

平成 30 年度 修士論文

SRGAN の 3 次元モデル超解像への拡張

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

学籍番号 T1703007

岡 和寿

概要

本論文では、3次元モデルの超解像を行う 3D-Super Resolution Generative Adversarial Networks (3D-SRGAN) を提案する。

低解像度データから高解像度データを復元・生成する技術は超解像と呼ばれ、画像分野では幅広く研究されている。しかし、3次元モデルを超解像する研究は見当たらない。そこで、3次元モデルの超解像を行う 3D-SRGAN を提案する。3D-SRGAN は画像超解像で有効性が知られている SRGAN を 3次元に拡張した手法である。3D-SRGAN と局所的な情報を用いた手法で比較実験を行い、3D-SRGAN の生成結果が正解の高解像度 3次元モデルにより近いことを示した。また、3D-SRGAN では学習データが 1 クラスの場合でも、超解像に多少の影響はあるが他クラスの超解像を行えることを実験により実証した。

目次

1. はじめに.....	4
2. 超解像.....	6
2.1. 超解像の定義.....	6
2.2. 画像の超解像の従来研究.....	7
2.3. SRGAN による画像の超解像.....	9
2.3.1. SRGAN.....	9
2.3.2. SRGAN の特徴.....	10
2.3.3. SRGAN のネットワーク構造.....	12
2.3.4. SRGAN による 2 値画像超解像の予備実験.....	13
2.4. 3次元モデルの超解像.....	17
3. 3D Super Resolution Generative Adversarial Network による超解像.....	18
3.1. 3D-SRGAN の構成.....	18
3.2. Generator の構造.....	19
3.3. Discriminator の構造.....	22
3.4. 3D-SRGAN の学習方法.....	24
3.4.1. Generator の学習更新.....	25
3.4.2. Discriminator の学習更新.....	26
4. 実験.....	27
4.1. 信号处理的なアプローチとの比較.....	27
4.1.1. 比較手法.....	27
4.1.2. 実験方法.....	29
4.1.3. 結果・考察.....	30
4.2. 学習データとは別クラスでの超解像.....	34
4.2.1. 実験方法.....	34

4.2.2. 結果・考察.....	34
4.3. 局所的情報を用いた超解像.....	38
4.3.1. 実験方法.....	38
4.3.2. 結果・考察.....	38
5. おわりに.....	41
謝辞.....	42
参考文献.....	43

1. はじめに

近年、個人向け 3D プリンターや 3D-CAD の登場によって 3 次元モデルを扱うことが多くなってきた。3 次元モデルの表現方法には、直線や曲線で構成されたワイヤーフレーム表現、平面や曲面で構成されたサーフェイス表現、複数の立体の集合で構成されたソリッド表現などがある[1]。ソリッド表現の 1 種に、ボクセル表現がある。ボクセル表現では、3 次元空間を格子状の細かい立方体(あるいは直方体)に分けて 3 次元モデルを表現する。各立方体をボクセルと呼ぶ。ボクセルは、3 次元空間のその位置が物体内部もしくは境界上である場合は値 1 を持ち、物体外部である場合は値 0 を持つ。これにより 3 次元モデルの形状を表現する。本研究では、このボクセル表現の 3 次元モデルを扱う。

Blender[2]や Fusion 360[3]などの 3 次元モデルを作成するソフトウェアではサーフェイス表現が多く使われる。しかし、手軽に 3 次元モデルを作成するためにボクセル表現が使われることもある。手軽に 3 次元モデルを作成する例として Minecraft[4]がある。Minecraft とは、Markus Persson らが開発したサンドボックスゲームである。Minecraft では、立方体のブロックを積み木のように積み重ねてボクセル表現の 3 次元モデルを手軽に作成することができる。しかし、高精細な 3 次元モデルを作成するには、多数のブロックを積み重ねる必要があり、手間が掛かる。粗い 3 次元モデルから高精細な 3 次元モデルを作成できれば、この手間を低減できる。

このような粗いデータから高精細なデータを生成する処理を超解像[5]と呼ぶ。画像分野では、粗い画像から高精細な画像を生成する超解像技術が多く研究されている。しかし、3 次元モデルの超解像技術の研究は見当たらない。

そこで、本研究では画像の超解像で有効性が知られている SRGAN を拡張した 3D-Super Resolution Generative Adversarial Networks (3D-SRGAN) を提案する。SRGAN は画像の超解像で効果があったため、3 次元モデルでも粗い 3 次元

モデルから高精細な 3 次元モデルを生成する超解像で効果が期待できる。

以下、2 章で画像と 3 次元モデルの超解像について述べ、3 章で提案手法である 3D-Super Resolution Generative Adversarial Networks について述べる。そして 4 章で 3 D-SRGAN での 3 次元モデル超解像の生成精度が高いことを実験により実証する。最後に 5 章で結論を述べる。

2. 超解像

2.1. 超解像の定義

超解像とは低解像度のデータから高解像度データを復元・生成する技術である。解像度とは、画素やボクセルが一定の長さの間にどれだけ存在しているかを表している。解像度が高いと、画像や 3 次元モデルを表現する格子が細くなる。低解像度データは粗いデータとなり、高解像度データは高精細なデータとなる。

超解像は、低解像度データから高解像度データを生成する技術であるが、この問題は不良設定問題(ill-posed problem)[6]である。不良設定問題とは、解を求めるための必要な情報が一部欠けている問題のことである。低解像度データを高解像度データに変換するためには、低解像度データに存在しない部分のデータを生成する必要がある。この低解像度データに存在しない部分には、無数の生成パターンがありえる。データが周波数分解でき、サンプリング定理を満たす場合は、信号処理的アプローチで補間が行えるが、多くの実データでは、ノイズが含まれる上、デジタルデータでは量子化誤差も含まれるため、実用的には良い結果が得られない。そのため、一般に低解像度データから適切な高解像度データを生成することは難しい問題である。

2.2. 画像の超解像の従来研究

近年、2次元画像超解像の研究が盛んである。2次元画像超解像では複数の画像から行う手法と1枚の画像から行う手法がある。

複数の画像から行う手法には、複数の Bayer 画像データからデモザイキング処理と超解像処理を同時に行う手法[7]や複数の観測画像から位置合わせ処理と再構成処理を同時に行う手法[8]などがある。これらの手法では、同じシーンに対する複数の画像から得られるサブピクセルの情報を利用して、超解像を実現している。これらは観測に基づく手法なので、正しい高解像度画像が得られるが、適切な設定で撮影した複数の画像が必要である。

1枚の画像から超解像する手法には、補間の考え方に基づく信号処理的なアプローチによる手法[9][10][11]と学習やデータベースに基づいて高周波成分を復元する手法[12]がある。前者の手法では、もともと高周波成分が含まれていない画像に対しては大きな改善が期待できない。一方後者の手法は、学習に使った画像やデータベース内の画像の情報を利用して、1枚の画像から高周波成分を付与できる。そのため高周波成分が含まれていない画像であっても、高周波成分を付与した高解像度画像を生成することができる。しかし、低解像度画像に存在しない部分のデータは、学習やデータベースに利用した異なる画像を参考にして生成するため、必ずしも正しい高解像度画像が得られるとは限らない。

近年では、学習の中でも深層学習を用いた超解像の研究[13][14]が多く見られる。深層学習とは、機械学習の1種であるニューラルネットワークにおいて深い階層構造を持ったネットワークを利用する学習方法である。深層学習の1つに畳み込みニューラルネットワーク(Convolution Neural Network : CNN)[15]がある。CNNは畳み込み層で複数のカーネル(フィルター)と呼ばれる格子状の数値データを用いて入力データに畳み込み処理を行い、特徴マップを出力する。畳み込み処理とは注目した座標とその周囲の座標の画素値に対し、カーネルの

対応する各値を掛けて加算したものを出力画素値とする処理である。CNNでは、入力の各チャンネル(層)でカーネルを少しずつずらして畳み込み処理を行っている。このCNNを使用した超解像の手法[16][17]では、多数の実画像を学習に利用することで、一般的な画像における高解像度データの生成パターンを学習できるため、人間にとってより自然な高解像度画像を生成することができる。また、数値評価結果でも、従来の手法よりも生成結果の精度が高くなっている。

2.3. SRGAN による画像の超解像

2.3.1. SRGAN

Super Resolution Generative Adversarial Network(SRGAN)[18]は深層学習を用いた超解像の手法である。この手法は Generative Adversarial Network(GAN)[19]と呼ばれる生成モデルを応用している。GAN は Generator(生成器)と Discriminator(識別器)の2つのニューラルネットワークで構成されている。Generator はより学習データに近いデータを生成し、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。このような学習により、最終的には Generator が学習データと同じようなデータを生成できることが期待される。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ため、Discriminator の正答率は約50%となる。このような仕組みにより GAN は、学習データに非常に似た新たなデータを生成できる。SRGAN は、GAN の持つこの特徴を利用して超解像を行う。そのため、自然な高解像度画像を生成することができる。

