

ステレオ画像からの直接的かつ高速な微小平面3Dサーフェス生成法

杉本 茂樹 奥富 正敏

東京工業大学大学院理工学研究科機械制御システム専攻

〒152-8550 東京都目黒区大岡山 2-12-1-S5-22

E-mail: {shige,mxo}@ok.ctrl.titech.ac.jp

あらまし 本論文では、ステレオカメラで撮影された画像を利用して、滑らかな曲面に対する効率的な3次元再構成手法を提案する。提案手法では、基準画像を小三角パッチによるメッシュに分割し、全パッチの頂点の奥行きを求めることにより曲面の3次元形状を得る。頂点の奥行きを求める際には、基準画像上のパッチと、対応する参照画像上のパッチとのSSD(Sum of Squared Differences)を考え、全パッチについて加算した値を最小にする各画素の奥行きを、Gauss-Newton法を用いて最適化する。最適化における繰り返し計算では、ICIA(Inverse Compositional Image Alignment)の考え方を利用することにより、繰り返し計算ごとの計算コストを低く抑えることが可能となる。合成画像と実画像を用いた実験を通じて提案手法の有効性を示す。

キーワード 微小平面サーフェスモデル, ステレオ3次元再構成, ICIA, Gauss-Newton法

A Direct and Efficient Method for Piecewise-Planar Surface Reconstruction from Stereo Images

Shigeki SUGIMOTO and Masatoshi OKUTOMI

Department of Mechanical and Control Engineering, Graduate School of Science and Engineering, Tokyo
Institute of Technology

Ookayama 2-12-1, Meguro-ku, Tokyo, 152-8550 Japan

E-mail: {shige,mxo}@ok.ctrl.titech.ac.jp

Abstract In this paper, we propose a direct method for 3D surface reconstruction from stereo images. In the proposed method, we reconstruct a 3D surface by estimating the depths of all vertices of piecewise triangular patches which composes a mesh generated on the reference image. We express the SSD (sum of squared differences) value between a single patch region in the reference and the corresponding region in the input as a function of three depths of the vertices of the patch. Then all depths are estimated by minimizing the accumulated SSD value with respect to all patches by Gauss-Newton optimization. For reducing the iterative computational costs for depth estimation, we incorporate an ICIA (inverse compositional image alignment) manner for expressing the SSD function. The validity of the proposed method is shown by some experiments using synthetic and real images.

Key words piecewise-planar surface model, stereo 3D reconstruction, ICIA, Gauss-Newton optimization

1. はじめに

伝統的なステレオ法では、一方の画像上の特定画素に注目し、もう一方の画像上にある対応点を探索することにより奥行き推定を行う。この方法によって得られる3次元情報は、3次元空間中の点の座標であり、対象物体の密な表面形状(サーフェス)を直接的に表現したものではない。

ステレオ画像からサーフェスを生成するための代表的なアプローチは、ステレオ法で得られた3次元計測点群を、特定のサーフェスモデルに当てはめる方法である。(例えば[9][2][11][14])。しかし、これらの手法のほとんどは、もとの3次元計測点の補間および変更に基づいており、対応点探索によって得られる3次元情報が、画像ノイズの影響を受けて誤差を含んでいることを考慮すると、3次元計測点を補間した後に、その点を画像上に再

び投影して、形状と画像との整合性を評価する必要があり、全体の計算コストは小さくない。

これに対し、ステレオ画像の輝度情報を利用して、直接的に物体サーフェスを生成するためのアプローチが提案されている。これらの手法は、画素値の差を評価値として含むコスト関数をサーフェスパラメータによって表現し、そのコスト関数を最小化することによりサーフェスを生成するアプローチである。このような方法として、ステレオ画像からポリゴンメッシュを生成する手法 [10] [6] や、Oriented Particle モデルを生成する手法 [5] などが挙げられる。また、近年では、Level Set 法に基づいて微分方程式を解く手法 [3] [7] も盛んに研究されている。しかし、これらの手法は、彫像や人形などの静止物体の正確な 3 次元モデルを得ることを目的にしているため、計算コストについてはほとんど考慮されておらず、一般に、数十秒から数分程度の計算時間を必要とする。利用する画像中の画素を間引くなどの方法で計算コストを抑える方法 [6] も考えられるが、この場合は、直接法による利点が損なわれる。

一方、ステレオ法は、2 枚の画像間の座標の対応推定に基づいた技術であるため、単眼カメラを利用した移動物体のトラッキング技術と深い関連がある。特に、ステレオ画像から滑らかな物体サーフェスを生成する技術は、2 枚の画像間の滑らかな対応を求める技術であるため、単眼カメラを利用した非剛体のトラッキング技術と結びつけて考えることができる。Active Blobs [12] では、三角パッチによるメッシュをテンプレート画像上に定め、パッチのアフィン変形を示すパッチ頂点の移動量をパラメータとしたコスト関数を最小化することにより、各頂点の移動量を求めている。このコスト関数に、既知のステレオカメラ配置 (カメラ移動量) を組み込むことができれば、パッチ頂点の移動量をより強く拘束できるため、三角メッシュによるサーフェスモデルを安定に生成できる可能性がある。

そこで、本論文では、三角メッシュモデルを利用した、ステレオ画像からの直接的かつ高速な物体サーフェス生成法を提案する。提案手法では、まず、基準画像を三角パッチによるメッシュに分割する。各三角パッチは 3 次元空間中で平面をなし、物体表面の形状をよく近似できていると仮定する。ステレオ画像では、単眼カメラのトラッキングとは異なり、微小カメラ移動量を想定できないため、ステレオ画像間におけるパッチの変形は、一般に平面射影変換 (homography) となる。そこで、提案手法では、画像間に成立するパッチの平面射影変換を各頂点の奥行きを用いて表現し、1 つのパッチに関するステレオ画像間の SSD (Sum of Squared Differences) を、3 頂点

