を行なうことができると考えられる。

第6章で述べた対象のモーション推定については、本章で述べた方法によって、非同期ステレオカメラでもモーション推定が可能である。非同期ステレオカメラを利用して、障害物に対するモーション推定と自車の運動推定により、6.5節のように障害物の絶対モーションが推定される。

7.6 本章のまとめ

本章では、非同期ステレオカメラを用いて、画像中の各点の3次元位置と3次元モーションと、加えてカメラ間の時間オフセットを同時に最適化して推定する手法を提案した。本手法では、画像中の各点ごとの3次元位置とモーションを求めつつ、画像全体の時間オフセットを同時に推定することで、画像中に複数の物体が含まれているような動的

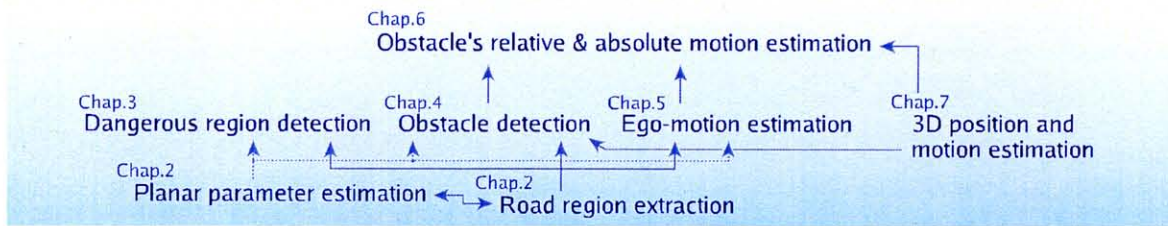


図 7.15: 非同期ステレオを用いた際の各手法の関連
 Fig. 7.15: Overview of proposed methods with unsynchronized stereo camera.

シーンにおいても、適用することができる。合成画像と実画像を用いて、従来手法と提案手法の比較を行い、提案手法によって、従来手法では推定できなかった3次元位置や、3次元モーションが推定できていることを示した。

最後に、非同期ステレオカメラを利用した場合に、本論文で述べてきた道路環境認識に関する各手法が適用可能であるのかについて考察をした。

第8章 結言

8.1 本論文の結論

本論文では、車載ステレオカメラを利用した車両の前方環境認識に関する手法を提案した。提案手法では、車載カメラの映像から空間中の道路領域を推定し、それらを活用して危険領域や障害物を検出し、その空間位置を特定した。さらに、自車両の運動を推定した。加えて、シーン中の各点の空間位置とモーションを同時に推定し、それを利用して障害物の運動を推定した。また、より一般的なカメラを利用した非同期ステレオに対しても、3次元計測やモーション推定が行える方法を提案した。以下に各章で述べた事項について簡単にまとめる。

まず、第2章「道路領域抽出と道路面の姿勢推定」において、道路環境を認識する上で基礎となる道路領域抽出手法と道路面とカメラ間の姿勢推定方法について述べた。本手法で平面領域は、空間中の平面とステレオ画像の幾何学的関係を利用することで抽出した。また、道路等の走行可能な領域を平面とみなすことで、本手法を利用し、道路領域を抽出することができることを示した。本手法では、道路面に対する大きなウィンドウを変形してマッチングをするため、テクスチャの少ない路面でも安定に抽出することができ、道路白線が必ずしも存在しない、市街地や雪道でも適用することができた。次に、道路面とカメラ間の姿勢推定方法について述べた。姿勢には、カメラに対する道路面の法線ベクトルと、カメラの光学中心と道路面までの距離があり、道路面に対するステレオ画像間の射影変換行列から求められることを示した。

第3章「仮想投影面画像の生成と危険領域検出」では、第2章で抽出された道路領域と、推定された道路面の姿勢を用いた危険領域検出手法を提案した。本手法は、まず、道路面とカメラ間の相対的な姿勢を用いて、道路領域のVPP画像を生成した。ここで、VPP画像とは、道路面を仮想的に垂直上方から観測した画像である。危険領域は、VPP画像に変換された道路領域の境界を、光軸からの方向別に距離を計測して、時系列間でその変化具合を測定することによって検出した。本手法では、トラッキング作業を行わずに危険性の判定が可能であったり、実空間中での危険領域の位置や距離を、VPP画像上の位置が実座標に対応している性質を利用することで、ステレオ対応点探索なしに求めたりす

ることが可能であった。そのため、簡単な処理で危険領域の検出やその位置の計測ができた。実験結果から、対向車などの大きな物体以外にも、一般的に検出が難しいと考えられる、ガードレールの先端などの細かい部分も、危険な領域として検出できることが確認できた。

しかし、第3章の方法では、物体が道路面に接していない場合には、対象までの距離を正確に測定できない問題があることから、第4章「道路面を基準とした障害物検出」では、第3章とは異なり、障害物を自車両が通過する可能性のある空間に属する立体物とし、陽に障害物を検出する手法を提案した。本手法は、ステレオ計測を利用して、画像中の各点の3次元位置を測定し、それらの点の集合から、道路上に存在する立体物を、道路面の姿勢を利用して分離し、さらにセグメンテーション処理を行なって、障害物を検出した。その際、道路領域抽出結果を用いることで、画像中の障害物が写っている領域である、非平面領域だけに限定して、対応点探索を行なうことにより、障害物を未検出することなく、計算コストを抑えた。さらに、動的に推定されたカメラと道路面間の姿勢を利用することで車両の振動による、道路面とカメラ間の変動に耐えられた。加えて、道路面と平行な投票グリッドを用いて、計測された3次元データを高さ方向に投票することによって、自動車、バイク、歩行者などの道路環境に存在する物体を、そのままの形状に近い形で安定に検出できた。このように、道路環境と障害物の性質をそれぞれの処理に巧みに取り入れることで、さまざまな実環境下において、機能することを確認した。さらに、実際のシーンの検出率を調べた結果、本手法の検出率は高いことが示された。

第5章「自車両の運動推定」では、自車両の運動推定手法を提案した。車両の運動推定をするには、静止している領域を基準として、道路面に対するモー션을推定する必要がある。そこで、まず、道路領域を抽出することによって静止領域を特定し、その際に推定されたカメラに対する道路面の姿勢を利用して、VPP画像を生成した。VPP画像は道路面を垂直上方から観察した画像に相当するため、車両の運動を2次元平面内の並進と回転の運動に限定できた。最初にVPP画像の道路領域内の移動ベクトルと車両の運動モデルを用いることで、特定の注目点のトラッキングを行なうことなく、簡便かつ安定に自車両の運動を推定した。次に、その結果を初期値として並進と回転を勾配法の考え方をを用いた解法によって、運動パラメータを更新し、より高精度に運動を推定した。本手法を合成画像実験に適用し、推定精度について検証した。また、さまざまな環境下で撮影された実画像を使って、実際に自車両の運動が推定できることを確認した。

