令和 6 年度修士論文

3D-SRCGAN を用いた
低解像度三次元ボクセルモデルから
サーフェイスモデルへの変換

指導教員　椋木 雅之

宮崎大学大学院　工学研究科　工学専攻

機械・情報系コース　情報システム工学分野

T2303736

ADAM MANATO MAEDA BIN AMIZAN

概要

本論文では、3D-Super Resolution Conditional Generative Adversarial Networks（3D-SRCGAN）を用いた低解像度三次元ボクセルモデルからサーフェイスモデルへの変換手法を提案する。

三次元ボクセルモデルからサーフェイスモデルを生成する手法として、Marching Cubes 法や DMTet 法がある。Marching Cubes 法は三次元の離散スカラーフィールドから等値面のポリゴンメッシュを抽出するためのアルゴリズムである。一方、DMTet 法は、四面体グリッドから等値面を抽出する Marching Tetrahedra アルゴリズムを深層学習に統合した手法である。

しかし、Marching Cubes 法には、生成されるサーフェイスが滑らかではないという問題がある。一方、DMTet 法に関しては、公開されているサンプルコードが論文の完全な実装ではなく、実行時に適切なサーフェイスが生成されない場合があった。

そこで、本論文では 3D-SRCGAN を用いて低解像度の三次元ボクセルモデルを超解像し、Marching Cubes 法を適用する手法を提案する。本手法により、より細かなサイズのボクセルに対してサーフェイスを生成することが可能となり、ボクセルサイズによる凹凸の影響が低減できる。さらに、生成したサーフェイスの頂点に対して Laplacian Smoothing と呼ばれる平滑化処理を適用することで、滑らかなサーフェイスの生成を実現する。本研究では、この一連の処理により、低解像度三次元ボクセルモデルから高品質なサーフェイスモデルの生成を目指す。

実験では、提案手法の有効性を ModelNet10 データセットを用いて検証した。その結果、提案手法が従来手法と比較して、低解像度の三次元ボクセルモデルからサーフェイスを生成する際に優れた結果を示すことが確認された。

目次

1. はじめに

近年、三次元モデルは建築や機械設計、エンターテインメント、医療等、幅広い分野で利用されている。二次元画像に比べて三次元モデルを利用すると、簡単に物体の形状を把握し、複雑な形状を設計することができる。

三次元モデルの表現方法には、ボクセル表現やサーフェイス表現などがある。ボクセル表現は、ボクセルと呼ばれる立方体を積み重ねて三次元モデルを表現する。サーフェイス表現は、物体の表面を三角形などのポリゴンの集合で表現する。ボクセル表現では、積み木のようにボクセルを積み合わせて三次元モデルを比較的容易に作成できる。しかし、ボクセル表現は必要なメモリ量が多く、表示や変形などの処理時間が長くなるという問題がある。そのため、三次元モデルを利用する際には、サーフェイス表現が使われることが多い。

そこで、三次元ボクセルモデルからサーフェイスモデルを生成する手法が提案されている。例えば、Marching Cubes 法[1]は、ボクセルモデルからサーフェイスモデルを生成するための代表的な手法である。三次元の離散スカラーフィールド（その要素はボクセルと呼ばれる）から等値面のポリゴンメッシュを抽出するためのアルゴリズムである。一方、Deep Marching Tetrahedra 法[2]（以下、DMTet 法呼ぶ）は、四面体グリッドから等値面を抽出する Marching Tetrahedra[3]アルゴリズムを深層学習に統合した手法である。ニューラルネットワークを用いて、低解像度の三次元ボクセルモデルから高精細なサーフェイスモデルを生成する。

しかし、Marching Cubes 法は、サーフェイスの生成が滑らかではないという問題がある。一方、DMTet 法は、論文で示された実験結果において、粗い三次元ボクセルモデルからサーフェイスモデルを良好に生成できている。しかし、公開されているサンプルコードは論文を完全に実装したものではなく、実行したところ適切な面が生成されない場合があった。さらに、論文の記述に近づくよう提供されたコードを改良したものの、得られた結果は十分な精度を満たさなかった。

本研究では、3D-Super Resolution Conditional Generative Adversarial Networks（3D-SRCGAN）[4]を用いた低解像度三次元ボクセルモデルからサーフェイスモデルへの変換手法を提案する。3D-SRCGAN は、低解像度の三次元ボクセルモデルから、より精細な三次元ボ

クセルモデルを生成する超解像手法である。超解像とは、低解像度のデータから高解像度のデータを生成・補完する技術である。解像度とは、画素やボクセルなどが一定区間内にどれだけ密に存在しているかを示す指標である。以下では、超解像により得られる精細な三次元ボクセルモデルを超解像三次元ボクセルモデルと呼ぶ。Marching Cubes 法によるサーフェイス生成は、ボクセルサイズの影響を受けるため、低解像度の三次元ボクセルモデルに直接適用すると、凹凸の目立つサーフェイスモデルとなる。3D-SRCGAN により超解像三次元ボクセルモデルに変換した後、Marching Cubes 法を適用することで、より細かなサイズのボクセルに対してサーフェイスを生成できるため、ボクセルのサイズに起因する凹凸の影響を低減できる。さらに、生成したサーフェイスの頂点に対して Laplacian Smoothing[5]と呼ばれる平滑化処理を適用することで、滑らかなサーフェイスを生成する。この一連の処理により、低解像度三次元ボクセルモデルから、滑らかなサーフェイスモデルを生成する。

以下、２章では、ボクセルモデルからサーフェイスモデルへの変換について述べる。３章では、提案手法である超解像処理によるサーフェイス生成について述べる。４章では、超解像処理によるサーフェイス生成の有効性について、実験により評価する。最後に５章では、結論と今後の課題を述べる。

2. ボクセルモデルからサーフェイスモデルへの変換

2.1. 三次元形状モデルの表現方法

三次元形状モデルの表現方法には、ボクセル表現やサーフェイス表現などがある。

ボクセル表現は、空間を均一な立方体グリッド（ボクセル）に分割し、物体の内外を 0/1 の数値情報で形を表す。ボクセルを積み重ねることで、比較的容易に三次元モデルを作成できる。その一方で、ボクセル表現は通常、必要なメモリ量が多く、表示や変形などの処理時間が長くなるという問題がある。

サーフェイス表現は、物体の表面や外層をポリゴンなどで面的に表現する方法である。内部の構造情報を持たず、主に外形や見た目の表現に特化している。サーフェイスは、多くの場合、ポリゴンメッシュ（三角形や四角形で構成されたポリゴンの集合）として表現される。主に、ゲーム開発や映画の CG 制作、建築デザインや製品デザインに使われることが多い。メリットとしては、三次元モデルの外部形状のみを扱うため、データ量を抑えやすい。また、高精細でリアルな外観を再現できる。一方デメリットとしては、三次元モデルを作成するときに、サーフェイス表現を用いるには、CAD（コンピュータ支援設計）などの専用ツールの操作や効率的なワークフローの習得が必要となる。そのため、初心者にとっては、操作習得に時間を要する場合がある。

2.2. Marching Cubes 法によるサーフェイス生成

ボクセルからサーフェイスを生成する手法として、Marching Cubes[1]法がある。Marching Cubes 法は、コンピュータ断層撮影（CT）や核磁気共鳴画像法（MRI）などで得られる三次元ボリュームデータから、等値面のサーフェイス表現を生成する手法である。

ボリュームデータは、三次元空間を賽の目状に分割したボクセルに、その地点での密度や温度、圧力などの値をボクセル値として保持することで、三次元的なデータ分布を表現したデータである。Marching Cubes 法、まず、2×2×2 の隣接するボクセルを頂点とする立方体を考える。次に、この立方体の各頂点のボクセル値が等値面の値以上であれば 1、以下であれば 0 に 2 値化する。これにより、立方体の 8 つの頂点は2^8=256 通りのパターンを取るが、回転や 0、1 の反転を考慮すると、図 1 の 15 種類の基本立体図形のいずれかとなる。Marching Cubes 法では、立方体の 8 つの頂点の 256 通りのパターンが基本立体図形のいずれであるか決定する。さらに、各辺上で、ボクセル値を内分することで、面の位置を決定し等値面を作成する。このため、生成される面はボクセルのサイズに依存したものとなる。

ボクセル表現の三次元モデルは、ボリュームデータの一種で、ボクセル値が 0、1 の 2 値であるものとみなせる。等値面の値を 0.5 として Marching Cubes 法を適用することで、物体形状の表面に面をはり、サーフェイス表現を得ることができる。

