

マシンビジョン技術による交通流計測

慶應義塾大学大学院 理工学研究科 開放環境科学専攻

2007 年 3 月

安倍 満

要 旨

交通流計測は ITS(Intelligent Transport Systems) における基礎技術の一つである。現在では超音波式やループコイル式の局所型センサによって車両の通過台数や速度情報の収集が行われており、渋滞緩和のための信号制御や到達時間予測に利用されている。今後、交通流計測システムは事故車両や違反車両検知システムを統合したものへと拡張されていくことが期待されているが、局所型センサでは本質的に情報量が不足しているため、カメラをセンサとして用いる広域型センサが新たに注目を浴びるようになった。広域型センサからは多くの情報を取得でき、単一のセンサで広範囲に渡って車両の振る舞いを解析することができる。そのため局所型センサの欠点を補うものとして有望視されている。

そこで本論文では、道路上の監視カメラを用いた、低コストで高性能な交通流計測システムを開発することを目的とする。本論文で提案する交通流計測手法は、次に示す5つの性質、すなわち「1. 1 台のモノクロカメラで計測が可能」「2. カメラの設置位置に関する制約が緩い」「3. 照明条件の変化に強い」「4. 車両の移動軌跡を取得可能」「5. オクルージョンに強い」を有する。

本手法は車両の追跡の問題を、車両から抽出した特徴点の軌跡群をクラスタリングする問題に帰着する。特徴点とは濃淡画像からも抽出できる基本的な特徴量であり、明度変化に比較的強いという性質を持つ。車両のような人工物からは、どの撮影方向においても充分多くの特徴点を検出できるため、交通流計測には適切な特徴量である。そこで車両から特徴点を抽出し、背景画像との正規化相関係数をもとに背景特徴点を除去する。残った点を車両特徴点とし、それぞれを時間方向に追跡することで軌跡群を得る。そして任意の2つの軌跡同士の類似度を算出し、これに基づいてクラスタリングを行う。ここでいう類似度とは、2つの軌跡の組み合わせごとに定義されるスカラー量であり、それら2つの軌跡が同一の車両から検出されたものであるか否かを評価する尺度である。この類似度を、画像のエッジ情報および画像平面上における軌跡同士の位置関係から求める。

複数の要素間に重みを持った連結関係がある場合、それらの構造を重み付き完全グラフとしてみなすことができる。そこで、グラフ分割アルゴリズムを適用し、辺重みの弱い部分で分割することで、個々の車両の位置を得る。グラフ分割アルゴリズムは、グラフの大局的な性質を示す評価値を最大化するような分割を能率よく探索するというものであり、局所的なエラーの影響を受けにくいという利点がある。この分割の際に、過去のフレームにおける結果を拘束条件として与え、解空間を制約することで、オクルージョンに強い追跡を実現する。

実験により、筆者が提案した手法は異なる時間帯・道路状況に対しても、システムのしきい値を調整することなく適用できることを示した。この結果により、従来手法にはない耐環境性能を持つ交通流計測システムを提案できたものといえる。交通流計測は多くの地点で長時間行われるべきものであり、本手法の目指す方向性は、交通流量の把握、事故・違法車両検知を目的とした道路交通流監視システムの実現へと繋がってゆくと考えられる。

目次

第 1 章	序論	1
1.1	近年の交通事情と高度交通システム (ITS)	1
1.2	ITS におけるマシンビジョン技術	2
1.2.1	道路インフラにおけるマシンビジョン	3
	交通流計測	3
	突発事象	4
	道路側からの運転支援	4
1.2.2	車載カメラにおけるマシンビジョン	5
	前方・後方障害物監視	5
	駐車支援	6
	運転手監視	6
1.3	総括	7
第 2 章	交通流計測とマシンビジョン	9
2.1	交通流計測の現状とマシンビジョンの利点	9
2.2	関連研究	10
2.2.1	テンプレートマッチングによる方法	10
2.2.2	領域に基づく方法	10
2.2.3	時空間画像による方法	11
2.2.4	2 次元および 3 次元の形状モデルによる方法	11
2.2.5	明度・色ヒストグラムを用いる方法	12
2.2.6	Condensation を用いる方法	12
2.2.7	特徴量のクラスタリングによる方法	13
2.2.8	特殊なカメラによる方法	13
第 3 章	本論文の目的および関連研究に対する位置付け	15
3.1	本論文の目的	15

3.2	従来研究における位置付け	16
3.3	本論文の構成	18
第 4 章	特徴点群抽出とグラフ分割を用いた車両認識	19
4.1	概要	19
4.2	適用範囲	20
4.3	車両認識の原理	21
4.4	グラフ構築	24
4.4.1	特徴点検出処理	24
4.4.2	軌跡取得処理	28
	軌跡の定式化	28
	動的計画法による軌跡の取得	30
4.4.3	類似度の算出	32
4.4.4	グラフの構築	34
4.5	グラフ分割	35
4.5.1	軌跡群のクラスタリング	35
4.5.2	グラフ分割の解法	36
	Tabu Search による 2 分割	36
	GSA による k 分割への拡張	37
4.6	実験	38
4.6.1	実験条件	38
4.6.2	実験結果と検討	38
	撮影位置に関するロバスト性	39
	照明条件に関するロバスト性	48
	グラフ分割に関する検討	48
	実行速度の検討	49
4.7	総括	51
第 5 章	動画像を入力情報とする車両計数	53
5.1	概要	53
5.2	車両計数の原理	54
5.3	特徴点の軌跡群抽出	55
5.3.1	車両特徴点抽出	55
5.3.2	車両特徴点の軌跡	57
5.4	軌跡群のクラスタリングによる追跡	59

5.4.1	軌跡間の類似度	59
	エッジによる類似度 E	60
	距離による類似度 D	62
5.4.2	グラフの定義と分割法	62
5.4.3	グラフ分割による追跡	63
	累積スコアによる車両の判定	64
	重みの修正	64
	V_t^i と V_{t-1}^j の対応付け	65
5.5	実験	66
5.5.1	対象シーン	66
5.5.2	しきい値の決定	70
5.5.3	実験結果	71
5.6	実験結果の検討	72
5.6.1	追跡精度に関する検討	72
5.6.2	特徴点の対応付けについて	72
5.6.3	グラフの辺重み修正による効果	73
5.6.4	追跡エラーの要因	74
5.6.5	本手法の適用可能な条件について	74
5.6.6	実行速度	75
5.7	総括	75
第 6 章	重交通流における車両計数	77
6.1	概要	77
6.2	オクルージョンに強い車両計数の原理	78
6.3	特徴点の軌跡群取得	81
6.3.1	車両特徴点抽出	81
6.3.2	車両特徴点の軌跡	81
6.4	軌跡間の類似度算出	82
6.4.1	エッジによる類似度 E	82
6.4.2	距離による類似度 D	83
6.4.3	前フレームの分割結果を用いた類似度の修正	83
6.5	拘束付きグラフ分割による車両追跡	84
6.5.1	問題の定義	84
6.5.2	大まかな流れ	85

6.5.3	拘束点群の選択	86
6.5.4	拘束付き Normalized Cuts	86
	焼きなまし法による分割	87
	再分割	88
6.6	実験	88
6.6.1	計測結果	91
6.6.2	近接する複数車両の分離	92
	離れていた二台が重なり合う場合	92
	重なり合っていた二台が離れていく場合	92
6.6.3	実行速度	95
6.7	総括	95
第 7 章	その他の応用事例	97
7.1	対象物体追跡のフレームワーク	97
7.2	応用事例 1：テニスの選手追跡	98
7.3	応用事例 2：車載カメラによる後方／前方車両監視	99
7.4	応用事例 3：監視用ステレオカメラによる侵入者検知	100
7.5	総括	102
第 8 章	結論	103
付録 A	特徴点	107
A.1	Harris 作用素	107
A.2	Harris 作用素の変形	107
付録 B	Normalized Cuts	109
B.1	k-means 法とグラフ分割	109
B.2	グラフ分割の定義	110
B.3	Multiclass normalized cuts	111

目次

1.1 ITS におけるマシンビジョン技術	3
4.1 入力画像	21
4.2 夜間における特徴点抽出	22
4.3 クラスタリングによる検出	23
4.4 エッジ画像	23
4.5 完全グラフ	23
4.6 全体の流れ図	24
4.7 背景成分の除去	25
4.8 特徴点の 3 次元分布 (1)	26
4.9 特徴点の 3 次元分布 (2)	26
4.10 特徴点の 3 次元分布 (3)	27
4.11 特徴点の 3 次元分布 (4)	27
4.12 特徴点の局所分布	28
4.13 軌跡の決定	29
4.14 動的計画法による決定過程	30
4.15 動的計画法の擬似コード	31
4.16 車両エッジ画像の作成	32
4.17 点間のエッジ成分	33
4.18 ずれの補正	33
4.19 完全グラフの例	35
4.20 GSA	38
4.21 検出結果	40
4.22 (a) 手前に直進 1	41
4.23 (a) 手前に直進 2	42
4.24 (b) 俯瞰画像 1	43

4.25 (b) 俯瞰画像 2	44
4.26 (c) 右側面	45
4.27 (d) 左側面	46
4.28 (e) 夜間	47
4.29 軌跡の直線性の評価による特徴点の棄却	49
4.30 3 台の車両における完全グラフ	49
4.31 実行速度	50
5.1 背景特徴点の棄却.	56
5.2 軌跡の延長	58
5.3 2 点間のエッジ強度	61
5.4 類似度の正規化	62
5.5 グラフ分割による追跡の流れ	64
5.6 V_t^i と V_{t-1}^j の対応付け	65
5.7 追跡結果 (昼 1)	67
5.8 追跡結果 (昼 2)	68
5.9 追跡結果 (夜)	69
5.10 時間推移に伴う車両位置の変化	71
5.11 重みの修正による効果	73
5.12 実行速度	75
6.1 近接する 2 車両の追跡過程.	80
6.2 背景特徴点の棄却.	81
6.3 2 点間のエッジ強度	83
6.4 グラフ分割による追跡の流れ	85
6.5 実験に用いた動画画像	90
6.6 計測結果	91
6.7 エラー要因	91
6.8 近接しつつある 2 車両	93
6.9 離れつつある 2 車両	94
6.10 実行速度.	95
7.1 拘束付きグラフ分割による追跡のフレームワーク	98
7.2 テニスの試合における選手追跡	99
7.3 後方接近車両の水平エッジ成分	100

7.4 監視カメラ画像	101
B.1 二重の輪	110
B.2 重み付きグラフ	110

表目次

3.1	従来手法それぞれの特徴	17
4.1	検出精度	39
5.1	しきい値	70
6.1	しきい値	89

第 1 章

序論

1.1 近年の交通事情と高度交通システム (ITS)

近年、人口の増加や個人の生活水準の向上、車両価格の低下などの理由により、車両の保有台数が個人、法人を問わず年々増加する傾向にある。国土交通省自動車交通局が発表した統計資料によると、個人、法人の保有する車両数は伸び続けており、平成 18 年 3 月末の時点では 7800 万台以上と報告されている。車両数は道路のキャパシティに比べ過剰状態になりつつあるとも言え、渋滞の慢性化や交通事故の増発、違反車両の増加が深刻な社会問題として認知されることとなった。

そこで、ITS(Intelligent Transport Systems:高度交通システム)のもとに民間、公的機関が集まり、これらの問題の解決を目指した研究開発が活発に行われるようになった。ITS が対象とする分野は広く、(1) ナビゲーションシステムの高度化、(2) 自動料金収受システム、(3) 安全運転の支援、(4) 交通管理の最適化、(5) 道路管理の効率化、(6) 公共交通の支援、(7) 商用車の効率化、(8) 歩行者等の支援、(9) 緊急車両の運行支援、といった目的のもとで研究開発が続けられている。

ITS が対象とする問題は学術的にも興味深いものが多い。通信技術、センシング、最適化理論など、異なる分野の研究者達が ITS のもとに集まり、それぞれの専門分野の垣根を越えて活発な研究活動を繰り広げている。この動きは世界的なものであり、多くの国際会議や海外の論文誌が ITS に関連する研究事例を取り扱っている。無論、日本でも多くの学会で ITS に関連する論文が発表されており、国内における ITS への関心が非常に高いことが伺える。ITS における研究活動に焦点を当てる国際会議や論文誌は数多く [1, 2, 3, 4]、その代表的な例としては IEEE Transactions on Vehicular Technology[1] が有名である。2000 年からは新たに IEEE Transactions on Intelligent Transportation Systems[2] が発刊され、ITS に関連する論文発表の場がさらに広がってきた。国際会議と

しては ITS 世界会議 [3] や IEEE Intelligent Vehicle Symposium[4] などが有名である。ITS の研究は産学連携のもとで取り組まれているものが多く、企業のみならず大学による研究活動も活発に展開されている。2004 年に名古屋で行われた ITS 世界会議では、ITS の研究に取り組む日本の大学が集まり大々的な展示発表が行なわれた。ITS 世界会議や IEEE Intelligent Vehicle Symposium などの国際会議は前述の論文誌よりも速報性が高く、また幅広い研究事例を扱っているため、ITS の研究動向を概観する上で非常に参考になる。また、各研究機関に所属する研究者達の情報交換の場としても機能しており、ITS の研究の流れを支える主軸としての役割を果たしている。

1.2 ITS におけるマシンビジョン技術

前節で述べた通り、様々な分野の研究者が多角的な視点から研究活動を展開しているが、中でも次世代の技術として近年注目されているものとして、マシンビジョン技術が挙げられる。マシンビジョンとは、特に実用を意識した画像認識によるセンシング技術の総称である。センシング技術自体の歴史は古く、これまでに超音波や誘電コイル、赤外線検知、ミリ波レーダなど、様々なセンシングデバイスが開発され、交通流量の計測や危険車両の検知など、多くの局面で実用化されてきた。だが、これらのセンサはその検知範囲が局所的であり、得られる情報量が乏しいという欠点がある。一方、マシンビジョンではセンシングデバイスとしてカメラを用いるのが特徴である。カメラは検知範囲が非常に広く情報量に富むため、従来の局所検知型のセンサとは異なる、高度な環境情報獲得を可能にすると期待されている。カメラから得られる情報量は膨大であり、得られた情報の解釈は非常に難しいが、近年では画像認識・コンピュータビジョンにおける理論が成熟してきており、また計算機の処理能力が飛躍的に向上したことから、画像を情報源とした環境センシングの技術が現実可能なものとなってきた。

前述の国際会議や論文誌 [1, 2, 3, 4] でも多くのマシンビジョン技術が取り上げられており、特に IEEE Transactions on Vehicular Technology の 2004 年 11 月号ではマシンビジョン技術の特集 [5, 6, 7, 8, 9, 10, 11] が組まれた。また、コンピュータビジョンの分野で高いレベルを誇る論文誌として知られている IEEE Transactions on Pattern Analysis and Machine Intelligence[12] の 2005 年 5 月号では、道路上の車両検出手法についてまとめた解説論文 [13] が発表された。これらの特集論文や解説論文を筆頭に、多くのマシンビジョン技術が発表されており、この分野の活発さをうかがい知ることができる。

ITS におけるマシンビジョン技術の多くは、道路インフラにおけるマシンビジョンと車載カメラによるマシンビジョンという二種類に分類できる (図 1.1)。前者は道路側に設置したカメラを用いることで、高速道路や駐車場、交差点などにおける車両や人の振る舞い

をとらえようと試みるものである。後者は車の内外に取り付けたカメラを用いることで、後方・前方車両の監視やレーンキーピング，運転手の状態監視などを行うものである。

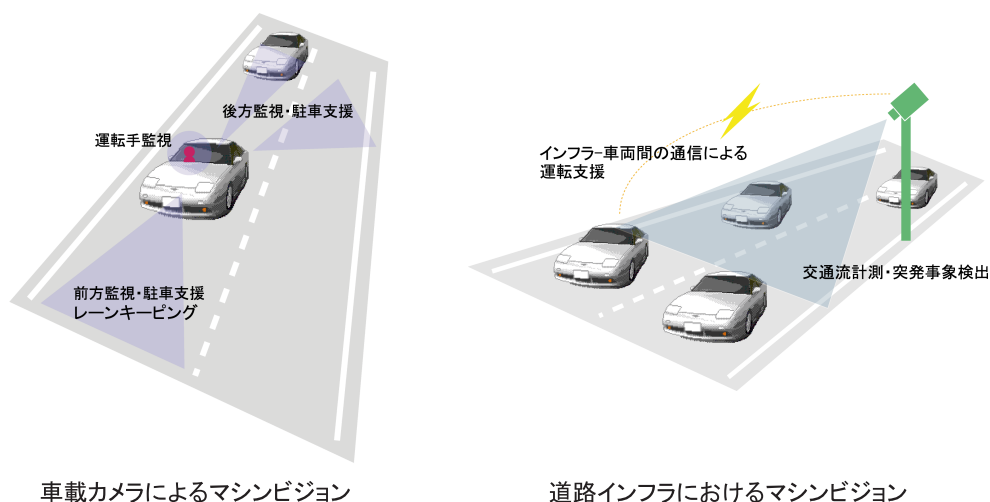


図 1.1 ITS におけるマシンビジョン技術

1.2.1 道路インフラにおけるマシンビジョン

道路側に設置されたカメラから得られた動画像を処理することで，道路交通流に関する様々な情報を獲得しようと試みるのが，道路インフラにおけるマシンビジョン技術である。主なものとしては，交通流計測，突発事象検知，道路側からの運転支援を目的としたシステムなどがある。以下，これらについて概説する。

交通流計測

交通流計測とは，道路上の車両の通過台数や通過速度，個々の車両の振る舞い，道路状況に関する様々なパラメータの解析・収集を行うものである。これらの情報は渋滞の緩和や目的地到達時間予測などに役立つと考えられており，現在では主に超音波センサや誘電コイル式センサなどの機器により行われている。だが，これらのセンサは検知範囲が局所的であり，得られる情報量が乏しいという欠点がある。そのため，近年では画像センサによる交通流計測に対する期待感が高まっている。

本論文の目的はマシンビジョン技術により交通流計測を行う手法を提案することにある。そこで，交通流計測とマシンビジョン技術の関係については2章で詳細に述べることにする。本章では交通流計測以外の応用におけるマシンビジョン技術を概説する。

突発事象

交通事故の多発は、近年における代表的な社会問題のうちの一つである。近年になって、これを自動的に検知することで、人命の救出、および交通流の復旧を迅速に行おうという試みが成されるようになった。しかし、事故が起こったか否かを判別するのは非常に難しい問題であるとされている。局所的に物体の有無を検知するタイプのセンサや、道路交通流の速度を計測するセンサなどでは情報が不足であり、事故が起こったか否かを判別するのは困難である。そもそも、突発事象をどう定義するかが難しく、これが突発事象検出システムの実現を阻んでいる主な要因である。

この突発事象の検出を一種の認識問題としてとらえた研究事例として、HMM による方法 [14] がある。このシステムは画像を入力情報とするのが特徴である。画像上の車両を追跡し、その振る舞いを解析することで突発事象の認識を行う。認識系には2車両の追突と接近、近い位置の通過という3パターンを認識するHMMを用いる。突発事象対策のためのマシンビジョンに関する研究は未だ黎明期にあるが、近年のコンピュータビジョンの理論の成熟とともに、今後の発展が期待される分野である。

道路側からの運転支援

道路と車とを無線による通信で結びつけることで、新たな運転支援の可能性を求めようとする研究事例がある。車両に取り付けたセンサの検知範囲には限界があり、障害物や対向車両の後方にはどうしても死角が生じてしまう。しかしながら、路上の高い位置に取り付けられたカメラならば、近辺の状況を一挙に取得できる。そこで大田らの研究グループ [15, 16] は、道路インフラに設置された監視カメラの映像を、通信を介して車両側へ提供することで運転者の状況把握を視覚的に支援する一連のシステム“NaviView”を提案した。初期の NaviView は、車両の上方における仮想カメラからの鳥瞰映像を生成し提示することによって、運転者の状況把握を補助するものであった [15]。その後、機能拡張が進み、対向車線において死角となる位置を見やすく提示するバーチャル坂道という新しい映像提示方法が提案された [16]。これは、対向車線において、本来であれば運転手から見えるはずのない位置にいる車両を、地上から空へと向かう擬似的な坂道の上に合成して提示することで、運転手の周囲環境把握を補助しようとするものである。このほかにも、死角となる位置をフロントガラス上に重畳表示するバーチャルミラーなど、これまでに無かった新しい運転支援技術が実装されつつある。道路側からの運転支援を実現するためには、道路側と車側の両方の環境が整備される必要があるため、実用化に向けては多くの障壁を越えねばならないが、道路と車とを通信させることで新たな可能性が開けることを示した事例として興味深い。

1.2.2 車載カメラにおけるマシンビジョン

車載カメラによるマシンビジョン技術としては、安全性の確保を目的とするものや、未熟な運転手の運転操作を支援をするものなどがある。現在、自動車メーカー各社は製品の差別化を図るために車載カメラにおけるマシンビジョン技術に力を入れている所が多く、産業的にも非常に活発な分野である。以下、前方・後方障害物監視、駐車支援、および運転手監視のためのマシンビジョン技術について概説する。

前方・後方障害物監視

前方あるいは後方に向けてカメラを取り付け、撮影された映像から障害物を早期に検出し、運転手に警報を与えようとする試みが数多く行われている [17, 18, 19, 20]。これを 1 台のカメラで試みたものとして、水平、垂直エッジのパターンから後方接近車両をとらえる方法 [17] がある。この手法では、まずレーン検出結果と水平エッジ成分から車両の探索領域を絞り込み、当該領域内の垂直エッジを検出し、領域を過分割する。その後、隣接する小領域内におけるテクスチャのクラス間分散を参考に統合していくことで、車両の領域を求めている。校正済みのカメラを用いれば、車両の下端の y 座標から接近車両までの相対距離と相対速度を求めることができるため、これらの情報に基づいて接近車両が安全かどうかの判別を行っている。また、同様に単眼カメラを用いたものとして、車両の対称性に基づく方法 [18] がある。一般に、車両のような人工物は水平方向に対称な明度パターンを持つことが多く、これが前方および後方車両を見つけるための良い手がかりとなる。この手法では、明度分布の対称性の他に、エッジ情報の対称性を同時に取り入れることで、ノイズ成分にロバストな対称軸の検出を行っている。以上の手法は検出対象とする障害物のテクスチャ情報を主な手がかりとした。一方、幾何学的な拘束条件を利用することで、単眼カメラによる障害物の検出を成功させたものがある。例えば、複比と消失線に基づく方法 [19] がこれに属する。複比とは同一直線上の 4 点に関して定義されるものであり、射影変換に関する幾何学的不変量である。この手法では、まず複比の不変性から 3 本の水平線が他車両に属する場合と、路面に属する場合それぞれについて運動の拘束条件を導いている。そして二つの拘束条件のうち、どちらの拘束条件を満たすかを調べることで障害物の検出を行っている。単眼カメラを用いる手法は低コストで実現可能という利点があるが、想定されていない状況では検出精度が低下することがあり、この性能低下は夜間・雨天・降雪などで特に顕著である。この問題をステレオカメラを用いて解決した手法 [20] が提案されている。これは背景が複雑な一般道路環境下での利用を想定したものである。路面とカメラとの姿勢パラメータを動的に推定し、画像のエッジ点の 3 次元位置を三

角測量の原理により求め、高さのある点を障害物のものとして認識している。降雪時やトンネル内など、撮影条件としてはかなり悪い状況でも障害物を検出できることが報告されている。

駐車支援

車庫入れや縦列駐車を苦手とする運転手は少なくない。そこで、近年になって駐車時の運転操作を支援するものが現れてきた [21, 22, 23]。例として、車の前方に取り付けたカメラを用いて路側の車両を検出し、駐車可能な空き領域を運転手に知らせる研究 [21] が挙げられる。これは、道路領域内の水平エッジ成分から路側車両を検出し、それらの車両の間隔を求めることで駐車可能領域を求めるものである。また、周囲環境を3次元復元することで駐車可能領域を見つける手法 [22] が提案されている。これは一種のステレオ法の概念を用いている。自車両の動きが既知である場合、連続した2フレーム間の画像の幾何学的関係が分かるため、三角測量の原理により対象物までの距離を計測することができる。そこで画像から特徴点群を検出および追跡し、それぞれの点の3次元位置から近似的に周囲の3次元モデルを構築することで、駐車可能領域を判別している。さらに、車庫入れ操作の全自動化を目的とした研究も現れてきた [23]。これは、車載 CCD カメラにより撮影した前景画像を車載ディスプレイに出力し、ディスプレイ上で指示された画素点に対応する駐車位置を幾何学的な計算により測量し、駐車位置へ自動走行するものである。この手法では、車両の走行経路を駐車位置付近までの幅寄せと、最終車庫入れ動作の二つに分類し、それぞれの段階における最適な経路を決定し、車両の制御を行っている。

運転手監視

運転手の注意がどこを向いているか、あるいは注意力が散漫になっていないかを判断するためには、顔の状態に着目するのが良いと考えられる。そこで、運転手自身を監視することで、安全性を確保しようと試みる研究が盛んに行われている [24, 25]。運転手の顔検出を目的としたものとして、近赤外光を用いた顔検出法 [24] が提案されている。これは、近赤外光の反射特性が材質ごとに異なることを利用したものである。顔の皮膚や髪の毛、座席の材質はそれぞれ異なる。そこで、異なる2種類の波長を持つ近赤外光を投光し、取得した2種類の画像の差分を取ることで、顔の皮膚領域のみを抽出している。また、運転手の顔の輪郭だけでなく、目・鼻・唇などの各部位を同時に決定できる方法として、Active Appearance Model による方法 [25] が提案されている。これは顔形状の輪郭および顔面のテクスチャ情報を包含する主成分空間を用いた統計的手法であり、1秒間に 200fps という高いフレームレートで顔の姿勢や表情を追跡できると報告されている。

1.3 総括

以上、ITSにおけるマシンビジョン技術を、道路インフラにおける研究と車載カメラにおける研究の2つの視点から概説した。他のセンシングデバイスに比べてカメラから得られる情報は多彩であり、その膨大な情報の中から有益な情報のみを抽出するのは非常に難しい問題である。だが、近年の計算機性能の向上とコンピュータビジョンの理論の成熟に支えられ、多くの研究者達が研鑽してきた結果、本章で紹介したような多くのマシンビジョン技術が生まれてきた。これらの研究事例から分かることは、今やマシンビジョン技術によって新たな可能性が開かれつつあるということである。マシンビジョン技術は一つのセンシング技術として、超音波や誘電コイル、レーザ、ミリ波センサなどに肩を並べており、よく研鑽すれば、他のセンサでは実現不可能なシステムが可能となることが示されてきた。

この研究の流れの中に、本論文の論点は位置する。本論文の目的はマシンビジョン技術による交通流計測システムを提案することにある。そこで、2章ではITSにおける交通流計測をとりまく事情について述べ、そこでいかなる研究活動が展開されているかを議論する。そして3章では、この分野における我々の研究の目的と位置付けについて述べる。

第 2 章

交通流計測とマシンビジョン

2.1 交通流計測の現状とマシンビジョンの利点

交通流計測は ITS(Intelligent Transport Systems) における基礎技術の一つである。その計測値は渋滞の緩和や到達時間予測に利用されており、現在では超音波式やループコイル式の局所型センサと総称される機器により計測が行われている。だが、これらのセンサは精度の面では良好であるものの、広域に渡って測定する場合にはセンサを多数配置しなければならず、設置コストの面で問題がある。また、今後の交通流計測システムは事故車両や違反車両検知システムへと拡張されていくことが期待されており、局所型センサでは本質的に情報量が不足しているためこの実現が困難であると考えられている。

これを受け、マシンビジョン技術を基礎とした広域型センサが広く研究されるようになった。広域型センサは局所型のセンサと比べると安定性や信頼性の面で若干劣るが、単一のセンサで広範囲に渡り多くの情報を取得できるため、局所型センサの欠点を補うものとして有望視されている。広域型センサには、モノクロカメラ、カラーカメラ、赤外線カメラ、ステレオカメラなど様々なバリエーションがあるが、得られる情報が 2 次元の信号、すなわち画像であるという点で共通している。画像に含まれている情報は、局所型センサから得られる 1 次元的な信号とは比較にならないほど膨大である。そこには様々な道路状況を認識するうえで十分な情報が含まれていると考えてよい。特に、画像から個々の車両の軌跡に関する情報が抽出できるということには大きな意義がある。車両が現れてから画面外に消えるまでの一連の軌跡を追うことができれば、通過車両台数や車両速度のみならず、これまでには不可能であった車両の「振る舞い」を解析することが可能になるからである。個々の車両の振る舞い、あるいは複数の車両の相互に干渉しつつ移動する状況を解析することにより、交通事故の認識や違法駐車車両の検出といった、より高次のアプリケーションが実現できるであろう。これは局所型センサには成しえることのできない広

域型センサ固有の利点である。広域型センサには、撮像系は従来のものを用いたままソフトウェア側をアップデートすることで飛躍的な性能向上が可能であるという利点もある。ハードウェアは同じものを使っているが、そこから得られた画像の情報を解釈するソフトウェア側を工夫することで異なるアプリケーションを実現可能である。そういう意味で汎用的かつ低コストであるともいえる。

だが多くの局面において、広域型センサには技術的な困難がつきまとう。画像には多くの情報が含まれているが、その解釈が非常に難しいからである。カメラが撮影対象とする道路状況は様々であり、晴れ、曇り、雨などの天候による変化、時間帯の推移による照明条件の変化、カメラの設置位置と路面との相対位置関係による車両の見え方の相違、車両同士の重なりの問題などが画像処理による追跡を困難なものとする。そのため、これらの点を如何にして解決するかが研究課題となっている。

マシンビジョン技術による交通流計測手法は多く提案されているが、適用対象とする環境、あるいは計測の目的に応じて様々な種類がある。そこで2.2節では、これらを分類して整理し、それぞれの特性について検討する。3章ではこの分野における我々の研究の位置付けについて述べる。

2.2 関連研究

2.2.1 テンプレートマッチングによる方法

車両を追跡する上で最も単純なアプローチとして、テンプレートマッチングによる方法が挙げられる。これは、車両が出現したときの周辺領域をテンプレートとして保存し、テンプレートに類似する明度パターンを時系列に沿って探索し続ける手法である。例えば、内村ら [26] はテンプレート逐次更新することで、通常のテンプレートマッチング処理よりも追跡性能を高めた手法を提案した。このとき、追跡対象の画像平面上における面積の変化量を参考にしてオクルージョンの有無を判定し、オクルージョンが発生していると判定された場合には、車両の移動ベクトルの情報のみを用いて個々の車両の位置を推定するという方法を採用している。テンプレートマッチングによる方法は処理が単純であるため、計算資源が限られている状況では効果的な手法である。しかし、照明条件の変化やオクルージョンへの耐性を求める場合には相応の工夫が必要である。

2.2.2 領域に基づく方法

背景差分法は一般的な対象物体抽出法として広く用いられており、無論、交通流計測にも応用することができる。だが、背景差分法は特に照明条件に弱いという欠点がある。な

ぜなら、ヘッドライトや影による明度の揺らぎの影響により、背景は時々刻々と変容するからである。妥当な背景画像が得られない場合、得られる差分画像はひどくノイズを含む画像になってしまう。そこで、背景の情報を一枚の画像として持つのではなく、それぞれのピクセルにおける明度値の揺れを統計的に解析し、混合ガウスモデルとして近似することで、照明変化に強い対象物体抽出を行う手法 [27] が提案されている。領域に基づく方法は、基本的にはフレーム内処理であるため、車両の時系列における振る舞いを取得するためには、フレーム間で検出結果を対応付けねばならない。そのときにオクルージョンの問題をどのように解決するかが焦点となる。H.Veeraraghavan ら [28] は Kalman フィルタを用い、車両の移動速度と方向を仮定することで、領域に基づく手法でありながらオクルージョンに対処できる方法を提案した。

2.2.3 時空間画像による方法

谷口ら [29] は、車両の大まかな進行方向を直線で近似し、その直線上における時空間画像を解析することで車両の通過台数を得た。時空間画像上において、車両は帯状に現れるため、これを抽出することで車両を検出できる。この手法は車両の移動軌跡が単純な直線である状況でしか適用できないが、条件を満たせば良好な結果が得られる。個々の車両の振る舞いを解析するような用途には向かないが、渋滞の検出や通過台数の計測など、交通流全体の状況を把握するための手法としては十分実用的な手法である。

2.2.4 2次元および3次元の形状モデルによる方法

ほとんどの場合、交通流計測における計測対象は二輪車、普通車、大型車のいずれかである。そこで、事前に追跡対象の形状モデルを作成し、車両の追跡の問題をモデルへのフィッティングの問題として考える手法 [30, 31, 32] が提案されている。形状モデルには様々な種類があり、画像平面上における2次元のモデルを想定するものと、実環境の座標系における3次元モデルを想定するものの2種類がある。形状モデルを想定することで部分的なオクルージョンに堅固になることが、このアプローチの利点である。

J.Malik ら [30] は、車両の輪郭線の移動方向と形状をフレームごとに更新することで車両の追跡を行った。これは車両を包含する多角形を車両モデルとしているもので、モデルベースの手法としては最も基本的な考え方に基づいた手法である。久保山・小沢ら [31] は想定するモデルを工夫し、車両の水平エッジと垂直エッジから成るL型フィルタを提案した。L型フィルタはトンネル内における重交通流に特化したモデルであり、車両が現れてから遠方へと消えてゆくまでの間、常に車両同士の重なり合いが起こるようなシーンでも追跡できるのが特徴である。以上は2次元モデルを用いた手法であるが、これに対して

Song ら [32] は 3 次元モデルを用いた手法を提案している。Song らは車両の 3 次元 CG モデルを用い、あらゆる方向から撮影したときの 2 次元画像群を合成しておき、これをカメラと路面の幾何学的関係に基づいて対象画像に当てはめることで、オクルージョンに非常に強い追跡法を提案した。Song らの実験により、車両の側面をとらえるような低いカメラ位置であっても対象を追跡できることが示されている。

モデルに基づく手法は事前に追跡対象となる車両モデルを準備しなければならないが、追跡対象が想定するモデルによく合致するのであれば、この種の手法は高い追跡性能を誇る。そのため、どれだけ妥当なモデルを事前に準備できるかが焦点となる。

2.2.5 明度・色ヒストグラムを用いる方法

P.Meer ら [33, 34] は、明度・色ヒストグラムを用いた一般的な対象物追跡の枠組みを提案した。これは Mean Shift 法と呼ばれており、車両や人物など剛体・非剛体に関わらず適用できる柔軟な手法である。Mean Shift 法では、対象画像の色ヒストグラムと画像の局所的な色ヒストグラム間の類似度を Bhattacharyya 係数で記述する。この類似度関数が空間的に連続になるという性質を利用し、山登り探索を繰り返すことで対象の追跡を行う。対象は幾何学的変換に不変な色ヒストグラムで表されるため、形状の変化にも頑健な追跡が可能である。また、部分的なオクルージョンが生じてても色ヒストグラムの分布はそれほど変化しないため、オクルージョン問題にも有効な手法として知られている。ただし、対象物体が単色である場合や、照明変化により色ヒストグラムがシフトしやすい状況では追跡の失敗が起こりやすくなるため、これを回避するためには状況に応じて相応の工夫が必要である。

小関ら [35] は Mean Shift 法を車載カメラ画像に適用し、後方接近車両の検出を試みた。安価な車載カメラは映像に歪みや雑音を多く含むため、カラーヒストグラムベースである Mean Shift 法では正確な追跡が困難な状況が発生する。そこで自車の動きを考慮した幾何変換差分を利用し、また協調的に作用する複数の Mean Shift トラッカを用いることでこの問題を回避している。

2.2.6 Condensation を用いる方法

Condensation は M.Isard と A.Blake[36, 37] によって提唱された対象物体追跡のための確率的枠組みであり、しばしばパーティクルフィルタや逐次モンテカルロフィルタなどとも呼ばれる。これは追跡対象を状態量と尤度を持つ多数の仮説群によって非正規分布の確率密度として表現することで、ノイズや環境変動に対してロバストな追跡を実現した手法である。Condensation は確率的状態推定の枠組みの一種であり、しばしば

Kalman Filter と比較して議論される．Kalman Filter では状態を正規分布として扱うが，Condensation では多数の仮説群によって近似的に任意の確率密度を表現するという点において決定的に異なる．Condensation を適用する際のポイントは，いかにして仮説の尤度を定義するかにある．目的が車両追跡であるならば，「ある位置に車両が存在するか否かをどのように評価するか」が性能を決める鍵となる．Condensation はあくまで追跡のための確率的枠組みに過ぎないため，個々の問題に対して有効な尤度関数を定義しなければならない．この尤度関数を車両のヘッドライトの位置を用いて定義し，夜間において車両の追跡を可能とした研究事例として，C.Idler ら [38] による手法などがある．

2.2.7 特徴量のクラスタリングによる方法

車両を直接的に求める代わりに，車両を幾つかの特徴量のまとまりとして考え，車両追跡の問題を時空間における特徴量のクラスタリングの問題として考える手法 [39, 40] が提案されている．このアプローチでは，車両は複数の特徴の集まりから成るという考え方を採用しており，車両の一部が欠けたとしても残りの特徴で補完できるという優れた能力を持つ．そのため，車両同士のオクルージョンに強いという利点がある．

J.Malik ら [39] は，車両とは特徴点の集合であると仮定し，特徴点を Kalman フィルタで追跡した後に，移動軌跡が似ている特徴点同士をまとめあげることで車両を追跡した．特徴点の移動方向に仮定が必要であること，車両のレーンチェンジを考慮していないなど，幾つかの問題点はあるが，照明変動とオクルージョンの両方の問題を克服した初期の研究事例として非常に興味深い．上條ら [40] は画像をブロック分割し，各ブロックを時空間 MRF モデルを用いて車両ごとにクラスタリングすることで車両を追跡した．この手法は，車両の動きが複雑であり，オクルージョンが多発するような状況を克服している．この手法は撮影対象に仮定を必要とせず，交差点のように車両の動きが入り組んでいるような状況においても有効であることが報告されている．上條らは，車両の追跡を時空間内におけるクラスタリング問題としてとらえ，確率的な記述で定式化した．解法として確率的緩和法を導入し，最適なクラスタを決定することでオクルージョンに強い追跡を達成している．

