疎な多視点画像からの樹木のボリュームモデリング

岡部 誠

静岡大学

E-mail: m.o@acm.org

1 はじめに

Computed tomography (CT) によるボリュームモデリングは、物体の表面形状だけでなく、物体の内部構造を可視化できるため、医療や教育、また、エンターテインメントのためのコンテンツ生成技術として広く利用されている。CT でボリュームをモデリングするには物体を多視点で撮影した画像が必要だが、それらの撮影には特別な装置が必要であるなど、しばしば多大なコストを要する。また、医療分野では多数の X 線画像を撮影する場合に、X 線が人体に与える悪影響についても議論されている。そこで我々は、既存の CT 技術を拡張し、なるべく少ない枚数の画像集合からでもボリュームをモデリングできるような技術を研究している。

少ない枚数の画像集合からボリュームをモデリングすることについて議論する。ここでは、樹木を正面（方位角が 0°）と真横（方位角が 90°）から撮影した 2 枚の画像のみが与えられているとする。CT 技術の 1 つである最小二乗法に基づく手法 [1] を用いてボリュームをモデリングし、それを様々な視点からレンダリングした画像を図 1 に示す。入力の 2 枚の画像と同じ方向からレンダリングすると、入力画像と同じ見た目の画像が得られる（図 1 の上段の 0° と 90°）。ところが、視点を変えるとレンダリングされる画像はぼやけた画像となり、そこに樹木の構造は見られない（図 1 の上段の 20° と 45° と 70°）。また、図 1 の下段に示すのは、上段と同じ方位角で、少し上空からボリュームを見下ろすようにレンダリングした画像だが、ぼやけて見えると同時に、入力の 2 枚の画像が最小二乗法によって投影された痕跡がグリッド線として見える。もし、樹木のボリュームが正しくモデリングされていたなら、視点を変えたとしても、このようにぼやけて見えたり、グリッド線が見えたりするようなことはないはずである。

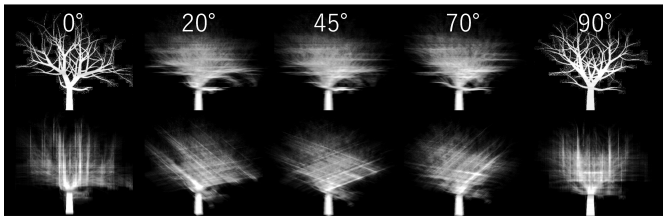

図 1: 2 枚の樹木の画像からモデリングしたボリューム。

そこで我々のアイデアは、視点を変えてレンダリングしても、入力画像と同様の質感となるようにボリュームを修正しようというものである。具体的には、レンダリングされた画像の質感と入力画像の質感との差をコスト関数とし、それを最小化するようにモデリングを行うことで、どこから見ても入力画像と同様の質感に見えるようなボリュームをモデリングする。結果的に少ない枚数の画像集合からでも良いボリュームがモデリングできる。

今回は対象を樹木に限定して実験を行ったので、その結果を報告する。提案手法を用いることで、入力画像が 1 枚しか与えられない場合でも図 2 のような樹木のボリュームをモデリングすることができる。

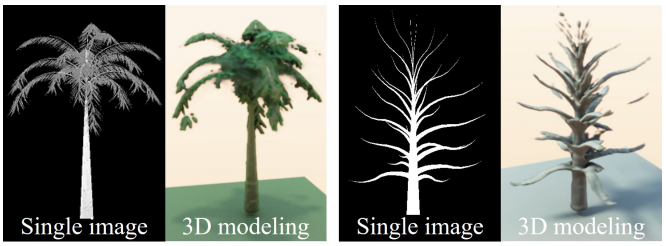

図 2: 1 枚の画像からモデリングした樹木のボリューム。

2 関連研究

画像に基づくボリュームモデリング　Gregson らは複数台のビデオカメラで撮影した映像を用いて流体現象をボリュームアニメーションとしてモデリングできる、stochastic tomography を提案した [2]。この手法の目的の 1 つは必要なカメラの台数を減らすことだが、依然として複数台のカメラが必要であった。岡部らは 1 台のビデオカメラの映像から流体現象をモデリングした [3]。この手法はモデリングできる形状に限界があり、樹木などの構造はモデリングが不可能とされている。近年、単一画像から 3 次元形状をモデリングする手法が盛んに研究されている。これらの中には形状表現にボリュームを用いるものもあるが、基本的には物体の表面形状を扱っていて [4, 5, 6]、樹木のような内部構造のモデリングは扱われていない。

