

逐次的四面体カービング法を用いた3Dモデリング

鳥居 秋彦¹ 杉浦 貴行¹ 奥富 正敏¹

概要：画像が次々と入力され、Structure from Motion (SfM) によって疎な3D点群とカメラポーズが与えられる場合、逐次的に効率良くサーフェス生成を行う手法を提案する。提案手法では、四面体を削り出すサーフェス抽出法を、視線と四面体の交差の効率的な検出方法と、ダイナミックグラフカットを適用したサーフェス抽出によって、逐次的処理が可能な手法へと発展する。これらにより、追加の入力に対して効率の良い処理でありながら、常に全体最適性を保証することが可能である。従って、提案手法で抽出されるサーフェスは、既存のバッチ処理によるサーフェス抽出手法を、入力毎に始めから繰り返す場合と、完全に同一のものとなる。実験では、数種類の既存手法と比較を行い、提案手法の効果を示す。

1. はじめに

SIFT [18] の提案後、Structure from Motion (SfM) [1], [5], [24], [25] の発展によって、多視点画像によるカメラの位置姿勢と3D点群の推定が比較的容易に行えるようになった。さらに、画像検索手法を利用することで、効率よく処理が行えるようになり [11], [23]、インターネット上で集めた画像群など、様々な入力から大規模なSfMを行うことも可能になりつつある [1], [5], [6], [32]。SfMで得られるカメラポーズを用い、入力に対するバッチ処理として、より密な3D点群の復元 [7] や、3Dサーフェスの抽出 [10], [12], [15], [28] も提案されている。

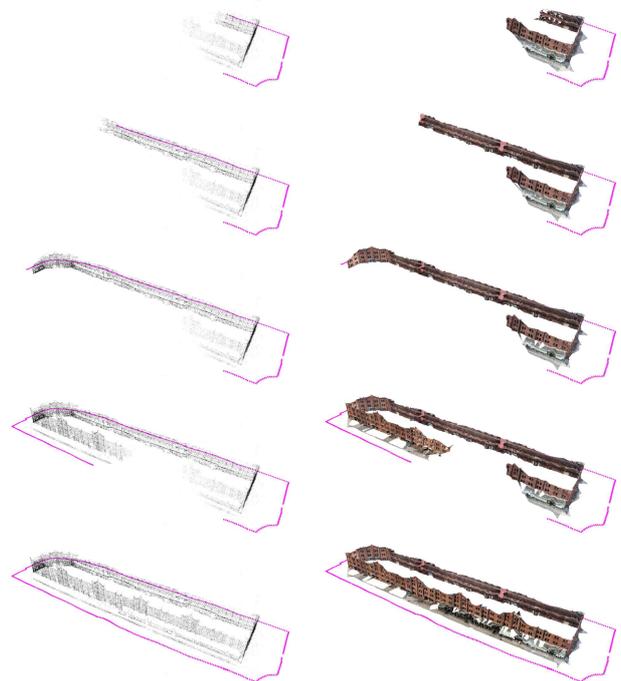
3Dモデリングの応用として、車両・UAVのナビゲーション [8], [21], [29] や変化検出 [22], [26] 等で用いる場合、刻々と追加される入力データに対して、3Dモデルを逐次的に生成・拡張し、視覚化することは重要である。ここで、SfMの各要素、特徴点抽出や画像ペアの選択とそのマッチング、カメラポーズの推定については、逐次処理に適した手法 [32] が提案され、グローバルバンドル調整についても、最近の研究 [9] で実現されている。

サーフェス生成に関して、逐次的な手法のひとつに、基準画像ごとに処理を行う方法 [13], [19], [21] が挙げられる。これらは、画素または画像上のパッチの奥行きを基準画像ごとに推定し、任意に設定した座標系に重ね合わせ続けるというアプローチをとる。処理の高速化が可能な反面、3Dサーフェスとしての全体最適性は保証されない。

特に、全体最適性を保証した3Dサーフェス抽出法として、空間に対するドロネー四面体分割と、その削り出しを



(a)



(b)

(c)

図1 提案手法による逐次的な3Dモデリング。(a) 入力画像例。(b) 逐次的SfMによる3D点群とカメラポーズ。(c) 逐次的に抽出される3Dサーフェス。

用いる手法 [15], [28] (本論文では四面体カービング法と呼ぶ) が、近年注目を集めている。ここで、全体最適性を保証するというのは、“シーン全体を考慮したうえで、3D