ステレオ動画画像を利用して、動的なシーンにおける物体上の各点の3次元位置と3次元モー션을推定するには、従来はステレオ画像間の対応点探索と時系列画像間のトラッキングを別々に行なう必要があった。そのため、アルゴリズムが複雑になったり、物体のモーションや見え方の変化などによって、推定精度が悪化する可能性があった。そこで、第6章「空間内の位置とモーションの同時推定」では、ステレオ画像と時系列画像間の対

応関係を同時に求めることによって、3次元位置と3次元モーションを直接的に、同時に最適化することができた。それによって、物体の奥行き方向の動きによる、画像中での見え方の変化を考慮して推定することが可能となった。また、画像中の各点ごとに独立して推定が行なえるため、あらかじめ同一運動する物体ごとにセグメンテーションをする必要がなかった。本手法の応用例として、障害物の3次元的なモーションが推定できることを示した。そのモーションは、第4章と第5章の結果を合わせることで、静止座標系における絶対モーションとして、推定を行なった。絶対モーションは、静止物体では速度が0となり、移動物体では速度を持つため、障害物の危険性の評価等に利用ができると考えられる。

これまで述べた方法は、主に同期ステレオカメラを想定した方法であった。しかし、市販されている安価なカメラを利用すると、多くの場合でステレオカメラ間で同時に撮影することができない、非同期ステレオカメラとなってしまふ。非同期ステレオでは、移動物体が画像中に写っていたり、カメラが移動していたりすると、各カメラで撮影されるタイミングの違いによって、対象の3次元位置が異なる。そのため、このままでは3次元位置やモーションを正しく推定することができない。このようなカメラを用いて、同期ステレオと同様にステレオ計測やモーション推定ができれば、安価で同期をとるための特別な装置の不要な、カメラを選ばないシステムが実現できる。そこで、第7章「非同期ステレオカメラを利用した試み」では、より一般的なカメラでも対応できるように、非同期ステレオカメラを利用した、3次元位置計測とモーション推定手法を提案した。本手法では、第6章の考え方を応用し、時系列非同期ステレオ画像間での対応を一度に求めることによって、画像中の各点ごとの3次元位置と3次元モーションを求めつつ、シーン全体の時間オフセットを同時に推定した。それにより、画像中に複数のモーションが含まれているような動的シーンにおいても、適用することができた。合成画像実験を通じて、本手法を用いた場合と従来方法との推定精度を比較した。実画像実験では、異種のカメラで撮影されたシーンの3次元復元とモーション推定を行なうことによって、本手法の有効性を確認した。

また、非同期ステレオカメラを利用した場合に、本論文で述べてきた道路環境認識に関する各手法がそのまま適用可能であるか、またはどのような仮定を設ければ適用できるのかについて考察を行なった。

8.2 今後の課題

全体的に計算速度の向上が必要であると考えている。本研究では、撮影された画像を車載した計算機に保存し、それらを研究室に持ち帰ってから処理を行なう、オフライン処理であった。これに対して、実際に車両にシステムを搭載し、その場で処理を行ない、その結果を確認することで、より多くのデータ処理が可能となり、これまで発見できなかつ

た問題が浮き彫りになったり、新たな用途が考案されたりすると期待される。そのためには、アルゴリズムの改良や計算並列化のような計算方法の工夫であったり、計算機の性能の高いものやGPUのようなハードウェアを利用したりして、各アルゴリズムの計算速度の向上方法を検討する必要がある。例えば、第3章の障害物検出処理では、 320×240 ピクセルのQVGA画像で1枚あたり、約1.9秒の計算時間がかかっているため、リアルタイムで処理を行なうためには、数倍計算速度を向上させる必要がある。

また、実環境における推定精度の検証方法を検討する必要があると考えている。第3、4章では、障害物までの実際の距離や障害物の横幅であるが、走行中にカメラから対象までの真の距離を正確に求めることは困難であると予想される。第5、6、7章の、モーション推定結果についても同様である。例えば、自車両の運動推定については、推定された運動を高精度なGPSなどと比較することで、検証ができると思われる。

以降では、各手法ごとの今後の課題について述べる。まず、3章の障害物検出手法では、時系列的な情報を処理に取り込む必要があると考えている。本研究では2眼ステレオカメラを利用したため、繰返しパターンが存在する場合には、ステレオ対応誤りが発生することがあった。そのため、異なった位置に物体が計測されたり、誤って検出されたりする場面があった。繰返しパターンによる影響は、カメラの配置と繰返しパターンの周期によって決まるが、自動車の場合は走行して対象までの距離が変化することで、そのパターンの周期が変わるため、シーケンス中の一部のフレームだけ誤検出が発生するといった現象が起こった。また、ステレオ対応誤りはオクルージョンの場合にも発生することがあった。このような問題を解決し、より障害物の検出率を上げつつ、誤検出率を下げるために、時系列情報を取り込む必要があると考えている。

第6章の3次元位置とモーションを同時に求める手法や、第7章の非同期カメラを利用した手法では、ステレオ画像間や時系列画像間に発生するオクルージョンに対する処理が必要であると考えている。また、本実験では初期値として画像内で一様としているが、他の方法である程度の初期値を推定しておけば、収束回数を早めることができ、計算時間を抑えられたり、ローカルミニマムに陥りにくくしたりできると考えられる。また、本論文では2眼ステレオカメラについて述べたが、ベースラインを複数持つような多眼カメラへ拡張し、初期値に対するロバスト性の向上や、推定精度の向上などが期待される。そして、本手法は、道路環境下で利用することを想定した車載画像処理向けの手法にとどまらず、一般的な環境下でステレオカメラからシーンの3次元位置と3次元モーションが求まるため、より広い使い方が期待される。今後はそのアプリケーションを多く考えていきたい。

