

I-049 多様なカメラ配置の多眼ステレオ法に対応可能な並列処理手法

A Parallelization Method for Multiview Stereo with Various Camera Configurations

福士 将†

Masaru Fukushi†

関口 直紀†

Naoki Sekiguchi†

阿部 亨‡

Toru Abe‡

1 はじめに

複数の画像を用い撮影対象の三次元形状を復元する手法の一つにボクセルベースの多眼ステレオ法がある [1]. この手法では, 図 1 に示すように, (1) 三次元空間に仮想的なボクセル空間 V を設定し, (2) V 内のボクセル v を各画像 $I_n (n = 1 \sim N)$ へ投影した箇所 $r_n(v)$ の類似度 $c(v)$ を求め, (3) 各 v の $c(v)$ に基づき物体表面に対応するボクセル (物体境界ボクセル) を判定することで, 物体の三次元形状を復元している. ボクセルベースの手法は, 多数の画像の情報を効果的に利用できるものの, 復元精度向上のため, ボクセル状態の相互参照を伴う判定処理 [2, 3] を導入すると, 膨大な処理時間が必要となる.

これに対し, V を複数の領域に分割し, 複数台の PE (processing element) で並列に処理する高速化手法が提案されている [4]. この手法は, 実際のカメラ配置からボクセルの参照関係を推定しボクセル空間の分割を行うため, PE 間通信の少ない効率的な並列処理が可能である. しかし, この手法では, 各視点 o_n が直線上に並ぶカメラ配置を前提としており, これを満たさない場合, PE 間通信が増加し並列処理の効率が大きく低下するという問題がある.

本稿では, この問題に対処するため, 視点対を単位としたボクセル空間分割に基づく新たな並列処理手法を提案する. 提案手法は, o_n が直線上に並ばない場合でも PE 間通信の増加を抑えることができ, 多様なカメラ配置に対し効率的な並列処理の実現を可能としている.

2 物体境界ボクセルの判定

まず, 物体境界ボクセルの判定法について説明する. 図 1 に示すように, 物体境界である v_1 の場合, 物体表面の同じ箇所が各 $r_n(v_1)$ へ投影されるため, その類似度 $c(v_1)$ は高く, 物体境界でない v_2 の場合は, 異なる箇所が各 $r_n(v_2)$ へ投影されるため $c(v_2)$ は低い. 従って, $c(v)$ に対する閾値処理により, v が物体境界ボクセルか否かの判

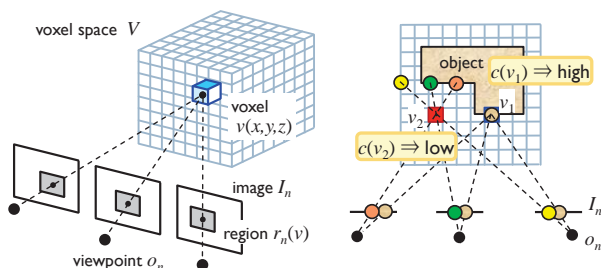


図 1 ボクセルベースの多眼ステレオ法

† 東北大学大学院情報科学研究科

‡ 東北大学サイバーサイエンスセンター

定が可能となる. しかし, 一部の画像で物体に遮蔽が生じる, 物体表面の異なる箇所が同じ輝度値となる等, $c(v)$ のみでは v を正しく判定できない場合があるため, 幾何学的拘束を導入し判定精度の向上を図る手法が提案されている [3]. この手法では, v と o_n を通る直線 $\phi_n(v)$ 上には o_n から可視となる物体境界ボクセルが最大 1 個しか存在しない (図 2 左に示すように, v が可視なら v より前に物体は存在せず, v が物体なら v より後は遮蔽され不可視となる) ことを拘束条件として用い, この条件が満たされるよう, 物体境界ボクセルの最終的な判定が行われる.

3 視点対を単位としたボクセル空間分割

ボクセル判定に幾何学的拘束を導入する場合は, 各 v が物体境界ボクセルである確からしさ $e(v)$ を $c(v)$ から求め, 二つの処理 — (1) $\phi_n(v)$ 上で v が拘束条件を満たすために必要となる $e(v)$ の修正量 $d_n(v)$ を求める処理, (2) $d_n(v)$ を全 n で統合し $e(v)$ を修正する処理 — を反復する [3]. その際, $d_n(v)$ を求める処理 (1) では, v の状態 $e(v)$ に加え, $\phi_n(v)$ 上の他のボクセルの状態も必要となるため, ボクセル間にデータ (状態) の参照関係が生じる.

