Vol.2022-CVIM-229 No.25
2022/3/10

点群からの表面再構成のための符号付き距離関数の学習におけるトポロジカルな制約の利用

中島 直道[†1,a)]　古川 遼[†2,b)]

概要：近年，点群データからの 3 次元物体の表面の再構成を行う際に，表面を表現する陰関数の一つである物体の符号付き距離関数 (SDF) をニューラルネットワークで近似する手法が盛んに研究されている． 多くのニューラルネットワークによる近似手法における課題の一つは，表面の大域的な幾何的性質が望ましいものになる保証がないことである．本論文では，向き情報を持たない点群から物体の表面再構成を行うタスクにおいて，表面の大域的かつトポロジカルな性質である連結性の制約を損失関数を通して導入し，ニューラルネットワークで符号付き距離関数を追加学習する方法を提案する．従来の連結性の制約を課す損失関数を用いた形状学習・修正方法に比べて，提案手法ではより直接的かつ効率的に制約を反映する工夫を行う．我々は，この手法を 3 次元の形状データセットを用いて実験し，再構成された表面の連結性と元の点群との差異を調査した．

キーワード：点群，表面再構成，符号付き距離関数，ニューラルネットワーク，陰関数表現，パーシステントホモロジー

Using topological constraints in learning signed distance functions for surface reconstruction from point clouds

Naomichi Nakajima[†1,a)]　Ryo Furukawa[†2,b)]

Abstract: In recent years, for the surface reconstruction of a 3-dimensional object from point cloud data, methods to approximate the signed distance function (SDF) of an object, which is one of the implicit functions representing the surface, by neural networks have been actively researched. One of the problems in many neural network approximation methods is that there is no guarantee that desired global geometric properties of a surface are satisfied. In this paper, in the task of a surface reconstruction of an object from a point cloud without normal information, we propose a method for additionally learning the signed distance function with a neural network by introducing the connectivity constraint, which is a global and topological property of a surface, through a loss function. Compared with the conventional shape learning and modification methods using loss functions that impose connectivity constraints, the proposed method devises a more direct and efficient way to reflect the constraint. We have experimented with this method on 3-dimensional shape datasets to investigate the connectivity of the reconstructed surface and the difference from the original point cloud.

Keywords: point cloud, surface reconstruction, signed distance function, neural network, implicit function, persistent homology

[†1] 現在，北海道大学大学院情報科学院
Presently with Hokkaido University
[†2] 現在，株式会社 ALBERT
Presently with ALBERT Inc.
[a)] nakajima-n@ist.hokudai.ac.jp
[b)] ryo_furukawa@albert2005.co.jp

1. はじめに

近年，点群からの 3 次元物体の表面再構成において，ニューラルネットワーク（NN）による陰関数表現を用いる手法が盛んに研究されている．陰関数表現を用いた 3 次元物体の表面再構成では，物体表面は NN で近似された 3 次元空間上の関数 $f_\theta\colon \mathbb{R}^3 \to \mathbb{R}$ のゼロレベルセットとして表現される：$S = \{x \in \mathbb{R}^3 \mid f_\theta(x) = 0\}$. NN を用いた陰関数表現を用いることで，従来のボクセル，メッシュ，octree を使った場合と異なり，解像度によらない表現を作ることが可能になる．本研究では，陰関数として，特に，物体の外部・内部で各々正・負の値をとる符号付き距離関数（signed distance function, 以下 SDF と呼ぶ）を考える．

SDF の学習方法として符号付き距離を教師データとして回帰を行うものもあるが，一般には距離の符号を定義する物体の内外や向き情報を観測点群から得ることは困難である．そこで，向き情報を持たない点群から SDF を学習する方法が提案されている [2], [3], [4], [11], [19]．NN を使った表面再構成では細い部分や薄い部分の再現が難しく，本来は繋がっている箇所が分断して再構成されてしまうといった場合も報告されている（例えば， [2]）．これらの問題に対して，我々は物体表面の大域的な幾何的性質の制約や事前知識を与えて学習することを考える．

