法線マップ推定と制約の対話的編集を用いた単一画像からの形状モデリング

岡部 誠　　山梨 傑　　岡村 徹

静岡大学

E-mail: {okabe.makoto, yamanashi.takeru.17, okamura.toru.19}@shizuoka.ac.jp

概要

我々は、1 枚の画像をもとに 3 次元形状を迅速かつ容易にモデリングするための手法を提案する。提案手法は 2 つの技術から構成される： 1) 法線マップ推定器と 2) 幾何学的制約の対話的編集である。法線マップ推定器は、U-net と ResNet を組み合わせた単純な形のネットワークで構成される。法線マップ推定器を学習させるために、カメラ、素材やテクスチャ、照明条件を変えて様々な 3 次元形状をレンダリングし、データセットを構築する。推定された法線マップは正確な形状を得るには十分な精度ではない。そこで、ユーザがインタラクティブに幾何学的制約を編集できる方法を提案する。ユーザは、平面性、直線性、垂直性、平行性などの基本的な幾何学的制約を指定することで、形状を修正することができる。これらの編集は、3 次元ではなく 2 次元の操作であるため、ユーザにとって容易である。提案した手法によって 3 次元モデリングした結果を紹介する。

1　はじめに

単一画像に基づく 3 次元形状モデリングはその利便性から注目を集めている。多視点画像に基づく 3 次元形状モデリングも、形状を正確にモデリングする上で便利だが、撮影プロセスに時間と労力が必要となる [1, 2, 3]。具体的には、1 枚の画像の獲得は容易な場合が多いが、多視点画像の獲得となると撮影機材の準備や取材のための出張が必要になることがある。特に対象物体が大規模な場合、その困難さは増す。そこで単一画像、またはできる限り少ない枚数の画像を元に 3 次元形状モデリングを行いたいという要望がある。

単一画像からの 3 次元形状、特に深度マップの推定に関する研究が活発に行われている。多数の手法が存在し、近年の手法は計算時間も短く精度も高くなってきた [4, 5, 6, 7]。だが、これらの手法により出力される形状の精度にはしばしば問題が存在する。例えば、図 1 に示すような、ZoeDepth[7] の結果は、形状を大まかにはよく再現できており、用途によってはこのままで十分使用可能である。しかし、これをそのまま適用すると多くの修正が必要となるようなアプリケーションもある。具体的には、平面であるべき箇所が歪んで曲面になっていたり、一部の面と面は垂直や平行の関係にあるべきなのにそうなっていないといった問題がある。既存の 3 次元モデリングソフトウェアを使用してこれらの問題を修正することが簡単かと言えば、むしろ困難な作業になることが多い。

このような背景から、我々は入力画像に対し簡単なアノテーションを追加することでこれらの問題を解決する方法を提案する。具体的には、入力画像に対し、平面である箇所や直線である箇所を指定できる。また、それら複数の平面や直線同士の関係性に制約（例えば垂直である、平行である等）を設けることが可能である。提案手法は、ユーザが指定した制約を満たすように深度マップを修正する。制約を考慮した深度マップの修正には thin plate 関数の最適化を採用する。

全てのアノテーション作業は 2 次元的な入力操作になるため、ユーザにとって扱いやすいと考えられる。今回の実験では、初期形状として ZoeDepth[7] のような手法で得られる深度マップではなく、法線マップ推定器を用いている。まず、入力画像から法線マップを推定し、次にその法線マップから 3 次元形状を求めて初期形状とする。提案手法によって生成されたいくつかの結果を紹介する。

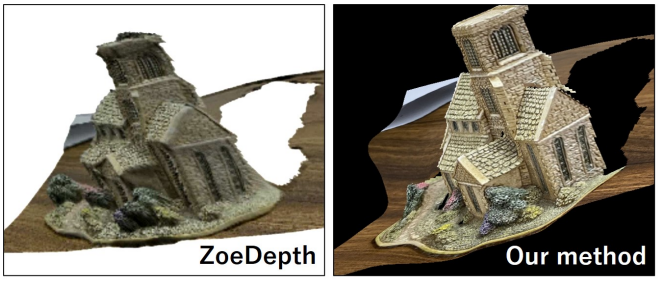

図 1　ZoeDepth[7] は形状を大まかには良く再現出来ているが、細かく見ると平面が歪んでいたり、垂直や平行になるべき箇所がそうなっていない。提案手法は簡単なアノテーションでこれらの問題を修正することができる。