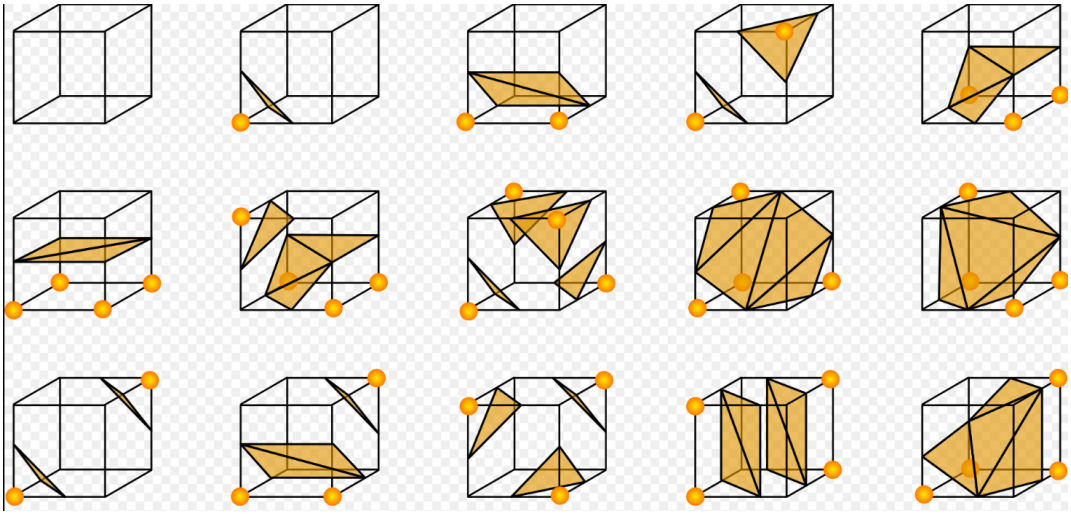


図 1：15 種類の基本立体図形

Marching Cubes 法によるサーフェイス生成はボクセルサイズに影響を受ける。ボクセルサイズが大きく（解像度が低く）なるほど、生成されるサーフェイスには凹凸が目立ちやすくなる。図 2 と図 3 に、解像度 16×16×16（16^3）と 64×64×64（64^3）の三次元ボクセルモデルに対する Marching Cubes 法によるサーフェイス生成の例を示す。解像度が低いと生成されるサーフェイスが滑らかではないことがわかる。

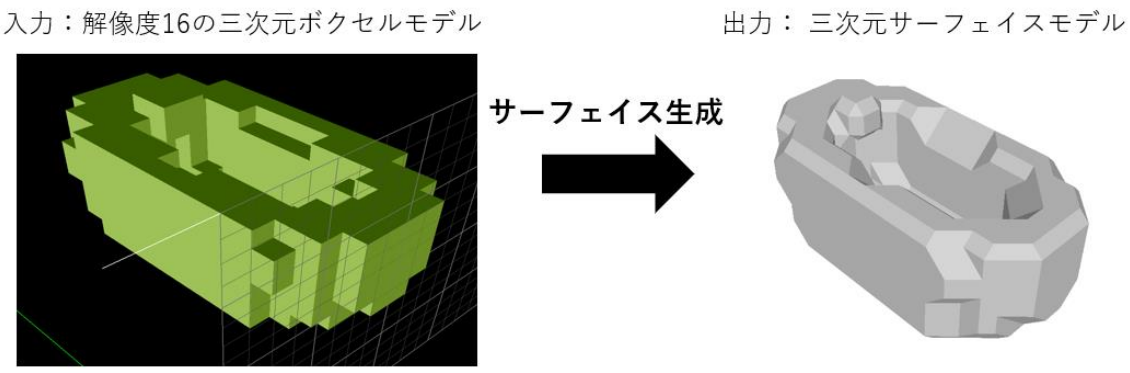


図 2：解像度16^3の三次元ボクセルモデルのときのサーフェイス生成

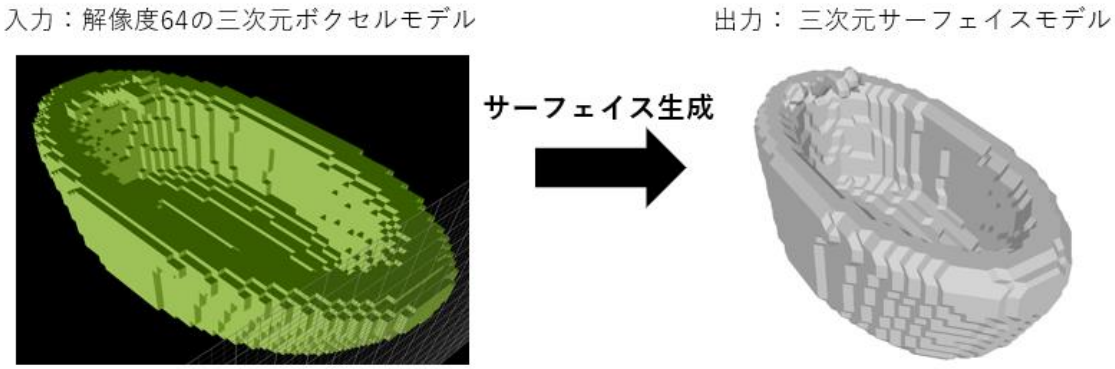


図 3：解像度64^3の三次元ボクセルモデルのときのサーフェイス生成

2.3. Deep Marching Tetrahedra 法によるサーフェイス生成

Deep Marching Tetrahedra（DMTet）法[2]は、低解像度の三次元表現（粗いボクセルまたは点群）から精細で高解像度の三次元サーフェイスモデルを生成する深層学習手法である（図 4）。この手法では、三次元の表現として Signed Distance Function（SDF）を利用する。SDF は、任意の空間内の点から曲面（形状表面）までの最短距離を表す関数で、曲面の外側では正、内側では負の値を取る。SDF の値が 0 となる点の集合が三次元空間におけるサーフェイスを表す。DMTet 法では、深層学習の一種である敵対的生成ネットワーク（GAN）を使って、低解像度の三次元表現から SDF を出力する生成器を学習する。生成器から得られた SDF を基に、Marching Tetrahedra 法により最終的な三次元サーフェイスモデルを生成する。ここで、Marching Tetrahedra 法は、Marching Cubes 法を立方体単位ではなく四面体単位に適用する手法である。これにより、高解像度かつ滑らかなサーフェイスモデルを生成することができる。

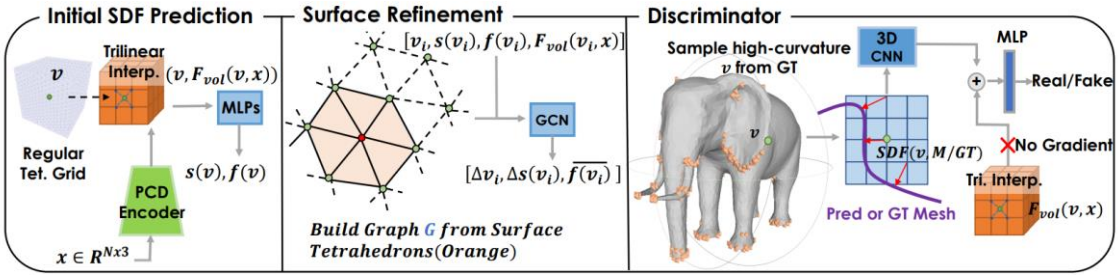


図 4：生成器と識別器の構造[2]

DMTet 法は、深層学習により生成器を学習する際、以下の 5 つの損失関数を組み合わせて最小化する。

1. 表面整合損失（Surface Alignment Loss）

表面整合損失は、正解のサーフェイスモデル（M_{gt}）と生成されたサーフェイスモデル（M_{pred}）の形状の一致度を評価するために用いられる損失関数である。具体的には、各サーフェイスモデルから点群を抽出し、その幾何学的整合性を測定する。

まず、正解のサーフェイスモデルM_{gt}から点群P_{gt}を抽出する。同様に生成されたサーフェイスモデルM_{pred}から点群P_{pred}を抽出する。次に、両者の間の Chamfer Distance および Normal Consistency を最小化することにより、形状の一致を促進する。

Chamfer Distance は、生成されたサーフェイスモデルの点群P_{pred}の各点から、正解のサーフェイスモデルの点群P_{gt}の最も近い点までの距離を測定し、その和を損失として定義する。これは、2 つの点群間の幾何学的な類似性を評価する指標として広く用いられる。Chamfer DistanceL_{cd}の損失関数は以下の式で表される：

$$L_{\text{cd}} = \sum_{p \in P_{pred}} \min_{q \in P_{gt}} ||p - q||_2 + \sum_{q \in P_{gt}} \min_{p \in P_{pred}} ||q - p||_2$$

この損失は、両点群の対応関係が近くなるように学習を促し、サーフェイス全体の形状が適切に再現されることを保証する。

Normal Consistency は、生成されたサーフェイスモデルの法線ベクトルと、正解のサーフェイスモデルの法線ベクトルの整合性を評価する指標である。具体的には、各点における法線ベクトル$\vec{n}_p$と$\vec{n}_{\hat{q}}$の内積を用いて、方向の一致度を測定する。Normal Consistency の損失関数L_{normal}は以下の式で表される：

$$L_{\text{normal}} = \sum_{p \in P_{pred}} (1 - |\vec{\mathbf{n}}_p \cdot \vec{\mathbf{n}}_{\hat{q}}|),$$