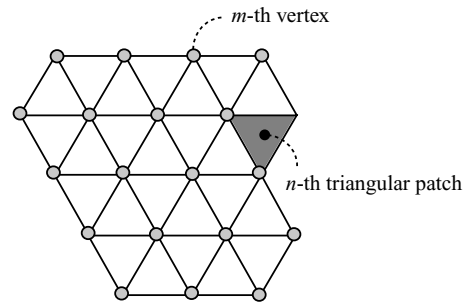


図 1 基準画像上に作成した三角メッシュの例

の奥行きパラメータを用いて表現する。そして、全パッチについて加算した値をコスト関数として最小化することにより、全パッチの奥行きを求める。ただし、このコスト関数を最小化するには、大きな計算コストを必要とする。そこで、平面射影変換と各頂点の奥行きパラメータとの間に介在する平面パラメータに注目し、その平面パラメータを高速に推定するアルゴリズム [13] を利用して、全パッチに関するコスト関数を高速に最小化する。

以下、本論文の構成は次のとおりである。まず、第 2 章では、微小平面モデルについて簡潔にまとめる。次に、第 3 章では、各パッチの平面射影変換と各頂点の奥行きとの関係について述べるとともに、全頂点の奥行きを求める基本的な考え方について説明する。第 4 章では、第 3 章にて述べた手法を高速化する枠組みについて説明し、奥行きを求めるための具体的な定式化を行う。そして、第 5 章において実験結果を示し、最後に本論文をまとめる。

2. 三角メッシュによる表面形状モデル

提案手法では、まず、基準画像を小さな三角形で分割してメッシュを作成する。メッシュを作成する方法は、画像全体を方形で分割した後に方形の対角線を結び三角形に再分割する方法や、画像上から特徴点を抽出した後に特徴点を母点としてドロネー分割する方法など、さまざまな方法が考えられるが、ここでは、画像領域全体が微小な三角形で分割されていれば、どのような方法を用いてもよい。正三角形によって構成されたメッシュの例を図 1 に示す。

メッシュの頂点を $\mathbf{v}_m (m = 1, 2, \dots, M)$ と表し、 \mathbf{v}_m は対象物体表面の 3 次元形状に対応した奥行き d_m を持つものとする。また、頂点によって構成される三角形をパッチと呼び、 $C_n (n = 1, 2, \dots, N)$ と表す。各パッチは 3 次元空間中において平面を構成するものとする。すなわち、パッチによって構成されるメッシュはポリゴンメッシュであり、メッシュが十分細かければ任意の 3 次元形状を精度よく近似できると考えられる。

本論文では、メッシュの細かさに関する検討は特に行わない。ただし、画像レジストレーション分野では広く知

られているように、粗い(大きな)メッシュを利用した場合には、より推定が安定になると考えられるため、後に3.にて述べる全頂点の奥行き推定では、最初は粗いメッシュを用いて奥行きを推定し、徐々にメッシュを細かくする粗密探索のアプローチが有効である(この方法は、5.の実画像実験において利用している)。また、細かい形状を持つ領域をなんらかの手法で推定し、その領域に対応するパッチのみを細かくするなど、subdivisionの考え方を利用したパッチの分割方法も有効であると考えられる。

3. 全頂点の奥行き推定に基づくサーフェス生成

本章では、全頂点の奥行きを求める基本的な考え方について説明する。ここでは、まず、1つの三角パッチに注目し、そのパッチの平面射影変換と各頂点の奥行きとの関係について述べ、1つの三角パッチに関する画像間のSSDを定式化する。そして、全てのパッチに関するSSDを加算したSSSDをコスト関数とし、そのコスト関数を最小化するアプローチについて説明する。

3.1 平面射影変換と3頂点の奥行き

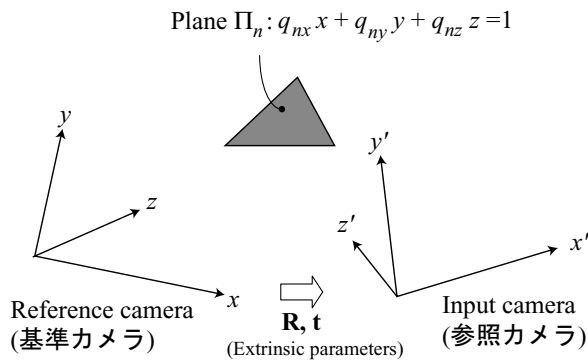


図2 3次元空間中の平面

基準画像座標を $\mathbf{u} = (u, v)^T$ 、参照画像座標を $\mathbf{u}' = (u', v')^T$ とする。ただし、本論文では、煩雑さを避けるため、以後、画像座標 \mathbf{u}, \mathbf{u}' は内部パラメータが正規化された座標(正規化画像座標)を示すものとする。

いま、ある三角パッチ C_n が3次元空間中の平面 Π_n をなすとする、 Π_n を2枚の画像上に射影したときの2枚の画像の座標関係は、同次座標表現 $\hat{\mathbf{u}}, \hat{\mathbf{u}'}$ を用いて、

$$\hat{\mathbf{u}} \sim \mathbf{P}_n \hat{\mathbf{u}'} \quad (1)$$

と書ける。ただし、 \mathbf{P}_n は 3×3 の平面射影変換行列を表している。

式(1)によって定められる2つの画像座標間の平面射影変換を、以下では、 $\mathbf{u}' = \mathbf{w}(\mathbf{u}; \mathbf{p}_n)$ と表す[1]。ただし、 \mathbf{p}_n は、行列 \mathbf{P}_n の要素をラスタ順に並べた9次元のベクトルとする。

基準画像を T 、参照画像を I とすると、 \mathbf{w} を用いた座標

変換表現により、基準画像上のパッチ C_n 領域内の座標における画像間のSSD(Sum of Squared Differences)は、次式で書ける。

$$\sum_{\mathbf{u} \in C_n} \{T[\mathbf{u}] - I[\mathbf{w}(\mathbf{u}, \mathbf{p}_n)]\}^2 \quad (2)$$

いま、射影変換パラメータ \mathbf{p}_n が、パッチ C_n の3頂点の奥行きによって表現できるものとする、式(2)をその頂点の奥行きによって表現されたコスト関数と考えることができ、そのコスト関数を最小化する3頂点の奥行きを計算することができる。以下では、その具体的な表現を説明する。

