

レンジデータの重ね合わせによるバスレリーフ形状生成

Bas-relief modeling by superposition of range data

○木村 拓海 (東京都立大学) *1

長井 超慧 (東京都立大学) *2

*1 Takumi Kimura, Tokyo Metropolitan University, 6-6 Asahigaoka, Hino-shi, Tokyo, Japan, 191-0065, kimura-takumi3@ed.tmu.ac.jp

*2 Yukie Nagai, Tokyo Metropolitan University, 6-6 Asahigaoka, Hino-shi, Tokyo, Japan, 191-0065, y-nagai@tmu.ac.jp

キーワード:バスレリーフ, レンジデータ, ポアソン方程式

1. 緒言

レリーフとは、浮き彫りのことで平面や曲面などに模様を付加したものである (図 1)。古来、権力や宗教の象徴として作られてきた。200 年程前からは芸術品として自由に制作されるようになり、現在では、硬貨・メダル・建築物・磁器など様々なところで用いられている。レリーフにはアンダーカットのあるハイレリーフとアンダーカットのないバスレリーフがある (図 2)。アンダーカットとは、土台に対して模様の一部が張り出している部分のことである。

職人の手作業によるレリーフは、まずスケッチでイメージ図を作り、粘土で立体形状を制作した後、鋳造による加工などにより制作されている。近年では、付加造形技術の発展により、3D プリンタを用いたレリーフ制作も注目されている。3D プリンタを用いた制作には、加工の自動化によるコストや制作時間の削減、様々な形状のレリーフを制作可能などのメリットがある。3D プリンタを用いるには、レリーフの 3D モデルを用意する必要がある。

レリーフの 3D モデルを生成する先行研究は、3D モデルからバスレリーフの 3D モデルを生成する方法⁽¹⁾や 3D モデルからハイレリーフを生成する方法⁽²⁾などがある。また、既存のレリーフの高さマップを入力にして、詳細な形状を保持したまま厚さを薄くしたレリーフの 3D モデルを生成する研究⁽³⁾もある。

先行研究の多くは、単体の 3D モデルや画像を入力にしてレリーフを生成している。もし、複数の 3D モデルや画像を組み合わせてレリーフを生成することができれば、表現できるレリーフの幅が広がると考えられる。そこで、本研究では、レリーフの土台部分と模様部分の 2 つのレンジデータを入力し、ポアソン方程式を用いてバスレリーフの 3D モデル(表面メッシュ)を生成する手法を提案する。



Fig. 1 bas-relief of medal



Fig. 2 Cross sections of bas-relief (left) and high-relief (right).

2. 提案手法

2.1. 提案手法の概要

入力は、バスレリーフの土台部分と模様部分の 2 つのレンジデータである。レンジデータとは、画素値がカメラから被写体までの距離を表す画像のことである。出力は、バスレリーフの表面メッシュ $\{(x, y, h(x, y))\}$ である。提案手法の流れを以下に示す。

1. レリーフ頂点の座標 $(x, y, h(x, y))$ を土台のレンジデータの画素の中心座標で初期化する。
2. 模様部分のレンジデータの勾配 $\mathbf{v}(x, y)$ を計算する。
3. 模様部分の領域 Ω において、レリーフの高さの勾配 $\nabla h(x, y)$ が $\mathbf{v}(x, y)$ に一致するまで、 $h(x, y)$ を繰り返し更新する。

図 3 に入力と提案手法の各ステップの結果を示す。図 3(a) と (b) は入力のレンジデータであり、それぞれ土台部分 (半球) と模様部分 (ドラゴン) である。図 3(c) はステップ 1 で形成された土台部分の半球である。図 3(d) はステップ 2 で計算した模様部分のレンジデータの勾配ノルム $\|\mathbf{v}(x, y)\|$ である。図 3(e) の網掛け部分はステップ 3 の領域 Ω を表している。図 3(f)(g)(h) は、ステップ 3 の繰り返しによりレリーフ頂点の高さ $h(x, y)$ が変化する様子である。模様部分が土台上に徐々に再現されていく。