2.2.8 特殊なカメラによる方法

撮影するデバイス自体を工夫することで，照明条件に関する問題の解決を試みたものとして，ステレオカメラによる方法 [41] や，赤外線カメラによる方法 [42] などがある．ステレオカメラによる方法は，異なる 2 地点から同一対象を撮影することで対象までの距離情報を正確に得ることが最大の利点である．距離情報が得られれば，車両の見えに関わ

らず対象を検出できるようになるため、照明やカメラの設置位置に対する条件の少ない、利便性の高いシステムを構築することが可能である。赤外線カメラは主に夜間を対象としたい場合に威力を発揮する。

デバイス側を工夫することで、単眼モノクロカメラを用いた手法よりも優れた結果を得られる可能性は高まるが、そのかわりに設置コストが高くなるという問題が生じる。観測したい地点がそう多くないが、そのかわりに精度良く交通流量を求めたいときにはこれらのアプローチは良い選択肢にはなり得るが、既に設置されてある監視カメラの映像を流用し、広大な範囲において交通流計測を行いたいような場合には適さない。つまり、状況に応じて使い分ける必要がある。

第 3 章

本論文の目的 および関連研究に対する位置付け

3.1 本論文の目的

本論文では、「道路上の監視カメラを用いた、低コストで高性能な交通流計測システム」を開発することを目的とする。そこで、以下の 5 つの性質を満たすマシンビジョン技術による交通流計測手法を提案する。

1. 1 台のモノクロカメラで計測が可能

赤外線カメラやステレオカメラなどの特殊な撮影デバイスを用いれば、環境変化に強い計測ができるようになると期待されるが、一般にこれらのカメラはコストが高く、広範囲に渡る普及を目指す場合には向かない。広範囲に渡って交通流計測を行うことが意図されている場合、既に設置されている監視カメラを流用することができれば、コストの面で最も都合が良い。そこで、一般的に普及しているモノクロカメラ 1 台のみを用いる。

2. カメラの設置位置に関する制約が緩い

マシンビジョン技術による交通流計測手法の多くは、適切な位置にカメラを設置しなければ動作しない。なぜなら、カメラの設置位置が異なると車両の見えも変化するからである。例えば、高い位置から見下ろすように設置した場合と、車両の後方をとらえるように設置した場合とでは、車両の見えは著しく異なる。だが、既存の監視カメラを用いて交通流計測を行おうとするのであれば、様々なカメラ位置を扱えることが望ましい。

3. 照明条件の変化に強い

時間帯・天候の変化に伴い、対象とする画像は時々刻々と変容する。しかし、統計

量の正確さを期するためには、24 時間続けて計測を行う必要がある。それゆえ、道路を取り巻く環境の変化に起因する明度変化に依存しない計測手法であることが望ましい。

4. 車両の移動軌跡を取得可能

超音波や誘電コイルセンサなどによる計測手法は、車両の台数や速度を得ることはできるものの、それらの個々の振る舞いを得ることはできない。しかしながら、車両が画面内に現れてから消えてゆくまでの一連の移動軌跡を得ることができれば、その情報を解析することで、違法駐車車両や突発事故の検出などの応用に結びつくものと考えられる。提案する手法は車両の移動軌跡を取得し、それらの記録を可能とするものである。

5. オクルージョンに強い

カメラの設置位置が低いとき、あるいは道路が渋滞したときなどには、車両同士が画像上で重なり合う状態、すなわちオクルージョンが頻繁に起こる。オクルージョンが起こると、本来複数台の車両として認識すべきものを 1 台として認識してしまい、計測結果に深刻な誤差が生じる。そこで、オクルージョンの問題を解決できる手法を提案する。

3.2 従来研究における位置付け

本節では、本論文と従来研究との関係について議論する。表 3.1 は 2.2 節で述べた 8 種の従来手法の特徴を示したものである。それぞれの手法が、本論文が目指す 5 つの目的をどの程度満たしているのかを 3 段階で評価した。それぞれの手法には様々なバリエーションがあるため一概には言えないが、おおむね表 3.1 で示した傾向が見受けられる。これを元に、8 種の方針のうちで我々の目的に最も適合するのはどれかを考える。

「特殊なカメラによる方法」は、計測精度という点においては非常に優れているが、設置済みの監視カメラを流用するという我々の意図から外れている。また、「時空間画像による方法」は、実行速度の面で非常に優れた手法ではあるが、車の移動方向が単純な直線、あるいは曲線で近似できなければならない。それゆえ扱える交通シーンが限定的であり、やはり我々の目的には向かない。「テンプレートマッチングによる方法」「領域に基づく方法」は、工夫次第で 5 つの目的を達し得るが、オクルージョン耐性という点で若干劣る。「2 次元および 3 次元の形状モデルによる方法」は、十分に研鑽すれば工夫次第で 5 つの目的を達し得るが、そのためには、おそらく信頼のおける車両モデルを充分多く用意することが不可欠である。だが、追跡対象とする車両の形状は様々であるため、車両モデルをどの程度用意すればよいかの指針は不明である。また、想定していない形状の車両が現れ

表 3.1 従来手法それぞれの特徴

手法	目的 1	目的 2	目的 3	目的 4	目的 5
テンプレートマッチングによる方法	○	△	△	○	△
領域に基づく方法	○	△	△	○	△
時空間画像による方法	○	×	△	×	×
2 次元および 3 次元の形状モデルによる方法	○	△	△	○	○
明度・色ヒストグラムを用いる方法	△	○	×	○	○
Condensation を用いる方法	○	△	△	○	○
特徴量のクラスタリングによる方法	○	△	△	○	○
特殊なカメラによる方法	×	△	○	○	△

目的 1:1 台のモノクロカメラで計測が可能

○満たす・向いている

目的 2:カメラの設置位置に関する制約が緩い

△どちらともいえない・工夫次第

目的 3:照明条件の変化に強い

×満たさない・不向きである

目的 4:車両の移動軌跡を取得可能

目的 5:オクルージョンに強い

た場合、それらを取りこぼす可能性があるので好ましくない。Mean Shift 法に代表される「明度・色ヒストグラムを用いる方法」は、対象の形状・撮影環境に特に仮定を置かないため非常に有効な手法ではあるが、照明条件の変化に弱いという欠点がある。また、モノクロカメラを用いる場合、カラーカメラと比較して明度ヒストグラムが特徴的でなくなるため、性能が低下すると思われる。そのため、既存の監視カメラを流用するという目的のもとでは残念ながら不適である。「Condensation を用いる方法」は適切な尤度が定義できるのであれば非常によい解決策になり得る。しかしながら、そもそも Condensation とは追跡対象の状態を示す変数の時間推移を確率的に推定するというものであり、複数対象の追跡を行うためには相応の工夫が必要である。また、1 つの対象物体を追跡するのに通常で数十～数百の仮説の尤度を計算しなければならないため、処理速度の点で難がある。

そこで本論文で提案する手法では、「特徴量のクラスタリングによる方法」を基礎とした。この手法は、車両を角点やエッジ、ブロックなど、複数の構成要素から成る集合体であるとする。そのため、オクルージョンによって車両の一部が欠けたとしても、残りの部分で補完できるという優れた特徴がある。特徴量をクラスタリングするというきわめてシンプルな考え方ではあるが、この考え方は Mean Shift や Condensation とは異なり、そもそも複数の物体を追跡することを念頭に置いた考え方である。Mean Shift や Condensation は、前フレームにおける追跡対象の位置が、次のフレームにおいてどのように遷移するのかを推定するという目的のもとで組み立てられた理論であり、単一、あるいは少数の物体の追跡には強力であるが、多数の物体が相互に干渉しあいながら複雑に動くような状況には不向きである。一方、「特徴量のクラスタリングによる方法」では、同一

の対象から抽出された特徴量をまとめ、その一方で異なる物体由来の特徴量同士を分離するという方針を採る。つまり、はじめから理論的枠組みとして多数の物体を同時に追跡することを考慮に入れている。また、クラスタリングの対象とする特徴量として照明条件、撮影位置に依存しないものを選択すれば、カメラの設置位置や照明変化の問題も解決可能である。そのため、撮影環境が様々であり、かつ多数の車両が画像中に現れるような状況を取り扱う場合に非常に適している。

3.3 本論文の構成

本論文では、以下の5つの目的を満たす交通流計測手法を提案する。

1. 1台のモノクロカメラで計測が可能
2. カメラの設置位置に関する制約が緩い
3. 照明条件の変化に強い
4. 車両の移動軌跡を取得可能
5. オクルージョンに強い

4章「特徴点群抽出とグラフ分割を用いた車両認識」では、上述の目的1,2,3を満たす車両の認識手法について述べる。5章「動画像を入力情報とする車両計数」では、これを時間方向に拡張し、目的1,2,3に加えて目的4を満たす車両追跡手法について述べる。6章「重交通流における車両計数」では、5章で述べた手法をさらに拡張し、目的1~5全てを満たす交通流計測手法を提案する。7章では、車両以外の追跡対象における本手法の適用可能性について議論する。最後に8章で、本論文の結論について述べる。

第 4 章

特徴点群抽出とグラフ分割を用いた 車両認識

4.1 概要

本章では、マシンビジョン技術による交通流計測のための車両認識手法を提案する。本章で述べる車両認識手法は、3.1 節で述べた 5 つの性質のうち、次の 1～3 の性質 (下記太字部分) を有する。

1. **1 台のモノクロカメラで計測が可能**
2. **カメラの設置位置に関する制約が緩い**
3. **照明条件の変化に強い**
4. 車両の移動軌跡を取得可能
5. オクルージョンに強い

我々が目指す交通流計測では、広い範囲の道路交通流の情報収集を目的としているため、低コストで広範囲に設置できることが望ましい。そのためには、すでに道路上に設置されているカメラを流用することが最もコスト面で有益である。道路上の監視カメラの多くは単眼のモノクロカメラである。そこで本章で提案する手法は、単眼のモノクロカメラから得られた画像を入力情報として用いる。この監視カメラが撮影対象とする道路状況は様々である。晴れ、曇り、雨などの天候による変化、時間帯の推移による照明条件の変化、カメラの設置位置と路面との相対位置関係による車両の見え方の相違などが画像処理による追跡を困難なものとする。そのため、これらの点を如何にして解決するかが研究課題となる。そこで、本章では上記 1～3 を満たす車両認識手法に焦点を当てる。

本論文で提案する手法は、3.2 節で述べた通り、「特徴量のクラスタリングによる方法」による考え方を基礎にしている。本手法は、車両の角点が昼夜問わず安定に検出できることに着目する。まず、特徴量画像を生成するフィルタとして微分情報に基づくフィルタ [43] を用い、得られた特徴点の軌跡を動的計画法により求める。このとき、車両を時空間内における特徴点の軌跡の集合としてみなし、これらの軌跡を同一の車両ごとにまとめることで車両を認識する。本手法では、このクラスタリングの問題を、特徴点の軌跡を頂点、軌跡間のエッジ強度を辺の重みとした完全グラフの分割問題とみなすことで解く。この原理によると、昼夜問わず同一のアルゴリズムで、多様な撮影位置、照明条件において得た画像に対して安定に車両を検出することができる。

特徴点の軌跡群をクラスタリングするという考え方を導入した初期の研究事例として、J.Malik ら [39] による手法がある。J.Malik らは特徴点を Kalman フィルタで追跡した後に移動軌跡が似ている特徴点同士をまとめあげることで車両を追跡した。この手法は照明変動に対してロバストであり、また限定的ではあるが、ある特定の状況下におけるオクルージョンに対処できるという利点がある。照明変動とオクルージョンという2点の問題を克服した初期の研究事例として非常に興味深い。しかし、当時の計算機環境は現在のものに比べ貧弱であったことから、クラスタリングには極めて単純なアルゴリズムが用いられていたという点に問題があった。クラスタリングのためには精細な特徴点の軌跡が必要であり、そのため Kalman フィルタを用いていたが、その代償として特徴点の移動方向が一定方向であるという仮定を用いねばならなかった。それゆえ、車両のレーンチェンジなどによる複雑な動きに対応できなかった。だが、J.Malik らの論文報告から 10 年が経ち、現在ではより高度なクラスタリング手法が比較的高速に適用できるようになった。本論文では、クラスタリング手法の一種であるグラフ分割に焦点を当て、Malik らが先駆的に導入した特徴点の軌跡群のクラスタリングという考え方をさらに発展させる。

4.2 適用範囲

本手法は図 4.1(a)–(f) に例示したように、夜間を含む様々なシーンを対象とする。天候は晴れ、曇り、小雨を対象とし、降雪や豪雨は対象外とする。複数のレーンを同時に含む画像でも良く、車両の移動方向は特に仮定しない。奥行き方向から手前に向かって走行するように、画像上での見かけの大きさが変化するような動画像も対象とする。車両の幅と高さは、少なくとも 30pixel 程度の解像度を確保できていることが望ましいものとする。これは後述する特徴点抽出処理で十分な数の特徴点を検出できるようにするための配慮である。風により背景の木が揺れるなど、移動物体以外の動きの成分が現れる場合、その領域は事前に監視領域から外しておく。この監視領域は手動で与える。



図 4.1 入力画像

4.3 車両認識の原理

本手法は様々な撮影角度や時間帯において撮影された映像を対象としている (図 4.1). 移動物体検出法の代表的な例としてはフレーム間差分法と背景差分法が挙げられるが、夜間は車両のヘッドライトや街灯の影響により明度値が不安定になるためにこれらの手法では検出が困難であると知られている. ここでは多様なシーン、時間帯を対象とした車両検出を目的としているため、撮影位置や照明条件の変化に不変な、撮影対象に依存しない特徴量を用いなければならない.

そこで本手法では、車両から抽出する特徴量として微分演算に基づく特徴点 [43] を採用した. 一般的に車両のような人工物は形状に直線的な部分が多く、その部分において高い微分値を示すため、特徴点を検出し易いという性質がある. また、微分値は照明変動に起因する全体的な明度変化に関係の無い量であるため、照明条件が悪いような環境下においても安定な検出が可能である. 図 4.2 は夜間における車両に対して特徴点検出処理を行ったものであり、このように街灯やヘッドライトの影響が強いシーンであっても特徴点を安定に検出できていることが分かる.

検出した特徴点は画像座標の x 軸, y 軸に時間 t 軸を加えた, xyt 軸から構成される 3 次元空間 (時空間) 内に分布するとみなせる (図 4.3). このとき、同一の車両からもたらされた特徴点の軌跡は空間内で柱状に集群する. この性質は対象を撮影した角度によらず成立するものであるから、これらの軌跡を車両ごとにクラスタリングすれば、撮影位置に

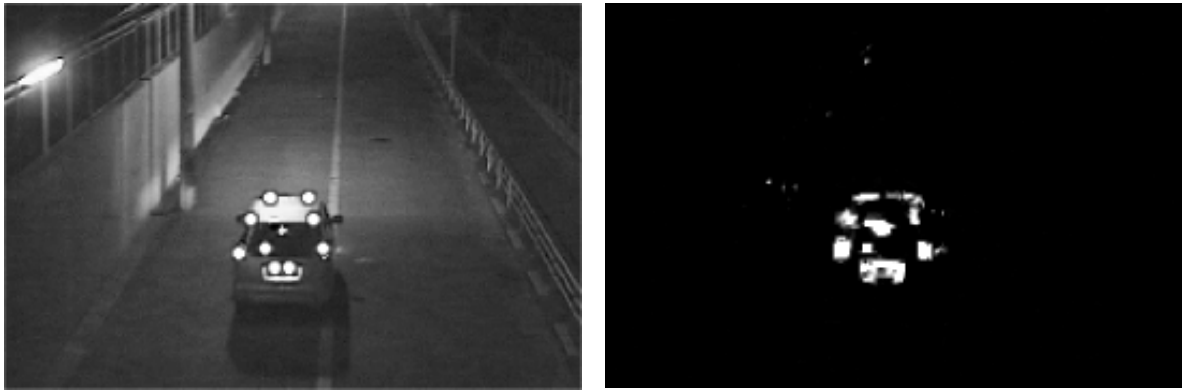


図 4.2 夜間における特徴点抽出

よらない検出を実現できる。

車両の追跡の問題を、画像平面上で疎に分布する特徴点のクラスタリング問題に帰着することで得られる利点が2つある。一つ目は、車両の一部であることが確からしい画素のみに対して処理を行うことで、ノイズの影響を回避できることである。画像は多くの画素からなるが、その全てを用いて処理を行うことは冗長である。車両追跡において有意な画素と、それほど意味を成さない画素とが同列に扱われることは精度の低下の原因となりえる。だが我々は、車両は人工物であり、特徴点が多く抽出できるという先験的な知見を持っている。そこで、特徴点こそが車両追跡において有意な画素であるとみなし、その他の冗長な画素を削減した上で処理を行うことで精度の向上を図ることができる。二つ目の利点としては、処理対象とする画素が減ることで、処理速度の向上が見込めることが挙げられる。一般にクラスタリングは計算コストのかかる処理であり、その処理速度は分類する要素の数に依存する。全ての画素をクラスタリングの対象とすると、リアルタイムな処理は困難である。だが事前に車両の特徴点を抽出し、クラスタリングの対象となる要素を減らしておくことで、大幅な処理速度の向上が望める。J. Shi ら [44] は画像上の画素全てをクラスタリングの対象とする手法を考案したが、その結果は精度および速度の面で不十分であった。したがって、疎に分布する特徴点を利用するという方針は理に適っているといえる。

特徴点の軌跡群をクラスタリングするためには、その手掛かりとなる類似度を定義する必要がある。ここでいう類似度とは、2つの軌跡の組み合わせごとに定義されるスカラー量であり、それら2つの軌跡が同一の車両から検出されたものであるか否かを評価する尺度である。類似度が高ければ、2つの軌跡は同一の車両のものである可能性が高いものとする。同一の車両から検出された特徴点間には強いエッジ成分が存在する(図4.4)という性質があるため、本章で述べる手法ではエッジ成分の強さに基づいて類似度を設定した。

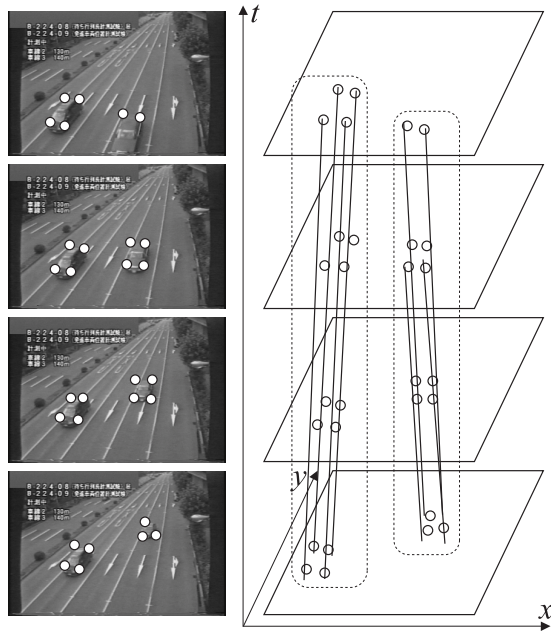


図 4.3 クラスタリングによる検出

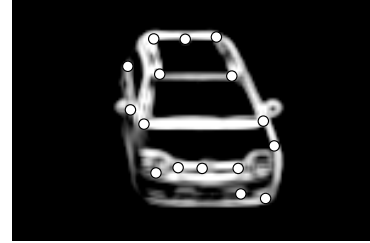


図 4.4 エッジ画像

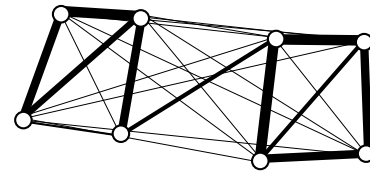


図 4.5 完全グラフ

以下、これを軌跡間の類似度、あるいは単に類似度と呼ぶことにする。

複数の要素間に重みを持った連結関係がある場合、それらの構造をグラフ (graph) としてみなすことができる。グラフとは、頂点集合と2つの頂点を端点とする辺の集合からなる構造である。図 4.3 における特徴点の軌跡群をグラフの頂点集合とみなし、2つ軌跡の全ての組み合わせを辺集合とみなし、それぞれの辺に対応する類似度を重みとして与えた重み付き完全グラフが図 4.5 である。軌跡間の類似度が強い部分は図中で太く示した。このように、同一の車両から検出された軌跡同士は強い連結関係を示すが、異なる車両から検出された軌跡間の連結は非常に弱くなる。つまり、この完全グラフを辺重みの弱い部分で切ること、軌跡を車両毎に分類することができる。この分割アルゴリズムには様々な種類があり、最適な分割を得るための手法が検討されている [45, 46]。

グラフ分割に帰着する利点は、その分割アルゴリズムのロバストさにある。従来の差分に基づく移動物体検出法と同様に、軌跡同士の類似度は車両間の近接やノイズ成分に影響を受けることがある。これにより、異なる車両間における軌跡同士が大きな類似度を持ってしまうことがある。図 4.5 に示したグラフでは、2 車両間で類似度が大きくなる軌跡が 1 組だけ存在している。このようなエラーを全て取り除けるような類似度を定義するのはきわめて難しい。しかし、グラフ分割アルゴリズムはグラフの大局的な性質を示す評価値を最大化（あるいは最小化）するような分割を能率よく探索するというものであるため、異なる車両間において高い類似度を持った軌跡が生じてしまったとしても、それが大局的

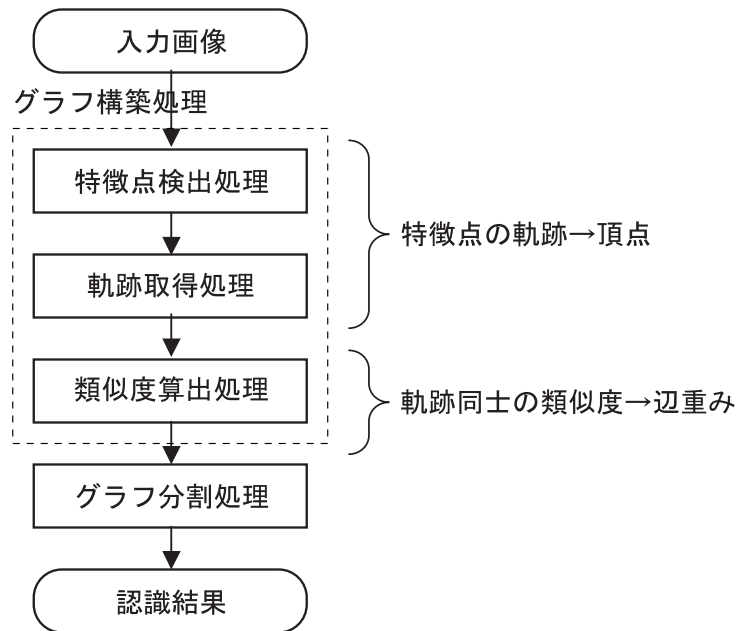


図 4.6 全体の流れ図

にみたときに些細なエラーなのであれば、これは他の連結に比べて非常に弱いので、車両は問題なく分離される。つまり、前段の類似度算出処理における失敗を、後段の分割処理によって修正できるという利点を持つ。これはシステムのロバストさを向上させるという意味で非常に有効である。

以上説明したように、本手法はグラフの構築処理と分割処理から成る。グラフの構築処理では特徴点の軌跡、および類似度を求め、それらを頂点集合と辺の重みとみなすことで重み付き完全グラフを構築する。分割処理では、前段で得た軌跡集合をグラフ分割アルゴリズムを用いてクラスタリングし、車両の位置を決定する。図 4.6 に本手法の全体の流れ図を示した。次節以降、これらの各段階の処理を詳細に述べる。

4.4 グラフ構築

4.4.1 特徴点検出処理

本手法では特徴点検出法として微分情報に基づくフィルタ [43] を採用した。これは、コンピュータビジョンの分野で標準的に用いられている特徴点抽出法であり、複数の方向に強い微分成分を持つ点を特徴点として選択するものである。車両は人工物であり、窓の隅やバンパー付近、ナンバープレート付近など、多くの場所において強い微分値を持つ。そのため、車両からは多くの特徴点を得ることができる。

入力画像 $I(\mathbf{x})$ にフィルタを適用すると、各座標位置における特徴点らしさを示す画像 $f(\mathbf{x})$ を得ることができる。通常の処理では、 $f(\mathbf{x})$ において局所最大であり、かつ閾値 λ_{th} を超える値を持つ座標を特徴点として採用すればよい。だが本手法で必要なのは移動物体の特徴点のみであり、このままだと背景成分の特徴点まで検出されてしまうため、これらを除く必要がある。そこで、事前に用意しておいた背景画像 $I'(\mathbf{x})$ にフィルタを適用して算出した $f'(\mathbf{x})$ を $f(\mathbf{x})$ から引くことで背景成分を除去する (図 4.7)。

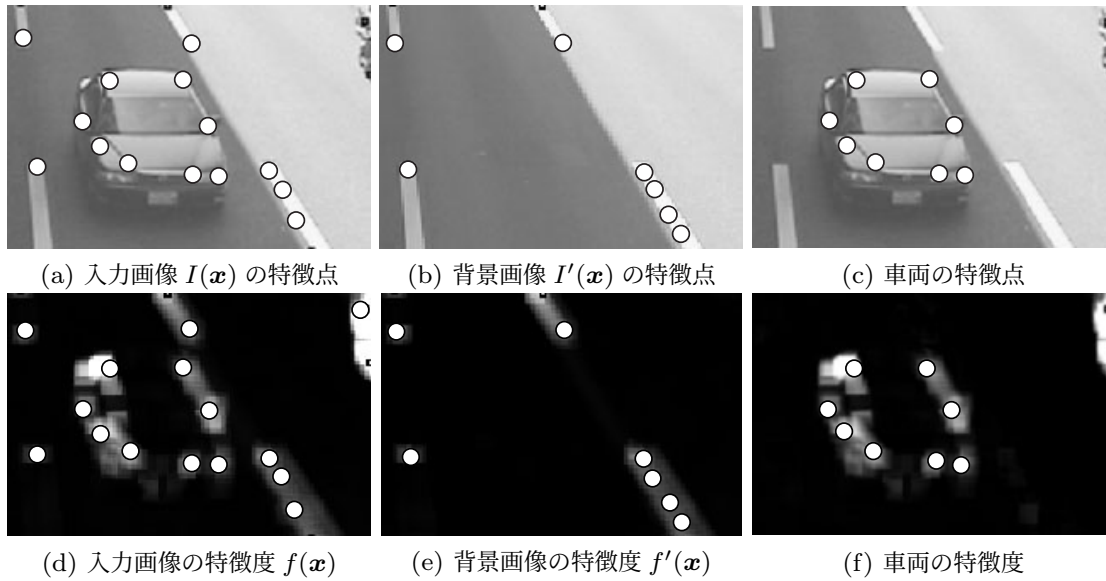


図 4.7 背景成分の除去

閾値 λ_{th} は特徴点検出の感度を示す値であり、これを小さく設定すればより多くの特徴点を検出できるが、ノイズによる不要な特徴点も混入し易くなってしまふ。従来用いられてきた画像間の差分による方法も同様な問題を抱えており、物体を取りこぼすことの無いように閾値を設定すると、今度は逆に不要な物体の誤検出を招きやすくなる。しかし、本手法では後半の軌跡取得処理やグラフ分割処理で不要な特徴点を棄却する機構を備えているため、 λ_{th} は小さめに設定しておけば良く、厳密に調整する必要は無い。具体的には $\lambda_{th} = 0.03$ 程度に設定すればよい。

以降、時間 t における入力画像から検出された特徴点の数を n_t 、 i 番目の特徴点を $\mathbf{p}_i^t$ で表し、時間 t において検出された特徴点の集合を $P_t = \{\mathbf{p}_1^t, \mathbf{p}_2^t, \dots, \mathbf{p}_{n_t}^t\}$ と表記する。フレームごとに車両の特徴点群 P_t を抽出し、これを xyt 軸から成る時空間内にプロットしたものが図 4.8, 図 4.9, 図 4.10, 図 4.11 である。同一車両から検出された点群が、時空間内で柱状に現れていることがわかる。

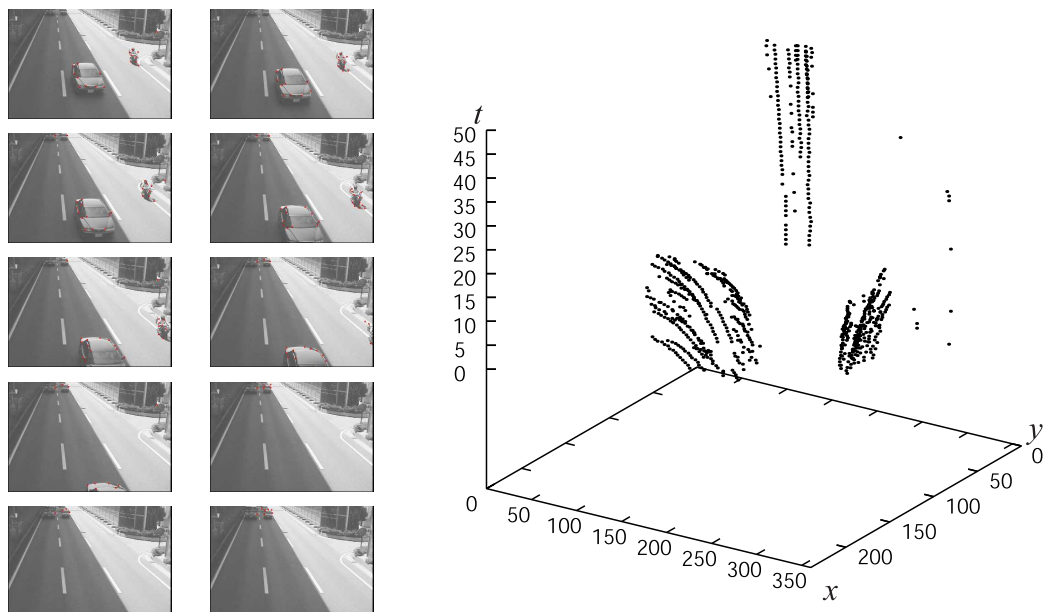


図 4.8 特徴点の 3 次元分布 (1)

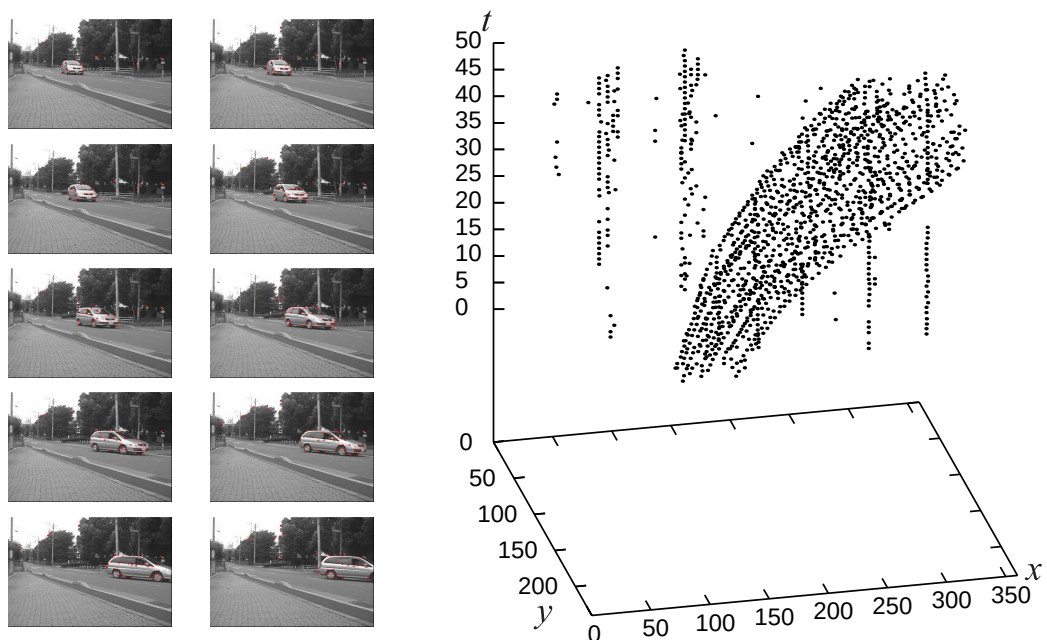


図 4.9 特徴点の 3 次元分布 (2)

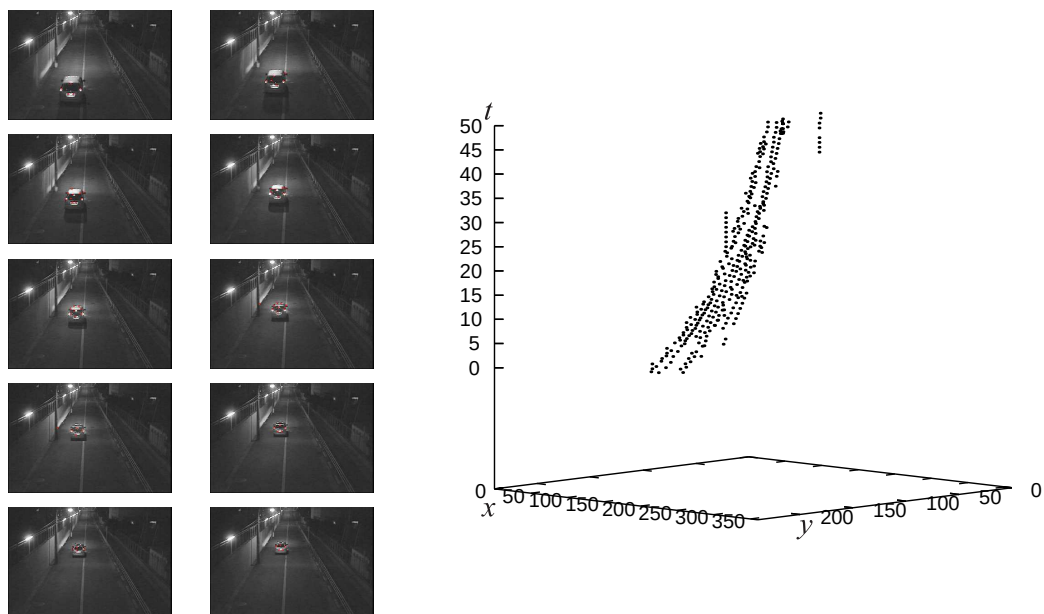


図 4.10 特徴点の 3 次元分布 (3)

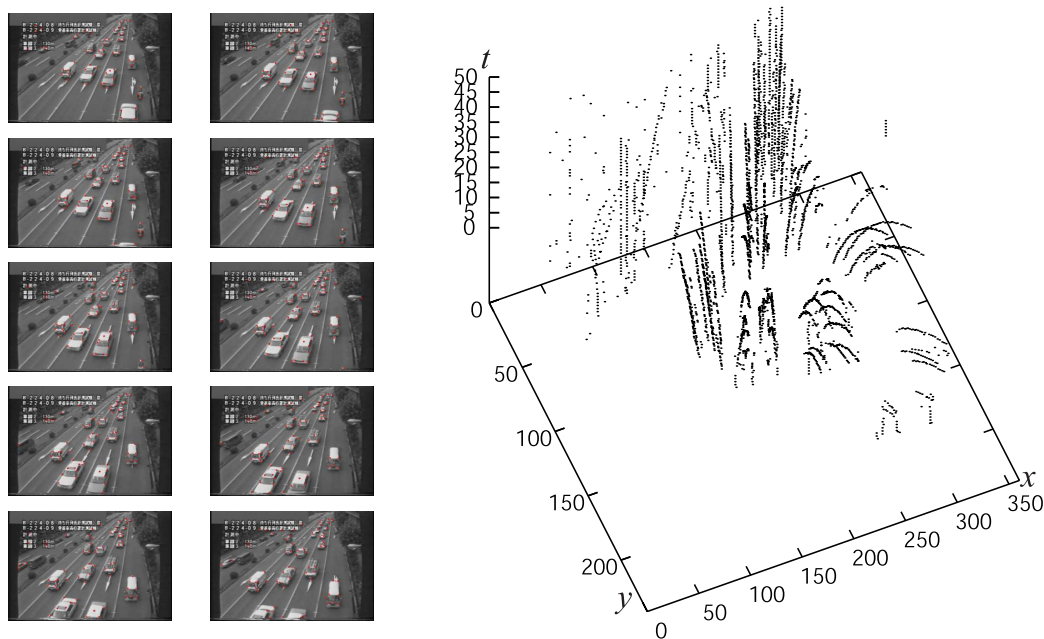


図 4.11 特徴点の 3 次元分布 (4)

4.4.2 軌跡取得処理

軌跡の定式化

次の処理では，フレーム単位に独立に検出された特徴点の軌跡を求める．本手法は特徴点の軌跡群を車両として考える方法を採用しており，ここで求めた軌跡は後段の処理において構成する完全グラフの頂点となる．軌跡を用いる利点として，特徴点を点のままクラスタリングせずに時間方向に対応付けられた軌跡をクラスタリングすることで，結果を時間方向に安定化できることが挙げられる．また，ノイズに起因する特徴点はその検出が時間方向に不安定であり，軌跡が安定に求められたか否かを評価することでノイズ成分を棄却することが可能である．

図 4.12 は，10 フレーム (15fps の動画像) 内での特徴点の分布の様子を示したものである．このように，局所時間内で特徴点は直線的な軌跡上に分布する．つまり軌跡を求めるには，時空間内の直線を検出すればよい．この仮定は局所時間内においてのみ成立するものであり，この期間を $N(\text{frame})$ とする．以後，この $N\text{frame}$ のまとまりを $Unit_{t-N+1}^t$ と記述する．

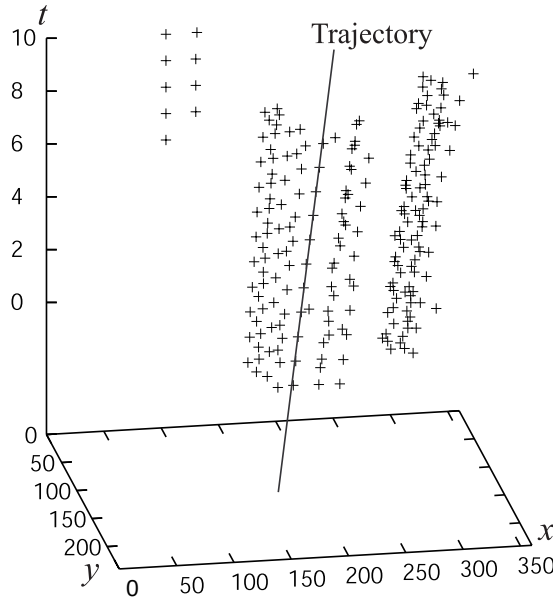


図 4.12 特徴点の局所分布

そこで，時間 $t - N + 1$ から t までの間に存在する特徴点群の軌跡を求めることを考える (図 4.13)．まず前提として，求める軌跡には，最も最近のフレーム t における i 番目の特徴点 p_i^t が含まれているものとする．この軌跡は式 (4.1) のように，集合として定義さ

れる.

$$L(\mathbf{p}_i^t) = L_i = \{\hat{\mathbf{p}}_i^{t-1}, \hat{\mathbf{p}}_i^{t-2}, \dots, \hat{\mathbf{p}}_i^{t-N+1}\} \quad (4.1)$$

$$\hat{\mathbf{p}}_i^{t-\tau} \in P_{t-\tau} \quad (\tau = 1, \dots, N-1) \quad (4.2)$$

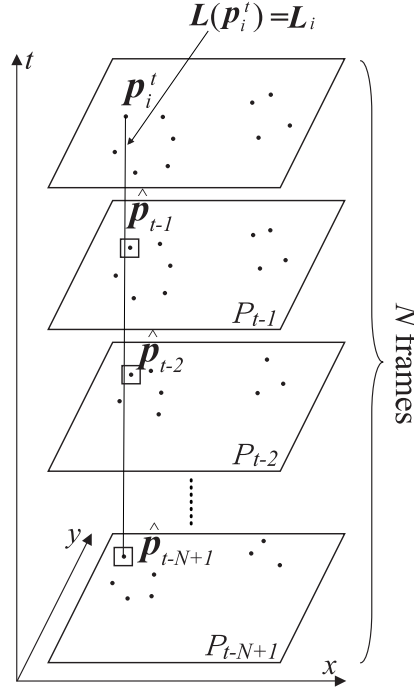


図 4.13 軌跡の決定

$L(\mathbf{p}_i^t)$ は特徴点 $\mathbf{p}_i^t$ が含まれる軌跡であり, $\hat{\mathbf{p}}_i^{t-1}, \hat{\mathbf{p}}_i^{t-2}, \dots, \hat{\mathbf{p}}_i^{t-N+1}$ は, 時間 t よりも過去のフレーム $t-1, \dots, t-N+1$ からそれぞれ一つずつ選んだ点である. 以下, 記述の簡略化のため, $L(\mathbf{p}_i^t)$ を L_i と略記する. 時間 t の特徴点数は n_t 個であるから, 検出される軌跡の総数も $L_1, \dots, L_{n_t}$ で計 n_t 本となる. ここで, $\mathbf{p}_i^t$ が属する軌跡を決定するためには, $\mathbf{p}_i^t$ 及び $\hat{\mathbf{p}}_i^{t-1}, \hat{\mathbf{p}}_i^{t-2}, \dots, \hat{\mathbf{p}}_i^{t-N+1}$ が直線上に乗るような最適な組み合わせ L_i を決定すればよいことになる.