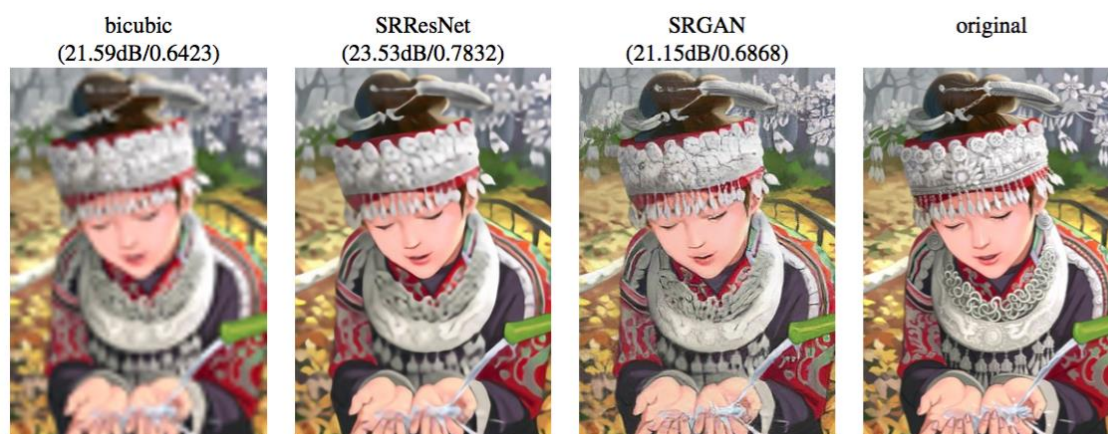


図 1. SRGAN と従来手法の画像超解像 (引用[18])

2.3.2. SRGAN の特徴

SRGAN の特徴として以下が挙げられる。

- (1) Content Loss[20]による主観的特徴の学習
- (2) 2つのネットワークによる敵対的な学習(GAN の利用)
- (3) Residual Network(ResNet)[21]による深いネットワーク
- (4) 逆畳み込みではなく Pixel Shuffler[22]を利用した画像拡大

1つ目の特徴は、学習の際に Content Loss という Loss 関数を用いている点である。Content Loss には画像に何らかの操作を行っても、元の画像の大まかな見た目を維持しようとする性質がある。SRGAN では、Content Loss を利用することで、画像に拡大の操作を行っても主観的な特徴を変えずに超解像を行うことができるようにしている。

2つ目の特徴は、GAN を利用している点である。SRGAN では、GAN の持つ「学習データと見分けのつかないデータを生成する」性質を利用している。しかし、GAN は学習データに共通する性質を残した画像を生成するが、入力とは別の画像を生成してしまう。一方、上記の Content Loss は大まかな見た目を維持するだけであり、これだけではぼやけた画像が生成されてしまう。例えば、学習データが顔画像であった場合、入力に顔画像を与えると、GAN だけでは入力と

は別人の顔画像が生成される。一方、Content Loss だけでは入力と同じ人の顔が生成されるが細部までは復元できない。そのため、SRGAN は Content Loss と GAN を適度に組み合わせて超解像を行っている (図 1)。

3 つ目の特徴は、ResNet を利用している点である。ResNet は、CNN の層を深くしたニューラルネットワークの構造である。CNN では、層を深くする(層の数を多くする)とより高度で複雑な特徴を抽出できるとされている。しかし、単純に層を深くすると学習の性能が悪化してしまう [23][24]。ResNet では層で求める最適な出力ではなく、層の入力と出力の差を学習している。これにより、ResNet では学習性能を悪化させることなく深いネットワーク構造を実現できるため、高度で複雑な特徴を抽出することが期待できる。

4 つ目の特徴は、Pixel Shuffler 処理により画像拡大を行っている点である。深層学習では、低次元の特徴マップから高次元の特徴マップを生成する際に逆畳み込み処理が利用される。しかし、逆畳み込み処理では処理速度が遅く、精度があまり高くないと考えられている [25]。Pixel Shuffler は、入力の特徴マップの各ピクセルを並び替えて高解像度な特徴マップを出力する。処理がメモリコピーだけであるため処理速度が速く、オーバーラップがなく生成画像がぼやけにくい性質がある。

SRGAN はこれらの特徴を組み合わせることで、より本物に近い画像が生成でき、超解像の課題である可能な無数のパターンから自然なパターンを選択することを達成している。

2.3.3. SRGAN のネットワーク構造

次に SRGAN の構造について説明する。SRGAN は GAN と同じく Generator と Discriminator のネットワークで構成されている。Generator と Discriminator はそれぞれ図 2 のような構造になっている。

Generator は大きく入力層、ResNet 層、Pixel Shuffler 層、出力層の 4 層に分けられる。Generator には入力として学習データの低解像度画像が与えられ、4 つの層で処理され、高解像度画像を出力する。ResNet 層は、5 つの Residual Block で構成されている。それぞれの Block では、入力を Block の最後に加算して出力している。5 つ目の Block の処理後には、畳み込み層で処理した出力に 1 つ目の Block の入力を加算している。この構造によって深いネットワークが構成できている。

Discriminator は入力層、Block 層、出力層の 3 層に分けられる。Discriminator には入力として、学習データの高解像度画像もしくは Generator が生成した高解像度画像が与えられ、0~1 の値を出力する。この値は、入力データが学習データである確率を表している。

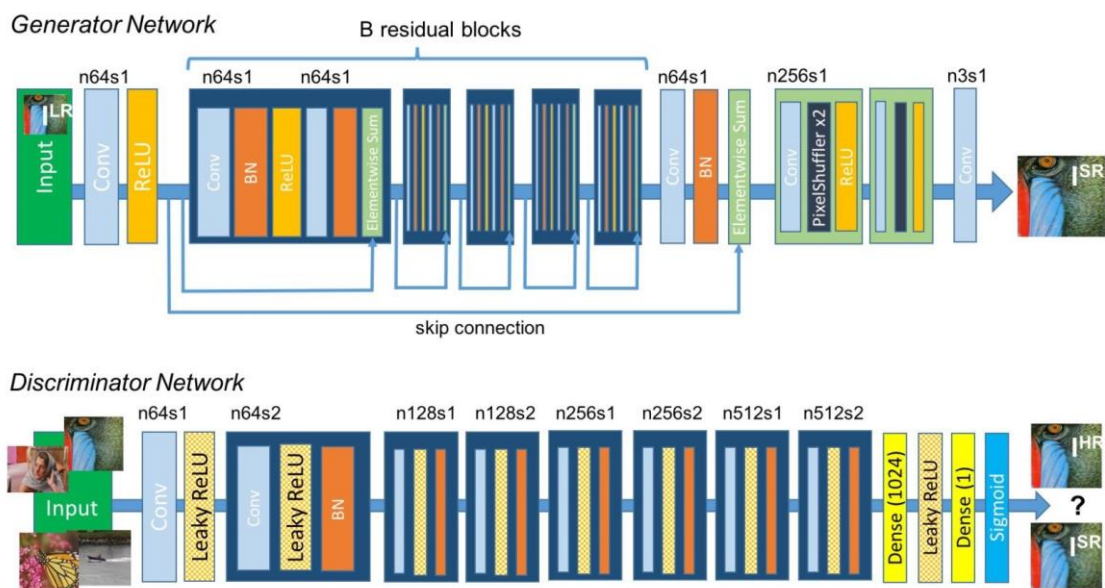


図 2. SRGAN の Generator と Discriminator (引用[18])

2.3.4. SRGAN による 2 値画像超解像の予備実験

SRGAN では、精度の高い超解像ができるという報告がある。SRGAN を 3 次元モデルに拡張できれば、3 次元モデルでも精度の高い高解像度 3 次元モデルが生成できると期待できる。本研究で扱うボクセル表現の 3 次元モデルでは、3 次元空間上のボクセルは、0 または 1 の 2 値のいずれかを格納している。これは画像では 2 値画像に相当する。しかし、2 次元画像超解像の研究はカラー画像や濃淡画像を対象にしており、2 値画像を対象にした研究は見当たらない。そこで、SRGAN で 2 値画像を対象にした予備実験を行った。

まず、SRGAN が本当に精度の高い超解像ができるか、カラー画像、濃淡画像で実験した。学習にはデータセット RAISE_HR、LR[26]を使用した。RAISE_HR、LR はカラー画像や濃淡画像であり、高解像度画像と低解像度画像がペアで与えられる。これらの学習データを SRGAN で学習した。学習回数は 70 万回である。学習した生成モデルでカラー画像と濃淡画像の超解像を行った。図 3 に生成結果を示す。図 3 の結果から、カラー画像ではひげや目などが復元され、濃淡画像では橋や植物の細部まで復元していた。どちらの生成結果でも元の画像と主観的にほとんど違いがない。このことから SRGAN は、カラー画像と濃淡画像で精度の高い超解像を行うことができることが分かった。