参考文献

- [1] 安部正人. 自動車の運動と制御 (第2版). 山海堂, 2003.
- [2] T. Aizawa, A. Tanaka, Higashikage H, Y. Asokawa, M. Kimachi, and S. Ogata. Road surface estimation robust against vehicle's existence for stereo-based vehicle detection. In *Proc. Intelligent Transportation Systems*, pp. 43–48, 2002.
- [3] S. Baker, A. Datta, and T. Kanade. Parameterizing homographies. Technical Report CMU-RI-TR-06-11, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, March 2006.
- [4] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *Proc. IEEE CVPR*, Vol. 1, pp. 1090–1097, 2001.
- [5] M. Bertozzi and A. Broggi. Real-time lane and obstacle detection on the gold system. In *Proc. IEEE Intelligent Vehicles Symposium*, pp. 213–218, 1996.
- [6] J.Y. Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/.
- [7] A. Broggi, M. Bertozzi, A. Fascioli, C. LoBianco, and A. Piazzzi. Visual perception of obstacles and vehicles for platooning. *IEEE Trans. Intelligent Transportation Systems*, Vol. 1, No. 3, pp. 164–176, 2000.
- [8] N. Cornelis, K. Cornelis, and L. Van Gool. Fast compact city modeling for navigation pre-visualization. In *Proc. IEEE CVPR*, Vol. 2, pp. 1339–1344, 2006.
- [9] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, Vol. 29, No. 3, pp. 159–179, 1998.
- [10] O.D. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 2, No. 3, pp. 485–508, 1988.

- [11] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, Vol. 6, pp. 35–49, 1993.
- [12] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, Vol. 12, No. 1, pp. 16–22, 2000.
- [13] J. Goldbeck and B. Huertgen. Lane detection and tracking by video sensors. In *Proc. Intelligent Transportation Systems*, pp. 77–79, 1999.
- [14] M. Gong and Y.H. Yang. Near real-time reliable stereo matching using programmable graphics hardware. In *Proc. IEEE CVPR*, Vol. 1, pp. 924–931, 2005.
- [15] K.J. Hanna and N.E. Okamoto. Combining and motion analysis for direct estimation of scene structure. In *Proc. IEEE*, pp. 357–365, 1993.
- [16] H. Hao, H.S. Sawhney, and R. Kumar. Dynamic depth recovery from multiple synchronized video streams. In *Proc. IEEE CVPR*, Vol. 1, pp. 118–124, 2001.
- [17] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision Second Edition*. Cambridge University Press, 2003.
- [18] 林洋. 实用自動車事故鑑定工学. 技術書院, 1992.
- [19] 本田技術工業株式会社. <http://www.honda.co.jp/>.
- [20] Z. Hu and K. Uchimura. U-v-disparity: An efficient algorithm for stereovision based scene analysis. In *Proc. IEEE Intelligent Vehicles Symposium*, Vol. 1, pp. 48–54, 2005.
- [21] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. In *Proc. IEEE CVPR*, pp. 454–460, June 1994.
- [22] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, Vol. 82, No. Series D, pp. 35–45, 1960.
- [23] 川又正太, 片原俊司, 青木正喜. 車載スリットカメラと路面マークを用いた位置計測システムとその応用. 第8回画像センシングシンポジウム, pp. 71–76, 2002.
- [24] Q. Ke and T. Kanade. Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection. In *Proc. IEEE CVPR*, pp. 390–397, June 2003.
- [25] D. Keller, Q.T. Luong, and J. Malik. Using binocular stereopsis for vision-based vehicle control. In *Proc. IEEE Intelligent Vehicles Symposium*, pp. 237–242, 1994.

- [26] 窪田進, 仲野剛, 岡本恭一. 車載ステレオ障害物検出システムのための実時間処理可能な全体最適化アルゴリズム. 画像の認識・理解シンポジウム (MIRU), pp. 106–113, 2006.
- [27] R. Labayrade, D. Aubert, and J.P. Rarel. Real time obstacle detection in stereovision on non flat road geometry through “v-disparity” representation. In *Proc. IEEE Intelligent Vehicles Symposium*, Vol. 2, pp. 646–651, 2002.
- [28] S. Lee, M. Hariyama, and M. Kameyama. Processor architecture for road extraction based on projective transformation. In *Proc. SICE-ICASE*, pp. 1446–1450, 2006.
- [29] G. Lefaix, E. Marchand, and P. Boutheymy. Motion-base obstacle detection and tracking for car driving assistance. In *Proc. ICPR*, Vol. 4, pp. 74–77, 2002.
- [30] R. Mandelbaum, G. Salgian, and H. Sawhney. Correlation-based estimation of ego-motion and structure from motion and stereo. In *Proc. ICCV*, pp. 544–550, 1998.
- [31] W.V.D. Mark, D.E. Fontijne, and L. Dorst. Vehicle ego motion estimation with geometric algebra. In *Proc. IEEE Intelligent Transportation Systems*, pp. 58–61, 2002.
- [32] 森屋崇, 岩佐まもる, 石川直人, 中島直人. ステレオ画像を用いた市街地における車両衝突防止システム. 第8回画像センシングシンポジウム, pp. 77–82, 2002.
- [33] 中森卓馬, 石川直人, 中島真人. 動画像処理による車両前方監視システム. 電子情報通信学会技術研究報告, pp. 1–6, 2002.
- [34] 中山啓満, 窪田進, 谷口恭弘, 小野口一則. 逆投影画像上での候補追跡処理による白線検出. 電子情報通信学会技術研究報告, Vol. 101, pp. 15–22, 2001.
- [35] NEC エレクトロニクス. <http://www.necel.com/>.
- [36] R. Okada, Y. Taniguchi, K. Furukawa, and K. Onoguchi. Obstacle detection using projective invariant and vanishing lines. In *Proc. ICCV*, pp. 330–337, 2003.
- [37] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 4, pp. 353–363, 1993.
- [38] M. Okutomi, K. Nakano, J. Maruyama, and T. Hara. Robust estimation of planar regions for visual navigation using sequential stereo images. In *Proc. IEEE ICRA*, pp. 3321–3327, 2002.