2　提案手法

提案手法の概要を図 2 に示す。図 2 の上段は入力画像と初期形状を示している。今回、入力画像から法線マップを推定し、その法線マップとユーザが入力した断絶線の情報を元に初期形状を復元した。初期形状として ZoeDepth[7] のような手法で得られる深度マップを用いても良い。尚、断絶線とは深度値が大きく変化するピクセル（言い換えると面が切断される箇所）を指定する線のことである（入力画像（Input）を拡大するとオレンジ色に見える）。

この初期形状では黄色い積み木の面の歪みが目立つ。そこで、図 2 の中段ではこの歪みを修正するため、ユーザは黄色い積み木の面上にマウス操作で四角形を描いた。この四

角形は平面性を課す制約となるため、最適化を適用すると歪みが修正され、平面的な形状を得ることができる。

次に、赤い積み木に注目すると、3 つの面が歪んでおり、互いに垂直となっていないため、本来の形である立方体が再現されていない。そこで、図 2 の下段では赤と青の積み木に対して 3 つの面を描き、さらにこれら 3 つの面が互いに垂直であることを指定することで赤および青の積み木の形状を修正している。この指定のための具体的な操作は、マウス操作で多角形を 3 つ描いた後、それら 3 つの多角形をクリックして選択状態にし、「垂直性」ボタンを押すことである。

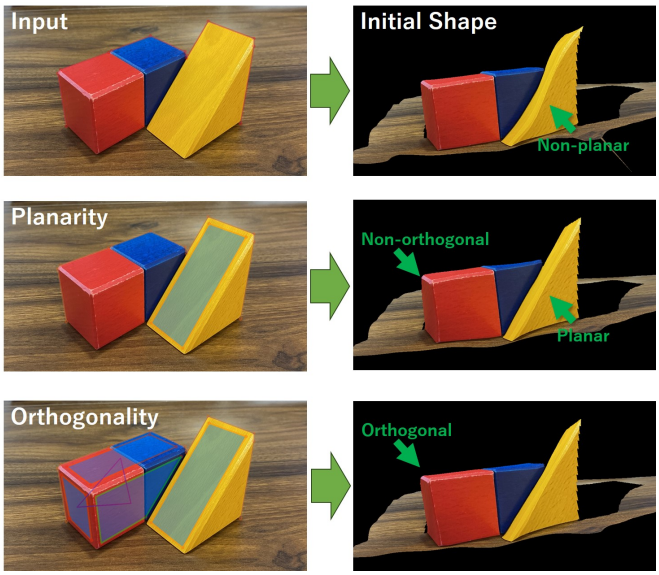

図 2　上段：入力画像と初期形状。中段：黄色い積み木の面の歪みを修正。下段：赤及び青の積み木の面同士が垂直となるように修正。

3　制約に基づく形状の修正

作りたい形状の深度マップにおいて、(x, y) ピクセルの深度値を $d_{x,y}$ と表現するとき、次のようなエネルギー関数を考える：

$$\begin{aligned} E = \sum_{x,y} [&\alpha_{x,y}(d_{x+1,y} - 2d_{x,y} + d_{x-1,y} - a_{x,y})^2 \\ &+ 2\beta_{x,y}(d_{x+1,y+1} - d_{x,y+1} - d_{x+1,y} + d_{x,y} - b_{x,y})^2 \\ &+ \gamma_{x,y}(d_{x,y+1} - 2d_{x,y} + d_{x,y-1} - c_{x,y})^2] \end{aligned} \quad (1)$$

このエネルギー関数は thin plate 関数に基づくものである [8]。Thin plate 関数は表面の滑らかさを表す尺度だが、ここでは 3 つの定数 $a_{x,y}$、$b_{x,y}$、$c_{x,y}$ が追加されている。この理由は、我々は既に初期形状を持っているからであり、初期形状の深度マップの (x, y) ピクセルの深度値を $e_{x,y}$ と表現すると、

$$\begin{aligned} a_{x,y} &= e_{x+1,y} - 2e_{x,y} + e_{x-1,y} \\ b_{x,y} &= e_{x+1,y+1} - e_{x,y+1} - e_{x+1,y} + e_{x,y} \\ c_{x,y} &= e_{x,y+1} - 2e_{x,y} + e_{x,y-1} \end{aligned}$$

となる。即ち、式 1 を最小化するような深度値 $d_{x,y}$ を求めることは、初期形状をなるべく満たすような形状を得ることに等しく、式 1 に加えて特に制約がない場合には初期形状がそのまま得られる。式 1 の最小化には最急降下法を用いる。