だが, 式 (4.2) から選択される組み合わせは無数にあり, これら全ての組み合わせに対し, 直線上にあるか否かを逐一検証していくのは困難である. 仮に 1 フレームあたりに検出された平均特徴点数を m とおくと, その組み合わせは m^{N-1} となる. 実際には $m = 100, N = 20$ 程度であり, 計算量が膨大になるため解を得ることができない. このような各段階 (フレーム) で一つずつ候補を選んでいく多段決定過程には, 動的計画法による計算量の削減が非常に有効である. これにより探索の計算コストを $o(m^{N-1})$ から $o(m^2(N-1))$ に削減できる.

動的計画法による軌跡の取得

L_i の要素は直線状に分布するという仮定に基づき、時間 τ において p_j^τ が選ばれたときの評価量 E_j^τ を次のように定義する．

$$E_j^\tau = \min_{1 \leq k \leq n_{\tau+1}} (\|\delta_j^\tau - d_k^{\tau+1}\|^2 + E_k^{\tau+1}) \quad (4.3)$$

動的計画法では、各段の決定候補 P_τ の中で仮に p_j^τ が選ばれたとした場合、最も評価量を向上させる前段の候補 $p_k^{\tau+1}$ を決定する．この選択関係を図 4.14 に線で示した．このように、現段階における決定は前段の決定に依存するため、評価量は式 (4.3) のような再帰的定義となる．

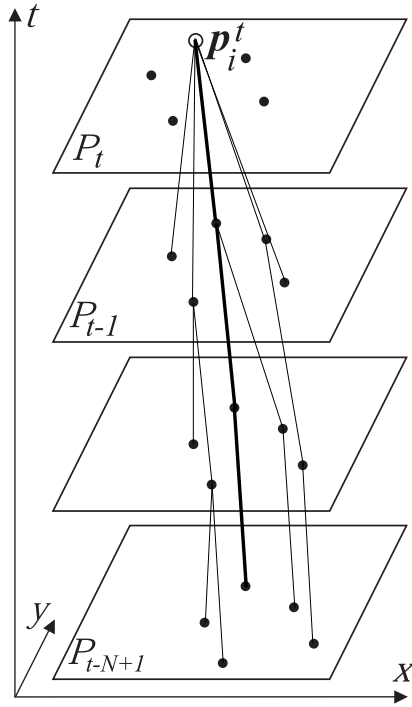


図 4.14 動的計画法による決定過程

$\|\delta_j^\tau - d_k^{\tau+1}\|^2$ は現フレームにおける決定 p_j^τ に関する評価量の項であり、直線分布に近いほど小さな値を示す． δ_j^τ は p_j^τ から p_i^t への移動ベクトルであり、式 (4.4) で定義される．

$$\delta_j^\tau = \frac{p_i^t - p_j^\tau}{t - \tau} \quad (4.4)$$

また、 $d_k^{\tau+1}$ はこれまでの決定における平均移動量であり、式 (4.3) の $\min$ 演算子によ

り決定された k を k' とおくとき、次のように更新される。

$$\mathbf{d}_j^\tau = \frac{(t - \tau - 1) \cdot \mathbf{d}_{k'}^{\tau+1} + \boldsymbol{\delta}_j^\tau}{t - \tau} \quad (4.5)$$

つまり、現フレームで選んだ $\mathbf{p}_j^\tau$ の移動ベクトル $\boldsymbol{\delta}_j^\tau$ が、前段までの決定における平均移動量 $\mathbf{d}_k^{\tau+1}$ に近いほど、良い評価量が付加される。

最終段まで式 (4.3) による評価量の付加を繰り返し、最終段において最小の評価量 E_i が付加された選択からこの線を遡って辿れば、最適な組み合わせを得ることができる。 E_i は直線 L_i の直線近似精度であり、式 (4.6) で算出できる。

$$E_i = \min_{1 \leq k \leq n_{t-N+1}} (E_k^{t-N+1}) \quad (4.6)$$

図 4.15 は動的計画法による軌跡取得の擬似コードである。

なお、式 (4.3) において選択候補となる特徴点のどれもが評価量を著しく大きく、すなわち悪くするような場合、そのフレームでの特徴点は検出漏れが起きているものとして選択せず、評価量に一定値のペナルティを加える。これにより、特徴点の欠けによる軌跡の検出漏れを防いでいる。

```

(Init)

for( $j = 1; j \leq n_{t-1}; j++$ ) {
     $E_j^{t-1} = 0$ ;
     $\mathbf{d}_j^{t-1} = \mathbf{p}_i^t - \mathbf{p}_j^{t-1}$ ; }

(DP)

for( $\tau = t - 2; \tau \geq t - N + 1; \tau--$ )
    for( $j = 1; j \leq n_\tau; j++$ ) {
         $\boldsymbol{\delta}_j^\tau = (\mathbf{p}_i^t - \mathbf{p}_j^\tau) / (t - \tau)$ ;
         $E_j^\tau = \min_{1 \leq k \leq n_{\tau+1}} (\|\boldsymbol{\delta}_j^\tau - \mathbf{d}_k^{\tau+1}\|^2 + E_k^{\tau+1})$ ;
         $\mathbf{d}_j^\tau = ((t - \tau - 1) \cdot \mathbf{d}_{k'}^{\tau+1} + \boldsymbol{\delta}_j^\tau) / (t - \tau)$ ; }

(End)

 $E_i = \min_{1 \leq k \leq n_{t-N+1}} (E_k^{t-N+1})$ ;

```

図 4.15 動的計画法の擬似コード

4.4.3 類似度の算出

特徴点の軌跡をクラスタリングするためには，それらの間の類似度を示す尺度を導入する必要がある．そこで，軌跡間に強いエッジ成分が存在する場合，それらの軌跡は同一の車両からもたらされたものである可能性が高いという性質に基づき，軌跡間のエッジの本数を類似度として採用する．そのため，それぞれの点間のエッジの有無を調べる必要がある．

入力画像 $I(\mathbf{x})$ 及び背景画像 $I'(\mathbf{x})$ にそれぞれ Sobel フィルタを適用して得られたエッジ画像を $S(\mathbf{x}), S'(\mathbf{x})$ とおくと，これらの差分をとることで車両のエッジ成分のみを抽出した車両エッジ画像 $\hat{S}(\mathbf{x})$ を得ることができる．図 4.16 は原画像 $I(\mathbf{x})$ 及び車両エッジ画像 $\hat{S}(\mathbf{x})$ の例である．



(a) 入力画像

(b) 車両エッジ画像

図 4.16 車両エッジ画像の作成

$\hat{S}(\mathbf{x})$ を利用し，画像上の 2 点 $\mathbf{p}, \mathbf{q} = (x, y)^\top$ 間のエッジ強度を定義する．図 4.17 は，エッジ付近を拡大した図である．ここで，車両エッジ画像 $\hat{S}(\mathbf{x})$ の 2 点 $\mathbf{p}, \mathbf{q}$ 間を結んだ直線上に存在する画素値を $e(k)$ ($0 \leq k \leq n-1$) で表すことにする．このとき，式 (4.7)(4.8)(4.9) で定義される $e(k)$ のエッジ強度に関する統計量 e_1, e_2, e_3 を用いて，エッジの強さを定義する．

- 平均

$$e_1 = \frac{1}{n} \sum_k e(k) \quad (4.7)$$

- 理想的なエッジ強度 e_{ideal} からの乖離度

$$e_2 = \sqrt{\frac{1}{n} \sum_k (d(k))^2} \quad (4.8)$$

$$d(k) = \begin{cases} e_{ideal} - e(k) & \text{if } e_{ideal} - e(k) > 0 \\ 0 & \text{otherwise} \end{cases}$$

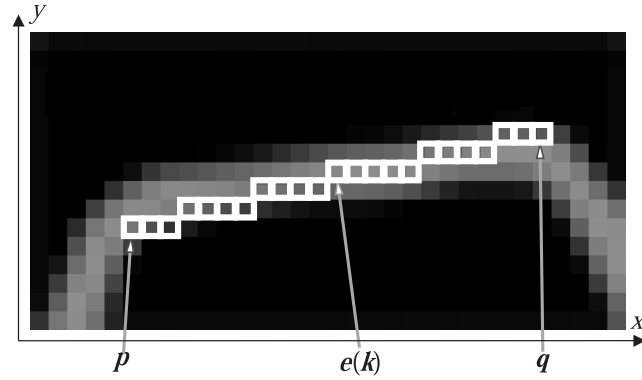


図 4.17 点間のエッジ成分

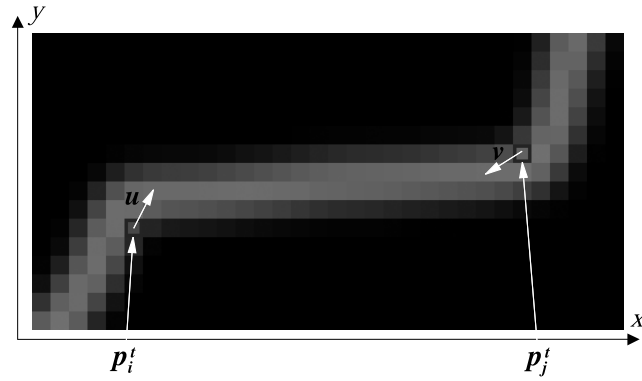


図 4.18 ずれの補正

- 隣接点間の差分

$$e_3 = \sqrt{\sum_k (e(k) - e(k-1))^2} \quad (4.9)$$

e_1 は p, q 間の平均値であり、大きいほどエッジらしいと言える。また、 e_2 は理想的なエッジ強度 e_{ideal} に近いかなを示す統計量であり、これは小さいほど良い。 e_3 はエッジ強度の変化量に関する項であり、これはエッジが途切れた場合に大きくなる。つまり、小さいほど途切れのない、安定したエッジであると言える。そこで、 p, q 間のエッジの強さを式 (4.10) で定義する。

$$e'(\mathbf{p}, \mathbf{q}) = -ae_1 + be_2 + ce_3 \quad (4.10)$$

a, b, c は正の定数である。 p, q 間に強いエッジが存在する場合、 $e'(\mathbf{p}, \mathbf{q})$ は小さくなる。

さて、このとき $\mathbf{p}_i^t, \mathbf{p}_j^t$ 間のエッジ強度は $e'(\mathbf{p}_i^t, \mathbf{p}_j^t)$ で求められるが、特徴点は厳密にエッジの両端に位置せず、若干ずれた位置から検出されることが多い (図 4.18)。そのため、なるべくエッジが強く検出されるようにこのずれを補正した方が、より正確にエッジの強さを求めることができると考えられる。そこで、このずれの量として $\mathbf{u}, \mathbf{v} = (x, y)^\top$

を新たに導入する．このずれの量が大きくなりすぎないように，エッジ強度にペナルティを与える項 e_4 を導入する．

- ずれによるペナルティ

$$e_4 = \|\mathbf{p}_i^t - \mathbf{u}\|^2 + \|\mathbf{p}_j^t - \mathbf{v}\|^2 \quad (4.11)$$

e_4 を用いて， $\mathbf{p}_i^t, \mathbf{p}_j^t$ 間のエッジの強さを式 (4.12) で定義する．

$$e(\mathbf{p}_i^t, \mathbf{p}_j^t) = \min_{\mathbf{u}, \mathbf{v}} (-ae_1 + be_2 + ce_3 + de_4) \quad (4.12)$$

式 (4.12) は $\mathbf{u}, \mathbf{v}$ に関する最適化問題であり，ここでは Local Search 法 [47] により $\mathbf{u}, \mathbf{v}$ を決定している．

$e(\mathbf{p}_i^t, \mathbf{p}_j^t)$ が閾値 e_{th} を超えるとき，その点間にはエッジがあるとみなす．軌跡 L_i, L_j 間の類似度 S_{ij} は，各フレームでの特徴点間に存在するエッジによる連結数となる． S_{ij} が大きいほど， L_i, L_j は同一の車両からもたらされた軌跡である可能性が高いと考えられる．

4.4.4 グラフの構築

以上で求めた軌跡群とそれらの類似度を用いて完全グラフ G を定義する．まずグラフの頂点集合 V を定義する．求めた軌跡 L_i が頂点となるが，このとき直線近似精度 E_i が悪いような L_i は軌跡として不適切であるため頂点から除く．式 (4.13) にグラフの頂点の定義を示した．

$$V = \{L_i | 1 \leq i \leq n_t, E_i < E_{th}\} \quad (4.13)$$

ここで， E_{th} は軌跡の精度に関する閾値であり，近似精度がこの値よりも悪い場合は頂点として登録されない．この段階でノイズなどの不要な成分からもたらされた軌跡が棄却されることになる．

次に，グラフの辺 E を定義する．

$$E = \{e_{ij} | \forall L_i, L_j \in V\} \quad (4.14)$$

辺は V の全ての要素間に関して定義されるため，このグラフは完全グラフとなる．さらに，類似度 S_{ij} を用いて各辺の重み関数 $w(e)$ を次のように定義する．

$$w(e_{ij}) = S_{ij} \quad (4.15)$$

4.5 グラフ分割

4.5.1 軌跡群のクラスタリング

図 4.19 は動画像から構築した完全グラフの例である．このように，完全グラフ内に強い連結関係を持った頂点群が車両の台数分だけ現われる．そこでこの連結関係に基づき，頂点群を車両毎に分割すればよい．



図 4.19 完全グラフの例

グラフ分割とは，クラスタリング手法の一種であるともいえる．クラスタリング手法には他にも様々な種類があり，その代表的な例として「ユークリッド距離に基づく手法」と「モデルに基づく手法」がある．

k-means 法，LBG アルゴリズムはユークリッド距離に基づく手法の代表例である．これらは，クラスタリングの対象となる要素を N 次元のベクトル $\mathbf{x}$ で表し， N 次元の特徴空間内の分布を最も良く近似する代表ベクトル（セントロイド）を求めることで個々の要素をクラスタリングするものである．これらの手法では，要素間の類似度は特徴空間内のユークリッド距離で測られる．そのため，特徴ベクトルをどのように定義するかがクラスタリングの結果を大きく左右する．本章においては，特徴点の軌跡同士の類似度として軌跡間のエッジの強さを採用したが，次章以降ではこれに加え軌跡の位置関係も含めた総合的な評価尺度を導入することでクラスタリングの安定化を図る．そのため，このクラスタリングの問題をユークリッド距離に基づく手法で解くのは容易ではない．それは，これら複数の類似尺度を含めた総合的な特徴空間を構成するのが非常に困難であるからである．特徴ベクトルを何次元とするか，ベクトルの N 個の要素をどのように定義するのか，また各要素の軸のスケールをどのように揃えるかなど決めるべき項目は数多い．特徴空間によっては主成分分析による次元削減が必要となる可能性もありえる．一方，グラフ分割では要素間の類似度を示す評価尺度が重みとして与えられているだけでよく，数学的記述は

非常に簡易である。要素間の類似度を複数の評価尺度で表そうとする場合、グラフとして記述する方が適切である。

モデルに基づくクラスタリング手法とは、要素群を先験知識の数学的記述である「モデル」へとフィッティングすることでクラスタリングする方法のことである。この種の手法は、事前に対象の形状モデルを網羅的に準備できる場合には非常に強力である。しかしながら交通流計測の場合、現れる車の形状は多種多様であり、それら全てに対してモデルを用意するのは非現実的である。しかしながら、網羅的にモデルが準備できなければモデルに合わない対象は検出できなくなってしまう。本論文では、道路の状況や撮影位置、車両の形状に仮定を置かないことで、様々な環境下でも動作する交通流計測手法を提案することを目的としているが、この「撮影対象に仮定を用いない」という趣旨にも大きく外れることになる。グラフ分割は対象物の形状に関する仮定を必要としないため、本手法の目的に良く合致している。撮影環境への耐性を追及するという目的のもとでは、車両の形状モデルを用意しないほうがよいと考えられる。

4.5.2 グラフ分割の解法

グラフ分割の解法は幾つか提案されているが、初期の研究の成果としてメタ戦略 [47] が有効であることが明らかになっている。文献 [47] で言及されている通り、メタ戦略は簡易でロバスト性が高く、アルゴリズム内部で要されるパラメータが多少変化したとしても性能が大幅に劣化しないという性質がある。特に、グラフ分割に対する組み合わせ最適化の解法としては Tabu Search による解法 [48] が有効であると報告されている。しかし、文献 [48] による方法はグラフを 2 分割することを対象としたものである。本手法はグラフを車両の台数分だけ分割する必要があるため、未知の k 個のグラフに分割するアルゴリズムを提唱する必要がある。

そこで、2 分割を繰り返し適用し任意の k 分割を得る Greedy Splitting Algorithm [45] を併用することで、Tabu Search による任意の k 分割を得ている。以下、分割のアルゴリズムについて解説する。

Tabu Search による 2 分割

Tabu Search 法とは、Local Search 法の近傍解への移行手順に手を加えたものである。TS 法は短期間内に移行した解をタブーリストに記憶しておき、最近遷移した解への移行を回避することで局所解からの脱却力を強め、大域解を得ようとするものである。この種の組み合わせ最適化アルゴリズムを適用するためには近傍構造及び評価関数をそれぞれ定義する必要がある。

- 近傍構造の定義

今、グラフ G を 2 つのグラフ $G_1 = (V_1, E_1)$, $w_1(e)$ 及び $G_2 = (V_2, E_2)$, $w_2(e)$ に分割することを考える. 組み合わせ最適化問題に対するメタ戦略は, 解の近傍の性質が似ており評価量が近くなることを前提としている. この前提条件を踏まえ, 本手法では V_1 から V_2 へ, あるいは V_2 から V_1 へ頂点をひとつだけ移動して得られる分割を近傍として採用した. つまり, 分割 (V_1, V_2) の近傍は常に $\|V\|$ 個存在することになる.

- 評価量

車両の軌跡は, 同一の車両に属する場合は強く連結しており, 異なる車両同士の場合の連結力は弱い. そこで, 分割内の辺重み総和を頂点数で正規化した値 “within” が大きく, 分割間の辺重み総和を分割間の辺数で正規化した値 “between” が小さいような最適な分割 G_1, G_2 を探せばよい (式 (4.16)(4.17)).

$$\text{within} = \frac{\sum_{e \in E_1} w_1(e) + \sum_{e \in E_2} w_2(e)}{\|V_1\| + \|V_2\|} \quad (4.16)$$

$$\text{between} = \frac{\sum_{e \in E} w(e) - \left(\sum_{e \in E_1} w_1(e) + \sum_{e \in E_2} w_2(e) \right)}{\|V_1\| \cdot \|V_2\|} \quad (4.17)$$

within 及び between を用い, 分割の評価関数 $cost$ を式 (4.18) で定義する.

$$cost = \frac{\text{between}}{\text{within}} \quad (4.18)$$

上記 $cost$ を最小にする分割が求める分割である. 以上の定義に基づき Tabu Search を適用し, 最適な分割を得る.

GSA による k 分割への拡張

前述の 2 分割アルゴリズムを GSA(Greedy Splitting Algorithm) を用いて未知の k 分割をできるように拡張する. 図 4.20 は GSA による k 分割の過程を示したものである. まず G に 2 分割アルゴリズムを適用し, 二つの部分グラフ G_1, G_2 を得る. 次に, G_1, G_2 の両方に 2 分割アルゴリズムを適用し, 分割の評価値がより小さかった方の分割を採用する. これを評価値が閾値を超えるまで繰り返すことで, 任意数の分割を得ることができる. このとき, 頂点数が少ない分割はノイズとみなし, 削除する. 最後に残った分割が動画像内の車両を示している.

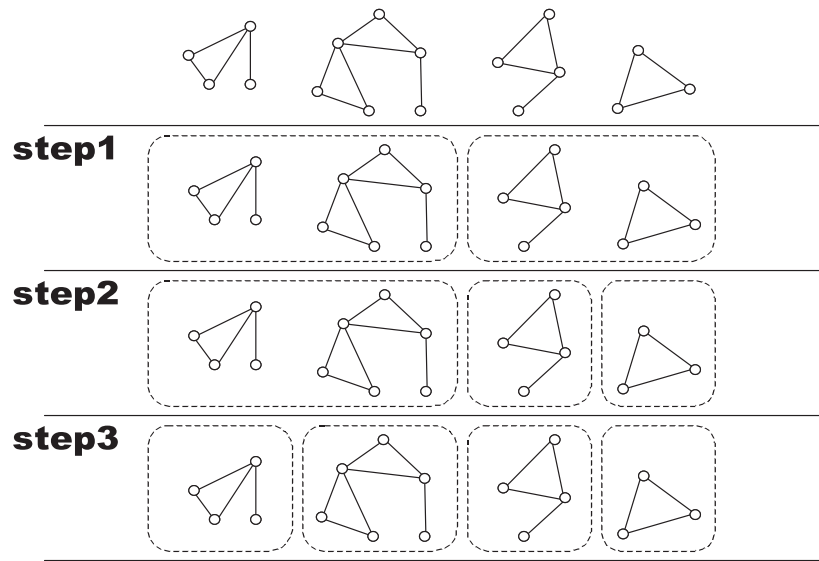


図 4.20 GSA

4.6 実験

4.6.1 実験条件

撮影位置の異なる 30fps の動画を 7 つ用意した (図 4.21). (a)~(d) は昼間に撮影されたものである. (e)~(g) は夜間に撮影されたものであり, (f)(g) のシーンでは小雨が降っている. 特に, (g) はヘッドライトがカメラの方向に向かっており, 7 つの動画の中で最も劣悪な照明条件のもとで撮影されている. (a)(b) からそれぞれ無作為に 10 $Unit$ を選び, それらを用いて閾値の調節を行った. 実験にはこの共通の閾値を用いた. また, 式 (4.1)(4.2) において $N = 20$ とした. 7 つの動画に対し, 100 フレーム間隔で検出アルゴリズムを適用した. すなわち, $Unit_{t-N+1}^t$ において, $t = 100k$ ($k = 0, 1, 2, \dots$) となる $Unit$ に対して検出処理を行った. これは, 閾値の調節に用いた $Unit$ とは別のものを使用した.

4.6.2 実験結果と検討

表 4.1 に計測結果を検出成功 (Success), 過検出 (False Positive:FP), 未検出 (False Negative:FN) の三つに分類し, 撮影条件, 処理 $Unit$ 数とともにまとめた. また, 図 4.21 に車両の検出結果を矩形で示した. 図 4.22~4.28 は車両認識の結果と構築したグラフを併記したものである.

表 4.1 検出精度

Scene	Time	Weather	Units	Success	FP	FN
(a)	Daytime	Fine	33	97	22	2
(b)	Daytime	Fine	24	19	2	7
(c)	Daytime	Fine	39	21	1	1
(d)	Daytime	Fine	39	12	2	2
(e)	Night	Fine	99	24	5	6
(f)	Night	Rainy	39	52	6	3
(g)	Night	Rainy	12	20	32	6

撮影位置に関するロバスト性

図 4.21(a)～(d) に示される 4 シーンはそれぞれ車両の側面, 上部, 前面を捉えたものであり, 画像上での車両の見え方が著しく異なっているが検出に成功している.

交通流計測は広域に渡って行われることが要求されており, (a)～(d) ような多様な撮影位置でも動作可能であることが期待される. 画像からの車両検出の枠組みの一つとして車両の形状モデルを用いた方法が有効であると考えられているが, (a)～(d) のような動画像全てに適用するためには撮影位置に応じたモデルを用意する必要がある, これは一般に困難である. だが本手法は特徴点の軌跡群を車両であると考えており, 特に車両の形状モデルなどを仮定していないため, 撮影位置に依存しない適用が可能である.

だがその一方で, 表 4.1(a)～(d) が示すように車両の過検出を引き起こしやすいという問題があることも否めない. それは (a) で特に顕著である. これは, 画像中で車の面積が大きくなり過ぎたときに軌跡間の距離が広がってしまうことで, それらの軌跡同士が十分大きな類似度を持てなくなることによる. 類似度が弱い場合, その部分でカットされてしまい, 結果的に 1 台であった車両を複数台として認識してしまうことになる. つまり, 撮影時には対象までの距離を十分に確保し, 広域をとらえるようにしておくことで検出に成功しやすい.