本研究の目標は，一つの物体表面から得られる向き情報を持たない点群に対して，大域的かつトポロジカルな制約である物体表面の連結性を満たすように，SDF を用いた表面再構成を行うことである．そのために，我々は，訓練途中の表面からサンプリングされた点群に対して損失関数を通して連結性の制約をかけ，NN のパラメータを更新する追加学習方法を提案する．データ点や関数のトポロジカルな情報を取り出す道具としてパーシステントホモロジーがあり，理論・実用の両面において発展している．3 次元形状に関わるタスクにおいても，物体の連結性などのトポロジカルな制約をかけるために応用されている [5], [9], [10], [15], [17] が，その使われ方は限定的である．本研究では，サンプリングされた点群のパーシステントホモロジーを用いて，表面の連結成分同士が大きく離れると考えられる部分を特定し，その周辺点群同士を比較する損失関数項を導入することで，物体表面の連結性に制約をかける．また，教師点群とサンプリングされた点群を集合として比較する損失関数項も導入する．これにより，連結性の制約をかけつつ再構成表面の質を大きく低下させない学習を実現する．損失関数の値からの逆伝播で NN のパラメータを更新するためには，損失関数が NN のパラメータに関して微分可能である必要がある．本研究では，この微分可能性を持たせるために，陰関数のレベルセットからサンプリングした点の座標を NN のパラメータに関して微分可能にする手法 [1] を用いる．

我々は，提案手法を検証するために，ShapeNet データセット [6] のメッシュからサンプリングされた点群を用いて，点群からの表面再構成において，ベースモデルを追加学習する実験を行った．再構成表面の連結性や元点群との差異について評価を実施した結果，提案手法が実験データに対して定性的・定量的に有効であることが確認された．

本論文の貢献は次の通りである：

- 物体の大域的かつトポロジカルな性質である連結性の制約を損失関数を通して導入し，SDF を学習する方法を提案する．
- 教師点群と再構成表面からの点群の類似性も考慮し，連結性の制約を与えつつ再構成表面の質を担保する．

2. 関連研究

2.1 陰関数表現を用いた向き情報なしの点群からの表面再構成

SDF の教師情報を用いた NN の陰関数表現による研究が行われる中，[2] では，向き情報を持たない点群などの幾何学的データから形状の陰関数表現を学習する方法 sign agnostic learning（SAL）を提案した．以降も，損失関数に微分情報も含む SAL の一般化 [3]，単位勾配の制約の損失関数項の導入 [11]，SDF とその勾配を用いた表面へ近づき方の学習 [4]，implicit moving least-square function で得られる SDF を併用する自己教師あり学習 [19] の研究がある．SAL [2] では，陰関数を表現する NN（MLP）の初期値が球面の SDF を近似するような重みの初期化（幾何的初期化）も提案した．この初期化は向き情報のない点群からの NN の陰関数表現において重要な役割を果たす．

本研究では，一つの向き情報なしの点群に対して SDF をオーバーフィットさせるタスクにおいて，基本的なモデルである SAL [2] のネットワーク構造・損失関数を用いてベースモデルの学習を行い，その後提案手法の損失関数を用いて追加学習を行う実験を行った．

2.2 パーシステントホモロジーによる 3 次元形状の制約

データ点や特徴量からのパーシステントホモロジーの計算を微分可能にし，パーシステントホモロジーから計算される損失関数を用いてモデルのパラメータを更新する手法が，様々なタスクにおいて用いられている．

NN とは限らない 3 次元形状に関わる機械学習では，パーシステント図に合わせた点群の変形 [10], 形状マッチングのための関数最適化 [17]，点群からの表面再構成 [5] に用いられている．特に，[5] では，パーシステント図から計算される損失関数を用いて各点周りの共分散行列をパラメータとする尤度関数の最適化を行った．[9] では，より一般の機械学習・NN に適用可能な層を提案し，正則化・生成モデル・敵対的攻撃に応用した．[15] では，NN を用いた 3 次

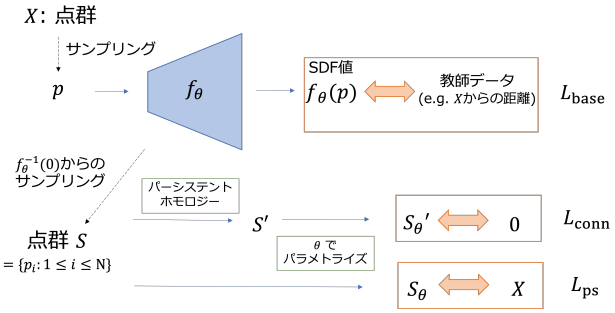


図 1 手法概要.
Fig. 1 Outline of the method.