¹ 東京工業大学
Tokyo Institute of Technology

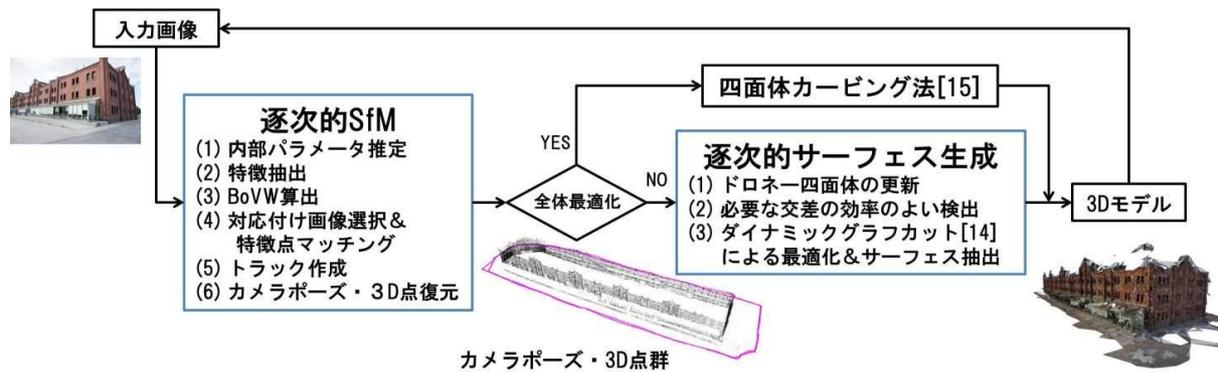


図 2 逐次的 3D モデリングシステムの概要.

サーフェス（物体境界面）抽出を行うために定義したコスト関数の最適解が求められている”，という意味である。四面体カービング法では，剛体であるという以外，対象とする物体形状に対する制約が無く，様々なシーンへの適用が可能である。Jancosek ら [10] は，この四面体カービング法を改良し，visual hull [16] を利用することで，半透明物体や，テクスチャのあまり無い物体も含め，非常に高品質なサーフェス抽出を実現している。

この四面体カービング法 [15] を利用した逐次的なサーフェス抽出法も存在する [17], [20], [29]。これらの手法では，例えば，時間的に近傍のカメラを選択するなど，経験則に基づいて，四面体の削り出しを効率良く行う反面，出力される 3D サーフェスの全体最適性は保証されない。

また，画像からの 3D モデリングを行うシステムとして，Hoppe ら [8] は，逐次的 SfM を行いつつ，一定フレーム毎に四面体カービング法をバッチ処理として適用する方法を提案した。この方法では，3D サーフェスの全体最適性が保証されるものの，逐次処理ではないため，撮影により 3D モデルが拡張し続ける場合，計算時間が増加する。逐次的で効率の良い処理を行いながら，全体最適性を保証するサーフェス生成法は，未だ確立されておらず，非常に挑戦的な課題である。

本論文では，次々と入力される画像に対して，逐次処理に適した SfM [31] によりカメラポーズと 3D 点群を推定し，逐次的なサーフェス抽出を行う逐次的 3D モデリングシステムを提案する (図 1)。サーフェス抽出は，[15] を拡張し，逐次的で効率の良い処理を行う。その際，全体最適性を保つサーフェス，つまり，四面体カービング法 [15] によるバッチ処理を繰り返した場合と，同一のサーフェスを抽出可能である。サーフェス抽出における，本論文の貢献を以下に挙げる：

- 全体最適性を保ちながら，逐次的で効率の良いサーフェスの抽出を行えるアルゴリズムの提案。
- 視線と四面体の面の種類に適応的な，2つの交差検出アルゴリズムの提案。

- 四面体の増減を取り扱うための孤立ノードの導入による，ダイナミックグラフカット [14] の逐次的サーフェス抽出への応用。

2. 逐次的 3D モデリングシステム

これまでに著者らは，逐次処理に適した SfM システムを提案してきた [31]。この SfM では，高速な特徴点マッチング，累積誤差に対して適応的なバンドル調整，画像検索を利用したループクロージング処理により，効率よく安定性の高い復元を実現する。さらに，SfM システムとは独立して，逐次的に SfM で得られるカメラポーズと点群を利用した高速なサーフェス生成法 [30] を提案してきた。本稿では新たに，高精度かつ様々な物体形状への適用が可能なる，四面体カービング法 [15] を発展した手法を提案する。これらの逐次処理に適した SfM とサーフェス生成法により，安定かつ逐次的に 3D モデリングを行うシステムを実現する (図 2)。

2.1 逐次処理に適した SfM

提案する逐次処理に適した SfM [31] を概略する。逐次入力される画像に対し，下記の処理を行う (図 2)：

- (1) EXIF から得られる焦点距離や CCD サイズを用い，カメラ内部パラメータの推定を行う；
- (2) 特徴量 (SIFT または SURF) を抽出する；
- (3) 抽出した特徴量から BoVW (Bag of Visual Words) を構成，処理中の画像と各画像間の類似度を計算する；
- (4) 時系列近傍画像と，十分に時刻の古い復元画像列の中からそれぞれ類似度の高い画像を数枚選択し，特徴点対応付けを行う；
- (5) 特徴点をリンクしたトラックを生成，拡張する；
- (6) 5 点アルゴリズム及び，2 点アルゴリズム^{*1}を用いて入力画像のカメラポーズを求め，3D 点を復元する；
- (7) 累積する復元誤差が一定閾値以上になった場合，全体でバンドル調整を行い，それ以外は現在処理している