ここで、$\hat{q}$はpに対する最も近い点であり、$\vec{n}_p$, $\vec{n}_{\hat{q}}$は点p, $\hat{q}$における法線方向を表す。また、$|\vec{n}_p \cdot \vec{n}_{\hat{q}}|$は、2 つの法線ベクトルの内積を表し、値が 1 に近いほど整合性が高いことを示す。Normal Consistency の最小化により、サーフェイスの滑らかさやリアルな形状の再現性が向上する。

2. 敵対的損失（Adversarial Loss）

敵対的損失（Adversarial Loss）は、生成された SDF（Signed Distance Function）が、正解の高解像度 SDF と区別がつかないように学習を行うための損失関数である。敵対的損失には、LSGAN[6] の手法を適用する。LSGAN は、従来の GAN におけるバイナリクロスエントロピー損失の代わりに、二乗誤差（最小二乗損失）を使用することで、学習の安定性向上と勾配消失問題の軽減を図る手法である。このアプローチにより、識別器の過学習を防ぎ、生成器がより現実的な SDF を出力することが期待される。

識別器の目的は、正解の高解像度 SDF（真のデータ）を正しく識別し、生成器が生成した SDF（偽のデータ）を区別することである。識別器の損失関数L_Dは以下の式で表される：

$$L_{\mathrm{D}} = \frac{1}{2}[(D(M_{gt}) - 1)^2 + D(M_{pred})^2]$$

ここで、M_{gt}は正解の高解像度 SDF サンプル、M_{pred}は生成器によって生成された SDF サンプル、$D(M_{gt})$は識別器が正解のデータを本物と判断する確率、$D(M_{pred})$ は識別器が生成データを偽物と判断する確率を表す。この損失を最小化することで、識別器は正解の高解像度 SDF と生成された SDF の違いをより正確に識別できるようになる。

生成器の目的は、識別器を欺き、生成された SDF を正解の SDF と区別できないようにすることである。生成器の損失関数L_Gは以下の式で表される：

$$L_{\mathrm{G}} = \frac{1}{2}[(D(M_{pred}) - 1)^2].$$

ここで、生成器は識別器を騙すために、$D(M_{pred})$を 1 に近づけるように学習を進める。この損失関数を最小化することで、生成器はより高品質な SDF を出力できるようになり、識別器がそれを本物と誤認するようになる。

3. SDF 整合損失（SDF Consistency Loss）

SDF 整合損失は、生成されたサーフェイスモデルの SDF 値が、正解の高解像度サーフェイスモデルの SDF 値と整合するように制約を課すための損失関数である。この損失を最小化することで、形状表現の正確性を向上させ、生成モデルが正解のサーフェイス形状に忠実であることを保証する。

SDF とは、三次元空間内の各点からサーフェイスまでの最短距離を示す関数であり、形状の外側では正の値、内側では負の値を持つ。SDF 整合性を保つことで、生成された三次元形状が、正解の形状と一致するように制御できる。

SDF 整合損失の損失関数L_{SDF}は以下の式で表される：

$$L_{\mathrm{SDF}} = \sum_{v_i \in V_T} |s(v_i) - SDF(v_i, M_{gt})|^2,$$

ここで、$SDF(v_i, M_{gt})$は学習した生成器が出力する点v_iにおける正解のサーフェイスM_{gt}に対する SDF 値、$s(v_i)$は点v_iにおける真の SDF 値を表す。この損失関数は、形状の細部までの再現精度を向上させることを目的としている。

4. 頂点変形損失（Deformation Loss）

頂点変形損失（Deformation Loss）は、生成の際にサーフェイスの頂点位置が不必要に大きく変形しないように制御するための損失関数である。この損失を導入することで、正解のサーフェイスの構造を維持しながら、局所的な歪みやノイズを抑え、より安定したサーフェイス生成が可能となる。

頂点変形損失の損失関数L_{def}は以下の式で表される：

$$L_{\text{def}} = \sum_{v_i \in V_T} ||\Delta v_i||_2$$

ここで、Δv_iは生成されたサーフェイスの頂点位置と正解のサーフェイスの頂点位置の間の誤差である。この損失関数を最小化することで、頂点の位置が正解のサーフェイスと一致し、無駄な変形が抑制される。

5. 最終損失関数の統合

最終的な損失関数Lは、以下の 5 つの損失関数の重み付き和として定義される。

$$L = \lambda_{\text{cd}} L_{\text{cd}} + \lambda_{\text{normal}} L_{\text{normal}} + \lambda_{\text{G}} L_{\text{G}} + \lambda_{\text{SDF}} L_{\text{SDF}} + \lambda_{\text{def}} L_{\text{def}},$$

ここで、λ_{cd}, λ_{normal}, λ_{G}, λ_{SDF}, λ_{def}は各損失関数の重みを調整するハイパーパラメータである。

2.4. 従来手法の問題

Marching Cubes 法は、15 種類の基本立体図形を利用してボクセルデータをサーフェイスに変換するため、計算が効率的に行われる。また、立方体単位で処理を行うため、大規模なデータセットにも対応可能になる。しかし、Marching Cubes 法は入力データのボクセル解像度に強く依存しており、解像度が低い場合は、滑らかな形状を表現できない。また、小さな特徴や複雑な形状がメッシュに反映されにくいという問題がある。

DMTet 法は、深層学習モデルにより形状データを補間する。そのため、従来手法よりも滑らかで正確な形状の表現ができ、ボクセルの解像度に制約されず、形状の複雑さに応じて高品質なサーフェイスモデルが生成できる。しかし、DMTet 法のサンプルコードが公開[7]されているものの、完全な実装にはなっていない。論文に記載されていた 5 つの損失関数に対して、サンプルコードには Chamfer Distance の損失関数L_{cd}と生成器の損失関数L_Gの 2 つしか実装されていなかった。そこで、不足していた SDF 損失関数L_{SDF}と識別器に対する敵対的損失関数L_Dを追加で導入した。しかし、生成されたサーフェイスの精度向上には大きく寄与しなかった。図 5 に追加実装した DMTet 法でのサーフェイス生成の例を示す。生成されたサーフェイスに多数の穴が存在し、サーフェイスが滑らかではなく、凹凸が目立つという問題がある。

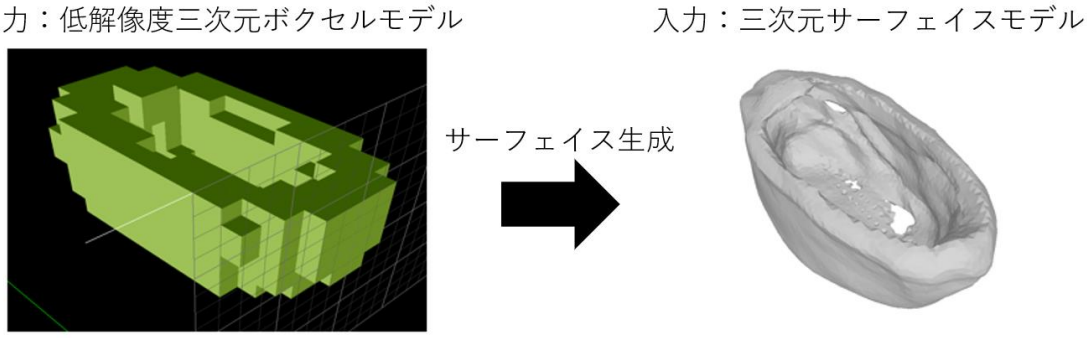


図 5：追加実装した DMTet 法によるサーフェイス生成

3. 超解像処理によるサーフェイス生成

3.1. 超解像処理の導入

超解像とは、低解像度のデータから高解像度のデータを生成・補完する技術である。解像度とは、画素やボクセルなどデータの構成単位が一定の区間内にどれだけ密に存在しているかを表したものである。本研究では、超解像の対象として、ボクセル表現の三次元モデルを扱う。また、超解像によって生成された高解像度データを超解像三次元モデルと呼ぶ。

超解像を導入することによって、2.4.節で述べた従来手法の問題を解決することができる。Marching Cubes 法では、低解像度の三次元モデルを入力とした場合、滑らかな形状を表現できないという課題があった。この問題に対処するため、まず低解像度の三次元モデルに対して超解像処理を適用し、その後、平滑化処理を施すことで、精細かつ滑らかなサーフェイスモデルを生成する。また、DMTet 法については、サンプルコードが不完全な実装で、サーフェイス生成が論文のような高品質なものにならなかった。これに対し、本研究では、3D-SRCGAN と Marching Cube 法を組み合わせることで、より高精度なサーフェイス生成を安定して実現する。3D-SRCGAN を用いることで、学習により最適な高解像度ボクセルデータを生成し、Marching Cubes 法によって効率的にサーフェイスモデルを作成することが可能となる。