- [39] 奥富正敏, 中野勝之, 丸山純一, 原智章. ステレオ動画像を用いた視覚誘導のための平坦部の連続推定. 情報処理学会論文誌, Vol. 43, No. 4, pp. 1061–1069, 2002.
- [40] M. Okutomi and W. Noguchi. Extraction of road region using stereo images. In *Proc. ICPR*, pp. 853–856, 1998.
- [41] 奥富正敏, 野口卓, 中野勝之. ステレオ画像からの射影変換行列の抽出による道路領域検出. 日本ロボット学会誌, Vol. 18, No. 8, pp. 1105–1111, 2000.
- [42] 奥富正敏, 関晃仁. ステレオ画像を用いた道路平面領域並びに障害物検出方法. 国際出願番号 PCT/JP2005/009701.
- [43] 奥富正敏, 関晃仁. ステレオ画像を用いた道路平面領域並びに障害物検出方法. 特願 2004-023493, 2004/1/30.
- [44] F. Paetzold and U. Franke. Road recognition in urban environment. *Int. Journal on Image and Vision Computing*, Vol. 18, No. 5, pp. 377–387, 2000.
- [45] S.K. Park and I.S. Kweon. Robust and direct estimation of 3-d motion and scene depth from stereo image sequences. *Pattern Recognition*, Vol. 34, No. 9, pp. 1713–1728, 2001.
- [46] M. Pollefeys, R. Koch, and L. Van Gool. A simple and efficient rectification method for general motion. In *Proc. ICCV*, pp. 496–501, 1999.
- [47] デジタル画像処理編集委員会 (編). デジタル画像処理. 財団法人画像情報教育振興協会, 2004.
- [48] 実吉敬二, 埴圭二, 十川能之, 荒井一真. ステレオ画像を用いた運転者支援のための前方状況認識システム. 電子情報通信学会技術研究報告, Vol. 30, pp. 39–46, 1997.
- [49] 関晃仁, 奥富正敏. ステレオ動画像を利用した道路面パターン抽出による自車両の運動推定. 情処学研報 (2004-CVIM-146-13), Vol. 2004, No. 113, pp. 17–24, 2004.
- [50] 関晃仁, 奥富正敏. ステレオ動画像を利用した平面抽出による障害物検出. 情処学研報 (2004-CVIM-143-3), Vol. 2004, No. 26, pp. 17–24, 2004.
- [51] 関晃仁, 奥富正敏. ステレオ動画像を利用した平面抽出による障害物検出. 情報処理学会論文誌, Vol. 45, No. SIG 13(CVIM10), pp. 1–10, 2004.
- [52] 関晃仁, 奥富正敏. 平面の検出・姿勢推定を利用した一般道路環境下におけるロバストな障害物検出. 信学技報, Vol. 105, No. 259, pp. 13–19, 2005.

- [53] A. Seki and M. Okutomi. Ego-motion estimation by matching dewarped road regions using stereo images. In *Proc.ICRA*, Vol. Vol.1, pp. 901–907, 2006.
- [54] A. Seki and M. Okutomi. Robust obstacle detection in general road environment based on road extraction and pose estimation. In *Proc. IEEE Intelligent Vehicles Symposium*, Vol. Vol.1, pp. 437–444, 2006.
- [55] 関晃仁, 奥富正敏. ステレオ動画像を用いた動的シーンのモーションと奥行きを同時推定. 画像の認識・理解シンポジウム (MIRU), pp. 352–357, 2006.
- [56] 関晃仁, 奥富正敏. ステレオ動画像を利用した道路領域の抽出と追跡による自車両の運動推定. 情報処理学会論文誌, Vol. 47, No. SIG 5(CVIM10), pp. 90–99, 2006.
- [57] 関晃仁, 奥富正敏. 車載ステレオカメラによる自車両の運動推定. 画像ラボ, 日本工業出版, Vol. 17, No. 10, pp. 37–41, 10 2006.
- [58] 関晃仁, 奥富正敏. 道路面の抽出・姿勢推定を元にした一般道路環境下におけるロバストな障害物検出. 電子情報通信学会論文誌, Vol. J89-D, No. 8, pp. 1859–1868, 2006.
- [59] 関晃仁, 奥富正敏. 平面の検出・姿勢推定に基づく一般道路環境下におけるロバストな障害物検出. 第12回画像センシングシンポジウム, pp. 370–373, 2006.
- [60] 関晃仁, 奥富正敏. 非同期ステレオ動画像を用いた動的シーンの位置とモーションの同時推定. 情処学研報 (2007-CVIM-157), Vol. 2007, No. 1, pp. 177–184, 2007.
- [61] 関晃仁, 奥富正敏, 藤巻朋. 移動体周辺監視装置. 国際出願番号 PCT/JP2005/14782.
- [62] 関晃仁, 奥富正敏, 藤巻朋. 物体検出装置. 国際出願番号 PCT/JP2005/14785.
- [63] 関晃仁, 奥富正敏, 藤巻朋. 移動体移動量算出装置. 特願 2004-234945, 2004/8/11.
- [64] 関晃仁, 奥富正敏, 藤巻朋. 移動体周辺監視装置. 特願 2004-234946, 2004/8/11.
- [65] 関晃仁, 奥富正敏, 藤巻朋. 物体検出装置. 特願 2004-234947, 2004/8/11.
- [66] J. Shi and C. Tomasi. Good features to track. In *Proc. IEEE CVPR*, pp. 593–600, June 1994.
- [67] 清水雅夫. 画像のサブピクセルマッチングにおける誤差解析と高精度推定手法の提案. PhD thesis, 東京工業大学大学院情報理工学研究科情報環境学専攻, 2003.
- [68] 清水彰一, 藤吉弘亘. カメラ間のシャッタータイミングのずれを利用した高速3次元位置推定. 画像の認識・理解シンポジウム (MIRU), pp. 428–433, July 2004.

- [69] S. Shimizu, H. Fujiyoshi, Y. Nagasaka, and T. Takahashi. A pseudo stereo vision method for unsynchronized cameras. In *Proc. ACCV*, Vol. Vol.1, pp. 575 – 580, 2004.
- [70] H.Y. Shum and R. Szeliski. Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, Vol. 36, No. 2, pp. 101–130, 2000.
- [71] G. Stein, O. Mano, and A. Shashua. A robust method for computing vehicle ego-motion. In *Proc. IEEE Intelligent Vehicles Symposium*, pp. 362–368, 2000.
- [72] G.P. Stein and A. Shashua. Direct estimation of motion and extended scene structure from a moving stereo rig. In *Proc. IEEE CVPR*, pp. 211–218, 1998.
- [73] 杉本茂樹, 奥富正敏. ステレオ画像を利用した平面姿勢推定手法と多眼カメラへの拡張. 情処学研報 (2005-CVIM-151), Vol. 2005, No. 112, pp. 131–138, 2005.
- [74] T. Suzuki and T. Kanade. Measurement of vehicle motion and orientation using optical flow. In *Proc. IEEE Intelligent Transportation Systems*, pp. 25–30, 1999.
- [75] M. Svedman, L. Goncalves, N. Karlsson, M. Munich, and P. Pirjanian. Structure from stereo vision using unsynchronized cameras for simultaneous localization mapping. In *Proc. IROS*, pp. 2 – 6, 2005.
- [76] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, Vol. 9, No. 2, pp. 137–154, 1992.
- [77] トヨタ自動車株式会社. <http://www.toyota.co.jp/>.
- [78] T. Tsuji, H. Hattori, M. Watanabe, and N. Nagaoka. Development of night-vision system. *IEEE Transactions on Intelligent Transportation System*, Vol. 3, No. 3, pp. 203–209, 2002.
- [79] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 3, pp. 475–480, 2005.
- [80] O. Veksler. Fast variable window for stereo correspondence using integral images. In *Proc. IEEE CVPR*, pp. 556–564, 2003.
- [81] T. Williamson. A high-performance stereo vision system for obstacle detection. Robotics Institute, Carnegie Mellon University CMU-RI-TR-98-24, September 1998.