^{*1} 回転成分は 5 点アルゴリズムから得られる基本行列から抽出して利用し，並進成分のみを推定する。

画像とそのカメラから見えている 3D 点のみを利用した局所的なバンドル調整を行う。

上記のアルゴリズムの中で、計算効率、復元精度を向上させるポイントを挙げる。(2)では、高速な画像類似度計算を用いることで、不要な画像ペアマッチング処理を省いている。(5)において、それまでに特徴点对応付けを行ってきたトラックと、矛盾が生じないかを確認することで、信頼性の高い点のみを復元するように促せる。また、(4)の段階で、画像群の中に過去に復元された画像中で類似度の高い画像を選択しておくことで、(5)、(6)により、自動的にループクローズが行われるため、復元精度の向上を期待できる。さらに、3D 点の復元角を確認し、復元角の小さい、精度の不安定なカメラ、3D 点を除去しつつ、復元を遂行することで、信頼性、復元精度の高い点とカメラを残すようにした。

2.2 全体最適性を保証した逐次的なサーフェス生成

四面体カービング法 [15] を拡張したアルゴリズムによって、逐次的にサーフェスを生成する方法を概略する (図 2) :

- (i) SfM によって生成される 3D 点群を入力として、それらを頂点とするドロネー四面体による、3D 空間の分割を更新する；
- (ii) カメラと 3D 点を結ぶ視線と、ドロネー四面体との交差を検出する。この時、更新に必要な交差のみを効率よく検出する；
- (iii) 四面体と視線の交差数に基づくコスト関数の、ダイナミックグラフカット [14] による効率の良い全体最適化により、各四面体が撮影対象の内部・外部どちらに属するか判定し、境界を抽出することで、サーフェスを得る。

これは、SfM によって生成された 3D 点は撮影したカメラから直接見えているはずである、という考察に基づき、より多くの視線と交差する四面体が撮影対象の外部に属するようにすることで、境界面の抽出 (削り出し) を行っている。文献 [15] においては、すべての 3D 点群とカメラ位置情報を用いて、上記各ステップの処理を行う。新たに提案する手法では、逐次入力に対して、文献 [15] の手法の全体最適性を保証しながら、各ステップにおいて逐次的に処理を進めることを可能にした。

2.3 統合における制約

提案手法による SfM とサーフェス生成は統合することが可能である。しかし、SfM において、グローバルバンドル調整によって生じる 3D モデル全体の歪みと摂動は、サーフェス抽出における逐次処理を複雑にしまう。実際に SfM と統合する場合は、基本的には、各更新ごとに提案する逐次的なサーフェス生成を適用し、グローバルバンドル調整後のみバッチ処理により四面体カービング法 [15] を行

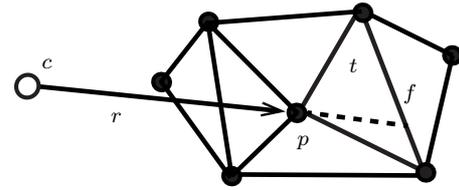


図 3 視線と面の交差。矢印の実線はカメラ c と 3D 点 p からなる視線 r 。破線は r の延長線を表す。 f は四面体 t の面である。

うアプローチが妥当である (図 2)。

3. 四面体カービング法を用いたサーフェス抽出

本章では、本論文の貢献を明確にするために、四面体カービング法 [15] について述べる。入力としては、3D 点群 $\mathcal{P} = \{p\}$ とカメラポーズ $\mathcal{C} = \{c\}$ 、視線 $\mathcal{R} = \{r\}$ が与えられる。ここで視線 r とは、 p とそれを見ている c の位置を結んだ線分である (図 3)。まず、 \mathcal{P} に対する 3D ドロネー分割によって空間を四面体群 $\mathcal{T} = \{t\}$ に分割する。以下では、対象に対する各四面体の内外判定について詳しく述べる。

3.1 2 値ラベル付け問題としての定式化

得られた \mathcal{T} に対して、各四面体が物体の内外どちらに属するのかを判定する。まず、四面体群の構造からグラフ $\mathcal{G} = (\mathcal{T}, \mathcal{E})$ を生成する。このグラフは各四面体をノード $\mathcal{T} = \{t\}$ とし、隣接する四面体間の面 $\mathcal{F} = \{f\}$ に対応するエッジ $\mathcal{E} = \{E = (t_p, t_q)\}$ を持つ。各四面体の内外の判定は、グラフのノードに対する 2 値のラベル付け問題として以下のコスト関数を最小化することで求めることができる。

$$E(\mathcal{L}) = \sum_{t \in \mathcal{T}} \{U_t(l_t) + \sum_{v \in V_t} B_{t,v}(l_t, l_v)\} \quad (1)$$