3.2. 処理の流れ

本研究での処理の流れは以下の通りである。

１． 3D-SRCGAN による三次元ボクセルモデルの超解像

２． Marching Cubes 法によるサーフェイス生成

３． Laplacian Smoothing によるサーフェイスモデルの平滑化

以下、各処理の詳細について述べる。

3.3. 3D-SRCGAN による三次元ボクセルモデルの超解像

3D-Super Resolution Conditional Generative Adversarial Networks（3D-SRCGAN）[4]は深層学習を用いて三次元ボクセルモデルを超解像する手法である。本手法の基盤となるのは、3D-Super Resolution Generative Adversarial Networks (3D-SRGAN)[8]である。これは二次元画像の超解像に有効な Super Resolution Generative Adversarial Networks （SRGAN）[9]を、ボクセル表現の三次元モデルへと拡張した手法である。SRGAN は Generative Adversarial Network（GAN）を応用しており、生成器 （Generator）と識別器（Discriminator）の 2 つのニューラルネットワークで構成される。Generator は、学習データに近いデータを生成しようとし、Discriminator は入力データが学習データか Generator が生成したデータかを識別しようとする。Generator は Discriminator を騙すことができるように学習し、Discriminator は Generator が生成したデータを見破ることができるように学習する。この学習は、Generator が学習データと同じようなデータを生成できるようになることが最終目標である。この状態になると、Discriminator は学習データと生成データの識別ができなくなり、正答率は 50%になる。以上のような仕組みで GAN は学習データに非常に近い新たなデータを生成することができる。3D-SRGAN は、GAN のこの特徴を利用して超解像を行う。

3D-SRGAN に対して 3D-SRCGAN は、GAN の Discriminator は Generator の入力にクラス識別情報を与えることで、複数クラスが混在するデータセットから任意のクラスを生成できるという特徴がある。これにより、3D-SRGAN による三次元ボクセルモデル超解像において、学習に使えるデータ数が少ない、学習に使っていないクラスの超解像精度が低いという問題が改善された。

3.3.1. 3D-SRCGAN のネットワーク構造[4]

3D-SRCGAN は GAN と同じく Generator と Discriminator の 2 つのネットワークで構成されており、入出力データとしてボクセル表現の三次元モデルを扱う（図 6）。学習データには低解像度三次元モデルと高解像度三次元モデルのペアを使用する。Generator は低解像度三次元モデルから高解像度三次元モデルを生成し、Discriminator は入力された三次元モデルが学習データの高解像度三次元モデルなのか Generator が生成した三次元モデルなのかを識別する。この 2 つは敵対的関係にあり、それぞれの目的関数は、Generator は

Discriminator を騙すように学習データと似た三次元モデルを生成することであり、Discriminator は学習データと生成された三次元モデルを見分けることである。3D-SRCGAN の最終的な目的は、Discriminator が識別できないような三次元モデルを Generator が生成できることである。また、学習の結果得られた Generator が生成モデルとなる。

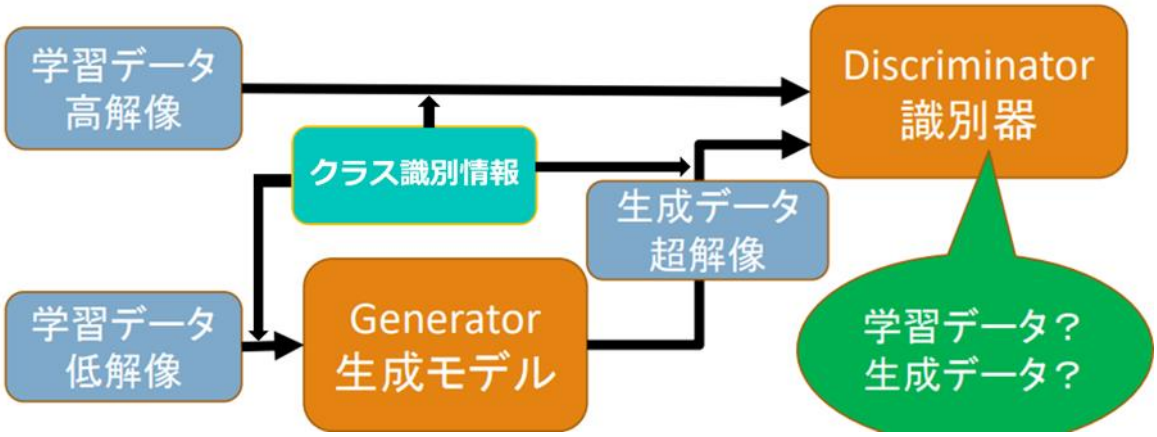
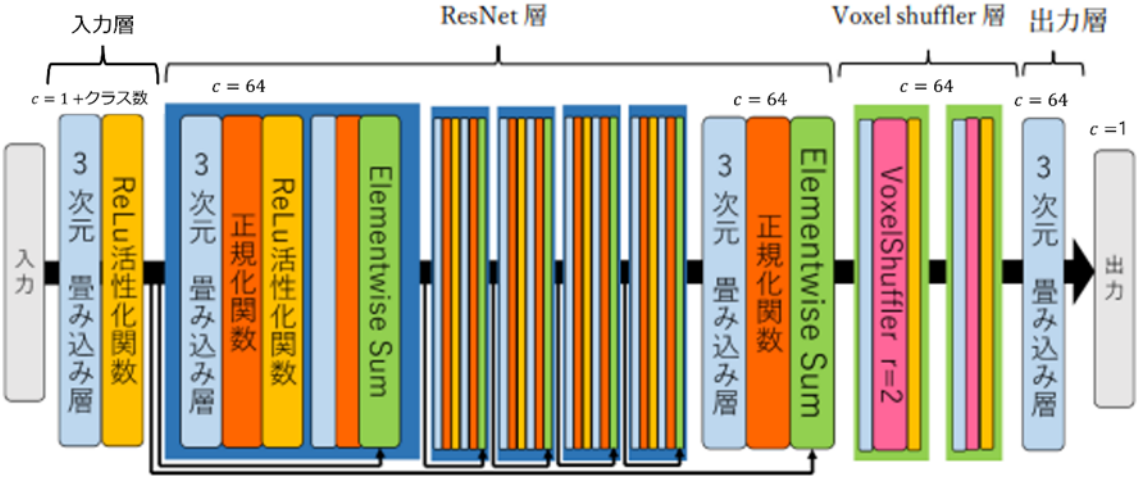


図 6：3D-SRCGAN の概要図



c：入力チャネル数

図 7：Generator の構造

3.3.2. 3D-SRCGAN の Generator の構造[4]

3D-SRCGAN の Generator は図 7 のような構造になっている。以降、入力のチャネル数をC_{in}、出力のチャネル数をC_{out}、ストライドをs、カーネルのサイズをkとする。ここでストライドとは、畳み込み処理を行うときにカーネルをずらしていく間隔のことである。Generator の三次元畳み込み層のストライドは全てs=1 である。

入力には学習データの低解像度三次元モデルが与えられ、出力として高解像度三次元モデルを生成する。

Generator の構造は大きく入力層、ResNet 層、Voxel Shuffler 層、出力層の 4 つに分けられる。入力層では、入力で与えられた三次元モデルを三次元畳み込み層によって、複数のチャネルとして ResNet 層に与えている。ResNet 層では三次元モデルの特徴を抽出し、Voxel shuffler 層で三次元モデルを拡大している。

最後に出力層で入力の特徴マップから三次元モデルを生成し、出力している。

3D-SRCGAN の Generator では、三次元畳み込み層を利用している。三次元の畳み込み層では、カーネルのサイズが$k \times k \times k$の畳み込み処理をする。

三次元畳み込み層の後の処理に、正規化関数や活性化関数がある。正規化関数では三次元畳み込み層の出力を正規化している。この処理をすることで学習速度を速くすることができ、学習の安定化を図ることができる[10]。Generator では活性化関数に ReLu 活性化関数(図 8)を使用しており、以下の式で表現できる。

$$f(x) = \max(0, x)$$

次にそれぞれの層の処理を説明する。まず、入力層では、$C_{in} = 1$+クラス数、$C_{out} = 64$、$k = 9$ の三次元畳み込み層、ReLu 活性化関数を順番に処理する。

ResNet 層は 5 つの Residual Block で構成されている。Residual Block の中身は、$C_{in} = 64$、$C_{out} = 64$、$k = 3$ の三次元畳み込み層、正規化関数、ReLu 活性化関数、$C_{in} = 64$、$C_{out} = 64$、$k = 3$ の三次元畳み込み層、正規化関数の順番で構成されている。それぞれの Residual Block では入力を出力と加算している。即ち各 Residual Block はそれぞれの入力と出力の差を学習することになる。5 つの Residual Block の処理の後、$C_{in} = 64$、$C_{out} = 64$、$k = 3$ の三次元畳み込み層、正規化関数で処理を行い、ResNet 層の入力を加算して出力している。

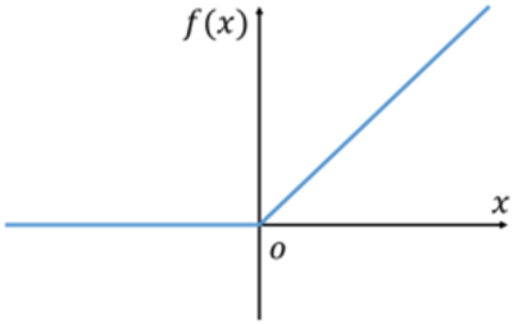


図 8：ReLu 活性化関数

Voxel shuffler 層では、実際に三次元モデルのサイズを拡大していく(図 9)。そのために入力の特徴マップを並び替えて三次元モデルを拡大することができる Voxel Shuffler を利用する。Voxel shuffler 層は 2 つのブロックで構成されており、1 つの Block で入力の特徴マップから 2 倍に拡大している。1 つのブロックの中身は$C_{in} = 64$、$C_{out} = 512$、s = 1、$k = 3$ の三次元畳み込み層、$C_{in} = 512$、$C_{out} = 64$の Voxel Shuffler 層、Relu 活性化関数を順に処理している。この 2 つのブロックを通過した出力は 4 倍に拡大されている。