元形状の生成・自己符号化・修正のタスクで，[9] の層を用いた連結性の制約を物理的安定性の制約とともに用いて，生成モデルの潜在変数を選択する学習方法を導入した．特に，陰関数として occupancy field や部分ごとの SDF を用いたモデル [7], [20] において有効性を検証した．陰関数の NN のパラメータの更新は行っていないことに注意する．近年では，従来のパーシステント図の計算を微分可能にする方法において逆伝播する箇所が一部に限られてしまうことに対処するために，関数最適化において merge tree を用いる方法 [16] や単体複体から作られる近似関数を用いる方法 [18] が提案されている．

本研究では，NN による陰関数表現を用いた SDF による表面再構成おいて，連結性の制約を与えるために [9] を元に点群の連結性が崩れる部分付近を [9] より集中的に取り出す（3.4.1 章）．それによって計算される損失関数項と点群の集合としての近さに制約を与える損失関数項を合わせて用いて，NN のパラメータを直接更新する．

3. 提案手法

3.1 概要

一つの連結な物体の表面からサンプリングされた点群 $X \subset \mathbb{R}^3$ に対し，その物体の SDF を NN $f_\theta \colon \mathbb{R}^3 \to \mathbb{R}$ で近似することを考える．ここで，$\theta \in \mathbb{R}^D$ は NN のパラメータである．学習を行うにあたり様々な損失関数 L_{base} が提案されている．

本手法では，L_{base} における学習の後，二種類の損失関数 L_{conn}, L_{ps} を加えて追加学習を行う（図 1）．これらの二種類の損失関数は，ゼロレベルセット $S = \left\{x \in \mathbb{R}^3 \mid f_\theta(x) = 0\right\}$ からサンプリングされた点群を用いて計算される．L_{conn} は，物体表面の 2 次元閉曲面の大域的な性質である連結性（connectivity）に制約をかけることを目的としたものであり，サンプリング点群のパーシステントホモロジーを用いて計算される（3.4.1 章）．L_{ps} は，教師点群との差異を局所的に捉えることを目的としたものであり，教師点群とサンプリング点群との間の chamfer discrepancy を用いる（3.4.2 章）．追加訓練時の損失関数 L_{total} は重み係数 $\lambda_{\text{conn}}, \lambda_{\text{ps}} \in \mathbb{R}_{\ge 0}$ を用いて次のように表される：

$$L_{\text{total}} := L_{\text{base}} + \lambda_{\text{conn}} L_{\text{conn}} + \lambda_{\text{ps}} L_{\text{ps}}. \tag{1}$$

追加学習の際には，Controlling Neural Level Sets [1] の方法を用いて，ゼロレベルセット S からの各サンプリング点の座標値を NN のパラメータ θ の関数として表す．その上でサンプリング点の座標値から計算される損失関数の値を逆伝播させることで NN f_θ のパラメータを更新する．

3.2 準備

ここでは，本手法で必要となる，パーシステントホモロジー，chamfer discrepancy 及びレベルセットのパラメータ付けの手法について説明する．

3.2.1 パーシステントホモロジー

ここでは，パーシステントホモロジー（詳細は，例えば [8], [21] を参照）とその計算を微分可能にして機械学習に用いる方法（例えば，[9]）について概観する．

ホモロジーは図形（特に，位相空間）を特徴づける代数的道具である．n 次元の図形 $\mathcal{X}$ に対しては，各次元 $k = 0, 1, \ldots, n$ に対してホモロジー群 $H_k(\mathcal{X})$ が定義される．$H_0(\mathcal{X})$ の生成元は $\mathcal{X}$ の連結成分に対応する．パーシステントホモロジーは，単体複体 K とフィルトレーションと呼ばれる K の部分複体の増大列 $(K_r)_{r \in \mathbb{R}_{\ge 0}}$ が与えられた時に，フィルトレーション内のホモロジーの遷移を特徴づける道具である．K が有限単体複体の場合は，フィルトレーションは単体複体 K 上の実数値関数 $F \colon K \to \mathbb{R}$ と考えられる．パーシステントホモロジーの情報は，各次元ごとにフィルトレーション内のホモロジーの生成元が誕生する時刻 birth とその生成元が消滅する時刻 death の組の集合（パーシステント図）として取り出すことができる．ここで，時刻はフィルトレーションの添字に対応する．

birth・death の組の集合 $\{(b_i, d_i)\}_{i \in \mathcal{I}_k}$（$k$ は次元）に対して F を用いて順序を入れることで，各 birth（または，death）に対して単体を対応させることができる:

$$\pi_F(k) \colon \{b_i, d_i\}_{i \in \mathcal{I}_k} \to K. \tag{2}$$

[9] で提案された層（以後，Topology Layer と呼ぶ）では，各 birth（または，death）に対応する単体 σ にパーシステント図から計算される損失関数の逆伝播が行われる：

$$\frac{\partial L}{\partial \sigma} = \sum_{i \in \mathcal{I}_k} \frac{\partial L}{\partial b_i} \mathbb{I}_{\pi_F(k)(b_i) = \sigma} + \sum_{i \in \mathcal{I}_k} \frac{\partial L}{\partial d_i} \mathbb{I}_{\pi_F(k)(d_i) = \sigma}. \tag{3}$$

ここで，$\mathbb{I}_{a=b}$ は $a = b$ の場合は 1 となり，それ以外の場合は 0 となる関数である．

本研究では，点群 X のドロネー図から作られるアルファ複体とその 0 次元（$k = 0$）のパーシステントホモロジーを使う．0 次元のパーシステント図は点群 X の要素数 $|X|$ 個の組 $\{(b_i, d_i)\}_{i=1}^{|X|}$ からなり，全て birth は 0 でありドロ

ネー図の各辺の長さ（及び $+\infty$）が death となる．実験の際は，Topology Layer [9] の公開実装内の AlphaLayer を改良して用いた．

3.2.2 Chamfer discrepancy

点群 $P, Q \subset \mathbb{R}^3$ の chamfer discrepancy とは次で定義される量で，点群間の類似性を測る指標である：

$$d(P,Q) := \frac{1}{|P|}\sum_{p\in P}\min_{q\in Q}||p-q||_2^2 + \frac{1}{|Q|}\sum_{q\in Q}\min_{p\in P}||q-p||_2^2. \tag{4}$$

ここで，$|P|$ は点群 P の点数を表し，$||\cdot||_2$ は L_2 ノルムである．Chamfer discrepancy は距離の公理を満たさないため距離ではないが，点群やメッシュ生成のタスクでは一般的に使われる指標である．

3.2.3 レベルセットのパラメータ付け

Controlling Neural Level Sets [1] では，NN f_θ でパラメータが $\theta = \theta_0$ の場合に表現する関数 $f_{\theta_0}\colon \mathbb{R}^3 \to \mathbb{R}$ と，その関数の正則値 c のレベルセット $S = \{p \in \mathbb{R}^3 \mid f_{\theta_0}(p) = c\}$ がある時に，S 上の点 p を NN のパラメータ θ の関数として表す近似を提案している：

$$p(\theta) \approx p - \frac{\nabla_x f_{\theta_0}(p)}{||\nabla_x f_{\theta_0}(p))||_2^2}\frac{\partial f_\theta(p)}{\partial\theta}(\theta-\theta_0) \tag{5}$$

$$\approx p - \frac{\nabla_x f_{\theta_0}(p)}{||\nabla_x f_{\theta_0}(p))||_2^2}(f_\theta(p)-c). \tag{6}$$

ここで，∇_x は空間変数 $x \in \mathbb{R}^3$ の各成分に関する偏微分を表す．一つ目の近似は c が f_{θ_0} の正則値という仮定の下での陰関数定理を用いた一次近似であり，二つ目の近似は f_θ の $f_{\theta_0}(p) = c$ の周りでの θ に関する一次近似である．

本手法では，訓練途中のゼロレベルセット付近からサンプリングされた点に対して，$c = 0$ として θ によるパラメータ付けを行う．

3.3 ゼロレベルセットのサンプリング

Controlling Neural Level Sets [1] における表面再構成では，ゼロレベルセットのサンプリングを行う際に，SDF を近似する NN の勾配を用いてガウス・ニュートン法

$$p^{\text{next}} = p^{\text{prev}} - \frac{\nabla_x f_{\theta_0}(p^{\text{prev}})}{||\nabla_x f_{\theta_0}(p^{\text{prev}}))||_2^2} f_{\theta_0}(p^{\text{prev}})$$