図 4.21 検出結果

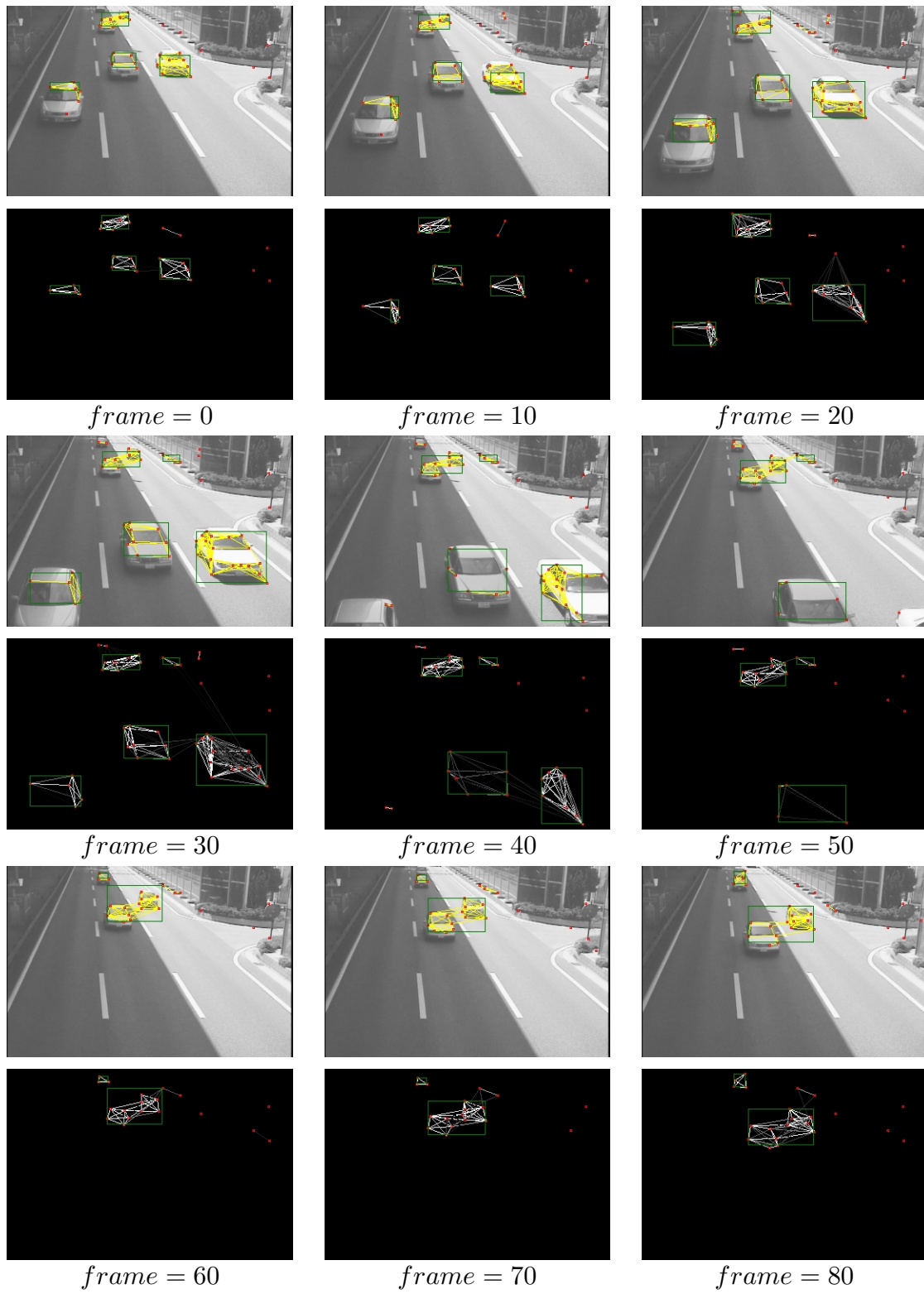


図 4.22 (a) 手前に直進 1

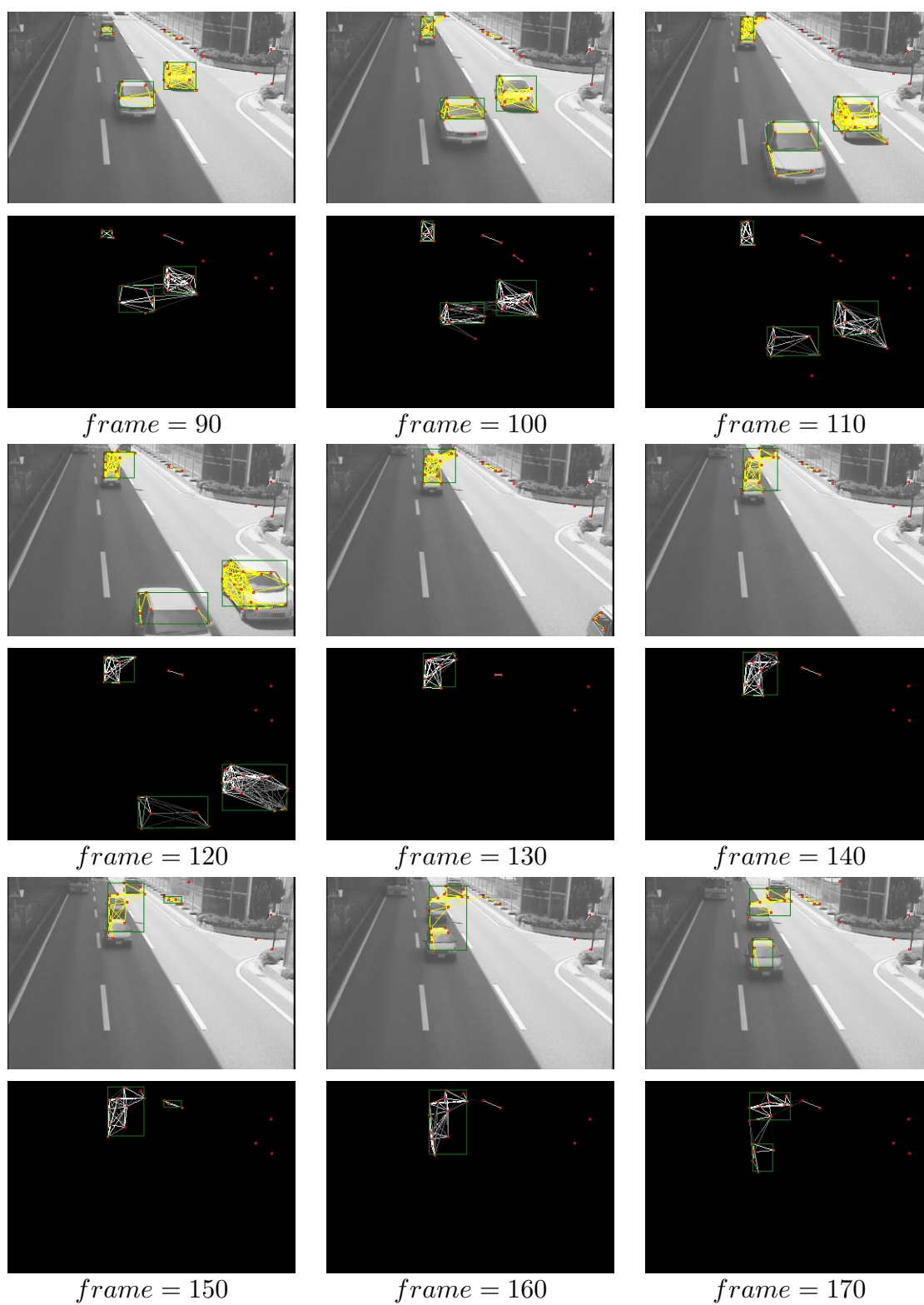


図 4.23 (a) 手前に直進 2

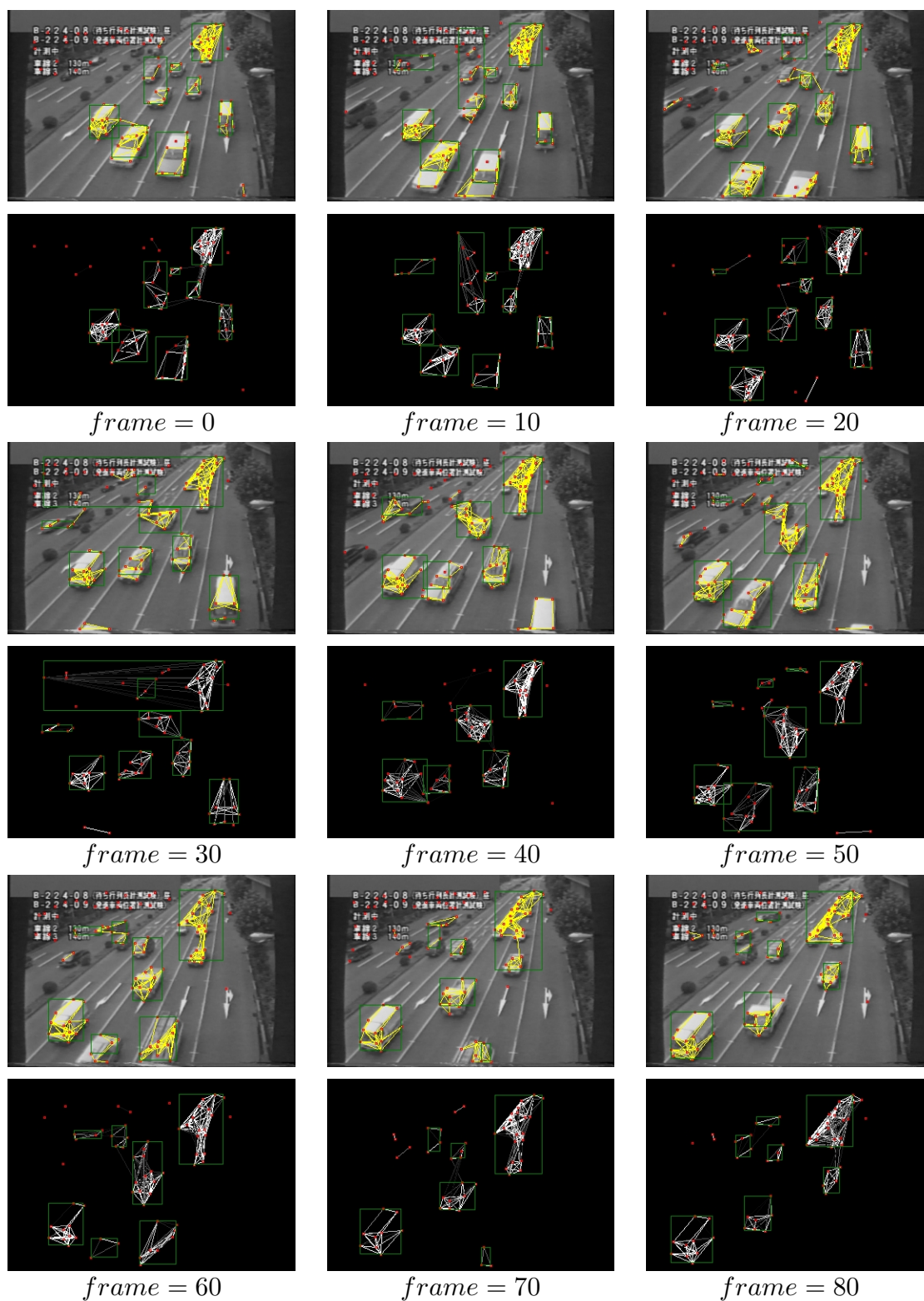


図 4.24 (b) 俯瞰画像 1

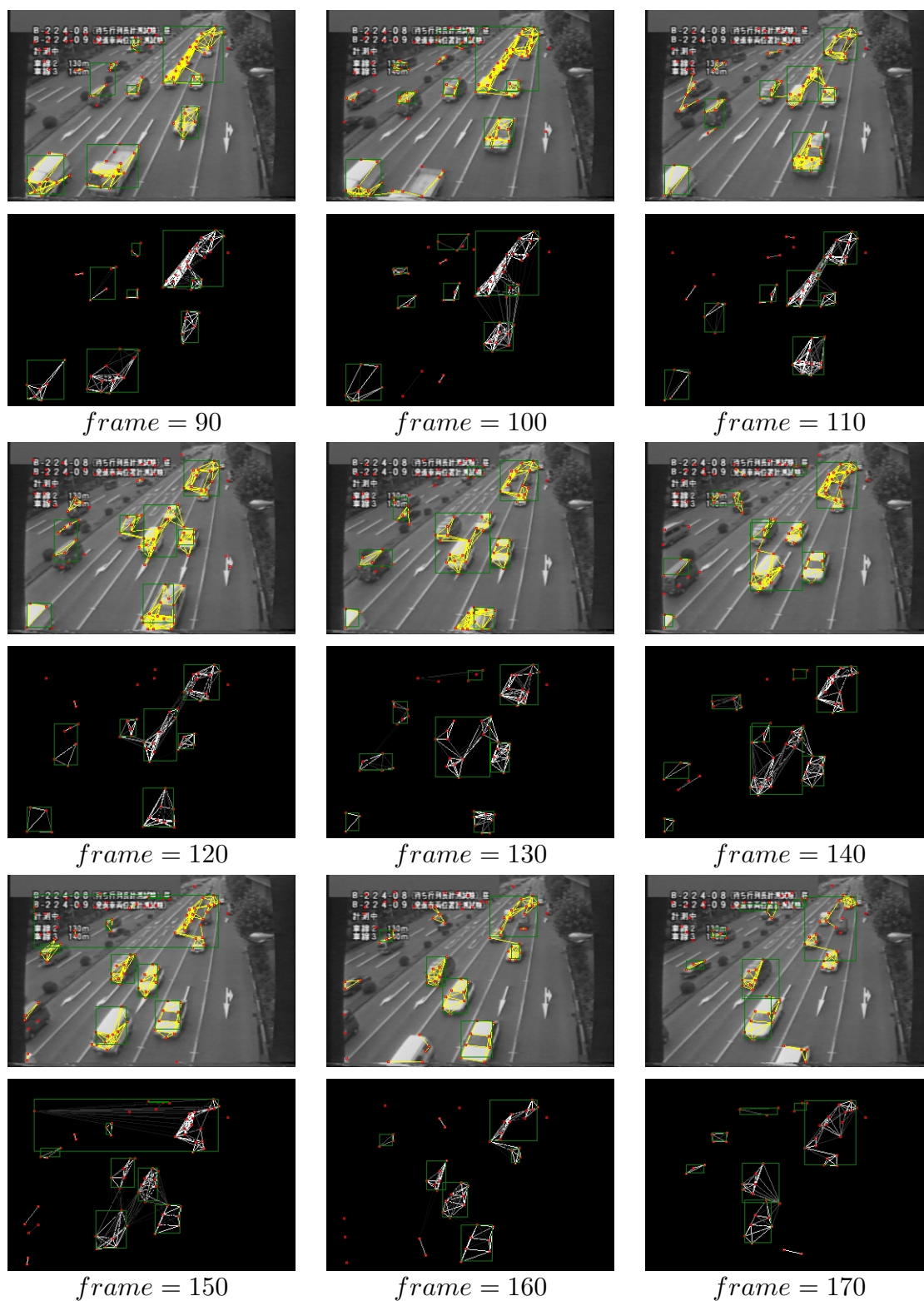


図 4.25 (b) 俯瞰画像 2

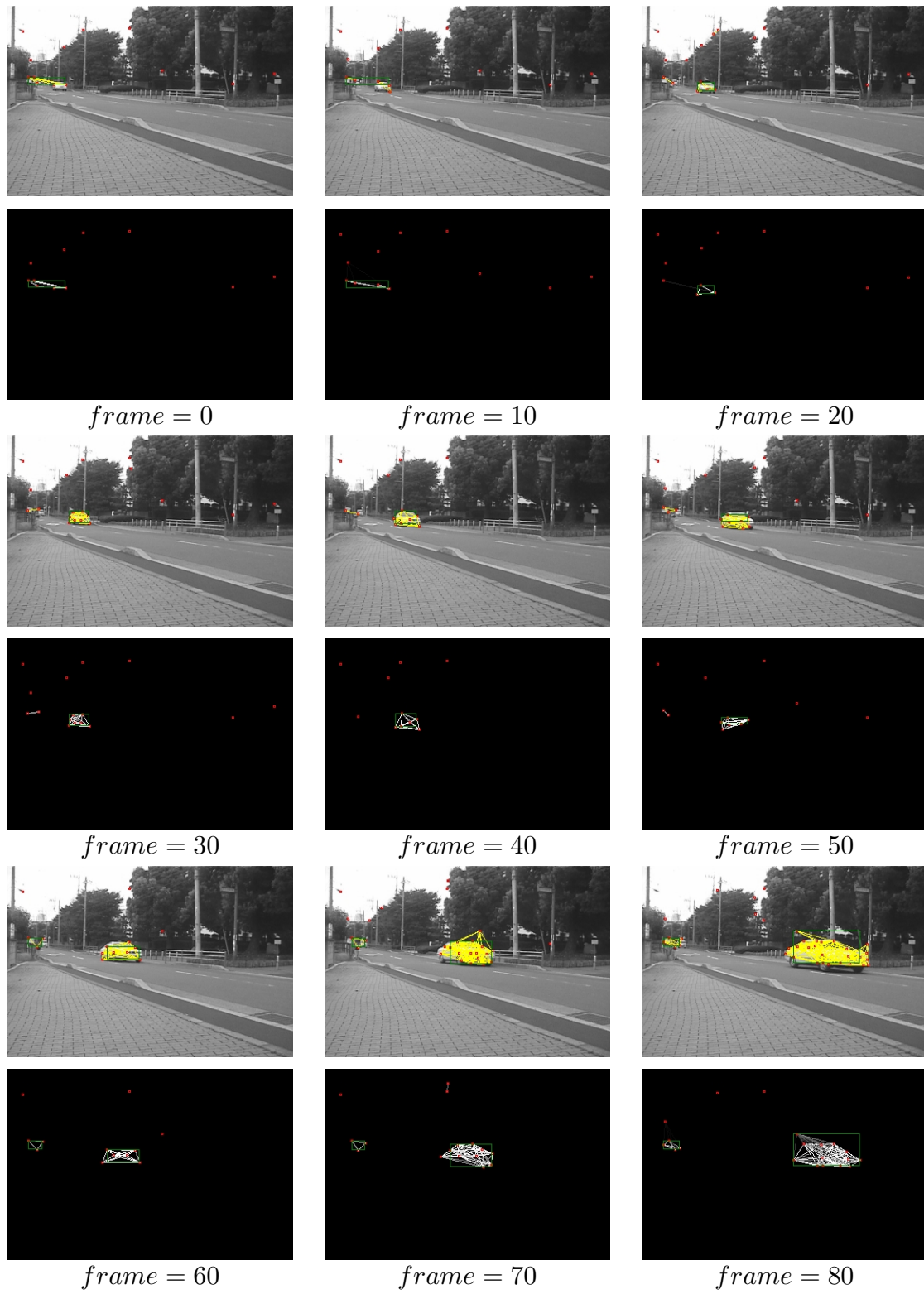


図 4.26 (c) 右側面

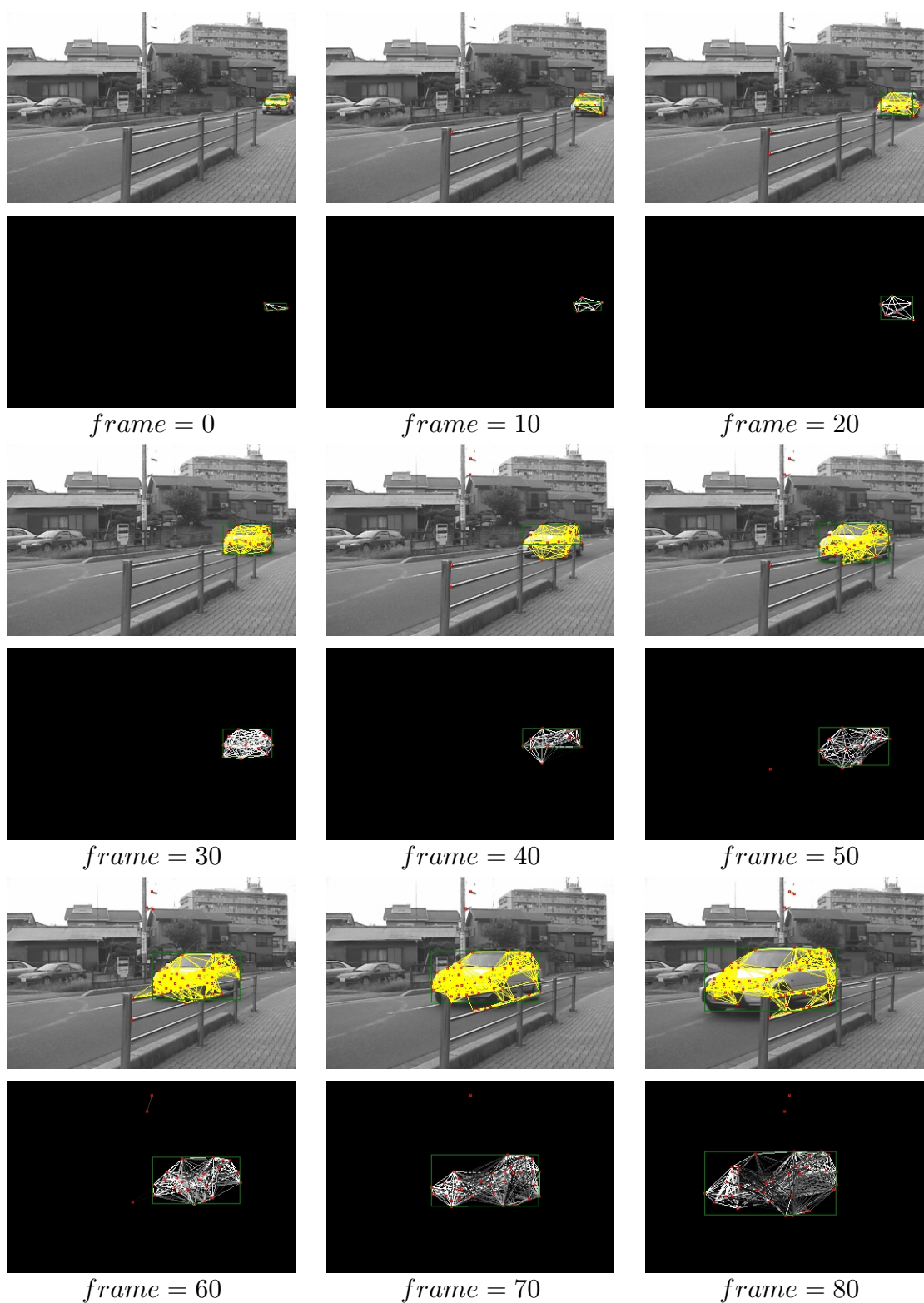


図 4.27 (d) 左側面

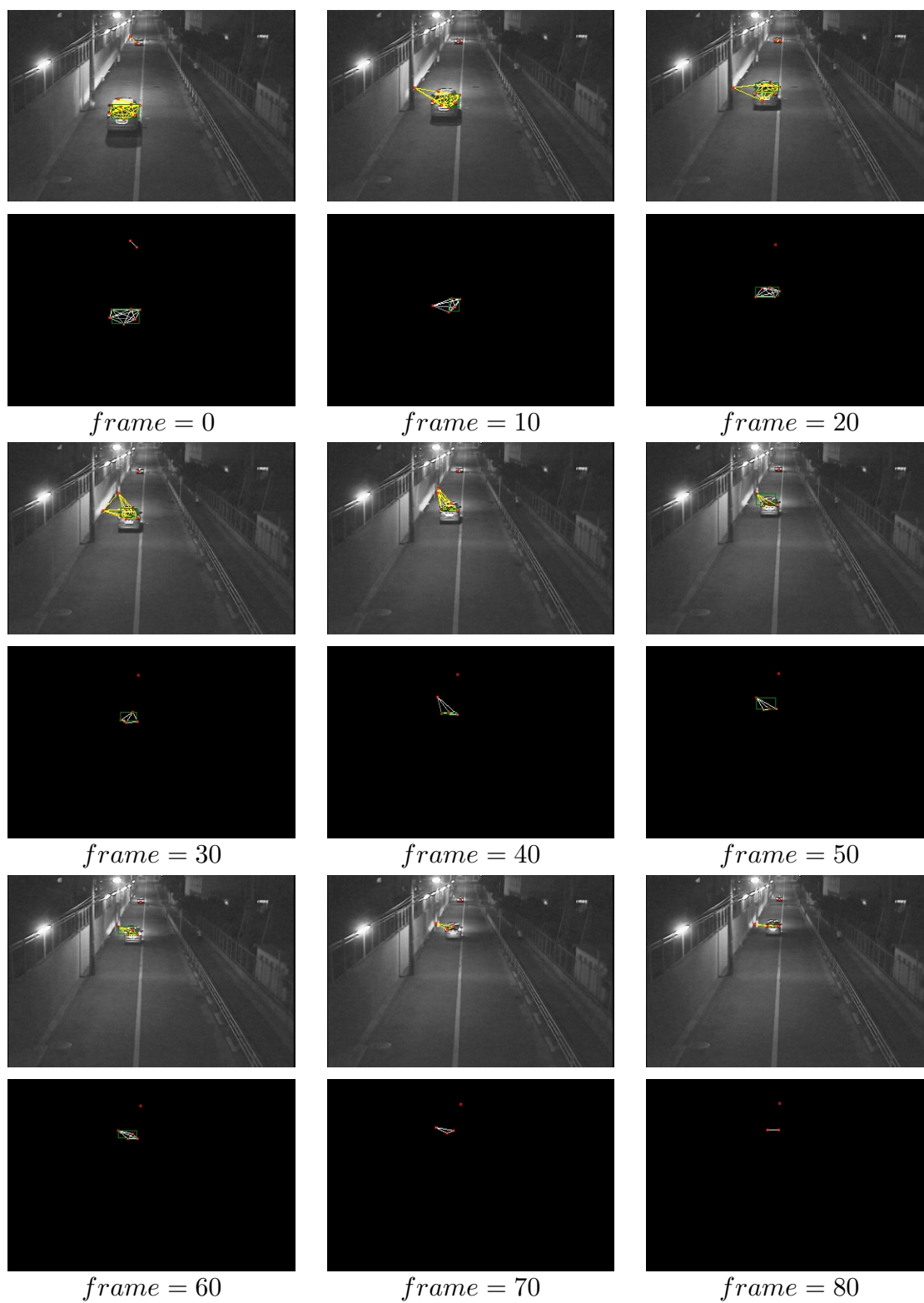


図 4.28 (e) 夜間

照明条件に関するロバスト性

図 4.21(e)～(g) は夜間に本システムを適用した結果である。これらのうち、(e)(f) は昼間と同程度の性能を示すことが表 4.1 から読み取れる。(f) では小雨が降っており、路面がややヘッドライトの光を反射しやすい状況であったが特に問題はなかった。夜間はノイズ成分や車両のヘッドライトなどの影響により明度値が安定しないが、特徴点検出処理はこうした明度変化に対して頑健である。それゆえ、システムの性能は夜間においても劣化せず、安定した性能を発揮する。

(g) は車両のヘッドライトがカメラの方向を向いており、さらにその光が路面を覆う雨で反射しているという非常に劣悪な照明条件下で撮影したものである。この画像列では、ヘッドライトが濡れた路面で鏡面反射し路面上に鋭い濃淡成分を生んでしまうため、不要な特徴点が検出されてしまった。また、ヘッドライトの影響により車両の輪郭のエッジ成分が相対的に弱くなってしまい、類似度がうまく生成されなかった。これらの理由から(g) では正常な検出が出来なかった。このように、サチュレーションが起こるほどに強い光が照射されるような条件が想定される場合では、ダイナミックレンジの広いカメラを用いるなどの工夫が必要であると思われる。

なお、特徴点検出処理には検出の感度を示す閾値 λ_{th} の設定が必要となるが、 λ_{th} を低めにすることで特徴点の過検出が起こったとしても、後に軌跡の直線性を評価することでノイズ成分による不要な特徴点が棄却されるため、 λ_{th} は照明条件にかかわらず低めに設定しておけば良い。図 4.29 に特徴点が棄却される様子を示した。このように、直線性を満たさないような特徴点群は削除され、車両の軌跡として妥当な特徴点群のみが処理の対象になるため、前段の処理の内容を後段の処理にて修正するような処理が実現されていることが分かる。

グラフ分割に関する検討

図 4.30 は画像中に 3 つの移動物体が現れたときの完全グラフである。理想的には異なる車両間の類似度が零になることが望ましいが、現実的には車両の近接、ノイズ、軌跡の誤検出などの要因により、異なる車両間にも弱い重みを持った辺が生じてしまう。だが、グラフ分割アルゴリズムは辺重みの弱い部分をカットするものであり、ある程度の過剰な連結が生じたとしても、それらは問題なくカットされる。つまり、ここでも同様に前段の処理の内容を後段の処理にて修正するようなアルゴリズムになっており、このために各処理の閾値に依存せず、ロバストな車両検出を可能なものとしている。

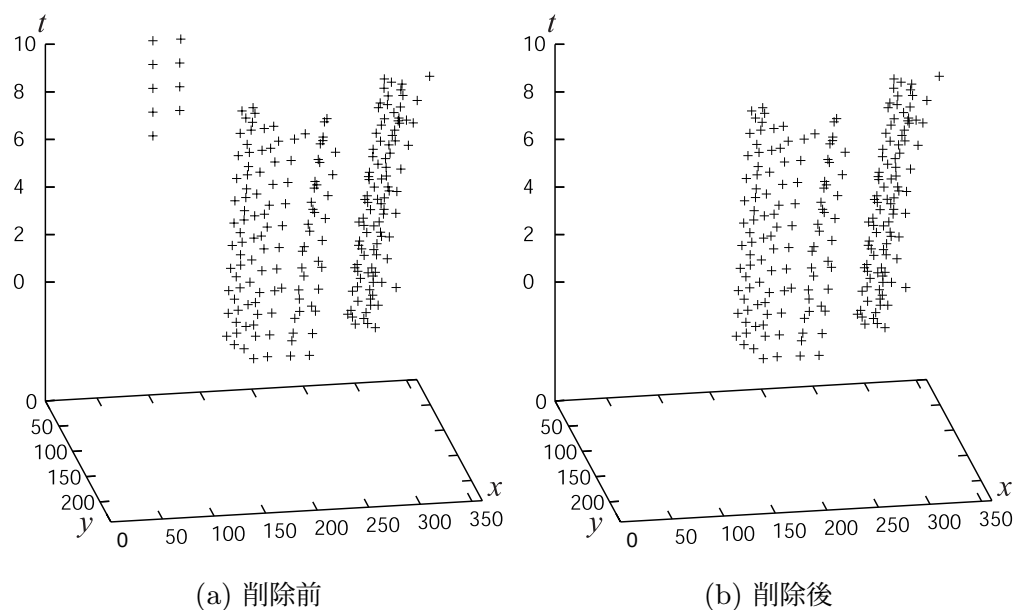


図 4.29 軌跡の直線性の評価による特徴点の棄却



図 4.30 3 台の車両における完全グラフ

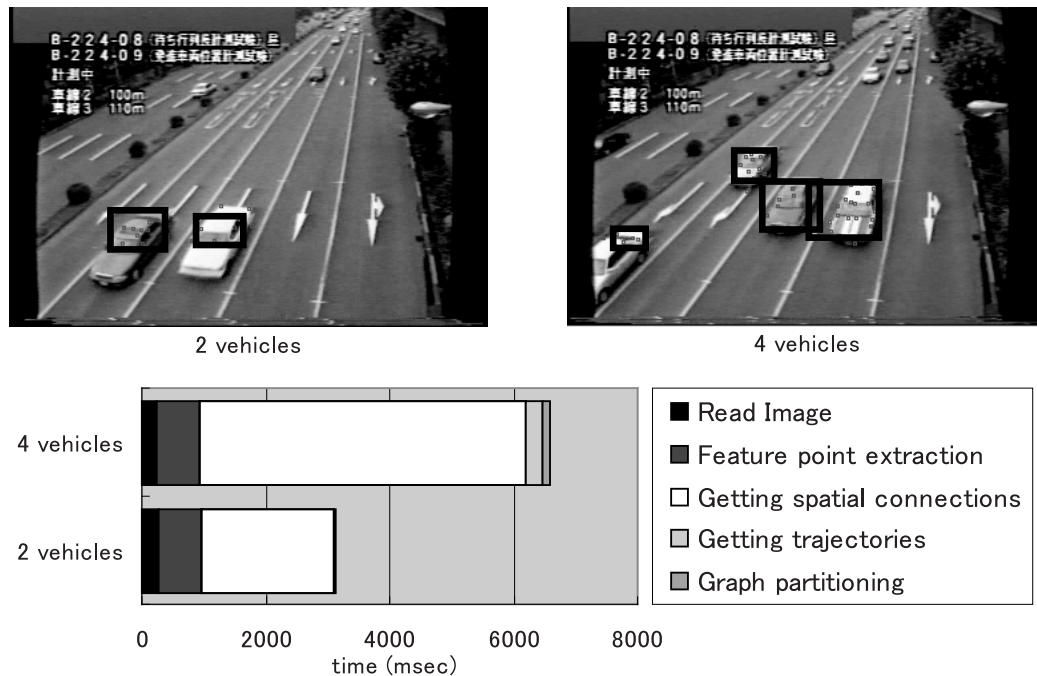
実行速度の検討

本手法は軌跡群をクラスタリングすることで車両の追跡を実現しているため、抽出した軌跡の本数によって処理時間が異なる。そこで、車両の台数が異なるシーンにそれぞれシステムを適用し、その実行速度を計測した（図 4.31）。図中の棒グラフは 20 フレームを処理するのにかった時間を示している。この図から、処理過程の中で大部分の計算時間を占めているのは類似度の算出処理であることが分かる。この処理は特徴点の組み合わせを列挙し、その全ての組み合わせに関して類似度を算出するものであり、特徴点数を n とす

ると、その計算量は $o(n^2)$ で増加することが予想される。

だが実際の実装では、特徴点間の距離が遠い場合、それらの点は同一の車両のものとは言いがたいため、類似度の計算を省略し、0 に設定することで高速化を図っている。そのため、計算量は単純に $o(n^2)$ で増加をするわけではない。図 4.31 の例の場合、軌跡数に 4 倍の差があるが、実行時間の増加は 2 倍程度に抑えられている。また、車両エッジ画像を低解像度化し、点間の画素値 $e(k)$ の数を減らすことでも高速化を期待できる。

この部分の処理は非常に並列性が高く、点間の類似度はそれぞれ独立に計算できる。そこで複数 CPU によるマルチスレッド化やハードウェア化による高速化が望めるので、30fps 程度の動画像であればリアルタイム化が可能であると考えられる。



	Trajectories	Total time (msec)	msec/frame
4 vehicles	67	6652.15	332.6
2 vehicles	17	3148.69	157.4

図 4.31 実行速度

4.7 総括

現在、画像処理に基づく交通流計測システムが幾つも提案されており、その中には実用化されているものもある。だが、今後の交通流計測システムは広域に渡って運用可能であることが要求されており、そのためには撮影対象に依存しない車両検出法を確立する事が重要である。この要求に応えるため、本論文では多様な撮影環境下においても適用可能な車両検出アルゴリズムを提案した。本手法は多様な撮影環境下においても同一のアルゴリズム、同一の閾値で動作するものであり、車両の位置決定をグラフ分割問題に帰着する事で安定な検出を実現した。条件の異なる7シーンに対してシステムを適用する事でその有効性を確かめた。

本手法の特徴は、特徴点の軌跡とそれらの類似度を求め、グラフ分割処理を適用することで車両の位置を決定することにあった。本章で述べた方法では、軌跡を求める際には動的計画法を用い、また類似度を求める際にはエッジ情報を用いた。グラフ分割の手法としては Tabu Search と GSA を併用する方法を用いた。だが、これら各段の処理は別の手法で置き換えることも可能である。例えば、本章では特徴点の軌跡を求める際に動的計画法を用いた。これは、車両は局所時間内で等速直線運動をするという最も単純な仮定を導入したためである。しかし、この仮定を外したい場合には、KLT Tracker[49] やその前身である LK Optical Flow[50]、あるいはテンプレートマッチングによる追跡手法などが有効であると考えられる。ともあれ、いずれの手法を用いるにしろ、特徴点の軌跡をグラフ分割でクラスタリングするというアプローチが車両認識をするうえで非常に有効であることが本章における議論で示されたといえる。

次章では、特徴点の軌跡群とグラフ分割の戦略により、交通流計測を行うための手法を提案する。本章で述べた手法で得られる車両の認識結果は、局所フレームごとにそれぞれの車両の位置を決定できるものであった。これはつまり、車両の認識結果は時間軸において離散的に得られるということである。次章では、車両が画像上に現れてから画面外に消えるまでの間、個々の車両の一連の軌跡を追い、車両の台数、および移動軌跡を取得する手法について検討する。

第 5 章

動画像を入力情報とする車両計数

5.1 概要

道路上を走行する車両の移動軌跡を得ることができれば，交通流量の把握，事故検知および違法車両検出までを含めた高度な道路交通流監視システムを実現できると期待される．そこで本章では，4 章で述べた車両認識手法を時間軸方向に拡張し，動画像を入力情報とする車両計数のための手法を提案する．本章で述べる手法は，3.1 節で述べた 5 つの性質のうち，前章で焦点を当てた次の 1～3 の性質に加え，4 番目の性質を満たす (下記太字部分)．

1. **1 台のモノクロカメラで計測が可能**
2. **カメラの設置位置に関する制約が緩い**
3. **照明条件の変化に強い**
4. **車両の移動軌跡を取得可能**
5. オクルージョンに強い

4 章では，20 フレーム程度の短い動画像を用いることで，異なる時間帯，天候，道路状況に対しても適用できる車両認識手法を提案した．本章では 4 章で提案した手法を拡張し，車両が監視領域に進入してから出てゆくまでの間，車両を追跡するマシンビジョン技術を提案する．本章で述べる手法の基本的な原理は 4 章で述べた手法と同一であり，特徴点の軌跡群を抽出し，グラフ分割処理で軌跡群をクラスタリングすることで車両の位置を決定する．

5.2 車両計数の原理

特徴点の軌跡群をグラフ分割でクラスタリングするという基本的なアイディアは4章で述べた手法と同様であるが、4章では局所時間内での車両位置を決定していたのに対し、本章では車両が監視領域に進入してから出てゆくまでの間の全てのフレームで車両位置を決定することを目的としている。

本章で述べる手法は特徴点の軌跡群抽出、軌跡群のクラスタリングによる追跡という2段階の処理から成る。まず、画像から特徴点を検出し、背景特徴点を除去する。次に車両の特徴点を時間方向に追跡し、軌跡群を得る。この軌跡群を頂点 V_t 、軌跡間の類似度を辺の重み $w_t(e)$ とした重み付き完全グラフ $G_t = (V_t, E_t)$ 、 $w_t(e)$ $e \in E_t$ (E_t は辺、 t は時間) を毎フレーム構築する。軌跡間の類似度は、同一の車両から検出された軌跡である可能性が高い場合、値が高くなるように設定しておく。この G_t を分割することで軌跡群をクラスタリングし、時間 t での車両の位置を決定する。このとき、 G_{t-1} の分割結果を利用し、 $w_t(e)$ を修正することでクラスタリングの安定化を図る。最後に G_t, G_{t-1} のそれぞれの分割結果を対応付けることで追跡を実現する。

特徴点の検出処理は照明条件に依存しにくいという性質があり、また軌跡間の類似度には撮影位置に依存しない量を採用しているため、完全グラフは撮影対象や照明条件によらない不変的なものとなる。それゆえ、追跡結果は撮影環境にロバストなものとなる。また、グラフ分割は大域的な構造が結果によく反映されるクラスタリング手法であるため、仮に類似度の算出が失敗し、本来別々の車両に属するべき軌跡同士のうちの幾つかが強い辺重みで連結し合ってしまうようなことがあったとしても、そのような局所的なエラーの影響を受けにくい。すなわち、前段の類似度算出処理で失敗が起こったとしても、それを後段の分割処理で修正できるという利点を持つ。この性質により特徴点の軌跡は多少のノイズには影響されること無く正確にクラスタリングされる。

本章で述べる手法は、4章で述べた手法を時間方向に拡張しており、手法の枠組みが大きく異なる。すなわち、(a) グラフの頂点の定義が、局所時間内の近似直線から時間方向に特徴点列を接続した折れ線へと変更されたこと、(b) 前フレームまでの追跡結果を用いて辺重みを修正する仕組みが追加されたこと、(c) G_{t-1}, G_t の分割結果を対応付ける処理が追加されたこと、という以上3点で大きく異なる。また細かな改良点として、(d) 背景特徴点の棄却処理が、より正確に不要な点を削除できるようになったこと、(e) 軌跡間の類似度の計算法が簡略化・高速化されたこと、(f) 分割結果がロバストであり、かつ収束性に保証のある高速なグラフ分割法を適用したこと、などがある。

本手法は様々な撮影環境においても同一のしきい値で稼働できるという利点を持つ。そ

こで、本章で現れるしきい値を全て $T_{\text{Subscript}}^{\text{Superscript}}$ の形で記述し、これらの調整の仕方について 5.5.2 節で議論する。

5.3 特徴点の軌跡群抽出

5.3.1 車両特徴点抽出

J.Shi ら [49] の手法によると、入力画像 $I(\mathbf{p})$ の各画素 $\mathbf{p} = (x, y)^\top$ における特徴点らしさは式 (5.1)(5.2) で与えられる。

$$\mathbf{C} = \sum_{\mathbf{x} \in \mathcal{N}(\mathbf{p})} \begin{pmatrix} (I_x(\mathbf{x}))^2 & I_x(\mathbf{x})I_y(\mathbf{x}) \\ I_x(\mathbf{x})I_y(\mathbf{x}) & (I_y(\mathbf{x}))^2 \end{pmatrix} \quad (5.1)$$

$$f = \min(\lambda_1, \lambda_2) \quad (5.2)$$

$\mathcal{N}(\mathbf{p})$ は $\mathbf{p}$ の 8 近傍における位置ベクトルの集合、 λ_1, λ_2 は $\mathbf{C}$ の固有値であり、式 (5.1)(5.2) を画素毎に計算する。 f の値が局所最大であり、この値が T_λ^{fp} を超える場合、その点を特徴点とみなせる。 T_λ^{fp} は特徴点検出の感度に相当する。式 (5.1)(5.2) は、その近傍で複数の方向に明度勾配を持つ画素を特徴点とするものである。車両の角点は、その周囲の明度パターンがこの性質に合致するため、比較的安定に検出される。

夜間の道路交通流を撮影した動画像を処理する場合、コントラストの低下により特徴点を取りこぼすことが懸念されるが、これは T_λ^{fp} を低く設定し、特徴点検出の感度を上げることで回避できる。また夜間の場合、車両のヘッドライトの影響を回避することも重要である。だが、車両のヘッドライトが作る明度分布はその変化が平面的であり、角点のような鋭い微分値が生まれにくい。それゆえ路面に照射されたヘッドライトに対しては反応が鈍く、不要な特徴点は混入しにくい。こうして得た特徴点群には背景成分が含まれている。特徴点の取りこぼしを防ぐために抽出の感度はできるだけ上げておく必要があり、そのため混入する背景成分はますます増えることになる。つまり、本来必要である車両の特徴点を消してしまうことなく、背景成分のみを正確に除去する枠組みが必要となる。さらに、その除去に必要なしきい値はできるだけ照明条件に依存せず、一定の値を利用できることが望ましい。

そこで、検出した特徴点の周囲のテクスチャと、背景画像上で相当する位置のテクスチャとの間で正規化相関 [51] を計算し、この値としきい値 $T_{\text{nc}}^{\text{fp}}$ を比較することで背景成分の除去を行う。このときに用いる窓は $T_{\text{window}}^{\text{fp}} \times T_{\text{window}}^{\text{fp}}$ の正方形の領域とする。ここで正規化相関を用いる理由は、背景輝度の緩やかな変動やヘッドライトや街灯による輝度変化は線型変換であるとみなすことができ、正規化相関がこのような線型の変化に頑健であることによる。

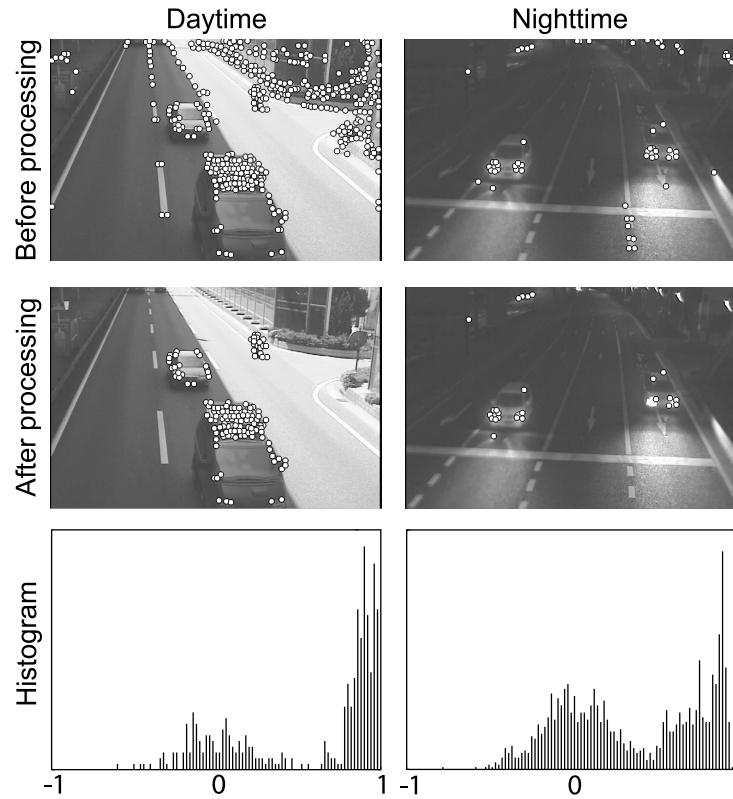


図 5.1 背景特徴点の棄却.