最後に$C_{in} = 64$、$C_{out} = 1$、s = 1、$k = 9$ の三次元畳み込み層で処理し、高解像度三次元モデルを出力している。

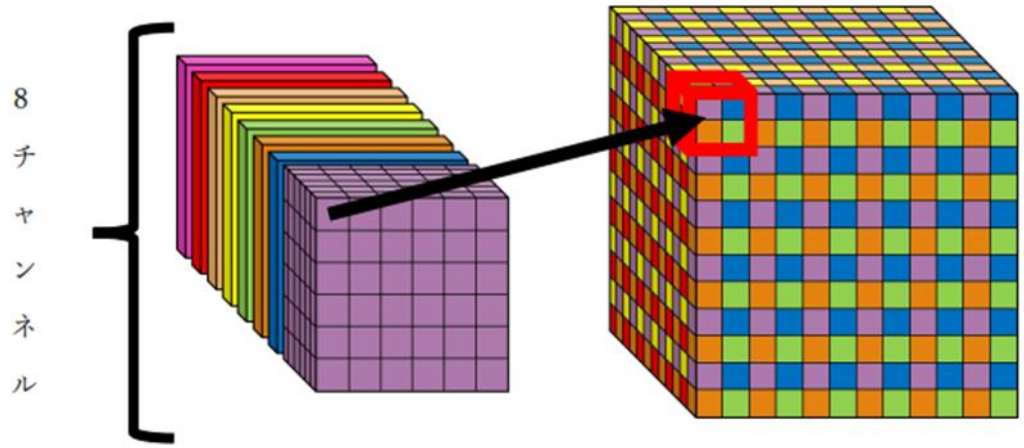


図 9：Voxel Shuffler

3D-SRCGAN により低解像度ボクセルモデルに対して超解像処理を行う。

本研究では、16×16×16（16^3）の低解像三次元モデルの入力から 64×64×64（64^3）の超解像三次元モデルを生成する。図 10 に超解像により得られた三次元モデルの例を示す。図 10 では、超解像によりボクセルのサイズが細かくなっていることがわかる。

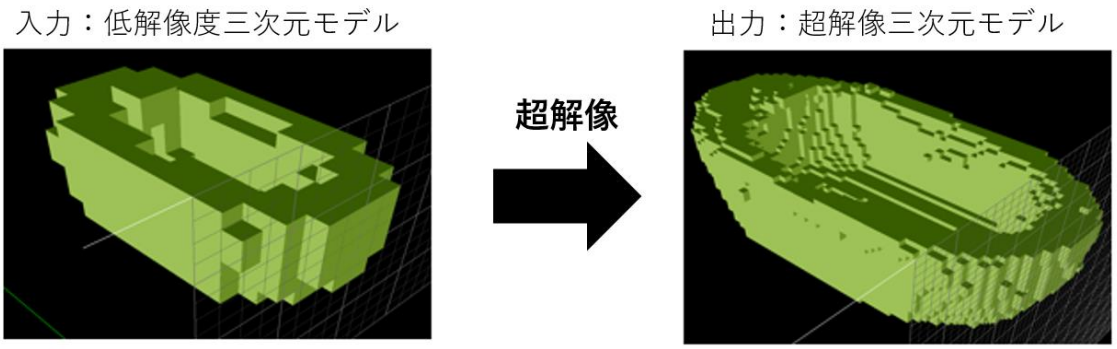


図 10：3D-SRCGAN による三次元ボクセルモデルの超解像

3.4. Marching Cubes 法によるサーフェイス生成

ボクセルからのサーフェイス生成の手法として、2.2.1.節で述べた Marching Cubes 法を用いる。入力は、64^3の超解像三次元モデルで、出力はサーフェイス表現の三次元モデルである。図 11 に超解像三次元モデルから Marching Cubes 法によりサーフェイス生成した例を示す。図 11 では、ボクセルサイズが小さいため、凹凸があまり目立たないサーフェイスモデルが生成されている。

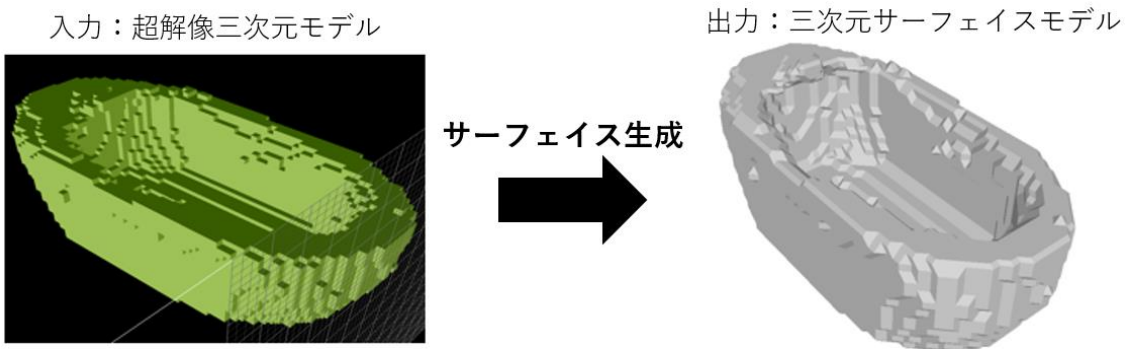


図 11：超解像三次元モデルからのサーフェイス生成

3.5. Laplacian Smoothing によるサーフェイスモデルの平滑化

Laplacian Smoothing[11]（ラプラシアン平滑化）は、ポリゴンメッシュの形状を滑らかにするための手法である。このアルゴリズムでは、各頂点の位置を近傍の頂点の平均位置に基づいて更新することで、メッシュの滑らかさを向上させる。Laplacian Smoothing における各頂点の位置更新は、以下の式で表される。

$$\bar{x}_i = \frac{1}{N}\sum_{j=1}^{N} \bar{x}_j$$

ここで、Nはノードiに隣接する頂点の数、$\bar{x}_j$はj番目の隣接頂点の位置、$\bar{x}_i$はノードiの新しい位置である。

本研究では、Marching Cubes 法によって生成されたサーフェイスモデルに対して Laplacian Smoothing を適用し、より滑らかなサーフェイスモデルを得る。図 12 に三次元サーフェイスモデルに Laplacian Smoothing を適用した例を示す。図 12 では、平滑化処理により滑らかなサーフェイスが生成されていることがわかる。

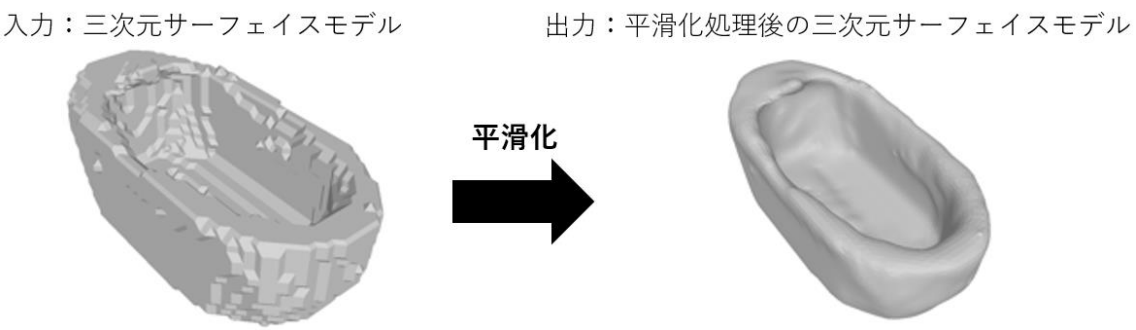


図 12：Laplacian Smoothing による平滑化処理

4. 実験

4.1. 目的

提案手法によるサーフェイス生成の精度を比較し、その有効性を評価する。

比較手法を含めた以下の 4 つの手法で実験を行う。

- 16^3手法：低解像度（解像度16^3）の三次元ボクセルモデルに対して Marching Cubes 法を直接適用。
- 64^3手法：高解像度（解像度64^3）の三次元ボクセルモデルに対して Marching Cubes 法を適用。
- DMTet 手法：低解像度の三次元ボクセルモデルに対して DMTet 法を適用。
- 提案手法：低解像度の三次元ボクセルモデルに対して 3D-SRCGAN を用いて超解像を行い、その後 Marching Cubes 法を適用。