を繰り返し用いる．

SDF を近似する NN の勾配を用いたガウス・ニュートン法によるサンプリングでは，サンプリング点が物体の角など曲率が大きい部分に偏る現象が見られた．一様でないサンプリングや実際のゼロレベルセットに近くない点を用いて形状の修正を行うことは難しいと考えられる．そこで，本手法では，ゼロレベルセット付近かつ一様にサンプリングするために，マーチングキューブ法 [12] で張られるメッシュからのサンプリングを用いた上で，ガウス・ニュートン法を行う．具体的には，まず，マーチングキューブ法で得られたメッシュからサンプリングを行う．マーチングキューブ法では，関数の符号変化を捉えてメッシュを張るため，メッシュ頂点の配置はマーチングキューブ法を適用する際の空間分割の位置と解像度に依存する．そこで，次に，空間分割の位置と解像度の影響を減らすため，ガウスノイズ $z \sim \mathcal{N}(0, \sigma^2 I)$ を加える．最後に，ノイズが加えられたサンプリング点を初期点としてガウス・ニュートン法を行い，得られた点をゼロレベルセットからのサンプリングの近似 $S := \{p_i\}$ とする．

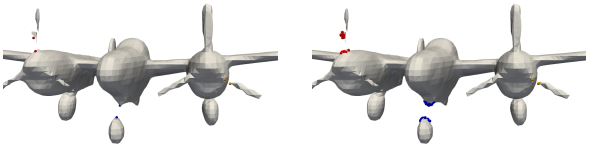


図 2 NN から取り出した物体表面（灰色）と death 対応点付近の損失関数が逆伝播する点集合の組（赤・青・橙色の 3 組）の例．左：Topology Layer の場合．右：本手法の場合．

Fig. 2 Examples of an object surface (gray) out of a NN and pairs of point sets (3 pairs of red, blue, and orange) where the loss function propagates backwards around the death corresponding points. Left: Topology Layer case. Right: In this method.

このようにして得られるゼロレベルセットのサンプリング点群 $S := \{p_i\}$ の各点の座標値を，式 (6) の近似を用いることで，NN のパラメータ θ の関数 $S_\theta := \{p_i(\theta)\}$ として損失関数の計算に用いる．

3.4 損失関数

本提案手法では，ベースとなる既存モデルの損失関数 L_{base} で学習を行なった後，連結性を考慮する L_{conn} と点群同士を比較する L_{ps} を加えた式 (1) の損失関数 L_{total} を用いて追加学習を行う．

3.4.1 連結性のための損失関数項

パーシステント図を用いて，連結性の制約を考えるためには次のようにする．まず，点群の 0 次のパーシステント図 $\{(b_i, d_i)\}_{i=1}^N$ を計算し，ライフタイム $l_i := d_i - b_i$ の降順，つまり，$i < j$ ならば $l_i \geq l_j$，となるように添字を付け替える．その上で，最も長いライフタイム $l_1 = d_1 - b_1$ 以外が小さくなるように制約をつける．

Topology Layer を用いて，0 次のパーシステント図から連結性の制約をつける場合（例えば，[9], [15]），各 death に対して式 (2) を用いて単体（辺）を対応付け，式 (3) を用いて計算した損失関数を逆伝播させるため，各 death に対し 2 点のみにしか損失関数が逆伝播されない（図 2 左）．そのため，学習によって離れた連結成分同士を繋げるように修正する能力が限定的になると考えられる．

そこで，本手法ではパーシステントホモロジーの計算によって異なる連結成分同士が結合されるドロネー図内の

辺を特定し，その辺の端点付近の点集合に対して損失関数を計算する．具体的には，まず，ゼロレベルセットからサンプリングされた点群に対し，ライフタイムの降順で添字づけられた 0 次のパーシステント図 $\{(b_i, d_i)\}$ を計算する．次に，各 birth・death ペア (b_i, d_i) に対して，式 (2) の $\pi_F(0)(d_i)$ に対応する辺の端点 $v_{i,1}$，$v_{i,2}$ をとり，それらの端点の周りで一定距離 δ_i 以内の近傍点集合 $V'_{i,1}$，$V'_{i,2}$ を取得する．その後，二つの集合の要素数が $\min(|V'_{i,1}|, |V'_{i,2}|)$ となるように，必要であれば距離が遠い点から順に除いて，改めて近傍点集合を $V_{i,1}$，$V_{i,2}$ を作る（図 2 右）．最後に，それらの集合間の chamfer discrepancy を計算する：

