イベントカメラを用いたオプティカルフロー推定 ~スパース・非同期推定を目指して~

2024年2月

長田 惇

# 学位論文 博士(工学)

イベントカメラを用いたオプティカルフロー推定 ~スパース・非同期推定を目指して~

2024年2月

慶應義塾大学大学院理工学研究科

長田 惇

#### 概要

オプティカルフロー推定は,移動物体の認識において重要な技術である.また,自動運転の衝突 回避のような緊急性の高いシステムでは,高レートでの推定が求められる.従来のフレームベース のカメラでは,画像間で輝度一貫性を測ることで推定を行うが,時間分解能の低さから画素同士の 対応づけは本質的に困難である.近年では,深層学習など高度なモデルが提案されているが,出 カレートを上げようとすると画像の枚数が多くなり,全画素を処理することは計算負荷が大きく なる.

イベントカメラは画素ごとに非同期的に輝度の変化を出力するカメラであり,高い時間分解能を 持ち,オプティカルフローの推定に適していると言える.その特性ゆえに,時空間イベントを局所 的に平面フィッティングすることで簡単にフローが得られるが,エッジに並行な成分に不定性が残 り,完全なオプティカルフローではない.また,イベントを密に処理してフローを推定する手法が 存在するが,古いイベントに対して再計算を要し,効率的な処理とは言えない.そこで本研究で は,スパースかつインクリメンタルな処理によってイベントデータから完全なオプティカルフロー を推定する手法を提案する.

第1章に、本研究の背景と目的を述べた。第2章では、従来の画像ベースのオプティカルフロー 推定技術について述べ,その原理的課題を述べた.次に,イベントカメラを導入し,既存手法につ いて概説した.第3章では、イベントデータから輝度を復元することで、従来の輝度一貫性の仮定 を利用する手法に着目した.高い時間分解能で輝度を復元しフローを推定可能であるが,データが スパースなため平坦な推定になってしまう、そこで、車載シーンの運動モデルを活用した正則化を 提案し、精度向上を確認した.第4章では、第3章の復元した輝度で一致度を測る仕組みは、差分 のみを処理するイベントの特性を活かしきれていないと考え、イベントの時間で直接対応づけを測 る損失関数を提案した.これにより、提案のロス関数を滑らかさの正則化とともに最適化すること で、輝度復元を介さずにオプティカルフローを推定可能とした。第5章では、非同期イベントデー タを画像と同様に扱い,全画素で同期的に最適化を行っていた第3,4章に対して,スパース,非 同期的, インクリメンタルにオプティカルフローを推定する手法を提案した. イベントデータから 局所的に得られる動きの情報を用いて、オプティカルフローの分布を細長い2次元のガウス分布で 表現し、滑らかさの事前情報を合わせたファクターグラフを構成した.信念伝搬法によってイベン トが来るごとに各画素の分布を非同期的に更新して周辺化することで、局所的な演算のみからオプ ティカルフローを推定することを可能とした.スパースな手法であるため並列化が可能で,マルチ コア CPU でリアルタイム処理を達成した.本手法は計算結果を任意のタイミングで取り出すこと

 $\mathbf{2}$ 

ができ,出力レートを自由に設定可能である.第6章では,結論として本研究で得られた成果を要約した.

# 目次

1	序論	1
1.1	研究背景と目的	1
1.2	用語	7
1.3	本論文の構成....................................	7
1.4	本章のまとめ....................................	8
2	予備知識	10
2.1	画像ベースオプティカルフロー	10
2.2	イベントベースオプティカルフロー	14
2.3	本章のまとめ	27
3	拡張焦点を用いた正則化	28
3.1	導入	28
3.2	拡張焦点 (FOE: focus of expansion)	29
3.3	FOE を用いた車載カメラ運動解析	29
3.4	拡張焦点を用いた正則化..................................	31
3.5	実験	33
3.6	本章のまとめ	37
4	Surface matching	39
4.1	導入	39
4.2	Surface matching loss	40
4.3	実験	42
4.4	本章のまとめ....................................	49

5

# 信念伝播法による非同期イベントオプティカルフロー推定

	TEGBP: Tangentially Elongated Gaussian Belief Propagation for Event-based Incre-	
	mental Optical Flow Estimation	51
5.1	導入	51
5.2	問題設定	53
5.3	Tangentially Elongated Gaussian (TEG)	55
5.4	事前情報と TEG の同時分布の周辺化	56
5.5	ファクターグラフ	57
5.6	信念伝搬法による周辺化	58
5.7	アルゴリズム	60
5.8	実験	60
5.9	本章のまとめ	69
6	結論	71
6 6.1	<b>結論</b> まとめ	71 71
6 6.1 6.2	結論         まとめ	<ul><li>71</li><li>71</li><li>72</li></ul>
6 6.1 6.2 6.3	結論         まとめ	<ul><li>71</li><li>71</li><li>72</li><li>73</li></ul>
6 6.1 6.2 6.3 付録 A	結論         まとめ         将来展望         ジョン         終わりに         FOE を用いた車載カメラの運動解析	<ul> <li>71</li> <li>71</li> <li>72</li> <li>73</li> <li>86</li> </ul>
6 6.1 6.2 6.3 <b>付録</b> A A.1	結論         まとめ         将来展望         ジネの運動解析         方メラ座標系での対応点	<ul> <li>71</li> <li>71</li> <li>72</li> <li>73</li> <li>86</li> <li>86</li> </ul>
6 6.1 6.2 6.3 <b>付録 A</b> A.1 A.2	結論         まとめ	<ul> <li>71</li> <li>71</li> <li>72</li> <li>73</li> <li>86</li> <li>86</li> <li>87</li> </ul>
6 6.1 6.2 6.3 <b>付録</b> A A.1 A.2 <b>付録</b> B	結論         まとめ	<ul> <li>71</li> <li>71</li> <li>72</li> <li>73</li> <li>86</li> <li>86</li> <li>87</li> <li>89</li> </ul>
6 6.1 6.2 6.3 <b>付録 A</b> A.1 A.2 <b>付録 B</b> B.1	結論         まとめ	<ul> <li>71</li> <li>71</li> <li>72</li> <li>73</li> <li>86</li> <li>86</li> <li>87</li> <li>89</li> <li>89</li> </ul>
6 6.1 6.2 6.3 <b>付録 A</b> A.1 A.2 <b>付録 B</b> B.1 B.2	結論         まとめ	<ul> <li>71</li> <li>71</li> <li>72</li> <li>73</li> <li>86</li> <li>86</li> <li>87</li> <li>89</li> <li>89</li> <li>89</li> </ul>

# 表目次

3.1	実験結果	35
4.1	オプティカルフロー推定の定量評価. 評価指標は AEE( $ ext{pix}/\Delta t$ ) を用いた	47
5.1	Basic experimental setup.	62
5.2	MVSEC における定量評価. オプティカルフローは真値が得られている 20Hz で評価	
	した	64
5.3	DSEC における定量評価.オプティカルフローは真値が得られている 10Hz で評価し	
	た	66
5.4	MVSEC における定量評価. オプティカルフローは真値が得られている 20Hz で評価	
	した. 比較はサーフェスマッチング (SM) と TEGBP で比較を行った	69

# 図目次

1.1	オプティカルフロー.一般的に 2 枚の画像間で計算され,各画素 2 次元のベクトルで	
	表現される.右図のように色相で向きを, 明度で大きさを表している.画像は, [Teed	
	and Deng, 2020] のものを編集して用いた. .	2
1.2	イベントデータの例.輝度の変化が発生した時の,時間 $t$ ,画素位置 $(x,y)$ ,極性 $p$	
	の 4 次元タプルの時系列データとなっている.xyt の 3 次元空間上のそれぞれの点	
	がイベントを表している.赤と黄緑はそれぞれ正負の極性を表している.データは	
	[Mueggler et al., 2017b] のものを使用した.................	3
1.3	標準フレームカメラ (上) とイベントカメラ (下) の比較. 図は, [Mueggler et al.,	
	2014] のものを一部改変して作成した. .	4
1.4	窓問題 (aperture problem). 図中央のような 2 つのエッジで構成されるパターンが	
	右向き (マゼンタの矢印) に運動しているシーンを想定する.左図のように一つの	
	エッジのみでは,輝度勾配方向の成分 (黄緑の矢印) しか分からず,対応点が一意に定	
	まらない.右図のように2つのエッジがあれば対応点が一意に定まる.......	5
1.5	本論文の構成	8
2.1	オプティカルフローの線形拘束式.拘束線 (緑) へのベクトルのうち,輝度勾配方向	
	であり,距離が最小となるベクトル (水色) はノーマルフローと呼ばれる. .....	11
2.2	タイムサーフェス.図は [Benosman et al., 2014] 中のものを編集している. .	15
2.3	実データによるタイムサーフェス. データは [Mueggler et al., 2017b] のものを使用	
	した	17
2.4	ある範囲の空間でノーマルフローの平均を取ることで窓問題の影響を軽減する手法	
	[Akolkar et al., 2022]	18

2.5	輝度復元を介したオプティカルフロー推定 [Bardow et al., 2016] のためのスライディ	
	ングウィンドウ.図は [Bardow et al., 2016] のものに変更を加えた.......	19
2.6	輝度復元を用いた手法 [Bardow et al., 2016] のコスト関数	20
2.7	コントラスト最大化 [Gallego et al., 2019]. イベントを運動パラメータでワープし	
	て作成した画像のコントラストが最大になるようにパラメータを推定する. 図は	
	[Gallego et al., 2019] のものに変更を加えた	21
2.8	教師あり学習手法の代表的手法である E-RAFT[Gehrig et al., 2021b] のネットワー	
	ク構造. 図は [Gehrig et al., 2021b] のものをそのまま用いた	23
2.9	半教師あり学習手法の代表的手法である EV-FlowNet[Zhu et al., 2018a] のネット	
	ワーク構造.図は [Zhu et al., 2018a] のものをそのまま用いた. .	24
2.10	教師なし学習手法の代表的手法である FireFlowNet[Paredes-Vallés and de Croon,	
	2021] の学習方法.図は [Paredes-Vallés and de Croon, 2021] のものをそのまま用い	
	た	25
2.11	SNN の代表的手法である Spike-FlowNet[Lee et al., 2020a] の構造.図は [Lee et al.,	
	2020a] のものをそのまま用いた.................................	26
3.1	FOE の定義. <i>X</i> , <i>Y</i> , <i>Z</i> はカメラ座標系, <i>x</i> , <i>y</i> は画像座標系を表す. .	30
3.2	車載カメラの運動モデル.................................	32
3.3	FOE の性質. オプティカルフロー $\mathbf{v}$ から回転による成分 $\mathbf{v}^{\omega}$ を除く ( $\mathbf{v}' = \mathbf{v} - \mathbf{v}^{\omega}$ )	
	と,FOE から放射状になるという性質を持っている. ..............	33
3.4	FOE を利用した正則化.回転による成分を除いたフローが FOE に従うように修正	
	する	33
3.5	FOE を用いた正則化手法の全体概要	33
3.6	推定されたオプティカルフローの例..................................	36

4.6	MVSEC の outdoor シーンにおける定性評価.左から,画像,イベント (黄緑と赤
	が正負それぞれのイベントの極性,黄色は両方発生していることを示している),
	Reconstruction, Variance, Surface Matching, Ground Truth のオプティカルフ
	ローを示している. オプティカルフローは, 右下のように, 色相で向き, 明るさで大
	きさを表している.行方向はサンプルを示している. ......... 48

4.7	HACD データにおける定性評価.左からイベント (黄緑と赤が正負それぞれのイベン
	トの極性,黄色は両方発生していることを示している),+ のタイムサーフェス,-の
	タイムサーフェス, Reconstruction, Variance, Surface Matching, Ground Truth
	のオプティカルフローを示している.オプティカルフローは,右下のように,色相で
	向き,明るさで大きさを表している. . . . . . . . . . .
4.8	MVSEC の outdoor シーンにおける pitch レートと AEE
4.9	MVSEC の outdoor データの振動が大きいシーン

- 5.1 バッチ処理とインクリメンタルな処理の模式的比較. ..... 52
- 5.3 インクリメンタルなオプティカルフロー推定.あるノードiの新たな時刻のノーマルフロー v<sup>⊥</sup><sub>i,t+1</sub>を入力とし、メッセージ交換によって周辺ノードのみの分布をインクリメンタルに更新する.出力は任意のタイミングで取り出すことができ、即時に対応するオプティカルフロー v<sub>i,t+1</sub>を出力することも可能である......54

5.4	TEG によるオプティカルフローのモデリング.ノーマルフロー v <sup>⊥</sup> が得られたとき,	
	オプティカルフロー v はノーマルフローを通り,ノーマルフローの向き (輝度勾配方	
	向) と垂直な線上のどこかであることがわかる (左図).そこで,その不確実性を,平	
	均がノーマルフロー <b>v</b> <sup>⊥</sup> ,接線方向に大きい分散を持つ 2 次元のガウス分布 (TEG)	
	で表現した (右図)	54
5.5	接線方向のフローの分布...................................	55
5.6	TEG のパラメータ $\sigma_{ m t}$ を変化させた時の,それぞれの TEG(赤,緑,青) と周辺分布	
	(マゼンタ). 黒が真のオプティカルフロー, マゼンタが推定したオプティカルフロー	
	を表す	56
5.7	ファクターグラフ.トポロジーはアクティブノードセット $\mathcal{A}_t$ に応じてダイナミック	
	に変動する.最近イベントが発生した画素のノードをアクティブとしている.同じ画	
	素のアクティブな変数ノード (青) と観測ノード (黄緑) は繋がっている.また,近所	
	のアクティブな変数ノード同士は事前情報ファクターノードを通して繋がっている.	57
5.8	一連のメッセージ送信スキーム................................	59
5.9	ESIM-bricks による実験結果.左から,ノーマルフロー,ARMS,TEGBP による推	
	定オプティカルフローを結果を示す. 正解のオプティカルフローは矢印によって示し	
	τδ	63
5.10	TEG の長径の標準偏差 σ <sub>t</sub> を変化させた時の ESIM-bricks の実験結果	63
5.11	DVS-stripes による実験結果.左から,ノーマルフロー,ARMS,TEGBP による推	
	定オプティカルフローを結果を示す.赤い線は,2 つの速度の正解のオプティカルフ	
	ローを示している	64
5.12	MVSEC における定性評価	65
5.13	DSEC における定性評価	66

5.14	メッセージ交換スキーム.黄色がイベントが発生した画素を示す.このノードから 8	
	近傍で <i>L</i> 種類の遠さのノードにメッセージを送る.この時, <i>l</i> 番目は 2 <sup><i>l</i>-1</sup> 個離れた	
	ノードとした..................................	67
5.15	コア数に応じた相対的な実行時間 (イベントデータの時間に対する実行時間).10 <sup>0</sup> が	
	リアルタイムを表している	68
6.1	CPU と GPU, IPU の違い	73
付録 B.	1変数ノード $x$ に $f_s$ を含むファクターノードの結合	90
付録 B.	2ファクター $f_s$ と変数 $x, M$ 個の他の変数の結合	90

# 1 序論

本章ではまず,従来のフレームベースのカメラを用いたコンピュータビジョン分野におけるオプ ティカルフロー推定の課題を挙げる.それが,フレームベースカメラを用いたセンシングの本質的な 課題であり,センシングデバイスを含めたパラダイムシフトが必要であることを述べる.そして,従 来のカメラと動きを効率的に捉えることができる生体の視覚システムの違いを明確にし,網膜を模倣 して作られたイベントベースカメラ (イベントカメラ) について導入する.イベントカメラのデータ 形式や優位性について述べた後,イベントカメラの持つスパース性と一般的なオプティカルフロー推 定における課題である窓問題 (aperture problem) について述べる.そして,研究目的や提案手法のア プローチ,本論文の構成について述べる.

## 1.1 研究背景と目的

オプティカルフローは、3 次元中の対象の動きを画像平面に投影したものであり、画素ごとの 2 次 元ベクトルで表現される.一般的には、図 1.1 のように 2 時刻の画像間の移動ベクトルを表す.オプ ティカルフローは、物体の運動情報を必要とする自動車や監視・見守り、人流解析、工場自動化など多 岐に渡って応用される.特に先進運転アシストシステムでは、前方の障害物や歩行者との衝突回避の ために、前方物体の運動情報をリアルタイムに把握することが重要であり、低レイテンシなオプティ カルフロー推定が求められる.

オプティカルフローは、古典的には2枚の連続したフレーム画像から輝度の一貫性の仮定を用いて、 画素同士の対応づけを行うことで推定される [Lucas and Kanade, 1981]. 輝度一貫性の仮定は、画像 中の物体が異なる時刻で同じ輝度を持つという仮定であり、照明条件が変動する実際の画像では成り 立たない場合がある.また、オプティカルフローは2枚間の移動量が1ピクセル以内と十分に小さい ものとして定式化される.しかし、フレームレートに対して移動量が大きいときにそのような仮定が 成り立たなくなる.さらに、物体の動きが早く低照度環境の時、ブレが発生することで、輝度での対

1



図 1.1 オプティカルフロー. 一般的に 2 枚の画像間で計算され,各画素 2 次元のベクトルで表現 される. 右図のように色相で向きを,明度で大きさを表している. 画像は, [Teed and Deng, 2020] のものを編集して用いた.

応づけが正確に測れない.最近では,畳み込みニューラルネットワーク (CNN: Convolutional neural network)を用いて輝度情報よりも高次元の特徴を抽出することで,よりロバストに対応づけを測る手 法が提案されている [Dosovitskiy et al., 2015; Teed and Deng, 2020]. しかし,複雑なネットワーク を用いるため,計算量が大きくなるという課題がある.

元より,画像は全ピクセルの絶対的な輝度値を一定時間ごとに全て記録する形式である.よって, 照明が一定で画像中で動きが発生していない場合,輝度の変化は発生せず,同じ輝度値が記録され続 ける.この場合,変化がない箇所ではオプティカルフローを推定する必要がないが,全画素について 処理が行われ無駄な計算が発生する.この問題は,高レートでオプティカルフローを得たい場合に, より顕著に現れる.高速カメラを使用すれば高レートに画像を取得することができるが,レートが上 がるにつれてデータの冗長性や処理負荷が大きくなる.

このように, 従来のフレームベースの画像認識で行われる, 時間的に離散な 2 枚の画像を入力とした 密なマッチングによるオプティカルフロー推定には本質的な課題がある.一方, 生体が持つ視覚シス テムにはフレームという概念はなく, 時間的に連続に視覚的な輝度の変化を捉え, 離散的なスパイクで 非同期的に伝えていると言われている [Hubel and Wiesel, 1959; Orchard and Etienne-Cummings, 2014]. このような生体の視覚システムを模倣したセンサであるイベントカメラ [Lichtsteiner et al., 2008; Posch et al., 2011; Brandli et al., 2014; Son et al., 2017] が近年注目されている [Gallego et al., 2019; Chen et al., 2020]. イベントカメラは, 全画素で同時に輝度の絶対値を出力する従来の



図 1.2 イベントデータの例. 輝度の変化が発生した時の,時間 t,画素位置 (x, y),極性 p の 4 次 元タプルの時系列データとなっている. xyt の 3 次元空間上のそれぞれの点がイベントを表してい る.赤と黄緑はそれぞれ正負の極性を表している.データは [Mueggler et al., 2017b] のものを使 用した.

フレームベースカメラとは異なり,ある画素で閾値以上の輝度の変化があった時に,その時間と画素 位置,明暗のどちらかを示す二値の値 (極性)を出力するカメラである.図 1.2 は,イベントデータの 例を表している.イベントデータは,右図のように N×4 の形の時系列データである.ここで,N は イベント数を表す.各イベントの時間 t と画素位置 (x,y)の3 つの値を,3 次元空間中に表示すると 左図のようになる.

このように、イベントデータは画素ごとに異なるタイミングのデータであり、このことを「非同期」 と呼ぶ.また、輝度が変化しない画素ではデータが出力されないため、変化のあった一部の画素のデー タしか得られない.このように、部分的な画素のみにデータがあること、または、部分的な画素を処 理することを「スパース」と呼ぶ.これらに対して、画像のように全画素で同時にデータを出力する ことを「同期」と呼び、全画素のデータが存在する、または、全画素を処理する様子を「密」または 「デンス」と呼ぶ.

このような仕組みを持つイベントカメラは,フレームベースカメラと比較して,低消費電力・高ダ イナミックレンジ (140dB)・高時間分解能 (マイクロ秒単位) などの特徴を持つ.画像中の物体の運 動に伴って発生する輝度の変化に反応するため,イベントカメラは物体の運動を捉えるのに適してい

 $\mathbf{3}$ 



図 1.3 標準フレームカメラ (上) とイベントカメラ (下) の比較. 図は, [Mueggler et al., 2014] の ものを一部改変して作成した.