4.2. 実験設定

本研究では、サーフェイス生成、評価に用いる三次元モデルとして、データセット ModelNet10[12]を使用した。このデータセットは bathtub、bed、chair、desk、dresser、monitor、night_stand、sofa、table、toilet の 10 クラスの三次元モデルがあり、それぞれ学習データとテストデータが用意されている。表 1 にデータセット ModelNet10 の内訳を示す。データセットはサーフェイスモデルによる提供されている。このサーフェイスモデルを正解サーフェイスとする。実験で扱う各手法への入力データとするために、学習データ、テストデータともにボクセル数が16^3の低解像度三次元ボクセルモデルと64^3の高解像度三次元ボクセルモデルのペアに変換した。テストデータから各手法を用いて三次元サーフェイスモデルを生成する。

提案手法は 3.2 節で述べた処理の流れでサーフェイスモデルを生成する。3D-SRCGAN は ModelNet10 の学習データを用いて学習する。バッチサイズを 30、Epoch 数を 50 として学習した。

比較評価に平等で行うために、16^3手法、64^3手法、DMTet 手法でそれぞれサーフェイス生成行った後、Laplacian Smoothing で平滑化処理を行う。

サーフェイス生成の評価は、定量評価と主観評価を通じて行う。定量評価では、同一の三次元ボクセルモデルから生成されたサーフェイスを比較し、下記の 4 つの評価項目について各評価項目の平均値を算出する。各評価項目は値が小さいほど、生成されたサーフェイスが正解サーフェイスに近いことを示す。

1 L1 Chamfer Distance

正解のサーフェイスモデルから点群を抽出する。同様に生成されたサーフェイスモデルから点群を抽出する。L1 Chamfer Distance は、生成されたサーフェイスモデルの点群の各点から、正解のサーフェイスモデルの点群の最も近い点までの絶対値距離を測定し、その和して定義する。これにより、生成されたサーフェイスと正解のサーフェイスの間の一致度と定量的に評価する。

2 L2 Chamfer Distance

L1 Chamfer Distance の拡張で、距離の 2 乗の和を計算する。これにより、大きな誤差に対してより大きなペナルティが加わるため、大きな距離のずれをより厳密に評価で

きる。L2 Chamfer Distance は、生成されたサーフェイスと正解サーフェイス間の形状の違いを、L1 Chamfer Distance よりも厳格に評価するための指標として用いる。

3 Light Field Distance（LFD）

正解のサーフェイスモデルと生成されたサーフェイスモデルを異なる角度からレンダリングし Zernike Moments などの特徴量を抽出する。取得した特徴量間の距離により、2 つのメッシュ間の LFD を算出する。これにより、サーフェイスの形状の正確さや滑らかさを定量的に評価することができる。特に、複雑な形状や微細なディテールを再現する能力を評価するために使用する。

4 Point-to-Surface

生成されたサーフェイスから抽出した点群内の各点が、正解サーフェイスにどれだけ近いかを評価する指標である。この評価は、生成されたサーフェイスが正解の形状にどれだけ忠実に再現されているかを示す。

表 1：ModelNet10 の内訳

クラス名	bathtub	bed	chair	desk	dresser
学習データ数	106	515	889	200	200
テストデータ数	50	100	100	86	86
クラス名	monitor	night_stand	sofa	table	toilet
学習データ数	465	200	680	392	344
テストデータ数	100	86	100	100	100

4.3. 実験結果

図 13 に monitor のクラスの三次元ボクセルモデルに各手法を適用したときに生成されたサーフェイスモデルの一例を示す。主観的に見ると、DMTet 手法によるサーフェイス生成は穴が多数存在して、サーフェイス滑らかではない。また、16^3手法ではボクセルサイズが大きいため、凹凸が目立つ。提案手法では、平滑化処理により滑らかな面が生成されている。64^3手法と提案手法（平滑化処理前）では、ボクセルサイズが小さいため凹凸が比較的目立たない。しかし、生成されたサーフェイスが少し粗いことがわかる。

表 2 に monitor のクラスに各手法を適用し生成したサーフェイスモデルの定量評価結果を示す。64^3手法が、各項目において定量評価が最も良い。提案手法によるサーフェイス生成は、各項目において定量評価が他の手法に比べて 2 番目に良い。DMTet 手法によるサーフェイス生成は、L2 Chamfer Distance、L1 Chamfer Distance、Point-to-Surface の項目においては他の手法に比べて 3 番目に良い。16^3手法は、LFD の項目においては他の手法に比べて 3 番目に良いが、それ以外の項目においては最も悪い評価である。

その他の実験結果の詳細については付録 A に示す。

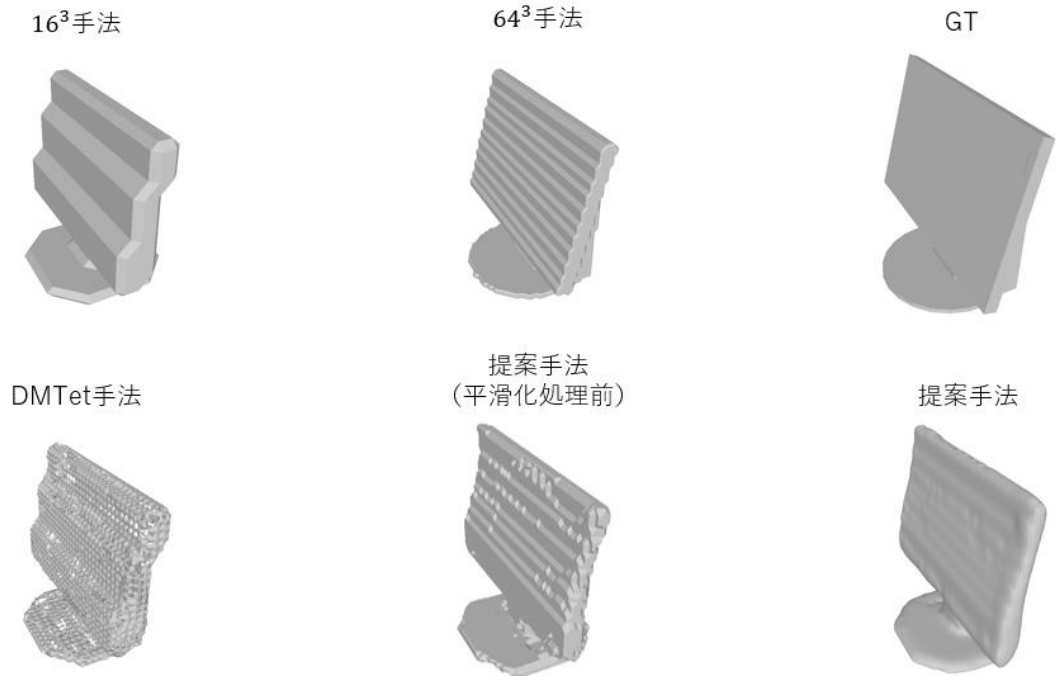


図 13：各手法で生成されたサーフェイスモデル

表 2：各手法の定量評価

手法	16^3手法	64^3手法	DMTet 手法	提案手法
L2 Chamfer Distance	0.035952	0.027969	0.035134	0.029615
L1 Chamfer Distance	0.193098	0.154968	0.187012	0.161413
LFD	3407	1421	4954	3202
Point-to-Surface	0.067025	0.050333	0.065687	0.054315

4.4. 考察

本実験の結果から以下のことが分かった。

まず、DMTet 法によるサーフェイス生成は完全な実装にはなっていないため、存在しないサーフェイスが生成されたり、穴が開いたりして論文上のような結果にはならなかった。そのため、定量評価に影響してあまり良い精度ではなかった。

次に、低解像度三次元ボクセルモデルを直接 Marching Cubes 法によりサーフェイス生成を行うとボクセルサイズが大きいため、凹凸が目立つという問題がみられた。しかし、解像度が上がるとボクセルサイズが小さくなり、凹凸が目立たくなった。定量評価でも64^3手法が16^3手法より良いことから、ボクセルサイズを小さくすることで、サーフェイス生成の精度を向上できることが分かった。

提案手法で生成されたサーフェイスモデルは、凹凸と穴が目立っていた。これは、3D-SRCGAN の精度が完璧ではなく、低解像度三次元ボクセルモデルを超解像する際に、穴が生じることがあるからである。一方、表 2 を見ると、提案手法の定量評価は、16^3手法より良い結果が得られた。本実験では、主観的に見ると提案手法で生成されたサーフェイスは少し粗いが、数値的に良い結果が得られた。これにより、提案手法でサーフェイス生成した方が16^3手法より良い結果が得られることがわかる。

以上のことから、本研究では、「低解像度三次元ボクセルモデルからサーフェイスモデルを生成する際に、低解像度三次元ボクセルモデルを直接 Marching Cubes 法でサーフェイス生成を行うより、超解像して Marching Cubes 法でサーフェイス生成を行うとより精細なサーフェイスモデルが得られる。」と結論づけた。

5. おわりに

本研究では、3D-SRCGAN を用いた低解像度三次元ボクセルモデルからサーフェイスモデルへの変換手法を提案した。従来手法である Marching Cubes 法および DMTet 法の課題を克服するため、超解像処理を導入し、より滑らかで精細なサーフェイスモデルを生成することを目指した。