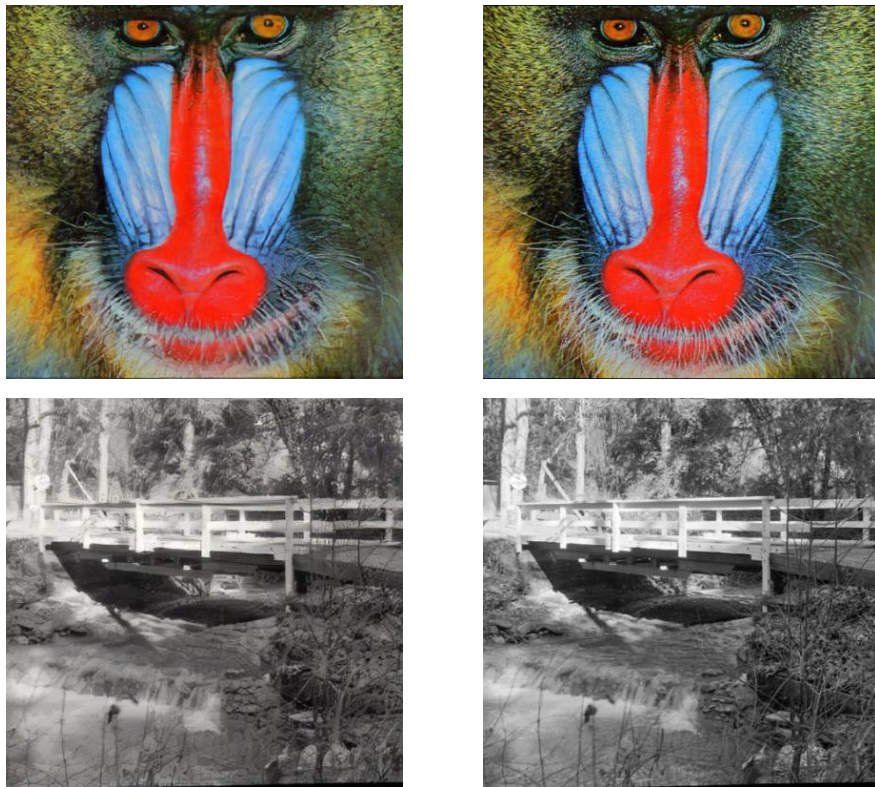


図 3. SRGAN によるカラー画像と濃淡画像の超解像

左: 生成画像、右: 正解画像

次に、先ほど学習した生成モデルで 2 値画像の超解像を行った。図 4 に生成結果を示す。2 値画像でも超解像を行うことができた。しかし、図 4 のマントヒヒの目や植物などの細かい部分の復元ができず、生成精度が悪いことが分かった。原因として、学習データが 2 値画像ではなかったことが考えられる。

そこで、学習の前に RAISE_HR、LR をそれぞれ 2 値化処理した学習データを用意した。学習回数は 70 万回とし、2 値化した画像の学習データで SRGAN の学習を行った。図 5 に新たに学習した生成モデルで 2 値画像の超解像を行った結果を示す。図 4 よりも目や植物などの細部が復元されており、正解画像に近い画像を生成した。SRGAN では、2 値データでも十分に精度が高いことが分かった。

これらの実験結果から、SRGAN では 2 値画像の超解像で精度の高い高解像度画像を生成することが可能であると言える。しかし、学習データの影響が大きく 2 値画像の超解像を行う場合、学習データはカラー画像や濃淡画像ではなく、2 値画像の方がより精度が高いことも分かった。以上のことから、SRGAN を拡張して 3 次元モデルの超解像を行っても精度の高い生成結果が期待できる。