- [82] T. Williamson and C. Thorpe. Detection of small obstacles at long range using multibaseline stereo. In *Proc. IEEE Intelligent Vehicles Symposium*, pp. 311–316, 1998.
- [83] T. Williamson and C. Thorpe. A specialized multibaseline stereo technique for obstacle detection. In *Proc. IEEE CVPR*, pp. 238–244, 1998.
- [84] J. Woetzel and R. Koch. Real-time multi-stereo depth estimation on gpu with approximative discontinuity handling. In *Proc. European Conference on Visual Media Production (CVMP)*, pp. 245–254, 2004.
- [85] 山口晃一郎, 加藤武男, 二宮芳樹. 車載単眼カメラによる車両前方の障害物検出. 情処学研報 (2005-CVIM-151), Vol. 2005, No. 112, pp. 69–76, 2005.
- [86] J.W. Yi and J.H. Oh. Estimation of depth and 3d motion parameter of moving object with multiple stereo images. *Image and Vision Computing*, Vol. 14, No. 7, pp. 501–516, 1996.
- [87] 吉沢博一, 松田一朗, 伊東晋. 道路領域の予測に基づいた車載カメラ映像からの車両検出. 電子情報通信学会技術研究報告, Vol. 101, pp. 21–26, 2001.
- [88] Q. Yu, H. Araujo, and H. Wang. Stereo-vision based real time obstacle detection for urban environments. In *Proc. International Conference on Advanced Robotics*, pp. 1671–1676, 2003.
- [89] C. Yuan and G. Medioni. 3d reconstruction of background and objects moving on ground plane viewed from a moving camera. In *Proc. IEEE CVPR*, Vol. 2, pp. 2261 – 2268, 2006.
- [90] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientation. In *Proc. IEEE ICCV*, 1999.
- [91] C. Zhou and H. Tao. Dynamic depth recovery from unsynchronized video streams. In *Proc. IEEE CVPR*, Vol. 2, pp. 351–358, 2003.
- [92] 古川賢司, 岡田隆三, 谷口恭弘, 小野口一則. 車載用画像処理 LSI を用いた車両周辺監視システム. 第 9 回画像センシングシンポジウム, pp. 227–232, 2003.

謝辞

本研究を進めるにあたり御指導頂いた東京工業大学 理工学研究科 奥富 正敏 教授に心より感謝致します。先生は本学位論文審査員の主査であり、研究室に所属して以来長きに渡って、研究の仕方やまとめ方、論文の書き方を始め、さまざまな御指導を承りました。本研究には先生の数々の発想やアイデアが生かされており、多岐に渡る御指導がなかったら、本研究もまとまっていなかったと思います。審査員の理工学研究科 機械制御システム専攻 北川 能 教授，同 藤田 政之 教授，同 大山 真司 助教授，同 山北 昌毅 助教授に感謝致します。審査員の先生方には、本論文をまとめるにあたり、適切なアドバイスを頂きました。

車載関連の実験装置の準備や画像の取得に際し、御協力いただいた(株)本田技術研究所 栃木研究所の飯星明氏，藤巻朋氏，植田恭史氏に感謝致します。

ビジュアルナビゲーション班と一緒に研究し、平面抽出処理プログラムを提供していただいた杉本 茂樹 研究員に感謝致します。また、修士の田中 健一 氏，学士の前田 敦史 氏，八重田 岳 氏には、同じ研究班として、有益な御討論を頂きました。そして、研究成果を残してくださった先輩の方々に感謝致します。

最後になりましたが、研究に関するアドバイスや発想の種子を提供して頂いたり、日々の気分転換の会話におつきあい頂いた奥富研究室の皆様へ深く感謝致します。そして、これまでご支援下さいました両親に心よりお礼申し上げます。

付録A 実験環境

車両の前方環境認識を行うための装置として、カメラを2台利用した。車載ステレオカメラは、図 A.1 に示すように車両の前方に向けて、2台をほぼ平行になるように設置した。道路面に対するカメラの俯角は約 5 [degree]、道路面からカメラまでの距離は約 1.25 [m] であり、走行中は車両の振動等により変化する。設置後の様子を図 A.2 に示す。ステレオカメラは、Sony 社製 XC003 や Point Gray 社製 Flea を利用した。XC003 や Flea のように通常の 8bit カメラと、実験の一部ではよりダイナミックレンジの高い 16bit カメラ (High

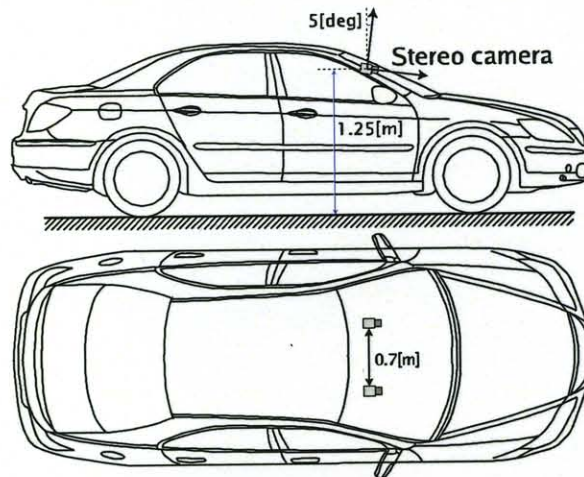


図 A.1: 実験車両
Fig. A.1: Experimental vehicle.