ると言える. 図 1.3 は,標準的なフレームカメラとイベントカメラを比較したものである. 従来のフ レームカメラは,一定の時間間隔で画像全体の輝度を出力するのに対して,イベントカメラは,輝度 の変化のみに反応し,高い時間分解能でデータを出力することができている. 後者のデータは,輝度 パターンの運動に反応して得られていることから,動きが捉えやすいことが分かる.

しかし、イベントデータは非同期的な輝度変化であり、従来のフレーム間での輝度一貫性の仮定を そのまま用いることができない. そのため、イベントデータを用いたオプティカルフロー推定では、 新たな手法で画素同士の対応づけを行う必要がある. イベントカメラは高い時間解像度で輝度変化を 出力することから、局所的な範囲でイベントの対応づけをとることで運動を推定可能である. 例えば、 局所的な範囲で速度が一定だと仮定して時空間の3次元イベント点群を平面にフィッティングするこ とで運動の推定が可能である [Benosman et al., 2014]. また、局所的な範囲で速度が一定となる3点 のイベントを探すことで運動を推定する手法もある [Shiba et al., 2023].

しかし,1次元の局所的な運動情報からでは2次元の運動が一意に定まらないという窓問題 (aperture problem) が存在する [Nakayama and Silverman, 1988; Orchard and Etienne-Cummings, 2014].



図 1.4 窓問題 (aperture problem). 図中央のような 2 つのエッジで構成されるパターンが右向き (マゼンタの矢印) に運動しているシーンを想定する. 左図のように一つのエッジのみでは, 輝度勾 配方向の成分 (黄緑の矢印) しか分からず, 対応点が一意に定まらない. 右図のように 2 つのエッ ジがあれば対応点が一意に定まる.

例として図 1.4 のように 2 つのエッジで構成される輝度パターンが並進しているシーンを考える. 左 図の水色の円で示すように 1 つのエッジのみを見た時,輝度勾配方向の成分のみしか分からず,点線 で示す拘束線上で不定となり,2次元ベクトルを一意に決めることができない. それに対して,右図 の赤色の縁で示すように 2 つのエッジを含む範囲を見た時,拘束線の交点により,2次元の運動を一 意に決めることが可能である. このように,局所的な推定のみでは真の運動方向を持ったオプティカ ルフローを推定できず,輝度勾配成分のみしか推定することができない. このオプティカルフローの 輝度勾配方向成分を「ノーマルフロー」と呼ぶ. 窓問題を解決するには,異なる向きを持つ輝度勾配 が含まれるより広い範囲を考慮する必要があり,一般的には近傍画素で動きが一定,または滑らかに 変化するといった仮定を追加する必要がある.

これまで述べたように、イベントカメラは高時間解像度で輝度の変化を出力する仕組みから、局所 的な運動の推定に優位性を持つ.一方で、窓問題により、局所的な範囲の運動情報のみから完全なオ プティカルフローを推定することはできず、広い範囲を見る必要がある.そこで、本研究の主題をイ ベントデータを用いたオプティカルフロー推定とし、以下の2点について検討する.一つ目は、どの ようにイベントデータから局所的な画素同士の対応づけを測りつつ、大域的に整合の取れたオプティ カルフローを推定するかということだ.もう一つは、どのようにスパース/非同期的な処理のままオプ ティカルフローを推定するかということだ.画像ベースの手法のように、広い範囲を同期的に見て滑 らかさの制約の下で最適化することで、大域的に整合の取れたオプティカルフローを推定することが 可能である.しかし、その範囲が大きいと、大部分を同時に処理することとなり、イベントデータの 非同期性、スパース性が損なわれてしまう.本研究では、最終的に、スパースで非同期な処理によっ て、大域的に整合の取れたオプティカルフローを推定する手法を提案する.

まず最初に、3章では、イベントから輝度を復元することで、従来の輝度一貫性の仮定を利用して 画素同士の対応づけを計算する手法 [Bardow et al., 2016] について着目した.ある大きい時間幅のイ ベント群を用いて、輝度とイベントの関係、輝度とオプティカルフローの関係、輝度とオプティカル フローの時空間的な滑らかさの正則化を含むコスト関数を最適化することで、輝度とオプティカルフ ローを同時に推定する.この手法では、滑らかさの正則化を加えてグローバルに最適化を行うことか ら、イベントデータから完全なオプティカルフローを推定することが可能である.しかし、スパース なイベントデータから輝度とオプティカルフローを復元するのは不良設定問題であり、滑らかさの正 則化だけでは不十分である.そこで、車載シーンに限定し、運動モデルを利用した車載特有の正則化 を提案し、性能向上を確認した.

しかし,高い時間分解能でスパースなセンシングを行ったにもかかわらず,複数時刻の密な画像の 復元を行うことは無駄な処理である.そこで,4章では、イベントの時間の差分を直接測ることで対 応づけを測る手法を提案する.異なる時刻の2つのイベント群から,各画素の最新イベントのみを扱 い,時間の差分を測るロス関数を提案した.このロス関数に滑らかさの正則化を加え、グローバルに 最適化することでオプティカルフローを推定する.イベントの高い時間分解能を活かしつつ,同期的 な最適化で効率的に推定が可能である.

しかし,これらの手法でより高レートに出力をしようとした際に,古いイベントに対して再計算が 生じてしまい,計算量が増大する.そこで,5章では,時系列で得られるイベントデータに対して,ス パースに扱いつつ,前の計算結果を利用して漸進的 (インクリメンタル) にオプティカルフローを推定 する手法を目指す.局所的に推定した動きの情報からオプティカルフローの分布を2次元ガウス分布 で表現し,その情報を周囲に伝播させることでオプティカルフローを推定する.各画素をノードとし, 画素間に事前情報 (滑らかさなど) ファクターを設定したファクターグラフを,信念伝搬法によって 各画素について周辺化することでオプティカルフローを推定する.信念伝搬法は,局所的な周辺化を メッセージ送信という形で再帰的に繰り返すことで,グラフ全体の周辺事後分布を計算するアルゴリ ズムである.イベントが発生したタイミングで,近傍のイベントが最近発生した画素のみでメッセー ジ送信を行うことで,スパースかつ非同期な推定でオプティカルフローの推定を可能とした.各ノー ドではガウス分布によって信念を持ち,インクリメンタルに更新し,任意のタイミングで平均を取り 出すことで出力とする.並列化も可能で,CPU のマルチコアを用いて実装しリアルタイム処理を実現 した.

### 1.2 用語

本論文では、以下に示すように用語を使用した.

同期 画像のように全画素が同じ時間であること

非同期 イベントデータのように画素ごとが異なる時間であること

密,デンス 画像全体,全画素

**スパース** データが存在するのが画像全体ではなく部分的であり, 疎であること

インクリメンタル 前の計算結果を再利用し、それを更新する形で漸進的に計算すること

ノーマルフロー オプティカルフローの輝度勾配方向の成分

## 1.3 本論文の構成

図 1.5 に本論文の構成について示す. 1 章では,本研究の背景について述べた. 2 章で,事前知識と して画像/イベントベースのオプティカルフロー推定について述べる. 3 章で,輝度復元を介すイベン トオプティカルフロー推定のための,拡張焦点 (FOE: Focus of expansion)を用いた正則化について 述べる. この主な内容は [Nagata et al., 2019a,b] を元にしている. 4 章で,イベントから輝度復元な



図 1.5 本論文の構成.

しで,イベントの時間で対応づけしてオプティカルフローを推定する手法について提案する.この主 な内容は [Nagata et al., 2021] を元にしている.5章で,非同期的な最適化によってオプティカルフ ローをインクリメンタルに推定する手法を提案する.この主な内容は [Nagata and Sekikawa, 2023] を元にしている.6章で,本論文の結びとする.

# 1.4 本章のまとめ

本章では研究背景と研究目的を説明した.従来のフレームベースカメラを用いたオプティカルフ ロー推定は,時間的に離散な2枚の画像から各画素の対応づけを行うことに本質的な難しさがある. 一方,生体の視覚システムは,フレームという概念はなく,時間的に連続的な情報を処理することで 効率的に動きを捉えている.生体の網膜の仕組みを模倣したイベントカメラは,輝度の変化のみを画 素ごとに非同期的に出力する仕組みを持ち,画像内の物体の運動を捉えるのに適していると言える.

本研究では、スパースなイベントデータから大域的に整合の取れた完全なオプティカルフローを推 定する問題設定に取り組んだ.復元した輝度によって対応づけを測る手法から検討を始め(3章)、イ ベントの時間を直接扱う手法を提案した(4章).しかし、いずれの手法も画像と同様の同期的で密な 推定であり、イベントのスパース性を活かしきれていない.最終的に、信念伝搬法を用いた手法を提 案し,局所的な運動の推定から非同期的な最適化で完全なオプティカルフローを推定することを可能 とした (5章).

# 2 予備知識

本章では、本研究のタスクであるオプティカルフロー推定について記述する.まずは、古くから研 究されている画像ベースのオプティカルフロー推定手法について述べる.その後、イベントカメラの 導入と、それを用いたオプティカルフロー推定手法について述べる.

### 2.1 画像ベースオプティカルフロー

#### 2.1.1 オプティカルフロー拘束式

画像上の輝度の変化と輝度パターンの運動の関係を導く. ある時刻 t の画像平面上の点 (x,y) の輝 度値を I(x,y,t) とする. 輝度パターンが移動した時,対応する点での輝度は一定であると仮定をする と,以下の式が得られる.

$$\frac{\mathrm{d}I}{\mathrm{d}t} = 0 \tag{2.1}$$

この式を、合成関数の偏微分における連鎖律を用いて以下のように展開する.

$$\frac{\partial I}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial I}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}y} + \frac{\partial I}{\partial t} = 0$$
(2.2)

ここで、オプティカルフローの x, y 方向成分をそれぞれ u = dx/dt, v = dy/dt とすると、以下のように簡単化される.

$$I_x u + I_y v + I_t = 0 (2.3)$$

ここで、下付き添え字は偏微分を表す. つまり、 $I_x$ はx方向の輝度勾配を表す. この式は、二つの未 知数u,vを持つオプティカルフローの線型方程式である. さらに、オプティカルフローと輝度勾配の ベクトルをそれぞれ $\mathbf{v} = (u,v)^{\top}, \nabla I = (I_x, I_y)^{\top}$ のように定義すると、以下のように表すことがで きる.

$$\nabla I \cdot \mathbf{v} = -I_t \tag{2.4}$$



図 2.1 オプティカルフローの線形拘束式. 拘束線 (緑) へのベクトルのうち,輝度勾配方向であり,距離が最小となるベクトル (水色) はノーマルフローと呼ばれる.

ここで、図 2.1 は、未知数 u, v の 2 次元の空間で (2.4) 式の拘束式をプロットしたものである.オ プティカルフロー (u, v) は、輝度勾配ベクトル  $(I_x, I_y)$  と垂直な線上のどこかに位置することが分か る.しかし、局所的な一つの拘束式のみでは、この線上のユニークな 1 点である完全なオプティカル フローを決定することはできない.この問題は、窓問題 (aperture problem) と呼ばれる [Nakayama and Silverman, 1988].

#### 2.1.2 ノーマルフロー

一つの拘束式からオプティカルフローを推定することはできないが,輝度勾配ベクトル ∇I の成分 を推定することはできる.これは,エッジに直交していることからノーマルフローと呼ばれている. つまり,図 2.1 における,原点から拘束式の直線までの距離が最短になるベクトルである.その大き さは、以下のように求められる.

$$|\mathbf{v}^{\perp}| = -\frac{I_t}{\sqrt{I_x^2 + I_y^2}} \tag{2.5}$$

よって、ノーマルフローは以下のように求められる.

$$\mathbf{v}^{\perp} = -\frac{I_t}{\sqrt{I_x^2 + I_y^2}} \frac{\nabla I}{\sqrt{I_x^2 + I_y^2}} = -\frac{I_t}{I_x^2 + I_y^2} \nabla I$$
(2.6)

### 2.1.3 空間的整合性

### Lucas-Kanade 法 [Lucas and Kanade, 1981]

輝度パターンのそれぞれの点が独立に動くと仮定すると、それらの動きを復元することはできない. 近傍の画素群 q<sub>1</sub>, q<sub>2</sub>, … q<sub>n</sub> が同じオプティカルフローを持つと仮定すると、一つの v に対する n 個の 拘束式が得られる.

÷

$$I_x(q_1) u + I_y(q_1) v = -I_t(q_1)$$
(2.7)

$$I_x(q_2) u + I_y(q_2) v = -I_t(q_2)$$
(2.8)

$$I_{x}(q_{n}) u + I_{y}(q_{n}) v = -I_{t}(q_{n})$$
(2.10)

これらの方程式を行列の形  $(A\mathbf{v} = \mathbf{b})$  で表すと以下のようになる.

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} u \\ v \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix}$$
(2.11)

このシステムは、2つの未知パラメータ *u*,*v* より多くの拘束式を持ち一般的には over-determined である. Lucas-Kanade 法は最小二乗法を用いて、以下のように解く.

$$\mathbf{v} = (A^{\top}A)^{-1}A^{\top}\mathbf{b} \tag{2.12}$$

#### 2.1.4 滑らかさ拘束

2.1.3 章の Lucas-Kanade 法では,輝度勾配がない平坦な部分では推定ができない.また,エッジな ど,依然として窓問題が残る部分で方程式が悪条件となり,推定が悪化する.また,領域を大きくと ると局所的に運動が一定という仮定が成り立たなくなる [Bruhn et al., 2005].ここで,隣り合う点が 滑らかに運動すると仮定し,正則化を加えて画像全体で最適化することでオプティカルフローを推定 する手法を紹介する.

#### Horn-Shunk 法 [Horn and Schunck, 1981]

オプティカルフロー拘束式の2乗に加え,滑らかさの正則化としてオプティカルフローの空間微分の 大きさの2乗をコスト関数に加える.

$$E_{HS}(\mathbf{v}) = \int_{\Omega} \left( \nabla I \cdot \mathbf{v} + I_t \right)^2 + \alpha^2 \left( \left| \nabla u \right|^2 + \left| \nabla v \right|^2 \right)$$
(2.13)

ここでは, Ω は画像空間, α は正則化の調整パラメータを示している. この最小化問題は, Euler-Lagrange 方程式で記述し, 繰り返し解くことで効率的に解を求めることができる.

#### TV-L1 法 [Zach et al., 2007; Sánchez Pérez et al., 2013]

Horn-Shunk 法で使用されているオプティカルフロー拘束式は, (2.4) 式で表されるもので, オプティ カルフローパラメータに対して線形になっている.一方,一般的な 2 枚の画像 *I*<sub>0</sub>, *I*<sub>1</sub> から非線形の定 式化で以下のように記述できる.

$$I_1(\mathbf{x} + \mathbf{v}) - I_0(\mathbf{x}) = 0 \tag{2.14}$$

左辺の第1項は非線形項であり, v に十分に近い v<sup>0</sup> の近傍でテイラー展開することで,以下のように 線形化することができる.

$$\rho(\mathbf{v}) = \nabla I_1(\mathbf{x} + \mathbf{v}^0) \cdot (\mathbf{v} - \mathbf{v}^0) + I_1(\mathbf{x} + \mathbf{v}^0) - I_0(\mathbf{x}) = 0$$
(2.15)

Horn-Shunk 法で使用される2乗のコストをL1ノルムに変更することで、フローの不連続性を許容す

る形に改良することができる [Zach et al., 2007; Sánchez Pérez et al., 2013; Chambolle and Pock, 2011]. (2.15) 式にフローの空間微分の L1 ノルムによる滑らかさの正則化を加えて,以下のようにコ スト関数を定義する.

$$E_{TVL1}(\mathbf{v}) = \int_{\Omega} |\nabla u|_1 + |\nabla v|_1 + \lambda |\rho(\mathbf{v})|_1$$
(2.16)

これらの古典的な画像ベースのオプティカルフローは,輝度の一貫性の仮定や画像間での運動が十 分小さい (<1 画素) という仮定を用いている.しかし,30fps などの一定の時間で撮影される画像で は,高速に動く物体やカメラの近くで動く物体において,この仮定が成り立たない場合がある.その 場合,解像度を落として複数解像度の画像のピラミッドを作成し,粗い階層から細かい階層へと順に 最適化することで対応される [Meinhardt-Llopis et al., 2013].

最近では,深層畳み込みニューラルネットワークを用いて,複雑な特徴抽出と画素同士の対応づけを 複数解像度で行う手法が提案されている [Dosovitskiy et al., 2015; Sun et al., 2018; Teed and Deng, 2020]. GPU による計算力の向上とモデルの高度化によって,精度の高いオプティカルフロー推定が 可能になってきている.しかし,依然としてロボットなどリソースが限られた中での高速なオプティ カルフロー推定が求められている.特に,ドローンなど高速に制御が必要なアプリケーションにおい て高レートの出力を要求した時,短い時間間隔で画像全体の処理をする必要がある画像ペースのオプ ティカルフロー推定は計算コストが高くなる.そのため,画像という入力形式にとらわれず,センシ ングを含めたシステム全体を考える必要がある.

## 2.2 イベントベースオプティカルフロー

画像中の物体が運動する時に輝度の変化が発生することから,高時間分解能で輝度の変化を記録す ることが可能であるイベントカメラは,オプティカルフロー推定に適していると言える.輝度の変化 のみを捉える仕組みによって,運動の情報を効率的に取得することができる.しかし,イベントデー タは,時間的に離散な画像データと形式が異なるため,2.1章で説明したような2枚の画像間で輝度に よって対応づけを測る手法をそのまま用いることができない.本章ではまず,イベントカメラのデー



図 2.2 タイムサーフェス. 図は [Benosman et al., 2014] 中のものを編集している.

タ形式について説明する.その後,イベントカメラを用いたオプティカルフロー推定手法について説 明する.

#### 2.2.1 イベントカメラ

イベントカメラは、従来のカメラのようにフレーム画像を取得することがなく、輝度の変化を画素 ごとに非同期的に検出する.ある画素  $\mathbf{x}_k \in \mathbb{R}^2$  において、時刻  $t_k$  で対数輝度値  $L := \log(I)$  が閾値 Cだけ変化した時に、一つのイベント  $e_k := (\mathbf{x}_k, t_k, p_k)$  を出力する.

$$L(\mathbf{x}_k, t_k) - L(\mathbf{x}_k, t_k - \Delta t_k) = p_k C$$
(2.17)

ここで,  $p_k \in \{+1, -1\}$ は, 輝度が明暗のどちらに変化したかを示す二値の値 (極性),  $\Delta t_k$ は同じピ クセルで前回イベントが発生してからの経過時間を示す.このように,一つのイベントは,2次元の 画素位置,タイムスタンプ,極性の4次元のタプルで表され,イベントデータはその時系列の集合で 表される.

#### 2.2.2 Surface of active events (SAE)

図 2.2 のように,あるエッジの運動に伴ってイベントが発生した時,イベントは *x,y,t* の 3 次元空間 で面を形成する.この面は Surface of active events (SAE),または,タイムサーフェスと呼ばれ,以 下のように表現される.

$$S_p: \mathbb{R}^2 \to \mathbb{R} \tag{2.18}$$

$$\mathbf{x} \mapsto S_p(\mathbf{x}) = t \tag{2.19}$$

つまり,各極性,画素ごとに最新のイベントの時間を保存する.図 2.3 は,実データによるタイムサー フェスを示している.タイムサーフェスはイベント群を画像サイズに集約し,計算効率をよくして おり,物体認識やコーナー検出など様々なタスクに利用されている [Lagorce et al., 2017; Mueggler et al., 2017a; Sironi et al., 2018; Ramesh et al., 2019; Manderscheid et al., 2019].

タイムサーフェスの空間方向の勾配ベクトル  $\nabla S_p(\mathbf{x}) = \begin{bmatrix} \frac{\partial S_p(\mathbf{x})}{\partial x}, \frac{\partial S_p(\mathbf{x})}{\partial y} \end{bmatrix}^{\top}$ は、空間に対する時間の 変化方向と変化率を示している. つまり、それぞれの要素は、**x** における運動ベクトルの逆数を表し ている. ここで、局所的な速度一定の仮定をすると、タイムサーフェスは平面 t = ax + by + c で表現 することができる. この平面を推定することで、ノーマルフローを推定することができる [Benosman et al., 2014; Aung et al., 2018; Rueckauer and Delbruck, 2016; Low et al., 2020; Nagata and Aoki, 2022]. 平面パラメータ  $\beta = [a, b, c]^{\top}$ は、局所窓での最小二乗法で推定可能である. ここで、平面の 基底を  $B = [x_i, y_i, 1]_{i=1,...,n} \in \mathbb{R}^{n \times 3}$ とし、ターゲットを $\mathbf{t} = [t_1, ..., t_n]^{\top}$ とすると、以下の式で推定 される.

$$\hat{\beta} = (B^{\top}B)^{-1}B^{\top}\mathbf{t} \tag{2.20}$$

ここで、nは局所窓内のイベント数とする.オプティカルフローは、推定された平面パラメータによっ



図 2.3 実データによるタイムサーフェス.データは [Mueggler et al., 2017b] のものを使用した.