画像座標が正規化画像座標で表されているため、 \mathbf{P}_n は、次式で書ける[4]。

$$\mathbf{P}_n = \mathbf{R} + \mathbf{t}\mathbf{q}_n^T \quad (3)$$

ただし、 \mathbf{R} と \mathbf{t} は、カメラ間の回転行列および平行移動ベクトルを表し、それぞれ既知とする。また、 $\mathbf{q}_n \equiv (q_{nx}, q_{ny}, q_{nz})^T$ は、図2に示すように、平面 Π_n の3次元空間中の方程式を表す3自由度のパラメータ(以下、平面パラメータと呼ぶ)である。すなわち、式(3)は、射影変換パラメータ \mathbf{p}_n が \mathbf{q}_n の関数となることを表している。

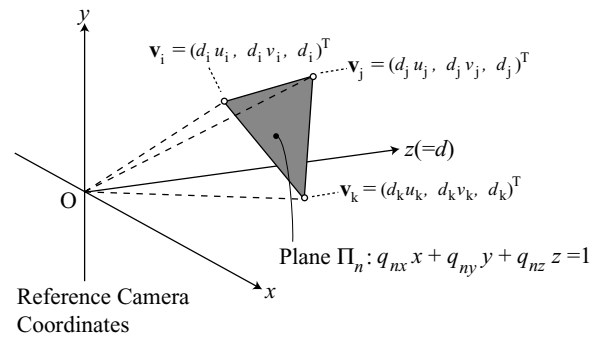


図3 パッチ頂点の3次元座標

また、パッチ C_n の3頂点の基準画像上の座標を $\mathbf{u}_m = (u_m, v_m)^T$ ($m = i, j, k$) とし、それらの奥行きを d_m とすると、図3に示すように、3頂点の3次元空間中の座標 \mathbf{v}_m は $\mathbf{v}_m = (d_m u_m, d_m v_m, d_m)^T$ ($m = i, j, k$) と表すことができる。このとき、 \mathbf{q}_n と d_m ($m = i, j, k$) の関係は、次式で表される。

$$\mathbf{q}_n = \mathbf{L}_n \boldsymbol{\gamma}_n \quad (4)$$

$$\text{where } \mathbf{L}_n \equiv \begin{bmatrix} u_i & v_i & 1 \\ u_j & v_j & 1 \\ u_k & v_k & 1 \end{bmatrix}^{-1}, \quad (5)$$

$$\boldsymbol{\gamma}_n \equiv (1/d_i, 1/d_j, 1/d_k)^T. \quad (6)$$

すなわち、式(4)は、 \mathbf{q}_n が、 $\boldsymbol{\gamma}_n = (1/d_i, 1/d_j, 1/d_k)^T$ の

関数として表現できることを意味している。

以上のことから、式 (2) は、次式で書ける。

$$\sum_{\mathbf{u} \in C_n} \{T[\mathbf{u}] - I[\mathbf{w}(\mathbf{u}, \mathbf{p}_n(\mathbf{q}_n(\gamma_n)))]\}^2 \quad (7)$$

よって、式 (7) を最小化する γ_n を求めることにより、三角パッチ C_n の 3 頂点の奥行きが推定できる。

3.2 全頂点の奥行き推定

式 (7) は、1 つのパッチに対する 3 頂点の推定であるが、これは全パッチの全頂点の推定に容易に拡張できる。 γ_n を拡張し、全頂点の奥行きを逆数を持つパラメータベクトルを、 $\Gamma \equiv (1/d_1, 1/d_2, \dots, 1/d_M)^T$ とする。そして、全ての三角パッチについて加算した SSSD (Sum of SSSD) を、次式のように定める。

$$\sum_n \left[\sum_{\mathbf{u} \in C_n} \{T[\mathbf{u}] - I[\mathbf{w}(\mathbf{u}, \mathbf{p}_n(\mathbf{q}_n(\Gamma)))]\}^2 \right] \quad (8)$$

式 (8) は、 Γ をパラメータとしたコスト関数であるため、最適化手法を用いて最小化することにより、全パッチの奥行きを求めることができる。

3.3 最適化における計算コスト

式 (8) は、全頂点の奥行きに関する簡潔な式であり、非常にシンプルなアイデアによって導かれたものである。しかし、式 (8) を最小化する奥行きを求めるには、膨大な計算コストを必要とする。以下では、最適化手法として Gauss-Newton 法を利用する場合について説明する。

奥行きパラメータ Γ を、現在値 (初期値) Γ_0 と微小変化量 Γ_Δ を用いて、 $\Gamma = \Gamma_0 + \Gamma_\Delta$ とする。Gauss-Newton 法に基づき、 $T[\mathbf{u}] - I[\mathbf{w}(\mathbf{u}, \mathbf{p}_n(\mathbf{q}_n(\Gamma)))]$ を 1 次テイラー展開し、式 (8) を Γ_Δ で微分してゼロとおくと、最終的に次式が得られる。

$$\Gamma_\Delta = -\mathbf{H}^{-1}\mathbf{b}, \quad (9)$$

$$\text{where } \mathbf{H} \equiv -\sum_n \left\{ \sum_{\mathbf{u} \in C_n} [(\mathbf{g}\mathbf{J}\mathbf{K}\bar{\mathbf{L}}_n)^T (\mathbf{g}\mathbf{J}\mathbf{K}\bar{\mathbf{L}}_n)] \right\}, \quad (10)$$

$$\mathbf{b} \equiv \sum_n \left\{ \sum_{\mathbf{u} \in C_n} [(\mathbf{g}\mathbf{J}\mathbf{K}\bar{\mathbf{L}}_n)^T e] \right\} \quad (11)$$

ただし、 e は、2 枚の画像間の画素値の差であり、 $T[\mathbf{u}] - I[\mathbf{w}(\mathbf{u}; \mathbf{p}(\mathbf{q}(\Gamma_0)))]$ で表される。また、 $\mathbf{g}, \mathbf{J}, \mathbf{K}$ は、それぞれ次式で表される。

$$\mathbf{g} \equiv \left[\frac{\partial I}{\partial \mathbf{w}} \right]_{\Gamma=\Gamma_0}, \quad \mathbf{J} \equiv \left[\frac{\partial \mathbf{w}}{\partial \mathbf{p}} \right]_{\Gamma=\Gamma_0}, \quad \mathbf{K} \equiv \left[\frac{\partial \mathbf{p}}{\partial \mathbf{q}} \right]_{\Gamma=\Gamma_0} \quad (12)$$