ここで、 $l_t \in \mathcal{L}$ は四面体 $t \in \mathcal{T}$ のラベルで、物体の内側 ($l_t = \text{IN}$) と外側 ($l_t = \text{OUT}$) のどちらに属するかを示す。単項コスト U_t は t につけられたラベル l_t と視線・四面体の関係が矛盾していないかを示す。

$$U_t(l_t) = \begin{cases} N_{\text{in}}(t) & \text{if } l_t \neq \text{IN} \\ N_{\text{out}}(t) & \text{if } l_t \neq \text{OUT} \end{cases} \quad (2)$$

$N_{\text{in}}(t)$ は、 t の頂点を見ている視線 r のうち延長線が t と交差する視線 (図 3) の数であり、 $N_{\text{out}}(t)$ は t が内部を含むカメラからの視線 r の数を表す。

2 項間コスト $B_{t,v}$ は隣接する 2 つの四面体間のラベルの変化を抑える平滑化項で以下のように表される。

$$B_{t,v}(l_t, l_v) = \begin{cases} N_{\text{intrsect}}(t, v) & \text{if } \begin{cases} l_t = \text{OUT} \\ l_v = \text{IN} \end{cases} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$N_{\text{intrsect}}(t, v)$ は、 t と隣接する四面体 $v \in V_t$ で共有される

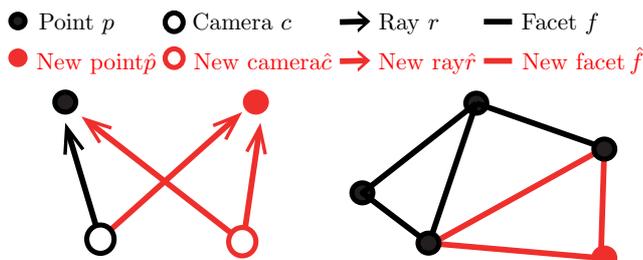


図4 更新後のグラフ G' の重みを構成する要素：3D点，カメラ，視線，面．詳細は本文を参照のこと．

面 $f(t, v)$ と， t 側から v 側に向かう視線 $r_{t \rightarrow v}$ の交差の数である．詳細は [15] を参照されたい．

3.2 視線と面の交差検出

コスト関数の計算には，視線と四面体の面との交差検出が必要である．ここでは，各視線 r が3D点 p からカメラ中心 c までの間で貫く四面体を検出する (図3)．これは， p を頂点に持つ四面体から c を含む四面体まで，隣接する四面体をたどることで効率的に検出することができる．本論文では4.1節で述べる提案法と区別するために，この処理を“ray-wise tracing”と呼ぶ．

3.3 グラフカットによる最適化

コスト関数の最適化には一般的な最大流/最小切断アルゴリズム [2] を用いる．グラフ G に対する s - t グラフカットにより，全体最適なラベル付けを求める．隣接する四面体間の面に対応するエッジには，2項間コスト $B_{t,v}$ による重み付けがされる．また，各ラベルに対応する特別なノード“source”と“sink”はすべてのノードに対してエッジを持ち，単項コスト U_t による重み付けを行う．グラフカットにより，四面体群 \mathcal{T} に対する，コスト関数 E を全体最小化するラベル付けを求めることで，全体最適解としてのサーフェスが抽出される．

4. 逐次的四面体カービング法

本章では，全体最適性を保ちながら，四面体カービング法を逐次処理に拡張する方法を述べる．つまり，追加の入力がされた場合でも，従来の手法 [15] を初めからやり直すことで抽出されたサーフェスと同一な出力を得ることができる．以降， $\hat{c} \in \hat{\mathcal{C}}$ と $\hat{p} \in \hat{\mathcal{P}}$ はそれぞれ，新たに追加されたカメラポーズと3D点を表すこととする (図4)．まず，追加された3D点に対して四面体群を更新する．点群に対するドロネー分割の一意性によって，追加する点の順序に関係なく同一な四面体群を得ることができる [4]．以下，更新された四面体群にラベルを付けるための2つの重要な処理を述べる．

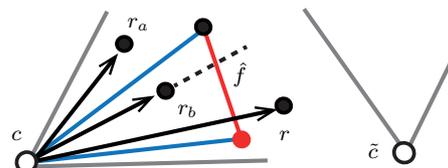


図5 Facet-wise detection による交差検出．カメラの視野によりカメラ \hat{c} からの視線は面 \hat{f} と交差しないことがわかる．さらに c に関する \hat{f} の視円錐により， r_a を除去可能で，最後に視線 r と r_b のように実際に \hat{f} と交差しているのかを判定する．

4.1 新しい交差の効率的な検出

新たなカメラと3D群の追加や四面体群の変化によって，グラフ G が G' に更新される．グラフカットによる最適化のために，更新後のグラフ G' の重みをすべて再計算してしまうと，入力が増加するにつれて計算のボトルネックになってしまう．そこで，全体最適性を保ちながら効率的な計算を行うためには， G' の中で，新たに発生した交差に対応するエッジの重みのみを更新することが必要である．本節では重みの更新に必要な視線と面の交差検出法を述べる．