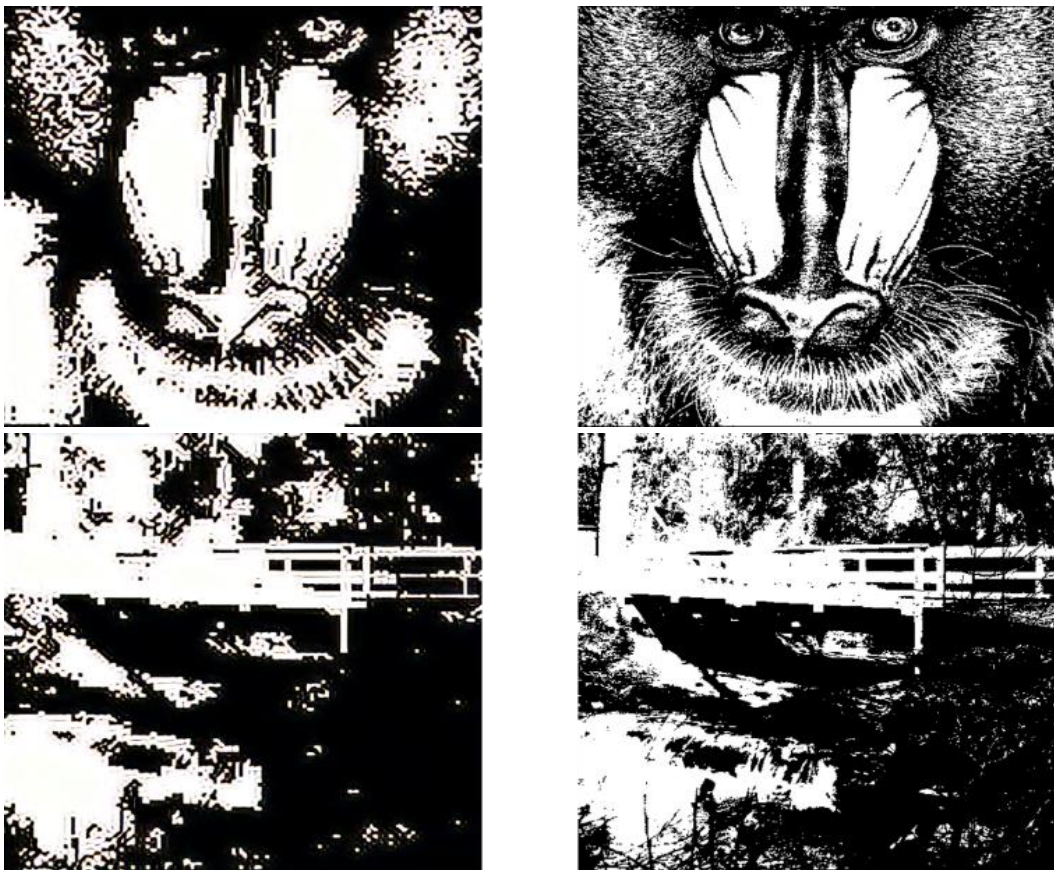


図 4. SRGAN による 2 値画像の超解像

左: 生成画像、右: 正解画像

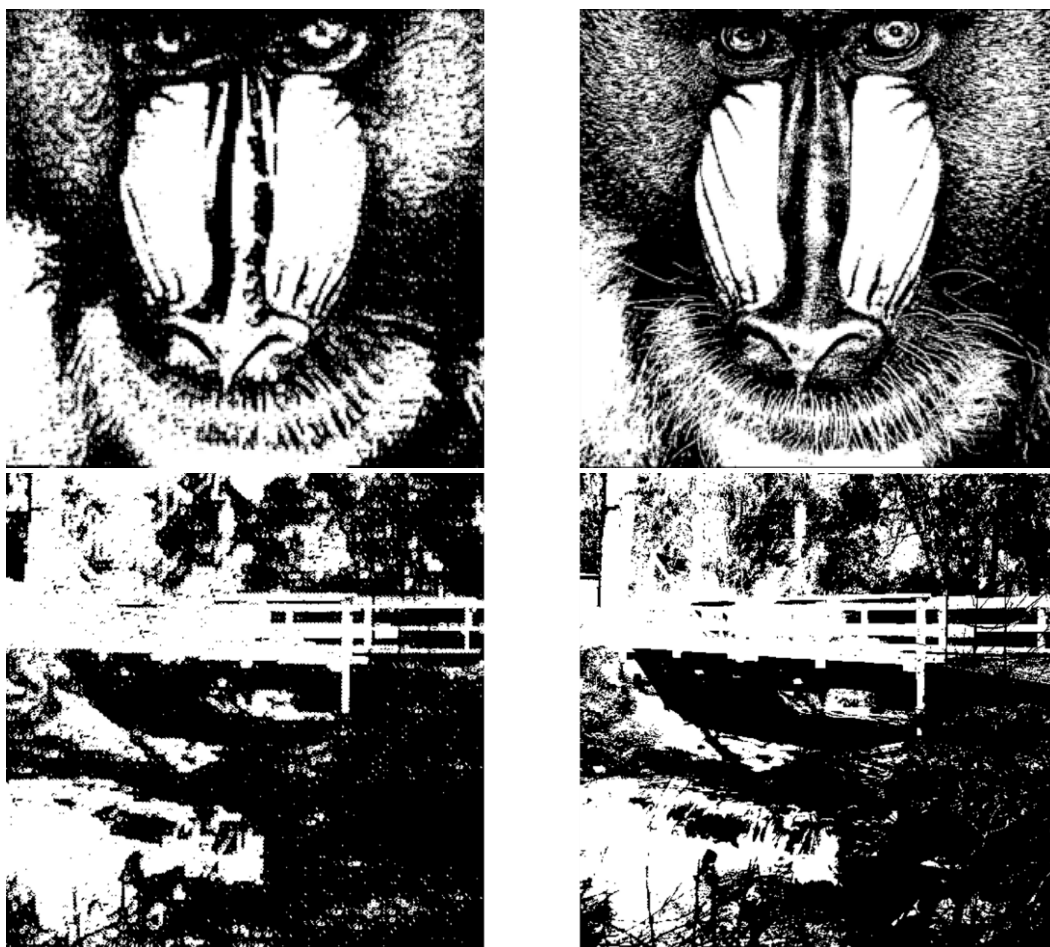


図 5.2 値画像を学習した SRGAN による 2 値画像の超解像

左: 生成画像、右: 正解画像

2.4.3 3次元モデルの超解像

近年、3D プリンターや 3D-CAD の登場によって 3次元モデルを対象にした研究は盛んである。複数の観測画像から 3次元データの復元を行う手法[27]や全方位カメラのデータに SfM を用いて 3次元モデリングを行う手法[28]、複数の解像度の 3次元モデルから形状の類似度を比較する手法[29]など幅広く研究されている。しかし、本研究の目的である 3次元モデルの超解像の研究は見当たらない。そこで、SRGAN の技術を利用して 3次元モデルの超解像を行う 3D-SRGAN を提案する。この手法では、学習データに低解像度 3次元モデルと高解像度 3次元モデルのペアを用いて、超解像を行う。学習した生成モデルでは、低解像度 3次元モデルから自動で高解像度 3次元モデルが生成できる。

3. 3D Super Resolution Generative Adversarial Network による超解像

3.1. 3D-SRGAN の構成

ここでは 3D-SRGAN の大まかな構成を述べる。3D-SRGAN は、SRGAN を 3 次元モデル超解像ができるように拡張したものである。3D-SRGAN では、Generator と Discriminator の構造は基本的に SRGAN と同様とし、入出力データとしてボクセル表現の 3 次元モデルが扱えるように拡張している。学習データには、低解像度 3 次元モデルと高解像度 3 次元モデルのペアを使用する。

3D-SRGAN は、Generator と Discriminator の 2 つのニューラルネットワークで構成されている。Generator は低解像度 3 次元モデルから高解像度 3 次元モデルを生成し、Discriminator は入力された 3 次元モデルが学習データの高解像度 3 次元モデルなのか Generator が生成した高解像度 3 次元モデルなのかを識別する(図 6)。この 2 つは敵対的な関係であり、それぞれの目的は、Generator は Discriminator を騙すように学習データと似た 3 次元モデルを生成することであり、Discriminator は学習データと 3 次元モデルを見分けられるようになることである。最終的には、Discriminator が識別できないような 3 次元モデルを Generator が生成できることが 3D-SRGAN の目的となる。また、この時の Generator が生成モデルとなる。

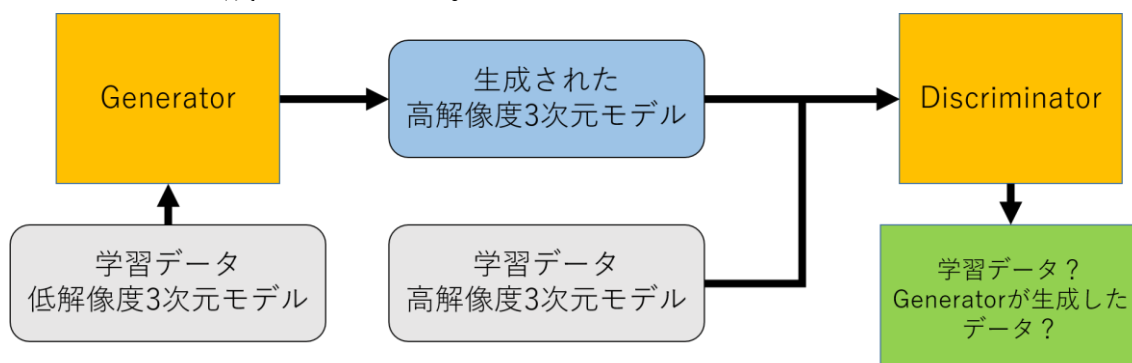


図 6. GAN の学習の概要図

3.2. Generator の構造

Generator のネットワークの構造について説明する。以降、入力チャンネル数を C_{in} 、出力チャンネル数を 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少しずつずらしていく間隔のことである。Generator の 3 次元畳み込み層のストライド数は全て $s=1$ である。

入力には学習データの低解像度 3 次元モデルが与えられ、出力として高解像度 3 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 3 次元モデルを 3 次元畳み込み層によって、複数のチャンネルとして ResNet 層に与えている。ResNet 層では 3 次元モデルの特徴を抽出し、Voxel Shuffler 層で 3 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 3 次元モデルを出力している。

3 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。まず、正規化関数では 3 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、学習の安定を図ることができる [30]。活性化関数にはいくつか種類がある。Generator では ReLu 活性化関数(図 7)を使用しており、式(1)で表現できる。

$$f(x) = \max(0, x) \quad (1)$$

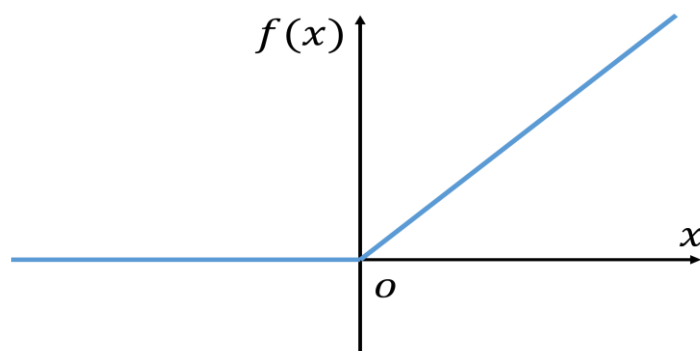


図 7. ReLu 活性化関数

3D-SRGAN の Generator では 3 次元畳み込み層を利用している。2 次元の畳み込み層でカーネルのサイズが $k \times k$ の畳み込み処理をしていた場合、3 次元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。

次にそれぞれの層の処理を説明する。まず、入力層では、 $C_{in}=1$ 、 $C_{out}=64$ 、 $k=9$ の 3 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 3 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 3 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。また、5 つの Residual Block の処理の後、 $C_{in}=64$ 、 $C_{out}=64$ 、 $k=3$ の 3 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。

次の層では、実際に 3 次元モデルのサイズを拡大していく。そのために Pixel Shuffler を 3 次元モデルに拡張した Voxel Shuffler を利用する。Pixel Shuffler はもともと画像を拡大するための手法である。入力の特徴マップを並び替えて画像を拡大することができる。この並び替えの処理を行う関数を \mathcal{PS} とし、入力された特徴マップの高さを H_{in} 、幅を W_{in} 、拡大する倍率を r とすると、式(2)で表すことができる。

$$\mathcal{PS}(T)_{x,y,c} = T_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, C_{out} \cdot r \cdot \text{mod}(y,r) + C_{out} \cdot \text{mod}(x,r) + c} \quad (2)$$

式(2)の T は入力マップ、 c は出力マップのチャンネル位置、 y は出力マップの縦位置、 x は出力マップの横位置を表す。また、 $\text{mod}(a, b)$ は a を b で割った余りを表す。 $C_{in} = C_{out} \cdot r^2$ とすると式(2)によって、入力の特徴マップ $C_{out} \cdot r^2 \times H_{in} \times W_{in}$ は $C_{out} \times r H_{in} \times r W_{in}$ に変形できる。関数 \mathcal{PS} は画像を拡大するための関数であるため、3 次元モデルを拡大するために特徴マップを並び替える関数として Voxel Shuffler を新たに導入する。

$$\mathcal{VS}(T)_{x,y,z,c} = T_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, \lfloor z/r \rfloor, C_{out} \cdot r^2 \cdot \text{mod}(z,r) + C_{out} \cdot r \cdot \text{mod}(y,r) + C_{out} \cdot \text{mod}(x,r) + c} \quad (3)$$

式(3)は Voxel Shuffler の関数であり、 z は出力マップの奥行きを表してい

る。ここで、入力の特徴マップの奥行きを D_{in} 、 $C_{in} = C_{out} \cdot r^3$ とすると、式(3)によって入力の特徴マップ $C_{out} \cdot r^3 \times H_{in} \times W_{in} \times D_{in}$ は $C_{out} \times rH_{in} \times rW_{in} \times rD_{in}$ に変形できる。これによって、3次元モデルを r 倍することができる。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3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。また、3次元畳み込み層の C_{out} はVoxel Shufflerの拡張に伴って、SRGANからパラメータを変更している。