提案手法の有効性を検証するため、ModelNet10 データセットを用いた実験を実施し、低解像度三次元ボクセルモデルに対する直接的な Marching Cubes 法適用との比較を行った。その結果、提案手法により、より高精度なサーフェイスモデルが得られ、特に Chamfer Distance や Point-to-Surface の評価指標において良好な結果が示された。

本研究の課題としては、DMTet 法の実装が不完全であったため、完全な実装を行うことが挙げられる。完全実装が行えることで、高品質なサーフェイスモデルを生成することができる可能性がある。また、生成されたサーフェイスモデルに一部凹凸や穴が残る問題があり、3D-SRCGAN モデルのさらなる精度向上や、後処理の改良が必要である。

謝辞

本論文の作成にあたり、適切なサポート、また丁寧に指導して下さった椋木雅之教授には深く感謝いたします。指導教員である椋木雅之教授には研究や論文執筆に関して様々なご助言をいただきました。本当にありがとうございました。

椋木研究室の皆様には、研究を進めるにあたって、様々な助言をいただきました。皆様のおかげで充実した研究生活を過ごすことができました。大変感謝しております。ありがとうございました。

参考文献

[1] Lorensen, William E.; Cline, Harvey E. "Marching cubes: A high resolution 3D surface construction algorithm", ACM SIGGRAPH Computer Graphics. 21 (4), pp. 163-169 (1987)

[2] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, Sanja Fidler, “Deep Marching Tetrahedra: a Hybrid Representation for High-Resolution 3D Shape Synthesis”, Advances in Neural Information Processing Systems (NeurIPS), pp. 6087-6101 (2021)

[3] Akio Doi、Akio Koide、”An efficient method of triangulating equi-valued surfaces by using tetrahedral cells”、IEICE TRANSACTIONS on Information and Systems、74(1)、pp. 214–224、（1991）

[4] 野村 淳也、椋木雅之、”クラス識別を導入した 3D-SRGAN によるボクセル超解像”, 宮崎大学 工学研究科 工学専攻 機械・情報系コース 修士論文、(2023)

[5] Leonard R. Herrmann, "Laplacian-isoparametric grid generation scheme", Journal of the Engineering Mechanics Division, 102 (5), pp. 749-756 (1976)

[6] Xudong Mao、Qing Li、Haoran Xie、Raymond YK Lau、Zhen Wang、Stephen Paul Smolley、”Least squares generative adversarial networks”、In Proceedings of the IEEE international conference on computer vision、pp. 2794–2802、（2017）

[7] “kaolin/example/tutorial/dmtet_tutorial.ipynb”、https://github.com/NVIDIAGameWorks/kaolin/blob/master/examples/tutorial/dmtet_tutorial.ipynb

[8] 岡和寿, 椋木雅之, “SRGAN の 3 次元ボクセルモデル超解像への適用”, 画像電子学会論文誌, vol. 48, no. 4, pp. 448-496（2019）

[9] C Ledig, L Theis, F Huszar, J Caballero, A Cunningham, A Acosta, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 105-114 (2017)

[10] A Radford, L Metz, S Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks", International Conference on Learning Representations (ICLR), （2016）

[11] “Laplacian Smoothing”、 https://en.wikipedia.org/wiki/Laplacian_smoothing

[12] ModelNet10, ModelNet40, http://modelnet.cs.princeton.edu/

A. 付録

図 A-1 から A-3 に各手法の入力になる三次元ボクセルモデルを示す。図 A-4 から A-7 に各手法から生成された三次元サーフェイスモデルを示す。表 A-1 から A-4 に各手法による定量評価を示す。