提案手法は thin plate 関数を用いた単一画像からの 3 次元モデリングの手法 [8] と関連している。既存手法との違いは初期形状から得られる 3 つの定数 $a_{x,y}$、$b_{x,y}$、$c_{x,y}$ が追加されていることと、以下に述べるような平面性や垂直性のような新たな制約が追加されたことである。

3.1　平面性

平面性制約は図 2 の中段に示すように、形状に平面性を課す制約である。ユーザがマウス操作で多角形を描いたとき、その多角形の内側に含まれるピクセルに対応する 3 次元点群を $\{(x_i, y_i, d_{x_i,y_i})\}$ と表現する。この 3 次元点群に対し主成分分析を適用することで、3 次元点群に最もフィットする 3 次元平面の式を得ることができる。その平面の式を $Ax + By + Cz + D = 0$ とすると、$z = \frac{-D-Ax-By}{C}$ であるから、以下の式を最小化するような深度値 $d_{x,y}$ を求めることは、平面性を課すことに相当する。

$$E_{planarity} = \sum_i (\frac{-D - Ax_i - By_i}{C} - d_{x_i,y_i})^2 \quad (2)$$

ユーザが指定した平面性制約に対応する上式を式 1 を加えて最小化問題を解くことによって平面性を持った形状を得ることができる。ただし、平面の式の定数 A, B, C, D は最急降下法のループの中で毎回変化することに注意する。

既存手法 [8] でも平面性制約は実現されており、その際は、平面性を課したいピクセル (x, y) に対し、3 つの定数 $a_{x,y}$、$b_{x,y}$、$c_{x,y}$ をゼロに設定するような形で実現されていた（厳密には最小二乗法としてではなく、ハード制約としてラグランジュの未定乗数法を用いて解く）。このような方法では平面性を課された部分に関しては完全な平面ができる一方、法線マップの推定によって得られた細かな凹凸が消えてしまう。そこで上記のような方法を提案した。上記の手法では 3 次元点群 $\{(x_i, y_i, d_{x_i,y_i})\}$ を構築する際、すべてのピクセルを対象とせず疎にサンプリングされたピクセル集合を使うことで、細かな凹凸を維持しながら平面性を課すことを試みている。

3.2　垂直性

垂直性制約は図 2 の下段に示すように、形状に垂直性を課す制約である。図 2 の下段では、ユーザはマウス操作によって 3 つの多角形を描いている。ここでは 3 つある多角形のうちの 2 つを選び、それぞれ p と q として、垂直性を課すための手法について説明する。多角形 p から得られる平面の式を $A_p x + B_p y + C_p z + D_p = 0$ とし、多角形 q から得られる平面の式を $A_q x + B_q y + C_q z + D_q = 0$ とする。垂直性とはこの 2 つの平面が垂直であることを意味するので多角形 p の平面の法線を $N_p = (A_p, B_p, C_p)$、多角形 q の平面の法線を $N_q = (A_q, B_q, C_q)$ と置けば、$N_p \cdot N_q = 0$

が満たすべき制約であるが、今はこの式は成り立っていない（まだ垂直になってないので）。そこで、$(N_p \cdot N_q)^2$ の値をより小さくするように最急降下法で N_p 及び N_q を更新すると、得られる法線は $N'_p = N_p - 2\lambda(N_p \cdot N_q)N_q$ 及び $N'_q = N_q - 2\lambda(N_p \cdot N_q)N_p$ となる。あとは多角形 p の内側に存在する 3 次元点群は N'_p の法線の平面を作るように、多角形 q の内側に存在する 3 次元点群は N'_q の法線の平面を作るように、3.1 章の方法で深度値の更新を行えばよい。

3.3 その他

現在のシステムは平面性に加え、直線性を指定できる。また直線同士の垂直性や、平面と直線の間に垂直性を設定することも可能である。また垂直性に加え平行性も同様の考え方で実現可能である。

4 法線マップに基づく初期形状の生成

初期形状を得るにあたっては既存の深度マップ推定器を使うことができるが、本実験ではまず法線マップを求めた後、ユーザが断絶線を入力し、それら 2 つの情報を元に積分計算によって初期形状を求めている。法線マップの利点は、深度マップを直接求めるよりも滑らかな表面を得られる場合があるためである。一方で深度マップ推定器の方が良い形状が求められる場合もあり、ケースバイケースなので、ことさら本手法の優位性を主張するわけではないが、以下に簡単に法線マップ推定器について説明する。