\mathbf{g} は参照画像 I の勾配を示す 1×2 の列ベクトル、 \mathbf{J} は、画像座標を射影変換パラメータ \mathbf{p} で微分した 2×9 のヤコビ行列であり、どちらも繰り返し計算のたびに再計算

が必要となる。一方、 \mathbf{K} は、式 (3) の関係から得られる 9×3 のヤコビ行列であり、一度計算したら再計算は不要である。また、 $\bar{\mathbf{L}}_n$ は、式 (5) の列数を M (M は頂点数) に拡張した $3 \times M$ の行列であり、 n 番目のパッチに対応する m ($m = i, j, k$) 行以外はゼロになる。この行列は、パッチごと値が異なるものの頂点の画像座標 \mathbf{u}_m のみに依存しているため、再計算は不要となる。

すなわち、このアプローチでは \mathbf{g} と \mathbf{J} の再計算が必要であるため、結果として、画素ごとの計算量が膨大となり、非常に大きな計算コストが必要であることがわかる。

4. 高速化

提案手法では、式 (8) に対する最適化計算を高速化するため、ICIA (Inverse Compositional Image Alignment) [1] の考え方を取り入れた平面パラメータ推定の高速化法 [13] を、複数の三角パッチに適用するように拡張する。

式 (1) によって表現される平面射影変換は、行列の現在値 \mathbf{P}_0 と微小変化量 \mathbf{P}_Δ を用いて、次式のように表すことができる。

$$\tilde{\mathbf{u}} \sim \mathbf{P}_0(\mathbf{I} + \mathbf{P}_\Delta)^{-1}\tilde{\mathbf{u}}' \quad (13)$$

このとき、画像の座標変換 \mathbf{w} は、 \mathbf{P}_0 による座標変換 $\mathbf{w}(\mathbf{u}, \mathbf{p}_0)$ と、 $(\mathbf{I} + \mathbf{P}_\Delta)$ による座標変換 $\mathbf{w}_\Delta(\mathbf{u}, \mathbf{p}_\Delta)$ を用いて、 $\mathbf{w}(\mathbf{u}, \mathbf{p}) = \mathbf{w}(\mathbf{u}, \mathbf{p}_0) \circ \mathbf{w}_\Delta(\mathbf{u}, \mathbf{p}_\Delta)^{-1}$ と表すことができる。ただし、 \mathbf{p}_0 と \mathbf{p}_Δ は、それぞれ \mathbf{P}_0 と \mathbf{P}_Δ の要素をラスタ順に並べたベクトルである。ICIA では、平面射影変換のこの特徴を利用して、式 (2) を

$$\sum_{\mathbf{u}} \{T[\mathbf{w}_\Delta(\mathbf{u}; \mathbf{p}_\Delta)] - I[\mathbf{w}(\mathbf{u}, \mathbf{p}_0)]\}^2 \quad (14)$$

と書くことにより、繰り返し計算ごとに微分要素の再計算不要なアルゴリズムを実現している [1]。

また、平面パラメータ \mathbf{q} を、現在値 \mathbf{q}_0 と初期値 \mathbf{q}_Δ を用いて $\mathbf{q} = \mathbf{q}_0 + \mathbf{q}_\Delta$ とし、 \mathbf{p}_Δ を \mathbf{q}_Δ の関数として表すと、ICIA と同様に、高速なアルゴリズムを実現できることが示されている [13]。

そこで、以下では、[13] の方法と同様に、射影変換パラメータの微小変化量 \mathbf{p}_Δ を、それぞれ、メッシュ全頂点の奥行きパラメータの微小変化量 Γ_Δ を用いて表すことにより、高速化効果が得られることを示す。

まず、式 (8) における $\mathbf{w}(\mathbf{u}, \mathbf{p}_n(\mathbf{q}_n(\Gamma)))$ を、

$$\mathbf{w}(\mathbf{u}, \mathbf{p}_n(\mathbf{q}_n(\Gamma))) = \mathbf{w}(\mathbf{u}, \mathbf{p}_{n0}(\mathbf{q}_{n0}(\Gamma_0))) \circ \mathbf{w}_\Delta(\mathbf{u}, \mathbf{p}_{n\Delta}(\mathbf{q}_{n\Delta}(\Gamma_\Delta)))^{-1} \quad (15)$$

のように合成変換として表現することを考える。ただし、各パラメータの添え字 n は、 n 番目のパッチに関するものであることを示している。このとき、 $\mathbf{p}_{n\Delta}$ と $\mathbf{q}_{n\Delta}$ の関

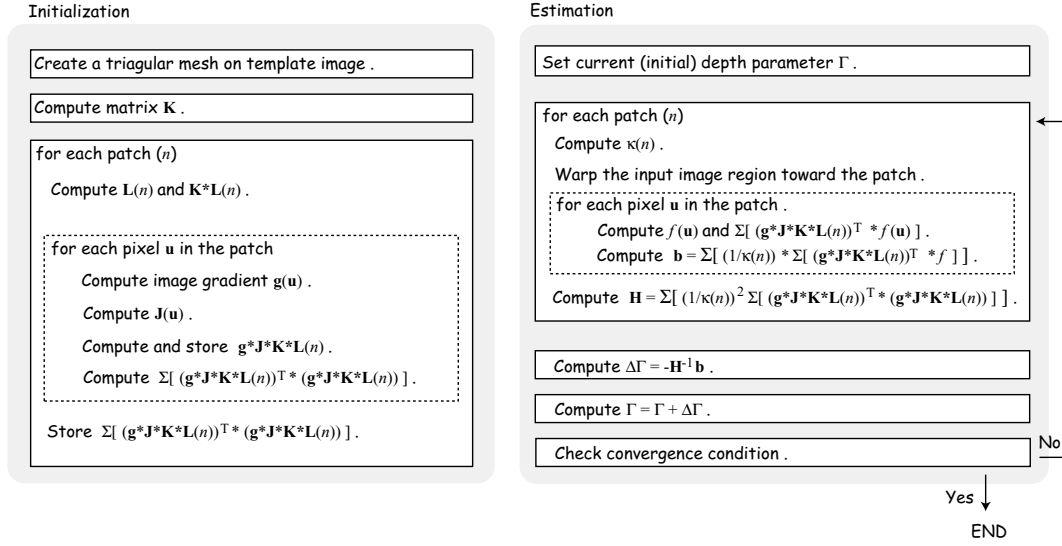


図4 アルゴリズムフロー

係は、次式で書ける [13].