まず，2種類の視線 \mathcal{R} と $\hat{\mathcal{R}}$ を考える． $r \in \mathcal{R}$ は1台のカメラ $c \in \mathcal{C}$ と1つの3D点 $p \in \mathcal{P}$ からなる視線である．また， $\hat{r} \in \hat{\mathcal{R}}$ は新たに追加された視線であり， \hat{r} は $\hat{c} \in \hat{\mathcal{C}}$ と $\hat{p} \in \hat{\mathcal{P}}$ ，もしくは $c \in \mathcal{C}$ と $\hat{p} \in \hat{\mathcal{P}}$ からなる (図4)．同様に，面に関しても \mathcal{F} と $\hat{\mathcal{F}}$ の2種類に分類する． \mathcal{P} に対するドロネー四面体群の中で新たな入力の追加後も存在する面を \mathcal{F} とし，新しく生成された面を $\hat{\mathcal{F}}$ とする．グラフの更新において，すでに検出されている交差の再計算を避けるためには，新たな入力によって変化することのない \mathcal{R} と \mathcal{F} の交差を避ける必要がある．

まず，新たな視線 $\hat{\mathcal{R}}$ には ray-wise tracing を適用する．面の種類に関係なく，面 $\mathcal{F} \cup \hat{\mathcal{F}}$ と \hat{r} の交差は検出する必要があるからである．残りの交差の中でグラフの更新に関係するのは \mathcal{R} と $\hat{\mathcal{F}}$ の交差である．そこで検出が必要な交差を得るために，すべての視線 \mathcal{R} に対して ray-wise tracing を適用してしまうと，初めから計算し直した場合と同じ計算量になってしまう．しかし，ray-wise tracing では面の種類を無視して四面体をたどるために，新たな面 $\hat{\mathcal{F}}$ と交差する視線を選び出して ray-wise tracing を適用することは不可能である．

そこで，1つの面に対して何本の視線が交差するかを検出する方法を導入する．これを“facet-wise detection”と呼ぶ．この facet-wise detection を新しい面 \hat{f} に適用することで \mathcal{R} の中で，交差する視線だけを選択可能であるため， $\hat{\mathcal{F}}$ と \mathcal{R} の交差を効率よく検出できる．

Facet-wise detection の処理について述べる (図5)．まず，新しい面 \hat{f} に対して各カメラの視野を考えることで，交差する可能性のある視線を持たないカメラ \hat{c} を除去する．続いて，選択されたカメラに関する \hat{f} の視円錐によって，各カメラの持つ視線の中から \hat{f} と交差しない視線 r_a を検

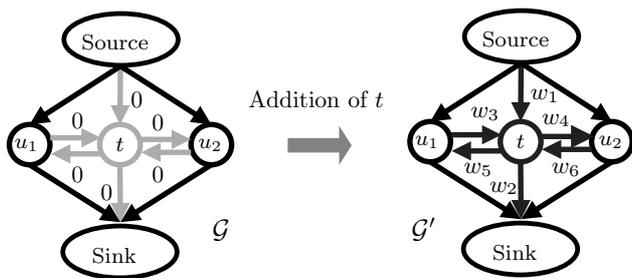


図 6 グラフ G から G' への更新における孤立ノード t の適用.

出する. ここで, 本論文では 3 章で述べたように, 視線をカメラから 3D 点までの線分としているので \hat{f} まで到達していない視線 r_b が存在することに注意されたい.

まとめると, 新しい視線 \hat{R} に対する ray-wise tracing と新しい面 \hat{F} に対する facet-wise detection によって, グラフの更新に必要な交差を過不足なく検出することができる. 提案手法による交差検出は, カメラや 3D 点の増加によって増え続ける \mathcal{R} と \mathcal{F} の交差を検出せずに行うことができる.

4.2 ダイナミックグラフカット

四面体群の更新と新たな交差の検出によってグラフ G が G' に更新される. このグラフの変化はダイナミックグラフとみなすことができ, Kohli ら [14] によって, 変化後のグラフでの全体最適解を求めるアルゴリズムが提案されている. エッジの重みの変化による流量の変化を局所的に計算することで, 流量全体を再計算することなく効率よく最大流を求めることができる. このアルゴリズムはグラフ全体に対して変化が局所的であった場合に大きな効果を発揮する. 提案手法の場合, 新たな入力の追加によるグラフの変化がシーン全体に対して一部であるという仮定は, 十分に妥当である.

また, 文献 [14] ではグラフのエッジの重みの変化を想定している. それに加えて提案するアルゴリズムでは, 新たな入力の追加により, 四面体群 \mathcal{T} が $\hat{\mathcal{T}}$ に更新され, グラフ G も G' に更新される際, 以下の 2 つの変化が起こりうる. (1) グラフ G' に新たなノード \hat{t} が生成される. (2) ドローン分割において除去された四面体群については, 対応するノードが G' から取り除かれる. 本論文では一度追加された 3D 点の除去や統合は発生しないと仮定しているが, ドローン分割の処理によって四面体の増減は生じる.