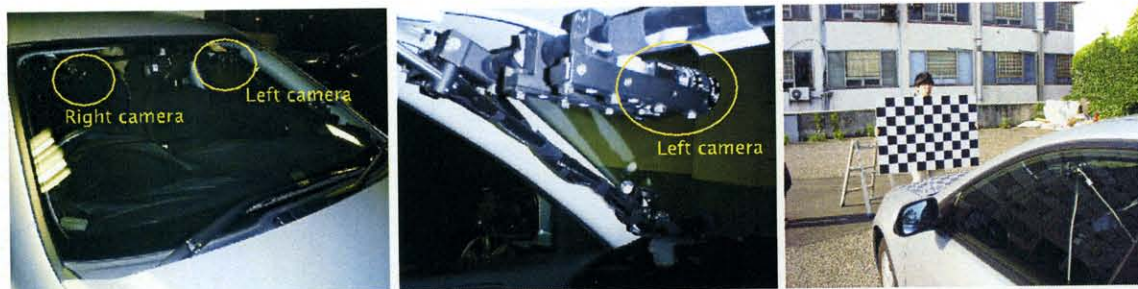


図 A.2: 実験機材 (左側: 外からみた様子, 右側: 車内の様子) 図 A.3: キャリブレーション風景
Fig. A.2: Experimental equipment.

Fig. A.3: Calibration operation.

Dynamic Range :HDR カメラ) も利用した。本研究の実験では、オフライン処理を行ったため、車載カメラで撮影された画像を車載した計算機に保存し、それを持ち帰って各種処理に利用した。

カメラのキャリブレーションは、実験前に図 A.3 に示すようにキャリブレーションターゲットを撮影し、Zhang ら [90] の手法に基づく Matlab Toolbox[6] を利用することで、カメラの内部パラメータおよび歪みパラメータ、外部パラメータを推定した。

付録B Rectification

ステレオ画像が与えられたとき、図 B.1 に示すように Rectification 処理を行なうことで、ステレオ画像の画像面 (同図赤色の平面) を同一面 (同図青色の平面) にし、かつエポピーラ線を平行に揃えることができる。これによって、ステレオ対応点探索をする際に、基準画像の注目点ごとのエポピーラ線を個々に計算しなくても、注目点に対してその水平線 (または垂直線) がエポピーラ線となる。それにより、ステレオ対応点探索を簡単に行なうことができる。

本論文で用いた Rectification は、Fusiello らの手法 [12] を利用した。 $\hat{m} = (u, v, 1)^T$ を同次座標系における画像上の位置とし、 $w = (x, y, z)^T$ をワールド座標系における 3次元座標位置、 \hat{w} はそれを同次座標系で表したものとする。このとき、透視投影行列を P とすると、

$$\hat{m} \simeq Pw \tag{B.1}$$

として表現することができる。 \simeq は、定数倍を許して等しいことを表している。

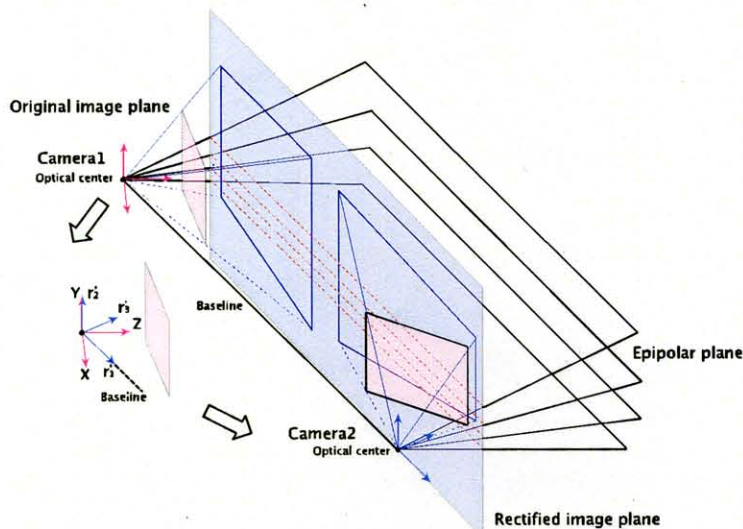


図 B.1: エポピーラ幾何と Rectification
Fig. B.1: Epipolar geometry and rectified image plane.

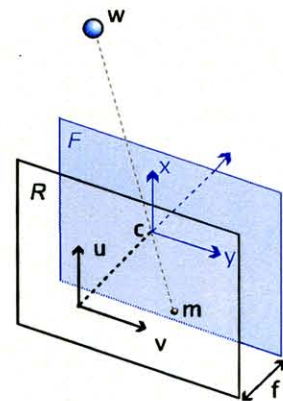


図 B.2: 画像面と焦点面
Fig. B.2: Image plane and focal plane.

透視投影行列 \mathbf{P} は、3行4列の行列であることから

$$\mathbf{P} = \left[\begin{array}{c|c} \mathbf{q}_1^T & q_{14} \\ \mathbf{q}_2^T & q_{24} \\ \mathbf{q}_3^T & q_{34} \end{array} \right] = [\mathbf{Q}|\tilde{\mathbf{q}}] \quad (\text{B.2})$$

と表現でき、画像座標は

$$\begin{cases} u = \frac{\mathbf{q}_1^T \mathbf{w} + q_{14}}{\mathbf{q}_3^T \mathbf{w} + q_{34}} \\ v = \frac{\mathbf{q}_2^T \mathbf{w} + q_{24}}{\mathbf{q}_3^T \mathbf{w} + q_{34}} \end{cases} \quad (\text{B.3})$$

と表せる。このとき、光学中心の3次元座標位置を \mathbf{c} とすると、光学中心は焦点面上に存在するため、

$$\mathbf{q}_3^T \mathbf{c} + q_{34} = 0 \quad (\text{B.4})$$

が成り立ち、かつ図 B.2 に示すように光学中心で画像面の垂直・水平軸とも交わることから、

$$\mathbf{q}_1^T \mathbf{c} + q_{14} = 0 \quad \text{かつ} \quad \mathbf{q}_2^T \mathbf{c} + q_{24} = 0 \quad (\text{B.5})$$

が成り立つ。これによって、

$$\mathbf{P} \begin{bmatrix} \mathbf{c} \\ 1 \end{bmatrix} = \mathbf{0} \quad (\text{B.6})$$