我々の法線マップ推定器は U-Net である [9]。入力は 256 × 144 ピクセルの RGB 画像であり、出力は 256 × 144 ピクセルの法線マップ（3 チャンネル）である。Pooling は 4 回行っている。エンコーダ部分には ResNet8[10] を用いている。アップサンプリングの際、中間の特徴マップからも画像を生成し (64 × 36 ピクセル及び 128 × 72 ピクセルの法線マップを出力し) 最終的な出力 (256 × 144 ピクセルの法線マップ) と合わせて、計 3 枚の解像度の異なる法線マップとの誤差を取りながら学習している。

法線マップ推定器の学習には図 3 のような CG シーンを用いる。CG シーンは地面を表す平面と、その上に配置された 3 次元物体から成る。配置する 3 次元物体は建物や自動車など様々なものを使用した。地面と物体のテクスチャはランダムに設定される。また、レンダリングの際のカメラの位置と方向もランダムに設定され、照明は複数の点光源から成り、位置や色や輝度はランダムに設定される。レンダリングには局所照明モデルとアンチエイリアシング処理を施したシャドウマップを用いた。現時点の法線マップ推定器は数百万枚から成る CG 画像のデータセットを用いて学習されている。

写真画像から推定された法線マップを図 4 に示す。ソファの画像の場合には，細部の形状は我々の手法の方が捉えられているものの、大まかな形状については既存手法 [11] の方が正確に捉えられているように思われる。

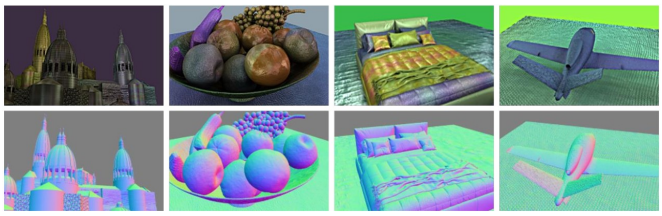

図 3　データセットの一部。上段が入力となる RGB 画像、下段が正解の法線マップ。

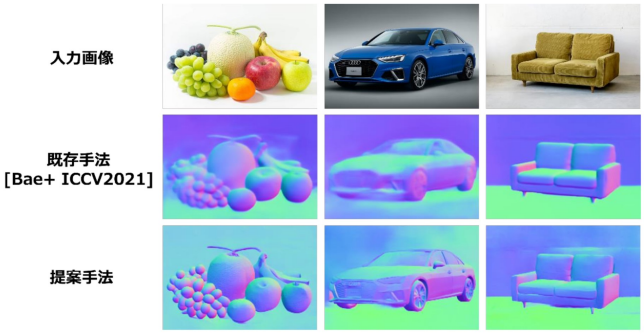

図 4　法線マップ推定器による写真画像の法線マップの推定の結果。既存手法よりも詳細で正確な法線を出力できる場合がある。

5 結果と議論

提案手法の結果を図 5 に示す。初期形状は推定された法線マップと、ユーザが入力した断絶線情報に基づいて計算したものである。いずれの初期形状も面が歪んでいたり垂直性が保たれていない箇所が目立つ。図 5 では、ユーザが指定した制約については多角形の情報のみを示している。制約に基づいて修正された形状では面の垂直性や平行性が再現されている。一方で修正が施された結果、法線マップに見られていたような細かな凹凸が消えてしまった箇所がある。

図 6 に既存手法 ZoeDepth[7] との比較を示す。図 6 の上段は積み木の画像に対する結果である。既存手法では形状の歪みが見えるのに対し提案手法の結果では平面性や垂直性が再現されており本来の積み木により近い形が出来ていると思われる。またこの結果について法線マップが上手く推定されたこともあり、既存手法よりも滑らかな表面を作ることができている。図 6 の下段でも同様のことが言え、既存手法より提案手法の結果の方が歪みが少ない。法線マップ推定器が効果的に働いた事に加え手作業での修正が上手くいった例である。

提案手法の限界は、初期形状の品質に依存していることである。つまり初期形状がある程度上手く求まれば、それを手作業で修正することには有用であるが、初期形状に多くの間違いがある場合、それらを修正するためには多くの制約を入力する必要が出てしまうとともに、制約があまりに多い場合には形状修正の最適化計算が破綻することもある。