ノードの増減を含んだ変化に対応し, グラフ G から G' への変化にダイナミックグラフカットアルゴリズム [14] を適用するため, 孤立ノードという概念を導入する. あるグラフ中の孤立ノードとはそのノードに接続されているすべてのエッジの重みが 0 であるノードのことを指す.

- ノード t を追加する場合, まず変化前のグラフ G に新たなノード t を孤立ノードとして追加する. グラフが G' に更新されると, 4.1 節で述べた視線と面の交差の

数を, 対応する G の孤立ノード t のエッジに重みとして加える (図 6).

- ノード t をグラフから取り除く場合, 変化後のグラフでは t に関するエッジの重みを 0 にすることで t を孤立ノードとする.

孤立ノードは存在している四面体と分離されているので, コスト関数に影響を与えることがなく, 全体最適なラベル付けを求めることができる.

5. 実験

本章では提案手法の評価実験を示す. 既存研究として新たな入力があるたびに [15] を始めからバッチ処理を繰り返す手法 (以下, ベースライン手法と呼ぶ) と提案手法によるサーフェスの更新との比較を行う.

5.1 実装の詳細

上記のように, 2 値のラベル付けによって内外の四面体群の境界として得られたサーフェスに対して 2 つの後処理を施すことで改善を行う. まず, 各パッチの最も長い辺の長さに対するパーセンタイル法を用いることで, 大きな 3 角パッチを取り除く. 次に, サーフェスの平滑化を行う. 以下に基づいた座標の更新を繰り返し, 各 3D 点に施す:

$$p_{i+1} = p_i + \lambda \sum_{q_i \in V_p} w_{p_i q_i} (q_i - p_i), \quad (4)$$

V_p はサーフェス上で点 p と隣接する点の集合, λ は平滑化パラメータである. $w_{p,q}$ は 2 点間の距離に応じた重みで, $w_{p,q} = \phi(p,q) / \sum_{k \in V_p} \phi(p,k)$ で表される. ここで, $\phi(p,q)$ は 2 点間の距離の逆数である. これらの処理は逐次処理に拡張されていないが, サーフェスモデルの最終結果の見た目を向上させるために用いる.

本実験は, Intel Core i7, CPU 3.20GHz, RAM 64GB のデスクトップ PC 上で, Linux 64bit OS, C++ を用いて行った. 実装において以下のライブラリを利用した: OpenCV [3], CGAL [27], ダイナミックグラフカットライブラリ [14]. 視線と面の交差検出においてカメラごとのマルチスレッド化を行った.

3D モデルの生成において, SfM による 3D 復元と, 提案手法によるサーフェス抽出を分け, 純粹に提案手法によるサーフェス抽出の効果を評価するために, あらかじめデータセットの画像すべてを用いてカメラポーズ \mathcal{C} と 3D 点群 \mathcal{P} を復元しておく. サーフェス抽出の入力には 1 台のカメラとそれに関係した 3D 点群を徐々に追加していく.

5.2 提案手法の評価

評価には 1 つの建物の周りを歩いて撮影した 978 枚の画像からなる赤レンガ倉庫データセットを用いた. 画像サイズは 2296×1528 で, 焦点距離固定の 1 台のカメラで撮影

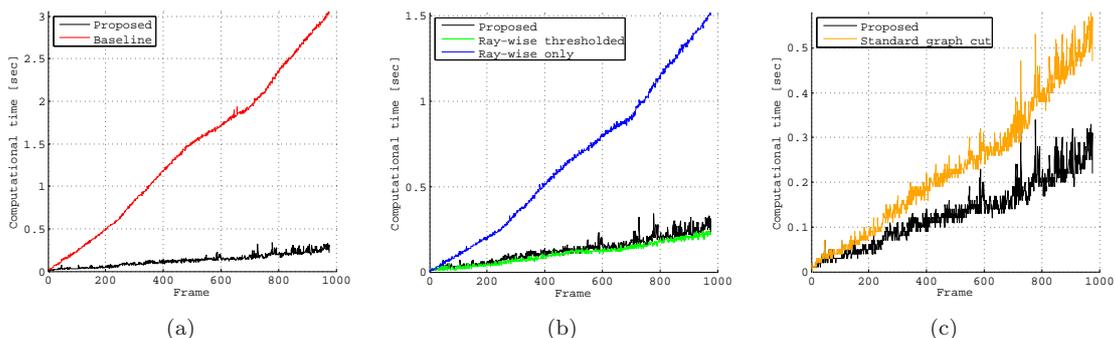


図7 既存手法との比較結果. 各グラフは1枚の画像を追加した時に各フレームでサーフェス抽出にかかった計算時間. (a) 提案手法(黒)とベースライン手法(赤). (b) 提案手法(黒)と ray-wise tracingのみを用いた手法(青), しきい値により視線の一部に対して ray-wise tracingを適用する手法 [29](緑). (c) 提案手法(黒)と一般的なグラフカット [2]を用いる手法(橙).