最後に $C_{in}=64$ 、 $C_{out}=1$ 、 $s=1$ 、 $k=9$ の3次元畳み込み層で処理し、高解像度3次元モデルを出力としている。

3D-SRGANのGeneratorのこれらのパラメータは基本的にSRGANと同じものを利用している(図8)。

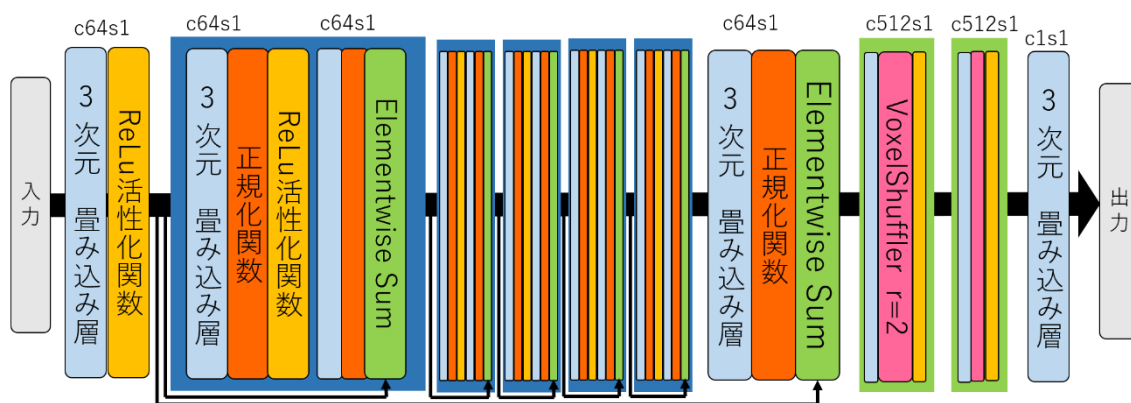


図 8. Generator の構造と各パラメータ

c: 出力チャンネル数、s: ストライド数

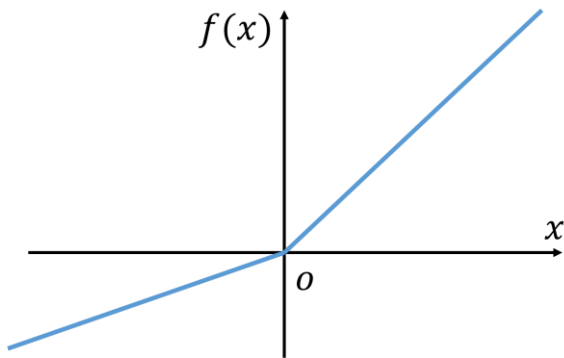


図 9. LeakyReLU 活性化関数

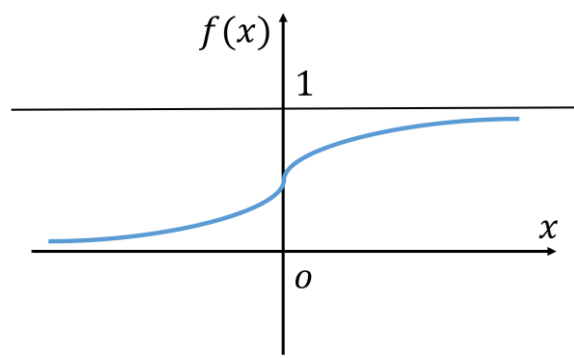


図 10. Sigmoid 活性化関数

3.3. Discriminator の構造

Discriminator の構造は大きく入力層、Block 層、出力層の 3 つに分けられる。入力層では入力の 3 次元モデルから複数のチャンネルを出力し、Block 層で 3 次元モデルの特徴を抽出し、出力層では入力の 3 次元モデルが学習データである確率が出力される。

3 D-SRGAN の Discriminator でも Generator と同様に 3 次元畳み込み層を利用し、3 次元モデルの畳み込み処理を行っている。

Discriminator では、活性化関数として Leaky ReLU 活性化関数(図 9)と Sigmoid 活性化関数(図 10)を使用しており、それぞれ式(4)(5)と表現できる。

$$f(x) = \max(\alpha x, x), 0 < \alpha < 1 \quad (4)$$

$$f(x) = \frac{1}{1 + \exp(-\beta x)}, 0 < \beta \quad (5)$$

次にそれぞれの層の処理を説明する。まず、最初の層では、 $C_{in} = 1, C_{out} = 64$ 、 $s = 1$ の 3 次元畳み込み層、Leaky ReLU 活性化関数の順に処理している。次の Block 層は 7 つのブロックで構成されている。1 つのブロックの中身は、3 次元畳み込み層、正規化関数、Leaky ReLU 活性化関数の順に処理している。また、それぞれの Block の 3 次元畳み込み層のパラメータは層毎に順に $C_{in} = 64, 64, 128, 128, 256, 256, C_{out} = 64, 128, 128, 256, 256, 512, 512, s = 2, 1, 2, 1, 2$ となっている。3 次元畳み込み層のカーネルサイズは全て $k = 3$ である。

7つのブロックの処理後、 $C_{out} = 1024$ の全結合、Leaky ReLU 活性化関数、 $C_{out} = 1$ の全結合、Sigmoid 活性化関数を順番に処理している。最終的な出力の値は[0,1]の間の値となる。この値が Discriminator の入力学習データである確率を表している。

3D-SRGAN の Discriminator のこれらのパラメータは、SRGAN と同じものを利用している(図 11)。

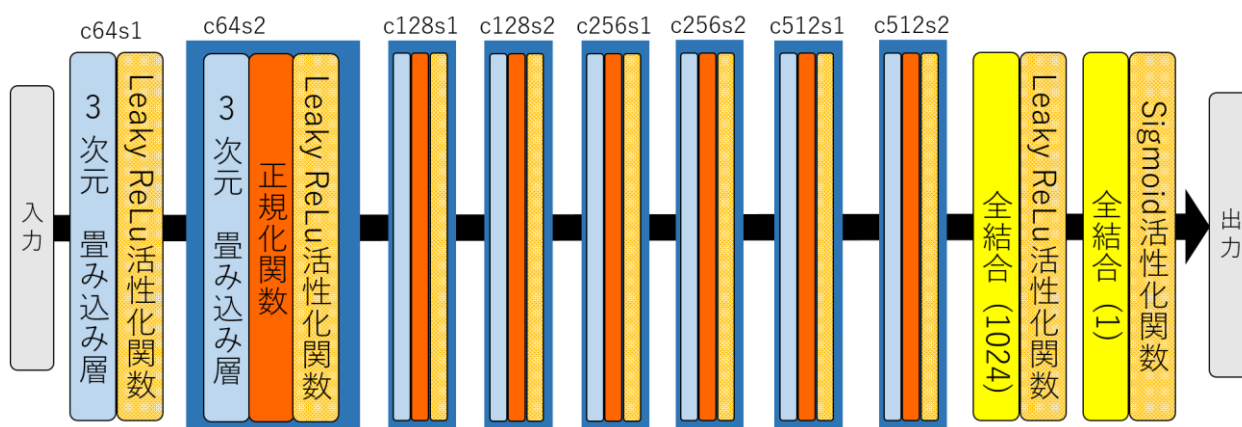


図 11. Discriminator の構造と各パラメータ

c: 出力チャンネル数、s: ストライド数

3.4. 3D-SRGAN の学習方法

3D-SRGAN の学習の進め方は基本的に SRGAN と同じである。学習は目的関数 $V(D, G)$ に従って進む。

$$\min_G \max_D V(D, G) = \mathbb{E}_{I^{HR} \sim p_{train}(I^{HR})} [\log D(I^{HR})] + \mathbb{E}_{I^{LR} \sim p_G(I^{LR})} [\log(1 - D(G(I^{LR})))] \quad (6)$$

式(6)の G 、 D はそれぞれ Generator、Discriminator のネットワークを表し、 I が入力データ、 HR 、 LR はそれぞれ高解像 3 次元モデル、低解像 3 次元モデルを表している。また、 $D(x)$ は入力データ x が学習データである確率を表している。そのため、Discriminator がうまく識別できるようになると $D(I^{HR})$ が大きくなり、 $\log D(I^{HR})$ も大きくなる。そして、生成データだと見破られてしまうため $D(G(I^{LR}))$ が小さくなり、 $\log(1 - D(G(I^{LR})))$ は大きくなる。一方、Generator が学習データと似ているデータを生成できるようになると、Discriminator の識別結果が悪くなり、 $D(G(I^{LR}))$ が大きく、 $\log(1 - D(G(I^{LR})))$ が小さくなる。そのため、Generator は $\log(1 - D(G(I^{LR})))$ を最小化しようとし、Discriminator は $D(I^{HR})$ を最大化しようとして学習を進めていく。