2.2. ポアソン方程式によるレリーフ頂点位置の決定方法

土台上に模様を再現するために、提案手法ではステップ 3 でレリーフの高さの勾配 $\nabla h(x, y)$ を模様部分の勾配 $\mathbf{v}(x, y)$ に一致させる。図 4 は $\nabla h(x, y)$ と $\mathbf{v}(x, y)$ が一致する様子を表した図である。青い矢印がレリーフの高さの勾配 $\nabla h(x, y)$ を表しており、緑の矢印が模様部分のレンジデータの勾配 $\mathbf{v}(x, y)$ を表している。模様部分において $\nabla h(x, y)$ が $\mathbf{v}(x, y)$ に一致するとき式 (1) が成り立つ。

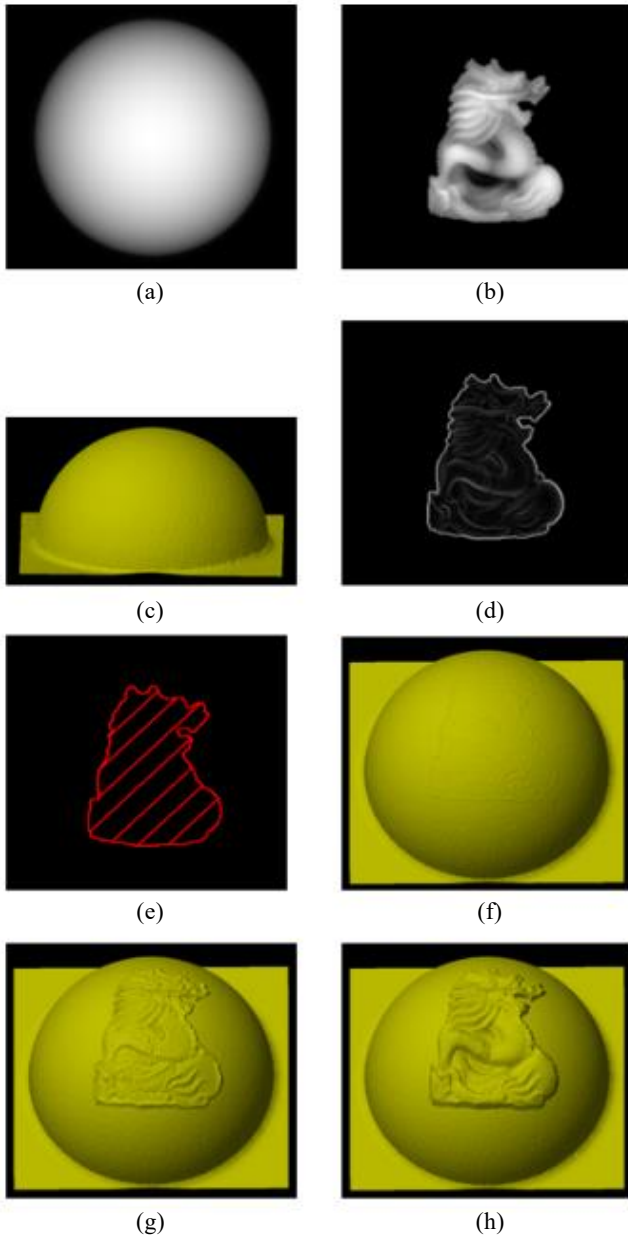


Fig. 3 (a) Input range data for the base, (b) input range data for the foreground pattern (image courtesy of Ji et al.), (c) initialized surface mesh (Step1), (d) gradient norm $\|\mathbf{v}(x,y)\|$ of (b) (Step2), (e) area Ω (red hatched region), (f)-(h) after 5, 100, and 30,000 iterations of Step 3.

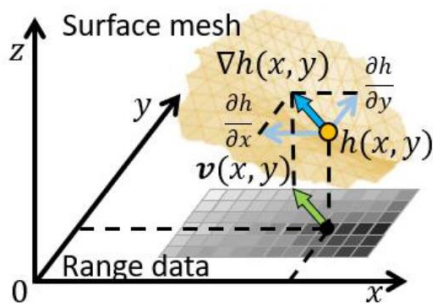


Fig. 4 Concept of matching the gradient of the height function $\nabla h(x,y)$ to the gradient of the intensity of foreground pattern $\mathbf{v}(x,y)$.