$$\mathbf{P}_{n\Delta} = -\frac{1}{1 + \mathbf{q}_{n0}^T \mathbf{R}^T \mathbf{t} + \mathbf{q}_{n\Delta}^T \mathbf{R}^T \mathbf{t}} \mathbf{R}^T \mathbf{t} \mathbf{q}_{n\Delta}^T \quad (16)$$

また、 $\mathbf{q}_{n\Delta}$ と $\mathbf{\Gamma}_{n\Delta}$ の関係は、式 (4) により、線形関係であることが容易に得られる。これにより、式 (8) は、次式に書き換えることができる。

$$\sum_n \sum_{\mathbf{u} \in C_n} (T[\mathbf{w}_\Delta(\mathbf{u}, \mathbf{p}_{n\Delta}(\mathbf{q}_{n\Delta}(\mathbf{\Gamma}_\Delta)))] - I[\mathbf{w}(\mathbf{u}, \mathbf{p}_{n0}(\mathbf{q}_{n0}(\mathbf{\Gamma}_0)))]^2) \quad (17)$$

$T[\mathbf{w}_\Delta(\mathbf{u}; \mathbf{p}_{n\Delta}(\mathbf{q}_{n\Delta}(\mathbf{\Gamma}_\Delta)))]$ を Taylor 展開し、式 (17) を $\mathbf{\Gamma}_\Delta$ で微分してゼロとおくことにより、式 (17) を最小にする $\mathbf{\Gamma}_\Delta$ は次式で得られる。

$$\mathbf{\Gamma}_\Delta = -\mathbf{H}^{-1} \mathbf{b}, \quad (18)$$

$$\text{where } \mathbf{H} \equiv -\sum_n \left\{ \frac{1}{\kappa_n} \sum_{\mathbf{u} \in C_n} [(\mathbf{g} \mathbf{J} \mathbf{K} \mathbf{L}_n)^T (\mathbf{g} \mathbf{J} \mathbf{K} \mathbf{L}_n)] \right\}, \quad (19)$$

$$\mathbf{b} \equiv \sum_n \left\{ \frac{1}{\kappa_n} \sum_{\mathbf{u} \in C_n} [(\mathbf{g} \mathbf{J} \mathbf{K} \mathbf{L}_n)^T e] \right\} \quad (20)$$

ここで、 e は、2枚の画像間の画素値の差であり、式 (11) と同じものである。ただし、 \mathbf{g} 、 \mathbf{J} および \mathbf{K} は、式 (11) とは異なり、以下ようになる。

$$\mathbf{g} \equiv \left[\frac{\partial T}{\partial \mathbf{w}_\Delta} \right]_{\mathbf{\Gamma}_0=0}, \quad (21)$$

$$\mathbf{J} \equiv \left[\frac{\partial \mathbf{w}_\Delta}{\partial \mathbf{p}_\Delta} \right]_{\mathbf{\Gamma}_0=0} = \begin{bmatrix} u & v & 1 & 0 & 0 & 0 & -u^2 & -uv & -u \\ 0 & 0 & 0 & u & v & 1 & -uv & -v^2 & -v \end{bmatrix} \quad (22)$$

$$\mathbf{K} \equiv \left[\frac{\partial \mathbf{p}_\Delta}{\partial \mathbf{q}_\Delta} \right]_{\mathbf{\Gamma}_0=0} = \begin{bmatrix} \alpha_1 & 0 & 0 & \alpha_2 & 0 & 0 & \alpha_3 & 0 & 0 \\ 0 & \alpha_1 & 0 & 0 & \alpha_2 & 0 & 0 & \alpha_3 & 0 \\ 0 & 0 & \alpha_1 & 0 & 0 & \alpha_2 & 0 & 0 & \alpha_3 \end{bmatrix}^T \quad (23)$$

$$\text{where } \alpha_1 \equiv t_x r_1 + t_y r_4 + t_z r_7, \quad (24)$$

$$\alpha_2 \equiv t_x r_2 + t_y r_5 + t_z r_8, \quad (25)$$

$$\alpha_3 \equiv t_x r_3 + t_y r_6 + t_z r_9. \quad (26)$$

ただし、 $\{r_1, r_2, \dots, r_9\}$ は、 \mathbf{R} の要素をラスタ順に並べたものであり、 $(t_x, t_y, t_z)^T \equiv \mathbf{t}$ である。また、 κ_n は次式で書ける [13].

$$\kappa_n \equiv -(1 + \mathbf{q}_{n0}^T \mathbf{R}^T \mathbf{t}), \quad (27)$$

式 (8) を用いた場合と異なる主な点は、 \mathbf{g} が基準画像の勾配を表し、 \mathbf{J} も画素値のみに依存しているため、どちらも一度計算すれば再計算が不要な点となる。ただし、ICIA による平面射影変換行列の推定 [1] や、平面パラメータ推定 [13] の場合とは異なり、ここでは、ヘッセ行列 \mathbf{H} を固定化することができない。これは、 κ_n が平面パラメータの現在値 \mathbf{q}_{n0}^T に依存 (すなわち $\mathbf{\Gamma}_0$ に依存) しているため、繰り返し計算ごとに、係数 κ_n を再計算する必要が生じるからである。