検出された特徴点が背景成分によるものならば、その点のまわりのテクスチャは背景画像と強い相関を示すが、車両などの移動物体から検出された特徴点の場合は、背景とは無相関になるはずである。図 5.1 は全ての特徴点で算出した正規化相関値をヒストグラムに示したものである。背景特徴点の場合は強く相関するため、値は 1 付近を中心とした正規分布に従う。その一方で、移動物体の特徴点の場合は無相関となるため、値は 0 付近を中心とした正規分布に従う。そのため、図 5.1 は双峰性を示すことになる。

この方法は、 T_{nc}^{fp} の設定が非常にロバストになるという利点がある。図 5.1 のヒストグラムは正規化された値であるため、いかなる照明条件であっても基本的には 0 と 1 付近を中心とした二つの正規分布が合わさったものとなる。図 5.1 にはそれぞれ照明条件が著しく異なる二つの事例を示したが、いずれもこの性質を満たしていることが分かる。そこで、 T_{nc}^{fp} は照明条件にかかわらず 0.5 付近に設定すればよい。ヒストグラムの双峰性により、背景特徴点のみが正確に棄却されるため、 T_{λ}^{fp} を低く設定することで混入した不要な点群はすべて棄却されることになる。それゆえ、 $T_{\lambda}^{fp}, T_{nc}^{fp}$ は照明条件にかかわらず一定の値で良く、しきい値の良し悪しが全体の性能にとって支配的な影響を及ぼすことが無い。これは撮影環境に対してロバスト性が要求される場合には重要な性質である。

4 章では、画像中の全ての画素 p に対し、式 (5.2) を用いて入力画像と背景画像の特徴

点らしさ f, f' をそれぞれ算出し、 $f - f'$ が局所最大であり、しきい値を超えるような点を車両の特徴点としていた。つまり、背景画像上で特徴点らしさが高い位置にある特徴点を全て削除していた。だが、これは簡易的な手法であり、状況によってはうまく背景の特徴点を除去できない場合がある。例えば路面に道路標示などによる強い微分成分が存在し、その上を車両が通過するような場合、 f と f' は共に大きくなる。この場合、 $f - f'$ は小さな値になってしまい、検出漏れを起こすという問題が生じる。一方、本章で述べた処理では、まず車両の特徴点と背景の特徴点を区別することなくまとめて検出し、その後に入力画像と背景画像の局所的な明度分布を比較して不要な点を判別するというアプローチを採っている。この方法だと、道路標示上を車両が通過した場合でも、道路標示上の明度分布と車両の明度分布は著しく異なるため、誤って特徴点を消してしまうことはほとんど無い。この点で、4章で述べた手法よりも優れている。

以降、時間 t における入力画像から検出された特徴点の数を n_t 、 i 番目の特徴点を p_t^i と表記する。

5.3.2 車両特徴点の軌跡

5.3.1 節で述べた車両特徴点の検出処理を用いて時空間内の軌跡を決定する。4章では時空間内における特徴点群の分布の直線性に着目して特徴点の軌跡を求めた。だが、この方法は画像平面上で特徴点が等速直線運動をすることを仮定しているため、状況によっては動作しない場合が有りえる。そこで、実用化に向けた改良が必要である。

連続する2フレーム間で特徴点を追跡する方法としては、多くの研究者によって改良が進められてきた KLT Tracker[49] が有名である。KLT Tracker は追跡位置のずれが少なく、また1ピクセル以下の精度で特徴点の位置を求めることができるため、コンピュータビジョンの分野では幅広く用いられている。だが、KLT Tracker は処理速度の点で問題が残る。我々の予備実験によると、KLT Tracker は100個前後の特徴点を全て追跡するのに1フレームあたり数百 msec 程度の時間を要した。動画像を逐次的に処理してゆく場合、少なくとも15~30fps程度で動作することが必要である。すなわち、1フレームの処理時間に割ける時間は高々33~67msec程度である。個々の特徴点の追跡に与えられる計算時間は多く見積もっても10msec程度、可能であれば数 msec 程度で終わることが望ましい。^{*1}

そこで、差の二乗和 (SSD) を基準としたテンプレートマッチングによって車両特徴点

^{*1} 幸いにして、KLT Tracker の計算量は特徴点の数を n とすると $o(n)$ である。近い将来には計算機の処理性能が向上し、KLT Tracker をリアルタイムで適用できるようになると予測される。その際には KLT Tracker により特徴点群の軌跡を決定することが望ましいであろう。

の軌跡を決定する方法を採用した．この方法は，連続する2フレーム間で特徴点の対応を見つけるアルゴリズムとしては最も簡単な方法であり，非常に高速に計算できる．SSDによる方法は誤った対応を与える可能性も高いが，本手法では局所的なエラーは後段のグラフ分割で修正される仕組みになっているため，必ずしも正確な対応付けは必要ではない．

ここで，まず記号を整理する． $\mathbf{p}_t^i$ の軌跡は， $\mathbf{p}_t^i$ 及び時間 $t-1$ 以前における特徴点の列から成り，

$$l_t^i = \{\mathbf{p}_t^i, \mathbf{p}_{t-1}^{i_1}, \mathbf{p}_{t-2}^{i_2}, \dots, \mathbf{p}_{t-N+1}^{i_{N-1}}\} \quad (5.3)$$

式 (5.3) のように，集合として記述できる．そこで，時間 t における軌跡の集合を次のように定義する．

$$\mathcal{L}_t = \{l_t^1, l_t^2, \dots, l_t^{n_t}\} \quad (5.4)$$

この集合 $\mathcal{L}_t$ の要素 l_t^i を全て求めることが，車両特徴点の軌跡を求めることに相当する．このように，各特徴点に関して過去の軌跡を求める理由は，後のクラスタリング結果を時間方向に安定化させるためである．

ここで， l_t^i を求めるにあたり， $\mathbf{p}_t^i$ が $\mathcal{L}_{t-1}$ のどの軌跡から延びてきたものなのかを考える．もし $\mathbf{p}_t^i$ が l_{t-1}^j を延ばした結果得られたものであるならば， $l_t^i = \{\mathbf{p}_t^i\} \cup l_{t-1}^j$ とすればよい．もし， $\mathbf{p}_t^i$ が時間 t で初めて現れた特徴点であるならば， $l_t^i = \{\mathbf{p}_t^i\}$ となる．

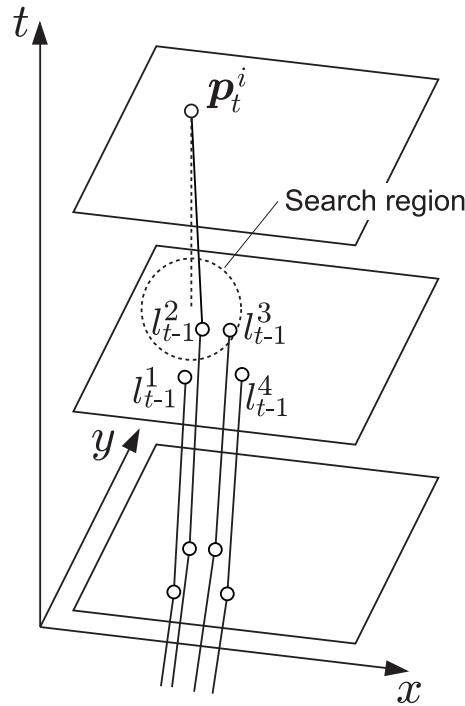


図 5.2 軌跡の延長

そこで、差の二乗和 (SSD) を基準としたテンプレートマッチングにより、 $\mathbf{p}_t^i$ に対応すべき $l_{t-1}^j \in \mathcal{L}_{t-1}$ を選出する (図 5.2). まず、 $\mathbf{p}_t^i$ の近傍に半径 $T_{\text{range}}^{\text{fp}}$ の探索領域を設け、この領域内に存在する軌跡を対応付けの候補 $l_{t-1}^{j_k}$ ($k = 1, 2, \dots$) とする. 次に、各候補 $l_{t-1}^{j_k}$ の時間 $t-1$ での座標 $\mathbf{p}_{t-1}^{j_k}$ に対し、SSD を式 (5.5) で求める.

$$\text{ssd}(\mathbf{p}_t^i, \mathbf{p}_{t-1}^{j_k}) = \frac{1}{|\mathcal{W}|} \sum_{\mathbf{d} \in \mathcal{W}} (I_t(\mathbf{p}_t^i + \mathbf{d}) - I_{t-1}(\mathbf{p}_{t-1}^{j_k} + \mathbf{d}))^2 \quad (5.5)$$

但し、

$$\mathcal{W} = \{ (u, v)^\top \mid |u|, |v| < T_{\text{window}}^{\text{fp}}/2 \} \quad (5.6)$$

とする. $I_t(\mathbf{p})$ は時間 t での入力画像である. この SSD の値が最も小さくなり、かつしきい値 $T_{\text{ssd}}^{\text{fp}}$ 以下となる候補 $l_{t-1}^{j_k}$ を $\mathbf{p}_t^i$ に対応すべき軌跡として選択する.

後に述べる類似度算出 (5.4.1 節), 辺重みの修正 (5.4.3 節), および分割したグラフの対応付け (5.4.3 節) の各処理では、時間 t における特徴点 $\mathbf{p}_t^i$ が接続している過去における軌跡 l_{t-1}^j が持つ情報を用いる. そこで、 $\mathbf{p}_t^i$ に対して選出した軌跡が l_{t-1}^j であるとき、この選択関係を関数 $c_t(i) = j$ で記録し、この関数 $c_t(i)$ を用いて各処理を定式化する. $c_t(i)$ を導入することで、例えば $\mathbf{p}_t^i$ が接続している過去における軌跡を $l_{t-1}^{c_t(i)}$ と表せるようになり、記述が簡潔になる.

5.4 軌跡群のクラスタリングによる追跡

前節で得た軌跡群 $\mathcal{L}_t$ は、画像の x, y 軸に時間 t 軸を加えた 3 次元空間内で折れ線として現れる. 時間 t における車両の位置を決定するためには、軌跡群 $\mathcal{L}_t$ をクラスタリングすればよい. 本手法では、このクラスタリング手法としてグラフ分割 [46] を用いる. $\mathcal{L}_t$ における任意の 2 つの軌跡 l_t^i, l_t^j の間に類似度を定義し、軌跡を頂点、類似度を辺重みとした完全グラフを分割することで $\mathcal{L}_t$ をクラスタリングし、車両の位置を決定する.

5.4.1 軌跡間の類似度

まず、2 つの軌跡 $l_t^i, l_t^j \in \mathcal{L}_t$ の類似度 S_t^{ij} を定義する. 我々は既に 4 章での議論において、軌跡間の類似度としてエッジの強度が有効であることを示した. だが、4 章では特徴点間のエッジ強度を局所探索法による最適化を施すことで精細に求めており、類似度の算出に時間がかかるため、実用上は問題がある. そこで、ブロック化されたエッジ画像を用いることで、エッジによる類似度をより高速に求められる方法を提案する. また、軌跡間の距離による類似度を定義し、これら二つの尺度を合わせた値を軌跡間の類似度として定義することで、クラスタリングのさらなる安定化を図る.

類似度 S_t^{ij} は式 (5.7) により、前フレームでの類似度 $S_{t-1}^{c_t(i)c_t(j)}$ を用いて帰納的に定義される。

$$S_t^{ij} = T_{\text{weight}}^{\text{cut}} ED + (1 - T_{\text{weight}}^{\text{cut}}) S_{t-1}^{c_t(i)c_t(j)} \quad (5.7)$$

E は入力画像 $I_t(\mathbf{p})$ 上の 2 点 $\mathbf{p}_t^i, \mathbf{p}_t^j$ 間のエッジの強さであり、エッジが強いほど 1 に近い値をとる。 D は $\mathbf{p}_t^i, \mathbf{p}_t^j$ 間の距離の近さを示す尺度であり、2 点が近いほど 1 に近い値をとる。 $T_{\text{weight}}^{\text{cut}}$ は重み係数である。つまり、前フレームでの類似度 $S_{t-1}^{c_t(i)c_t(j)}$ を、現在フレームの情報 E, D を用いて更新していくという方法で S_t^{ij} を得る。以後、これら E, D をそれぞれエッジによる類似度、距離による類似度と呼ぶ。

なお、 $|l_t^i| = 1$ または $|l_t^j| = 1$ であるような場合は $S_{t-1}^{c_t(i)c_t(j)}$ を得ることができないため、そのときは単に $S_t^{ij} = ED$ とする。

エッジによる類似度 E

2 点間のエッジ強度を得るために、次の手順でブロック化された車両エッジ画像を作成する (図 5.3(a)~(f))。まず、入力画像 $I_t(\mathbf{p})$ に Sobel フィルタをかけ、エッジ画像 $I'_t(\mathbf{p})$ を作成する。これを式 (5.8) により $T_{\text{block}}^{\text{cut}}$ 画素単位にブロック化する。

$$B_t(u, v) = \max_{0 \leq x, y \leq T_{\text{block}}^{\text{cut}} - 1} I'_t(T_{\text{block}}^{\text{cut}}u + x, T_{\text{block}}^{\text{cut}}v + y) \quad (5.8)$$

つまり、各ブロック内での最大値を当該ブロックでの値として採用する。 $B_t(u, v)$ は v 行 u 列目のブロックに対して与えられたエッジの強さを示す。 $B_t(u, v)$ は背景のエッジ成分を含むため、 $I_t(\mathbf{p})$ と背景画像 $J(\mathbf{p})$ を比較することで背景成分の除去を行う。

今、ブロック (u, v) が背景のエッジか否かを判別することを考える。このとき、ブロック (u, v) の周辺に $T_{\text{window}}^{\text{cut}} \times T_{\text{window}}^{\text{cut}}$ 画素の大きさを持つ窓を用意し、 $I_t(\mathbf{p})$ と $J(\mathbf{p})$ の間で正規化相関値 B_{NC} を求める。このとき、式 (5.9) により $B_t(u, v)$ を更新し、背景成分の除去された $B'_t(u, v)$ を得る。

$$B'_t(u, v) = \begin{cases} B_t(u, v) & B_{\text{NC}} < T_{\text{nc}}^{\text{cut}} \\ 0 & B_{\text{NC}} \geq T_{\text{nc}}^{\text{cut}} \end{cases} \quad (5.9)$$

$T_{\text{nc}}^{\text{cut}}$ は、5.3.1 節で述べた内容と同様の理由で、0.5 とすれば良い。

最後に、式 (5.10) によりエッジを空間方向に太める。

$$\hat{B}_t(u, v) = \max_{-1 \leq x, y \leq 1} B'_t(u + x, v + y) \quad (5.10)$$

こうして求めた $\hat{B}_t(u, v)$ を用い、2 点間のエッジの強さを評価する。

ここで、2 点 $\mathbf{p}_t^i, \mathbf{p}_t^j$ を結んだ直線上の $\hat{B}_t(u, v)$ の値を $e(s)$ で表すものとする。このとき、 $\mathbf{p}_t^i, \mathbf{p}_t^j$ 間のエッジの強さ e を

$$e = \min_s e(s) \quad (5.11)$$

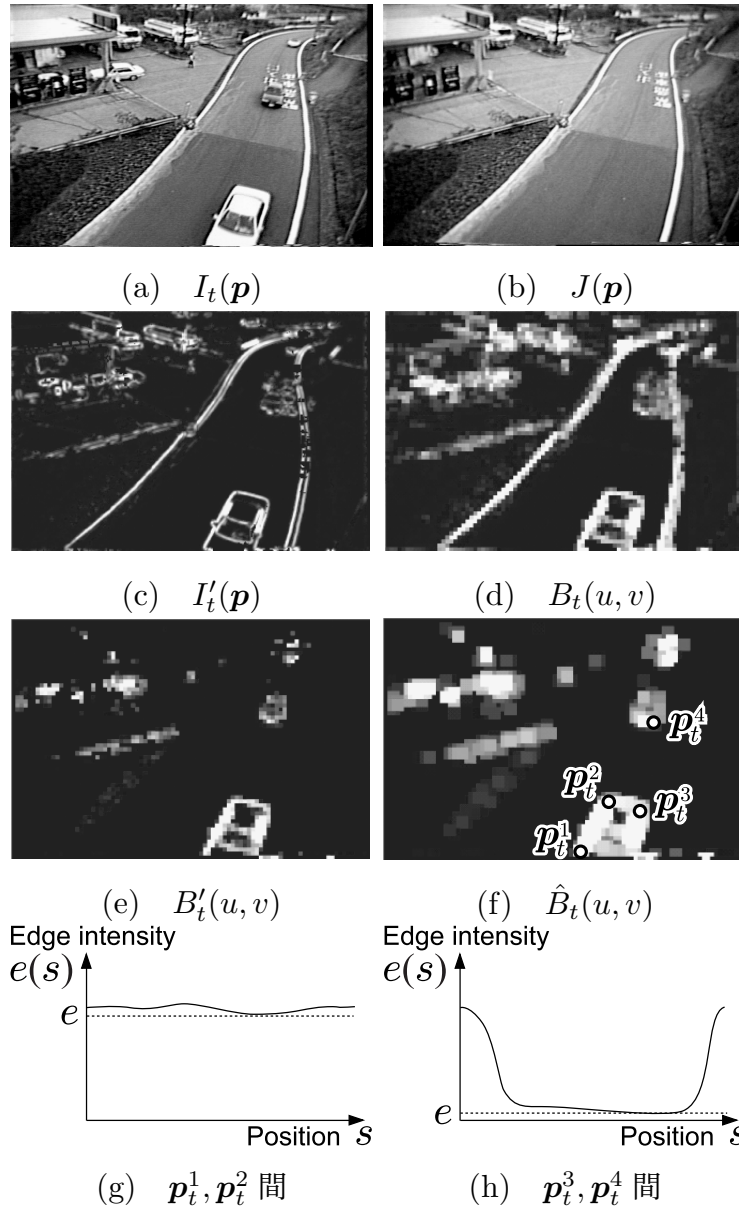


図 5.3 2 点間のエッジ強度

とする。

図 5.3(f) における p_t^1, p_t^2 間, p_t^3, p_t^4 間の $e(s)$ を示したのが図 5.3(g)(h) である。 $e(s)$ の最小値が大きくなるということは、2 点間に途切れなくエッジが存在することを意味する。その一方で、2 点間のエッジが途切れている場合は $e(s)$ の最小値が著しく小さい値をとる。

こうして求めた e を図 5.4(a) の写像で 0~1 の値に正規化する。この正規化された値を p_t^i, p_t^j 間のエッジによる類似度 E として採用する。基本的に e と E は比例するが、特に e が大きい値を取る場合、それがクラスタリングにおいて有意に働き過ぎないようにする

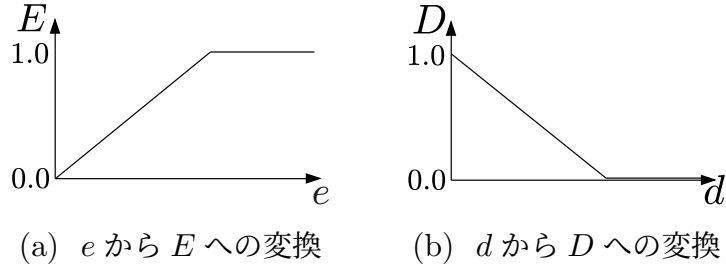


図 5.4 類似度の正規化

ために、一定値 $T_{\text{normE}}^{\text{cut}}$ を超えた場合は類似度を一律で 1 になるように設定している．なお、この変換にはシグモイド関数のような非線型変換を用いることもできるが、予備実験を行い検討したところ、線型変換でも十分な追跡性能を得られることを確認している．

距離による類似度 D

E と同様に、 $\mathbf{p}_t^i, \mathbf{p}_t^j$ 間の画像平面上での距離 $d = \|\mathbf{p}_t^i - \mathbf{p}_t^j\|$ を図 5.4(b) の写像で 0～1 の値に正規化する．この正規化された値を $\mathbf{p}_t^i, \mathbf{p}_t^j$ 間の距離による類似度 D として採用する．この写像は距離が遠くなるほど類似度 D が減少し、ある一定の距離 $T_{\text{normD}}^{\text{cut}}$ を超えると一律で 0 になるように設計する．

5.4.2 グラフの定義と分割法

グラフ $G_t = (V_t, E_t)$, $w_t(e)$ $e \in E_t$ を次のように定義する．

$$V_t = \mathcal{L}_t \quad (5.12)$$

$$E_t = \{e_t^{ij} = \langle l_t^i, l_t^j \rangle \mid l_t^i, l_t^j \in V_t, i < j\} \quad (5.13)$$

$$w_t(e_t^{ij}) = S_t^{ij} \quad (5.14)$$

$e_t^{ij} = \langle l_t^i, l_t^j \rangle$ は l_t^i, l_t^j 間の辺である． G_t は全ての頂点間に辺を持ち、その重みとして類似度 S_t^{ij} が与えられている．完全グラフ G_t において、同一の車両から検出された軌跡同士は強い辺重みで連結し合う．各車両の位置を決定するためには G_t を分割すればよい．

グラフ分割アルゴリズムはオペレーションズ・リサーチの分野で深く議論されており、最適な分割を得るための様々な手法が提案されているが [47]、ここでは Normalized Cuts [46] を用いた．これは一般化固有値問題とグラフ分割の関連性に基づく分割手法であり、画像の領域分割に有効な方法であるとしてコンピュータビジョンの分野で注目されているものである．

Normalized Cuts は、分割する際の「評価値」と「解法」にそれぞれ工夫がある手法である．Normalized Cuts の評価値は、ノイズに起因する孤立点や小さなクラスタをカット

するような場合には悪くなり、一方で大きなクラスタをカットするときには評価値が良くなる。それゆえ、「直感的に」非常に良いクラスタを得られると言われている。また、グラフ分割において、未知の k 分割を得ようとする場合、2 分割を繰り返すことで 3 以上の分割を得るという方法がとられるが、このとき分割の停止条件として評価値に対するしきい値 $T_{\text{stop}}^{\text{cut}}$ が必要である。Normalized Cuts の評価値は頂点の数で正規化されているため、グラフの頂点数が変わっても評価値のスケールが変わらない。それゆえしきい値 $T_{\text{stop}}^{\text{cut}}$ の設定が容易であるという利点もある。Normalized Cuts はこの $T_{\text{stop}}^{\text{cut}}$ 以外に一切のしきい値を要さない*2。また、評価値を最大化（あるいは最小化）する最適解は固有値解析で近似的に得られることが示されており、 G_t の重みを示す行列が疎であれば、計算量は $o(n)$ 程度 (n は頂点数) である。以上の性質から、Normalized Cuts は本手法に適用すべき分割法として適切であると考えられる。

考案者を含め、コンピュータビジョンの分野に属する研究者のほとんどは、Normalized Cuts を画像の領域分割に適用する方向で研究を展開しているが [46, 52, 53, 54, 44]、コンピュータビジョン以外の分野においてもしばしば適用例が見られる。例えばバイオインフォマティクスの分野においては、DNA マイクロアレイから有意な情報を抽出するためのクラスタリング手法としても用いられている [55]。Normalized Cuts は一種のクラスタリング手法であり、このような多方面への応用が可能である。本章でも軌跡群をクラスタリングするための手法として Normalized Cuts を用いる。

5.4.3 グラフ分割による追跡

図 5.5 はグラフ分割による追跡の流れを示したものである。まず、前フレームまでの追跡結果を活かしてクラスタリングを安定化させるために、完全グラフ G_t の重みの修正を行う。この修正したグラフを G'_t とする。次に G'_t に対して Normalized Cuts を適用し、頂点集合 V_t を複数の部分集合 $V_t^1, V_t^2, \dots$ に分割する。なお、これらの分割 V_t^i に重複は無く、 $\bigcup_i V_t^i = V_t$ および $V_t^{i_1} \cap V_t^{i_2} = \phi$ ($i_1 \neq i_2$) を満たすものとする。これらと過去の分割 V_{t-1}^j を対応付けることで車両の追跡を行う。このとき、各分割 V_t^i に累積スコア A_t^i を付与し、累積スコアが十分高い分割を車両として判別する。

*2 疎行列の固有値を求める数値計算法は幾つかのパラメータの調整を要するが、それらは算出の精度や速度に関係するものであり、分割結果に支配的に影響を及ぼすようなものではないため、ここではこれらを見捨てる。

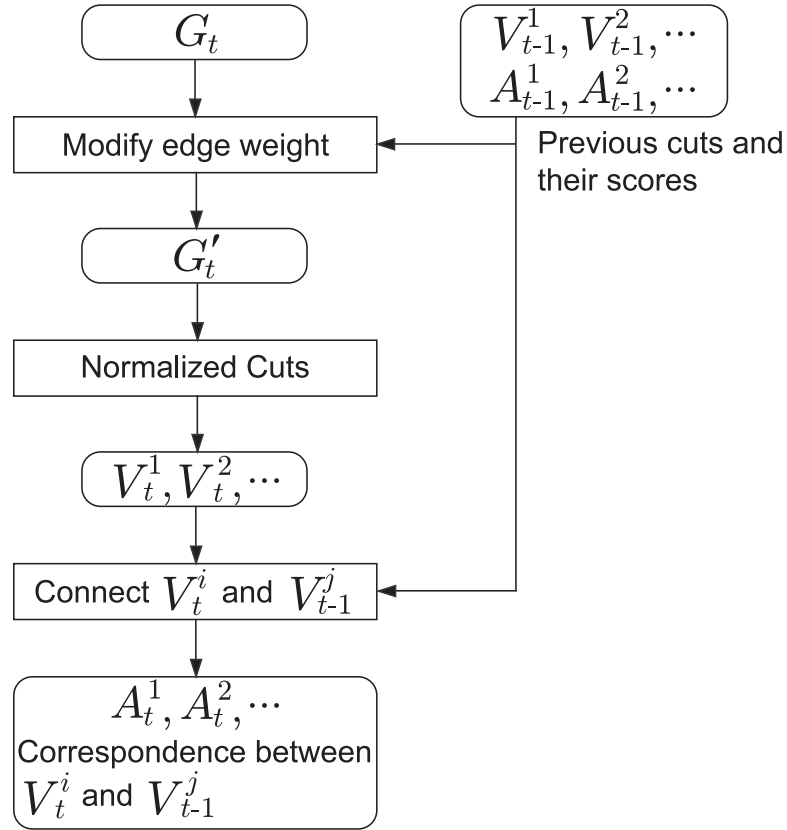


図 5.5 グラフ分割による追跡の流れ

累積スコアによる車両の判定

分割 V_t^i は、必ずしも車両の軌跡群を表していない。例えば 1~2 本の軌跡から成るような小さな分割はノイズ由来のものであると考えられる。そこで、各分割 V_t^i に累積スコア A_t^i を付与し、このスコアが一定値を超えたときに車両として判別する。 V_t^i に対して V_{t-1}^j が対応付けられているとき、 V_t^i の累積スコア A_t^i を式 (5.15) で更新する。

$$A_t^i = \begin{cases} A_{t-1}^j + 1 & |V_t^i| \geq T_{\text{num}}^{\text{cut}} \\ A_{t-1}^j & |V_t^i| < T_{\text{num}}^{\text{cut}} \end{cases} \quad (5.15)$$

つまり、分割の頂点数が十分多い場合は前フレームでの累積スコア A_{t-1}^j に 1 を加算する。このスコアが一定値 $T_{\text{score}}^{\text{cut}}$ 以上になったときに V_t^i を車両として判別する。 V_t^i に対応すべき V_{t-1}^j が見つからなかった場合は $A_t^i = 0$ とする。

重みの修正

例えば軌跡 $l_t^i, l_t^j \in V_t$ が時間 $t-1$ において同一の車両の軌跡であった場合、これらは時間 t においても同一の分割に属する可能性が非常に高いものと考えられる。そこで、辺

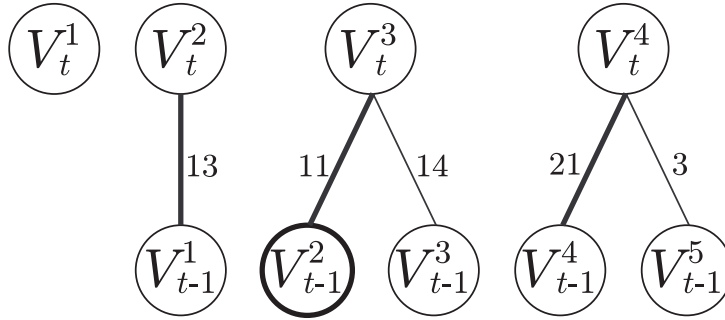


図 5.6 V_t^i と V_{t-1}^j の対応付け — 図中太線の丸で囲った分割は既に車両として判別されているものを示す．太い直線は最終的に選ばれた対応関係を表す．直線に付随する数字は各分割が共有する軌跡数．

重み $w_t(e_t^{ij})$ に 1 よりも大きい定数係数 $T_{\text{modify}}^{\text{cut}}$ を掛け、重みを強化する．

一方、軌跡 $l_t^i, l_t^j \in V_t$ が時間 $t-1$ において異なる車両の軌跡であった場合、これらは時間 t においても異なる分割に属する可能性が非常に高いものと考えられるため、辺重み $w_t(e_t^{ij})$ をゼロに設定しなおす．これをまとめると式 (5.16) で表すことができる．

$$w'_t(e_t^{ij}) = \begin{cases} T_{\text{modify}}^{\text{cut}} w_t(e_t^{ij}) & \text{if } l_{t-1}^{c_t(i)}, l_{t-1}^{c_t(j)} \in V_{t-1}^k \\ & \text{and } A_{t-1}^k \geq T_{\text{score}}^{\text{cut}} \\ 0 & \text{if } l_{t-1}^{c_t(i)} \in V_{t-1}^{k_1} \\ & \text{and } l_{t-1}^{c_t(j)} \in V_{t-1}^{k_2} \\ & \text{and } A_{t-1}^{k_1} \geq T_{\text{score}}^{\text{cut}} \\ & \text{and } A_{t-1}^{k_2} \geq T_{\text{score}}^{\text{cut}} \\ & \text{and } k_1 \neq k_2 \\ w_t(e_t^{ij}) & \text{otherwise} \end{cases} \quad (5.16)$$

$w'_t(e)$ は修正した重みである． $w_t(e)$ のかわりに、 $w'_t(e)$ を重み関数として持つ完全グラフを G'_t とし、これに Normalized Cuts を適用することで分割 G_t^i を得る．

この重み修正処理は、複数車両が近接したときに効果を発揮する．このような場合、異なる車両間でもやや強い重みを持った辺が現れることがある．だが、前フレームでの分割結果に基づき、過去の分割結果が保存されるように重みが修正されるため、結果として正常な分割が得られる．

V_t^i と V_{t-1}^j の対応付け

まず、 V_{t-1}^j に対して対応すべき V_t^i を探すことを考える．時間 t での軌跡 l_t^k は、前のフレームでは $l_{t-1}^{c_t(k)}$ である．そこで、 V_t^i と V_{t-1}^j の両方に含まれる軌跡、すなわち $l_t^k \in V_t^i$ かつ $l_{t-1}^{c_t(k)} \in V_{t-1}^j$ となるような軌跡を数え、この数が最大となる V_t^i を選択する．

すると、 V_t^i から見て 1 対多の対応となる (図 5.6 : V_t^i, V_{t-1}^j 間の線)．次にこれを 1 対

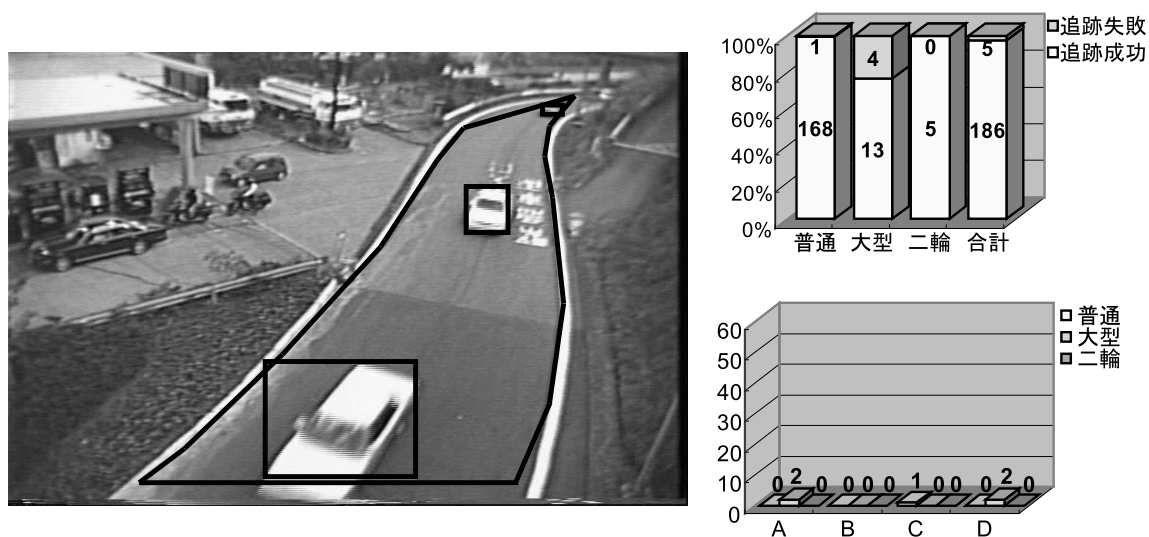
1 の対応に絞ることを考える．基本的には，共有する軌跡数が最も大きくなるものを選択することで1 つに絞るが，すでに車両として判別されているものがある場合，それを優先して対応付ける (図 5.6 : V_t^i, V_{t-1}^j 間の太線)．

5.5 実験

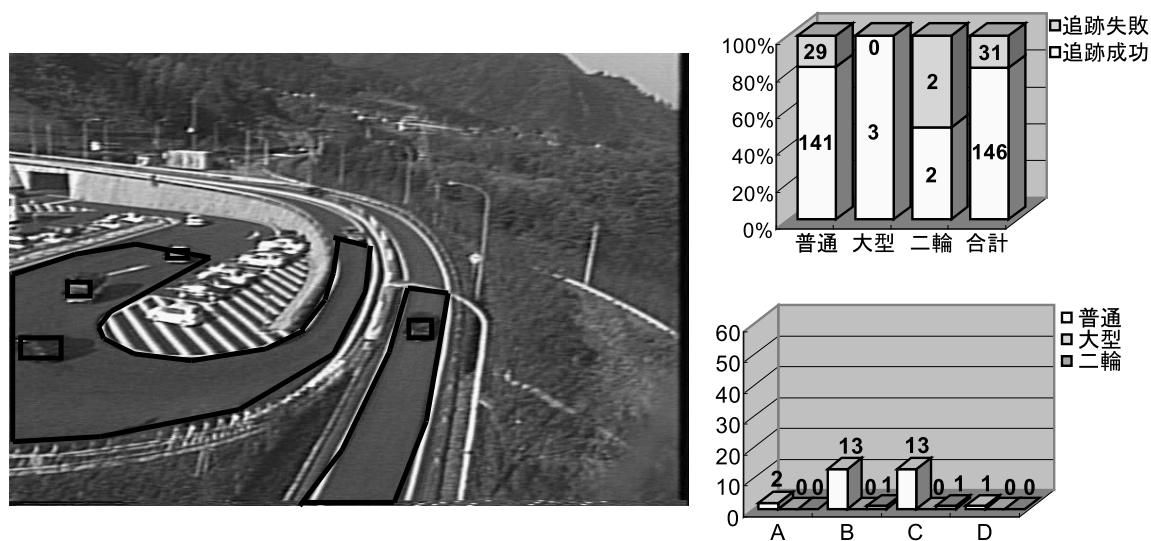
5.5.1 対象シーン

空間解像度 360×240 ，時間解像度 30fps の 6 種類の動画像 (図 5.7, 図 5.8, 図 5.9 Scene1 ~6) を用いて提案手法の有効性を評価した．それぞれの動画像において，交通流量の少ない 1000 フレーム程度の部分を選択し，これを時間方向に平均化して合成した画像を背景画像として用いた．

Scene1~4 は昼間に撮影したものであり，Scene5,6 は夜間に撮影したものである．Scene1 は最も好条件で，車両の停滞や重なりが非常に少ない．Scene2 は画像上での車両の大きさが，6 シーン中最も小さい．Scene3 は信号の手前 5 車線を同時に撮像したものであり，1 分前後の停滞が 3 回起こる．Scene4 は奥から手前に向かって車両が走行するシーンであり，近づくにつれて車両が非常に大きくなる．Scene5 は夜の事例であり，小雨が降っている．車両を後方から捉えており，遠方では車両同士が著しく近接する．Scene6 もまた夜に小雨が降っている事例である．信号の手前の停止線付近を撮影しているため，車両が頻繁に停車する．このとき，右側 3 車線ではヘッドライトを点灯させた車両が密集するため，条件としては非常に悪い画像であると言える．



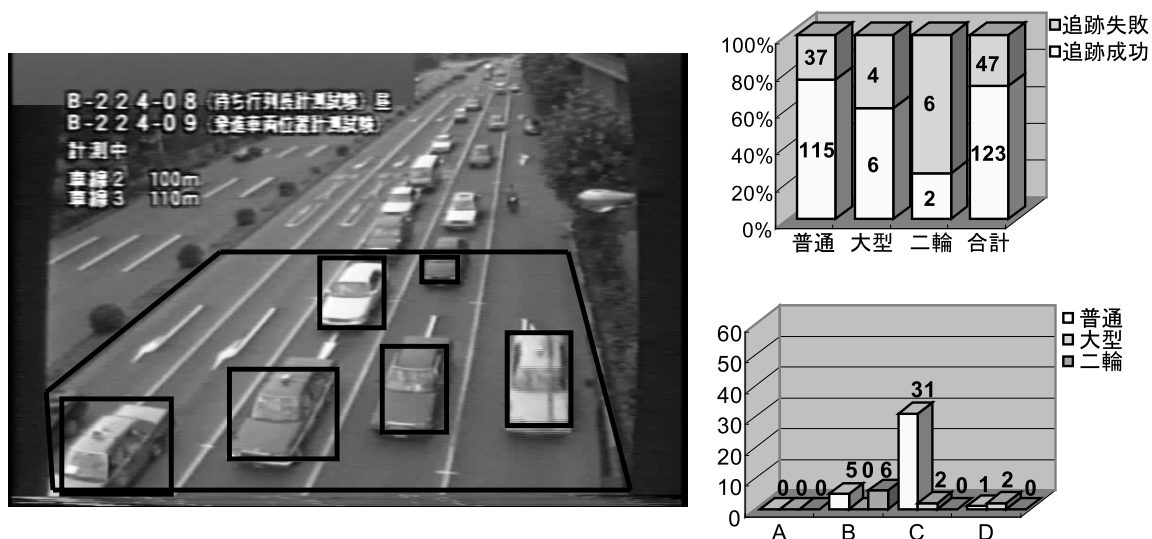
Scene1 — 15 分 00 秒 追跡成功率：97.38% 平均実行時間：56.52 msec/frame



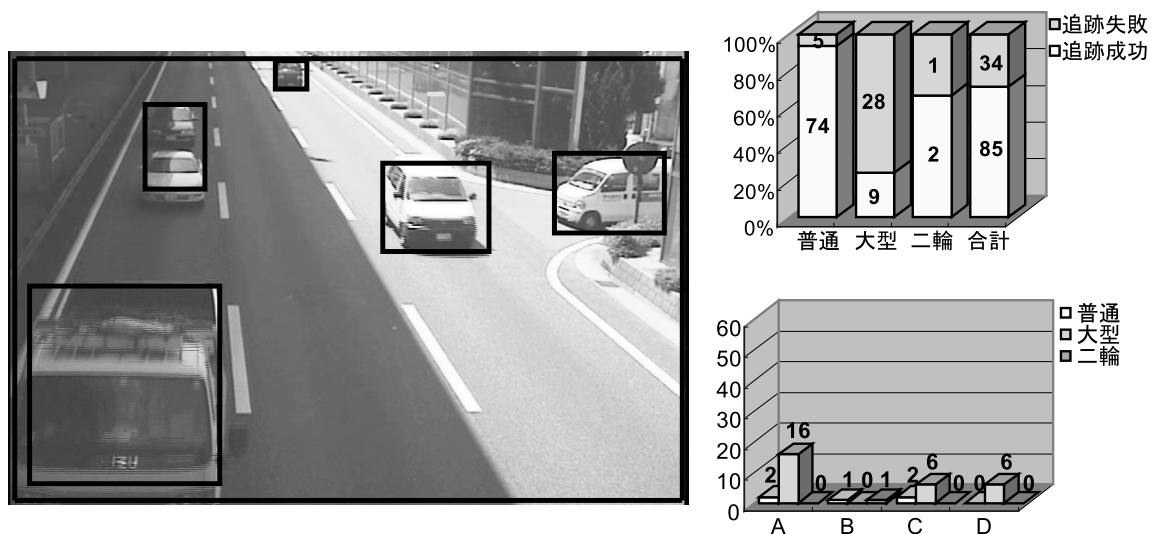
Scene2 — 15 分 00 秒 追跡成功率：82.49% 平均実行時間：43.45 msec/frame

- (A) 1 台を複数の車両として追跡してしまった場合
- (B) 追跡は行われたが、累積スコアが十分高くなりず車両として認識されなかった場合、あるいはそもそも十分な数の特徴点を抽出できず、車両を見落とした場合
- (C) 複数の車両を 1 台として追跡してしまった場合
- (D) 外接方形の位置や大きさが実際と著しく異なるような場合

図 5.7 追跡結果 (昼 1)



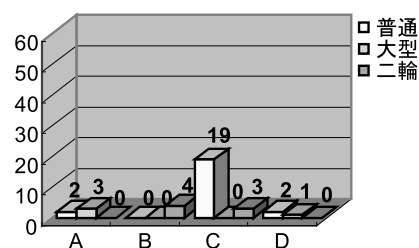
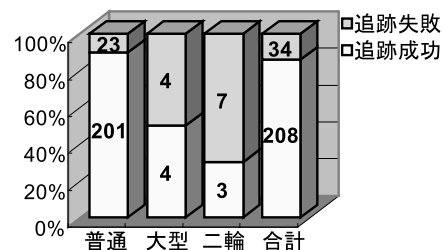
Scene3 — 7 分 19 秒 追跡成功率：72.35% 平均実行時間：90.30 msec/frame



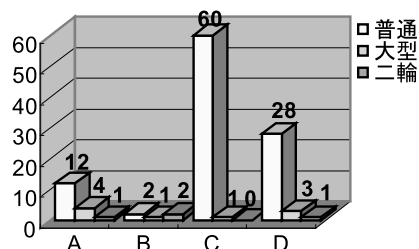
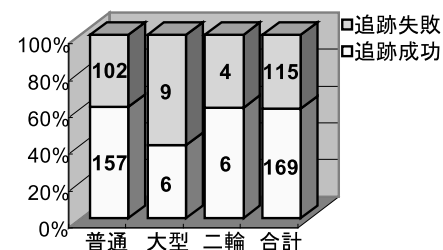
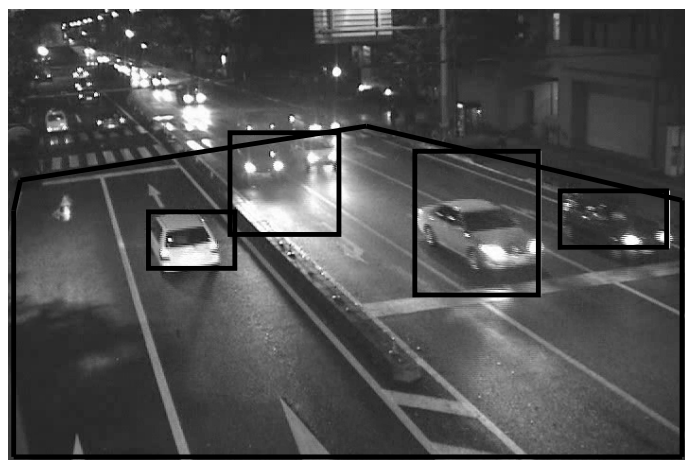
Scene4 — 2 分 55 秒 追跡成功率：71.43% 平均実行時間：142.65 msec/frame

- (A) 1 台を複数の車両として追跡してしまった場合
- (B) 追跡は行われたが、累積スコアが十分高くなりず車両として認識されなかった場合、あるいはそもそも十分な数の特徴点を抽出できず、車両を見落とした場合
- (C) 複数の車両を 1 台として追跡してしまった場合
- (D) 外接方形の位置や大きさが実際と著しく異なるような場合

図 5.8 追跡結果 (昼 2)



Scene5 — 15 分 00 秒 追跡成功率：85.95% 平均実行時間：44.00 msec/frame



Scene6 — 15 分 00 秒 追跡成功率：59.51% 平均実行時間：64.20 msec/frame

- (A) 1 台を複数の車両として追跡してしまった場合
- (B) 追跡は行われたが、累積スコアが十分高くなりず車両として認識されなかった場合、あるいはそもそも十分な数の特徴点を抽出できず、車両を見落とした場合
- (C) 複数の車両を 1 台として追跡してしまった場合
- (D) 外接方形の位置や大きさが実際と著しく異なるような場合

図 5.9 追跡結果 (夜)

5.5.2 しきい値の決定

5.3 節, 5.4 節で説明した手法では 15 個のしきい値が用いられている. これらを全てのシーンに対して適切な値にするために予備実験を行い, 以下の指針に従い決定した. また, 決定したしきい値を表 5.1 にまとめた.