$$\nabla h(x,y) = \mathbf{v}(x,y) \text{ over } \Omega \quad (1)$$

理想は、領域 Ω の全ての点 (x,y) において式 (1) が成り立つことである。この最適な点 $h(x,y)$ を得るために、式 (2) の最小化問題を解く。

$$\iint_{\Omega} \|\nabla h(x,y) - \mathbf{v}(x,y)\|^2 dx dy \rightarrow \min \quad (2)$$

$$\text{s.t. } h(x,y)|_{\partial\Omega} = I_b(x,y)|_{\partial\Omega}$$

ただし、 $\partial\Omega$ は領域 Ω の境界、 $I_b(x,y)$ は土台部分のレンジデータの画素値である。最小化問題 (2) は、領域 Ω の全ての点において、 $\nabla h(x,y) - \mathbf{v}(x,y)$ の L^2 ノルムを計算し、その総和を最小にするという意味である。この最小化問題を、Poisson Image Editing⁽⁴⁾を参考にして解く。式 (1) の両辺の発散を計算し、式 (3) のポアソン方程式を得る。

$$\Delta h(x,y) = \text{div } \mathbf{v}(x,y) \quad (3)$$

式 (3) を満たす領域 Ω の注目画素 p の高さ h_p は、式 (4) の更新式を繰り返し解くことにより求まる。

$$h_p \leftarrow \frac{1}{4} (\sum_{q \in N_p} h_q + \sum_{q \in N_p} v_{pq}) \quad (4)$$

$$v_{pq} = (g_p - g_q)/s$$

ただし、 N_p は注目画素 p の4近傍の画素の集合、 q は N_p の要素、 g_p は模様部分の画素 p の画素値、 v_{pq} はベクトル $\mathbf{v}(x,y)$ の pq 方向の成分、 s は画素辺の長さである。実験では、 v_{pq} に急激な高さ変化を抑えるために $\lambda = 0.03$ をかけた。

3. 実験結果

実験環境は、Windows 10 Home 64-bit, 3.9 GHz AMD Ryzen 7 3800XT CPU, 32GB RAM である。プログラムはC++で実装した。

図5は、半球とドラゴンのレンジデータを入力にした実験結果である。解像度は 675×615 、領域 Ω の画素数は87,623である。図5(c)(d)は半球にドラゴンを重ね合わせた結果である。半球上にうまくドラゴンを再現することができた。

図6は、半球とスタンフォードドラゴンのレンジデータを入力にした実験結果である。解像度は 700×700 、領域 Ω の画素数は116,588である。本実験では、ステップ2の $\mathbf{v}(x,y)$ に対してメディアンフィルターを適用し、突出した値を抑制した。こちらも半球上に上手くドラゴンを再現することができた。

図7は、階段とスタンフォードバニーのレンジデータを入力にした実験結果である。解像度は 700×700 、領域 Ω の画素数は61,620である。図7(c)(d)は階段上にウサギの模様を重ね合わせた結果である。ウサギの模様を土台の段々の影響を受けずに再現することができた。

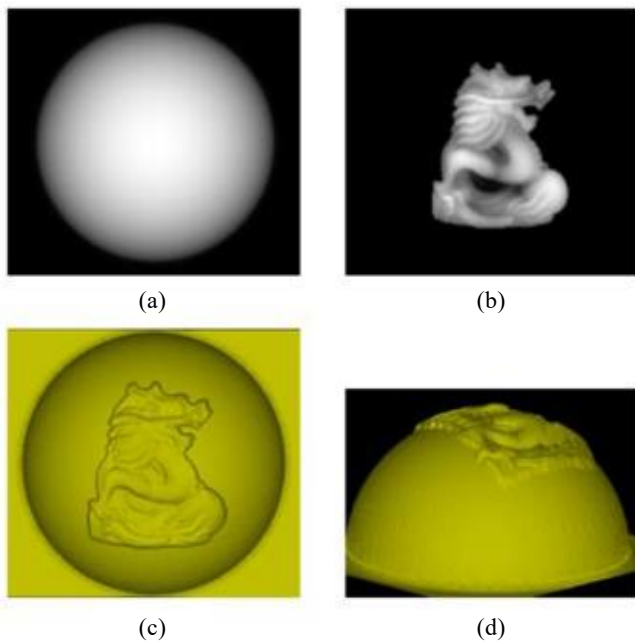


Fig. 5 Result for hemisphere and dragon. (a) and (b) input range data, (c) and (d) result relief from different viewpoints.