この参照関係を考慮し, 従来ボクセル空間分割法 (GDD: Gradient-dependent Domain Division) [4] では, 図 2 右に示すように, o_n の並びを近似した直線を通る平面で V の分割を行い, 各領域を異なる PE へ割り当てている. これにより, 相互参照が必要な $\Phi(v) = \bigcup_{n=1}^N \phi_n(v)$ 上のボクセルを同一 PE へ割り当て, PE 間通信の削減を図っている. しかし, o_n が直線上にない場合は, $\Phi(v)$ が同一平面上に存在するとは限らず, GDD では, $\Phi(v)$ 上のボクセルが異なる PE へ割り当てられるため, PE 間通信が増加し並列処理の効率が低下するという問題が発生する.

この問題に対処するため, 本稿では, 視点対を単位としたボクセル空間分割 (VGDD: Viewpoint-pair based Gradient-dependent Domain Division) を提案する. VGDD では, 図 3 に示すように, 視点 N 個を $N/2$ 組の対に分け, 視点対毎に, GDD による V の分割を行う. 具体的には, P 台の PE で並列処理を行う場合, 各視点対を通る平面で

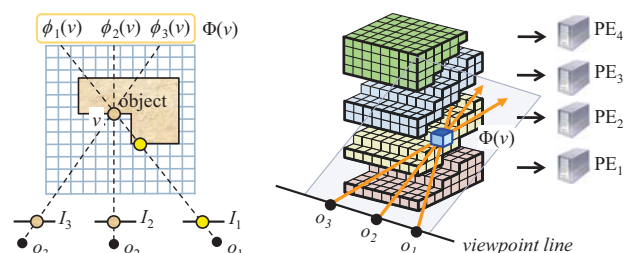


図 2 ボクセル判定への幾何学的拘束の導入 (左) と GDD (右)

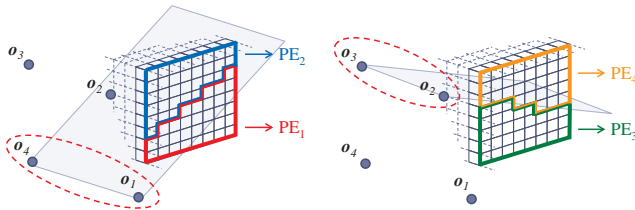


図3 視点対を単位としたボクセル空間分割 (VGDD)

V を P 個の領域に分割し, 得られた全 $P \times N/2$ 個の領域を $N/2$ 個ずつ各 PE へ割り当てる.

従来の GDD とは異なり, VGDD では, 視点対 o_m, o_n に対し処理 (1) が実行される. 従って, 相互参照が必要な $\phi_m(v) \cup \phi_n(v)$ 上のボクセルは常に同一平面上に存在し, それらを同じ PE へ割り当てることができるため, カメラ配置によらず, 処理 (1) で生じる PE 間通信の大幅な削減が可能となる. 一方, 処理 (2) に関しては, GDD では, 各 v に対する N 個の $d_n(v)$ が同じ PE で計算されるのに対し, VGDD では, $d_n(v)$ が最大 $N/2$ 個の異なる PE で別々に計算されるため, $d_n(v)$ を統合する際に新たな PE 間通信が発生する. しかし, この PE 間通信は, 処理 (1) のものに比べて通信量が少なく, また, バタフライ型の通信を行うことで効率的に実現できるため, 全体としては PE 間通信の増加を抑えることができ, VGDD により, 多様なカメラ配置に対しても効率的な並列処理が実現できる.

4 評価結果

提案手法 (VGDD) の効果を検証するために, GDD と VGDD を用い, ボクセルベースの多眼ステレオ法により三次元形状を復元する評価実験を行った. 実験は, PC クラスタ上でを行い, 入力として, Seitz らが公開しているデータセット Temple [5] から 8 枚 1 組の画像セット 3 組 (a), (b), (c) を選択した. PC クラスタの仕様を表 1 に, 各画像セットでの視点位置を図 5 に示す. 図 5 から分かるように, 画像セット (a)<(b)<(c) の順で, 視点の並びが直線から外れる程度が大きくなっている. また, 設定したボクセル空間のサイズは $100 \times 100 \times 100$ ボクセルである (1 ボクセルは $1.1\text{mm} \times 1.7\text{mm} \times 0.9\text{mm}$).