また現在の平面や直線、また、それらの垂直性や平行性だけでは修正しきれない形状も多い。今後制約のバリエーションを増やし、対称性や円形や球形などのさまざまの制約

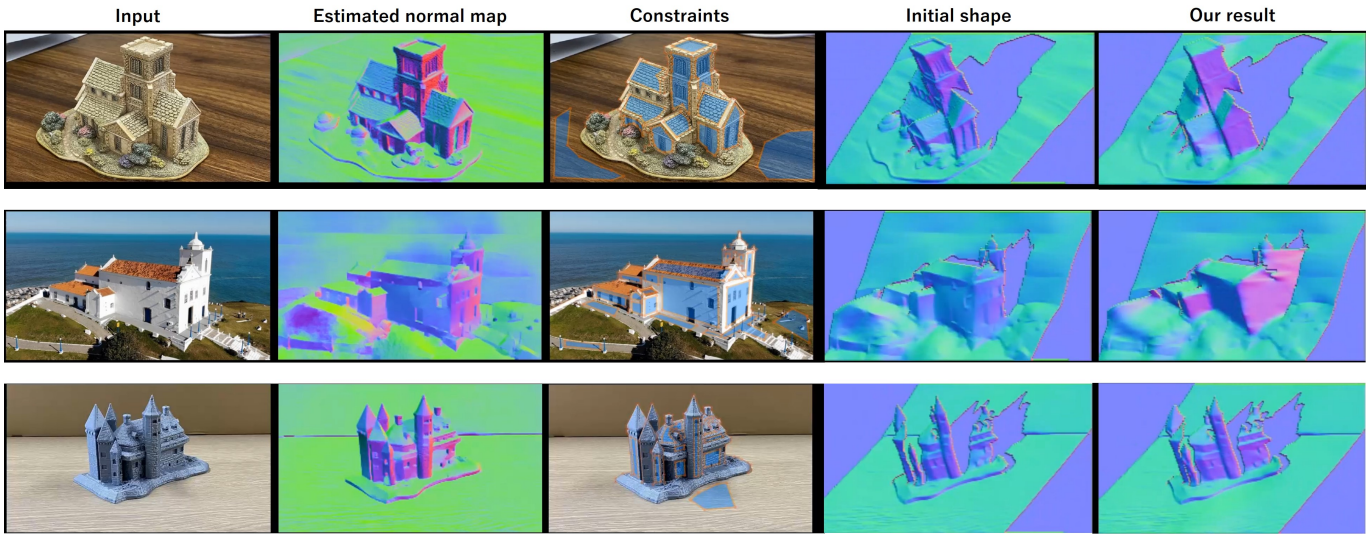

図 5　提案手法の結果。左から入力画像、推定された法線マップ、ユーザが指定した制約、初期形状、提案手法の結果。

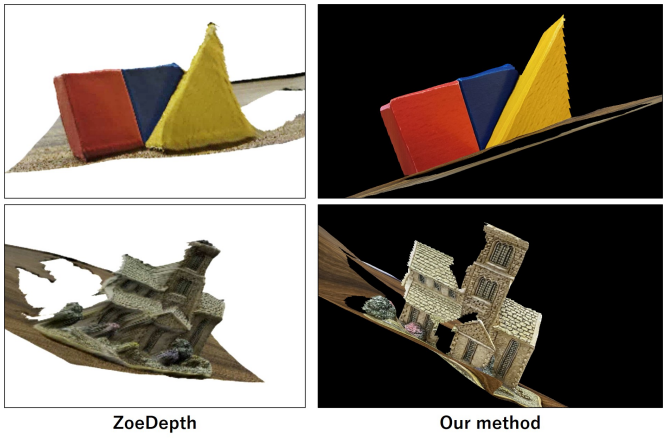

図 6　既存手法 ZoeDepth[7] との比較。

を導入して使いやすいツールを作っていきたい。

参考文献

[1] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *CoRR*, vol. abs/2003.08934, 2020. [Online]. Available: https://arxiv.org/abs/2003.08934

[3] M. Worchel, R. Diaz, W. Hu, O. Schreer, I. Feldmann, and P. Eisert, "Multi-view mesh reconstruction with neural deferred shading," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 6187–6197.

[4] M. Ramamonjisoa and V. Lepetit, "Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 2109–2118.

[5] R. Hu, N. Ravi, A. C. Berg, and D. Pathak, "Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

[6] X. Long, C. Lin, L. Liu, W. Li, C. Theobalt, R. Yang, and W. Wang, "Adaptive surface normal constraint for depth estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 12 849–12 858.

[7] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "Zoedepth: Zero-shot transfer by combining relative and metric depth," 2023. [Online]. Available: https://arxiv.org/abs/2302.12288

[8] L. Zhang, G. Dugas-Phocion, J.-S. Samson, and S. M. Seitz, "Single view modeling of free-form scenes," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015, cite arxiv:1505.04597Comment: conditionally accepted at MICCAI 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[11] G. Bae, I. Budvytis, and R. Cipolla, "Estimating and exploiting the aleatoric uncertainty in surface normal estimation," in *ICCV 2021*, 2021.