画像に基づく樹木のモデリング　CT 技術を拡張して多視点画像集合から樹木をモデリングする手法がある [7]。スケッチベースモデリングでは、ユーザの描いた 1 枚のスケッチ画像から 3 次元樹木をモデリングできる [8, 9]。また 1 枚の画像から 3 次元樹木をモデリングする手法が存在する

図 3: テクスチャ合成の例。上段が模範となるテクスチャ画像 $\vec{e}$、下段が合成結果 $\vec{i}$。

[10]。多視点画像集合から植物の枝構造を推定する手法が存在する [11]。これらの手法は対象物体を樹木や植物に限定することで見栄えの良い形状のモデリングを可能にしている。一方、提案手法は技術的には対象物を限定せず、あらゆる物体のモデリングに適用できる可能性のある手法である。

3 提案手法

入力画像の集合を $\{\vec{e_\theta} : 1 \leq \theta \leq N^e\}$ とする。N^e は入力画像の枚数とする。入力画像の幅を w、高さを h とすると、$\vec{e_\theta}$ は $w \times h$ 次元のベクトルである。モデリングしたいボリュームを $\vec{v}$ とする。$\vec{v}$ は $w \times h \times w$ 次元のベクトルである。エネルギー関数 E を次のように定義する:

$$E_i(\vec{v}) = |F(B_i\vec{v}) - F(\vec{e_\phi})|^2, \quad (1)$$

$$E = \sum_i E_i(\vec{v}) + \lambda|\vec{v}| \quad \text{subject to } 0 \leq v_{xyz}. \quad (2)$$

ただし、i はカメラの番号とし、B_i はレイキャスティング法によるレンダリング操作を意味する行列とする。つまり、i 番目のカメラでボリューム $\vec{v}$ をレンダリングした画像が $B_i\vec{v}$ となる。F は引数に与えた画像の特徴量を抽出する関数である（詳細は 3.1 章）。$\vec{e_\phi}$ は i 番目のカメラに一番近いカメラで撮影された入力画像とする。式 1 によって、$B_i\vec{v}$ と $\vec{e_\phi}$ から抽出される画像特徴量が同様であってほしい、つまり、両者は同様の見た目の画像であってほしい、という意図を表現している。v_{xyz} は $\vec{v}$ の各要素である。

エネルギー関数 E を最小化するようなボリューム $\vec{v}$ を求めたいが、この最小化問題を一度に解くことは難しい。そこで、i 番目のカメラ毎に勾配 $\nabla E_i(\vec{v})$ を求め、その方向に $\vec{v}$ を少しずつ更新するアプローチを採る。更新後のボリュームを $\vec{v'}$ とし、また、更新の重みを α 及び β とし、以下の更新式を用いる：

$$\vec{v'} = \vec{v} + \alpha\nabla E_i(\vec{v}) - \beta\vec{u} \quad (3)$$

カメラを変えつつ、この更新式を繰り返し計算する。勾配 $\nabla E_i(\vec{v})$ は逆誤差伝搬法によって求めることができる。$-\vec{u}$ は式 2 の L^1 正則化項の勾配であるが、$\vec{v}$ の各要素は 0 以上の値なので、$\vec{u}$ は $w \times h \times w$ 次元の定数ベクトルとなる。また、式 2 の $0 \leq v_{xyz}$ を満たすため、毎回の更新後に $\vec{v'}$ の要素で負の値になったものは強制的に 0 に修正する。

また、繰り返し計算を始めるために $\vec{v}$ の初期値が必要である。入力画像の集合 $\{\vec{e_\theta} : 1 \leq \theta \leq \Theta\}$ から最小二乗法 [1] を用いて初期ボリュームをモデリングする。

3.1 テクスチャ合成

式 1 はテクスチャ合成のためのエネルギー関数である。テクスチャ合成とは、模範となるテクスチャ画像 $\vec{e}$ を入力とし、それと同様の見た目を持つ画像 $\vec{i}$ を合成することである。式 1 のエネルギー関数を $\vec{e}$ と $\vec{i}$ を用いて、

$$E^{texture}(\vec{i}) = |F(\vec{i}) - F(\vec{e})|^2 \quad (4)$$

と書き直す。

N^f 個のフィルタ集合 $\{\vec{f_p} : 1 \leq p \leq N^f\}$ と画像 $\vec{t}$ があるとき、$F(\vec{t})$ は大きさが $N^f \times N^f$ のグラム行列 C を与える。C の各要素 c_{pq} は画像 $\vec{t}$ のフィルタ応答の相互相関として与えられる。

$$r_{\vec{p}} = ReLU(\vec{t} \circledast \vec{f_p}), \quad (5)$$

$$c_{pq} = r_{\vec{p}} \cdot r_{\vec{q}}, \quad (6)$$

ただし、$\circledast$ は畳み込み演算、$\cdot$ は内積を表す。

グラム行列は既存のテクスチャ合成手法でも見た目を制約する特徴量として用いられているが、提案手法が異なるのはフィルタ集合 $\{\vec{f_p} : 1 \leq p \leq N^f\}$ の作り方である。