しかし、 \mathbf{H} の計算に注目すると、行列 $\sum [(\mathbf{g} \mathbf{J} \mathbf{K} \mathbf{L}_n)^T (\mathbf{g} \mathbf{J} \mathbf{K} \mathbf{L}_n)]$ は各パッチごと異なるものの、繰り返しごとに変化せず、さらに、この行列は実質的に $3 \times 3 = 9$ 個の要素しかもっていないため、全パッチに関して加算しても計算量は比較的小さい。また、 \mathbf{b} の計算は、実質的に画素数に依存しており、単一平面のパラメータ推定や、単一平面の3頂点の奥行き推定を、同じ画素数 (同じ画像

領域サイズ)を用いて計算した場合とほぼ同じ計算量となる。すなわち、式(18)の計算量全体を評価すると、同じ画素数を用いて単一平面のパラメータ推定をした場合と比較して、 \mathbf{H} の逆行列計算の分だけが計算コストの増加といえる。ただし、 \mathbf{H} の行列サイズは、 $M \times M$ であり、頂点の数が多くなるほど、その計算量は大きくなる。

本章の最後に、微小平面モデルを用いた曲面形状推定の提案アルゴリズムを図4に示す。このアルゴリズムは、初期化処理と推定処理に分けられる。初期化処理では、推定処理にて利用する固定値(繰り返しごとに変化しない行列やベクトル)を計算している。また、推定処理では、式(18)の計算に必要な値を各パッチについて求め、 Γ_{Δ} の計算と Γ の更新を収束するまで繰り返す。

5. 実験結果

本章では、提案手法の有効性を確認するために行った実験結果を示す。ここでは、合成画像を利用したさまざまなシミュレーション実験の結果について示し、最後に、実画像を用いた実験結果を示す。

5.1 シミュレーション実験

シミュレーション実験では、 512×512 の画像を利用し、ステレオカメラの設定パラメータは、以下に示す値を利用した。カメラの内部パラメータは、基準および参照カメラとも同じであり、次式の \mathbf{A} で示す。

$$\mathbf{A} = \begin{bmatrix} 600 & 0 & 255.5 \\ 0 & 600 & 255.5 \\ 0 & 0 & 1 \end{bmatrix} \quad (28)$$

また、カメラの外部パラメータは、 $\mathbf{R} = \mathbf{I}$, $\mathbf{t} = (0.3, 0, 0)^T$ とした。

図5は、基準画像の光軸上15m先を中心とした半径約7mの球体に、テクスチャを貼り付けることによって生成したステレオ画像である(実形状は図6(a)(b)を参照)。図5(a)(b)は左画像と右画像を示しており、本実験では、右画像を基準画像としている。また、それぞれの画像は、 512×512 の画像における中央部分の 420×420 領域を表示している。このようにして作成した画像に対し、同図(c)のメッシュを基準画像上に生成し、提案手法による表面モデル生成を行った。同図(c)のメッシュの頂点数は61、パッチ数は96である。また、頂点の奥行き初期値は、全頂点において10mとした。

本実験では、収束条件を特に設定せず、30回まで繰り返し計算を行った。プログラムはC言語で実装し、逆行列計算にはCLapackライブラリを利用した。図5(c)で示したメッシュを利用した場合、Pentium IV(2.8GHz)のLinux OSにおいて、30回の繰り返し計算に要した時間は、約0.9秒であった。

図6に、図5の物体の実形状と、30回の繰り返し計算の後に得られた形状を示す。同図により、推定が適切に行われていることがわかる。また、メッシュが収束してゆく様子を図7に示す。図7では、基準画像上のメッシュを白線で示し、真値から計算される参照画像上の頂点を、推定された奥行きによって得られる平面射影変換を用いて基準画像上に投影し、その点によって構成されるメッシュを灰色線で示した。この場合、正しい奥行きが推定されれば、白線と灰色線のメッシュは完全に一致する。同図により、繰り返し計算により、2つのメッシュが一致してゆくことがわかる。さらに、繰り返し計算における画像RMSEの変化、および奥行き真値と推定値との誤差(絶対値)を、図8に示す。図8(a)は画像RMSEを示しており、いくつかのパッチにおける画像RMSEと、全パッチに関する画像RMSEを示している。また、図8(b)は、いくつかの頂点の絶対誤差と、全頂点に関するRMSEを示している。これらにより、ほぼ20回の繰り返し後に形状が正しく推定できていることがわかる。

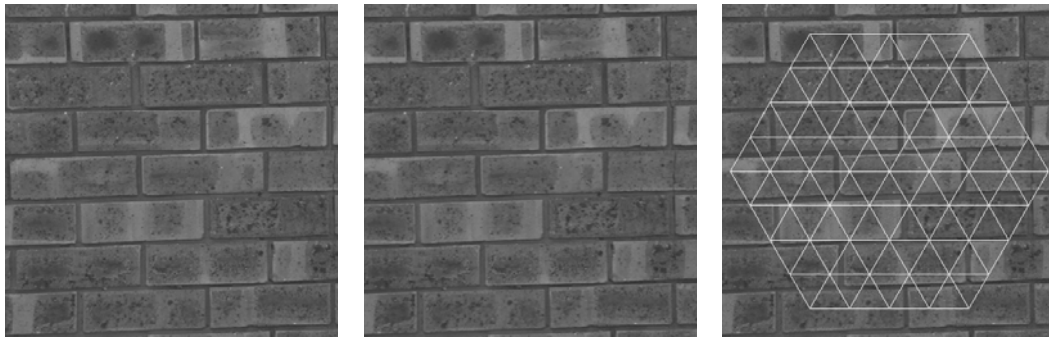
5.2 実画像実験

実画像実験に利用した画像を図9に示す。この画像は、月表面を模擬した 1280×960 のステレオ画像である。本実験では、粗いメッシュを徐々に細かくしてゆく方法を用いて表面モデルを生成した。ここでは、一辺の長さが464[pix]の三角パッチから始め、次の推定ではパッチサイズを半分にするるとともに、前回の頂点の奥行きを初期値とした推定を行った。図9(c)の基準画像上のメッシュは、最終パッチサイズとして使用した29[pix]のメッシュを示している。このとき、メッシュの頂点数は817、パッチ数は1536であった。