て以下のように計算される.

$$\mathbf{v}^{\perp} = \frac{1}{\|\nabla S_p(\mathbf{x})\|} \nabla \hat{S}_p(\mathbf{x}) = \frac{1}{a^2 + b^2} \begin{bmatrix} a\\ b \end{bmatrix}$$
(2.21)

ここで、 $\nabla \hat{S}_p(\mathbf{x}) = \frac{\nabla S_p(\mathbf{x})}{\|\nabla S_p(\mathbf{x})\|}$ は $\nabla S_p(\mathbf{x})$ の単位ベクトルを表している.しかし、画像と同様に、窓問 題により局所的な推定では完全なオプティカルフローを得ることができず、ここで平面パラメータから計算できるフロー  $\mathbf{v}^{\perp}$ はノーマルフローである.

Lucas Kanade 法をイベントカメラ用に変更した手法 [Benosman et al., 2012] では,輝度の変 化をイベントによって近似している.また,Principal Component Analysis (PCA) を用いた手法 [Khairallah et al., 2022] や,局所窓内で速度が一定となる 3 つのイベントを探す Triplet matching と いう手法 [Shiba et al., 2023] もある.これらの手法は,どれも局所的な推定のみで,ノーマルフロー しか推定することができない.

これらのノーマルフローからマルチスケールで平均をとり,その平均したフローが最大となるス ケールを選択することで窓問題を緩和する手法 [Akolkar et al., 2022; Stumpp et al., 2022] が提案さ れている. 図 2.4 は,この手法のアイデアを表している. 左列のように,エッジが運動方向と垂直に なる時,ノーマルフローとオプティカルフローが一致し,ノーマルフローの大きさが最大となる.中



図 2.4 ある範囲の空間でノーマルフローの平均を取ることで窓問題の影響を軽減する手法 [Akolkar et al., 2022].

列のように、エッジが運動方向に対して角度を持つ時、ノーマルフローは小さくなる.よって、右図 のような複雑なパターンが同じ運動をしていると仮定した時、オプティカルフローと一致している大 きいノーマルフローを取り込むことで完全なオプティカルフローに近い推定ができるというアイデア である.この手法は、イベントが来るごとにそのイベントに対してのみ推定が可能なスパースな手法 である.しかし、平均をとる範囲に角度がついたエッジが存在する時、ノーマルフローの平均は完全 なオプティカルフローと一致しない.

このように、イベントカメラは輝度変化を高時間分解能で報告する仕組みから、局所的に運動を推 定することに優れている.一方、窓問題により局所的な範囲のみから完全なオプティカルフローを推 定することは、多くの場合に原理的に不可能である.そこで、本研究ではどのように局所的な対応づ けをとりつつ、周囲の情報と合わせて大域的に整合の取れたオプティカルフローを推定するか、また、 それをスパースかつ非同期的な処理で実現するかという問題に取り組む.



図 2.5 輝度復元を介したオプティカルフロー推定 [Bardow et al., 2016] のためのスライディング ウィンドウ. 図は [Bardow et al., 2016] のものに変更を加えた.

#### 2.2.3 輝度の復元を介したオプティカルフロー推定

イベントデータから完全なオプティカルフローを推定する手法として,輝度復元を介する手法 [Bardow et al., 2016] が提案されている. (2.4) 式で表されるオプティカルフローの拘束式をそのまま 適用するためにはイベントから輝度を復元する必要がある.輝度とオプティカルフローの関係式と輝 度とイベントの関係式の双方を同時に最適化することで,輝度とオプティカルフローを同時に推定す る.コスト関数は以下の式で表現される.

$$\int_{\Omega} \int_{T} (\lambda_1 \| \mathbf{v}_{\mathbf{x}} \|_1 + \lambda_2 \| \mathbf{v}_t \|_1 + \lambda_3 \| L_{\mathbf{x}} \|_1 + \lambda_4 \| \langle L_{\mathbf{x}}, \delta_t \mathbf{v} \rangle + L_t \|_1 + \lambda_5 h_\theta (L - L(t_p))) \, \mathrm{d}t \mathrm{d}\mathbf{x}$$
$$+ \int_{\Omega} \sum_{k=2}^{|P(\mathbf{x})|} \| L(t_k) - L(t_{k-1}) - Cp_k \|_1$$
(2.22)

 $\Omega$  は画像空間, T は時間幅,  $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$  は各項の重み, C はイベント発生の閾値を表している. このコスト関数を最小化することで,オプティカルフロー v と輝度 L を同時に推定する. 最小化



図 2.6 輝度復元を用いた手法 [Bardow et al., 2016] のコスト関数.

のために,画像空間 Ω を *M* × *N* のピクセルグリッドに離散化し,時間幅 *T* をそれぞれが Δ*t* の幅を 持つ *K* 個のセルに離散化する.離散化されたコスト関数は,図 2.5 のようなスライディングウィンド ウ上に定義され,1 セルずつスライドしながら最小化を行う.

図 2.6 は、コスト関数の 6 つの項の説明を表している. 1 項目から順にそれぞれ、オプティカルフ ローの空間方向の滑らかさ、オプティカルフローの時間方向の滑らかさ、輝度の空間方向の滑らかさ、 輝度の時間方向の整合性 (オプティカルフローの拘束式)、イベント未発生項、イベント発生項を表し ている.

5項目のイベント未発生項に含まれる $h_{\theta}$ は、以下の式で表される.

$$h_{\theta}(x) = \begin{cases} |x| - \theta & (|x| > \theta) \\ 0 & (otherwise) \end{cases}$$
(2.23)

L – L(t<sub>p</sub>) は前回のイベント発生時の輝度と現在の輝度の差を表しており,これがイベント発生の閾 値以上になっていたらその分ペナルティを与えるものである.超えていなければ正常のため,ペナル



図 2.7 コントラスト最大化 [Gallego et al., 2019]. イベントを運動パラメータでワープして作成 した画像のコントラストが最大になるようにパラメータを推定する. 図は [Gallego et al., 2019] のものに変更を加えた.

ティは与えない. 6項目のイベント発生項は,イベント発生の条件式である (2.17) 式を意味しており, 各画素で前後のイベント時刻での対数輝度の差がイベント発生の閾値になるようにペナルティを与え るものである.

画像ベースの手法と同様に,オプティカルフローの拘束式と空間的な滑らかさの制約をコスト関数 に加えており,完全なオプティカルフローを推定することができる.また,カメラの運動やシーンに 関する制約もない.

しかし, *K* = 128 枚もの輝度画像とオプティカルフローが最適化対象のパラメータとなっており, 非常に多い.また,これらをスパースなイベントから推定することは劣条件になりやすい.この輝度 復元を介す手法において,さらに精度を向上させるために,3章では車載シーンに着目した正則化を 提案した.

2.2.4 Contrast Maximization (CMax)

イベントデータからオプティカルフローなどの運動パラメータを推定する手法として、コントラス ト最大化 (CMax: Contrast Maximization)[Gallego et al., 2019; Stoffregen and Kleeman, 2019] が 提案されている. この手法は、イベントのアライメントを最大化する軌跡を探すことでオプティカル フローなどの運動パラメータを推定する手法である. ここで、イベントのアライメントは現在の候補 の運動パラメータ v によってイベントをワープして作成した Image of Warped Events (IWE) のコン トラストの大きさによって測られる. 図 2.7 は,手法の概要を示している. イベントを運動パラメー タでワープして作成した画像のコントラストが最大になるように運動パラメータを推定する.

運動パラメータは全画素で共通で 2 次元のベクトル **v** ∈ ℝ<sup>2</sup> とすると,イベントのワープによって 作成される IWE は以下のような式で表される.

$$I(\mathbf{x}; \mathbf{v}) = \sum_{k=1}^{N_e} p_k \delta(\mathbf{x} - \mathbf{x}'_k(\mathbf{v}))$$
(2.24)

ここで,  $\mathbf{x}'_k(\mathbf{v}) = \mathbf{x}_k - (t_k - t_{ref})\mathbf{v}$  はイベントのワープを表している. 時刻  $t_k$  で位置  $\mathbf{x}_k$  のイベント を, オプティカルフロー  $\mathbf{v}$  で基準時刻  $t_{ref}$  にワープしている.  $N_e$  はイベント数,  $\delta$  はデルタ関数を表 している. 以下の式で表すように, IWE の分散によって候補の軌跡にどれだけイベントがサポートし ているかを測ることができる.

$$Var(I(\mathbf{v})) = \frac{1}{N_p} \sum_{\mathbf{x}} \left( I(\mathbf{x}; \mathbf{v}) - \mu_I \right)^2$$
(2.25)

ここで、 $\mu_I = \frac{1}{N_p} \sum_{\mathbf{x}} (I(\mathbf{x}; \mathbf{v}))$ は IWE の平均値であり、 $N_p$  は画素数を表している. この指標を最 大化するように運動パラメータを更新することで、最善の軌跡を得ることができる. しかし、高い自 由度のモデルを設定するとイベントが、限られたピクセルにワープしてしまい局所解に陥ってしまう、 Event collapse という現象が発生すると知られている [Nunes et al., 2022; Zhu et al., 2019; Shiba et al., 2022a]. CMax のフレームワークをマルチスケール、マルチリファレンスに拡張することで、 より自由度の高いパラメタライズを行うことが可能である [Shiba et al., 2022b].

#### 2.2.5 Aritficial Neural Networks (ANN) を用いた学習ベース手法

近年では、画像を入力とした深層学習モデルによるオプティカルフロー推定手法の発展により、イベ ントデータに対しても同様のアプローチがなされている. イベントをタイムサーフェスやボクセルグ リッドなど密なテンソル表現に変換し、ネットワークに入力してオプティカルフローを推定する [Zhu et al., 2018a, 2019, 2018c; Ye et al., 2020; Gehrig et al., 2021b]. ネットワークの学習の仕方は、教 師あり、自己教師あり、教師なしの3つに大別される.


図 2.8 教師あり学習手法の代表的手法である E-RAFT[Gehrig et al., 2021b] のネットワーク構 造. 図は [Gehrig et al., 2021b] のものをそのまま用いた.

#### 教師あり学習

教師あり学習を用いた手法では、ネットワークによって出力されたオプティカルフローと正解のオプ ティカルフローの誤差が小さくなるように学習する.このような教師あり手法では、学習時と推論時 のデータの分布が同じであることが好ましい.特に、オプティカルフロー推定においては、カメラの 解像度が合っている必要がある.シミュレーションによって真値データを作成する方法は、真値情報 が得るのが困難であるオプティカルフローにおいて非常に有用な手段である.しかし、イベントカメ ラのさまざまなノイズ [Delbruck et al., 2021; Grac et al., 2023] を模擬することは困難であり、依然 として現実のデータと乖離があると言われている [Stoffregen et al., 2020].現実のデータの真値のオ プティカルフローは、静止シーンにおいて LiDAR による深度データと Inertial Measurement Unit (IMU) で取得されたカメラ運動によって作成される [Zhu et al., 2018b].しかし、データレートや視 野角、移動物体などの影響で正確な真値を得ることが難しいといった問題もある.教師あり学習手 法の代表例の一つ、E-RAFT[Gehrig et al., 2021b] は図 2.8 のように、画像分野で性能が良かった RAFT[Teed and Deng, 2020] の構造を用いた手法である.



図 2.9 半教師あり学習手法の代表的手法である EV-FlowNet[Zhu et al., 2018a] のネットワーク 構造. 図は [Zhu et al., 2018a] のものをそのまま用いた.

### 半教師あり学習

イベントデータによる半教師あり学習を用いた手法では、イベントと同時に輝度画像が得られるイベ ントカメラ (DAVIS[Brandli et al., 2014] など)を用いる. 2.1 章で述べた従来のフレームベースオプ ティカルフロー推定のように、2 枚の画像間で対応する輝度の差が一致するように学習される.入力に は、その画像間のイベントを用いる.この手法では学習時に、1 章で述べたフレームベースカメラの欠 点である、ダイナミックレンジの低さやモーションブラーなどの影響を受けることとなる.シーンの 明るさや運動の速さによってイベントに含まれるノイズが異なるため、これらのシーンを学習データ として用いることができないことは大きな問題となる.また、フレームレートの間隔でしか学習できな いことも問題の一つとして挙げられる.半教師あり学習手法の代表例の一つ EV-FlowNet[Zhu et al., 2018a] は、図 2.9 のように、U-Net[Ronneberger et al., 2015] の構造を用いている.EV-FlowNet[Zhu et al., 2018a] は、イベントデータを用いた学習ベースのオプティカルフローの先駆的な手法であり、 画像データのみを用いたオプティカルフロー推定の精度を上回る結果が報告されている.

### 教師なし学習



図 2.10 教師なし学習手法の代表的手法である FireFlowNet[Paredes-Vallés and de Croon, 2021] の学習方法. 図は [Paredes-Vallés and de Croon, 2021] のものをそのまま用いた.

教師なし学習を用いた手法では、タイムスタンプの平均 [Zhu et al., 2019; Tian and Andrade-Cetto, 2022] など、イベントの軌跡の整合性を測るロス関数が用いられる. FireFlowNet[Paredes-Vallés and de Croon, 2021] では、図 2.10 のように比較的軽いネットワークを用いてイベントデータからオプ ティカルフローと輝度を推定し、コントラスト最大化と輝度の整合をとることで学習する.

これらの学習ベースの手法は、イベントデータを画像のような密な表現に変更しており、スパース な推定ができていない.特に出力レート上げようとした場合も、同様の時間幅の入力を用いることと なり、オーバーラップしたイベントに対して再計算が必要となる.

#### 2.2.6 Spiking Neural Networks (SNN) を用いた学習ベース手法

生体の神経ネットワークでは、タイミング情報を持つ離散的なスパイクによって情報の伝達を行っ ている [Maass, 1997]. SNN は、2.2.5 章で述べたコンピュータビジョン分野で広く用いられている 一般的な ANN とは異なり、より生体に近いニューロンモデルを用いたスパイクベースのニューラ ルネットワークである. 一つのニューロンモデルである Integrate and Fire (IF) ニューロンモデル [Burkitt, 2006] では、膜電位と言われる内部状態を持ち、時間経過とともに入力されるスパイクによ



図 2.11 SNN の代表的手法である Spike-FlowNet[Lee et al., 2020a] の構造. 図は [Lee et al., 2020a] のものをそのまま用いた.

り膜電位が積算され、閾値を超えたときにスパイクを出力するモデルである. このような仕組みはイ ベントカメラの原理と共通しており、イベントデータをスパースかつ非同期的に、低消費電力で処理 するモデルとして着目されている. しかし、深い層においてスパイクの数が急速に減り、性能が悪化 することが示されている [Lee et al., 2020b]. また、その離散的なスパイク表現から、一般的な誤差逆 伝播による学習を適用できず、ネットワークの学習が困難である. 現在に至るまで単純なケースで検 討が進められ [Orchard et al., 2013; Haessig et al., 2017; Paredes-valles et al., 2020], 最近ではよ り複雑な実シーンでのオプティカルフロー推定など高度なタスク [Lee et al., 2020a; Chaney et al., 2021; Paredes-Vallés et al., 2021] に適用され始めている. しかし、代表的な Spike-FlowNet は、図 2.11 のようにデコーダー部分は ANN であったり [Lee et al., 2020a], ANN と比較して依然として精 度に課題がある [Chaney et al., 2021; Paredes-Vallés et al., 2021]. また、これらを処理するための ハードウェアとして、IBM の TrueNorth[Merolla et al., 2014] や Intel の Loihi[Davies et al., 2018] など、神経生物学的なアーキテクチャを模倣する回路が開発されている.

### 2.3 本章のまとめ

本章では、本研究の主題であるオプティカルフロー推定について、古典的な画像ベースのオプティ カルフローとイベントベースのオプティカルフローについて述べた.古典的な画像ベースのオプティ カルフローでは、輝度の一貫性の仮定や画像間での運動が十分小さい (<1 画素) という仮定を用いて いる.しかし、30fps などの一定の時間で撮影される画像では、高速な運動の時にこの仮定が成り立た ない場合がある.また、ニューラルネットを用いたオプティカルフロー推定手法も提案されているが、 密な処理で計算量が大きく、リソースが限られた中での高速なオプティカルフロー推定は困難である. それに対して、イベントカメラは輝度の変化のみを検出するカメラであり、時間分解能が高く、オプ ティカルフロー推定に適していると言える.イベントデータからオプティカルフローを推定する手法 についていくつか紹介した.局所的なイベントを用いてを平面フィッティングなどで対応づけを行う 手法は、ノーマルフローのみしか推定することができない.完全なオプティカルフローを推定する手 法では、輝度復元を行うものや、コントラストを最大化する手法などが存在する.また、近年では学 習ベースの手法が増えている.

# 3 拡張焦点を用いた正則化

3.1 導入

イベントカメラは,高時間分解能やハイダイナミックレンジといった特性を持つことから,逆光や 夜間という環境下での動的なシーンが含まれる車載への応用が期待される.その中で,オプティカル フロー推定は前方物体の運動解析において重要な役割を担う [Sato et al., 2011].本研究では,車載 シーンにおけるイベントカメラを用いたオプティカルフロー推定の高精度化を目的とする.

2.2 章では、イベントカメラを用いたオプティカルフロー推定手法について紹介した.その中でも 2.2.3 節で述べた輝度復元を介す手法では、非学習手法であり、カメラ運動によらず完全なオプティカ ルフローを推定可能である.この手法は、従来の画像認識と同様の、輝度による対応づけから導かれ るオプティカルフロー拘束式とオプティカルフローの滑らかさの正則化に加え、イベントと輝度の関 係性の項、輝度の滑らかさの項を含めたコスト関数を最適化する.しかし、従来の画像ベースの手法 と異なり、未知パラメータに輝度を追加で含んでおり、パラメータ数が多く最適化が困難であると言 える.また、イベントデータは空間的にスパースであり、密な推定では大部分が滑らかさの正則化に よってされることになり全体として平坦な推定になる.

オプティカルフローは、カメラ運動による静止した環境からのものと移動物体からのものと、2つ に大別することができる.前者は、6自由度のカメラ運動のパラメータと環境の各画素での深度情報 から構成され [Irani et al., 1994; Zhu et al., 2018a],その自由度は各画素2のオプティカルフローよ り小さい.特に、着目している車載シーンでは、カメラの並進は光軸方向が支配的である.また、カ メラの回転はハンドル操作によるヨー方向の変動が支配的である.本研究では、自車運動による発生 した静止環境のオプティカルフロー推定に限定し、オプティカルフローの正則化を提案する.

古くからカメラの並進運動と拡張焦点 (FOE: focus of expansion) の性質を用いた運動の推定が行 われている [Irani et al., 1994; Hu et al., 1999]. 本研究では, イベントデータからオプティカルフ ローを推定する最適化の中にこの性質を取り入れ, より高精度なフロー推定を目指す.

### 3.2 拡張焦点 (FOE: focus of expansion)

FOE は、図 3.1 のように、カメラが並進運動のみを行う場合、カメラの微小時間における並進軸と 画像平面の交点として定義される [Hu et al., 1999]. ここで、X, Y, Z はカメラ座標系、x, y は画像座 標系を表す. FOE の画像座標系における位置を  $(x_0, y_0)$ , 並進ベクトルを  $\mathbf{T} = [T_X, T_Y, T_Z]^{\top}$  とする と、カメラ座標系と画像座標系の関係から、焦点距離 f を用いて以下の式を導くことができる.

$$\frac{x_0}{T_X} = \frac{y_0}{T_Y} = \frac{f}{T_Z}$$
(3.1)

ゆえに、FOEの座標は次の式のようになる.

$$\begin{cases} x_0 = f \frac{T_X}{T_Z} \\ y_0 = f \frac{T_Y}{T_Z} \end{cases}$$
(3.2)

車載カメラは、カメラが車体に固定されていて進行方向の運動成分が支配的、つまり、 $T_X \ll T_Z$ 、  $T_Y \ll T_Z$ という関係があることから、カメラが一旦取り付けられると FOE は一定であり、その位置 はほぼ画像中心になるという性質がある。カメラの光軸と車の進行方向が完全に一致していれば FOE は画像中心になるが、ずれていれば進行方向以外の並進運動が生まれるため、画像中心ではなくなる。 本研究内では、FOE は画像中心として扱った。

### 3.3 FOE を用いた車載カメラ運動解析

FOE を用いると車載カメラの運動解析が容易になる. 図 3.2 のように,カメラの並進運動 T に加 え,回転運動  $\omega = [\omega_X, \omega_Y, \omega_Z]^{\top}$ を仮定する. この時,カメラ座標系の点 P の点 P' への移動を考え ると,画像座標系の対応点 p(x, y), p'(x', y')は以下の関係式になる [Hu et al., 1999]. 詳しい式展開



図 3.1 FOE の定義. X, Y, Z はカメラ座標系, x, y は画像座標系を表す.