したものである. 提案手法による復元結果(図1)と計算時間(図7(a))を示す. 画像枚数が増えると計算時間が増加するベースライン手法(赤)に比べて, 提案手法(黒)は比較的一定な時間でサーフェス抽出を行うことができている. 977回の更新にかかった合計時間を比較しても, 提案手法は127.67秒, ベースライン手法では1385.23秒となった. 最終的に出力されたサーフェスの三角パッチの枚数を比較することでベースライン手法と提案手法のサーフェスが完全に一致している, つまり提案手法で全体最適性の保たれたサーフェスを抽出できていることが確認された.

5.2.1 facet-wise detection の効果

提案手法における facet-wise detection の効果を評価するために, 視線と面の交差検出の方法を3種類の方法で行った計算時間の結果を図7(b)に示す. ray-wise tracingのみを用いた手法(青)は, 画像が増え, シーンが拡張していくにしたがって計算時間が増えている. また, ray-wise tracingを視線の一部に適用する手法 [29](緑)は, 入力された時間の近傍にあるカメラをしきい値によって選び, そのカメラに含まれる視線のみに対して処理を行う手法である. この手法による結果は部分的に提案手法(黒)よりも短い計算時間でサーフェスを抽出できているが, 最終的に出力されたサーフェスは全体最適性を保つベースライン手法と比べて47%のパッチ数の差を生じた. 一方で提案手法(黒)は facet-wise detection の効果により ray-wise tracingのみを用いた手法に比べて比較的一定な計算時間を達成している.

5.2.2 ダイナミックグラフカットの効果

グラフの変化に対して, 部分的な流量を計算するダイナミックグラフカット [14]を用いた提案手法(黒)と始めから流量を計算し直す一般的なグラフカット [2]を用いた手法(橙)の計算時間を図7(c)に示す. 他の処理の効果に比べて削減できる時間は少ないが, 毎フレーム平均で30%の削減は, 実時間応用を見込んだ場合有効であるといえる.

表1 赤レンガ倉庫データセットの3次元点と3角パッチの数.

	Proposed	[10] (SfM)	[10]	[7]
#points	124,216	196,994	1,797,387	5,141,170
#patches	276,367	227,355	2,503,592	-

5.2.3 定性的な評価

提案手法による3Dサーフェスモデルと既存の3Dモデリングの手法の結果との定性的な比較実験を行った. 利用可能なソフトウェアとしてPMVS [7]とCMPMVS [10]を用いた. CMPMVSについては平面走査法を用いた密な点群に対する結果と平面走査法を行わないSfMの出力である疎な点群に対する結果 *2を示す. 図8に各手法の3Dモデルを, 表1に出力された3D点や3角パッチの数をそれぞれ示した. 提案手法によるモデルは他の手法に劣らない結果であることが確認できる.

6. 結論

本論文では, 次々と入力される画像に対して, 効率良く逐次的にSfM, サーフェス生成を行う手法を提案した. 特にサーフェス生成に関しては, 既存の四面体カービング法を拡張することで, 逐次処理を行いながらも, 全体最適性を保証するアルゴリズムを提案した. 実験では提案手法による逐次的なサーフェス抽出によって画像を追加した時のモデルの更新を効果的に行えることを示した.

謝辞 本研究は科研費25240025, 24700161の一部助成を受けたものである.

参考文献

- [1] Agarwall, S., Snavely, N., Simon, I., Seitz, S. and Szeliski, R.: Building Rome in a Day, *Proc. ICCV*, pp. 72-79 (2009).
- [2] Boykov, Y. and Kolmogorov, V.: An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy

*2 平面走査法を行わないCMPMVSの実装は [10]の著者らの協力によるものである.



図 8 既存手法との定性的な比較. (a) 提案手法. (b) PMVS [7] による色付けされた密な点群. (c) 提案手法と同じ入力を与える, 平面走査法を行わない CMPMVS. (d) CMPMVS [10].