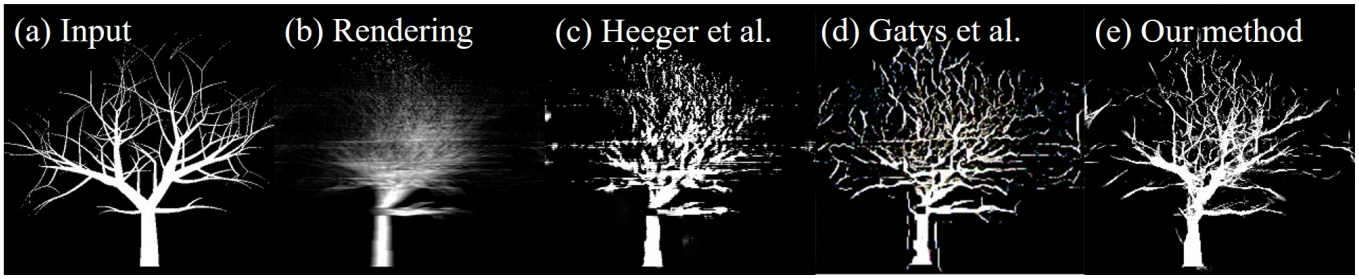

図 4: 提案するテクスチャ合成手法を樹木のモデリングに適用した際の結果。

Portilla らは手作りのフィルタ集合を用いたが、フィルタの数が少ないために、表現できるテクスチャの見た目に限界があった [12]。Gatys らは学習済み VGG モデルを用いて多くのフィルタ応答を得ることで、多種多様のテクスチャの合成に成功した [13] が、合成されるテクスチャの見た目は学習済みモデルに左右され、また、VGG などの大規模なモデルは、自ら学習し直すには多大なコストを要する。

そこで我々は模範となるテクスチャ画像 $\vec{e}$ からフィルタ集合 $\{\vec{f_p} : 1 \leq p \leq N^f\}$ を、オートエンコーダを用いてその場で学習する。$w^f \times w^f$ ピクセルのフィルタを N^f 個作成したい場合には、$\vec{e}$ から $w^f \times w^f$ ピクセルのパッチを抽出し、それらをデータセットとして、中間層に N^f 個のユニットを持たせたオートエンコーダを学習する。学習後、オートエンコーダの重みが $\vec{e}$ の局所的な特徴を捉えたフィルタとして利用できる。

図 3 にテクスチャ合成の例を示す。上段が模範となるテクスチャ画像 $\vec{e}$、下段が合成結果 $\vec{i}$ である。$\vec{i}$ はランダムなノイズを初期画像とし、逆誤差伝搬法による勾配 $\nabla E^{texture}(\vec{i})$ の計算とそれによる $\vec{i}$ の更新を 200 回繰り返した。また、処理の最後にヒストグラムマッチングを適用し、$\vec{i}$ の輝度分布が $\vec{e}$ と同じになるようにした。オートエンコーダによるフィルタ集合の学習に 42 秒、合成の繰り返し計算に 33 秒、合計時間は 75 秒を要した。

このテクスチャ合成手法をボリュームモデリングに適用した際の結果を図 4 に示す。図 4-a が入力画像である。図 4-b は、最小二乗法でモデリングした初期ボリューム $\vec{v}$ を、45° の方向からのカメラでレンダリングした画像である。この画像を初期画像として、3 種類のテクスチャ合成手法を実験した。岡部らの手法 [3] で用いられていた Heeger らの steerable pyramid とヒストグラムマッチングに基づく手法 [14] を用いて、図 4-b の見た目が図 4-a と同様になるように合成した画像が図 4-c である。計算時間は 0.5 秒と短いが、枝が繋がっておらず、樹木の構造が復元されていない。次に Gatys らの手法 [13] で合成した画像が図 4-d であり、最後に提案手法で合成した画像が図 4-e である。共に枝が繋がっており、樹木の構造が復元できているが、Gatys らの手法は 40 回の反復計算に掛かった時間が 34 秒と長かったのに対し、提案手法は同じ 40 回の反復計算に 5.3 秒を要した。また、提案手法の方が樹木の幹から先端に向かうに従って枝が細くなるなどの特徴が上手く表現できている。入力画像からフィルタ集合を直接学習することで、樹木の特徴をより正確に捉えることができているのではないかと推察する。