となるので、式を整理すると、

$$\mathbf{c} = -\mathbf{Q}^{-1}\tilde{\mathbf{q}} \quad (\text{B.7})$$

が得られる。

これにより、式(B.2)は

$$\mathbf{P} = [\mathbf{Q}|\tilde{\mathbf{q}}] = [\mathbf{Q} | -\mathbf{Q}\mathbf{c}] \quad (\text{B.8})$$

となる。

さて、透視投影行列 \mathbf{P} は、カメラの内部パラメータ行列 \mathbf{A} と、外部パラメータである回転行列 \mathbf{R} と、並進ベクトル \mathbf{t} から成り立っており、内部パラメータと外部パラメータを用いることにより、 \mathbf{P} を

$$\mathbf{P} = \mathbf{A}[\mathbf{R}|\mathbf{t}] \quad (\text{B.9})$$

と表現できる¹。ここで、式(B.8)と式(B.9)を考慮することで、透視投影行列は

$$\mathbf{P} = \mathbf{A}[\mathbf{R}] - \mathbf{R}\mathbf{c} \quad (\text{B.10})$$

となる。

Rectificationによって、(1)ステレオ画像のそれぞれの画像面を同一平面に乗せ、かつ(2)エピポーラ線を平行にすることから、(1)に対してはステレオカメラの焦点距離を一致させ、(2)はワールド座標からの回転行列をステレオカメラ間で同じにし、かつその回転行列のうちの1つの軸はベースラインに平行になるように決めればよい。そこで、Rectification後の透視投影行列を

$$\mathbf{P}'_b = \mathbf{A}'[\mathbf{R}'] - \mathbf{R}'\mathbf{c}_b \quad (\text{B.11})$$

$$\mathbf{P}'_r = \mathbf{A}'[\mathbf{R}'] - \mathbf{R}'\mathbf{c}_r \quad (\text{B.12})$$

とおく。A'の内部パラメータは、任意に決定することができるため、例えば、ステレオカメラの内部パラメータのうちの1つを利用すればよい。c_b, c_rは、Rectification前の透視投影行列から求めることができる。続いて、R'を

$$\mathbf{R}' = \begin{bmatrix} \mathbf{r}'_1{}^T \\ \mathbf{r}'_2{}^T \\ \mathbf{r}'_3{}^T \end{bmatrix} \quad (\text{B.13})$$

とすると、図B.1の左側に示すように、Rectification後のX軸は、ベースラインベクトルと平行であるので、

$$\mathbf{r}'_1 = \frac{\mathbf{c}_1 - \mathbf{c}_2}{\|\mathbf{c}_1 - \mathbf{c}_2\|} \quad (\text{B.14})$$

で与えられる。Y軸は、X軸(r'₁)と任意の単位ベクトルk(例えば一方のカメラのRectification前のZ軸)の外積によって

$$\mathbf{r}'_2 = \mathbf{k} \times \mathbf{r}'_1 \quad (\text{B.15})$$

として与えられる。最後にZ軸は、X軸とY軸の外積で

$$\mathbf{r}'_3 = \mathbf{r}'_1 \times \mathbf{r}'_2 \quad (\text{B.16})$$

として求められる。

次に、画像をRectification後の画像へ変換する方法について説明する。まず、カメラ1のRectification前の透視投影行列をP_{o1}とし、3次元位置 $\tilde{\mathbf{w}}$ が写る画像位置を $\tilde{\mathbf{m}}_{o1}$ とすると、

$$\tilde{\mathbf{m}}_{o1} \simeq \mathbf{P}_{o1}\tilde{\mathbf{w}} = [\mathbf{Q}_{o1}|\tilde{\mathbf{q}}_{o1}]\tilde{\mathbf{w}} \quad (\text{B.17})$$

¹もしPが既知でA, R, tを求めたいときには、因子分解をすればよい。

と表される. 式 (B.8) より,

$$\tilde{\mathbf{m}}_{o1} \simeq [\mathbf{Q}_{o1} | \tilde{\mathbf{q}}_{o1}] \begin{bmatrix} \mathbf{w} \\ 1 \end{bmatrix} = [\mathbf{Q}_{o1} | -\mathbf{Q}_{o1}\mathbf{c}_1] \begin{bmatrix} \mathbf{w} \\ 1 \end{bmatrix} \quad (\text{B.18})$$

この式を整理すると,

$$\mathbf{w} = \mathbf{c}_1 + \lambda_o \mathbf{Q}_{o1}^{-1} \tilde{\mathbf{w}} \quad (\text{B.19})$$

となる. λ_o は任意の定数である. 同様に, Rectification 後の透視投影行列を \mathbf{P}_{n1} とし, 次元位置 $\tilde{\mathbf{w}}$ が写る画像位置を $\tilde{\mathbf{m}}_{n1}$ とすると,

$$\tilde{\mathbf{m}}_{n1} \simeq \mathbf{P}_{n1} \tilde{\mathbf{w}} = [\mathbf{Q}_{n1} | \tilde{\mathbf{q}}_{n1}] \tilde{\mathbf{w}} \quad (\text{B.20})$$

と表され,

$$\mathbf{w} = \mathbf{c}_1 + \lambda_n \mathbf{Q}_{n1}^{-1} \tilde{\mathbf{w}} \quad (\text{B.21})$$

が得られる. 式 (B.19) と式 (B.21) より,

$$\tilde{\mathbf{m}}_{n1} = \lambda \mathbf{Q}_{n1} \mathbf{Q}_{o1}^{-1} \tilde{\mathbf{m}}_{o1} \simeq \mathbf{H}_1 \tilde{\mathbf{m}}_{o1} \quad (\text{B.22})$$

となり, 3行3列の射影変換行列 \mathbf{H}_1 が Rectification 画像へ変換するための変換行列となる. カメラ2についても同様に, 射影変換行列 \mathbf{H}_2 を求めて, 画像を変換すればよい.

この手法では, 例えば単眼車載カメラで直進している時系列ステレオ画像のように, 画像中に光学中心が含まれている場合には, 適用できない. このように, Rectification 画像への変換が射影変換行列では表せない場合には, Pollefeys らの手法 [46] を利用すればよい. この手法では, Rectification 後のカメラは透視投影モデルではなくなるため, 3次元位置を算出するには, 一度 Rectification 前の座標系へ戻すなどの手順が必要である.