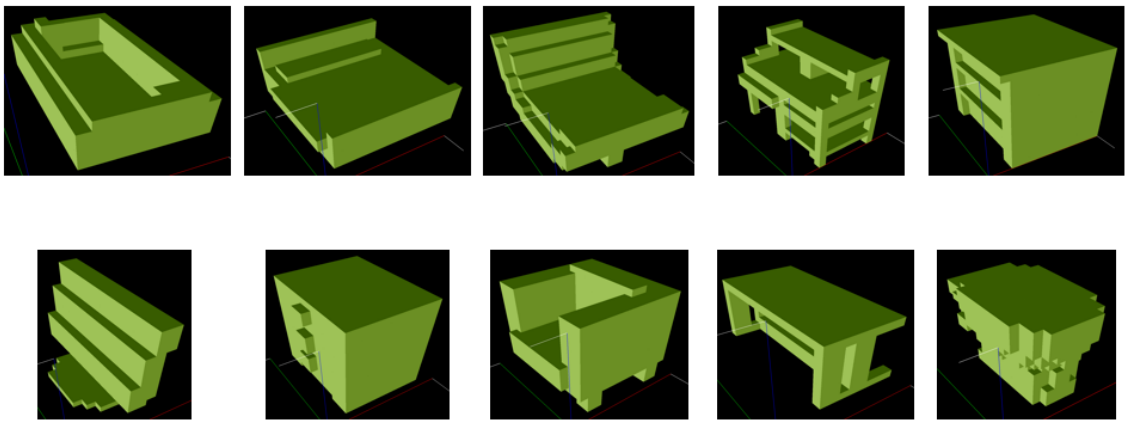


図 A-1：ModelNet10 による解像度16^3の三次元ボクセルモデル

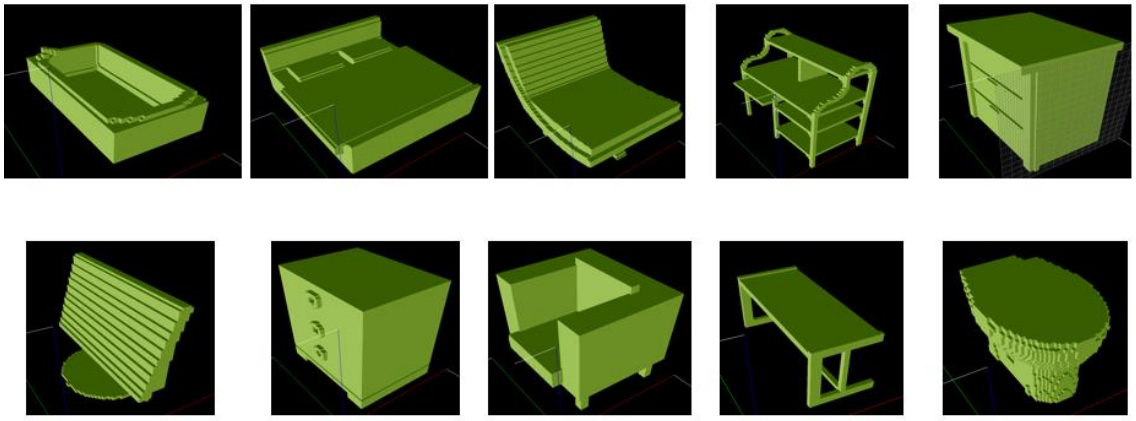


図 A-2：ModelNet10 による解像度64^3の三次元ボクセルモデル

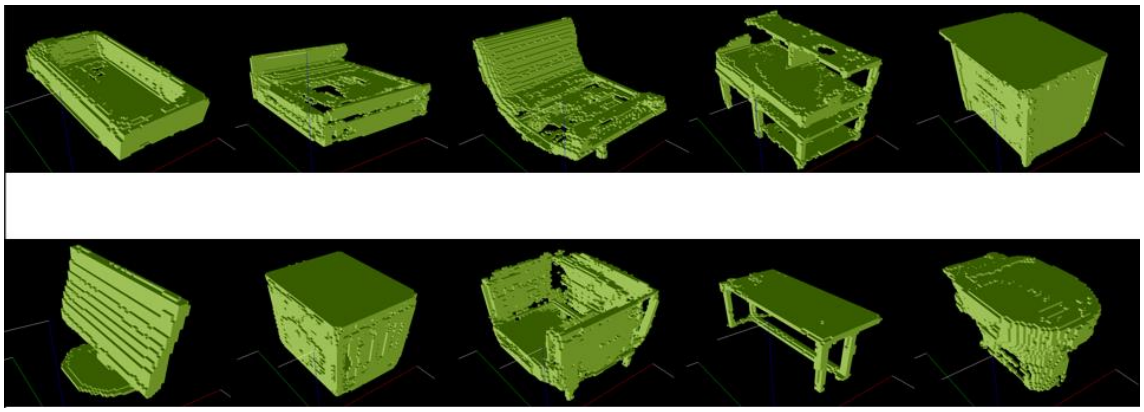


図 A-3：ModelNet10 による解像度64^3の超解像三次元ボクセルモデル

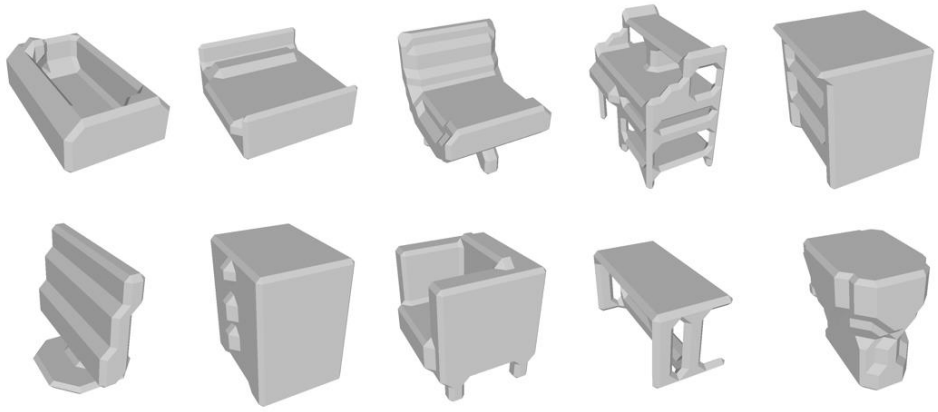


図 A-4：16^3手法によるサーフェイス生成

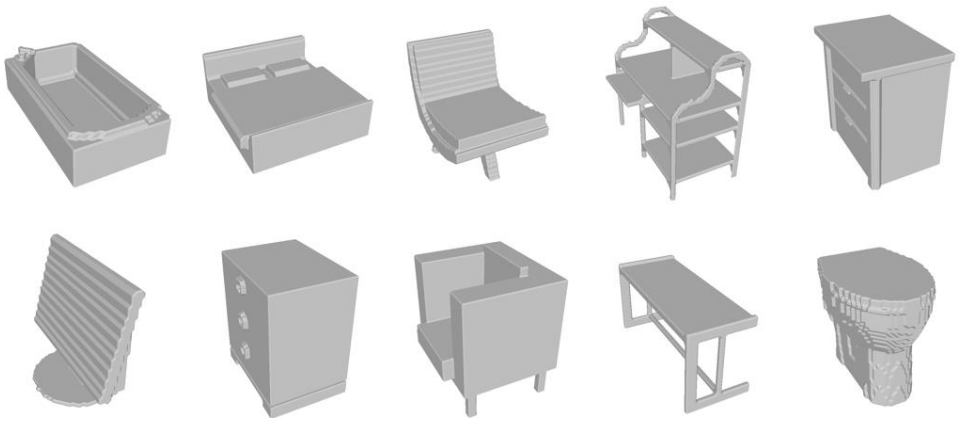


図 A-5：64^3手法によるサーフェイス生成

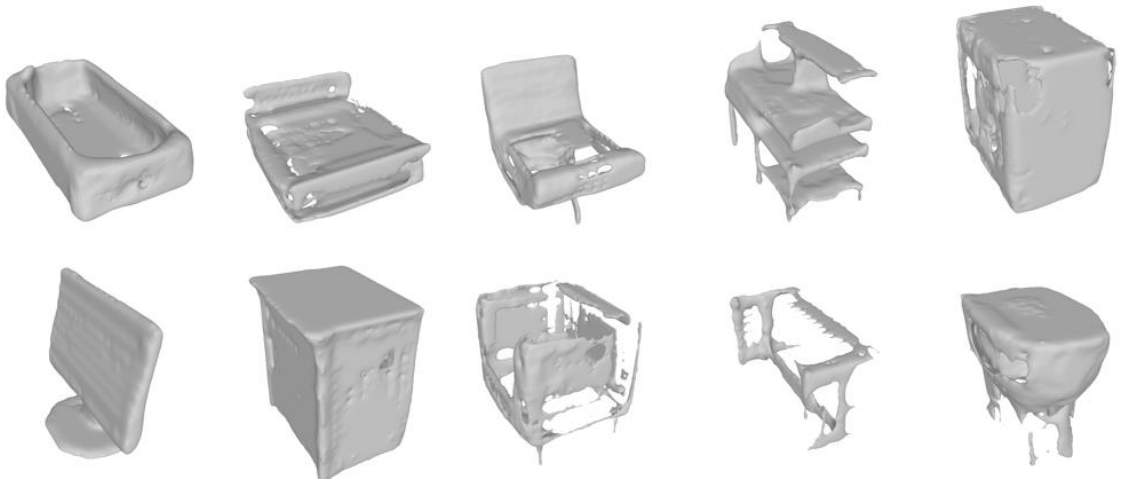


図 A-6：提案手法によるサーフェイス生成

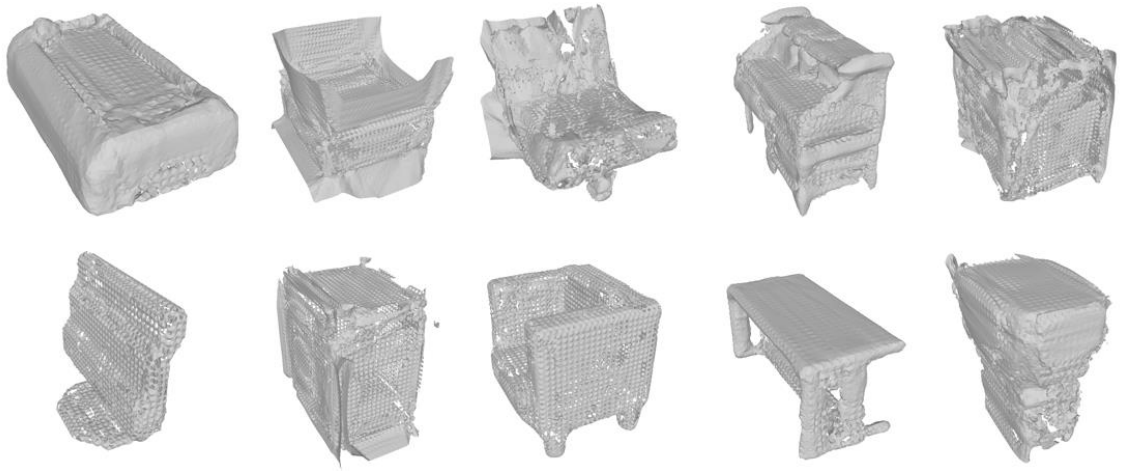


図 A-7： DMTet 手法によるサーフェイス生成

表 A-1：16^3によるサーフェイス生成の定量評価

クラス名	bathtub	bed	chair	desk	dresser
L2 Chamfer Distance	0.04673	0.048678	0.021721	0.034268	0.077209
L1 Chamfer Distance	0.212287	0.228043	0.157072	0.191965	0.319930
LFD	2534	3469	4431	3013	1815
Point-to-Surface	0.063649	0.090575	0.057056	0.068270	0.137840
クラス名	monitor	night_stand	sofa	table	toilet
L2 Chamfer Distance	0.035952	0.058126	0.022265	0.034344	0.024437
L1 Chamfer Distance	0.193098	0.262223	0.166869	0.196857	0.171524
LFD	3407	1890	3288	4117	2622
Point-to-Surface	0.067025	0.103913	0.080652	0.080872	0.077313

表 A-2：64^3手法によるサーフェイス生成の定量評価

クラス名	bathtub	bed	chair	desk	dresser
L2 Chamfer Distance	0.040543	0.024947	0.017255	0.029306	0.060232
L1 Chamfer Distance	0.189169	0.152769	0.130391	0.172826	0.281970
LFD	1306	1549	1895	1387	781
Point-to-Surface	0.062292	0.056484	0.055847	0.070402	0.125543
クラス名	monitor	night_stand	sofa	table	toilet
L2 Chamfer Distance	0.027969	0.048209	0.012194	0.047898	0.018152
L1 Chamfer Distance	0.154968	0.237516	0.117978	0.235966	0.140088
LFD	1421	702	1732	1819	1153
Point-to-Surface	0.050333	0.101111	0.056431	0.116339	0.065589

表 A-3：提案手法によるサーフェイス生成の定量評価

クラス名	bathtub	bed	chair	desk	dresser
L2 Chamfer Distance	0.043687	0.028252	0.018994	0.029147	0.055302
L1 Chamfer Distance	0.201474	0.156066	0.141084	0.178011	0.260814
LFD	4034	3515	4096	4536	1892
Point-to-Surface	0.071591	0.053063	0.063935	0.077259	0.105385
クラス名	monitor	night_stand	sofa	table	toilet
L2 Chamfer Distance	0.029615	0.045138	0.014576	0.048286	0.018014
L1 Chamfer Distance	0.161413	0.230788	0.127545	0.244296	0.139319
LFD	3202	3168	5380	7363	4329
Point-to-Surface	0.054315	0.099663	0.065021	0.131472	0.062933

表 A-4： DMTet 手法によるサーフェイス生成の定量評価

クラス名	bathtub	bed	chair	desk	dresser
L2 Chamfer Distance	0.044318	0.044440	0.020481	0.033870	0.064374
L1 Chamfer Distance	0.202548	0.219875	0.150425	0.185321	0.273861
LFD	4191	4894	6139	5142	4199
Point-to-Surface	0.059569	0.089232	0.051470	0.068806	0.116756
クラス名	monitor	night_stand	sofa	table	toilet
L2 Chamfer Distance	0.035134	0.042446	0.017070	0.051154	0.026682
L1 Chamfer Distance	0.187012	0.224392	0.143995	0.241897	0.173075
LFD	4954	3733	5256	5877	4313
Point-to-Surface	0.065687	0.093775	0.070916	0.094451	0.077229