3.2 勾配 $\nabla E_i(V)$ の計算ついて

上記で、「勾配 $\nabla E_i(\vec{v})$ は逆誤差伝搬法によって求めることができる」と書いたが、提案手法の実装においては少し違うアプローチをとっている。現在のボリューム $\vec{v}$ のレンダリング画像 $\vec{i} = B_i\vec{v}$ が与えられたら、まず、この画像を初期画像として 3.1 章のテクスチャ合成を適用し、修正された画像 $\vec{i'}$ を得る。そして、その差分画像 $\vec{d} = \vec{i'} - \vec{i}$ を計算し、これをカメラの方向にスウィープしたようなボリュームを計算することで勾配 $\nabla E_i(\vec{v})$ を得ている。

3.3 多重解像度

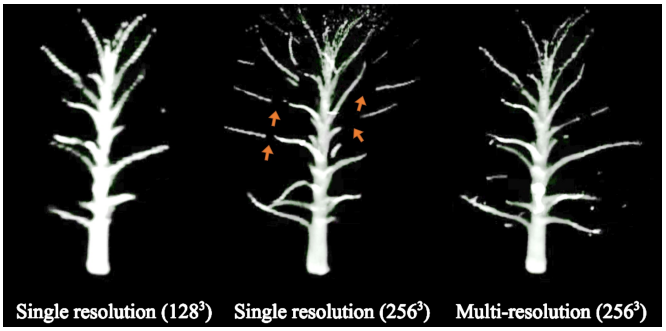

図 5: 左：単一解像度でモデリングした 128^3 ボクセルのモデル、中央：単一解像度でモデリングした 256^3 ボクセルのモデル、右：128^3 ボクセルのモデル（左）を初期ボリュームとしてモデリングした 256^3 ボクセルのモデル。

低い解像度のボリュームのモデリングから始め、徐々に高い解像度のボリュームをモデリングするような多重解像度のアプローチを採ることで、計算コストを抑えつつ、質の高いモデルが得られる。

図 5 に例を示す。元の入力画像は 256^2 ピクセルだが、これを 128^2 ピクセルに縮小した画像を入力とし、上記の手法でモデリングした結果が図 5-左である。樹木の大局的な形状がモデリングできているが解像度が低い。一方で、元の 256^2 ピクセルの画像を入力とし、モデリングした結果が図 5-中央である。樹木の細い枝が表現できていて解像度は高いが、それらはしばしば分断されてしまっている（図中のオレンジ色の矢印で示した箇所）。

多重解像度のアプローチでは、縮小した 128^2 ピクセルの画像でモデリングした 128^3 ボクセルのボリューム（図 5-左）を 256^3 ボクセルにアップサンプリングする。これを初期ボリュームとして用い、元の 256^2 ピクセルの画像を用いてモデリングした 256^3 ボクセルのボリュームを図 5-右に示す。枝の分断などが抑えられつつ、高い解像度が実現できている。

4　結果と議論

実験に用いた入力画像の解像度は全て 256^2 ピクセルで、モデリングしたボリュームの解像度は全て 256^3 ボクセルである。また、全ての結果を 3.3 章で述べたように 2 段階の多重解像度のアプローチでモデリングしている。また、今回用意した入力画像は全て Weber らの手法 [15] で作成した樹木の CG モデルをレンダリングすることによって入手した。実際の樹木の写真からのボリュームモデリングは現在は未だ取り組めていないが、近い将来実施したい。

1 枚の画像からモデリングした結果を図 2 に示す。入力画像が 1 枚しかない場合は、樹木を正面 (方位角が 0°) と真横 (方位角が 90°) から撮影した 2 枚の画像が同一であるものとしてモデリングを行う。そのため完成するボリュームは対称性を帯びることになる。図 2-右の結果については、128^3 ボクセルのボリュームをモデリングするにあたって、入力画像から 7×7 ピクセルのフィルタ 128 枚を学習してテクスチャ合成した。また、256^3 ボクセルのボリュームをモデリングするにあたって、入力画像から 13×13 ピクセルのフィルタ 128 枚、3×3 ピクセルのフィルタ 32 枚を学習してテクスチャ合成した。モデリングにはトータルで 396 秒を要した。