は付録 A に記述する.

$$\frac{(x'-x) - \left(-\frac{xy}{f}\omega_X - \frac{f^2 + x^2}{f}\omega_Y - y\omega_Z\right)}{(y'-y) - \left(-\frac{f^2 + y^2}{f}\omega_X - \frac{xy}{f}\omega_Y - x\omega_Z\right)} = \frac{x - x_0}{y - y_0}$$
(3.3)

この式は、対応点座標と回転運動パラメータ、FOE の座標の3つで構成されている.本研究では、カ メラの回転運動は Y 軸周りの回転速度  $\omega_Y$  のみを考慮する.よって、 $\omega_X = \omega_Z = 0$ を (3.3) 式に代入 して以下の式を得る.

$$\frac{(x'-x) - \left(-\frac{f^2 + x^2}{f}\omega_Y\right)}{(y'-y) - \left(-\frac{xy}{f}\omega_Y\right)} = \frac{x - x_0}{y - y_0}$$
(3.4)

対応点を結ぶベクトルはオプティカルフローそのものであり、以下のように置き換える.

$$\mathbf{v}_{(x,y)} = \begin{bmatrix} v_{x,(x,y)} \\ v_{y,(x,y)} \end{bmatrix} = \begin{bmatrix} x' - x \\ y' - y \end{bmatrix}$$
(3.5)

ここで下付き添え字の (x,y) は画素の位置を表している.また,(3.4) 式の分子分母の 2 項目はオプ ティカルフローの回転による成分を表しており,以下のように置き換える.

$$\mathbf{v}_{(x,y)}^{\omega} = \begin{bmatrix} v_{x,(x,y)}^{\omega} \\ v_{y,(x,y)}^{\omega} \end{bmatrix} = \begin{bmatrix} -\frac{f^2 + x^2}{f} \\ -\frac{f^2 + x^2}{f} \\ -\frac{f^2 + x^2}{f} \end{bmatrix}$$
(3.6)

以上のように, (3.5) 式, (3.6) 式を用いて, (3.4) 式を以下のように書き直す.

$$\frac{v_{x,(x,y)} - v_{x,(x,y)}^{\omega}}{v_{y,(x,y)} - v_{y,(x,y)}^{\omega}} = \frac{x - x_0}{y - y_0}$$
(3.7)

この式は,図 3.3 で示すように,オプティカルフローからカメラの回転成分を除くと,FOE から放射 状になるという性質を表している.この性質を利用して,車載イベントカメラによるオプティカルフ ロー推定における正則化を提案する.

### 3.4 拡張焦点を用いた正則化

3.3 章で述べたように、回転による成分を除いたオプティカルフロー  $\mathbf{v}' = \mathbf{v} - \mathbf{v}^{\omega}$  は、FOE から放 射状になるという性質を持つ.この性質を利用するために、ヨーレート  $\omega_Y$  を推定し、(3.6) 式からオ プティカルフローの回転成分  $\mathbf{v}^{\omega}$  を求める必要がある.ヨーレート  $\omega_Y$  は、(3.6) 式と (3.7) 式から、 以下のように推定される.

$$\hat{\omega}_Y = f \frac{v_{x,(x,y)}(y-y_0) - v_{y,(x,y)}(x-x_0)}{(f^2 + x^2)(y-y_0) - xy(x-x_0)}$$
(3.8)

ヨーレートは,各対応点 (オプティカルフロー) で一つ推定されるため,各対応点で推定されたヨー レートの中央値を用いた.ヨーレート ŵ<sub>Y</sub> が求まれば,オプティカルフローの回転による成分は (3.6)



図 3.2 車載カメラの運動モデル.

式で表される.

ここで、FOE $(x_0, y_0)$ から各画素(x, y)までのベクトルを $\mathbf{v}_{\text{foe},(x,y)} = [x - x_0, y - y_0]^{\top}$ とする.また、FOE から放射状で、大きさが $\|\mathbf{v}'\|$ のベクトルを $\mathbf{v}'_{\text{foe}} = \|\mathbf{v}'\| \cdot \frac{\mathbf{v}_{\text{foe}}}{\|\mathbf{v}_{\text{foe}}\|}$ とする、図 3.4 のように、回転による成分を除いたオプティカルフロー $\mathbf{v}'$ を以下のように $\mathbf{v}'_{\text{foe}}$ に近づける、

$$\hat{\mathbf{v}}' = (1 - \alpha)\mathbf{v}' + \alpha \mathbf{v}_{\text{foe}}' \tag{3.9}$$

ここで, αは0から1の値をとる重みである.そして,最後に回転成分を戻してオプティカルフロー を得る.

$$\hat{\mathbf{v}} = \hat{\mathbf{v}}' + \mathbf{v}^{\omega} \tag{3.10}$$

最後に,全体のアルゴリズムを図 3.5 に示す.提案の正則化は,繰り返し最適化の中で輝度復元およ びオプティカルフロー推定後に行う.



図 3.3 FOE の性質. オプティカルフロー  $\mathbf{v}$  から回転による成分  $\mathbf{v}^{\omega}$  を除く ( $\mathbf{v}' = \mathbf{v} - \mathbf{v}^{\omega}$ )と, FOE から放射状になるという性質を持っている.



図 3.4 FOE を利用した正則化. 回転による成分を除いたフローが FOE に従うように修正する.



図 3.5 FOE を用いた正則化手法の全体概要.

### 3.5 実験

提案の FOE を用いた正則化手法の有用性を検証するために、実験を行った.

#### 3.5.1 データセット

本手法は、車載シーンに限定した正則化手法であることから、車載シーンのイベントカメラ公開デー タセットである Multi-Vehicle Stereo Event Camera dataset (MVSEC)[Zhu et al., 2018b] を用い て実験を行った. このデータセットは、ステレオのイベントカメラや LiDAR, IMU, GPS が搭載さ れているデータ収集系を、車やバイク、ドローンに搭載して収集された. 撮影されたシーンには、日 中と夜間の運転シーン、ドローンによる室内シーンが含まれている. 使用されたイベントカメラは、 DAVISm346B であり、解像度は 346×260 である. このイベントカメラは、イベントデータと同時に フレーム画像を 50fps で出力することができる. LiDAR は Velodyne Puck LITE を使用しており、 20Hz で 100m までの深度情報を得ることができる. オプティカルフローの真値は、LiDAR による深 度データと IMU による回転情報を用いて、静止シーンという仮定の下に計算されている [Zhu et al., 2018a].

#### 3.5.2 評価指標

推定されたオプティカルフローの精度評価には、以下の式で表される Average Endpoint Error (AEE)を用いた.この評価指標は、正解のオプティカルフローの推定したオプティカルフローの差分 のフローベクトルの大きさの平均を表しており、小さいほど良い評価指標である.

$$AEE = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{v}_{i}^{pred} - \mathbf{v}_{i}^{gt}\|_{2}$$
(3.11)

この時,評価にはイベントが発生かつ,真値が存在する画素のみを用いた.電柱などの物体の深度情報が取得できなかった場合,その画素は評価から除外した.*N*は評価に用いた画素数,**v**<sup>pred</sup>は推定されたオプティカルフロー,**v**<sup>gt</sup>は真値のオプティカルフローを表す.また,手法の途中で推定したヨーレートの精度評価には,平均二乗誤差 (RMSE)を用いた.

	Optical flow (AEE)		Yaw rate (RMSE)	
	(pix/frame)		(rad/s)	
	day1	day2	day1	day2
[Bardow et al., 2016]	0.297	0.304	$1.58\times 10^{-3}$	$0.57\times 10^{-3}$
Ours	0.260	0.269	$1.46 imes 10^{-3}$	$0.50 imes10^{-3}$
Improvement	12.4%	11.7%	7.56%	12.0%

表 3.1 実験結果.

#### 3.5.3 実験設定

比較は, Bardow らの手法 [Bardow et al., 2016] と,提案した正則化を加えた本手法の2つの場合 で行った.使用したシーンは outdoor\_day1, outdoor\_day2 の2つで,推定したフレーム数はそれぞ れ 30000, 78000 フレームである.また1フレームの長さ  $\Delta t$  を 7.5ms, スライディングウィンドウ内 のセル数 K を 128 とした.提案手法の正則化パラメータ  $\alpha$  は 0.5 とした.

#### 3.5.4 実験結果

実験結果を表 3.1 に示す.提案手法は,Bardow らの手法と比較して,オプティカルフローの AEE で 12.4%,11.7%,ヨーレートの RMSE で 7.56%,12.0% の改善が見られた.本手法によって FOE から放射状になるという性質を利用することで,車載シーンにおけるオプティカルフローの正則化が可能であることが分かった.

図 3.6 に, 推定されたオプティカルフローの例を示す. 左の列から順にイベントデータ,本手法で推 定した輝度画像,標準フレーム画像, Bardow らの手法で推定したオプティカルフロー,提案手法で推 定したオプティカルフロー,真値のオプティカルフローを表示している.オプティカルフローの可視 化では,画像右下の色環のように,色相で向きを,明度で大きさを表している.右3列のオプティカル フローの色相に着目すると,Bardow らの手法ではL1 正則化の効果で空間微分がスパースになり,上 下左右方向に偏り,赤,黄緑,水色の3色が多くなっているが,提案手法により FOE に従う向きに制 約がかかったことで放射状に正しい向きに推定されていることが,中間の色が増えたことから分かる.



図 3.6 推定されたオプティカルフローの例.



図 3.7 ヨーレート推定結果. 上図は outdoor\_day1 の,下図は outdoor\_day2 の結果を表している.

図 3.7 に, 推定されたヨーレートを時系列で示す. 上図は outdoor\_day1 の結果であり, 下図は outdoor\_day2 の結果である. 値が上下に大きく振れるときはカーブをしているシーンであり, その時 においてもヨーレートが求まっている. これは, (1) 車載カメラの運動において前進方向の運動が支配 的である, (2)FOE が画像中心に固定されるという 2 つの仮定が概ね正しいことを示している.

#### 3.6 本章のまとめ

本章では、車載シーンにおけるオプティカルフロー推定のための正則化手法を提案した.提案手法 は、車載カメラの運動において前進方向の運動が支配的であり FOE が画像中心に固定されるという 仮定と、回転成分を除いたオプティカルフローが FOE から放射状になるという性質を利用して、オプ ティカルフローに制約を与える手法である.実験では、車載シーンのイベントカメラ公開データセッ トである MVSEC を用いて、提案手法の有用性を検証した.その結果、オプティカルフローの精度が 約10%程度向上したことが確認された.

導入でも述べたように、本手法は自車の運動によって発生した静止環境のオプティカルフロー推定 において有効である.よって、移動体が存在する場合、背景点かどうかの判別によってマスクする必 要がある.推定したオプティカルフローから、FOE に従わない消失点をグループ化することで、移動 体の検出が可能であると考えられる [Sato et al., 2011].また、検出した移動体のカメラに対する相対 運動が並進のみの場合、その移動体上のオプティカルフローは消失点を持つ [Sato et al., 2011] こと から、本手法と同様の正則化が適用できると考える.

# 4 Surface matching

### 4.1 導入

前章では,イベントデータから輝度復元とオプティカルフロー推定を同時にする手法における, FOE を用いた車載シーン特有の正則化を提案した.車載シーンであるという事前情報を積極的に用い ることで,オプティカルフローの推定精度を向上させることができた.

しかし, 輝度の変化を検出することでスパースなセンシングを実現したのにも関わらず, 全画素の輝 度を復元し, 輝度で対応づけを測ることでオプティカルフローを推定することは冗長な処理であると 言える.また, イベントデータから輝度を復元するために, 長い時間のイベントと大量の輝度画像の未 知パラメータを用意している.具体的には,時間幅 KΔt の長さのイベントを用い, K + 1(K = 128 が使われている)時点の画像を未知パラメータとしている.長い時間幅のイベントを用いることで, イ ベントタイムスタンプと画像時刻から線形補間によって輝度差を扱い,時間方向の微分の L1 ノルムを 正則化として加えることができる.オプティカルフローは, Δt のインターバルで復元した輝度から輝 度一貫性の仮定で推定しているが,輝度を離散化した情報であるイベントから推定した輝度画像は平 坦な箇所が多く好ましい処理だとは言えない.

そこで本章では、イベントデータから輝度復元を介さないでオプティカルフローを推定する手法を 提案する.提案手法では、従来の輝度の代わりに、イベントデータの時間を使って対応づけを測るこ とでオプティカルフローを推定する.固定時間幅の輝度復元を介さず、イベントの時間で直接対応づ けを測ることから、イベントの高い時間解像度を活用していると言える.

実験では,先行研究よりロス関数の形状が穏やかで,最適化がしやすいことを定性的に示す.また, 実際に密なオプティカルフロー推定において高精度に推定できることを定量的に示す.

### 4.2 Surface matching loss

2.2.2 章で定義した,極性ごとの各画素における最新イベントの時間表すタイムサーフェスという表現について考える.図 4.1 の左図の矢印のような,剛体シーンの非常に短い時間間隔  $\Delta t$  での物体の線形な運動を考えると,画素  $\mathbf{x} := (x, y)$  における  $\Delta t$  間の変位  $\mathbf{v}(\mathbf{x})$  を使って,イベントの時間の一致を表す以下の式が導かれる.

$$S_p(\mathbf{x}) = S_p(\mathbf{x} + \mathbf{v}(\mathbf{x})) - \Delta t \tag{4.1}$$

ここで、右辺はタイムサーフェスを Δt だけ時間方向にシフトしたものを表している. タイムサーフェ スは、イベントの時間で構成されているため、任意の時間を足し引きすることで時間方向にシフトす ることができる. Δt だけシフトしたタイムサーフェスを以下のように定義する.

$$S'_{p}(\mathbf{x}) := S_{p}(\mathbf{x}) - \Delta t \tag{4.2}$$

(4.1) 式をシフトしたタイムサーフェスを用いて書き直すと以下のようになる.

$$S_p(\mathbf{x}) = S'_p(\mathbf{x} + \mathbf{v}(\mathbf{x})) \tag{4.3}$$

この式は,タイムサーフェス  $S_p$  とシフトしたタイムサーフェス  $S'_p$  の間の画素の対応づけを表している.実際には,それぞれのタイムサーフェスは,参照時刻  $t_0$  から  $\tau$  の間のイベントを集めて構成する. それぞれのタイムサーフェスの値域は, $S_p, S'_p \in [t_0 - \tau, t_0]$ である.

(4.3) 式は, (2.14) 式の非線形オプティカルフロー拘束式をタイムサーフェスに適用したものであり, 同様に, 近傍点 v<sub>0</sub>の周りでテイラー展開することで以下の式を得る.

$$\rho_p(\mathbf{v}) = \nabla S'_p(\mathbf{x} + \mathbf{v}^0) \cdot (\mathbf{v} - \mathbf{v}^0) + S'_p(\mathbf{x} + \mathbf{v}^0) - S_p(\mathbf{x})$$
(4.4)

サーフェスマッチングロスは、(4.4) 式のL1 ノルムを各極性、全ての画素で足し合わせることで計算

される.

$$E_{\text{surface}} = \sum_{\mathbf{x}} \sum_{p} \|\rho_p(\mathbf{v})\|_1 \tag{4.5}$$

滑らかさの正則化項を加えると以下のようになる.

$$E = \sum_{\mathbf{x}} \left( \sum_{p} \|\rho_{p}(\mathbf{v})\|_{1} + \lambda \|\nabla \mathbf{v}\|_{1} \right)$$
(4.6)

(4.6) 式のコスト関数を最小化することでオプティカルフローを推定する.

本手法において,速度一定の仮定をおいているのは非常に短い時間幅の  $\Delta t$  の間のみであり, $\tau$ では ない.図 4.2 に速度が変化した時の一次元のタイムサーフェスを示す.ここで示すように,推定され るオプティカルフロー  $\mathbf{v}(\mathbf{x})$ は,サーフェス上の位置  $\mathbf{x}$ ,時刻  $S_p(\mathbf{x})$ における推定値であり, $\tau$  の間で の速度一定は仮定していない.

2.2.2 章では、タイムサーフェスの勾配が速度の逆数になっており、速度一定の仮定で平面フィッ ティングすることで速度ベクトルを計算していた.ここで、初期値として  $\mathbf{v}_0 = \mathbf{0}$  を考えると、(4.4) 式は  $\rho_p(\mathbf{v}) = \nabla S'_p(\mathbf{x}) \cdot \mathbf{v} + \Delta t$  となり、 $\Delta t$  の間でタイムサーフェスのフィッティングをしているもの と考えられる.本手法では、局所速度一定ではなく、オプティカルフローが近傍で滑らかに変化する という仮定の下、滑らかさの正則化と繰り返しの最適化によってオプティカルフローを推定している ものと解釈することができる.また、タイムサーフェスを活用することで、各時刻でのフィッティン グを画像サイズで同期的に行うことで GPU を用いて並列化が容易にしているとも言える.

図 4.3 に提案手法の概要を示す.まず,時空間のイベントから,各極性においてタイムサーフェス  $S_p$  とシフトしたタイムサーフェス  $S'_p$  を作成する.次に,オプティカルフローを使ってシフトしたタ イムサーフェス  $S'_p$  をワープして,元のタイムサーフェス  $S_p$  と差分を測り,小さくなるように最適化 する.最適化は,画像ベースの TV-L1 法 [Sánchez Pérez et al., 2013] と同様に primal-dual アルゴ リズムによって行う.



図 4.1 タイムサーフェスとオプティカルフローの関係. 左図はある Y 軸方向に伸びているエッジ が X 軸方向に一定の速度で移動した時のタイムサーフェスである. 右図はそれを  $\Delta t$  だけシフト したタイムサーフェスを表してる.



図 4.2 速度がが変化した時の一次元のタイムサーフェス.



図 4.3 手法の概要.時空間のイベントから,  $\tau$ 間のイベントで各極性のタイムサーフェス  $S_p$  と,  $\Delta t$ だけシフトしたタイムサーフェス  $S'_p$  を作成する.シフトしたタイムサーフェス  $S'_p$  を現時点の オプティカルフローの推定値を使ってワープして,元のタイムサーフェス  $S_p$  と差分を測り,小さ くなるように勾配法によってオプティカルフローを最適化する.

### 4.3 実験

提案したロス関数の有効性を検証するために、ロス関数の形状の可視化とオプティカルフローの精 度検証を行った.

#### 4.3.1 データセット

データセットは、以下の3つを用いた.

### ESIM

An Open Event Camera Simulator (ESIM)[Rebecq et al., 2018] は,任意のカメラ運動やシーンで イベントとオプティカルフローを出力可能なイベントカメラのシミュレータである.ESIM では,検 証のために,矩形 (rectangle) のシーン,チェッカーボード (checkerboard) のシーン,レンガ (bricks) のシーン,芝 (grass) のシーンで生成した.解像度は,240×180とした.

### MVSEC

3.5.1 章で述べた MVSEC[Zhu et al., 2018b] を用いた.

#### HACD

The HVGA ATIS Corner Dataset (HACD)[Manderscheid et al., 2019] は, コーナー検出のための データセットで, 平面のパターンをイベントカメラで撮影したものである. 使用されたイベントカメラ は Asynchronous Time-based Image Sensor (ATIS)[Posch et al., 2011] であり, 解像度は 480 × 360 である. このデータセットには, 10ms ごとに計測されたポスターの4つのコーナーのマーカーの位置 を含んでいる. 任意のタイミングでマーカーの位置を線形補間によって求めることで, 平面のホモグ ラフィーを推定することができ, 真値のオプティカルフローを得ることができる. 真の運動モデルの 自由度は 8 であるが, 全画素で密にオプティカルフローを推定する各画素 2 パラメータの設定として 実験を行った.

#### 4.3.2 ロス関数の形状の可視化

[Gallego et al., 2019] で行われてるように、ロス関数の形状を可視化することで、ロス関数の最適 化の難易度を評価することができる.本実験では、オプティカルフローを全ピクセルで一定の2パラ メータとし、パラメータ空間 v ∈ ℝ<sup>2</sup> を可視化した.ロス関数の計算は、30pix 四方の領域内で計算し た.比較は以下の2つで行った.



図 4.4 シミュレーションデータによるロス関数の形状の観察. (a) 矩形シーンの概要. 矩形が画 像平面上で  $\mathbf{v} = (2, -1)^{\top} (pix/\Delta t)$  で並進しているシーン. (b) オプティカルフローパラメータ空 間  $\mathbf{v} \in \mathbb{R}^2$  でのロス関数のプロット. それぞれのロスは [0,1] の範囲で正規化している. 下図は (c)  $\mathbf{v} = (2, 0)^{\top} (pix/\Delta t)$ , (d)  $\mathbf{v} = (1, 0)^{\top} (pix/\Delta t)$  における輝度画像と IWE, タイムサーフェスと それらのマゼンタの四角部分を拡大したものを示している. マゼンタの矢印は勾配ベクトルを表し ている.