付録C 運動推定のフィルタリング

C.1 “Roughness” と分散の関係

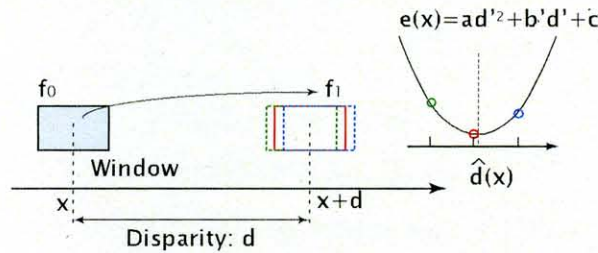


図 C.1: 1次元マッチングとパラボラフィッティング
Fig. C.1: One dimensional matching and parabola fitting.

本節では、カルマンフィルタで用いる“Roughness”と分散の関係について、最初に1次元のマッチングを考え、次に2次元の場合について述べる。まず、図C.1に示すようにSSDを類似度評価関数として2枚の画像間のマッチングをとり、視差 \hat{d} を求めるには、

$$e(\tilde{d}; x) = \int w(\lambda) [f_1(x + \tilde{d} + \lambda) - f_0(x + \lambda)]^2 d\lambda \quad (C.1)$$

を最小にする位置 \hat{d} を求めればよいため、

$$e(\hat{d}; x) = \min_{\tilde{d}} e(\tilde{d}; x)$$

を推定する問題になる [37]。視差 d に関して各画像に投影されるパターンは同じであり、ノイズが含まれていると仮定すると、

$$f_0(x) = f(x) + n_0(x), \quad f_1(x + d) = f(x) + n_1(x) \quad (C.2)$$

と表すことができる。式(C.2)を式(C.1)に代入すると、

$$e(\tilde{d}; x) = \int w(\lambda) [f(x + \tilde{d} - d + \lambda) - f(x + \lambda) + n_1(x + \lambda) - n_0(x + \lambda)]^2 d\lambda \quad (C.3)$$

が得られる。次に $\tilde{d} \simeq d$ として、上式をテーラー展開すると、

$$e(\tilde{d}; x) = a(x)(\tilde{d} - d)^2 + 2[b_1(x) - b_0(x)](\tilde{d} - d) + c(x) \quad (C.4)$$

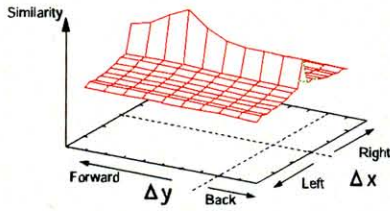


図 C.2: 推定成功時の例
Fig. C.2: Succeeded case.

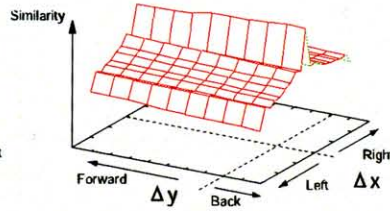


図 C.3: 推定失敗時の例
Fig. C.3: Failed case.

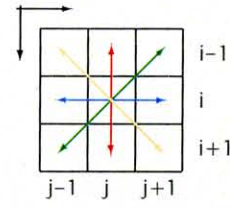


図 C.4: 2次元マッチング時のパラボラ方向の選択
Fig. C.4: Parabola direction of 2D matching case.

ここで,

$$\begin{aligned} a(x) &= \int w(\lambda)[f'(x+\lambda)]^2 d\lambda \\ b_i(x) &= \int w(\lambda)f'(x+\lambda)n_i(x+\lambda)d\lambda \\ c(x) &= \int w(\lambda)[n_1(x+\lambda) - n_0(x+\lambda)]^2 d\lambda \end{aligned} \quad (C.5)$$

であり, $a(x), b_1(x), b_0(x), c(x)$ は, $e(\hat{d}; x)$ の形状を決める要素であり, a が類似度分布に当てはめらるパラボラの“Roughness”である.

さて, 推定したい視差 \hat{d} は, 式 (C.4) を 1 次微分して 0 にすればよいので,

$$\hat{d} = d + \frac{b_0(x) - b_1(x)}{a(x)} \quad (C.6)$$

となる. b_i の分散は,

$$\text{Var}(b_i(x)) = \sigma_n^2 \int w^2(x)[f'(x+\lambda)]^2 d\lambda \quad (C.7)$$

$w(x) = 1$ とすると, $\sigma_n^2 a(x)$ となるので, 分散 $\text{Var}(\hat{d})$ は,

$$\text{Var}(\hat{d}) = \frac{2\sigma_n^2}{a(x)} \quad (C.8)$$

となる. つまり, 画像間の類似度分布から得られるパラボラの形状である, $a(x)$ と誤差分散には反比例の関係があることがわかる.

次に, 今回の運動推定の場合に準じ, 2次元に拡張した場合について考える. VPP 画像同士のマッチングで, 正しく対応を取ることができた場合とできなかった場合について, その類似度をプロットしたものが図 C.2 と図 C.3 である. 図 C.2 は, 正しくできた場合であり, 図 C.3 は, 誤った場合である. 類似度分布が一定に近い状態となってしまったことで, 正しく対応がとれなかったと考えられる.

2次元の類似度空間から, カルマンフィルタで用いる“Roughness”を算出する. “Roughness”は, 図 C.4 に示すように縦横斜めの合計 4 つが算出されるが, 最も小さいものをカルマンフィルタの観測誤差分散へ反映させる.

以下に，“Roughness”の算出方法を述べる。類似度とその位置を利用して，2次関数

$$y = a_2x^2 + a_1x + a_0 \quad (\text{C.9})$$

に最小二乗フィッティングさせる。 x は類似度を計算した位置， y は類似度に相当する。 a_i は係数である。そこで，各位置 $x_1 = -1, x_2 = 0, x_3 = 1$ とその位置における類似度 y_i を以下のように行列でまとめる。

$$\mathbf{X} = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{bmatrix}, \quad \mathbf{Y} = [y_1 \ y_2 \ y_3]^T. \quad (\text{C.10})$$

以下の計算により，2次関数の係数を求める。

$$\mathbf{a} = [a_2 \ a_1 \ a_0]^T = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \quad (\text{C.11})$$

ここで， a_2 が “Roughness” である。続いて縦，横方向に “Roughness” を計算し，小さい方を保存する。同様に，斜め方向のペアについても， $x_1 = -\sqrt{2}, x_2 = 0, x_3 = \sqrt{2}$ として “Roughness” をそれぞれ計算し，小さい方を保存する。最後に，算出された縦横方向と斜め方向の “Roughness” のうち，小さい方をフィルタリングに用いる “Roughness” とする。