樹木を正面 (方位角が 0°) と真横 (方位角が 90°) から撮影した 2 枚の入力画像からモデリングした樹木のボリュームを図 6 に示す。

ボリュームのレンダリングには Kroes らの *Exposure Render* を用いた [16]。この手法はボリュームレンダリング方程式の積分を確率的に効率よく行うことで、インタラクティブに伝達関数や環境照明を変更することが可能になっている。図 6-下段では樹木の枝が茶色に、葉が緑色になるようにレンダリングしているが、これは枝のボクセル値と葉のボクセル値の違いを利用し、色を割り当てるような伝達関数を用いてレンダリングしている。

提案手法にはいくつか改良の余地がある。1 つは、現在、Portilla らと同様に相互相関を制約にしたテクスチャ合成手法を用いているが、低レベルな特徴量しか用いていないために良い画像が合成できない場合がある。例えば、図 6-下段-中央のヤシの木のような例では、木の幹の部分に葉のギザギザな特徴が合成されてしまうことがある。これはテクスチャ合成手法が木の幹と葉の区別をしていないために起こることである。画像中に現れるパーツを分類して理解できるような高レベルな特徴量が扱えるような画像合成手法を開発することが、より高精度なボリュームをモデリングするために必須であることが伺える。

モデリングに掛かる計算時間が長いことも課題である。現在のボトルネックはテクスチャ合成の処理に時間が掛かっていることと同時に、最適化が収束するまでの繰り返しの多さも問題である。手法を見直して高速化を図り、スケッチベースモデリングなど、インタラクティブなモデリングのためのユーザインタフェースを開発したい。

今回はボリュームモデリングについて技術開発を行ったが、同様の考え方はサーフェスモデリングにも適用できると考えている。Structure from motion などの手法と、提案手法の考え方を組み合わせるような実験も行っていきたい。

参考文献

[1] I. Ihrke and M. Magnor, “Image-based tomographic reconstruction of flames,” in *Proc. of SCA ’04*, 2004, pp. 365–373.

[2] J. Gregson, I. Ihrke, N. Thuerey, and W. Heidrich, “From capture to simulation: Connecting forward and inverse problems in fluids,” *ACM Trans. Graph.*, vol. 33, no. 4, pp. 139:1–139:11, 2014.

[3] M. Okabe, Y. Dobashi, K. Anjyo, and R. Onai, “Fluid volume modeling from sparse multi-view images by appearance transfer,” *ACM Transactions on Graphics (Proc. SIGGRAPH 2015)*, vol. 34, no. 4, pp. 93:1–93:10, 2015.

[4] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee, “Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision,” in *Advances in Neural Information Processing Systems 29*.

[5] H. Kato, Y. Ushiku, and T. Harada, “Neural 3d mesh renderer,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] S. Lunz, Y. Li, A. Fitzgibbon, and N. Kushman, “Inverse graphics gan: Learning to generate 3d shapes from unstructured 2d data,” 2020.

[7] A. Reche-Martinez, I. Martin, and G. Drettakis, “Volumetric reconstruction and interactive rendering of trees from photographs,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 720–727, 2004.

[8] M. Okabe, S. Owada, and T. Igarashi, “Interactive design of botanical trees using freehand sketches and example-based editing,” *Computer Graphics Forum (proceedings of Eurographics 2005)*, vol. 24, no. 3, pp. 487–496, 2005.

図 6: 樹木を正面 (方位角が 0°) と真横 (方位角が 90°) から撮影した 2 枚の入力画像からモデリングした樹木のボリューム。

[9] X. Chen, B. Neubert, Y.-Q. Xu, O. Deussen, and S. B. Kang, "Sketch-based tree modeling using markov random field," in *ACM SIGGRAPH Asia 2008 Papers*, 2008.

[10] P. Tan, T. Fang, J. Xiao, P. Zhao, and L. Quan, "Single image tree modeling," *ACM Trans. Graph.*, vol. 27, no. 5, 2008.

[11] T. Isokane, F. Okura, A. Ide, Y. Matsushita, and Y. Yagi, "Probabilistic plant modeling via multi-view image-to-image translation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, 2018.

[12] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 49–70, 2000.

[13] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in *Advances in Neural Information Processing Systems 28*.

[14] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," in *Proc. of SIGGRAPH '95*, 1995, pp. 229–238.

[15] J. Weber and J. Penn, "Creation and rendering of realistic trees," in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, 1995, p. 119–128.

[16] T. Kroes, M. Eisemann, and E. Eisemann, "Visibility sweeps for joint-hierarchical importance sampling of direct lighting for stochastic volume rendering," in *Proceedings of Graphics Interface 2015*.