生成された表面モデルを図10に示す。メッシュの端に位置する頂点は、その外側にメッシュが存在しないため、他のメッシュによる十分な拘束を受けることができず推定が不安定になる傾向が見られる。しかし、メッシュ内部の領域の推定は安定であり、メッシュ中央部に見られるエッジを持った勾配や、左上部にみられる小さなクレータの穴がおおむね良好に復元されている。

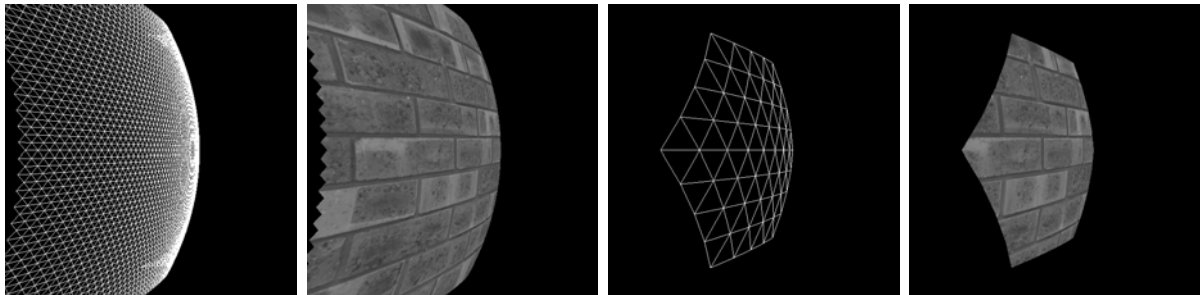
6. まとめ

本論文では、ステレオ画像を利用して、対象物体の表面形状を効率的に推定する手法を提案した。提案手法では、複数の小三角パッチによるポリゴンメッシュを用いて物体表面を近似し、パッチ頂点の奥行きを、全パッチに関するSSSDを最小にする値として、Gauss-Newton法を用いて最適化した。さらに、ICIAの考え方を利用することにより、効率的な表面形状推定が可能であることを示した。



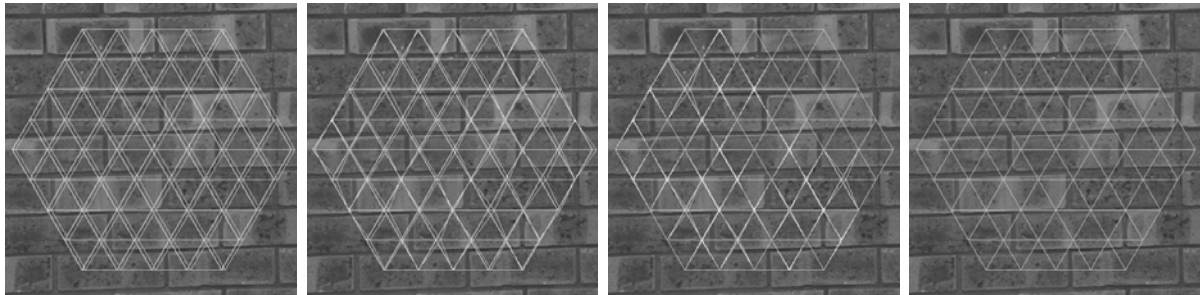
(a) 参照画像 (左) (b) 基準画像 (右) (c) 推定時使用メッシュ (右)

図 5 合成画像および推定時使用メッシュ. (a)(b):対象は光軸上の 15m 先に中心を持つ半径約 7m の球体. 表面にテクスチャを貼り付けた画像を生成. (c):推定に利用したメッシュ(50 ピクセルの辺を持つ正三角形によって生成).



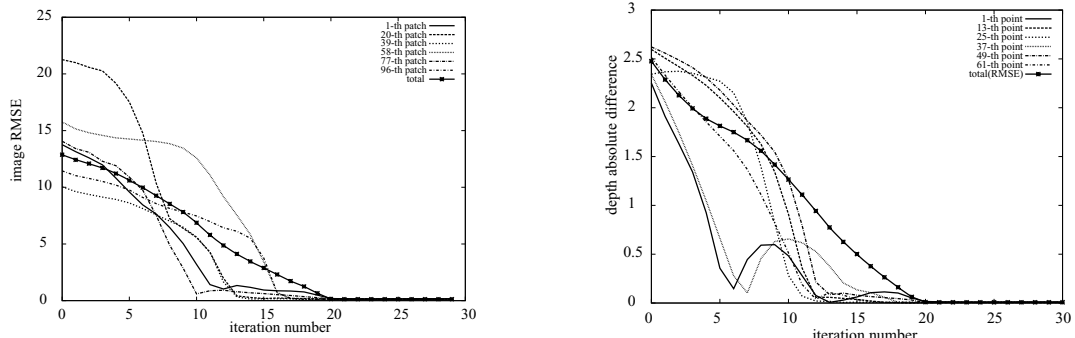
(a) 実形状 (メッシュ表示) (b) 実形状 (テクスチャ表示) (c) 推定形状 (メッシュ表示) (d) 推定形状 (テクスチャ表示)

図 6 図 5 の合成画像を用いたの実験における実形状と推定形状. (a)(b):実形状を表示したもの. (c)(d):推定形状を表示したもの. (a)~(d) は全て同一視点から見たもの.



(a) 初期状態 (b) 繰り返し 7 回 (c) 繰り返し 14 回 (d) 繰り返し 21 回

図 7 図 5 の合成画像を用いた実験におけるメッシュの収束の様子. 基準画像上のメッシュを白で示す. 灰色は, 真値の奥行きから計算される参照画像上の頂点を, 推定された奥行きによって得られる平面射影変換を用いて基準画像上に投影したもの.