学習では Generator と Discriminator がそれぞれ後述する Loss 関数で求めた Loss が小さくなるように交互に更新を行っている。

3.4.1. Generator の学習更新

Generator の loss 関数 l_G^{3DSR} は式(7)で表現できる。

$$l_G^{3DSR} = l_{con}^{3DSR} + 10^{-3} l_{Gen}^{3DSR} \quad (7)$$

$$l_{con}^{3DSR} = \frac{1}{r^3 WHD} \sum_{x=1}^{rW} \sum_{y=1}^{rH} \sum_{z=1}^{rD} (I_{x,y,z}^{HR} - G(I^{LR})_{x,y,z})^2 \quad (8)$$

$$l_{Gen}^{3DSR} = \sum_{n=1}^N -\log D(G(I^{LR})) \quad (9)$$

式(7)の l_{con}^{3DSR} は Content Loss、 l_{Gen}^{3DSR} は Adversarial Loss を表している。

Content Loss は、式(8)で計算される。式(8)では、学習データの高解像度3次元モデル I^{HR} と Generator が生成した高解像度3次元モデル $G(I^{LR})$ の平均二乗誤差を計算している。Content Loss には、元のデータの見た目をある程度そのままにして、別の操作を加えることができる特徴がある。そのため、超解像しても元のデータの見た目から大きく変化する心配がない。

Adversarial Loss は、式(9)で計算される。式(9)では、 $G(I^{LR})$ が Discriminator で識別された結果を計算している。Adversarial Loss は、GAN で通常使われている Loss 関数で、学習データと見分けがつかないように学習をする特徴がある。Content Loss だけでは、見た目の変化の問題が解決するだけで、はっきりとした3次元モデルが生成されず、Adversarial Loss だけでは学習データの性質を保持しているが、入力とは別の3次元モデルが生成されてしまう。そのため、この2つを適切に組み合わせることで超解像を行っている（式(7)では $1:10^{-3}$ ）。

3.4.2. Discriminator の学習更新

Discriminator の Loss 関数 l_D^{3DSR} は式(10)で表現できる。

$$l_D^{3DSR} = \frac{1}{m} \sum_{i=1}^m \left[\log D(I^{HR}) - \log(1 - D(G(I^{LR}))) \right] \quad (10)$$

$D(x)$ は Discriminator の出力であり、入力データ x が学習データの 3次元モデルである確率を表している。そのため、Discriminator の識別がうまくいくと $D(I^{HR})$ の値が大きくなり、 $D(G(I^{LR}))$ の値は小さくなる。

4. 実験

4.1. 信号処理的なアプローチとの比較

4.1.1. 比較手法

実験では、比較手法として画像処理の最近傍補間(nearest neighbor)[31]と線形補間(bi-linear)[32]を3次元モデルに拡張したものをを用いる。これらの手法は画像の輝度値の局所的な情報を利用してデータを補間している。本研究では、補間での3次元モデルの拡大は1回の処理で2倍拡大とする。

画像の最近傍補間は、最も近い位置にある画素の輝度値を参照して補間する手法である。求める画素間の座標 (x, y) の位置の輝度値を $Dst(x, y)$ とし、元の画像の輝度値を $Src(i, j)$ とすると、式(11)で表現できる。

$$Dst(x, y) = Src([x + 0.5], [y + 0.5]) \quad (11)$$

ただし、式(11)の[]は小数点以下切り捨てを表す。

3次元モデルで最近傍補間を行う。求めるボクセル間の座標 (x, y, z) の位置のボクセル値を $Vox_{HR}(x, y, z)$ とし、元の3次元モデルのボクセル値を $Vox_{LR}(x', y', z')$ とする。本研究では、2倍拡大であるため、式(12)で表現できる。

$$Vox_{HR}(2x + ix, 2y + iy, 2z + iz) = Vox_{LR}(x, y, z) \\ ix, iy, iz \in \{0, 1\} \quad (12)$$

式(12)では、元と3次元モデルの1ボクセルを8ボクセルに拡大して補間している。

線形補間は、画像の輝度値を近傍画素値の内分比により直線的に補間する手法である。線形補間では、求める座標 (x, y) の周辺の 2×2 画素(4画素)を使う。求める画素間の座標 (x, y) の位置の輝度値を $Dst(x, y)$ とし、座標 (x, y) の周辺の4画素の輝度値を $Src(x', y'), Src(x' + 1, y'), Src(x', y' + 1), Src(x' + 1, y' + 1)$ とすると、式(13)で表現できる。

$$\begin{aligned}
 Dst(x,y) &= (1 - dx)\{(1 - dy)Src(x',y') + dySrc(x',y' + 1)\} \\
 &\quad + dx\{(1 - dy)Src(x' + 1,y') + dySrc(x' + 1,y' + 1)\} \quad (13) \\
 dx &= x - x', \quad dy = y - y'
 \end{aligned}$$

線形補間は最近傍補間よりも滑らかな画像を生成することができる。しかし、この手法では輝度値を計算して補間しているため、2値データである3次元モデルの拡大を行うために工夫する必要がある。そこで、3次元モデルの求める座標 (x,y,z) の周辺の $2 \times 2 \times 2$ ボクセル(8ボクセル)のパターンによって補間する値を決める。図12に8ボクセルのパターンによる補間方法を示す。図12では、白いマスは値0、黒いマスは値1を表す。また、灰色のマスの値は0と1のどちらでも良いとする。2マスのパターンでは、AB、AC、AEのパターン、4マスのパターンでは、ABCD、ABEF、ACEGの面のパターン、8マスのパターンでは、ABCDEFGHのパターンを示す。また、2マスのパターンでa、b、d、4マスのパターンでc、e、f、8マスのパターンでgをそれぞれ補間する。図のパターンの時に値1、それ以外のパターンの時に値0を補間する。

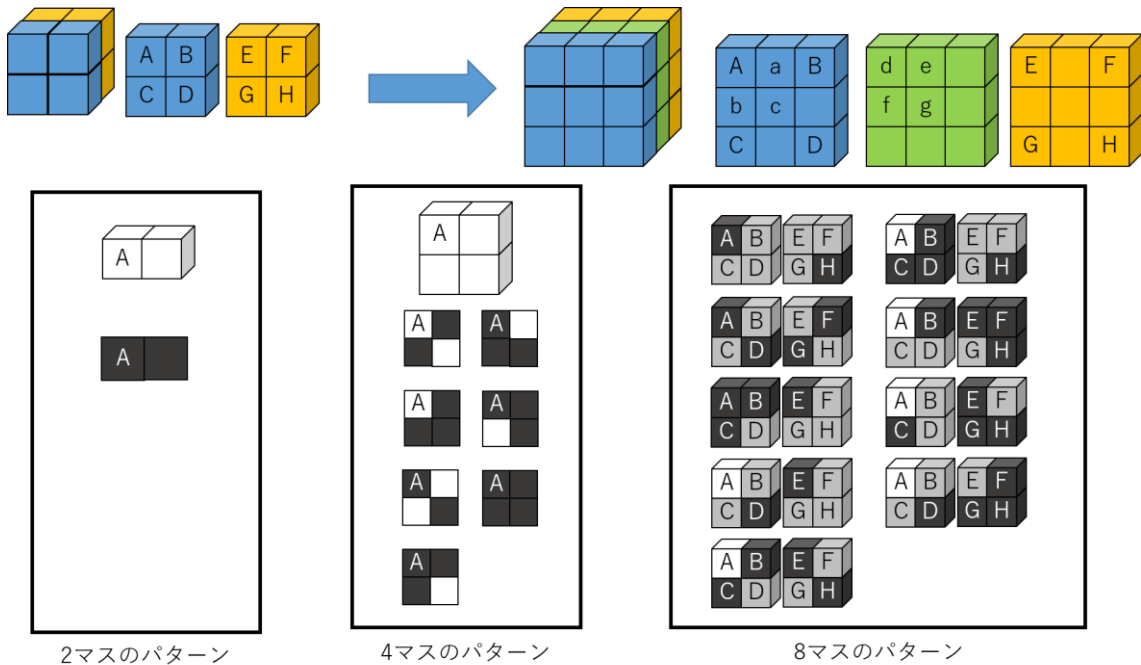


図 12. 3次元モデルの線形補間を利用した超解像

表 1. データセット ModelNet10

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

4.1.2. 実験方法

この実験では 4.1.1 節の 3 次元モデルの補間に拡張した最近傍補間と線形補間、提案手法である 3D-SRGAN の生成結果の比較を行った。

本研究では、3 次元モデルとしてデータセット ModelNet10[33]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 3 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。

学習前の準備として、学習データとテストデータをそれぞれ、ボクセル数が $16 \times 16 \times 16$ と $64 \times 64 \times 64$ の 3 次元モデルに変換した。前者を低解像度 3 次元モデル、後者を高解像度 3 次元モデルと呼ぶ。3D-SRGAN で chair の学習データの低解像度 3 次元モデルと高解像度 3 次元モデルのペアを学習させた。1000 回学習させ、最も適切な超解像を行う生成モデルを手作業で選択した。

それぞれの手法で超解像を行い生成した 3 次元モデルを超解像 3 次元モデルと呼び、データセットの高解像度 3 次元モデルと区別する。chair のテストデータの高解像度 3 次元モデルと超解像 3 次元モデルの同じ座標間でのボクセルの有無の違いを誤差としてそれぞれ計算し、比較した。また、主観的な評価として、それぞれの超解像 3 次元モデルを同じ角度から見た結果を画像として見比べた。