表 5.1 しきい値

T_{λ}^{fp}	0.03	$T_{\text{weight}}^{\text{cut}}$	0.9	$T_{\text{normD}}^{\text{cut}}$	75
$T_{\text{window}}^{\text{fp}}$	5	$T_{\text{block}}^{\text{cut}}$	4	$T_{\text{stop}}^{\text{cut}}$	0.3
$T_{\text{nc}}^{\text{fp}}$	0.5	$T_{\text{window}}^{\text{cut}}$	12	$T_{\text{num}}^{\text{cut}}$	5
$T_{\text{range}}^{\text{fp}}$	8	$T_{\text{nc}}^{\text{cut}}$	0.5	$T_{\text{score}}^{\text{cut}}$	10
$T_{\text{ssd}}^{\text{fp}}$	10000	$T_{\text{normE}}^{\text{cut}}$	50	$T_{\text{modify}}^{\text{cut}}$	5.0

$T_{\lambda}^{\text{fp}}, T_{\text{nc}}^{\text{fp}}$ は車両特徴点抽出に必要なしきい値であり, 5.3.1 節で述べた理由から, T_{λ}^{fp} は感度が十分高くなるように, また $T_{\text{nc}}^{\text{fp}}$ は 0.5 に設定した. $T_{\text{window}}^{\text{fp}}, T_{\text{block}}^{\text{cut}}, T_{\text{window}}^{\text{cut}}$ は窓やブロックサイズを決めるしきい値であり, これらは撮影対象よりもむしろ画像の空間解像度に依存すると考えられる. そこで, 空間解像度 360×240 に適切な値を選んだ. $T_{\text{range}}^{\text{fp}}$ は軌跡を求める際の探索範囲であるから, 車両が大きく撮影されている場合は探索範囲が広い方が好ましい. そこで, 車両が大きく映っている画像に合わせて調整した. $T_{\text{ssd}}^{\text{fp}}$ は高めに設定し, 対応付けられるべき過去の軌跡が見つかり易いようにした. $T_{\text{weight}}^{\text{cut}}$ は, 過去の類似度と現在の類似度との比率を決めるものであるが, ここでは現在の類似度を重視し, 大きめの値に設定した. $T_{\text{nc}}^{\text{cut}}$ は $T_{\text{nc}}^{\text{fp}}$ と同様の理由で 0.5 とした. $T_{\text{normE}}^{\text{cut}}$ はエッジの強さに関わるしきい値であるため, 夜の画像でもエッジを捉えられるよう, 低めに設定した. $T_{\text{normD}}^{\text{cut}}$ は, $T_{\text{range}}^{\text{fp}}$ と同様の理由により, 車両が大きく映っている画像に合わせて調整した. $T_{\text{num}}^{\text{cut}}$ は車両として認識されるために最低限必要な特徴点の数であり, 車両が小さく映っているような場合でも動作するように調整した. $T_{\text{score}}^{\text{cut}}$ は累積スコアの定義から, 移動物体が現れてから車両と認識されるまでに必要な最短フレーム数であると言える. ここでは 1 秒以下で認識することを目指し, $T_{\text{score}}^{\text{cut}} = 10$ とした. $T_{\text{modify}}^{\text{cut}}$ は撮影対象に依存する値ではない. そこで, 経験的に良いと思われる値を用いた.

5.5.3 実験結果

図 5.10 は、横軸を時間、縦軸を車両の中心の y 座標として、正解となる軌跡、本章で提案した手法の結果、および 4 章の手法による車両認識の結果をプロットしたものである。ここでは、Scene1 において連続して通過する 4 台を対象とした。4 章における車両認識手法は局所時間に対して適用するものであるため、20 フレームごとに手法を適用し、最後のフレームでの結果を代表としてプロットしてある。

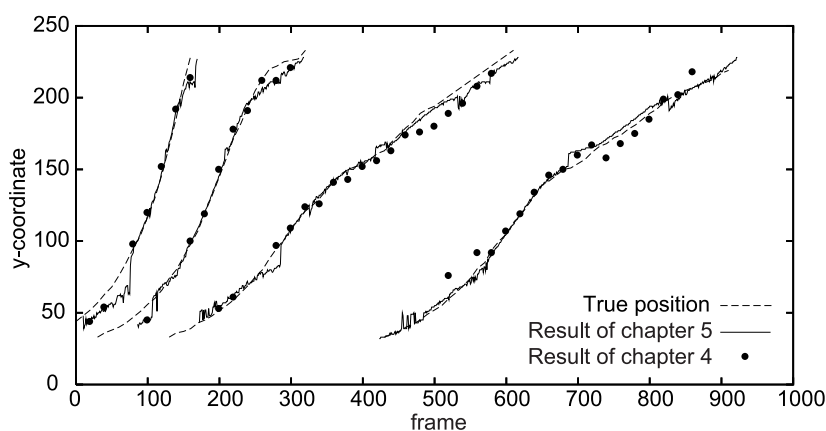


図 5.10 時間推移に伴う車両位置の変化

図に示されるように、本章による結果は真の解に非常に近い曲線となっており、車両が進入してから出てゆくまでの間、一連の車両の軌跡を得られていることが分かる。一方、4 章による手法は真の解に近い分布を示しているものの、それぞれの局所時間での結果が時間方向に接続されておらず、離散的なものとなる。本章で提案した手法ではこの点について改良されていることが分かる。

図 5.7, 図 5.8, 図 5.9 の左は入力画像とシステムが出力した車両の外接方形を示したものであり、監視領域を黒枠で囲っている。右上は追跡精度を示したグラフであり、普通車、大型車、二輪車の三種別々に集計した結果も示している。右下は追跡に失敗したときの要因を 4 種に分けて集計したものである。A~D はそれぞれ次の通りである。

- (A) 1 台を複数の車両として追跡してしまった場合
- (B) 追跡は行われたが、累積スコアが十分高くなり車両として認識されなかった場合、あるいはそもそも十分な数の特徴点を抽出できず、車両を見落とした場合
- (C) 複数の車両を 1 台として追跡してしまった場合
- (D) 外接方形の位置や大きさが実際と著しく異なるような場合

通過した車両数のうち、追跡に成功した数の割合を追跡成功率であると定義し、図 5.7, 図 5.8, 図 5.9 のシーン毎に記述した。本システムの開発環境は Microsoft Visual C++ 6.0 であり、Normalized Cuts で必要となる固有ベクトルの算出には MATLAB7.0.1 の数値計算エンジンを用いた。計測に利用した計算機の CPU は Pentium4 3.20GHz, メモリは 1GByte である。この環境で計測したときの 1 フレームあたりの実行速度を図 5.7, 図 5.8, 図 5.9 のシーン毎に記述した。

5.6 実験結果の検討

5.6.1 追跡精度に関する検討

Scene6 を除いた 5 つのシーンで 7 割以上の追跡に成功し、特に Scene1 では 95% 以上の高い成功率を達成できた。全て同一のしきい値で計測できたことを考えると、これらは非常に有意な結果であると言える。Scene6 は特に条件が悪い場合を検証するために実験対象として選んだシーンであるが、このシーンでは右側 3 車線においてヘッドライトを点灯させた車両が信号待ちにより密集するため、他のシーンに比べ著しく追跡精度が低下した。なお、左側 2 車線の追跡精度が 78.6% であったのに対し、右側 3 車線では 44.3% であった。Scene5 や Scene6 の左側 2 車線ではヘッドライトやテールランプにより路面が明度変化するものの、追跡結果は良好であった。Scene6 の右側 3 車線のような映像で追跡性能が低下するのは、路面に照射しているヘッドライトが広い範囲でサチュレーションを起こすためである。輝度が飽和した領域の境界付近では不要なエッジ成分が生じてしまう。これを避けるためには、ダイナミックレンジの広いカメラを用いるか、あるいはレンズにフィルタをかけることでヘッドライトの影響を低減させるなど、撮像系側で対処する必要があると考えている。いずれの方法でも、飽和領域付近の不要な微分成分を排除できると期待される。

5.6.2 特徴点の対応付けについて

本手法では、連続する 2 フレームにおける特徴点の対応付けの方法として、SSD によるテンプレートマッチングを用いた。これは周囲のテクスチャが似ている点同士を対応付けるというものであるから、理想的には毎フレーム同じ部位から特徴点を検出できていることが望ましい。なぜなら、検出部位がずれると対応付けに失敗する可能性が高くなるからである。また車の境界付近においては、車両が動くテクスチャがずれるため、ここでも SSD による対応付けが失敗する可能性が高い。特に、路面に道路標示がある場合などにはこれが顕著である。

だが、これらに起因する局所的なエラーは、後段のグラフ分割処理ですべて修正される。なぜなら、Normalized Cuts はグラフの大局的な構造を記述する評価値をもとにして分割を行うため、局所的なエラーには左右されにくいからである。このことは、類似度の算出についてもいえる。類似度の算出や SSD による特徴点の対応付けの結果には、ある程度エラーを含んでいても良い。異なる車両の特徴点同士が高い類似度を持ったとしても、またある特徴点が次のフレームにおいて異なる車両の特徴点に対応付けられてしまったとしても、その局所的エラーが全体から見れば些細なものであれば、分割は正常に行われる。グラフ分割に帰着する利点は正にここにある。

5.6.3 グラフの辺重み修正による効果

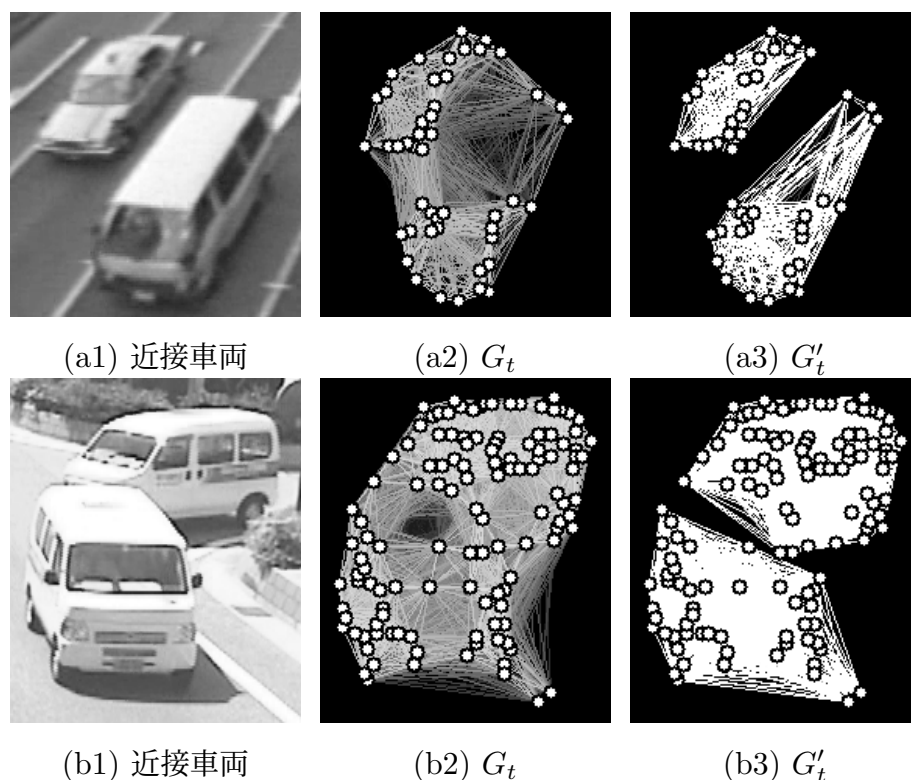


図 5.11 重みの修正による効果

図 5.11(a1)(a2)(a3) は 2 車両が近接した時の完全グラフ G_t 及び G'_t を示したものである。図中において、重みが大きな辺は明るくなるように示した。これらを比較すると、 G_t に比べ G'_t は 2 車両間の辺重みが弱められ、また同一の車両内では辺重みが互いに強められていることが分かる。また、図 5.11(b1)(b2)(b3) は一方の車両が他方を隠している場合であるが、このケースでは重なりがそれほど大きな面積ではなく、かつ短期間であったため、(b3) のように各車両同士の境界が明瞭に分かる G'_t が得られた。これにより、それ

それを正確に追跡することができた。

図 5.11 における 2 つのケースでは近接する車両を分離することができたが、車両の重なり合いが深く、かつ長時間その状態が継続するような場合は 2 車両を分離することができなかった。本章では過去の分割の情報に基づいてグラフの辺重みを修正したが、この方法では、過去において異なる車両であると認識されていた点群が、現在においても異なる車両に分類されることが理論的に保証されるわけではない。6 章で述べる内容は、この点をさらに改良するものである。

5.6.4 追跡エラーの要因

要因 (A) は、画像中で車両が大きく撮影されている場合に起こりやすく、特に大型車の追跡中に頻繁に起こった。これは、同一の車両内における点間の類似度が十分に高くないことが原因である。逆に要因 (B) では、画像上での車両の面積が非常に小さい場合に起こり易かった。これは、車両から十分な数の特徴点が検出されず、式 (5.15) の評価で累積スコアが十分に加算されなかったためである。信号待ちで車両が密集する場合、あるいは車両同士の重なりが深い場合などでは、要因 (C) によるエラーが多く見られた。要因 (D) は要因 (A) と同じく画像中で車両が大きく撮影されている場合に起こりやすいようである。以上の傾向が、図 5.7, 図 5.8, 図 5.9 右下のグラフに表れている。

5.6.5 本手法の適用可能な条件について

以上の実験を総括すると、「(a) 追跡対象から十分な数の特徴点を得ることができる」、「(b) 追跡対象がエッジ成分に富み、検出された特徴点同士がエッジで十分に結びつく」、「(c) 追跡対象が、他の追跡対象によって完全に隠されることが無い」という 3 点が、本手法がうまく動作するための主たる必要条件であることが分かる。また、これら 3 つの条件を満たす範囲内で対象を撮影することが、カメラの設置位置に関する制約条件であるとも言える。

条件 (a) は画像平面上における追跡対象の面積が十分大きければ満たされる。従って、大型車・普通車・二輪車の 3 種全てにおいて追跡を成功させたいのであれば、これらの中で最も小さい二輪車が、最低でも 30×30 画素程度の大きさに撮像されていれば良い。カメラから対象物までの距離は、条件 (a) を満たす範囲内にする必要がある。

大型車・普通車・二輪車は全て人工物であり、明度分布が起伏に富む。すなわち、条件 (b) はこれらの追跡対象において基本的には満たされている。だが、あまり接写しすぎると画像上の特徴点が疎に分布するようになり、条件 (b) が成立しにくくなる傾向がある。この問題は、特に大型車で起こりやすい。このようなカメラの設置位置は適切ではなく、

カメラと追跡対象との距離を広げることが望ましい。また、入力画像を低解像度化するという対処法も考えられる。

条件 (c) は、路面とカメラの光軸の成す角度に関係している。例えばカメラを設置した位置が低く、角度が浅すぎる場合、条件 (c) を満たさないため好ましくない。特に二輪車は小さく、大型車には隠されてしまいやすい。本手法は部分的なオクルージョンには対処できることから、二輪車が完全に隠れてしまわない程度に角度を大きくとればよい。なお、路面とカメラの光軸が直角に交わるとき、条件 (c) は最も良好に満たされる。

5.6.6 実行速度

30fps の動画像をリアルタイムで処理するためには、1 フレームを 33.3msec 以下で処理できれば良い。Scene1,2,5,6 では 43~65msec という結果を得ており、リアルタイムにはほぼ近い処理速度を達成できた。一方、Scene4 では 142.65msec 程度の時間を要した。図 5.12 は Scene2,4 の実行速度を処理ごとにまとめたものである。これを見ると、グラフ分割による追跡処理の部分で差が出ていることがわかる。Scene4 のように、車両が大きく撮像されている映像では多くの特徴点が検出されるため、グラフの頂点数が増大する。Normalized Cuts の計算コストはグラフの頂点数に依存するため、この部分がボトルネックとなる。

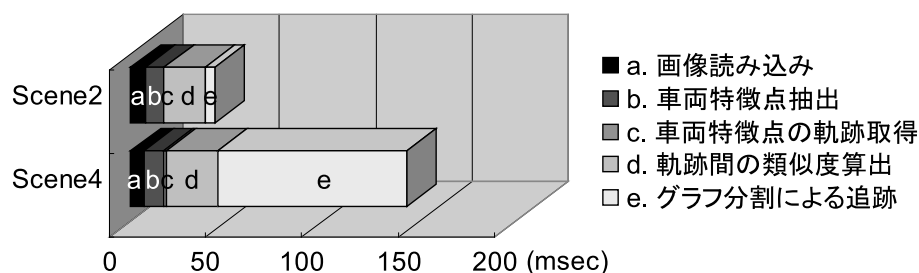


図 5.12 実行速度

だが、軌跡が正確に求められているのであれば、グラフの分割は数フレーム程度省略しても追跡精度には影響しないと考えられるため、グラフ分割処理を数フレームおきに飛ばして処理することで、大幅な高速化を望めるものと思われる。

5.7 総括

本章では、異なる時間帯、道路状況に対しても、システムのしきい値を調整することなく適用できる画像処理による車両追跡法を提案した。交通流計測は多くの地点で長時間行

われるべきものであるため、本手法の目指す方向性は、交通流量の把握、事故検知を目的とした道路交通流監視システムの実現へと繋がってゆくと考えられる。

本章で提案した手法はオクルージョン耐性という点で問題を残している。すなわち、複数の車両が近接し、深く重なり合ったまま長時間その状態を維持するような場合は複数車両の分離に失敗するという問題がある。オクルージョンの問題は、過去のフレームにおける分割結果を現在フレームに反映させることで解決できると考えられる。そこで本章ではその手段としてグラフの辺重みを修正するというアプローチを採ったが、実験の結果から、これだけでは不十分であるという結果が導かれた。そこで、次章ではグラフを分割するアルゴリズムを工夫し、過去のフレームにおける分割結果を現在フレームに伝播させる方法についてより詳細に検討し、オクルージョンに強い車両計数手法を実現する。

第 6 章

重交通流における車両計数

6.1 概要

本章では、重交通流における車両計数法について議論する。監視カメラが撮影対象とする道路状況は様々であり、カメラの設置位置と路面との相対位置関係によっては車両同士の重なりの問題が起こりやすくなる。これはオクルージョン問題と呼ばれ、性能低下の原因となる。本章では 5 章で述べた手法を拡張し、強力なオクルージョン耐性を持つ車両計数手法を提案する。本章で述べる手法は 3.1 節で述べた 5 つの性質全てを満たす (下記太字部分)。

1. **1 台のモノクロカメラで計測が可能**
2. **カメラの設置位置に関する制約が緩い**
3. **照明条件の変化に強い**
4. **車両の移動軌跡を取得可能**
5. **オクルージョンに強い**

4 章および 5 章での議論において、我々は 1~4 の性質を満たす車両計数手法を提案した。本章では、上記の 5 つの目的のうち 5 番めのオクルージョン耐性について焦点を当てる。オクルージョンの問題は、過去のフレームにおける分割結果を現在フレームに伝播させることで解決できると考えられる。そこで特徴点群のクラスタリングの問題を、過去における追跡情報を拘束条件としたグラフ分割問題として定式化することで、オクルージョンに頑健な追跡を実現する。

6.2 オクルージョンに強い車両計数の原理

特徴点の軌跡群をクラスタリングすることで車両の追跡を行うという基本的な枠組みは、4章および5章で述べた方法と同様である。特徴点群のクラスタリングの問題は、特徴点群を頂点 V_t とし、軌跡間の類似度を重み関数 $w_t(e)$ とした重み付き完全グラフ $G_t = (V_t, E_t)$, $w_t(e)$ $e \in E_t$ (E_t は辺, t は時間) を分割する問題として定式化できる。5章では、Normalized Cuts[46, 52] を適用することでこれを分割した。図 6.1(a1)(a2) に分割した例を示す。

Normalized Cuts は、それぞれの分割内の辺重みが大きく、かつ分割間の辺重みが小さいときに値が大きくなるような評価値 N_{cut} を定義し、 N_{cut} 値が最大になるようにグラフ G_t の頂点 V_t を K_t 個の頂点集合 $V_t^1, V_t^2 \dots V_t^{K_t}$ に分割する。Normalized Cuts は大域的な構造が結果によく反映されるクラスタリング手法であるため、仮に類似度の算出が失敗し、本来別々の車両に属すべき特徴点同士のうちの幾つかが強い辺重みで連結し合ってしまうようなことがあったとしても、そのような局所的なエラーの影響を受けにくいという利点がある。そのため、図 6.1(b1)(b2) のように、異なる車両間が強い辺重みで接続してしまった場合でも、これらを分割することができる。だが、図 6.1(c1)(c2) のように、車両が非常に深く重なり合ってしまうような場合は分割することができない。これは Normalized Cuts による分割の本質的な限界である。

そこで5章では、過去の分割の結果が現在フレームにおいてもある程度保存されるようにグラフの辺重みを修正することでこの問題に対処していた。この対策方法は、車両同士の近接がそれほど顕著でなければ十分に機能する。しかしながらこの方法は、過去において個別に認識されていた車両が、現在フレームにおいても個別に認識されることを保証するわけではない。そのため、車両同士が長時間重なり合うような場合には対処できないという問題があった。

本章では、過去において複数の車両と認識されていた車両は、以降のフレームで重なり合ったとしても、必ず個別に分離されることが保証される分割手法を提案する。多くの特徴点は過去のフレームの特徴点との対応がとれるため、前フレームにおいて対応する特徴点がどの車両に属していたのかを調べれば、当該特徴点がどの車両に属する可能性が高いのかどうかをあらかじめ知ることができる。そこで、幾つの特徴点を選択し、前フレームにおける分割結果をそのまま与え、残りの点群に対して N_{cut} 値を最大化するような組み合わせを探索する。これにより、解空間の自由度が小さくなり、前フレームにおける分割の情報が保存されるような解空間のみを探索することになるため、車両が近接してもそれらを必ず分離できるようになる。図 6.1(c3) は、図 6.1(b2) での分割結果を元にして、

事前に一部の解を与えた様子を図示したものであり、解を与えた頂点は四角形と三角形で示してある。本論文では、事前に解を与えた頂点を拘束点と呼ぶ。

従来の Normalized Cuts では一般化固有値解析を用いて Ncut 値を最大化する解を得るが、本手法では事前に一部の解を与えるため、同じ解法を用いることができない。Stella X. Yu ら [53, 54] は、一般化固有値解析を用いた解法を拘束された解空間においても適用できるように拡張を試みている。しかし、文献 [53] による方法では解空間の拘束のしかたが不十分であり、また文献 [54] では解の精度の面で問題があり、どちらも本論文の目的には適さない。そこで本手法では、組み合わせ最適化法的一种である焼きなまし法 (Simulated Annealing)[47] を用い、拘束のもとで Ncut 値を最大化する。前フレームにおける分割結果を用いれば、最適解に充分近い初期解を事前に与えることができるため、解は非常に高速に収束する。図 6.1(c4) は拘束点を用いて分割を行った結果であり、それぞれの車両を分離できていることがわかる。

以上がオクルージョン耐性を向上させるための原理である。拘束のもとでグラフ分割アルゴリズムを適用するという点が本章で新しく提案する部分である。特徴点抽出、軌跡の取得および類似度の算出については 5 章で述べた内容と同一であるため、簡潔にまとめるにとどめる (6.3, 6.4 節)。一方、6.5 節「拘束付きグラフ分割による車両追跡」はオクルージョンに対してロバストになる論拠となる部分であり、本章では特に 6.5 節の内容について詳細に述べ、実験および検討を行う。また 5 章と同様に、しきい値を示す記号を全て $T_{\text{Subscript}}^{\text{Superscript}}$ の記述で統一し、他の記号と区別する。

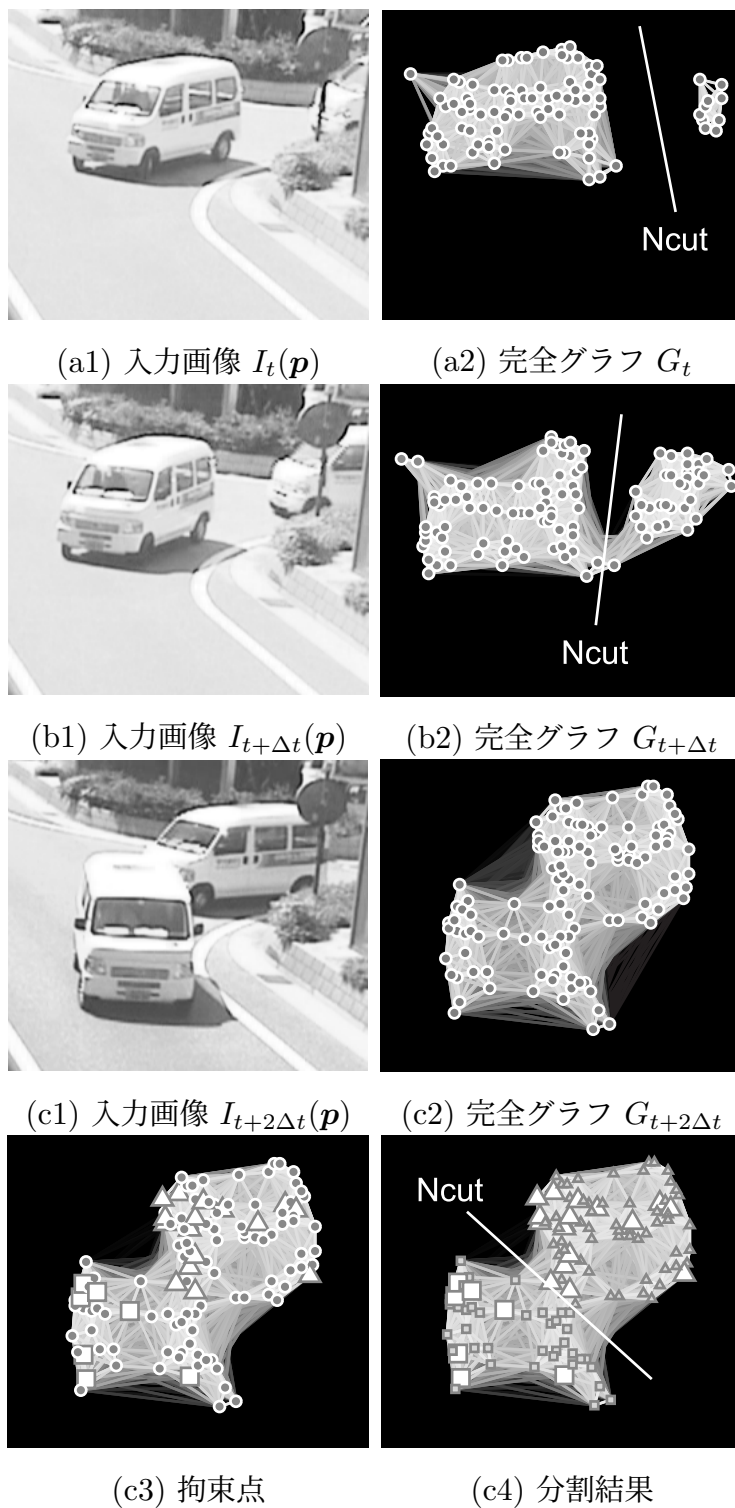


図 6.1 近接する 2 車両の追跡過程.

6.3 特徴点の軌跡群取得

6.3.1 車両特徴点抽出

微分成分に基づく方法 [49] により特徴点を抽出し、それぞれの点の周りのテクスチャと背景画像上で相当する位置のテクスチャとの間で正規化相関 [51] を計算する．背景特徴点の場合は強く相関し、移動物体の特徴点の場合は無相関となるため、相関値としきい値 T_{nc}^{fp} を比較することで背景成分を除去することができる．図 6.2 は全ての特徴点で算出した正規化相関値をヒストグラムに示したものである．相関値は正規化されているため、ヒストグラムは 0 と 1 付近を中心とした二つの正規分布が合わさったものになる．この性質は昼夜ともに変わらないため、照明条件にかかわらず $T_{nc}^{fp} = 0.5$ とすればよい．

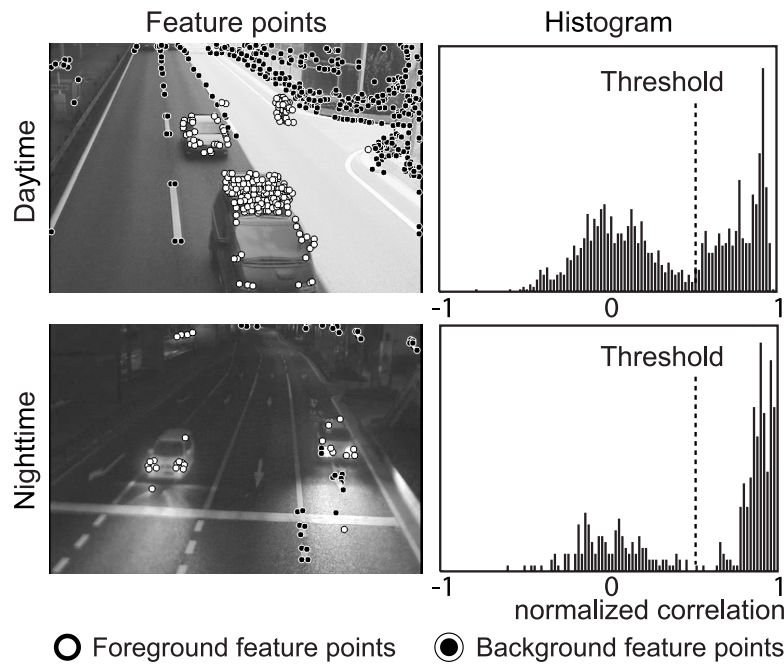


図 6.2 背景特徴点の棄却.

以降、時間 t における入力画像から検出された特徴点の数を n_t 、 i 番目の特徴点を p_t^i で表し、時間 t において検出された特徴点の集合を $\mathcal{P}_t = \{p_t^1, p_t^2, \dots, p_t^{n_t}\}$ と表記する．

6.3.2 車両特徴点の軌跡

車両特徴点の検出処理を毎フレーム適用すると、特徴点群 $\mathcal{P}_{t-2}, \mathcal{P}_{t-1}, \mathcal{P}_t, \dots$ が得られる．これらを時間方向に対応付けることで、特徴点の軌跡を得ることができる．ここで、

まず記号を整理する． $\mathbf{p}_t^i \in \mathcal{P}_t$ の軌跡は， $\mathbf{p}_t^i$ 及び時間 $t-1$ 以前における特徴点の列から成り，式 (6.1) に示すように集合として記述できる．

$$l_t^i = \{\mathbf{p}_t^i, \mathbf{p}_{t-1}^{i1}, \mathbf{p}_{t-2}^{i2}, \dots, \mathbf{p}_{t-N+1}^{iN-1}\} \quad (6.1)$$

そこで，時間 t における軌跡の集合を次のように定義する．

$$\mathcal{L}_t = \{l_t^1, l_t^2, \dots, l_t^{n_t}\} \quad (6.2)$$

この集合 $\mathcal{L}_t$ の要素 l_t^i を全て求めることが，車両特徴点の軌跡を求めることに相当する．

ここで， l_t^i を求めるにあたり， $\mathbf{p}_t^i$ が $\mathcal{L}_{t-1}$ のどの軌跡から延びてきたものなのかを考える．もし $\mathbf{p}_t^i$ が l_{t-1}^j を延ばした結果得られたものであるならば， $l_t^i = \{\mathbf{p}_t^i\} \cup l_{t-1}^j$ とすればよい．もし， $\mathbf{p}_t^i$ が時間 t で初めて現れた特徴点であるならば， $l_t^i = \{\mathbf{p}_t^i\}$ となる．

そこで，差の二乗和 (SSD) を基準としたテンプレートマッチングにより， $\mathbf{p}_t^i$ に対応すべき $l_{t-1}^j \in \mathcal{L}_{t-1}$ を選出する． $\mathbf{p}_t^i$ に対して選出した軌跡が l_{t-1}^j であるとき，この選択関係を，関数 $c_t(i) = j$ で記録しておく．

6.4 軌跡間の類似度算出

次に，2つの軌跡 $l_t^i, l_t^j \in \mathcal{L}_t$ の類似度 S_t^{ij} を定義する．これは，エッジによる類似度および軌跡間の距離による類似度から成る．類似度 S_t^{ij} は式 (6.3) により，前フレームでの類似度 $S_{t-1}^{c_t(i)c_t(j)}$ を用いて帰納的に定義される．

$$S_t^{ij} = T_{\text{weight}}^{\text{cut}} ED + (1 - T_{\text{weight}}^{\text{cut}}) S_{t-1}^{c_t(i)c_t(j)} \quad (6.3)$$

E は入力画像 $I_t(\mathbf{p})$ 上の2点 $\mathbf{p}_t^i, \mathbf{p}_t^j$ 間のエッジの強さであり，エッジが強いほど1に近い値をとる． D は $\mathbf{p}_t^i, \mathbf{p}_t^j$ 間の距離の近さを示す尺度であり，2点が近いほど1に近い値をとる． $T_{\text{weight}}^{\text{cut}}$ は重み係数である．つまり，前フレームでの類似度 $S_{t-1}^{c_t(i)c_t(j)}$ を，現在フレームの情報 E, D を用いて更新していくという方法で S_t^{ij} を得る．以後，これら E, D をそれぞれエッジによる類似度，距離による類似度と呼ぶ．

6.4.1 エッジによる類似度 E

2点間のエッジ強度を得るために，次の手順でブロック化された車両エッジ画像を作成する．まず，入力画像 $I_t(\mathbf{p})$ に Sobel フィルタをかけ，エッジ画像を作成する．計算量を削減するためにエッジ画像をブロック化し，各ブロック内での最大値を当該ブロックでの値として採用する．ブロックごとに入力画像と背景画像との正規化相関を計算し，相関値をしきい値 $T_{\text{nc}}^{\text{cut}}$ と比較することで背景成分を除去する．最後に，それぞれのブロックに

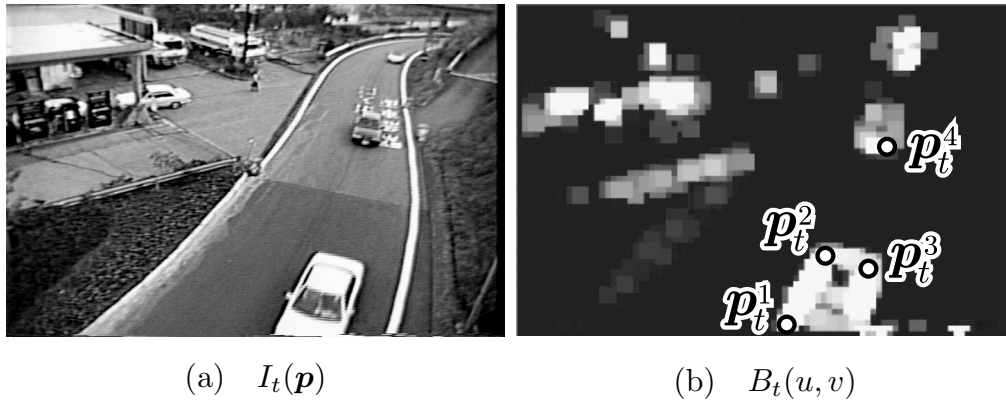


図 6.3 2 点間のエッジ強度

において、隣接するブロックの最大値を当該ブロックの値として採用することでエッジを太める．これをブロック化された車両エッジ画像 $\hat{B}_t(u, v)$ とする (図 6.3(b)).