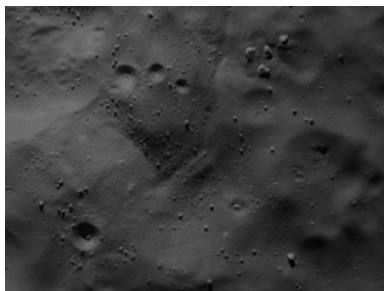


(a) 画像の RMSE (b) 奥行き誤差の絶対値

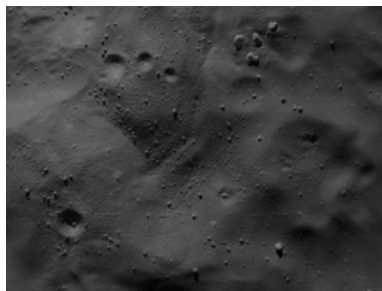
図 8 図 5 の合成画像を用いた実験における画像 RMSE および奥行き誤差の変化

実シーンのモデル化は, 奥行き不連続部分やオクルージョンの検出, および, それに伴うパッチの移動や削除などが有効であり, その具体的な手法の考案が今後の課

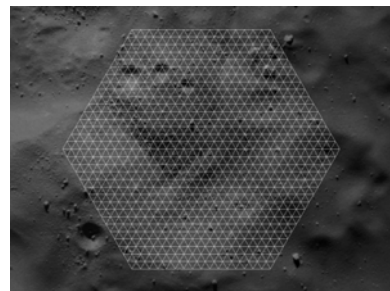
題である. また, ステレオ画像間でのパッチの明るさの違いや, 鏡面反射の影響などを排除する方法についても検討する予定である.



(a) 基準画像 (左)

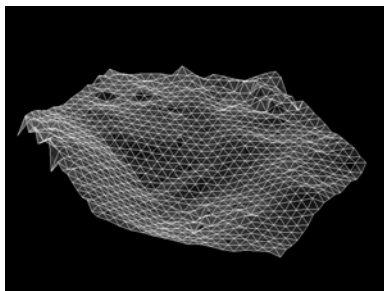


(b) 参照画像 (右)

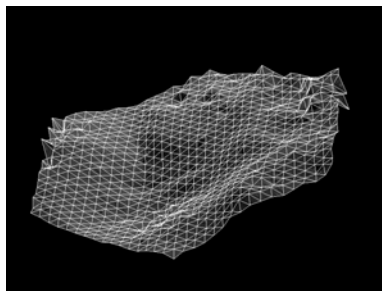


(c) 基準画像上のメッシュ

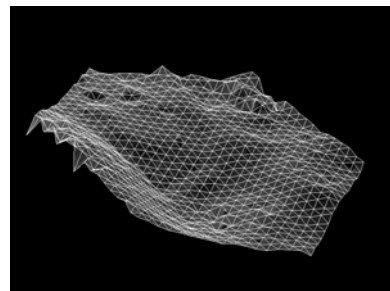
図 9 実画像実験に使用したステレオ画像



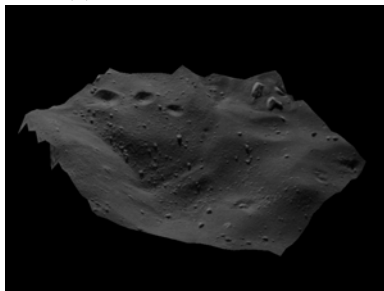
(a) 推定されたメッシュ形状



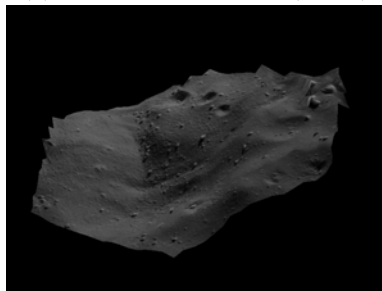
(b) 推定されたメッシュ形状 (別視点)



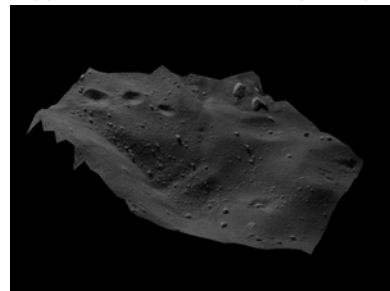
(c) 推定されたメッシュ形状 (別視点)



(d) (a) のテクスチャマッピング結果



(e) (b) のテクスチャマッピング結果



(f) (c) のテクスチャマッピング結果

図 10 生成されたサーフェス

謝 辞

実画像実験画像を提供して下さった、宇宙航空研究開発機構の片山保宏氏に深く感謝いたします。

文 献

- [1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, Vol. 56, No. 3, pp. 221–255, 2004.
- [2] I. Cohen, L. D. Cohen, and N. Ayache. Introducing new deformable surfaces to segment 3d images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 738–739, 1991.
- [3] O. D. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *European Conference on Computer Vision*, Vol. I, pp. 379–393, 1998.
- [4] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *Report de Recherche de l'INRIA*, 1988.
- [5] P. Fua. From multiple stereo views to multiple 3D surfaces. *International Journal of Computer Vision*, Vol. 24, No. 1, pp. 19–35, August 1997.
- [6] P. Fua and Y. G. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *International Journal of Computer Vision*, Vol. 16, No. 1, pp. 35–56, September 1995.
- [7] M. Lhuillier and L. Quan. Surface reconstruction by integrating 3D and 2D data of multiple views. In *IEEE International Conference on Computer Vision*, Vol. 2, pp. 1313–1320, October 2003.
- [8] B. D. Lucas and T. Kanade. An iterative image registration technique with an approach to stereo vision. In *Image Understanding Workshop*, pp. 121–130, 1981.
- [9] D. R. Murray. *Patchlets: A method of interpreting correlation stereo 3D data*. PhD thesis, Department of Computer Science, The University of British Columbia, 2003.
- [10] P.J. Narayanan, P. W. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. In *IEEE International Conference on Computer Vision*, pp. 3–10, 1998.
- [11] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, pp. 715–729, 1991.
- [12] S. Sclaroff and J. Isidoro. Active blobs. In *IEEE International Conference on Computer Vision*, pp. 1146–1153, 1998.
- [13] 杉本茂樹, 奥富正敏. ステレオ画像を利用した平面姿勢推定手法と多眼カメラへの拡張. 情報処理学会研究報告 (コンピュータビジョンとイメージメディア 2005-CVIM-151), Vol. 2005, No. 112, pp. 131–138, November 2005.
- [14] R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. In *Computer Graphics (SIGGRAPH)*, Vol. 26, pp. 185–194, 1992.