

GPGPU 向け 2 値画像の連結成分抽出と同時穴埋め処理の高速化手法の提案と実装

木綱 啓人† 佐藤 裕幸†

岩手県立大学大学院 ソフトウェア情報学研究科†

1. はじめに

連結成分抽出は画像処理において最も基本的なアルゴリズムの一つであり、古くから様々なアルゴリズム、アーキテクチャでの実装が提案されている [1]. 連結成分抽出処理は、同じ値を持つ連結画素を抽出する. この用途は、光学画像を用いた製品の異常診断やノイズ領域の抽出である. 特に、2 値化後のノイズ除去処理には穴埋め処理が用いられるが、これは画像から本来製品である背景成分を前景成分として補完抽出するものである. この処理概要は、背景成分の連結成分を抽出し、面積の小さな成分を前景化後、前景成分の連結成分を抽出することでノイズ除去済みの前景成分を抽出できる. しかし、穴埋め処理は他の画像処理フィルタと比べ低速な連結成分抽出を 2 回行う必要があり、リアルタイム処理においてボトルネックとなりやすい. 本稿では、前・背景を同時に抽出した連結成分を用いた穴埋め処理についての GPGPU 向けの並列アルゴリズムを提案、実装し、検証した.

2. 利用技術

連結成分抽出とは、True (1) と False (0) の値を持つ 2 値画像に対して、同一画素値を持つ連結した画素群には 1 以上のユニークなラベル ID を、それ以外の画素には 0 のラベル ID を付加し、ラベル画像を出力する処理である.

この連結成分抽出アルゴリズムには大きくラスタ型と伝播型の 2 つが存在する. なお本研究では 4 近傍連結を前提として説明する.

ラスタ型は 2 値画像を左上端から逐次走査し、注目画素の 2 近傍の中で最も小さなラベル ID を対象画素に貼る. これによりすべての画素へラベル付けが完了した後、Look-Up Table (以後、LUT) を用いてラベル ID 同士の同一性を評価し、ラベルを書換える. 従って、2 回の走査で連結成分を抽出できる.

伝播型は、ラベル画像の全画素へ配列インデックスを格納して初期化し、注目画素の 4 近傍を探索し、最小なラベル ID が存在する場合、注目画素値をそのラベル ID へ更新する.

ラスタ型は伝播型より相対的に計算量は低いが、逐次走査する必要がある、単純な並列性はない. 対して伝播型は最小のラベル ID を各成分内で伝播す

る必要があるため、6-8 回程度反復処理する必要があるが、すべての処理が画素単位の並列性を持つため、GPGPU に適している.

物体領域推定のために、その物体の色特徴等を用い入力画像を 2 値化し、マスクを生成することがある. しかし、このような単純な 2 値化処理では抽出出来ない物体領域(欠損)が存在する. それらを補完するために穴埋め処理が用いられる. 欠損領域は外光等の色ノイズにより物体の内側に小さな穴のような形状として存在するため、その欠損領域(=背景)を前景化することで物体領域を補完する. この穴埋め処理は、背景領域の連結成分を抽出し、背景成分の面積ヒストグラムを生成し、小さい背景成分の前景化により完了する.

3. 提案手法

3.1. 前背景同時連結成分抽出

初期化処理では、前背景同時に連結成分抽出するため、ラベル画像周辺 1px を除くすべての画素に配列インデックスを格納する. 前背景同時伝播処理における伝播処理の相違点は大きく 2 つある. 1 つ目は注目画素が背景なら連結判定しないが、ここでは前背景を同時抽出するため、注目画素値に関係なく連結性を判定する. 2 つ目は、注目・近傍画素との値の同一性判定をした後に連結性判定を行う.

3.2. 前背景連結成分を用いた穴埋め処理

説明のために、前背景のラベルの関係を簡略化した図 1 を用いて説明する. (a) は前背景の連結成分のラベル画像であり、これを初期値とする. 図にあ

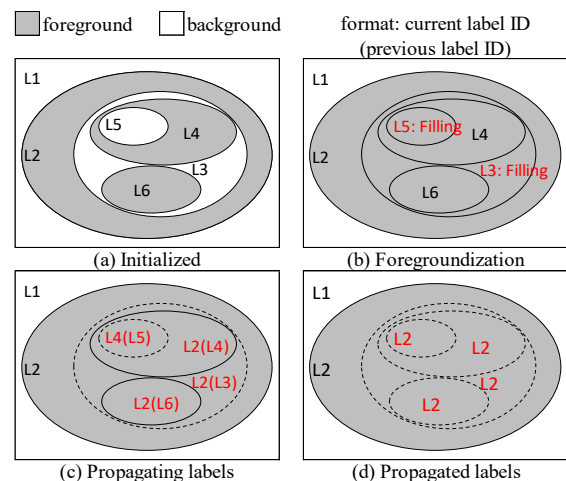


図 1 前背景のラベルの関係の簡略図

Proposal and implementation of the connected-component labeling of binary images and filling holes for GPGPU
Hiroto Kizuna† Hiroyuki Sato†
† Iwate Prefectural University