### Variance

(2.25) 式で表された分散. 使用したイベントの時間間隔は  $\Delta t + \tau$  として提案手法と揃えた.

### Surface Matching Loss

(4.6) 式で表されたロス関数. タイムサーフェス  $S_p$  とシフトしたタイムサーフェス  $S'_p$  の差分によっ

て計算される.

### 実験結果

図 4.4 に矩形シーンのロス関数の形状を示す. (b) を見ると,提案のロス関数の形状は,全方位に穏 やかに変化していることが分かる. (d) で勾配ベクトルを見ると, Variance は, x 方向の勾配が大き くなっている. 一方,提案の Surface Matching Loss は, x, y 方向に正しい勾配を持っていること



図 4.5 シミュレーションデータによるロス関数の形状の観察.上から checkerboard, bricks, grass のシーンを示している.全てのシーンで,  $\mathbf{v} = (1,0)^{\top} (pix/\Delta t)$  で並進しているシーンとした.

が分かる.また,図 4.5 に checkerboard, bricks, grass のシーンのロス関数の形状を示す. Variance は,シーンのテクスチャに依存して,極値周辺が鋭くなっていることが分かる.このように,コント ラスト最大化は,シーンのテクスチャに依存して,イベントが多く重なる方に引き寄せられることが ある [Zhu et al., 2019].一方,提案のロス関数の形状は,全てのシーンで全方位に穏やかに変化して いることが分かる.

#### 4.3.3 オプティカルフロー推定

提案のロス関数の効果を検証するために,密なオプティカルフロー推定の精度を比較した.実験は, 他の要素を排除するために,学習ベースではなくオンライン最適化ベースの手法で行った.比較は以 下の3つで行った.

#### Reconstruction

2.2.3 章で述べた輝度復元を介する手法 [Bardow et al., 2016]. 128Δ*t* の時間幅のスライディングウィ ンドウで,時空間の滑らかさの仮定の下, 129枚分の輝度を復元し,輝度の対応づけを測ってオプティ カルフローを推定する. 輝度とオプティカルフローは交互に最適化される.

#### Variance

(2.25) 式で表される IWE の分散を最大化するようにオプティカルフローを推定する手法 [Gallego et al., 2019]. オプティカルフローは画素ごとに 2 パラメータとし, L1 ノルムを用いた滑らかさの正則化を加えたコスト関数とした.

$$\min_{\mathbf{v}} \sum_{\mathbf{x}} \left( \|\nabla \mathbf{v}(\mathbf{x})\|_1 - \lambda Var(I(\mathbf{x}; \mathbf{v}(\mathbf{x}))) \right)$$
(4.7)

### Surface Matching Loss

以下のように, (4.6) 式で表されるサーフェスマッチングロスと滑らかさの正則化を合わせたロス関数 を最小化するようにオプティカルフローを推定する手法.

$$\min_{\mathbf{v}} \sum_{\mathbf{x}} \left( \|\nabla \mathbf{v}(\mathbf{x})\|_1 + \lambda \sum_p \|\rho_p(\mathbf{x}, \mathbf{v})\|_1 \right)$$
(4.8)

タイムサーフェス  $S_p, S'_p \in [t_0 - \tau, t_0]$ は [0, 255]に正規化した.

オプティカルフローを計算する時間間隔は、MVSEC の outdoor シーンで 7.5ms, indoor シーンで 10ms, HACD のすべてのシーンで 10ms とし、有効なタイムサーフェスの時間幅  $\tau$  はすべて  $10\Delta t$  とした.

#### 実験結果

MVSEC, HACD における定性評価を表 4.1 に示す. MVSEC の outdoor シーンでは 5 シーン中 2 シーンで, MVSEC の indoor シーン, HACD の全シーンで, 提案手法が最も精度が高い結果となっ た. MVSEC の outdoor シーンにおける定性評価を図 4.6 に, HACD データにおける定性評価を図 4.7 に示す. Reconstruction は,時間方向の正則化に合わないシーンで,ノルムが小さく出る傾向に ある. 提案手法の Surface Matching の結果は,正解値 (GT) に近いオプティカルフローが推定され ていることが分かる.

精度が低いシーンを詳しく見ると,outdoor シーンは地面の凹凸によって車が上下に大きく振れる シーンがあった.しかし,正解のオプティカルフローは,移動平均フィルタを用いて平滑化した速度を

	$outdoor_day1$	$outdoor_day2$	
Reconstruction	0.267	0.307	-
Variance	0.479	0.479	
Surface Matching	0.257	0.350	
	$outdoor\_night1$	$outdoor\_night2$	$outdoor\_night3$
Reconstruction	0.283	0.313	0.365
Variance	0.418	0.368	0.438
Surface Matching	0.334	0.363	0.356
	indoor_flying1	indoor_flying2	indoor_flying3
Reconstruction	0.348	0.525	0.468
Variance	0.351	0.525	0.469
Surface Matching	0.278	0.422	0.377
	guernica	paris	graffiti
Reconstruction	1.99	2.79	1.91
Variance	4.01	3.11	1.90
Surface Matching	1.50	2.30	1.36

表 4.1 オプティカルフロー推定の定量評価. 評価指標は AEE(pix/ $\Delta t$ ) を用いた.

用いているため,滑らかに変動する. Reconstruction は,時間方向に 128△t の長い時間窓を用いて, 時間方向の滑らかさの正則化を加えているため,振動の影響を抑えて評価を高くすることができたと 考えられる. 実際に,AEE と IMU による pitch レートを図 4.8 に示す. グラフの左の方では pitch レートが小さく振動が少なく,右の方では振動が多くなっている. 振動が少ない時は,提案法の精度 が高いことが分かる. 図 4.9 に振動が大きいシーンの例を示す. Surface Matching では,振動が大き いシーンでは,振動に合わせてオプティカルフローが推定されていることが分かる. GT に近いオプ ティカルフローを推定するには,Reconstruction のように時間方向の滑らかさの正則化を加える必要 がある.

#### 実行時間

提案手法の実行時間は、CPU での MATLAB による実装で 3fps 程度であった.しかし、最適化は画



図 4.6 MVSEC の outdoor シーンにおける定性評価. 左から, 画像, イベント (黄緑と赤が正 負それぞれのイベントの極性, 黄色は両方発生していることを示している), Reconstruction, Variance, Surface Matching, Ground Truth のオプティカルフローを示している. オプティカ ルフローは, 右下のように, 色相で向き, 明るさで大きさを表している. 行方向はサンプルを示し ている.



図 4.7 HACD データにおける定性評価. 左からイベント (黄緑と赤が正負それぞれのイベントの 極性, 黄色は両方発生していることを示している), + のタイムサーフェス, -のタイムサーフェス, Reconstruction, Variance, Surface Matching, Ground Truth のオプティカルフローを示して いる. オプティカルフローは, 右下のように, 色相で向き, 明るさで大きさを表している.

像ベースの TV-L1 法とほぼ同等で,画素ごとに並列化可能であり,GPU の実装でリアルタイム実行 が可能である [Zach et al., 2007]. さらに,画像ベースの手法とは異なり Δ*t* を任意の幅に調整できる ことから,複数解像度のピラミッドを用いる必要がなくさらに早くなると考えられる.また,初期値 として前時間ステップの推定値を用いることで,最適化にかかる繰り返し回数を減らすことが可能で あると考えられる.



図 4.8 MVSEC の outdoor シーンにおける pitch レートと AEE.



図 4.9 MVSEC の outdoor データの振動が大きいシーン.

### 4.4 本章のまとめ

本章では、イベントデータから輝度復元を介さないでオプティカルフローを推定する手法を提案した. た.従来の輝度の代わりに、イベントデータの時間を使って一致度を計算するロス関数を提案した. 滑らかさのL1 正則化を加えて最適化することで、周囲で整合性をとりつつサーフェスのフィッティ ングを行い、オプティカルフローを推定可能とした.提案したロス関数は、テクスチャに依存しにく く、勾配が全方向に穏やかであることを示した.実際に、密なオプティカルフロー推定を行ったところ、提案手法は従来の手法よりも高い精度を達成した.

本手法は,タイムサーフェスを用いた画素ごとの処理が多く GPU を用いた同期的な並列化が容易 である.一方で,タイムサーフェスをタイムサーフェスを形成する時間幅  $\tau$  の間でオクルージョンが 発生するとその時点の推定が出来なくなってしまう.また,高いレートで出力しようとすると,使用 するイベントのオーバーラップが大きくなり,再計算が多くなってしまう.

# 5 信念伝播法による非同期イベントオプティカルフロー推定

TEGBP: Tangentially Elongated Gaussian Belief Propagation for Event-based Incremental Optical Flow Estimation

5.1 導入

3章,4章で提案したオプティカルフロー推定手法は、どちらもイベントデータを用いた手法であり ながら、イベントのバッチに対してフレームベースの手法と同様に一定の時間間隔で画素全体で最適 化を行う手法であった.しかし、出力レートを上げようとした時、時間的にオーバーラップするイベ ントデータについて再計算をすることとなり計算量が増加する.また、イベントデータは輝度の変化 がない箇所では発生しないことから、画像全体での処理は無駄な計算となる.このことは、2.2.4章で 述べた CMax、2.2.5章の畳み込みニューラルネットワークを用いた手法においても同様である.

それに対して、2.2.2 章で述べた局所的な平面フィッティングの手法は、スパースで非同期的な手法 であり、インクリメンタルに計算が可能である.つまり、究極には新しく入ってきたイベント一つに 対して、周囲の古い計算結果を再利用して推定が可能である.しかし、このような局所的な推定では ノーマルフローしか推定することができない.

図 5.1 はバッチ処理とインクリメンタルな処理を比較したものである. イベントをバッチで扱う手 法では, 推定のたびに決まった数または時間のイベント群を処理するため, 推定のレートを上げるこ とが難しい. それに対して, インクリメンタルな処理では新しいイベントのみを部分的に処理するこ とから, 推定のレートを高くすることが可能である.

そこで、本研究ではスパースで非同期的でありながら、完全なオプティカルフローを推定すること を目的とする.以下のような性質を持つ手法を目指す.

 スパース/ローカル: 画像全体を密に処理するのではなく、イベントのあった箇所だけを部分 的/局所的に計算する.



図 5.1 バッチ処理とインクリメンタルな処理の模式的比較.

- 完全なオプティカルフロー: 輝度勾配成分であるノーマルフローではなく、オプティカルフローを推定する.
- インクリメンタル:古いイベントに対して再計算をせず、新しい観測に対して古い計算結果を 再利用して漸進的に推定を行う.

手法のキーとなるアイデアは,信念伝播法 (BP: belief propagation) を用いて,局所的に分かるオプ ティカルフローの分布を周囲に伝播することで大域的に整合の取れたオプティカルフローを推定する ことである.信念伝播法は,確率モデルをグラフ構造で表したファクターグラフにおいて,局所的な 計算によるメッセージ交換を行うことで,近似的にベイズ確率推論を行う手法である.信念伝搬法に ついては付録 B で追加の説明を行った.また,詳細は [Davison and Ortiz, 2019; Ortiz et al., 2021] を参照されたい.

画素をノードとしたファクターグラフにおいて,各画素に観測ファクター,画素間に事前情報ファク ターを設定し,信念伝搬法によってそれぞれのノードの周辺分布を求めることでオプティカルフロー を推定する.各ノードの信念は2次元のガウス分布で表現され,メッセージ交換を通して更新される. 2.2.2節で述べたように,オプティカルフローの輝度勾配に平行な成分であるノーマルフローは,局所 的に推定可能である.そこで,図5.2のように,推定したいオプティカルフローの分布を,ノーマルフ ローを中心とし拘束線方向に大きな分散を持つ2次元のガウス分布でモデリングし,観測ファクター



図 5.2 提案手法 TEGBP の概要. RGB の楕円で表すように,各ノードでのオプティカルフロー の分布を細長い 2 次元のガウス分布 (TEG) でモデリングし,適切な事前分布の下に統合すること によってマゼンタで表されるオプティカルフローの事後分布を得る.

とした. また, 画素間には滑らかさの事前情報をガウス分布でモデリングした.

イベントが入ってくるごとにその周辺のノードのみでメッセージ交換を行うことで,スパースで非 同期的な推定でありながら,近傍に情報を伝播し完全なオプティカルフローを推定可能である.非同 期的/インクリメンタルに各ノードの分布 (信念)を更新する仕組みにより,古いイベントに対する再 計算を要さず,任意のタイミングで現在の信念の平均を取り出すのみで推定結果を出力することがで きる.これにより,任意に出力レートを調整することができる.

### 5.2 問題設定

現時刻を t として,  $[t - \tau, t]$  の間でノーマルフローの集合  $V_t^{\perp}$  が観測されたときの,対応するオプ ティカルフローの事後分布  $p(V_t \mid V_t^{\perp})$  を求めたい.事後分布が求まれば,オプティカルフローの推定 値は事後分布最大 (MAP: maximum a posteriori) で求めることができる.

$$\hat{V}_t = \operatorname*{arg\,max}_{V_t} p(V_t \mid V_t^{\perp}) \tag{5.1}$$

本章の目的は、ノーマルフローを用いてインクリメンタルにオプティカルフローを推定することで ある. 各ノーマルフローが得られるごとに、 $V_t$ の同時分布の argmax を計算することは、計算量やメ モリ量の観点で扱うことができない. そこで、図 5.3 で表されるように、新しいノーマルフロー  $\mathbf{v}_{i\,t+1}^{\perp}$ 



図 5.3 インクリメンタルなオプティカルフロー推定.あるノード*i*の新たな時刻のノーマルフ ロー $\mathbf{v}_{i,t+1}^{\perp}$ を入力とし、メッセージ交換によって周辺ノードのみの分布をインクリメンタルに更新 する.出力は任意のタイミングで取り出すことができ、即時に対応するオプティカルフロー $\mathbf{v}_{i,t+1}$ を出力することも可能である.



図 5.4 TEG によるオプティカルフローのモデリング. ノーマルフロー  $\mathbf{v}^{\perp}$  が得られたとき,オプ ティカルフロー  $\mathbf{v}$  はノーマルフローを通り,ノーマルフローの向き (輝度勾配方向) と垂直な線上 のどこかであることがわかる (左図). そこで,その不確実性を,平均がノーマルフロー  $\mathbf{v}^{\perp}$ ,接線 方向に大きい分散を持つ 2 次元のガウス分布 (TEG) で表現した (右図).

が得られた時,以下のように事後分布  $p(V_t \mid V_t^{\perp})$  を更新する.

$$[p(V_t \mid V_t^{\perp}), p(\mathbf{v}_{i,t+1}^{\perp})] \mapsto p(V_{t+1} \mid V_{t+1}^{\perp}).$$
(5.2)

このようなインクリメンタルなオプティカルフロー推定を実現する手法を提案する.



## 5.3 Tangentially Elongated Gaussian (TEG)

2.1.1 章で述べたように、ノーマルフロー  $\mathbf{v}^{\perp}$  が得られたとき、オプティカルフロー  $\mathbf{v}$  はノーマルフ ローを通り、ノーマルフローの向き (輝度勾配方向) と垂直な線上のどこかであることがわかる. そこ で、その不確実性を、図 5.4 の右図に示すような、平均がノーマルフロー  $\mathbf{v}^{\perp} = \begin{bmatrix} v_x^{\perp}, v_y^{\perp} \end{bmatrix}^{\top}$  で接線方向 に大きい分散を持つ 2 次元のガウス分布 (TEG: Tangentially Elongated Gaussian) で表現した. そ の精度行列  $\Sigma^{\text{teg}}$  を以下のように定義する.

$$\Sigma^{\text{teg}} = \mathbf{R}(\theta) \begin{bmatrix} \sigma_{\text{r}}^2 & 0\\ 0 & \sigma_{\text{t}}^2 \end{bmatrix} \mathbf{R}(\theta)^{\top}$$
(5.3)

 $\mathbf{R}(\theta)$  は回転行列であり、 $\theta$ はノーマルフローの角度 $\theta = \arctan(v_y^{\perp}/v_x^{\perp})$ とする.

ここで, σ<sub>r</sub> = 0, σ<sub>t</sub> = ∞ の場合は, ノイズの無いノーマルフローが得られた時のフローの拘束線と 一致する.しかし, 実際には得られるノーマルフローにはノイズが含まれるため, σ<sub>r</sub> > 0 とする.ま た,オプティカルフローは接線方向に均一に分布せずノーマルフローの周りに集中するため, σ<sub>t</sub> を有 限の値に設定する.これらの事前情報は,本来 ill-posed なオプティカルフロー復元問題において重要 な手掛かりとなる.実際に実データで真値のオプティカルフローを利用して接線方向のフローの分布 を調べると図 5.5 のようになった.よって,本研究では 2 次元のガウス分布でモデリングすることを 提案した.



図 5.6 TEG のパラメータ  $\sigma_t$  を変化させた時の,それぞれの TEG(赤,緑,青) と周辺分布 (マゼ ンタ). 黒が真のオプティカルフロー,マゼンタが推定したオプティカルフローを表す.

### 5.4 事前情報と TEG の同時分布の周辺化

ここで、図 5.2 の例について考える. 2 つの向きのエッジを持ったコーナーが右方向に進んでいる とする. ある 3 つの画素  $p_1, p_2, p_3$  があり、 $p_1$  と  $p_2$  は同じエッジに属しているとする. 各画素でノー マルフロー  $\mathbf{v}_1^{\perp}, \mathbf{v}_2^{\perp}, \mathbf{v}_3^{\perp}$  が得られたとする (図中の赤、黄緑、青矢印に対応する). ここで、3 つの画素 は同じ剛体上にあるとし、同じオプティカルフロー (図中の黒矢印) によって動いているとする. この 時、3 つの画素の 3 つの TEG の同時分布  $p(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$  を、画素 2 のフロー  $\mathbf{v}_2$  について周辺化する と、周辺分布は 3 つの TEG の重ね合わせになる.

$$p(\mathbf{v}_2) = \mathcal{N}(\mathbf{v}_2; \mathbf{v}_1^{\perp}) \mathcal{N}(\mathbf{v}_2; \mathbf{v}_2^{\perp}) \mathcal{N}(\mathbf{v}_2; \mathbf{v}_3^{\perp})$$
(5.4)

実際に計算を行うと、図中のマゼンタの分布になり、その平均は黒い矢印と重なっている. これは、単 に平均をとったものではなく、3 つの TEG の同時分布を考慮したものになっている. このように、適 切な事前情報 (この例では 3 つのフローが同じ) の下に TEG を合わせることで、オプティカルフロー の推定を行うことができる.

実際に, TEG のガウス分布の長径の標準偏差 σ<sub>t</sub> を変化させた時の, それぞれの TEG と周辺事後 分布を図 5.6 に示す. σ<sub>t</sub> が大きいほど, オプティカルフローの推定値は真値に近づくことが分かる.

この例では,隣接ノードで同じフローであるという仮定を置いたが,次の 5.5 章で説明するように, 本手法では単純な滑らかさの事前情報をガウス分布で表現し,近傍のノード間にファクターとして設



図 5.7 ファクターグラフ.トポロジーはアクティブノードセット *A*t に応じてダイナミックに変動する.最近イベントが発生した画素のノードをアクティブとしている.同じ画素のアクティブな変数ノード(青)と観測ノード(黄緑)は繋がっている.また,近所のアクティブな変数ノード同士は事前情報ファクターノードを通して繋がっている.

定した.