4.1.3. 結果・考察

表 2 に超解像 3 次元モデルと高解像度 3 次元モデルの誤差の平均を示す。小数点以下を四捨五入して示している。また、それぞれの手法で chair のテストデータを超解像した結果を図 13 に示す。さらに、図 14 にそれぞれの手法での超解像 3 次元モデルと高解像度 3 次元モデルのデータ毎の誤差を示す。

まず、表 2 の結果から、3D-SRGAN の生成結果が最も高解像度 3 次元モデルとの誤差が少ない。また、最近傍補間と線形補間では線形補間の方が誤差は少ないが、3D-SRGAN と線形補間の差は大きく、3D-SRGAN がより高解像度 3 次元モデルに近いことが分かる。

図 13 からは主観的に見ると、高解像度 3 次元モデルに一番近いのは 3D-SRGAN であり、次に線形補間であった。高解像度 3 次元モデルとそれぞれを見比べると、直線や平面ではどれも形が整っている。しかし、曲線や曲面では、最近傍補間は階段状の角ばったような形状になっている。線形補間では最近傍補間よりも滑らかであるが、多少角ばった部分もある。一方、3D-SRGAN では、曲線や曲面がきれいに生成されており、より高解像度 3 次元モデルに近づいている。しかし、3D-SRGAN の生成結果の中にも、図 13 の上から 3 番目の椅子の背もたれでは穴の部分を埋めているように、形が少し崩れている 3 次元モデルもあった。

図 14 は、横軸が各超解像 3 次元モデルを表し、縦軸がその超解像 3 次元モデルと高解像度 3 次元モデルの誤差を表す。横軸は 3 D-SRGAN の誤差の昇順にソートして並べている。グラフの右の方で 3 D-SRGAN が線形補間より誤差が大きい箇所もあるが、ほとんどの 3 次元モデルで 3 D-SRGAN の誤差が最も少ない。

表 2. 3つの手法の超解像 3次元モデルと高解像度 3次元モデルの誤差の平均

	最近傍補間	線形補間	3D-SRGAN
誤差の平均	22851	15374	7512



図 13. 最近傍補間、線形補間、3D-SRGAN による 3次元モデル超解像

左から順に最近傍補間、線形補間 3次元モデル、3D-SRGAN によって生成した超解像 3次元モデル、高解像度 3次元モデル。

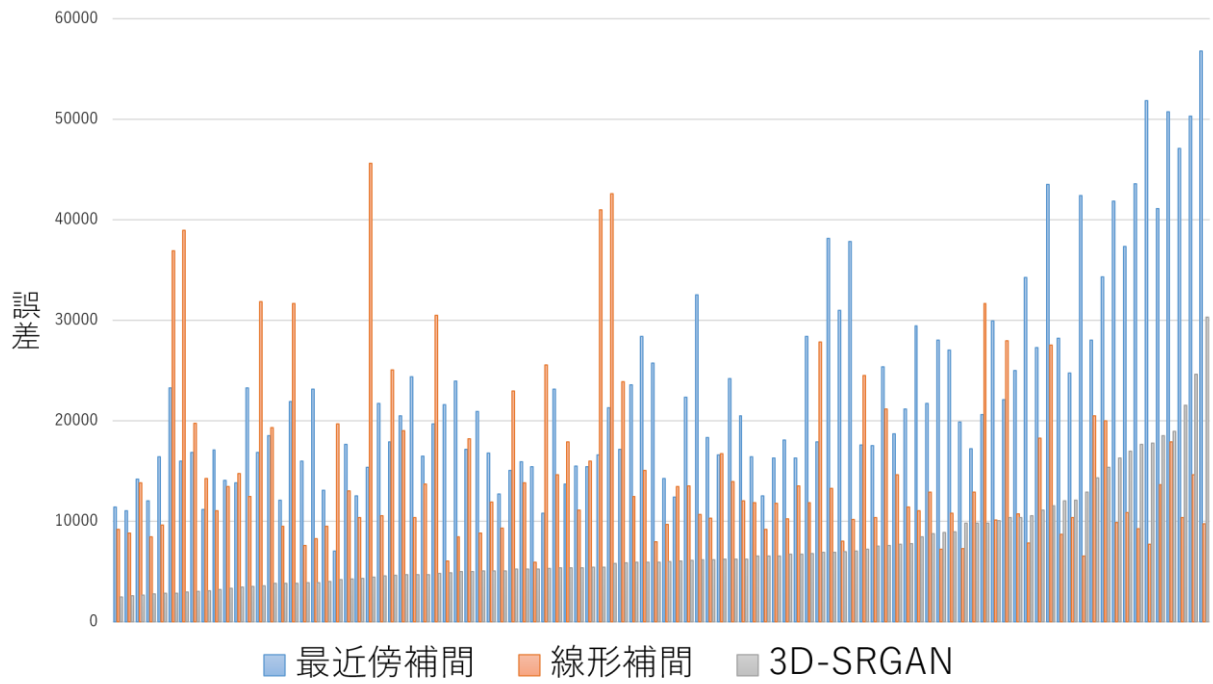


図 14. 3つの手法の超解像3次元モデルと高解像度3次元モデルの誤差

4.2. 学習データとは別クラスでの超解像

4.2.1. 実験方法

前節の実験では、chair を学習し、chair の超解像を行った。本節の実験では、3D-SRGAN に用いた学習データのクラスとは異なるクラスの3次元モデルの超解像を行い、学習データが超解像結果にどのような影響があるかを調査した。また、比較には4.1.1節の比較手法を用いた。

3D-SRGAN の生成モデルは前節と同じ chair の学習データを学習したものをを用いた。超解像を行った3次元モデルは、データセット ModelNet10 の chair 以外の9クラスの低解像度3次元モデルである。

4.2.2. 結果・考察

表3に超解像3次元モデルと高解像度3次元モデルの誤差の平均を示す。また、それぞれのクラスの生成結果を図15、16に示す。

表3の結果を見ると、3D-SRGAN が最近傍補間や線形補間よりも誤差が小さくなっている。

次に図15、16を見てみると、ほとんどの結果で超解像3次元モデルが高解像度3次元モデルに近いものとなっている。また、最近傍補間や線形補間よりもかなり滑らかな3次元モデルが生成されていることが分かる。また、table は他のクラスよりも自然な生成結果が見られる。表3でも、前節の chair よりも誤差が少なくなっている。table の生成結果が良くなっている理由として、chair と table の形状が似ていたため、超解像の精度も高くなったと考えられる。しかし、3D-SRGAN の超解像3次元モデルの中には形が少し歪なものもみられる。この原因は学習データの影響を受けているのだと考えられる。

このような結果から、3D-SRGAN では学習データが超解像結果に多少の影響を与えるが、学習データと異なるクラスの超解像でも生成精度が高いことが

分かった。また、学習データに形状が似ている3次元モデルを超解像すると精度が高くなる傾向があると言える。

表 3. 各クラスの超解像3次元モデルと高解像度3次元モデルの誤差の平均

クラス名	bathtub	bed	desk	dresser	monitor
最近傍補間	23073	28399	22205	39910	24647
線形補間	16449	21178	16526	27980	17620
3D-SRGAN	10323	15442	11972	22827	12313
クラス名	night_stand	sofa	table	toilet	
最近傍補間	44388	23528	14937	35537	
線形補間	34636	17278	9849	26427	
3D-SRGAN	24907	12550	7092	16281	



図 15. SRGAN による別クラスの超解像(1)

左から順に最近傍補間、線形補間 3 次元モデル、3 D-SRGAN によって生成した超解像 3 次元モデル、高解像度 3 次元モデル。



図 16. SRGAN による別クラスの超解像(2)

左から順に最近傍補間、線形補間 3 次元モデル、3 D-SRGAN によって生成した超解像 3 次元モデル、高解像度 3 次元モデル。

4.3. 局所的情報を用いた超解像

4.3.1. 実験方法

4.1 節と 4.2 節で扱っていた 3 D-SRGAN では 3 次元モデルの全体の情報を用いて学習を行っていた。そこで 3 次元モデルの局所的情報を用いて学習をした 3 D-SRGAN で超解像を行い、全体的情報と局所的情報ではどちらの学習の方が超解像の精度が高くなるか調査した。

局所的情報を扱う 3 D-SRGAN は、学習で 3 次元モデルを入力する際に、3 次元モデルの一部を切り取って入力とする。この切り取り処理をクロップと呼ぶ。16 × 16 × 16 の低解像度 3 次元モデルから 4 × 4 × 4 と 8 × 8 × 8 の 3 次元モデルをそれぞれクロップする。64 × 64 × 64 の高解像度 3 次元モデルからは低解像度 3 次元モデルでクロップした部分に対応する部分 16 × 16 × 16 と 24 × 24 × 24 をクロップする。4 × 4 × 4 と 16 × 16 × 16 の 3 次元モデルのペアと 8 × 8 × 8 と 24 × 24 × 24 の 3 次元モデルのペアをそれぞれ 3 D-SRGAN で学習する。前者を学習した 3 D-SRGAN の生成結果を 4 -超解像 3 次元モデル、後者で学習した 3 D-SRGAN の生成結果を 8 -超解像 3 次元モデルと呼ぶ。学習データには、4.1 節と同じ chair の 3 次元モデルを利用した。