2 つの特徴点 $\mathbf{p}_t^i, \mathbf{p}_t^j$ の間のエッジの強さ e は, $\mathbf{p}_t^i, \mathbf{p}_t^j$ 間を結ぶ線分上における $\hat{B}_t(u, v)$ の最小値とする．図 6.3(b) において, $\mathbf{p}_t^1, \mathbf{p}_t^2$ 間には途切れなくエッジが繋がっているため, e は大きな値をとる．一方, $\mathbf{p}_t^3, \mathbf{p}_t^4$ 間ではエッジが途切れているため, e は小さな値をとる．こうして求めた e を 0~1 の値に正規化し, エッジによる類似度 E とする．

6.4.2 距離による類似度 D

$\mathbf{p}_t^i, \mathbf{p}_t^j$ 間の画像平面上での距離 $d = \|\mathbf{p}_t^i - \mathbf{p}_t^j\|$ を求め, これを 0~1 の値に正規化する．このとき, 距離が遠くなるほど類似度 D が減少するようにする．

6.4.3 前フレームの分割結果を用いた類似度の修正

例えば軌跡 $l_t^i, l_t^j \in V_t$ が時間 $t-1$ において同一の車両の軌跡であった場合, これらは時間 t においても同一の分割に属する可能性が非常に高いものと考えられる．そこで, 類似度 S_t^{ij} に 1 よりも大きい定数係数 $T_{\text{modify}}^{\text{cut}}$ を掛け, 重みを強化する．一方, 軌跡 $l_t^i, l_t^j \in V_t$ が時間 $t-1$ において異なる車両の軌跡であった場合, これらは時間 t においても異なる分割に属する可能性が非常に高いものと考えられるため, 類似度 S_t^{ij} に 1 よりも小さい定数係数 $T_{\text{weaken}}^{\text{cut}}$ を掛け, 重みを弱める．この修正処理により, 前フレームの分割結果が保持されるように促される．

6.5 拘束付きグラフ分割による車両追跡

6.5.1 問題の定義

グラフ $G_t = (V_t, E_t)$, $w_t(e)$ $e \in E_t$ を次のように定義する.

$$V_t = \mathcal{L}_t \quad (6.4)$$

$$E_t = \{e_t^{ij} = \langle l_t^i, l_t^j \rangle \mid l_t^i, l_t^j \in V_t, i < j\} \quad (6.5)$$

$$w_t(e_t^{ij}) = S_t^{ij} \quad (6.6)$$

$e_t^{ij} = \langle l_t^i, l_t^j \rangle$ は l_t^i, l_t^j 間の辺である. G_t は全ての頂点間に辺を持ち, その重みとして類似度 S_t^{ij} が与えられている.

完全グラフ G_t において, 同一の車両から検出された軌跡同士は強い辺重みで連結し合う. そこで, グラフ分割法の一つである Normalized Cuts[46, 52] を適用し, 特徴点群をクラスタリングすることで各車両の位置を得る. Normalized Cuts では, 式 (6.7) に基づき, グラフ G_t の頂点 V_t を K_t 個の頂点集合 V_t^k ($1 \leq k \leq K_t$) に分割する.

$$\text{Ncut} = \frac{1}{K_t} \sum_{k=1}^{K_t} \text{linkratio}(V_t^k, V_t) \quad (6.7)$$

但し,

$$\text{linkratio}(V_t^k, V_t) = \frac{\sum_{l_t^i \in V_t^k, l_t^j \in V_t^k} w(e_t^{ij})}{\sum_{l_t^i \in V_t^k, l_t^j \in V_t} w(e_t^{ij})} \quad (6.8)$$

であり, V_t^k に重複は無く, $\bigcup_{k=1}^{K_t} V_t^k = V_t$ および $V_t^{k_1} \cap V_t^{k_2} = \phi$ ($k_1 \neq k_2$) を満たすものとする. Ncut 値を最大化する V_t^k が最良の分割結果である.

車両同士が近接する場合, 車両間の辺重みが大きくなるために適切に分割されなくなるが, 多くの特徴点は過去のフレームの特徴点との対応がとれているため, 前フレームにおいて対応する特徴点がどの車両に属していたのかを調べれば, 当該特徴点がどの車両に属する可能性が高いのかをあらかじめ知ることができる.

そこで, 前のフレームの分割結果を事前知識として用い, 探索すべき解空間から可能性の低い解を事前に取り除き, 前のフレームの分割結果が保持されるような解のみを探索することで, オクルージョンに対処する. ここで, c 個の拘束点群 $C_k \subset V_t$ ($1 \leq k \leq c$) を考える. これは, 過去のフレームにおける分割結果を参考にし, 少なくとも同じ車両に属するであろう頂点群をそれぞれ集合としてまとめたものである. 車両追跡の問題は,

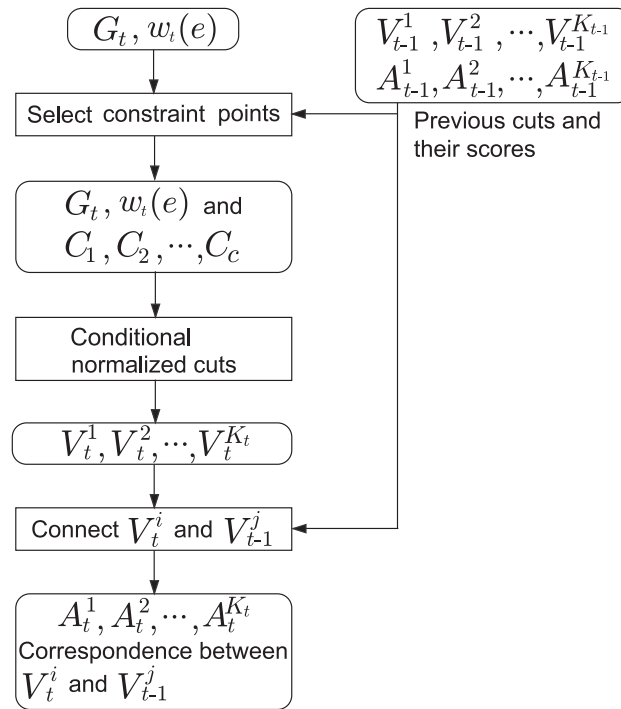


図 6.4 グラフ分割による追跡の流れ

式 (6.9) の拘束条件のもとで、Ncut 値を最大化する V_t^k および K_t を求める問題に帰着される。

$$C_k \subset V_t^k \quad (1 \leq k \leq c) \quad (6.9)$$

式 (6.9) の拘束条件により、 C_k の拘束点群は必ず同じグループに属するように分割される。一方、異なる拘束点群 C_{k_1}, C_{k_2} に属する頂点同士は、必ず異なるグループに属するように分割される。図 6.1(c3) の例では 2 つの拘束点群が入っており、四角形と三角形がそれぞれ C_1, C_2 に対応する。この拘束条件のもとで Normalized Cuts を計算することで、重なり合った車両を分離できるようになる (図 6.1(c4))。

ここで問題となるのは、拘束点群 C_k をどのように選択するのが望ましいか、また、Ncut 値を式 (6.9) の拘束のもとで最大化する最適化問題をどのようにして解くかの 2 点である。以下、これら 2 点を含め、全体の流れについて述べる。

6.5.2 大まかな流れ

図 6.4 はグラフ分割による車両追跡の流れを示したものである。まず、前フレームまでの追跡結果を用いて拘束点群 C_k を生成する。次に式 (6.9) の拘束のもとで G_t に Normalized Cuts を適用し、分割 $V_t^1, V_t^2, \dots, V_t^{K_t}$ を得る。これらと過去の分割 V_{t-1}^j を対応付けることで車両の追跡を行う。このとき、各分割 V_t^i に累積スコア A_t^i を付与し、

累積スコアが十分高い分割を車両として判別する.

なお, 累積スコアの与え方, および分割 V_t^i, V_{t-1}^j の対応付け方法は5章で提案した方法と同様であるため, ここでは説明を省略する.

6.5.3 拘束点群の選択

時間 t の頂点集合 V_t のうち, 前のフレームにおいて同一の車両 V_{t-1}^k に属していた頂点集合を C'_k とする. 頂点集合 C'_k に属するそれぞれの頂点は時間 $t-1$ においても同一の車両に属する可能性が高いものと考えられる.

C'_k をそのまま拘束点群として用いることもできるが, 拘束点群が多過ぎると, 前フレームの分割結果に強く依存し過ぎてしまう. そこで, 拘束点群はある程度間引いて選択する. 選択の仕方は幾つか考えられるが, ここでは一様乱数を用い, 平面的に一様分布するように間引く. 一様分布にする理由は, 前フレームでの車両の形状を保つためである. 間引く割合はしきい値 $0 \leq T_{\text{rate}}^{\text{cut}} \leq 1$ で決める. $T_{\text{rate}}^{\text{cut}}$ が大きいほど多く間引かれる. C'_k から間引かれた点群を拘束点群 C_k とする.

なお, 累積スコアが充分高く, 車両として既に認識されている車両 V_{t-1}^k のみを拘束点群を与える対象とする. 時間 $t-1$ において車両が c 台存在したのであれば, c 個の拘束点群 $C_1, C_2, \dots, C_c$ が生成される. これらの拘束点群を用い, 拘束のもとで Normalized Cuts を計算する.

6.5.4 拘束付き Normalized Cuts

c 個の拘束点群 $C_1, C_2, \dots, C_c$ により制約された解空間内において, V_t を K_t 個の頂点集合 $V_t^1, V_t^2, \dots, V_t^{K_t}$ に分割する解を探索する. このとき K_t は未知であり, V_t^i とともに K_t も定めなければならない.

Ncut 値を最大化する最適解は, 一般化固有値解析によって近似的に得られることが示されている [46, 52]. しかしながら, この解法は拘束点を用いることを考慮しておらず, 解空間が制約されている場合には適用することができない. 拘束を付与したうえで Normalized cuts を計算する手法は幾つか提案されているが [53, 54], 本論文の問題定義に適する解法は提案されていない. 文献 [53] では文献 [46] で提案された Normalized Cuts の解法を拡張し, 指定した拘束 C_k の頂点集合は必ず同じグループに属するように分割する解法が提案されているが, この手法は異なる拘束 C_{k_1} および C_{k_2} の頂点同士が異なるグループに分割されることは保証していない. 文献 [54] では異なる拘束 C_{k_1} および C_{k_2} の頂点同士が異なるグループに分割されるように分割する手法が提案されているが, この手法は解の近似精度が悪く, 良好な解が得られないことが知られている. また, これ

らの手法は双方ともに分割数 K_t を事前に決めておく必要がある．しかしながら，我々は画面上に存在する車両数を事前に知ることはできない．処理しているフレームで新たに車両が現れたり，消えたりするからである．

そこで，組み合わせ最適化法的一种である焼きなまし法 [47] を適用し，拘束のもとで Ncut 値を最大化する解を探索し， c 個に分割する．その後，分割したそれぞれに対して，拘束を用いない通常の Normalized Cuts を適用することで， K_t 個の分割を得る．

焼きなまし法による分割

焼きなまし法を用い， c 個の拘束点群 $C_1, C_2, \dots, C_c$ をもとで V_t を c 個の頂点集合 $V_t^1, V_t^2, \dots, V_t^c$ に分割する．焼きなまし法は評価関数の定義や解空間の構造によらずに適用できる汎用的な手法であるため，式 (6.9) の拘束のもとでも Ncut 値を最大化する解を探索することができる．焼きなまし法を適用するためには，(a) 解の近傍構造，(b) 初期解，(c) 評価関数，(d) 温度パラメータの制御，(e) 停止条件の 5 つを決める必要がある．

(a) 解の近傍構造

焼きなまし法は，解を少しずつ変化させることで徐々に評価関数を最大化してゆき，最適解を探索する．最適解までの到達速度は，解の変化のさせ方に強く依存する．解を大きく変化させると，大局解の探索能力は高まるが，その反面，局所解の精細な探索能力は低くなる．本手法では，充分最適解に近い初期解が得られるため，式 (6.10) に従い，1 つの頂点 l を入れ替えることで，解を遷移させる．

$$V_t^i \leftarrow V_t^i \setminus l, \quad V_t^j \leftarrow V_t^j \cup l \quad (i \neq j) \quad (6.10)$$

最も簡単な方針としては，頂点 l を V_t からランダムに選び，入れ替える方法が考えられる．しかしながら，目的とする解は平面的に連続性を保っているはずである．そのため，各分割の境界付近の頂点から l を選ぶ方が効率が良い．

そこで，特徴点の画像平面上の座標 p_t^i に従ってドロネー三角形分割アルゴリズムを適用し，グラフ G_t を平面化したグラフ G_t^{plain} を別に作成する．この平面グラフの構造を参照し，各分割の境界に位置する頂点集合 $\mathcal{N}$ を求める． $\mathcal{N}$ から拘束点群を除いた集合 $\hat{\mathcal{N}}$ が入れ替える頂点 l の候補となる (式 (6.11)).

$$\hat{\mathcal{N}} = \mathcal{N} \setminus \bigcup_{k=1}^c C_k \quad (6.11)$$

式 (6.11) により，拘束点群 $C_1, C_2, \dots, C_c$ は入れ替える頂点 l として選択されることはない．そのため，拘束点群に与えられた初期解は常に固定されることになるため，結果的に式 (6.9) の拘束のもとで Ncut 値を最大化することになる．

(b) 初期解

初期解はランダムに与えても良いが、ここで最適解に充分近い解を与えておけば、最適化の際の反復計算が少なくて済む。ここでもまた、前フレームによる分割結果を用いることが効果的である。そこで、 $V_t^i \leftarrow C_i'$ として初期解を与えておく。

過去のフレームとの対応がとれていない点については、すでに初期解が決定した点との結びつきの強さを調べ、最も結びつきが強いと思われる点の解を与えておく。時間 t において新しく出現した頂点の中には、初期解が決定しているどの点とも結びついていないものが有る。そのような点は V_t^1 を解として与えておく。

(c) 評価関数

式 (6.7) で示した Normalized Cuts の評価式をそのまま用いる。

(d) 温度パラメータの制御

焼きなまし法では、温度と呼ばれるパラメータ T_{SA}^{cut} を制御することで解の動きを制御する。通常、はじめの段階では温度 T_{SA}^{cut} を高く設定することでランダムな解の移動を生じやすくする。そして探索が進むにつれて T_{SA}^{cut} 徐々に小さくし、探索範囲を絞ってゆくことで、最適解を見つける。しかし、本手法では始めの時点で充分良い初期条件が得られているため、はじめから T_{SA}^{cut} を充分小さく設定し、温度の制御は行わない。

(e) 収束条件

解の遷移と評価を繰り返し、一定回数 T_{count}^{cut} の間、解の改善がみられない場合、最適化を終了する。そのときまでに得た最良の解を求める解とする。

再分割

以上の処理で得た c 個の分割それぞれに対して Normalized Cuts を適用し、さらに分割する。Ncut 値がしきい値 T_{ncuts}^{cut} 以下になるまで2分割を繰り返す。ここでは拘束点を用いずに分割するため、焼きなまし法ではなく通常の一般化固有値解析による手法を適用する。以上の処理により、 K_t 個の頂点集合 V_t^i を得る。

6.6 実験

空間解像度 360×240 、時間解像度 30fps の6種類の動画像を用いて提案手法の有効性を評価した (図 6.5)。Scene1~4 は昼間に撮影したものであり、Scene5,6 は夜間に撮影したものである。Scene1 は最も好条件で、車両の停滞や重なりが非常に少ない。Scene2 は画像上での車両の大きさが、6 シーン中最も小さい。Scene3 は信号の手前5車線を同時に撮像したものであり、1分前後の停滞が3回起こる。Scene4 は奥から手前に向かって車両が走行するシーンであり、近づくにつれて車両が非常に大きくなる。Scene5 は夜の事例であり、小雨が降っている。車両を後方から捉えており、遠方では車両同士が著しく

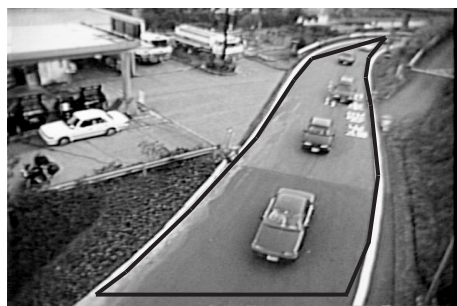
表 6.1 しきい値

T_{λ}^{fp}	0.03	$T_{\text{block}}^{\text{cut}}$	4	$T_{\text{score}}^{\text{cut}}$	10
$T_{\text{window}}^{\text{fp}}$	5	$T_{\text{window}}^{\text{cut}}$	12	$T_{\text{modify}}^{\text{cut}}$	1.5
$T_{\text{nc}}^{\text{fp}}$	0.5	$T_{\text{nc}}^{\text{cut}}$	0.5	$T_{\text{weaken}}^{\text{cut}}$	0.3
$T_{\text{range}}^{\text{fp}}$	8	$T_{\text{normE}}^{\text{cut}}$	50	$T_{\text{rate}}^{\text{cut}}$	0.75
$T_{\text{ssd}}^{\text{fp}}$	10000	$T_{\text{normD}}^{\text{cut}}$	75	$T_{\text{SA}}^{\text{cut}}$	0.0001
$T_{\text{weight}}^{\text{cut}}$	0.9	$T_{\text{num}}^{\text{cut}}$	5	$T_{\text{count}}^{\text{cut}}$	1000
				$T_{\text{ncuts}}^{\text{cut}}$	0.98

近接する。Scene6 もまた夜に小雨が降っている事例である。信号の手前の停止線付近を撮影しているため、車両が頻繁に停車する。このとき、右側 3 車線ではヘッドライトを点灯させた車両が密集するため、条件としては非常に悪い画像である。

図中の黒枠は監視領域であり、本実験ではこの領域内のみを計測対象とした。それぞれの動画像において、交通流量の少ない 1000 フレーム程度の部分を選択し、これを時間方向に平均化して合成した画像を背景画像として用いた。表 6.1 は実験で用いたしきい値の一覧である。Scene1～6 からそれぞれ 1 分程度の動画像を無作為に選択し、それらを用いてしきい値を調整した。以後の実験結果は、シーンによらず表 6.1 のしきい値を用いて得たものである。

5 章では、 $T_{\text{modify}}^{\text{cut}} = 5.0$ および $T_{\text{weaken}}^{\text{cut}} = 0.0$ としていたが、本章では $T_{\text{modify}}^{\text{cut}} = 1.5$ および $T_{\text{weaken}}^{\text{cut}} = 0.3$ とした。すなわち、重みの修正量を減らしている。拘束点を用いることにより、オクルージョン時に車両同士を分離する性能が向上したため、重みを大きく修正する必要が無くなったからである。その他の閾値については、5 章で用いた値を設定した。



(a) Scene 1 (15 分 00 秒)

追跡成功率：97.91%

処理速度：61.18msec/frame



(b) Scene 2 (15 分 00 秒)

追跡成功率：83.05%

処理速度：50.55msec/frame



(c) Scene 3 (7 分 19 秒)

追跡成功率：86.47%

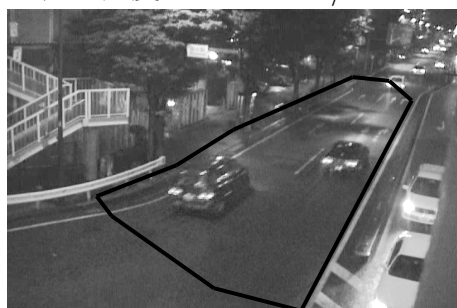
処理速度：63.64msec/frame



(d) Scene 4 (2 分 55 秒)

追跡成功率：82.35%

処理速度：122.33msec/frame



(e) Scene 5 (15 分 00 秒)

追跡成功率：95.04%

処理速度：50.13msec/frame



(f) Scene 6 (15 分 00 秒)

追跡成功率：81.34%

処理速度：59.49msec/frame

図 6.5 実験に用いた動画像

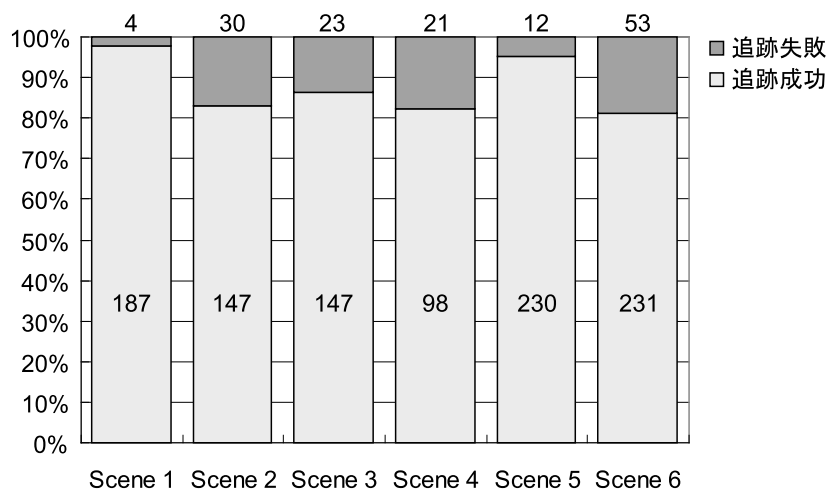


図 6.6 計測結果

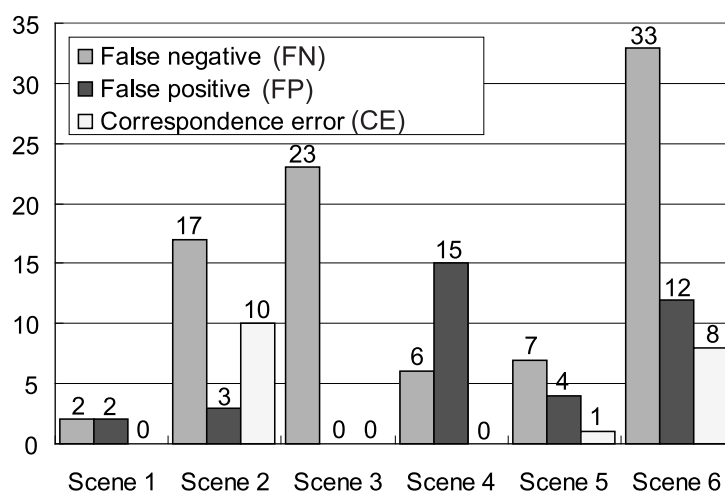


図 6.7 エラー要因

6.6.1 計測結果

図 6.6 に計測結果をまとめた。また、図 6.7 はそれぞれのシーンについてエラーの要因をまとめたものである。FN(False Negative) は見逃した車両の数を集計したものであり、FP(False Positive) は本来 1 台として認識すべきところを、複数の台数であるとして認識してしまった車両の数を集計したものである。また、CE(Correspondence Error) は、 V_t^i と V_{t-1}^j とを対応付ける処理が失敗することで、追跡がうまくいかなかった車両の台数を示している。

本手法を用いることで、全てのシーンについて、80% 以上の追跡精度を達成することができた。Scene3,6 は近接する車両が多く現れるため、5 章での追跡精度は 6~7 割程度で

あったが、本手法では拘束点群を導入したことにより近接車両を分離する性能が向上したため、追跡精度が10~20%程度向上した。

以下、さらなる性能改善の可能性について検討する。現状の処理では、1台の車両から抽出される特徴点の数は様々であり、これは画像平面上における面積に依存する。Scene2では車両の面積が小さいために十分な数の特徴点が検出できないことがあり、そのためにFNが多発する。また、特徴点の数が少ないと対応付けが失敗しやすくなり、CEも同時に多発するようである。一方、Scene4は手前に向かって車両が進行してくるため、1台から検出される特徴点の数が多くなり、またそれぞれの特徴点間の距離が遠くなる。特徴点間の距離が遠くなると類似度は小さくなる傾向があるため、FPが多発することになる。この問題を解決するためには、カメラの設置位置と路面との幾何学的関係を用い、一定の距離 x_{cm} ごとに特徴点が分布するように検出するようにし、1台から検出される特徴点の数をカメラの設置位置に依存しないようにすることが効果的であると考えられる。

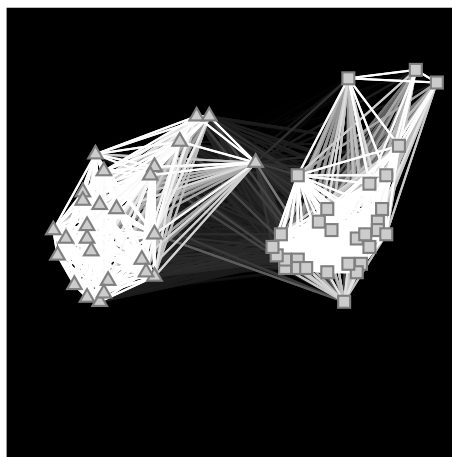
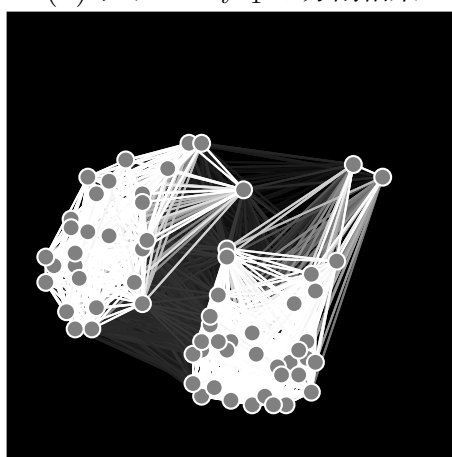
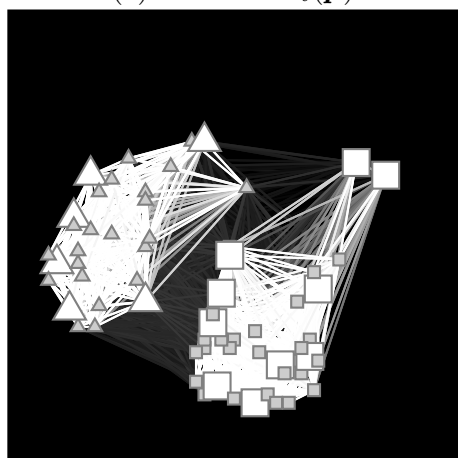
6.6.2 近接する複数車両の分離

離れていた二台が重なり合う場合

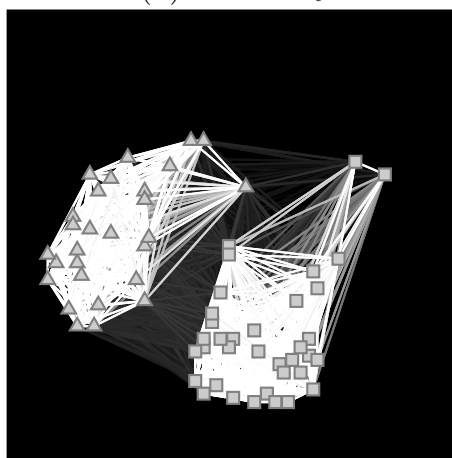
図6.8は始めに個別に追跡されていた2車両が近接し、重なりあったときの例である。(a)(b)は過去のフレームにおける入力画像、およびその特徴点のクラスタリング結果を示している。(c)は現在のフレームにおける入力画像 $I_t(\mathbf{p})$ であり、(d)は $I_t(\mathbf{p})$ から構築したグラフ G_t である。(b)のクラスタリング結果を参照し、拘束点群を用いて焼きなまし法による拘束付き分割を適用した結果が(e)である。ここで拘束点群が過去のフレームにおける分離の情報を保存するように働くため、近接する車両が2つに分離される。(f)は(e)で得たそれぞれの分割に対して再分割処理を適用した結果である。それぞれの分割の頂点群は、既に充分強い重みで結びついているため、ここでは分割は行われず、最終的な結果として2台の車両が得られる。

重なり合っていた二台が離れていく場合

図6.9は始めに重なり合っていた2車両が後に分離する過程を示したものであり、(a)-(f)は図6.9と同じものを示している。図6.9の事例では、始めの段階ではシステムは2車両を1台として認識する。しかし、徐々に2車両が離れ、車両間の重みが充分弱くなったときに、(f)における再分割処理によって分離が行われる。以後それぞれが個別の車両として追跡されるようになる。2台に分かれたとき、一方はこれまで追跡してきた車両として認識され、もう一方はその時点で新しく現れた車両として認識されることになる。

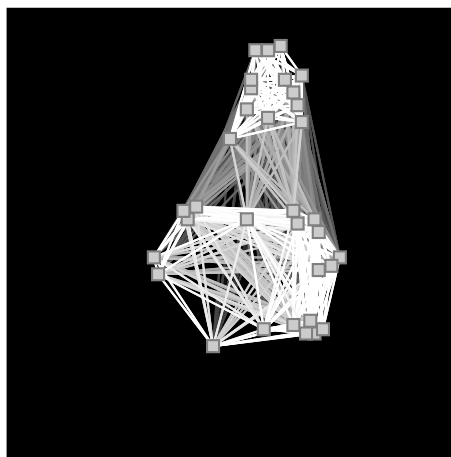
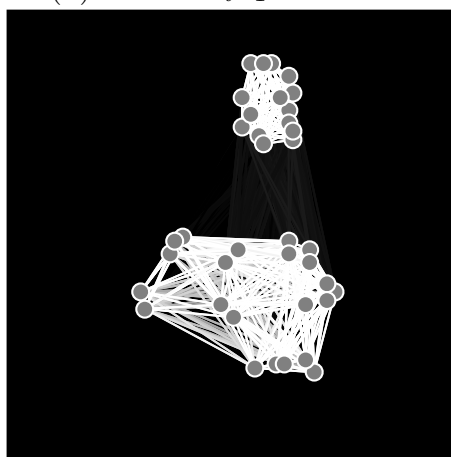
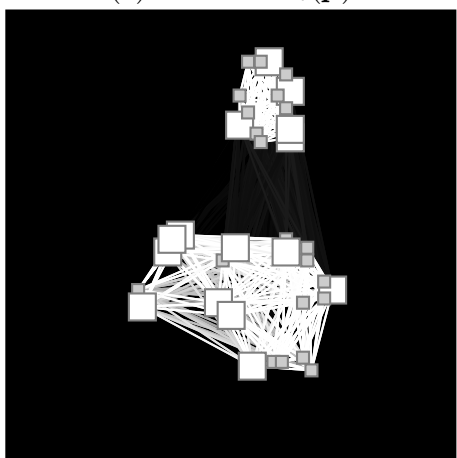
(a) 入力画像 $I_{t-1}(p)$ (b) グラフ G_{t-1} と分割結果(c) 入力画像 $I_t(p)$ (d) グラフ G_t 

(e) 拘束付きグラフ分割

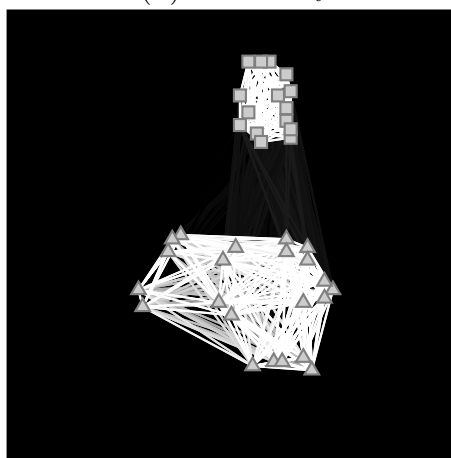


(f) 再分割

図 6.8 近接しつつある 2 車両

(a) 入力画像 $I_{t-1}(p)$ (b) グラフ G_{t-1} と分割結果(c) 入力画像 $I_t(p)$ (d) グラフ G_t 

(e) 拘束付きグラフ分割



(f) 再分割

図 6.9 離れつつある 2 車両

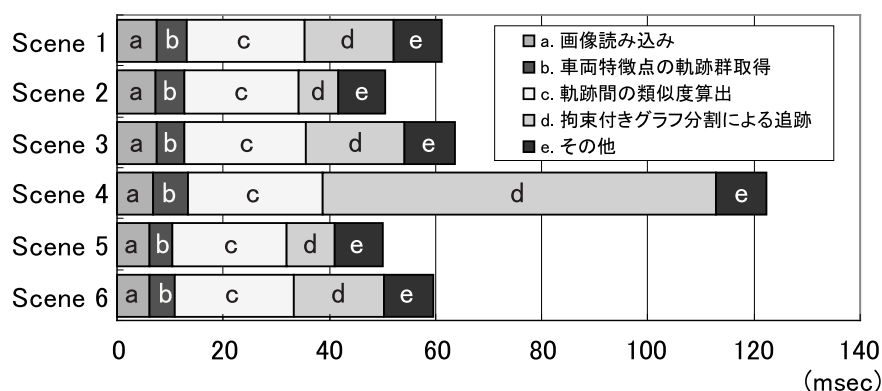


図 6.10 実行速度.