### 5.5 ファクターグラフ

(5.1) 式の同時分布  $p(V_t | V_t^{\perp})$  はベイズ則を用いてファクター分解をすることができる.

$$\hat{V}_t = \operatorname*{arg\,max}_{V_t} p(V_t^{\perp} \mid V_t) p(V_t)$$
(5.5)

観測の独立性と近傍ピクセルの相関を考慮し, (5.5) 式をさらに分解することで, 図 5.7 で示すよう なスパースなファクターグラフを構築する. 観測ファクター ψ と事前情報ファクター φ を以下のよう に定義する.

$$p(V) = \prod_{n \in \mathcal{A}_t} \psi(\mathbf{v}_n) \prod_{(i,j) \in \mathcal{E}_t} \phi(\mathbf{v}_i, \mathbf{v}_j)$$
(5.6)

ここで,  $A_t$  は時刻 t におけるアクティブなノードの集合,  $\mathcal{E}_t$  はアクティブなノード間のエッジの集合 とする.ここでは, アクティブなノードは時刻 t から時間幅  $\tau$  以内にイベントが発生したノードとし た.観測ファクター  $\psi$  は, 5.3 節で定義した TEG を用いて以下のように定義した.

$$\psi(\mathbf{v}_n) = \mathcal{N}^{-1}(\cdot | \boldsymbol{\eta}_n, \boldsymbol{\Lambda}_n^{\text{teg}})$$
(5.7)

ここで、精度行列は  $\Lambda_n^{\text{teg}} = \text{inv}(\Sigma_n^{\text{teg}})$ ,情報ベクトルは  $\eta_n = \Lambda_n^{\text{teg}} \mathbf{v}_n^{\perp}$ とする.また、事前情報ファク ター  $\phi$  は、オプティカルフローの滑らかさを仮定して、以下のように設定した.

$$\phi\left(\begin{bmatrix}\mathbf{v}_i\\\mathbf{v}_j\end{bmatrix}\right) = \mathcal{N}^{-1}\left(\cdot;\mathbf{0},\mathbf{\Lambda}^{\mathrm{p}}\right),\tag{5.8}$$

ここで、精度行列 Λ<sup>P</sup> は以下のように表される.

$$\Lambda^{\mathrm{p}} = \mathbf{J}_{\mathrm{p}}^{\top} \begin{bmatrix} \frac{1}{\sigma_{\mathrm{p}}^{2}} & 0\\ 0 & \frac{1}{\sigma_{\mathrm{p}}^{2}} \end{bmatrix} \mathbf{J}_{\mathrm{p}}$$
(5.9)

ここで、J<sub>p</sub>は、2つのノード間の変数の差分を計算するヤコビアンである.

$$\mathbf{J}_{\mathbf{p}} = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$
(5.10)

また,外れ値にロバストな推定にするため,観測ファクター,事前情報ファクターの両方を Huber モデルにした. [Agarwal et al., 2013; Davison, 2018] の共分散をスケーリングするテクニックを用い て,ガウシアンの形を保ったまま計算が可能である.

また,平面フィッティングによるノーマルフロー推定は,コーナーのような線形で表現できない箇 所でモデルエラーが発生してしまう.この情報を手法に反映させるために,ノーマルフローの信用度 によって観測ファクターの分散をスケーリングすることで,モデルエラーを考慮した推定とした.具 体的には,平面フィッティング時の,平面をサポートするイベントの割合の逆数を (5.3) 式にかけて分 散を大きくした.このように,柔軟に分布をモデリングできることが,提案手法の特徴の一つである.

### 5.6 信念伝搬法による周辺化

前節で定義したファクターグラフを信念伝搬法によって各ノードに対して周辺化を行った.ファク ターは全てガウス分布で表現しているため,信念伝搬は精度行列と情報ベクトルの非常に小さい行列 演算によって計算可能である (今回の例では2×2の逆行列).ここで,メッセージ交換はノーマルフ ローが得られたごとに非同期的に開始される.観測ファクターノードは,観測を受け取った時に変数


図 5.8 一連のメッセージ送信スキーム.

ノードにメッセージを送り、そのノードの信念を更新する.

$$\boldsymbol{\eta}_{i}^{\mathrm{b}} = \sum_{j \in \mathcal{C}_{i} \cap \mathcal{A}_{t}} \boldsymbol{\eta}_{j \to i}, \quad \boldsymbol{\Lambda}_{i}^{\mathrm{b}} = \sum_{j \in \mathcal{C}_{i} \cap \mathcal{A}_{t}} \boldsymbol{\Lambda}_{j \to i}$$
(5.11)

ここで、 $b(\mathbf{v}_i) = \mathcal{N}^{-1}(\cdot; \boldsymbol{\eta}_i^{\mathrm{b}}, \boldsymbol{\Lambda}_i^{\mathrm{b}})$ はノードiの信念、 $\mu_{j \to i} = \mathcal{N}^{-1}(\cdot; \boldsymbol{\eta}_{j \to i}, \boldsymbol{\Lambda}_{j \to i})$ はノードiからノードjへのメッセージを表しており、いずれも 2 次元のガウス分布である。また、 $C_i$ はノードiに接続しているノードの集合を示している。

その後,事前情報ファクターノードを通して,以下のようなメッセージによって,更新した信念の 情報を接続したノードに送る.

$$\boldsymbol{\eta}_{j \to i} = -\Lambda_{01}^{\mathrm{p}} (\Lambda_{11}^{\mathrm{p}} + \Lambda_{j}^{\mathrm{b}} - \Lambda_{i \to j})^{-1} (\boldsymbol{\eta}_{j}^{\mathrm{b}} - \boldsymbol{\eta}_{i \to j})$$
(5.12)

$$\Lambda_{j \to i} = \Lambda_{00}^{\rm p} - \Lambda_{01}^{\rm p} (\Lambda_{11}^{\rm p} + \Lambda_j^{\rm b} - \Lambda_{i \to j})^{-1} \Lambda_{10}^{\rm p}$$
(5.13)

ここで、 $\Lambda^{p}_{lphaeta}$ は、以下のように分割した、それぞれが $2 \times 2$ の行列を示している.

$$\Lambda^{\mathrm{p}} = \begin{bmatrix} \Lambda^{\mathrm{p}}_{00} & \Lambda^{\mathrm{p}}_{01} \\ \Lambda^{\mathrm{p}}_{10} & \Lambda^{\mathrm{p}}_{11} \end{bmatrix}$$
(5.14)

つまり,送信する方向のノードを除いたノードから受信したメッセージを足し合わせ(信念からメッ

セージを除くのと同意), ファクターノードで送信元のノードについて周辺化をすることで計算される. これらのメッセージ送信スキームを図 5.8 に示す. 観測であるノーマルフローが発生したノードで, 信念を更新し, 繋がっているノードにメッセージを送る. その先のメッセージを受け取ったアクティ ブなノードで信念を更新する. そして, また周囲にメッセージを送り, その先のメッセージを受け取っ たアクティブなノードで信念を更新する. このような一連の流れを, ノーマルフローが得られるごと に非同期的に行う.

収束を早くするために, coarse-to-fine にメッセージ送信するスキームを提案する. [Felzenszwalb and Huttenlocher, 2004] では, 画像のピラミッドを作成し, 粗い階層から順にメッセージを送ること で, 収束を加速できることが示されている. 粗い階層 *l* の観測ファクターノードを, 細かい階層 *l* – 1 の2×2のブロックのアクティブな観測ファクターノードの和によって計算する.

$$\boldsymbol{\eta}_{i}^{l} = \sum_{j \in \mathcal{B}_{i} \cap \mathcal{A}_{t}} \boldsymbol{\eta}_{j}^{l-1}, \quad \boldsymbol{\Lambda}_{i}^{l} = \sum_{j \in \mathcal{B}_{i} \cap \mathcal{A}_{t}} \boldsymbol{\Lambda}_{j}^{l-1}, \tag{5.15}$$

ここで, *B<sub>i</sub>*はノード*i*に対応するブロックを示している.メッセージ送信は,粗い階層から順に行い, 粗い階層でのメッセージを細かい階層へとコピーし,初期値として開始する.これにより,遠い距離 のノードにメッセージを送ることができ,収束を早くすることができる.

## 5.7 アルゴリズム

TEGBP のアルゴリズムを Algorithm 1 に示す. また, Algorithm 1 内のメッセージ交換の部分は Algorithm 2 に分けて示す.

#### 5.8 実験

提案した TEGBP の性能を評価するために、オプティカルフロー推定の精度の比較を行った.比較 対象は、同じくインクリメンタルなフロー推定手法である Aperture-robust multi-scale (ARMS) フ ロー推定 [Akolkar et al., 2022] とした. ARMS は、ノーマルフローを入力とし、適切な空間的範囲で

## Algorithm 1 TEGBP

1:	for all incoming normal flow $e_i = \{\mathbf{x}_i, t_i, \mathbf{v}_i^{\perp}\} \mathbf{do}$
2:	Calculate the message from the data factor by Eq. 5.3 and Eq. 5.7 and set to node at pixel
	$\mathbf{x}_i.$
3:	Update hierarchical data factor by Eq. 5.15.
4:	for all $l$ in $[L, \cdots, 1]$ do
5:	Update belief by Eq. 5.11
6:	Message passing (Algorithm 2)
7:	$\mathbf{if} \ l>1 \ \mathbf{then}$
8:	Copy the messages from a coarser layer $l$ to a finer layer $l-1$ .
9:	end if
10:	end for
11:	Determine the estimated value $\mathbf{v}_i$ as the mean of Gaussian belief.
12:	end for
Al	gorithm 2 Message passing at node <i>i</i>
1:	Add node <i>i</i> to queue <i>Q</i> .
2:	for all $k$ in $K$ hops do
3:	Pop all nodes in $Q$ and add to the temporary queue $Q'$ .
4:	for all $j$ popped from $Q'$ do
5:	Message passing by Eq. 5.12.
6:	for all $k \in \mathcal{A}_t \cap \mathcal{C}_j$ do
7:	Update belief by Eq. 5.11.
8:	Add node $k$ to queue $Q$ .
9:	end for
10:	end for
11:	end for

ノーマルフローの平均を計算することで窓問題にロバストなフローを推定する手法である. その空間

的範囲は平均したフローが一番大きくなるように全探索で決定する.

二つの手法はどちらも、スパースなノーマルフローから非同期的にフローを推定するインクリメン

タルな手法である.公平な比較のため、ノーマルフロー推定には共通のアルゴリズムを用いた.

Paramet	ESIM	DVS	MVSEC	DSEC	
Window size	r [pix]	5	5	3	5
window size	$\Delta t_{\rm pf} \ [{\rm ms}]$	40	40	40	40
Active duration	$\tau \; [ms]$	50	50	100	100
Prior std	$\sigma_{\rm p}$ [pix]	0.1	1	0.5	1
Tangential std	$\sigma_{\rm t}$ [pix]	10	10	10	10
Radial std	$\sigma_{\rm r}$ [pix]	3	3	3	3
Num. of layers	L	5	5	5	5
Batch size	$N_b$	100	100	1000	1000

表 5.1 Basic experimental setup.

#### 5.8.1 設定

ノーマルフロー推定には、2.2.2 章に述べたシンプルな平面フィッティング手法を用いた. 生のイベ ントに対して、各ピクセルで最新イベントから 40ms の間のイベントを除去する refractory フィルタ を適用した後、 $r \times r \times \Delta t_{pf}$ の大きさの時空間ウインドウ内のイベントを用いて x - y - tの3次元空 間で平面フィッティングを行った. 平面フィッティングの際、[Benosman et al., 2014] で提案されて いる外れ値除去を3回繰り返して適用した. ノーマルフロー推定のパラメータを表 5.1の上部に示す.

どちらの手法も非同期的なアルゴリズムで、ノーマルフローが得られるレートでオプティカルフ ローを推定することができる.しかし、計算効率を考え、 $N_b$  個のイベントを同時に処理するバッチ処 理を行った.TEGBP アルゴリズムのパラメータは表 5.1 の下部に示す.TEGBP では、収束するま で全ノードでメッセージ交換を行うべきであるが、K = 2 ホップで1 イベントにつき1 回の繰り返し ( $N_{\rm itr} = 1$ )で十分であることが分かった.

#### 5.8.2 ESIM-bricks を用いた実験

まず,イベントカメラシミュレータ (ESIM)[Rebecq et al., 2018] を使用して作成したデータを用い て,提案手法の性能を評価した.作成したシーンは,斜めに敷き詰められた煉瓦のシーンに対して,カ メラを並進させ作成したシーンである.運動方向と画像のエッジが垂直になっていないシーンであり, オプティカルフローとノーマルフローが一致しないシーンである.図 5.9 に, ESIM-bricks による実



図 5.9 ESIM-bricks による実験結果. 左から, ノーマルフロー, ARMS, TEGBP による推定オ プティカルフローを結果を示す. 正解のオプティカルフローは矢印によって示してる.



図 5.10 TEG の長径の標準偏差  $\sigma_t$  を変化させた時の ESIM-bricks の実験結果.

験結果を示す. ノーマルフローは, レンガのエッジに垂直な方向を向いていることが分かる. ARMS は, ノーマルフローの平均を取るという操作によって,大きさが小さくなって推定されていることが 分かる. 一方で,提案手法は,推定されたフローが真値の方向(右向き)に向いていることが分かる. オプティカルフローの分布を TEG によってモデリングしたことにより,正しい向きのフローの尤度 を高くすることが可能であるからと考えられる.

#### 5.8.3 TEG の分散パラメータに関する実験

TEGBP は、拘束線方向に大きな分散を持つ 2 次元のガウス分布で表現することが重要なポイント であった.この効果を確認するために、ESIM-bricks データにおいても、ガウス分布の長径の標準偏 差 σ<sub>t</sub> を変化させた時のオプティカルフローを観察した.図 5.10 に実験結果を示す.σ<sub>t</sub> が大きくなる と、オプティカルフローの分布が細長くなり、ノーマルフローより遠い方向のフローの尤度が高くな るため、正解値に近いオプティカルフローを推定することができる.



図 5.11 DVS-stripes による実験結果. 左から, ノーマルフロー, ARMS, TEGBP による推定オ プティカルフローを結果を示す. 赤い線は, 2 つの速度の正解のオプティカルフローを示している.

表 5.2 MVSEC における定量評価. オプティカルフローは真値が得られている 20Hz で評価した.

	indoor_flying1		indoor_flying2 in		indoor	indoor_flying3		outdoor_day1	
	$AEE\downarrow$	%Out↓	$AEE\downarrow$	%Out↓	$AEE\downarrow$	%Out↓	$AEE\downarrow$	%Out↓	
Norm	2.30	24.5	3.61	42.9	3.13	36.3	3.44	43.0	
ARMS	1.71	12.7	2.67	26.6	2.28	21.6	2.64	25.7	
Ours	1.14	6.25	1.87	16.4	1.54	11.8	1.46	11.1	

#### 5.8.4 DVS-stripes を用いた実験

次に,異なるフローが混在するシーンでの推定オプティカルフローを評価するために,ダイナミッ クビジョンセンサ (DVS)[Lichtsteiner et al., 2008] によって 2 つの違う速度の縞模様を撮影したデー タ [Mueggler et al., 2015] を用いた.図 5.11 に,DVS-stripes による実験結果を示す.ARMS の推 定結果は、半径で決まった範囲で平均を取るため、2 つのフローが混ざってしまい推定結果がバイア スを持ってしまっていることが分かる.平均フローが大きくなるようにスケールを選択するため、小 さいフローのエリアで大きなスケールの選択をしてしまい、大きいフローが混ざることでノルムが大 きくなる方にバイアスを持って推定してしまう.それに対して、TEGBP は 2 つの速度をそれぞれ推 定できていることが分かる.



図 5.12 MVSEC における定性評価.

#### 5.8.5 MVSEC を用いた実験

実践的なシーンで ARMS と比較するために MVSEC[Zhu et al., 2018b] のデータを用いた. 使用 したシーンは車によるアウトドアシーンと, ドローンによるインドアシーンである. 評価指標とし て, 平均エンドポイントエラー (AEE) とエンドポイントエラーが 3 ピクセル以上あるピクセルの割 合 (%Out と示す)を用いた. どちらの指標も, 正解値が存在し, かつ, 平面フィッティングが成功し たピクセルのみを用いて計算した. どちらの手法でも非同期的に推定することができるが, 正解値の レート 20Hz の一定間隔で評価を行った.

表 5.2 に, MVSEC による数値的な実験結果を示す. 前シーンにおいて, ARMS はノーマルフロー を改善しているが, 提案手法がさらに改善していることが分かる. 図 5.12 に定性的な実験結果を示 す. 特に, ノーマルフローがオプティカルフローと一致しない (輝度勾配が運動方向と一致しない) シーンにおいて, 大きな改善があることが分かる. これは, ARMS は単に平均をとっているのに対し て, TEGBP は適切にオプティカルフローの分布をモデリングできているからであると考える.

	thun_00_a		zurich_c	city_01_a	zurich_city_02_a	
	$AEE\downarrow$	%Out↓	$AEE\downarrow$	%Out↓	$AEE\downarrow$	%Out↓
Norm	7.09	68.7	8.26	37.9	12.8	86.8
ARMS	5.31	60.0	6.32	23.8	8.98	78.7
Ours	4.26	45.5	5.61	23.2	8.70	72.6

表 5.3 DSEC における定量評価. オプティカルフローは真値が得られている 10Hz で評価した.

	zurich_city_08_a		zurich_city_11_a		
	$AEE\downarrow$	%Out $\downarrow$ AEE $\downarrow$		%Out↓	
Norm	8.42	73.3	5.90	63.4	
ARMS	6.22	64.4	4.57	51.4	
Ours	5.08	<b>49.5</b>	3.27	32.2	



図 5.13 DSEC における定性評価.

### 5.8.6 DSEC を用いた実験

さらに多様なシナリオで比較を行うため,最近リリースされた高解像度なデータセットである DSEC[Gehrig et al., 2021a] を用いた. DSEC は,自動車によるアウトドアシーンであり,MVSEC よりも高解像度の 480 × 640 である.イベントカメラには,Prophesee 社の Gen3 を用いている.こ のデータセットは,画像全体の密なオプティカルフロー推定のために作成されたものである.テスト データの正解値が隠されており,推定結果を提出すると評価され,リーダーボードに結果が表示され る.ここでは,学習用シーケンスを用いてスパースなオプティカルフローとして AEE を計算して比 較を行った.



図 5.14 メッセージ交換スキーム. 黄色がイベントが発生した画素を示す. このノードから 8 近 傍で L 種類の遠さのノードにメッセージを送る. この時, l 番目は  $2^{l-1}$  個離れたノードとした.

表 5.3 に, DSEC による数値的な実験結果を,図 5.13 に定性的な実験結果を示す. MVSEC の結果 と同様に, ARMS の精度を大きく改善した.

5.8.7 マルチコア並列化

本提案手法は局所的な演算のみからなっており、一つのイベントをきっかけとする一連のメッセージ交換を、複数イベントで並列化できるという利点がある.この優位性を検証するために、CPUのマルチコアを利用して並列化を行った [Sekikawa and Nagata, 2023].実装には、OpenMP[Chandra et al., 2000] を利用した.

マルチスケールのメッセージ交換は, Algorithm 2 から, 図 5.14 で表されるようなアルゴリズムに 変更した. 粗い階層から順に行うのではなく, 2<sup>*l*-1</sup> 個離れたノードへ直接送ることで, 並列化を行っ た. また, 4 近傍から 8 近傍へとメッセージを送るように変更して実装した. 使用した CPU はインテ ル<sup>®</sup> Core<sup>™</sup> i9-9820X, クロック周波数は 3.30GHz である.

図 5.15 は、コア数に応じた相対的な実行時間 (イベントデータの時間に対する実行時間) を示す.相対的な実行時間が1以下になると、リアルタイムに処理できていることとなる.結果から、使用する コア数を増やすと実行時間が短くなることが分かる.これは、本提案手法が並列化によって高速化で きることを示している.



図 5.15 コア数に応じた相対的な実行時間 (イベントデータの時間に対する実行時間). 10<sup>0</sup> がリア ルタイムを表している.

現状のアルゴリズムでは、イベント発生からの経過時間によってアクティブかどうかを決定し、メッ セージ送信をするかを判断していたが、[Ranganathan et al., 2007] のようにメッセージの差分が大き い時のみ送信するようなアルゴリズム (wild fire) を用いることで、メッセージの送信数を減らすこと ができると考えられる.

#### 5.8.8 同期的な手法との比較

本提案手法は、イベントごとに推定可能な非同期的な手法であるが、メッセージをアクティブな全 画素において一斉に送信する同期的な推定も可能である.このような設定で、4章で提案したサーフェ スマッチングの手法と比較を行った.比較には MVSEC データセットを用いた.それぞれの手法にお いて現時点から有効なイベントの時間幅τは同じ 100ms とした.TEGBP では、平面フィッティング が成功した後の画素で現時刻からτ以内、サーフェスマッチングではタイムサーフェスが現時刻から τ 以内で、かつ真値が存在する画素で評価を行った.

表 5.4 に,実験結果を示す.全てのシーンにおいて TEGBP が上回った.この理由として 3 点考え られる.1つ目に,平面フィッティングによるノイズ除去があると考えられる.TEGBP では,ノー マルフロー推定の際に平面フィッティングを繰り返す際にノイズイベントを除去することができるが, サーフェスマッチングではサーフェスをそのまま使うためノイズが入ってしまう.2つ目に,分布の明

	indoor_flying1		indoor_flying2		indoor_flying3		outdoor_day1	
_	$AEE\downarrow$	%Out↓	$AEE\downarrow$	%Out↓	$AEE\downarrow$	%Out↓	$AEE\downarrow$	%Out↓
$\mathbf{SM}$	1.58	10.3	2.68	30.6	2.28	24.1	2.13	23.1
TEGBP	1.39	9.79	2.14	24.4	1.86	19.1	1.64	16.3

表 5.4 MVSEC における定量評価. オプティカルフローは真値が得られている 20Hz で評価した. 比較はサーフェスマッチング (SM) と TEGBP で比較を行った.