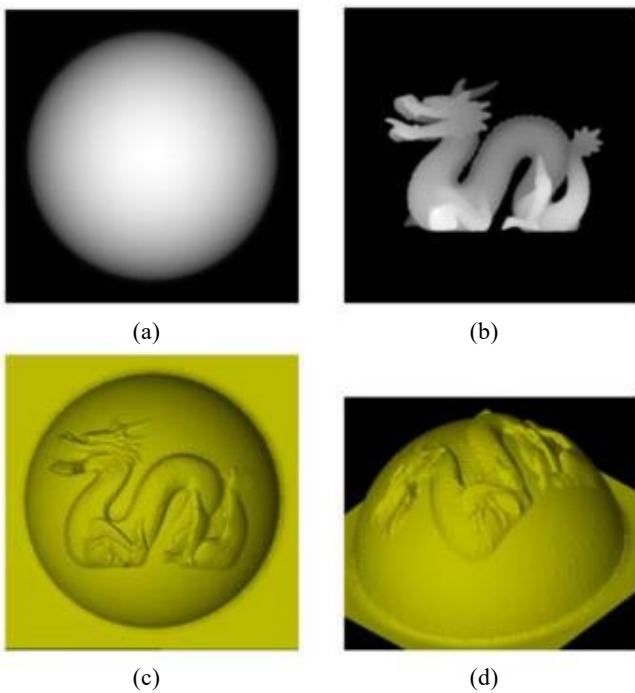


Fig. 6 Result for hemisphere and Stanford dragon. (a) and (b) input range data, (c) and (d) result relief from different viewpoints.

4. 結 言

本研究では、レリーフの土台部分と模様部分の2つのレンジデータを入力とし、土台上に模様を再現したレリーフの表面メッシュを生成する手法を提案した。レリーフの高さの勾配が模様部分のレンジデータの勾配に一致するようなポアソン方程式を解くことで、模様の特徴を損なわずに土台上に模様を再現することができた。

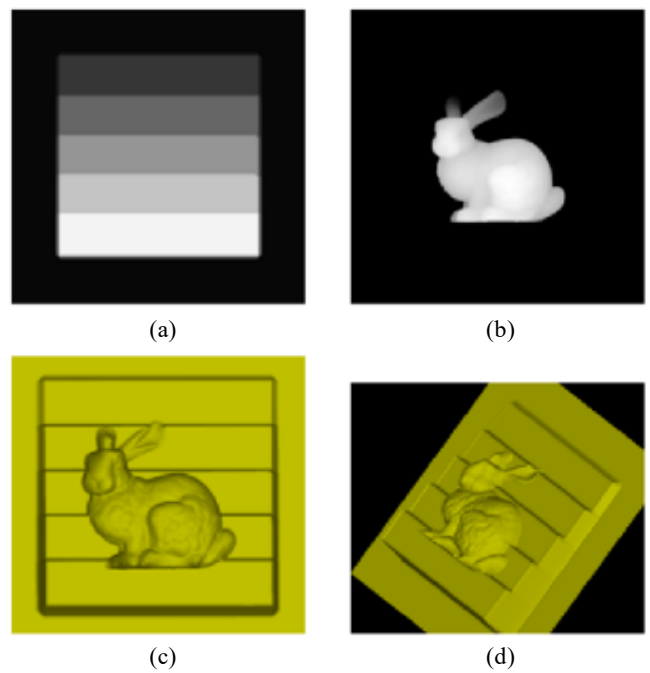


Fig. 7 Result for staircase and Stanford bunny. (a) and (b) input range data, (c) and (d) result relief from different viewpoints.

文 献

- (1) Tim Weyrich et al. Digital Bas-Relief from 3D Scenes. ACM Transactions on Graphics (Proc. SIGGRAPH) 26(3), 2007.
- (2) Sami Arpa et al. High Reliefs from 3D Scenes. Computer Graphics Forum vol. 34, Issue 2, pp. 253-263, 2015
- (3) Zhongping Ji et al. Bas-Relief Modeling With Detail Preservation and Local Significance Enhancement. IEEE Access vol. 8, pp. 44190-44201, 2020.
- (4) Patrick Pérez et al. Poisson image editing. ACM Transactions on Graphics vol. 22, No. 3, pp. 313-318, 2003.