4.3.2. 結果・考察

表 4 にそれぞれの超解像 3 次元モデルと高解像度 3 次元モデルの誤差の平均を示す。また、生成結果を図 17 に示す。

表 4 の結果から、クロップありの生成結果はどちらもクロップなしの生成結果より誤差が大きくなっていた。また、3 D-SRGAN の中では 4 -超解像 3 次元モデルの結果が最も誤差が大きかった。

図 17 の結果から、クロップありの生成結果では、平面や曲面に穴が空いているものが複数存在していた。また、クロップなしの超解像 3 次元モデルと見比べ

ると、本来つながっている部分が途切れていたり、本来存在しない座標にボクセルが生成されていたりして、形が崩れているものが多い。特に4-超解像3次元モデルでは、形の崩れているものが他の生成結果よりも多く見られた。

これらの結果から、3D-SRGANでは局所的な情報を用いるよりも全体的な情報を用いて学習をした方が精度の高い超解像を行えた。3D-SRGANは全体的な情報によって、より自然な超解像が行えていることが分かった。

表 4. クロップあり・なしの3D-SRGANでの超解像3次元モデルと

高解像度3次元モデルの誤差の平均

	4-超解像3次元モデル 3D-SRGAN	8-超解像3次元モデル 3D-SRGAN	クロップなし 3D-SRGAN
誤差の平均	8707	8457	7512
	最近傍補間	線形補間	
誤差の平均	22851	15374	



図 17. クロップありとクロップなしの 3D-SRGAN による超解像

左から順に、4-超解像 3 次元モデル、8-超解像 3 次元モデル、超解像 3 次元モデル、高解像度 3 次元モデル。

5. おわりに

本論文では、3次元モデルの超解像を行う3D-SRGANを提案した。3D-SRGANでは低解像度3次元モデルから高解像度3次元モデルを得ることができた。画像超解像の手法として用いられていた最近傍補間と線形補間を3次元モデルに拡張して、3D-SRGANとの超解像の生成結果の比較を行った。その結果、3D-SRGANが最も精度の高い3次元モデルを生成できた。また、3D-SRGANは学習データと異なるクラスの超解像では、生成結果に多少影響を与えるが、比較手法より精度の高い超解像3次元モデルを生成できることが分かった。また、クロップありとなしの3D-SRGANで超解像の生成結果の比較を行った。その結果、クロップなしの学習の方がより正解に近い3次元モデルを生成できた。このことから3D-SRGANでは、3次元モデル全体の情報を学習し、より自然な超解像が行えることが分かった。

今後の課題として、2つの課題が挙げられる。

1つ目は、3D-SRGANの学習データによる影響の調査である。本研究の実験では1クラスのデータで学習した。その結果、他クラスの生成結果では、同じクラスの生成結果よりも悪いものとなっていた。この原因は学習データの影響だと考えられる。そのため、学習データを複数のクラスで行ったり、数を増やしてみたりして学習データの影響を調査することが課題の1つに挙がる。

2つ目は、3D-SRGANの細かい設定の見直しである。本研究の3D-SRGANはSRGANを拡張した際に、必要最低限の変更しかしていない。そのため、GeneratorやDiscriminatorのパラメータや構造を、3D-SRGANに最適化することが課題として挙がる。

謝辞

本研究を行うにあたり、多くの方々に協力をしていただきました。大変お世話になりましたことをこの場を借りてお礼申し上げます。

指導教員である椋木雅之教授には、論文作成や研究に関して様々なご助言やご指導していただきました。ゼミなどでの質問にも対応していただきました。3年間、本当にありがとうございました。

伊達准教授、坂本准教授には、副査を務めていただきました。お忙しい中、時間を割いて下さりましてありがとうございました。

高塚佳代子技術職員には、ゼミを通して多くのご助言をいただきました。大変感謝しております。

椋木研究室の皆様には、研究を進める中で様々な助言をいただきました。楽しく、充実した研究室生活を過ごすことができました。大変感謝しています。

参考文献

- [1] 田村秀行, “コンピュータ画像処理”, オーム社, 9章 (2002)
- [2] blender.org - Home of the Blender project - Free and Open 3D Creation Software, <https://www.blender.org/>
- [3] Fusion 360, <https://www.autodesk.co.jp/>
- [4] Minecraft, <https://minecraft.net/en-us/>
- [5] 安藤繁, 土井康弘, “超解像”, 計測と制御, vol. 22, no. 10, p. 828-836 (1983)
- [6] 池谷彰彦, 広明敏彦, “超解像ソリューション (組込みソフトウェア・ソリューション特集)--(イメージ/音声処理コンポーネントソリューション)”, NEC 技報, vol. 60, no. 2, pp. 24-26 (2007)
- [7] 清水雅夫, 矢野高宏, 奥富正敏, “画像変形を表す N パラメータの高精度同時推定法と超解像への応用”, 情報処理学会論文誌コンピュータビジョンとイメージメディア(CVIM), vol. 45, pp. 93-98 (2004)
- [8] 田中正行, 奥富正敏, “画素数の壁を打ち破る 複数画像からの超解像技術”, 映像情報メディア学会誌, vol. 62, no. 3, pp. 337-342 (2008)
- [9] A. M. Tekalp, M. K. Ozkan and M. I. Sezan, "High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration", IEEE International Conference on Acoustics, Speech, and Single Processiong (ICASSP), vol. 3, pp. 169-172 (1992)
- [10] 後藤知将, 奥富正敏, “画像復元とレジストレーションの同時最適化の実験的検証”, 信学論誌, vol. 6, pp. 1632-1635 (2007)
- [11] 小松隆, 斎藤隆弘, “TV 正則化法を用いた JPEG 圧縮画像の超解像デコーディング”, 信学論誌, vol. 7, pp. 1671-1674 (2007)
- [12] W. T. Freeman, T. R. Jones and E. C. Pasztor, "Example-based super-resolution", IEEE Computer Graphics and Applications, vol. 22, no. 2, pp. 56-

65 (2002)

- [13] Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, Thomas Huang, “Deep Networks for Image Super-Resolution With Sparse Prior”, IEEE International Conference on Computer Vision (ICCV), pp. 370-378 (2015)
- [14] Zhen Cui, Hong Chang, Shiguang Shan, Bineng Zhong, Xilin Chen, “Deep Network Cascade for Image Super-resolution”, European Conference on Computer Vision (ECCV), pp. 49-64 (2014)
- [15] 岡谷貴之, “機械学習プロフェッショナルシリーズ 深層学習”, 講談社, 6章 (2015)
- [16] C. Dong, C. C. Loy, K. He and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 38, no. 2, pp. 295-307 (2016)
- [17] Y. Tai, J. Yang and X. Liu, "Image Super-Resolution via Deep Recursive Residual Network", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2790-2798 (2017)
- [18] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 105-114 (2017)
- [19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets”, Advances in Neural Information Processing Systems (NIPS), pp. 2672–2680 (2014)
- [20] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, “Image Style Transfer Using Convolutional Neural Networks”, IEEE Conference on

- Computer Vision and Pattern Recognition (CVPR), pp. 2414-2423 (2016)
- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778 (2016)
- [22] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, Zehan Wang, “Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1874-1883 (2016)
- [23] Kaiming He, Jian Sun, “Convolutional Neural Networks at Constrained Time Cost”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5353-5360 (2015)
- [24] Rupesh Kumar Srivastava, Klaus Greff, Jürgen Schmidhuber, “Highway Networks”, arXiv:1505.00387 (2015)
- [25] Odena, Augustus and Dumoulin, Vincent and Olah, Chris, “Deconvolution and Checkerboard Artifacts”, Distill (2016)
<http://doi.org/10.23915/distill.00003>
- [26] RAISE - The Raw Images Dataset, <http://loki.disi.unitn.it/RAISE/>
- [27] 查万志, Hyongdoo Jang, 宍戸英彦, 亀田能成, 北原格, “岩石の多視点画像から推定した3次元点群のクラスタリング”, 第80回全国大会講演論文集, vol. 2018, no. 1, pp. 475-476 (2018)
- [28] 山下淳, 原田知明, 金子透, “全方位カメラ搭載移動ロボットによる Structure from Motion を用いた3次元環境モデリング”, 日本機械学会論文集 C 編, vol. 73, no. 726, pp. 512-519 (2007)
- [29] 武井毅, 大淵竜太郎, “多重解像度表現を用いたポリゴンスープ三次元モ

- デルの形状類似比較”, 画像電子学会誌, vol. 34, no. 4, pp. 356-369 (2005)
- [30] A. Radford, L. Metz, S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks", International Conference on Learning Representations (ICLR) (2016)
- [31] Belur V. Dasarathy, "Nearest neighbor (NN) norms: nn pattern classification techniques", IEEE, p. 447 (1991)
- [32] 美濃導彦, “画像処理論-Web 情報理解のための基礎知識-”, オーム社, 5章 (2014)
- [33] ModelNet10, <http://modelnet.cs.princeton.edu/>