

# フレームバッファの不要な 1-pass 線分抽出法

清水 嘉泰<sup>†</sup> 秋山 隼哉<sup>††</sup> 渡邊 孝信<sup>†</sup>  
<sup>†</sup> 早稲田大学大学院 基幹理工学研究科 <sup>††</sup> 早稲田大学 基幹理工学部

## 1. はじめに

画像中の線分情報は消失点検出, ステレオマッチング, SLAM といった高位な画像処理に利用されており, 現在までに様々な線分抽出法が提案されている. しかしながら, 組込みシステムでのリアルタイム処理を考えた場合, 既存手法はメモリ使用量および計算量の双方で依然問題を抱えている. 本稿では, フレームバッファの不要な 1-pass 線分抽出法を提案し, メモリ使用量および計算量の双方を改善したことを示す.

## 2. 提案手法

提案手法は, 図 1 に示す 4 つの処理からなり, 一連の処理が 1-pass 処理として実装可能である.

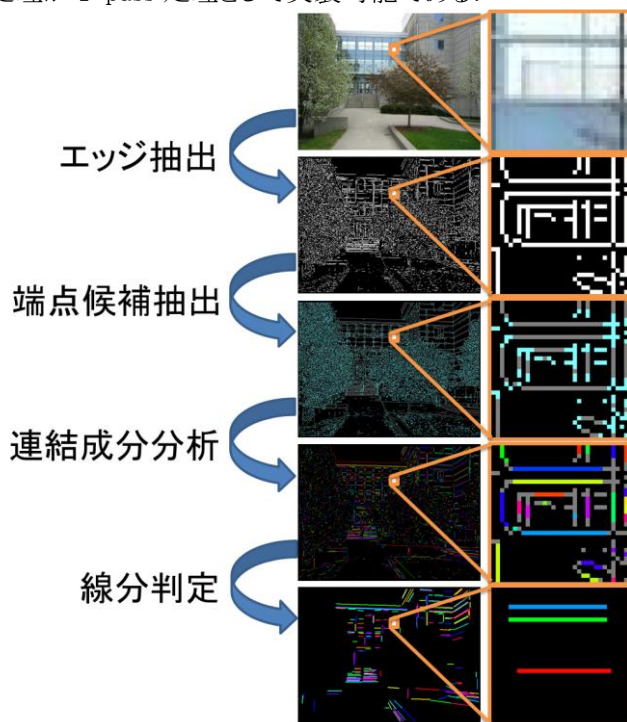


図1 提案手法の処理フロー

まず, エッジ抽出処理で入力画像中のエッジを 1 画素の線幅で抽出する. 端点候補抽出処理ではエッジ抽出で得られたエッジ画素の内, 線分の端点となりうる画素を端点候補として抽出する. 連結成分分析処理では, 端点候補画素で挟まれたエッジ画素をラベル付けていき, ラベル付けが完了したエッジ領域を仮線分として抽出する. 線分判定処理では, 連結成分分析で得られた仮線分が線分として相応しいかどうか判定する. 短すぎたり, 折れ曲がっていたりする仮線分が排除され, 長く真っ直ぐ伸びた仮線分のみが線分として抽出される.

## 3. 実験結果

今回は TESTIMAGES GRAY のテスト画像群 (100 枚, 1200×1200) を用いて提案手法と既存手法の比較実験を行なった. メモリ使用量 (推定値) と平均計算時間の結果を表 1 に, 線分抽出結果の一例を図 2 に示す.

図 2 の線分抽出結果を見ると, 提案手法は既存手法よりも抽出される線分数が若干少ないものの, 木や枝の線分をほとんど抽出していない. 消失点検出などへの応用を考えた場合, 木などの構造物でない線分は極力抽出されないことが望まれる. したがって, 図 2 の結果は, 提案手法が高位な画像処理への応用に適していることを示唆している. また, 表 1 を見ると提案手法が既存手法の 1/72 倍以下のメモリ使用量, 2.4 倍以上の高速化を達成していることが分かる.

表 1 メモリ使用量と平均計算時間 (TESTIMAGES GRAY)

	LSD[1]	EDLines[2]	提案手法
メモリ使用量	11.4 MiB	5.5 MiB	77.9 KiB
平均計算時間	311.9 ms	44.8 ms	18.6 ms

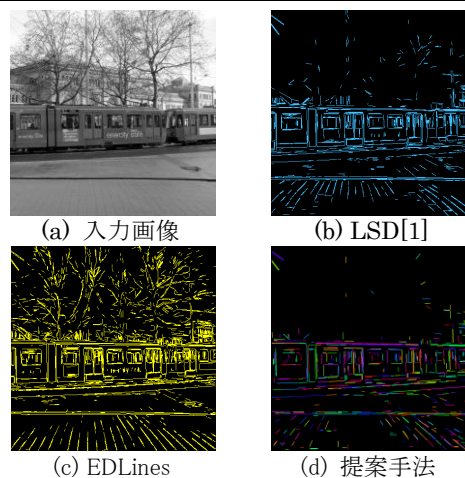


図 2 各手法の線分抽出結果 (GRAY\_OR\_1200x1200\_018)

## 4. まとめ

今回, フレームバッファの不要な 1-pass 線分抽出法を提案した. 提案手法は既存手法と比べて実用的な線分抽出精度を有しており, 1/72 倍以下のメモリ使用量, 2.4 倍以上の高速化を達成した. 今後は, 提案手法を用いた高位な画像処理の研究を行う予定である.

## 参考文献

- [1] R. G. von Gioi *et al.*, "LSD: A Fast Line Segment Detector with a False Detection Control," *IEEE trans. PAMI*, 32(4), pp. 722–732, 2010.
- [2] C. Akinlar *et al.*, "Edlines: Real-time line segment detection by Edge Drawing (ed)," *ICIP*, pp. 2837–2840, 2011.