6.6.3 実行速度

図 6.10 は全てのシーンの実行速度を処理ごとにまとめたものである．計測に用いた計算機の CPU は Pentium4 で駆動周波数が 3.2GHz, メモリは 2GByte である．開発環境は Visual C++ 6.0 であり，Normalized Cuts で必要となる固有値計算のエンジンとしては MATLAB7.0.1 を用いた．

Scene4 を除く 5 つのシーンでは，1 フレームにかかる処理時間の平均値は 60msec 程度であった．30fps の動画像をリアルタイムで処理するためには，1 フレームを 33.3msec 以下で処理できれば良いため，リアルタイムにほぼ近い処理速度を達成できたといえる．

一方，Scene4 における 1 フレームあたりの平均処理時間は 122.3msec であり，他のシーンよりも若干遅かった．これは，Scene4 は画面上における車両のサイズが大きいいため，他のシーンよりも多くの特徴点が検出されるためである．そのため，Normalized Cuts を計算するのに時間がかかってしまう．

この対策としては，特徴点を間引いて検出することや，グラフ分割処理を数フレームおきに飛ばして処理することなどが有効であると考えられる．いずれの方法も，大幅な高速化を望めるものと期待される．

6.7 総括

本章では，車両のオクルージョン問題に焦点を当て，グラフ分割に拘束を付与するという新たな概念を導入することでこの問題を解決した．4 章および 5 章で得た成果と本章での成果を合わせることで，車両のオクルージョン，および照明条件の変化とカメラの設置位置の相違に起因する見えの変化に頑健な車両計数手法を実現した．

性能評価のために用いた6シーンのうちScene2~6は、あえて条件が厳しいものを選んだものである。それでも全てのシーンにおいて80%以上の精度を得たことから考えると、実験で用いたシーン以外の多様な環境下でも、高い精度が見込めるものと期待される。特に高速道路のように車両の移動軌跡が単純であり、オクルージョンが生じる回数が少ないような状況では超音波や誘電コイルなどの局所型センサに迫る性能が期待できるであろう。オクルージョン性能をさらに検討するためには、駐車場や交差点など、車両同士が複雑に動くシーンに対しても詳細に評価するべきであるが、現時点では動画像を入手できなかったため、この評価は今後の課題としたい。だが、Scene3,6のようなオクルージョンが多発するシーンにおいても80%以上の性能を示したことから、同程度の性能は見込められると思われる。

本章までの議論で提案した車両計数手法は、3.1節「本論文の目的」で述べた5つの性質全てを満たしており、**「道路上の監視カメラを用いた、低コストで高性能な交通流計測システム」**として実用上、非常に有益な手法を提案できたものといえる。

第 7 章

その他の応用事例

7.1 対象物体追跡のフレームワーク

前章までの議論により，車両を特徴点の群としてモデル化し，特徴点間の類似度を重みとした完全グラフの分割問題に帰着することで，車両追跡の問題を効果的に解けることが示された．本章では，この追跡手法の枠組みを多層から成るシステムとして再解釈することで，本手法のさらなる応用可能性について議論する．

図 7.1 は拘束付きグラフ分割による追跡モデルの概念図を示したものである．我々が提案した追跡モデルは Low level layer と High level layer の 2 階層に分けることができる．Low level layer の主な役割は，対象物の構成要素となる特徴量の抽出と，それらの間の類似度を定義することである．High level layer の役割は対象物の構成要素をクラスタリングすることであり，拘束付きグラフ分割が High level layer に相当する．

拘束付きグラフ分割は「対象物の構成要素」「要素間の類似度」「事前知識 (拘束点)」を与えることで適切なクラスタを出力するものである．前章までの交通流計測では，「対象物の構成要素」を「特徴点」とし，「要素間の類似度」を「エッジ・ユークリッド距離による類似度」で与え，「事前知識 (拘束点)」を「前のフレームにおける分割結果」から与えた．これらは車両の形状や道路状況に関する事前知識を用いずに構成したため，この Low level layer による対象物体追跡システムは他の追跡対象，例えば非剛体である人物などにも適用可能であると考えられる．そこで，この点について 7.2 節で検討する．また，前章までの交通流計測では 1 台の固定カメラを用いていたが，Low level layer の構成を適切に変更すれば，車に設置された移動カメラや，監視用のステレオカメラなど，交通流計測とは全く異なる撮影状況・撮影機器でも図 7.1 で示したフレームワークは有効であると考えられる．そこで，7.3 節では車載カメラによる後方／前方車両監視，7.4 節では監視用ステレオカメラによる侵入者検知を例にとって検討を行う．

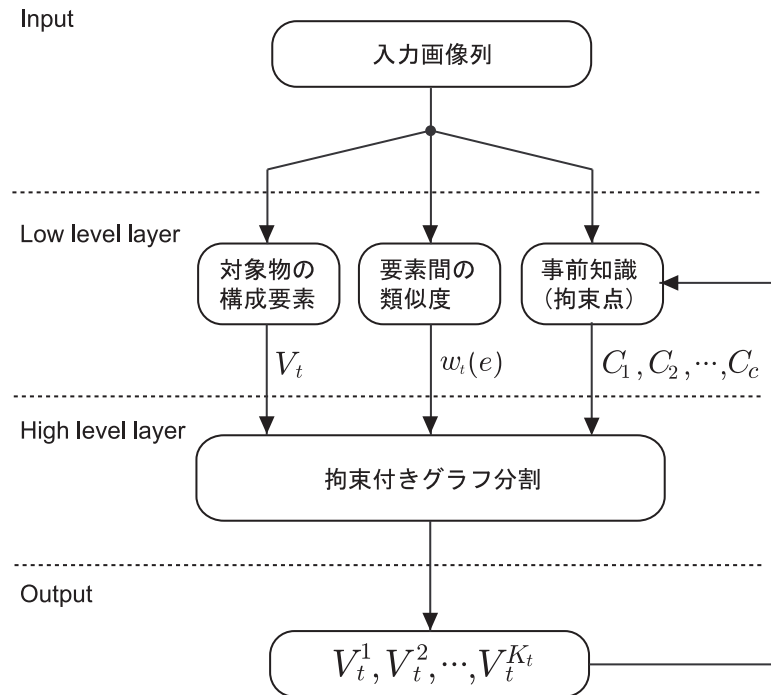


図 7.1 拘束付きグラフ分割による追跡のフレームワーク

7.2 応用事例 1：テニスの選手追跡

前章までに提案した交通流計測のための Low level layer は、追跡対象の形状について一切の仮定を置かずに構成した。それゆえ、特徴点さえ安定に抽出できるのであれば、車両以外の様々な対象物へも適用可能である。

図 7.2 はテニスの試合を撮影した動画像に対して本手法を適用した例である。動画像の空間解像度は 360×240 pixel であり、画像上における選手の大きさは 50×80 pixel 程度である。この程度の大きさで撮影されていれば、個々の選手からは十分な数の特徴点が検出できることがわかった。車両のような剛体と、人物に代表される非剛体の決定的な違いは、明度パターンが時々刻々と変容することにある。人間は動くときに間接部分が大きく動くため、画像上では明度パターンが動作に合わせて様々に変化する。そのため、特徴点の時間方向の対応付けは、剛体に比べて比較的難しい。特徴点の時間方向の対応付けの精度は、剛体に適用するときと比べて悪化していると考えるのが妥当である。しかしながら、グラフ分割による追跡の枠組みはこの対応付けの精度をそれほど要さない。ある程度正しい対応付けが成されていればそれで十分である。本手法をテニスの試合の動画像に適用してみると、人物のような非剛体であっても問題なく追跡できるという興味深い結果が

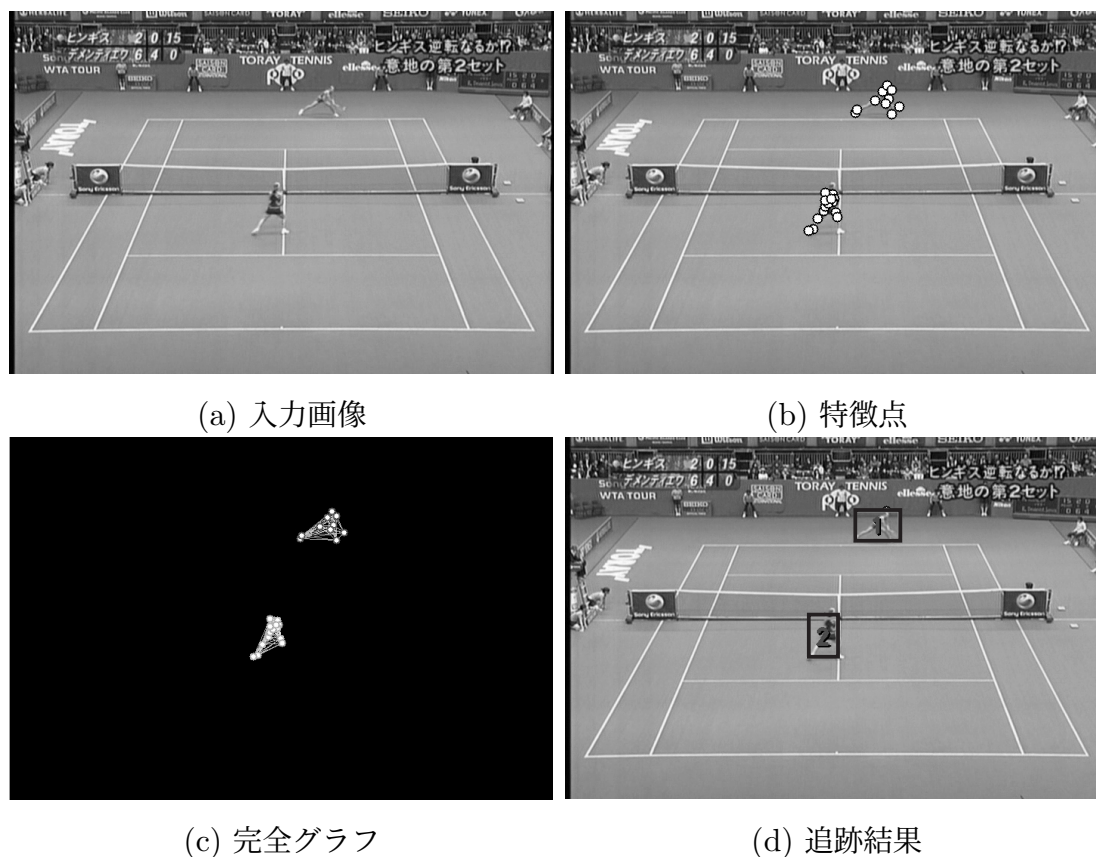


図 7.2 テニスの試合における選手追跡

得られた．このことから，本手法は交通流計測の他にも様々な応用可能性を持った柔軟な手法であるといえる．

7.3 応用事例 2：車載カメラによる後方／前方車両監視

車にカメラを搭載し，前方および後方を監視することで運転手の安全確認作業を補助しようと試みる研究事例が数多くある．我々が扱ってきた状況とは異なり，この目的においてはカメラが移動するという特色を持つ．交通流計測では固定された監視カメラを用いるのが一般的である．時間帯の推移やヘッドライトによる照明条件の変化はあるものの，背景は基本的には変わらない．そこで背景画像との正規化相関を計算することで，不要な背景成分を除去していた．一方，車載カメラは移動するため，背景画像を得ることができない．そのため 6 章までで述べた手法をそのまま適用することはできないが，Low level layer の構成を次のように変更することで対応できると考えられる．

- 対象物の構成要素

車載カメラから前方および後方を撮影した画像に現れる車両は，屋根やバンパー，

車両の底部などに多くの水平エッジ成分を含んでいる．そこで，一定以上の長さを持つ水平エッジ成分の集団として車両をモデル化する．

- 要素間の類似度

検出対象は剛体であることから，同一の車両から検出された水平エッジの動きベクトルは，画像平面上でアフィン拘束，あるいは射影拘束を満たすものと考えられる．また一方で，異なる移動物体同士の水平エッジや，移動物体の水平エッジと背景起因の水平エッジの間ではこの拘束を満たさないものと考えられる．そこで，この拘束を満たすか否かを評価することで，要素間の類似度を決定する．

- 事前知識 (拘束点)

前フレームにおける分割結果において同一の車両であると判断されていた水平エッジのうちの幾つかを，事前に拘束点として選択しておく．これにより，クラスタリングの安定性およびオクルージョンへの耐性が向上するものと考えられる．また，水平エッジの中心点の x 座標がほぼ厳密に一致し，長さもまた同じような水平エッジ同士は同じ車両に属する可能性が極めて高い．そこでそれらを拘束点として選択し，同一の車両として認識するように促すという方法も有効であると考えられる．



図 7.3 後方接近車両の水平エッジ成分

7.4 応用事例 3：監視用ステレオカメラによる侵入者検知

監視カメラで撮影した動画像内に現れる人物を追跡し，その挙動についての記録を取ることができればセキュリティの面で非常に有意である．これまでに我々の研究グループが

行った研究事例として、ステレオカメラを用い、特定の位置に侵入した人物を検出するという研究がある [56]。これは、距離画像を用いることで侵入者を検出するというものであり、対象の 3 次元的な位置が求められるという利点があった。しかしながら採用した追跡モデルが単純であったため、侵入者の移動軌跡の記録を取るという目的には不十分であった。我々が提案した拘束付きグラフ分割による追跡モデルを適用することで、この問題は改善できると考えられる。以下に、Low level layer の構成についての一案を示す。



図 7.4 監視カメラ画像

- 対象物の構成要素

ステレオカメラが使える状況であれば距離画像を得られるため、距離画像の変化量を観察することで対象物の有無を判定することができる。そこで距離画像をブロック分割し、ブロック内における距離の変化が著しいものを侵入者の一部であるとみなす。侵入者はこれらブロックの集まりとしてモデル化される。

- 要素間の類似度

画像平面上において近接しているブロック同士は、同一の追跡対象を示している可能性が高い。また、対象物までの距離の値が似ているブロック同士も同様に、同一の追跡対象を示している可能性が高いものと考えられる。そこで、これら 2 つの条件を両方満たすブロックの組み合わせに対して高い類似度を設定する。

- 事前知識 (拘束点)

複数の人物がオクルージョンを起こす状況を想定し、前フレームにおける分割結果において同一の人物であると判断されていたブロックのうちの幾つかを、拘束点として選択する。

7.5 総括

本章では、グラフ分割による追跡手法の枠組みを Low level layer と High level layer から成る一般的な追跡モデルとしてとらえ、他の産業的応用にどのように結びつくのかを具体例を交えて議論した。High level layer は拘束点群と重み付きグラフを入力とする層であり、分割された頂点群を出力する。この入出力構造は前段の Low level layer に依存しない。対象に応じて Low level layer をカスタマイズすることで様々な対象を追跡できるようになるであろうと期待される。

第 8 章

結論

本論文では、「道路上の監視カメラを用いた、低コストで高性能な交通流計測システム」を開発することを目的とし、以下の 5 つの性質を満たすマシンビジョン技術による交通流計測手法を提案した。

1. 1 台のモノクロカメラで計測が可能
2. カメラの設置位置に関する制約が緩い
3. 照明条件の変化に強い
4. 車両の移動軌跡を取得可能
5. オクルージョンに強い

1 章の総括

1 章では、近年の道路交通事情と高度交通システム (ITS) について概説し、同時に ITS におけるマシンビジョン技術の位置づけについて述べた。ITS におけるマシンビジョン技術を「道路インフラにおけるマシンビジョン」と「車載カメラによるマシンビジョン」の 2 種類に分類し、それぞれにおいてどのような研究活動が展開されているかを体系的に分類して概説した。

2 章の総括

2 章では、交通流計測という目的のもとで開発されたマシンビジョン技術について調査した内容を述べた。従来研究を 8 種の分類、すなわち「テンプレートマッチングによる方法」「領域に基づく方法」「時空間画像による方法」「2 次元および 3 次元の形状モデルによる方法」「明度・色ヒストグラムを用いる方法」「Condensation を用いる方法」「特徴量のクラスタリングによる方法」「特殊なカメラによる方法」に分け、それぞれの特性について述べた。

3 章の総括

3 章では、本論文の目的、および従来研究における本論文の位置付けについて述べた。まず、本論文の目的を「道路上の監視カメラを用いた、低コストで高性能な交通流計測システム」を開発することにあると位置付け、そのためには5つの性質が必要であることを述べた。次に、2 章で調査した8種のマシンビジョン技術を比較することでそれぞれの特徴を明らかにし、我々の目的のためにはどのアプローチが望ましいのかを検討した。その結果、「特徴量のクラスタリングによる方法」を基礎とした方法が最も望ましいという結論を得た。

4 章の総括

4 章では、3つの性質（「1 台のモノクロカメラで計測が可能」「カメラの設置位置に関する制約が緩い」「照明条件の変化に強い」）を有する車両認識手法を提案した。車両を時空間内における特徴点の軌跡の集合としてみなし、これらの軌跡を同一の車両ごとにまとめることで車両を認識する新しい枠組みを提案した。このクラスタリングの問題は、特徴点の軌跡を頂点、軌跡間の類似度を辺重みとした完全グラフの分割問題とみなすことで効果的に解けることを示した。この原理によると、昼夜問わず同一のアルゴリズムで、多様な撮影位置、照明条件における画像に対して安定に車両を認識できるということを実験的に示した。

5 章の総括

5 章では、4 章で実現した3つの性質に加え、「車両の移動軌跡を取得可能」という性質を備えた車両計数のための手法を提案した。5 章では4 章で提案した手法を拡張し、車両が監視領域に進入してから出てゆくまでの間、車両を追跡することを可能とした。基本的な原理は4 章で述べた手法と同一であるが、特徴点の軌跡群を抽出し、グラフ分割処理で軌跡群をクラスタリングするという枠組みを、逐次的に毎フレーム適用することで車両の移動軌跡を得た。条件の異なる6つのシーンに対して実験を行い、異なる時間帯、道路状況に対しても、システムのしきい値を調整することなく適用できることを示した。

6 章の総括

6 章では、5 章で実現した4つの性質に加え、「オクルージョンに強い」という性質を備えた車両計数のための手法を提案した。オクルージョンの問題は、過去のフレームにおける分割結果を現在フレームに伝播させることで解決できるという考えに基づき、過去における追跡情報を拘束条件としたグラフ分割を適用することで、オクルージョンに頑健な車両追跡を実現した。5 章における方法では、渋滞などのように車両同士が頻繁に近接し合

う状況においては性能の低下が見られたが、拘束付きグラフ分割という概念を導入することで、性能を改善することに成功した。

7 章の総括

7 章では、6 章までに提案した拘束付きグラフ分割による追跡の枠組みを Low level layer と High level layer から成る一般的な追跡モデルとしてとらえ、他の産業的応用にどのように展開できるのかについて、具体例を挙げて議論した。

以上、各章において述べた内容を総括した。最後に、本論文を総括した結論を述べる。一般に、監視カメラが撮影対象とする道路状況は様々であるため、マシンビジョン技術でこれらの画像を取り扱うのは難しい問題であると考えられてきた。なぜなら、晴れ、曇り、雨などの天候による変化、時間帯の推移による照明条件の変化、カメラの設置位置と路面との相対位置関係による車両の見え方の相違、あるいはオクルージョンなどの諸問題がつきまとうからである。しかし我々はこれらの問題に対し、特徴点群の抽出と拘束付きグラフ分割によるクラスタリングという概念が有効な解決策になるということを示した。我々が提案した手法は、異なる時間帯、道路状況に対しても、システムのしきい値を調整することなく適用できるものであり、従来手法にはない耐環境性能を持つ車両計数手法を提案できたものといえる。交通流計測は多くの地点で長時間行われるべきものであり、本手法の目指す方向性は、交通流量の把握、事故検知を目的とした道路交通流監視システムの実現へと繋がってゆくと考えられる。

付録 A

特徴点

特徴点とはコンピュータビジョンにおいて最も基本的かつ重要な特徴量である。特徴点とは、同一対象を異なる時間、あるいは異なる視点から撮影したときに、それらの画像間において対応の取りやすい画素のことを指す。通常、対象物の角や線の交わった部分から選ばれることが多い。

A.1 Harris 作用素

Harris 作用素 [57] は画像の明度勾配に基づく特徴点抽出手法である。座標 $\mathbf{p} = (x, y)^\top$ における画像の明度を $I(\mathbf{p})$ で表し、画像の 1 階微分を $f_x = \frac{d}{dx}I(\mathbf{p})$, $f_y = \frac{d}{dy}I(\mathbf{p})$ で表すとき、Harris 作用素は次のように定義される。

$$\text{Harris} = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad (\text{A.1})$$

$$\mathbf{C} = \begin{pmatrix} G_\sigma(f_x^2) & G_\sigma(f_x f_y) \\ G_\sigma(f_x f_y) & G_\sigma(f_y^2) \end{pmatrix} \quad (\text{A.2})$$

ただし k は定数であり、 $G_\sigma(\cdot)$ は標準偏差 σ のガウス分布による平滑化を表す。 λ_1, λ_2 は $\mathbf{C}$ の固有値である。式 (A.1)(A.2) で定義される Harris 作用素が大きいほど、その画素は特徴的であるといえる。そこで、Harris 作用素を全ての画素について計算し、極大になる部分を特徴点として選択すればよい。このとき、式 (A.1) の極大値を大きいものから順に指定した数だけ選ぶ、という方法が取られることが多い。

A.2 Harris 作用素の変形

Harris 作用素は λ_1, λ_2 が共に大きい位置を探すことを目的としており、このためには他の基準も考えられる。Tomasi ら [49] は、特徴点を追跡する過程において、特徴点近傍

の幾何学的変換がアフィン変換で近似できるような場合には、式 (A.1) で定義した Harris 作用素よりも、 $\min(\lambda_1, \lambda_2)$ の方が性能が良いということを示した。

また、式 (A.2) ではガウス分布による平滑化を用いたが、計算を単純にするために正方形の窓内の平均で代用することも多い。その場合、 $\mathbf{C}$ は次のようになる。

$$\mathbf{C} = \sum_{\mathcal{N}(\mathbf{p})} \begin{pmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{pmatrix} \quad (\text{A.3})$$

なお、 $\mathcal{N}(\mathbf{p})$ は $\mathbf{p}$ の近傍に設定した正方形の窓内における位置ベクトルの集合である。

付録 B

Normalized Cuts

Normalized Cuts とは, J.Shi と J.Malik によって 1997 年に提唱されたグラフ分割アルゴリズムの一種であり, 主に画像の領域分割への応用を目的として考案された [46, 58, 52]. グラフ分割とは, グラフの頂点と頂点を結ぶ辺に割り当てられた重みをもとにし, 相互に結びつきの強い頂点群ごとにクラスタリングを行うものである. J.Shi と J.Malik は, 人間の直感に近いクラスタリング結果を与える評価基準 $Ncut$ を定義し, $Ncut$ 値を最大化する解法として固有値解析を用いることにより, 安定にグラフ分割の解を得られることを示した.

B.1 k-means 法とグラフ分割

クラスタリングにおける代表的な手法の一つに k-means 法がある. これは, あらかじめクラスタ数が k 個であると分かっているとき, おのおののクラスタに代表ベクトルを与え, それぞれの要素との距離が最も小さくなるような k 個の代表ベクトルの位置を探索するものである. この手法では, 各要素と代表ベクトルとのユークリッド距離の総和が小さくなるように代表ベクトルの位置を更新する. 多変量の数値データの場合, クラスタの代表ベクトルとして平均値 (mean) を用いることから, k-means 法と呼ばれる. k-means 法はアルゴリズムが簡潔でありながら多くの場合で良い結果を示すためしばしば用いられるが, そのクラスタリングの能力において本質的な限界がある. 例えば図 B.1 に示すように, 二重の輪のように分布する要素群が与えられたとき, k-means 法では適切なクラスタを求めることができない. k-means 法では, それぞれの要素を最も近い代表ベクトルに割り当てる. そのため特徴空間内においてクラスタの分布は等方的な球状にまとまっている必要がある. 図 B.1 のようなケースではこの条件を満たさないため, 適切なクラスタを得ることができない.

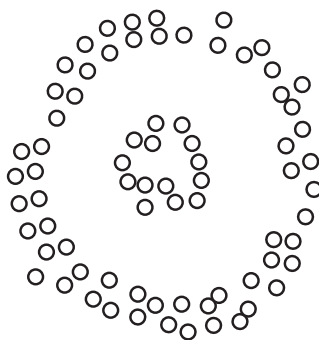


図 B.1 二重の輪

このケースにおいては，外側の輪の部分と内側の輪の部分で分けるのが自然であろう．グラフ分割によるクラスタリングは，こうした人間の直感的な分割に近い結果を与えるものである．グラフ分割では，クラスタリングの対象となる要素間にスカラー量である類似度を与え，相互に強く結びつき合っている要素群を抽出する．つまり，代表ベクトルを用いないという点で k-means 法と決定的に異なる．k-means 法のようにクラスタが特徴空間内で球状にまとまっている必要はなく，図 B.1 の事例においても適切なクラスタリング結果を得られる．

B.2 グラフ分割の定義

グラフ G とは頂点集合 V と，2つの頂点の組み合わせとして定義される辺集合 E から成り，まとめて $G = (V, E)$ と記述する．また，それぞれの辺が1つのスカラー量である重み $w(e)$ ($e \in E$) を持つとき，これを重み付きグラフという (図 B.2)．

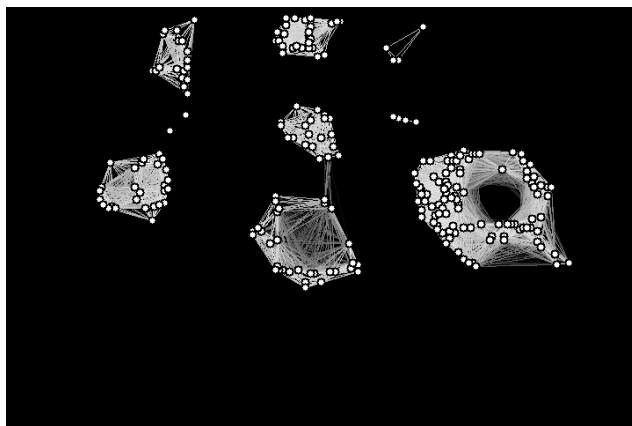


図 B.2 重み付きグラフ

グラフ分割とは、重み関数 (e) を用いて定義された評価関数をもとに、グラフ G の頂点 V を K 個の頂点集合 V^k ($1 \leq k \leq K$) に分割する手法のことである。なお、 V^k に重複は無く、 $\bigcup_{k=1}^K V^k = V$ および $V^{k_1} \cap V^{k_2} = \phi$ ($k_1 \neq k_2$) を満たすものとする。

ここで重要なのは、評価関数をどのように決めるのか、そして評価関数を最大化（もしくは最小化）する解 V^k ($1 \leq k \leq K$) をどのように見つけるか、という 2 点である。J. Shi と J. Malik は、この 2 点について詳細に検討し、人間の直感に近いクラスタリング結果を与える評価関数 Ncut を提案した。また、評価関数 Ncut を最大化する解 V^k ($1 \leq k \leq K$) を、固有値解析を用いることにより近似的に得られることを示した。

B.3 Multiclass normalized cuts

J. Shi と J. Malik が初めに提案した評価関数 Ncut は 2 分割を対象としたものであったが [46, 58]、その後研究が進み、Stella X. Yu および J. Shi によって k 分割を対象とした評価関数 Ncut が提案された [52]。これは、Multiclass normalized cuts、あるいは k -way normalized cuts と呼ばれる。Multiclass normalized cuts では、式 (B.1) で定義される評価関数に基づき、グラフ G の頂点 V を K 個の頂点集合 V^k ($1 \leq k \leq K$) に分割する。

$$\text{Ncut} = \frac{1}{K} \sum_{k=1}^K \text{linkratio}(V^k, V) \quad (\text{B.1})$$

但し、

$$\text{linkratio}(V^k, V) = \frac{\sum_{i \in V^k, j \in V^k} w(e^{ij})}{\sum_{i \in V^k, j \in V} w(e^{ij})} \quad (\text{B.2})$$

とする。式 (B.1) で定義した Ncut を最大化する V^k が最良の分割結果となる。

この定義式は次のような意味を持つ。式 (B.2) における $\text{linkratio}(V^k, V)$ の分子は、 k 番目の分割 V^k の各頂点間の結びつきの強さを示す。また分母は、 k 番目の分割 V^k と全ての頂点集合 V との結びつきの強さを示す。すなわち、linkratio を最大化するということは、分割内における結びつきが強くなり、かつ分割間の結びつきが弱くなるような解を探索することに相当する。

謝 辞

本研究は、著者が慶應義塾大学大学院理工学研究科在学中に、同大学理工学部小沢慎治教授のもとで行なったものである。本研究を遂行するにあたり、終始御指導くださり、本研究の内容の詳細にわたり貴重な御助言を賜りました小沢慎治教授に心から感謝いたします。小沢慎治教授には、著者が学部2年生の頃から後期博士課程卒業に至るまでの8年もの間、研究の面、生活の面、その他あらゆる点において貴重な御助言を頂きました。心より感謝の念を申し上げます。

そして、本研究の副査をご快諾頂き、また貴重なご意見を下さいました浜田望教授、川嶋弘尚教授、斎藤英雄教授に深く感謝いたします。副査の先生方からは技術的な観点からのご意見のみならず、本技術の今後の運用に関する方向性など、実に様々な視点からのご意見を頂きました。斎藤英雄教授とは研究室における輪講や発表の場を共有する機会も多く、そこでの議論も著者にとっては非常に有意義なものでした。心より感謝申し上げます。

著者が後期博士課程に在学してからは、幸運にも佐藤幸男教授と森山剛助手と議論を交わす場に恵まれました。お二人からは、何よりも研究者としての心構えを学びました。今後も初心を忘れずに、信念を持って研究活動に励む所存です。この場を借りて心より感謝申し上げます。また、著者の研究生活を支えてくれた小沢・斎藤研究室の諸先輩方に深く感謝いたします。著者が今後研究者として活動を展開してゆくことについて悩んでいたとき、木村誠氏には何度も相談に乗って頂きました。それがどれだけ著者の不安を拭い去ってくれたことか知れません。原寛徳氏、須藤智氏には配属時から長きに渡ってお世話になりました。心より感謝申し上げます。そして、研究室で多くの時間を共に過ごし、精神的支えになってくれた同期生の皆様、ならびに小沢・斎藤研究室の諸氏に、深く感謝いたします。

また、高校時代からの友人である上野正樹氏、森大祐氏、小田也寸志氏とは、日常の中で多くの知的好奇心を刺激する議論を交わすことができました。彼らとの議論は、筆者にとって何にも代えがたい貴重な時間でした。心より感謝いたします。

末筆になりますが、後期博士課程への進学を快諾してくれた両親に、そして研究者として道ゆくことを支えてくれた婚約者・清水智恵子にこの場をかりて感謝申し上げます。

2007年3月

安倍 満

参考文献

- [1] IEEE Transactions on Vehicular Technology,
<http://transactions.vtsociety.org/>,
<http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?puNumber=25>.
- [2] IEEE Transactions on Intelligent Transportation Systems,
<http://www.ewh.ieee.org/tc/its/trans.html>,
<http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?puNumber=6979>.
- [3] ITS World Congress,
<http://www.itsworldcongress.com/>.
- [4] IEEE Intelligent Vehicle Symposium 2006,
<http://www.cvl.iis.u-tokyo.ac.jp/iv2006/>.
- [5] Y.M. Liang, H.R. Tyan, S.L. Chang, H.Y.M. Liao, and S.W. Chen, "Video stabilization for a camcorder mounted on a moving vehicle," vol.53, no.6, pp.1636–1648, Nov. 2004.
- [6] C. Demonceaux, A. Potelle, and D. Kachi-Akkouche, "Obstacle detection in a road scene based on motion analysis," vol.53, no.6, pp.1649–1656, Nov. 2004.
- [7] X. Liu, and K. Fujimura, "Pedestrian detection using stereo night vision," vol.53, no.6, pp.1657–1665, Nov. 2004.
- [8] M. Bertozzi, A. Broggi, A. Fascioli, T. Graf, and M. Meinecke, "Pedestrian detection for driver assistance using multiresolution infrared vision," vol.53, no.6, pp.1666–1678, Nov. 2004.
- [9] Y. Fang, K. Yamada, Y. Ninomiya, B. k. Horn, and I. Masaki, "A shape-independent method for pedestrian detection with far-infrared images," vol.53, no.6, pp.1679–1697, Nov. 2004.
- [10] M.M. Trivedi, S.Y. Cheng, E.M.C. Childers, and S.J. Krotosky, "Occupant posture analysis with stereo and thermal infrared video: Algorithms and experimental evaluation," vol.53, no.6, pp.1698–1712, Nov. 2004.

- [11] S. Todorovic, and M.C. Nechyba, “A vision system for intelligent mission profiles of micro air vehicles,” vol.53, no.6, pp.1713–1725, Nov. 2004.
- [12] IEEE Transactions on Pattern Analysis and Machine Intelligence,
<http://www.computer.org/tpami/>,
<http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=34>.
- [13] Z. Sun, G. Bebis, and R. Miller, “On-road vehicle detection: A review,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.28, no.5, pp.694–711, May 2005.
- [14] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, “Traffic monitoring and accident detection at intersections,” vol.1, no.2, pp.108–118, June 2000.
- [15] 市原栄太郎, 高尾広行, 大田友一, “Naviview: 仮想車載カメラ映像による運転者の視覚支援,” 信学論 D-2, vol.J82-D-2, no.10, pp.1816–1825, Oct. 1999.
- [16] F. Taya, Y. Kameda, and Y. Ohta, “Naviview: Virtual slope visualization of blind area at an intersection,” Proceedings of 12th World Congress on Intelligent Transport Systems, San Francisco, Nov. 2005.
- [17] T. Yasumasu, and S. Ozawa, “Detection of dangerous vehicles from rear scene,” IEEJ Trans. Electronics, Information and Systems, vol.125, no.4, pp.570–575, Apr. 2005.
- [18] A. Bensrhair, M. Bertozzi, A. Broggi, P. Miche, S. Mousset, and G.Moulminet, “A cooperative approach to vision-based vehicle detection,” Proceedings of Intelligent Transportation Systems, pp.209–214, 2001.
- [19] 岡田隆三, 古川賢司, 谷口恭弘, 小野口一則, “複比と消失線に基づく車載単眼障害物検出,” 信学論 D-2, vol.J87-D-2, no.12, pp.2165–2175, Dec. 2004.
- [20] A. Seki, and M. Okutomi, “Robust obstacle detection in general road environment based on road extraction and pose estimation,” IEEE Intelligent Vehicle Symposium, pp.437–444, June 2006.
- [21] A. Hashizume, S. Ozawa, and H. Yanagawa, “An approach to detect vacant parking space in a parallel parking area,” Proceedings of 5th European Congress on Intelligent Transport Systems, June 2005.
- [22] K.Fintzel, R.Bendahan, C.Vestri, and S.Bougnoux, “3d parking assistant system,” Proceedings of 11th World Congress on Intelligent Transport Systems, Oct. 2004.
- [23] 小越咲子, 小越康宏, 木村春彦, 広瀬貞樹, “CCD カメラ画像に基づいた自動車の車庫入れの自動化,” 信学論 A, vol.J87-A, no.2, pp.253–264, Feb. 2004.

- [24] Y. Suzuki, K. Yamamoto, K. Kato, M. Andoh, and S. Kojima, "Skin detection by near infrared multi-band for driver support system," Proceedings of Asian Conference on Computer Vision (ACCV), Jan. 2006.
- [25] S. Baker, I. Matthews, J. Xiao, R. Gross, T. Ishikawa, and T. Kanade, "Real-time non-rigid driver head tracking for driver mental state estimation," CMU Technical Report CMU-RI-TR-04-10, 2004.
- [26] 内村圭一, 松島宏典, "オクルージョンを考慮した交通流計測," 電学論 C, vol.122-C, no.12, pp.2120–2127, Dec. 2002.
- [27] S. C., and G. W.E.L., "Adaptive background mixture models for real-time tracking," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.246–252, 1999.
- [28] H. Veeraraghavan, O. Masoud, and N. Papanikolopoulos, "Computer vision algorithms for intersection monitoring," vol.4, no.2, pp.78–89, June 2003.
- [29] 谷口博康, 関明伸, 古澤春樹, 黒田伸一, 池端重樹, "時空間画像を用いた動画画像処理手法の提案——DTT 法——," 信学論 D-2, vol.J77-D-II, no.10, pp.2019–2026, Oct. 1994.
- [30] D. Koler, J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning," Institute of Transportation Studies : California Partners for Advanced Transit and Highways (PATH), 1994.
- [31] 久保山英生, 小沢慎治, "連続画像からのトンネル内における重交通流計測," 信学論 D-2, vol.J85-D-2, no.2, pp.210–218, Feb. 2002.
- [32] X. Song, and R. Nevatia, "A model-based vehicle segmentation method for tracking," Proceedings of International Conference on Computer Vision (ICCV), pp.1124–1131, 2005.
- [33] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol.2, 2000.
- [34] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.25, no.5, 2003.
- [35] 小関亮介, 箕浦良文, 藤吉弘亘, 秋田時彦, 柿並俊明, "協調的な複数の mean-shift トラッカによる後方車両追跡," 画像の認識・理解シンポジウム (MIRU2005), pp.419–426, July 2005.
- [36] M. Isard, and A. Blake, "Condensation – conditional density propagation for visual tracking," Int. J. Computer Vision, vol.29, no.1, pp.5–28, 1998.

- [37] The Condensation Algorithm Home Page,
http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/ISARD1/condensation.html.
- [38] C. Idler, R. Schweiger, D. Paulus, M. Mählich, and W. Ritter, “Realtime vision based multi-target-tracking with particle filters in automotive applications,” IV2006, IEEE Intelligent Vehicles Symposium, pp.188- 193, Tokyo, 2006.
- [39] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, “A real-time computer vision system for measuring traffic parameters,” Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.495–501, June 1997.
- [40] 上条俊介, 松下康之, 池内克史, 坂内正夫, “時空間 markov random field モデルによる隠れにロバストなトラッキングアルゴリズム,” 信学論 D-2, vol.J83-D-2, no.12, pp.2597–2609, Dec. 2000.
- [41] M. Kagesawa, S. Ueno, K. Ikeuchi, and H. Kashiwagi, “Recognizing vehicles in infrared images using IMAF parallel vision board,” vol.2, no.1, pp.10–17, March 2001.
- [42] 諏訪正樹, “ステレオビジョンによる交通流の検知,” 画像センシングシンポジウム講演論文集, vol.11, pp.397–400, June 2005.
- [43] C. Tomasi, and T. Kanade, “Shape and motion from image streams: a factorization method-part3: Detection and tracking of point features,” CMU Technical Report, 1991.
- [44] J. Shi, and J. Malik, “Motion segmentation and tracking using normalized cuts,” Proceedings of International Conference on Computer Vision (ICCV), 1998.
- [45] 永持仁, “グラフの最小分割問題に対するアルゴリズム,” 信学論 D-1, vol.J86-D-I, no.2, pp.53–68, Feb. 2003.
- [46] J. Shi, and J. Malik, “Normalized cuts and image segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.22, no.8, pp.888–905, Aug. 2000.
- [47] 柳浦睦憲, 茨木俊秀, “組み合わせ最適化問題に対するメタ戦略について,” 信学論 D-1, vol.J83-D-I, no.1, pp.3–25, Jan. 2004.
- [48] 藤沢克樹, 久保幹雄, 森戸晋, “Tabu search のグラフ分割問題への適用と実験的解析,” 電学論 (C), vol.114-C, no.4, pp.430–437, Apr. 1994.
- [49] J. Shi, and C. Tomasi, “Good features to track,” Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.593–600, June 1994.
- [50] B.D. Lucas, and T. Kanade, “An iterative image registration technique with an

- application to stereo vision,” International Joint Conference on Artificial Intelligence, pp.674–679, 1981.
- [51] 池田光二, 吉田昌司, 中島啓介, 浜田長晴, 依田晴夫, “正規化相関演算の単調関数化による高速テンプレートマッチング,” 信学論 (D-II), vol.J83–D–II, no.9, pp.1861–1869, Sept. 2000.
- [52] S.X. Yu, and J. Shi, “Multiclass spectral clustering,” Proceedings of International Conference on Computer Vision (ICCV), pp.313–319, Nice, France, Oct. 2003.
- [53] S.X. Yu, and J. Shi, “Segmentation given partial grouping constraints,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.26, no.2, pp.173–183, Feb. 2004.
- [54] S.X. Yu, and J. Shi, “Grouping with bias,” CMU Technical Report, CMU-RI-TR-01-22, July 2001.
- [55] X. EP, and K. RM, “CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts,” Bioinformatics, vol.17, 2001.
- [56] 岡部亜梨子, 小沢慎治, “距離画像と濃淡画像を統合した侵入者検出システム,” Intruder Detection System Integrating Texture and Distance Images, vol.IP-04, no.10-25, pp.51–54, Sept. 2004.
- [57] C. Harris, and M. Stephens, “A combined corner and edge detector,” Proc. 4th Alvey Vision Conf., pp.pp.147-151, 1988.
- [58] J. Shi, and J. Malik, “Normalized cuts and image segmentation,” Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1997.