るLを接頭辞とするインデックスは各成分のラベルIDである。本来、穴埋め処理では背景成分を前景化するが、(b)のL5のように前背景成分を用いた穴埋めでは、背景の近傍成分は必ず前景である。したがって背景成分を前景化した後に、前景成分のラベルIDを統合する必要がある。本処理手順は大きく、前背景成分の面積算出及び小成分の抽出、小さい背景成分の前景化、前背景ラベルの統合探索から成る。

3.3. 前背景連結成分での穴埋め処理の最適化

3.2. では前背景連結成分を用いた穴埋め処理を説明した。そこでは、前景化された背景成分は必ず周辺の前景成分と連結する。この探索処理オーダは入力画像全画素であったが、文献[2]により各連結成分のラベルID集合の取得により、オーダを前背景成分数へ削減できた。この削減手法を説明する。

前景成分の統合は画像の全画素の近傍との連結性を探索し、より小さな前景ラベルIDを伝播することで実現する。この伝播するラベルIDは連結成分抽出時の非連続なラベルIDを用いる。このラベルIDは配列インデックスであり、各成分のラベルIDは各成分の左上に存在する画素の配列インデックスである。したがって、前景化する成分と統合すべき近傍成分は、注目成分の配列インデックスの上か左側に存在する。今まで全画素を探索していたが、全成分のラベルIDの上と左側のみ探索すれば、必ずすべての成分同士の連結性を探索することが出来る。これにより、逐次処理時は全ラベル数Ln回の探索で、並列実行時でも、全ラベル分を2回探索すれば統合できることを確認した。

4. 検証評価

検証にはNVIDIA社のJetson TX2[3]を用い、GPU実装には同社が提供するCUDA環境で実装した。また、CPU上で2方式の連結成分抽出を実装したところ、伝播型よりラスタ型の方が高速だったため、ラスタ型を採用した。5種類の画像サイズに変換した葉画像5枚と人工画像2枚を検証用画像として用いる。

4.1. 穴埋め処理の処理時間と高速化

提案手法をTX2上で実行した際の処理時間を表1に示す。処理時間は各画像に対して穴埋め処理を100回実行した平均である。葉画像の処理時間の標準偏差は最大で約5.5%であり、ゆらぎが十分に小さいため、葉画像の処理時間は全5種類の平均を載せている。各画像の画素数は前後で4倍ずつ大きくなっており、概ね画素数に処理時間が比例していることがわかる。提案手法による高速化率は、葉画像においては約26-46%程度、人工画像においても、Circleは約20-28%、Rectangleは約8-15%程度処理時間を改善することが出来た。

表1から画像の種類ごとの処理時間の間には約20-60%前後の差があり、これは連結成分抽出時間の変動によるものである。この原因として大きく2つ

表1 CPU と GPU 上での穴埋め処理の計測時間(us)

Method	Image size	Leaf Avg.	Circle	Rectangle
Conventional Method on GPU	480x270	3,741	3,461	3,933
	960x540	10,242	11,023	8,730
	1920x1080	30,336	26,858	23,968
	3840x2160	104,895	99,272	85,397
	7680x4320	392,934	412,114	327,606
Conventional Method on GPU	480x270	2,157	2,113	2,291
	960x540	7,837	7,656	8,294
	1920x1080	28,006	26,719	30,424
	3840x2160	107,833	103,982	123,516
	7680x4320	431,381	419,810	495,925
Proposed Method on GPU	480x270	1,719	1,760	2,091
	960x540	5,653	6,048	7,742
	1920x1080	19,750	20,844	26,582
	3840x2160	76,049	83,898	123,941
	7680x4320	295,810	339,748	539,600

の要因が推測できる。1つにGPUで同時に動作するスレッドが処理するラベル画像に含まれるラベルの数が多く、配列参照時のキャッシュヒット率が低下した点、2つに、Rectangleの各成分のリング幅は1pxであり、ラベルIDを1方向へしか伝搬できない点が性能劣化の原因として考えられる。

5. まとめ

本稿では、前背景を同時抽出した連結成分を使った穴埋め処理の並列アルゴリズム、その最適化手法を提案した。前背景成分を同時抽出することで2回の連結成分抽出を1回へ削減し、深刻なSIMD演算におけるコアのアイドル時間を削減した。また、その前背景連結成分を用いた穴埋め処理と探索計算量の削減手法を提案した。これらにより、TX2において、従来手法のGPU実行時と比べて実践的な葉画像の処理時間を約1.26-1.46倍高速化した。加えて、従来手法のCPU実装に画像を入力した際の処理時間と比較すると、約1.33-2.18倍の高速化を実現した。

我々は穴埋め処理における無駄なアイドルに着目し、それらを削減、補完する手法を提案し、従来手法よりも高速な穴埋め処理を実現することができた。

参考文献

- [1] L. He, Q. Gao, X. Zhao, et al, "The connected-component labeling problem: A review of state-of-the-art algorithms," Pattern Recognition 70: 25-43, 2017.
- [2] 木網啓人, 佐藤裕幸: 入退室管理システムにおける人物特定精度向上に向けた顔検出の高速化, 電子情報通信学会技術研究報告, Vol.117, No.484, pp.229-234 (2018).
- [3] NVIDIA: NVIDIA JETSON The embedded platform for autonomous everything (online), available from <<https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/>> (accessed 2019-01-07).