- Minimization in Vision, *PAMI*, Vol. 26, No. 9, pp. 1124–1137 (2004).
- [3] Bradski, G.: The OpenCV Library, *Dr. Dobb's Journal of Software Tools* (2000).
- [4] Devillers, O. and Teillaud, M.: Perturbations and vertex removal in a 3D delaunay triangulation, *ACM-SIAM, SODA '03*, pp. 313–319 (2003).
- [5] Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S. and Pollefeys, M.: Building Rome on a Cloudless Day, *ECCV*, pp. 368–381 (2010).
- [6] Furukawa, Y., Curless, B., Seitz, S. M. and Szeliski, R.: Towards Internet-scale multi-view stereo, *CVPR*, pp. 1434–1441 (2010).
- [7] Furukawa, Y. and Ponce, J.: Accurate, Dense, and Robust Multiview Stereopsis, *PAMI*, Vol. 32, pp. 1362–1376 (2010).
- [8] Hoppe, C., Klopschitz, M., Rumpler, M., Wendel, A., Kluckner, S., Bischof, H. and Reitmayr, G.: Online Feedback for Structure-from-Motion Image Acquisition, *BMVC*, pp. 70.1–70.12 (2012).
- [9] Indelman, V., Roberts, R., Beall, C. and Dellaert, F.: Incremental Light Bundle Adjustment, *BMVC* (2012).
- [10] Jancosek, M. and Pajdla, T.: Multi-view reconstruction preserving weakly-supported surfaces, *CVPR*, pp. 3121–3128 (2011).
- [11] Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P. and Schmid, C.: Aggregating local image descriptors into compact codes, *PAMI*, pp. 1704–1716 (2012).
- [12] Kazhdan, M., Bolitho, M. and Hoppe, H.: Poisson surface reconstruction, *Eurographics, SGP '06*, pp. 61–70 (2006).
- [13] Kim, K., Sugiura, T., Torii, A., Sugimoto, S. and Okutomi, M.: Instant Surface Reconstruction for Incremental SfM, *MVA*, pp. 371–374 (2013).
- [14] Kohli, P. and Torr, P. H. S.: Efficiently Solving Dynamic Markov Random Fields Using Graph Cuts, *ICCV*, pp. 922–929 (2005).
- [15] Labatut, P., Pons, J.-P. and Keriven, R.: Efficient Multi-View Reconstruction of Large-Scale Scenes using Interest Points, Delaunay Triangulation and Graph Cuts, *ICCV*, pp. 1–8 (2007).
- [16] Laurentini, A.: The Visual Hull Concept for Silhouette-Based Image Understanding, *PAMI*, Vol. 16, No. 2, pp. 150–162 (1994).
- [17] Lovi, D., Birkbeck, N., Cobzas, D. and Jagersand, M.: Incremental Free-Space Carving for Real-Time 3D Reconstruction, *3DPVT* (2010).
- [18] Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints, *IJCV*, Vol. 60, No. 2, pp. 91–110 (2004).
- [19] Newcombe, R. A., Lovegrove, S. and Davison, A. J.: DTAM: Dense tracking and mapping in real-time, *ICCV*, pp. 2320–2327 (2011).
- [20] Pan, Q., Reitmayr, G. and Drummond, T.: ProFORMA: Probabilistic Feature-based On-line Rapid Model Acquisition, *BMVC*, pp. 1–11 (2009).
- [21] Pollefeys, M., Nistér, D., Frahm, J.-M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S. J., Merrell, P., Salmi, C., Sinha, S. N., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R., Welch, G. and Towles, H.: Detailed Real-Time Urban 3D Reconstruction from Video, *IJCV*, Vol. 78, No. 2-3, pp. 143–167 (2008).
- [22] Sakurada, K., Okatani, T. and Deguchi, K.: Detecting Changes in 3D Structure of a Scene from Multi-view Images Captured by a Vehicle-mounted Camera, *CVPR*, pp. 137–144 (2013).
- [23] Sivic, J. and Zisserman, A.: Video Google: Efficient Visual Search of Videos, *CLOR*, pp. 127–144 (2006).
- [24] Snavely, N., Seitz, S. and Szeliski, R.: Photo tourism: exploring photo collections in 3D, *SIGGRAPH*, pp. 835–846 (2006).
- [25] Snavely, N., Seitz, S. and Szeliski, R.: Modeling the World from Internet Photo Collections, *IJCV*, Vol. 80, No. 2, pp. 189–210 (2008).
- [26] Taneja, A., Ballan, L., Pollefeys, M. and Pollefeys, M.: Image based detection of geometric changes in urban environments., *ICCV*, pp. 2336–2343 (2011).
- [27] CGAL: Computational Geometry Algorithms Library, <http://www.cgal.org>.
- [28] Vu, H.-H., Labatut, P., Pons, J.-P. and Keriven, R.: High Accuracy and Visibility-Consistent Dense Multi-view Stereo, *PAMI*, Vol. 34, No. 5, pp. 889–901 (2012).
- [29] Yu, S. and Lhuillier, M.: Incremental Reconstruction of Manifold Surface from Sparse Visual Mapping, *3DIM-PVT*, pp. 293–300 (2012).
- [30] 半澤悠樹, 鳥居秋彦, 奥富正敏: オンライン 3 次元復元システム, 画像センシングシンポジウム (SSII), pp. DS1-08–1–1 (2012).
- [31] 半澤悠樹, 鳥居秋彦, 奥富正敏: オンライン処理に適した Structure from Motion システム, 画像の認識・理解シンポジウム (MIRU), pp. IS2-17–1–8 (2012).
- [32] 半澤悠樹, 鳥居秋彦, 奥富正敏: オンライン撮影に適した実用的な SfM システム, 電子情報通信学会論文誌 D, Vol. J96-D, No. 8, pp. 1753–1763 (2013).