示的なモデリングが挙げられる. TEGBP では, ノーマルフローに近くなるという事前情報を盛り込 んだ TEG によって 2 次元の楕円でオプティカルフローのモデリングをしているが, サーフェスマッ チングでは拘束線となっているため外れ値が入ってきた時に値が大きくずれやすい. 前者ではさらに, ノーマルフロー推定のフィッティングの良さによって TEG の分散を調整しているため, 外れ値を除 きやすい仕組みになっている. 3 つ目に, 時間的な平滑化が挙げられる. TEGBP は, インクリメンタ ルな手法であり, 分布を更新する形で推定されることから時間方向の平滑作用がある. MVSEC デー タは, カメラ姿勢を時間平滑して作成されているため, TEGBP の精度が高くなったと考えられる.

#### 5.9 本章のまとめ

3章,4章の手法では、スパースで非同期的なイベントデータに対して、画像のように画像全体で 同期的に扱っていた.それに対して本章では、イベントデータからスパースでインクリメンタルにオ プティカルフローを推定するための新しいアルゴリズムを提案した.局所的に推定可能なノーマルフ ローを用いてオプティカルフローの分布を細長いガウス分布 (TEG)でモデリングした.また、スパー スなグラフ上で近傍の画素とメッセージ交換をすることで、近傍の画素の情報と合わせて大域的に整 合の取れた完全なオプティカルフローを推定可能とした.メッセージ交換は、エッジで繋がっている ノードにおいて局所的な演算のみで行われるため、画像全体での処理を必要としない.このように、 提案手法はスパースかつ非同期的、インクリメンタルなアルゴリズムであり、出力レートを任意に設 定することができる.

実験では,オプティカルフローの精度において,従来のスパースな手法を上回った.また,並列化 が容易であり, CPU のマルチコアで実装することでリアルタイムに推定可能なことを示した. 現状のアルゴリズムの欠点としては、ノーマルフローの推定精度が低いとオプティカルフローの推 定精度が低くなることが挙げられる.実験では、単純な繰り返し平面フィッティングを用いたが、この 部分に CNN などより高精度なノーマルフロー推定手法を用いることで、オプティカルフローの推定精 度を向上させることができると考えられる.また、同時にノイズ除去 [Baldwin et al., 2020; Nagata and Aoki, 2022] を行うことで、ノーマルフローの推定精度を向上させることができると考えられる.

他にも、精度を向上させる方法として、事前情報をよりデータにあったものにするということが考 えられる. 今回はすべてのシーンで同じ、一様な滑らかさの事前情報を与えた. 実際に、この事前情 報はよく機能したが、車載シーンにおいて大きく深度が異なる場合好ましくない. 3 章で述べたよう に、車載というアプリケーションに限定した場合、FOE の性質を利用した事前情報を入れ込むことで 改善できる見込みがある.また、[Dexheimer and Davison, 2023] のように、深度情報を学習し事前 情報として用いることも可能だと考えている.また、物体ごとに運動モデルを考え、領域分割と運動 の推定を同時に行うモーションセグメンテーション [Mitrokhin et al., 2019; Stoffregen et al., 2019; Zhou et al., 2021; Lu et al., 2021] への発展なども考えられる.

## 6 結論

本章では、結論として本論文を総括し、今後の課題と将来展望を示す.

#### 6.1 **まとめ**

本論文では、イベントカメラを用いたオプティカルフロー推定に取り組んだ.従来のフレームベー スのカメラでは、一定の時間間隔の画像から輝度によって対応を測ってオプティカルフローを推定し ていたが、その時間分解能の低さから画素同士の対応づけは原理的に難しい.また、出力の時間的な レートを上げるにつれ、データの冗長性が大きくなり、密な処理による推定手法は計算負荷が大きい という問題がある.それに対してイベントカメラは、輝度の変化のみに反応し、画素ごとに非同期的 にデータを出力するカメラであり、運動の推定において高いポテンシャルを持っている.

2.2.2 章で述べたように、その高い時間分解能から、単にイベントを時空間で平面にフィッティング するだけで、対応づけを測ることができ、局所的な運動の推定が容易である.しかし、局所的な推定 のみでは不定性が残り、窓問題により完全なオプティカルフローを得ることはできない.

3章では、輝度復元を介し、画像ベース同様の輝度一貫性のコストと滑らかさの正則化を加えてオ プティカルフローを推定する手法に着目し、車の運動モデルを活用した正則化を提案した.

4章では、画素同士の対応づけをイベントの時間を用いて直接測り、近傍の画素での滑らかさの正則 化を行うことで、オプティカルフローを推定した.計算は画素ごとに独立して行われるため、並列化 が可能であり、高速化が期待される.しかし、これらの手法はインクリメンタルな処理ではなく、全 画素のイベントのバッチ処理であり、高レートに出力しようとすると再計算が多くなり、計算量が増 大するという問題がある.

それに対して、5章では、非同期的な最適化によってオプティカルフローをインクリメンタルに推 定する手法を提案した.局所的に計算可能なノーマルフローから、オプティカルフローの拘束線方向 に大きな分散を持つ2次元のガウス分布でオプティカルフローの分布をモデリングし、滑らかさの事 前情報を含めたファクターグラフを構築した.このグラフを,非同期的なメッセージ送信によって周 辺化することで,オプティカルフローを推定した.これにより,イベントデータの持つスパース性を 活かし,局所的に運動を推定しつつ,周囲に伝播させることで完全なオプティカルフローを推定する ことを可能とした.また,非同期な局所的な演算を CPU で並列化を行うことでリアルタイム処理を 実現した.

### 6.2 将来展望

プロセッサとアルゴリズムの両面からの発展を目指す必要がある.3章,4章で扱った手法はイベン トを画像のようにして扱った同期的なアルゴリズムであり,GPUを用いた並列計算が適している.し かし,イベントデータはスパースで非同期的な特性を持ったデータであり,その性質を最大限活かす ためには,データ同様にスパースで非同期的なアルゴリズムが必要である.5章で述べた TEGBP は スパースで非同期なアルゴリズムであり,イベントデータに適した手法であると言える.また,並列 化が可能であり,CPUのマルチコアを用いて高速化することができた.しかし,依然としてメモリへ のアクセスがボトルネックとなっている.

この問題を解決するには、メモリを含めた分散処理が必要である.近くにメモリが配置されたプロ セッサを大量に積むことで、メモリへのアクセス時間を短縮しつつ大規模な並列化が可能となる.この ようなプロセッサはすでに開発が進んでおり、GraphCore の Intelligence Processing Unit (IPU)\*1な どがある.CPU、GPU、IPUの違いについて図 6.1 に示す.IPUは、非常に多くの並列プロセッサ と分散メモリを有しており、ニューラルネットワークやグラフの推論に有用である.実際に、Bundle adjustment を IPU上で実装し、高速化された事例もある [Ortiz et al., 2020].本提案手法の TEGBP は、各ノードで信念を持ちエッジで結合された局所的な演算のみを行うため、このようなプロセッサ との親和性が非常に高い、今後はこのようなプロセッサを用いて分散処理を行うことで、より高速な オプティカルフロー推定が可能になると考えられる.

 $<sup>^{*1}</sup>$  https://www.graphcore.ai/



図 6.1 CPU と GPU, IPU の違い.

## 6.3 終わりに

今日のコンピュータビジョンは、GPU などの計算リソースの発展や、巨大なパラメータを持つ ニューラルネットワークの登場により、画像認識の精度が飛躍的に向上した.一方、生物の脳はそれよ り遥かに効率的な処理によって多様なタスクをこなしている.本研究では、生物の脳のような非同期 的な処理を模倣したイベントカメラを活用し、データの特性を活かした低レベルなモデリングで、ス パースに処理することが可能なオプティカルフロー推定アルゴリズムを提案した.このようなスパー スな視覚信号を効率的に処理することができるようなアルゴリズムの開発は、今後のコンピュータビ ジョンにおいて重要であると考えられる.本研究が、その一助となることを願い、本論文を終わる.

# 謝 辞

本研究は,著者が慶應義塾大学大学院理工学研究科後期博士課程在学中に,同大学理工学部電気情報 工学科の青木義満教授の指導のもとに行われました.本論文の執筆にあたり,多くの人からご指導, ご支援いただいたことに感謝いたします.

はじめに、本論文の主査であり、指導教官である青木教授に深く感謝いたします. 学部4年生から修 士課程,博士課程にわたり、多くのご指導をいただきました. 青木研究室で学んだことが今後の研究 活動にとって大きな糧となると信じています. 改めて深く感謝申し上げます. また,研究室で出会っ たメンバーにも感謝いたします.

そして、本研究の副査を快く了承頂きました池原雅章教授、斎藤英雄教授、五十川麻理子准教授に 深く感謝いたします.学位論文の審査を通して、研究の内容だけでなく、研究に対する姿勢、論文の 書き方や発表の仕方まで、幅広く学ばせていただきました.3人の副査の先生方に改めて感謝申し上 げます.

加えて、デンソーアイティーラボラトリの社員一同に感謝いたします.特に、関川雄介博士とは研 究を始めた学部4年の時から共同研究を進めてきました.数々の議論を通して、研究の方向性や論文 の書き方、研究への姿勢など、研究者としての基礎を学ぶことができました.深く感謝申し上げます. 最後に、家族に深く感謝いたします.温かく見守り経済的にも生活を支えてくれた両親、応援して くれていた姉弟に感謝申し上げます.

74

## 参考文献

- Agarwal, P., Tipaldi, G. D., Spinello, L., Stachniss, C., and Burgard, W. (2013). Robust Map Optimization using Dynamic Covariance Scaling. *IEEE International Conference on Robotics* and Automation (ICRA), pages 62–69.
- Akolkar, H., Ieng, S. H., and Benosman, R. (2022). Real-Time High Speed Motion Prediction Using Fast Aperture-Robust Event-Driven Visual Flow. *IEEE Transactions on Pattern Analysis* and Machine Intelligence (TPAMI), 44(1):361–372.
- Aung, M. T., Teo, R., and Orchard, G. (2018). Event-based Plane-fitting Optical Flow for Dynamic Vision Sensors in FPGA. *IEEE International Symposium on Circuits and Systems* (ISCAS), pages 1–5.
- Baldwin, R. W., Almatrafi, M., Asari, V., and Hirakawa, K. (2020). Event Probability Mask (EPM) and Event Denoising Convolutional Neural Network (EDnCNN) for Neuromorphic Cameras. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1710.
- Bardow, P., Davison, A. J., and Leutenegger, S. (2016). Simultaneous Optical Flow and Intensity Estimation from an Event Camera. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 884–892.
- Benosman, R., Clercq, C., Lagorce, X., Sio-Hoi Ieng, and Bartolozzi, C. (2014). Event-Based Visual Flow. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 25(2):407– 417.
- Benosman, R., Ieng, S.-H., Clercq, C., Bartolozzi, C., and Srinivasan, M. (2012). Asynchronous frameless event-based optical flow. *Neural Networks*, 27:32–37.
- Bishop, C. M. (2006). Pattern recognition and machine learning.

- Brandli, C., Berner, R., Yang, M., Liu, S. C., and Delbruck, T. (2014). A 240  $\times$  180 130 dB 3  $\mu$ s latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341.
- Bruhn, A., Weickert, J., and Schnörr, C. (2005). Lucas / Kanade Meets Horn / Schunck : Combining Local and Global Optic Flow Methods. International Journal of Computer Vision, 61(3):211–231.
- Burkitt, A. N. (2006). A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input. *Biological Cybernetics*, 95:1–19.
- Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145.
- Chandra, R., Dagum, L., Kohr, D., Menon, R., Maydan, D., and McDonald, J. (2000). Parallel programming in openmp.
- Chaney, K., Panagopoulou, A., Lee, C., Roy, K., and Daniilidis, K. (2021). Self-Supervised Optical Flow with Spiking Neural Networks and Event Based Cameras. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5892–5899.
- Chen, G., Cao, H., Conradt, J., Tang, H., Rohrbein, F., and Knoll, A. (2020). Event-Based Neuromorphic Vision for Autonomous Driving: A Paradigm Shift for Bio-Inspired Visual Sensing and Perception. *IEEE Signal Processing Magazine*, 37(4):34–49.
- Davies, M., Srinivasa, N., Lin, T. H., Chinya, G., Cao, Y., Choday, S. H., Dimou, G., Joshi, P.,
  Imam, N., Jain, S., Liao, Y., Lin, C. K., Lines, A., Liu, R., Mathaikutty, D., McCoy, S., Paul,
  A., Tse, J., Venkataramanan, G., Weng, Y. H., Wild, A., Yang, Y., and Wang, H. (2018). Loihi:
  A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro*, 38(1):82–99.
- Davison, A. J. (2018). FutureMapping: The Computational Structure of Spatial AI Systems. arXiv.

- Davison, A. J. and Ortiz, J. (2019). FutureMapping 2: Gaussian Belief Propagation for Spatial AI. arXiv.
- Delbruck, T., Hu, Y., and He, Z. (2021). V2E: From video frames to realistic DVS event camera streams. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 1–11.
- Dexheimer, E. and Davison, A. J. (2023). Learning a Depth Covariance Function. *IEEE/CVF* Conference on Computer Vision and Pattern Recognition (CVPR), pages 13122–13131.
- Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazrbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. (2015). FlowNet: Learning Optical Flow with Convolutional Networks Alexey. *IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766.
- Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient Belief Propagation for Early Vision. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8.
- Gallego, G., Gehrig, M., and Scaramuzza, D. (2019). Focus Is All You Need: Loss Functions for Event-Based Vision. In *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), pages 12272–12281.
- Gehrig, M., Aarents, W., Gehrig, D., and Scaramuzza, D. (2021a). DSEC: A Stereo Event Camera Dataset for Driving Scenarios. *IEEE Robotics and Automation Letters*, 6(3):4947–4954.
- Gehrig, M., Millhäusler, M., Gehrig, D., and Scaramuzza, D. (2021b). E-RAFT: Dense Optical Flow from Event Cameras. *International Conference on 3D Vision (3DV)*, pages 197–206.
- Grac, R., Mcreynolds, B., Delbruck, T., Uzh, N., and Zurich, E. T. H. (2023). Shining light on the DVS pixel : A tutorial and discussion about biasing and optimization. *IEEE/CVF Conference* on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 4044–4052.
- Haessig, G., Cassidy, A., Alvarez, R., Benosman, R., and Orchard, G. (2017). Spiking Optical Flow for Event-based Sensors Using IBM's TrueNorth Neurosynaptic System. *IEEE Transac*-

tions on Biomedical Circuits and Systems, 12(4):860 - 870.

- Horn, B. K. B. and Schunck, B. G. (1981). Determining Optical Flow. Artificial Intelligence, 17(1-3):185–203.
- Hu, Z., Uchiyama, K., and Kawaji, S. (1999). Determining Motion Parameters for Vehicle-Mounted Camera Using Focus of Expansion. T. IEE Japan, 119-D(1):50–57.
- Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. The Journal of Phisiology, 148(3):574–591.
- Irani, M., Rousso, B., and Peleg, S. (1994). Recovery of Ego-Motion Using Image Stabilization. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 454–460.
- Khairallah, M. Z., Bonardi, F., Roussel, D., and Bouchafa, S. (2022). PCA Event-Based Optical Flow for Visual Odometry. *IEEE International Conference on Image Processing (ICIP)*, pages 3521–3525.
- Lagorce, X., Orchard, G., Galluppi, F., Shi, B. E., and Benosman, R. B. (2017). HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1346–1359.
- Lee, C., Kosta, A., Zhu, A. Z., Chaney, K., Daniilidis, K., and Roy, K. (2020a). Spike-FlowNet: Event-based Optical Flow Estimation with Energy-Efficient Hybrid Neural Networks. *European Conference on Computer Vision (ECCV)*, pages 1–16.
- Lee, C., Sarwar, S. S., Panda, P., Srinivasan, G., and Roy, K. (2020b). Enabling spike-based backpropagation for training deep neural network architectures. *Frontiers in Neuroscience*, 14:1–26.
- Lichtsteiner, P., Posch, C., Delbruck, T., and Member, S. (2008). A 128 × 128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576.

- Low, W. F., Gao, Z., Xiang, C., and Ramesh, B. (2020). SOFEA: A non-iterative and robust optical flow estimation algorithm for dynamic vision sensors. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 368–377.
- Lu, X., Zhou, Y., and Shen, S. (2021). Event-based Motion Segmentation by Cascaded Two-Level Multi-Model Fitting. IEEE International Conference on Intelligent Robots and Systems (IROS), pages 4445–4452.
- Lucas, B. D. and Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. International joint conference on Artificial intelligence (IJCAI), pages 674–679.
- Maass, W. (1997). Networks of Spiking Neurons : The Third Generation of Neural Network Models. Neural Networks, 10(9):1659–1671.
- Manderscheid, J., Sironi, A., Bourdis, N., Migliore, D., and Lepetit, V. (2019). Speed Invariant Time Surface for Learning to Detect Corner Points With Event-Based Cameras. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10237–10246.
- Meinhardt-Llopis, E., Sánchez Pérez, J., and Kondermann, D. (2013). Horn-Schunck Optical Flow with a Multi-Scale Strategy. *Image Processing On Line*, 3:151–172.
- Merolla, P. A., Arthur, J. V., Alvarez-icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., Jackson, B. L., Imam, N., Guo, C., Nakamura, Y., Brezzo, B., Vo, I., Esser, S. K., Appuswamy, R., Taba, B., Amir, A., Flickner, M. D., Risk, W. P., Manohar, R., and Modha, D. S. (2014).
  A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface. *Science*, 345(6197):668–673.
- Mitrokhin, A., Ye, C., Fermuller, C., Aloimonos, Y., and Delbruck, T. (2019). EV-IMO: Motion Segmentation Dataset and Learning Pipeline for Event Cameras. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6105–6112.

- Mueggler, E., Bartolozzi, C., and Scaramuzza, D. (2017a). Fast Event-based Corner Detection. British Machine Vision Conference, 1:1–11.
- Mueggler, E., Forster, C., Baumli, N., Gallego, G., and Scaramuzza, D. (2015). Lifetime Estimation of Events from Dynamic Vision Sensors. *IEEE International Conference on Robotics and* Automation (ICRA).
- Mueggler, E., Huber, B., and Scaramuzza, D. (2014). Event-based, 6-DOF pose tracking for high-speed maneuvers. *IEEE International Conference on Intelligent Robots and Systems*, pages 2761–2768.
- Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., and Scaramuzza, D. (2017b). The Event-Camera Dataset and Simulator: Event-based Data for Pose Estimation, Visual Odometry, and SLAM. International Journal of Robotics Research, 36(2):142–149.
- Murphy, K., Weiss, Y., and Jordan, M. I. (1999). Loopy Belief Propagation for Approximate Inference: An Empirical Study. Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, pages 467–475.
- Nagata, J. and Aoki, Y. (2022). Self-Supervised Learning of Inlier Events for Event-based Optical Flow. British Machine Vision Conference (BMVC), pages 1–12.
- Nagata, J. and Sekikawa, Y. (2023). Tangentially Elongated Gaussian Belief Propagation for Event-based Incremental Optical Flow Estimation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21940–21949.
- Nagata, J., Sekikawa, Y., and Aoki, Y. (2021). Optical flow estimation by matching time surface with event-based cameras. Sensors (Switzerland), 21(4):1–14.
- Nagata, J., Sekikawa, Y., Hara, K., and Aoki, Y. (2019a). FOE-based Regularization for Optical Flow Estimation from an In-vehicle Event Camera. Proceedings of SPIE - The International Society for Optical Engineering, 11049.

- Nagata, J., Sekikawa, Y., Hara, K., and Aoki, Y. (2019b). FOE-based Regularization for Optical Flow Estimation from an In-vehicle Event Camera. *IEEJ Transactions on Electronics*, *Infomation and Systems*, 139(10):1113–1118.
- Nakayama, K. and Silverman, G. H. (1988). The aperture problem—I. Perception of nonrigidity and motion direction in translating sinusoidal lines. *Vision Research*, 28(6):739–746.
- Nunes, U. M., Member, S., Demiris, Y., and Member, S. (2022). Robust Event-Based Vision Model Estimation by Dispersion Minimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):9561–9573.
- Orchard, G., Benosman, R., Etienne-Cummings, R., and Thakor, N. V. (2013). A spiking neural network architecture for visual motion estimation. *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, (1):298–301.
- Orchard, G. and Etienne-Cummings, R. (2014). Bioinspired visual motion estimation. *Proceedings* of the IEEE, 102(10):1520–1536.
- Ortiz, J., Evans, T., and Davison, A. J. (2021). A visual introduction to Gaussian Belief Propagation. *arXiv*, pages 1–20.
- Ortiz, J., Pupilli, M., Leutenegger, S., and Davison, A. J. (2020). Bundle Adjustment on a Graph Processor. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2413–2422.
- Paredes-Vallés, F. and de Croon, G. C. (2021). Back to Event Basics: Self-Supervised Learning of Image Reconstruction for Event Cameras via Photometric Constancy. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pages 3445–3454.
- Paredes-Vallés, F., Hagenaars, J., and de Croon, G. (2021). Self-Supervised Learning of Event-Based Optical Flow with Spiking Neural Networks. *Advances in Neural Information Processing*

Systems, 34:7167–7179.