図 6 に, PE 数を変えて並列処理を行った場合の処理時間, および, 速度向上比 (PE 数 1 での処理時間/PE 数 P での処理時間) を示す. この結果から, GDD を用いた場合は, PE 数が増えると, 画像セット (a)>(b)>(c) の順で速度向上比が大きくなり低下していることが分かる. これは, 視点の並びが直線から外れるほど, 相互参照の必要なボクセルが異なる PE に割り当てられる場合が多くなり, PE 間通信が増加したためである. これに対し, VGDD を用いた場合は, PE 数が増えても, 各画像セットでの速度向上比

表 1 PC クラスタの仕様

| | |
|----------|---|
| Node | 8 nodes connected with 1Gbps Ethernet (each node has two PEs) |
| PE (CPU) | AMD Opteron 2.0GHz |
| Memory | PC3200 ECC Registered, 2GB |
| OS | Cent OS 4.4 |



図4 入力画像の例 (画像セット (a))

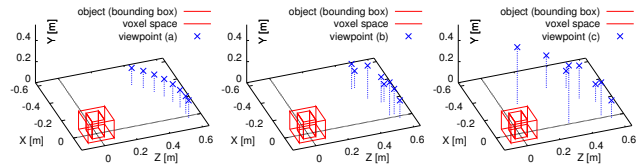


図5 画像セット (a), (b), (c) での視点位置

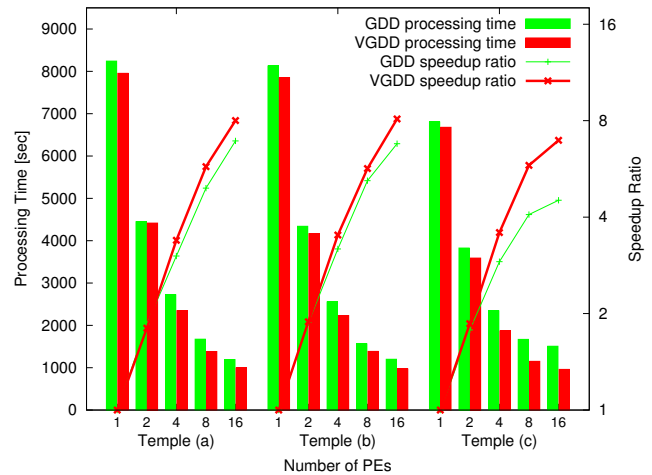


図6 各 PE 数での処理速度と速度向上比 (GDD, VGDD)

に大きな違いは生じていない. また, GDD を用いた場合に比べ, 処理速度自体も削減されている. これは, VGDD を用いることで, カメラ配置によらず, PE 間通信が効果的に削減され並列処理が効率的に行われた結果であると考えられる.

5 おわりに

本稿では, ボクセルベースの多眼ステレオ法の高速度を図るため, 視点対を単位としたボクセル空間分割に基づく新たな並列処理手法 (VGDD) を提案した. PC クラスタを用いた評価実験から, VGDD により, 多様なカメラ配置において従来よりも効率的な並列処理が可能であることが確認できた. 今後は, 大規模な対象 (多数の入力画像, 多数のボクセル) を用いた実験, 異なるアーキテクチャの並列計算機を用いた実験を行い, VGDD の詳細な評価と改良を進める予定である.

参考文献

- [1] S. Seitz, et al.: "A comparison and evaluation of multi-view stereo reconstruction algorithms", Proc. CVPR2006, pp. 519–528 (2006).
- [2] K. Kutulakos and S. Seitz: "A theory of shape by space carving", Int. J. Computer Vision, **38**, 2, pp. 199–218 (2000).
- [3] 川上: "複数の入力画像を修正に用いる三次元形状復元手法の研究", 東北大学大学院情報科学研究科修士論文 (2008).
- [4] M. Fukushi, et al.: "A parallelization method for multiview stereo", Proc. MVA2011, pp. 385–388 (2011).
- [5] "http://vision.middlebury.edu/mview" (2011).