- Paredes-valles, F., Scheper, K. Y. W., and de Croon, G. C. H. E. (2020). Unsupervised Learning of a Hierarchical Spiking Neural Network for Optical Flow Estimation : From Events to Global Motion Perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):2051–2064.
- Pearl, J. (1988). Probabilistic reasoning in intelligent systems: Networks of plausible inference.
- Posch, C., Matolin, D., and Wohlgenannt, R. (2011). A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. *IEEE Journal of Solid-State Circuits*, 46(1):259–275.
- Ramesh, B., Yang, H., Orchard, G. M., Le Thi, N. A., Zhang, S., and Xiang, C. (2019). DART: Distribution Aware Retinal Transform for Event-based Cameras. *IEEE Transactions on Pat*tern Analysis and Machine Intelligence, 42(11):2767–2780.
- Ranganathan, A., Kaess, M., and Dellaert, F. (2007). Loopy SAM. International Joint Conference on Artificial Intelligence (IJCAI), pages 2191–2196.
- Rebecq, H., Gehrig, D., and Scaramuzza, D. (2018). ESIM: an Open Event Camera Simulator. Conf. on Robotics Learning (CoRL), 87:969–982.
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention (MIC-CAI), 9351:234–241.
- Rueckauer, B. and Delbruck, T. (2016). Evaluation of Event-Based Algorithms for Optical Flow with Ground-Truth from Inertial Measurement Sensor. *Frontiers in Neuroscience*, 10(176):1– 17.
- Sánchez Pérez, J., Meinhardt-Llopis, E., and Facciolo, G. (2013). TV-L1 Optical Flow Estimation. Image Processing On Line, 3:137–150.

- Sato, I., Yamano, C., and Yanagawa, H. (2011). Crossing obstacle detection with a vehiclemounted camera. *IEEE Intelligent Vehicles Symposium (IV)*, pages 60–65.
- Sekikawa, Y. and Nagata, J. (2023). Live Demonstration: Tangentially Elongated Gaussian Belief Propagation for Event-Based Incremental Optical Flow Estimation. *IEEE Conference* on Computer Vision and Pattern Recognition Workshops, pages 3931–3932.
- Shiba, S., Aoki, Y., and Gallego, G. (2022a). Event Collapse in Contrast Maximization Frameworks. MDPI Sensors, 22(14):1–20.
- Shiba, S., Aoki, Y., and Gallego, G. (2022b). Secrets of Event-Based Optical Flow. European Conference on Computer Vision (ECCV), pages 1–23.
- Shiba, S., Aoki, Y., and Gallego, G. (2023). Fast Event-based Optical Flow Estimation by Triplet Matching. *IEEE Signal Processing Letters*, pages 1–5.
- Sironi, A., Brambilla, M., Bourdis, N., Lagorce, X., and Benosman, R. (2018). HATS: Histograms of Averaged Time Surfaces for Robust Event-based Object Classification. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1731–1740.
- Son, B., Suh, Y., Kim, S., Jung, H., Kim, J. S., Shin, C., Park, K., Lee, K., Park, J., Woo, J., Roh, Y., Lee, H., Wang, Y., Ovsiannikov, I., and Ryu, H. (2017). A 640 × 480 dynamic vision sensor with a 9 μ m pixel and 300Meps address-event representation. *IEEE International* Solid-State Circuits Conference (ISSCC), pages 66–67.
- Stoffregen, T., Gallego, G., Drummond, T., Kleeman, L., and Scaramuzza, D. (2019). Event-Based Motion Segmentation by Motion Compensation. *IEEE/CVF International Conference* on Computer Vision (CVPR), pages 7244–7253.
- Stoffregen, T. and Kleeman, L. (2019). Event Cameras, Contrast Maximization and Reward Functions: an Analysis. *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), pages 12300–12308.

- Stoffregen, T., Scheerlinck, C., Scaramuzza, D., Drummond, T., Barnes, N., Kleeman, L., and Mahony, R. (2020). Reducing the Sim-to-Real Gap for Event Cameras. *European Conference* on Computer Vision (ECCV), pages 534–549.
- Stumpp, D. C., Akolkar, H., George, A. D., and Benosman, R. B. (2022). hARMS: A Hardware Acceleration Architecture for Real-Time Event-Based Optical Flow. *IEEE Access*, 10:58181– 58198.
- Sun, D., Yang, X., Liu, M. Y., and Kautz, J. (2018). PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 8934–8943.
- Teed, Z. and Deng, J. (2020). RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. European Conference on Computer Vision (ECCV), pages 402–419.
- Tian, Y. and Andrade-Cetto, J. (2022). Event Transformer FlowNet for optical flow estimation. British Machine Vision Conference (BMVC), pages 1–13.
- Weiss, Y. and Freeman, W. T. (2000). Correctness of belief propagation in Gaussian graphical models of arbitrary topology. Neural Information Processing Systems (NIPS), pages 673–679.
- Ye, C., Mitrokhin, A., Fermüller, C., Yorke, J. A., and Aloimonos, Y. (2020). Unsupervised Learning of Dense Optical Flow, Depth and Egomotion with Event-Based Sensors. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5831–5838.
- Zach, C., Pock, T., and Bischof, H. (2007). A Duality Based Approach for Realtime TV-L1 Optical Flow. *Pattern Recognition*, pages 214–223.
- Zhou, Y., Gallego, G., Lu, X., Liu, S., and Shen, S. (2021). Event-based Motion Segmentation with Spatio-Temporal Graph Cuts. *IEEE Transactions on Neural Network and Learning* Systems (TNNLS), pages 1–13.
- Zhu, A., Yuan, L., Chaney, K., and Daniilidis, K. (2018a). EV-FlowNet: Self-Supervised Optical

Flow Estimation for Event-based Cameras. *Proceedings of Robotics: Science and Systems*, pages 1–8.

- Zhu, A. Z., Thakur, D., Ozaslan, T., Pfrommer, B., Kumar, V., and Daniilidis, K. (2018b). The Multi Vehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039.
- Zhu, A. Z., Yuan, L., Chaney, K., and Daniilidis, K. (2018c). Unsupervised event-based optical flow using motion compensation. *European Conference on Computer Vision (ECCV) Work*shop, pages 711–714.
- Zhu, A. Z., Yuan, L., Chaney, K., and Daniilidis, K. (2019). Unsupervised Event-based Learning of Optical Flow, Depth, and Egomotion. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 989–997.

## 付録 A FOE を用いた車載カメラの運動解析

本章では, FOE を用いた車載カメラの運動解析について述べる. 3.3 節で, FOE を導入することに よって導いた対応点とカメラの回転運動の関係を表した (3.3) 式の導出を示す.

## A.1 カメラ座標系での対応点

図 3.2 のように、カメラが並進運動  $\mathbf{T} = [T_X, T_Y, T_Z]^\top$ と回転運動  $\omega = [\omega_X, \omega_Y, \omega_Z]^\top$ をした時、 カメラ座標系の点 P(X, Y, Z) の点 P'(X', Y', Z')への移動を考える. この時、点 Pのカメラに対す る相対運動ベクトルは  $\mathbf{T}' = -\mathbf{T}, \omega' = -\omega$ となることから、点 Pと点 P'の関係は以下のように表さ れる.

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{T}'$$
(付録 A.1)

この時, *R* は相対回転行列で,回転角が小さい場合,正弦及び余弦の Taylor 展開の 2 次以上の項が省略できるため,各軸の回転パラメータを用いて以下のように表される.

$$R = \begin{bmatrix} 1 & -\omega_Z & \omega_Y \\ \omega_Z & 1 & -\omega_X \\ -\omega_Y & \omega_X & 1 \end{bmatrix}$$
(付録 A.2)

(付録 A.2) 式を用いて, (付録 A.1) 式を展開することで,以下の式が導かれる.

$$\begin{cases} X' = X + \omega_Y Z - \omega_Z Y - T_X \\ Y' = Y + \omega_Z X - \omega_X Z - T_Y \\ Z' = Z + \omega_Z Y - \omega_Y X - T_Z \end{cases}$$
(付録 A.3)

さらに、右辺の第1項を左辺に移項することで、各方向の変化量は以下の式のように表される.

$$\begin{cases} \Delta X = \omega_Y Z - \omega_Z Y - T_X \\ \Delta Y = \omega_Z X - \omega_X Z - T_Y \\ \Delta Z = \omega_Z Y - \omega_Y X - T_Z \end{cases}$$
(付録 A.4)

## A.2 画像座標系での対応点

次に, 点 P から点 P' への移動する場合の画像平面上の点 p(x,y) から点 p'(x',y') への移動を考える. 画像上の点 p(x,y) は投影変換式から, カメラ座標系の点 P(X,Y,Z) を用いて以下の式で表現できる.

$$\begin{cases} x = f \frac{X}{Z} \\ y = f \frac{Y}{Z} \end{cases}$$
(付録 A.5)

画面上の点 p の移動速度を  $\mathbf{v} = [v_x, v_y]^\top$  とすると、(付録 A.5) 式を微分することで以下の式が導かれる.

$$v_x = \dot{x} = \frac{d}{dt} \left( f \frac{X}{Z} \right) = f \left( \frac{\dot{X}}{Z} - X \frac{\dot{Z}}{Z^2} \right)$$

$$v_y = \dot{y} = \frac{d}{dt} \left( f \frac{Y}{Z} \right) = f \left( \frac{\dot{Y}}{Z} - Y \frac{\dot{Z}}{Z^2} \right)$$
(付録 A.6)

ここで,サンプリング間隔が十分小さいと仮定し微分値を差分値で置き換えることで,フレーム間の 画像上の対応点の関係は以下のようになる.

$$\begin{cases} x' - x = f\left(\frac{\Delta X}{Z} - X\frac{\Delta Z}{Z^2}\right) \\ y' - y = f\left(\frac{\Delta Y}{Z} - Y\frac{\Delta Z}{Z^2}\right) \end{cases}$$
(付録 A.7)

この (付録 A.7) 式に, 自車の運動によるカメラ座標系での差分値 (付録 A.4) 式と FOE の定義式の (3.2) 式を代入し,  $T_Z/Z \neq 0$ を仮定して整理することで,以下の式 (3章 (3.3) 式と同じ)を得る.

$$\frac{(x'-x) - \left(-\frac{xy}{f}\omega_X - \frac{f^2 + x^2}{f}\omega_Y - y\omega_Z\right)}{(y'-y) - \left(-\frac{f^2 + y^2}{f}\omega_X - \frac{xy}{f}\omega_Y - x\omega_Z\right)} = \frac{x - x_0}{y - y_0}$$
(付録 A.8)

この式は, FOE の座標と回転運動パラメータ,対応点座標の3つで構成されている. つまり,あらか じめ FOE を求めておき,対応点を推定することで,回転パラメータを推定することができる.

## 付録 B ガウシアン信念伝搬法

本章では、5章の理解の助けのため、ガウシアン信念伝搬法について説明する.より詳細な説明は [Davison and Ortiz, 2019; Ortiz et al., 2021] を参照されたい. 信念伝搬法は、ファクターグラフの ノード間でメッセージを交換することで、近似的にベイズ確率推論を行う手法である. ファクターグ ラフとは、確率モデルをグラフ構造で表したもので、ノードは確率変数、エッジは確率変数間の依存関 係を表す. ガウシアン信念伝搬法は、全てのファクターがガウス分布である場合の信念伝搬法である. ここでは、簡単のため線形化した後のファクターについて説明する.

## B.1 ファクターグラフ

ファクターグラフとは、確率モデルをグラフ構造で表したもので、ノードは確率変数、エッジは確 率変数間の依存関係を表す.ファクターグラフG = (X, F, E)は、ノード集合 $X = \{x_i\}_{i=1,\dots,N_x}$ 、 ファクター集合 $F = \{f_s\}_{s=1,\dots,N_f}$ 、エッジ集合Eからなる.それぞれのファクター $f_s$ は、変数集合 の部分集合 $x_s \subset X$ を引数に持つ関数 $f_s(X_s)$ で、確率的な依存関係を表現する.ファクターグラフ は、ファクターノードをそれぞれが依存する変数ノードとエッジで陽に結ぶことで因子分解を表現し ている.よって、変数の同時分布は以下のように独立な項の積で表現できる.

$$p(X) = \prod_{s=1}^{N_f} f_s(X_s) \tag{dds B.1}$$

## B.2 信念伝搬法

信念伝搬法は,同時分布の周辺化を行うための分散推論アルゴリズムである [Pearl, 1988; Bishop, 2006]. ある一つの変数 x<sub>i</sub> についての周辺分布は,同時分布を他の全ての変数について積分することで得られる.

$$p(x_i) = \int p(X) dx_1 \cdots dx_{i-1} dx_{i+1} \cdots dx_{N_x}$$
 (figs B.2)



図付録 B.1 変数ノード x に  $f_s$  を含むファクターノードの結合.



図付録 B.2 ファクター  $f_s$  と変数 x, M 個の他の変数の結合.

信念伝搬法はファクターグラフ上で局所的な計算によるメッセージ交換をすることで効率的に周辺分 布を計算する.

グラフ構造が木構造である場合,1スイープ (根ノードから葉ノードへの往復) で全ての変数につい て周辺分布を計算することができる.グラフにループが存在する場合,収束するまでメッセージ交換 を繰り返す.ループあり信念伝搬法は,収束性が保証されていないが,実際には収束することが多い [Murphy et al., 1999].分布がガウシアンである場合,ループありガウシアン信念伝搬法は正しい周 辺事後平均に収束することが知られている [Weiss and Freeman, 2000].

## B.3 メッセージ交換

ある変数 x についての周辺分布は,同時分布を他の全ての変数について足し合わせることで得られる.

$$p(x) = \sum_{X \setminus x} p(X) \tag{diff B.3}$$

ここで, *X*\*x* は *X* のうち *x* を除いた集合である. 図付録 B.1 のような, 複数のファクターグラフ *f<sub>s</sub>* が直接結合されているある変数ノード *x* に着目する. その他のファクターは, 直接結合されている ファクターを通じて, 間接的に *x* に結合されている. よって, グラフ全体を *f<sub>s</sub>* と同じ数の部分集合と して表現することができる. 全体の同時分布はそれらの積によって表現できる.

$$p(X) = \prod_{s \in n(x)} F_s(x, X_s)$$
 (付録 B.4)

ここで, n(x) は変数ノード x に結合されているファクターの集合である.  $F_s$  は  $f_s$  に関係する全ての ファクターの積を表している.  $X_s$  は  $f_s$  を通じて x に結合されている変数の集合である. (付録 B.3) 式と (付録 B.4) 式を組み合わせることで以下を得る.

$$p(x) = \sum_{X \setminus x} \left[ \prod_{s \in n(x)} F_s(x, X_s) \right]$$
(付録 B.5)

ここで、sum と product の順序を入れ替えることができる.

$$p(x) = \prod_{s \in n(x)} \left[ \sum_{X_s} F_s(x, X_s) \right]$$
( $\mathfrak{f}\mathfrak{g} \operatorname{B.6}$ )

この順序の入れ替えが非常に重要で,先に sum(周辺化) を行うことで,xの枝のファクターの積を計 算するだけで周辺分布を計算することができる.

ファクターノード  $f_s$  から変数ノード x へのメッセージ  $\mu_{f_s \to x}(x)$  を以下のように定義する.

$$\mu_{f_s \to x}(x) = \sum_{X_s} F_s(x, X_s) \tag{fdd B.7}$$

メッセージは、その木構造の一つの枝の全てのファクターについて考慮した結果の *x* の周辺分布であ る.このようなメッセージを全ての枝から受け取ることで、変数ノード *x* についての周辺分布を計算 することができる.これを再帰的に繰り返すことで、全ての変数ノードについての周辺分布を計算す ることができる.

メッセージ交換の手続きは以下の3つである.

• 最終的な周辺分布は、ファクターノードから変数ノードへのメッセージの積で表現される.

$$p(x) = \prod_{s \in n(x)} \mu_{f_s \to x}(x) \tag{fdd B.8}$$

 ファクターノードから変数ノードへのメッセージは、変数ノードからファクターノードへの メッセージの積を集め、行き先以外の変数で周辺化することで計算できる。

$$\mu_{f_s \to x}(x) = \sum_{x_1, \dots, x_M} f_s(x, x_1, \dots, x_M) \prod_{m \in n(f_s)} \mu_{x_m \to f_s}(x_m)$$
(付録 B.9)

 変数ノードからファクターノードへのメッセージは、行き先のファクターノード以外の全ての ファクターノードからのメッセージの積で計算できる。

$$\mu_{x_m \to f_s} \left( x_m \right) = \prod_{l \in n(x_m) \setminus f_s} \mu_{f_l \to x_m} \left( x_m \right) \tag{figs B.10}$$

## B.4 ガウシアン信念伝搬法

#### 変数ノードでのメッセージ送信

ガウシアン信念伝搬法は,全てのファクターがガウス分布である場合の信念伝搬法である.この時, メッセージは情報ベクトルと精度行列の組で表現される.(付録 B.10)式は,和を用いて以下のように 表現できる.

$$\boldsymbol{\eta}_{ms} = \sum_{l \in n(x_m) \setminus f_s} \boldsymbol{\eta}_{ml} \tag{fds B.11}$$

$$\Lambda_{ms} = \sum_{l \in n(x_m) \setminus f_s} \Lambda_{ml} \tag{fds B.12}$$

#### ファクターノードでのメッセージ送信

ファクターノードは、変数ノードからメッセージを受け取り、ターゲットの変数ノードへのメッセージを計算する. (付録 B.9) 式に従って、入ってきたメッセージの積を計算し、ファクターの分布自身を掛け合わせる. そして、出力変数以外の全ての変数について積分を行って周辺化し、メッセージとする.

ここで, m<sub>1</sub> が入力ノード, m<sub>2</sub> が出力ノードである, 2 つの変数が結合されているファクターを考える.

$$\mathbf{x}_{s} = \begin{pmatrix} \mathbf{x}_{sm_{1}} \\ \mathbf{x}_{sm_{2}} \end{pmatrix} \tag{fdde B.13}$$

また、情報ベクトルと精度行列を以下のように分割する.

$$\eta_s = \left( \begin{array}{c} \eta_{sm_1} \\ \eta_{sm_2} \end{array} \right)$$
(付録 B.14)

$$\Lambda_s = \begin{pmatrix} \Lambda_{sm_1m_1} & \Lambda_{sm_1m_2} \\ \Lambda_{sm_2m_1} & \Lambda_{sm_2m_2} \end{pmatrix}$$
(付録 B.15)

入力ノード m1 からメッセージが入ってきたとすると以下のようになる.

$$\boldsymbol{\eta}_{Cs} = \begin{pmatrix} \boldsymbol{\eta}_{sm_1} + \boldsymbol{\eta}_{m_1s} \\ \boldsymbol{\eta}_{sm_2} \end{pmatrix}$$
(付録 B.16)

$$\Lambda_{Cs} = \begin{pmatrix} \Lambda_{sm_1m_1} + \Lambda_{m_1s} & \Lambda_{sm_1m_2} \\ \Lambda_{sm_2m_1} & \Lambda_{sm_2m_2} \end{pmatrix}$$
(付録 B.17)

ファクターノードでのメッセージ送信を行うには, (付録 B.16) 式と (付録 B.17) 式で表される同時分 布から,出力ノード以外の全ての変数について積分を行う必要がある.ここで,分割したガウス分布 の周辺化は次のように計算できる.同時分布を以下のように分割する.

$$\eta = \begin{pmatrix} \eta_{\alpha} \\ \eta_{\beta} \end{pmatrix}$$
( $\mathfrak{f}\mathfrak{g} \mathsf{B}.18$ )  
$$\Lambda = \begin{bmatrix} \Lambda_{\alpha\alpha} & \Lambda_{\alpha\beta} \\ \Lambda_{\beta\alpha} & \Lambda_{\beta\beta} \end{bmatrix}$$
( $\mathfrak{f}\mathfrak{g} \mathsf{B}.19$ )

出力変数を上に並び替え, β について周辺化し, α のみの分布を得る.

$$\boldsymbol{\eta}_{M\alpha} = \boldsymbol{\eta}_{\alpha} - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\boldsymbol{\eta}_{\beta} \tag{fds B.20}$$

$$\Lambda_{M\alpha} = \Lambda_{\alpha\alpha} - \Lambda_{\alpha\beta} \Lambda_{\beta\beta}^{-1} \Lambda_{\beta\alpha} \tag{fdg B.21}$$

これにより,ファクターノード  $f_s$ の出力変数ノード  $m_2$ へのメッセージ  $\mu_{f_s \to m_2}$ の精度行列と情報ベクトルが得られた.