$$L_{\mathrm{conn}} := \sum_{i=2}^{K_1} d(V_{i,1}, V_{i,2}). \tag{7}$$

ここで，修正を要するサンプリング点に集中的に逆伝播が行われるよう，ライフタイムの上位 K_1 個の death に対応する近傍点集合のみを用いる．

3.4.2 点群同士を比較する損失関数項

教師点群 X とゼロレベルセットのサンプリング点群 S_θ の不一致性をとらえるために chamfer discrepancy を用いる．$S_\theta = \{p_i(\theta)\}$ は $i < j$ ならば $\min_{x\in X} ||p_i(\theta) - x||_2 \geq \min_{x\in X} ||p_j(\theta) - x||_2$ となるように添字を付け替えておく．同様に，$X = \{x_i\}$ に対しても添字に対して S_θ との距離が単調減少になるよう添字を付け替える．このとき次の損失関数 L_{ps} は次のようになる：

$$L_{\mathrm{ps}} := \frac{1}{K_2} \sum_{i=1}^{K_2} \left(\min_{x\in X} ||p_i(\theta) - x||_2^2 + \min_{p(\theta)\in S_\theta} ||x_i - p(\theta)||_2^2 \right). \tag{8}$$

ここで，修正を要するサンプリング点に集中的に逆伝播が行われるよう chamfer discrepancy の式 (4) の二つの項の各々で上位 K_2 個の点のみ用いる．

4. 実験

4.1 問題設定

与えられた一つの連結な物体表面からサンプリングされた向き情報なしの点群に対して，SDF を近似する NN をオーバーフィットさせる問題を考える．ここでは，NN のパラメータを更新することを学習と呼ぶ．

4.2 データセット

本研究では，3D 形状のデータセット ShapeNet [6] の airplane クラスのメッシュからサンプリングされた点群データを用いた．ShapeNet のメッシュは，部品が一枚のメッシュからなる場合やメッシュ同士が交わっている場合などが見られ，一般には水密メッシュではなく，特に閉じた曲面とは限らない（図 3 左）．そこで，occupancy networks [14] の公開リポジトリの方法に従って，水密メッシュを作成し（図 3 右），そのメッシュからサンプリングした点群を用いた．評価データに用いるメッシュデータは Meshing Point [13] に従った．実際に作成した各物体のメッシュの中で，頂点数が 1% 未満の連結成分を無視して一つの連結成分からなる物体のメッシュを採用した．その結果 airplane クラスの 300 件のメッシュのうち 288 件が採用された．これらのメッシュからサンプリングした各 100,000 点を評価データとした．学習時には，各 100,000 点のうち一部のみを用いた（4.4 章）．

図 3 正解メッシュ例．左：元メッシュ．右：作成した水密メッシュ．
Fig. 3 An example of the ground truth mesh. Left: The original mesh. Right: the created watertight mesh.

4.3 訓練方法

本研究では，SAL [2] を再実装したものを用いて表面再構成を行った．損失関数 L_{base} を用いてベース学習を行なった後，提案手法の項を加えた式 (1) の損失関数 L_{total} を用いて追加学習を行った．L_{base} として NN の出力と符号なしの L_2 距離とを比較する関数を採用した：

$$L_{\mathrm{base}} = \mathbb{E}_{x\sim D_X} \big| |f_\theta(x)| - h_{2,X}(x) \big|, \tag{9}$$

$$h_{2,X}(x) = \min_{x'\in X} ||x - x'||_2. \tag{10}$$

ここで，D_X はデータ点群 X から定まる分布である．

4.4 データ準備

ベース学習・追加学習ともに，各物体の正解点群 100,000 点の中から，20,000 点を用いて行なった．各物体でベース学習・追加学習時に使用する 20,000 点は同一のものとした．学習の際は，点群の重心が原点となるように平行移動し，座標の最大値の絶対値が 1 となるようにスケールすることで，点群が $[-1, 1]^3 \subset \mathbb{R}^3$ に収まるようにした．

4.4.1 ネットワーク構造

SDF を近似する NN として，中間層 8 層の MLP $f_\theta \colon \mathbb{R}^3 \to \mathbb{R}$ を用いた（図 4）．ネットワークパラメータは重み正規化（weight normalization，WN）を行なった．各中間層の次元は 512 次元で，4 層目に入力層を連結した．活性化関数は ReLU（rectified linear unit) を用いた．

ベース学習の際は，NN の重みの初期化として幾何的初期化 [2] を用いることで，訓練開始時の NN が概ね半径 1.0 の球面の SDF となるようにした．

4.4.2 ハイパーパラメータ

L_{total} 式 (1) において L_{conn}，L_{ps} の重みは，$(\lambda_{\mathrm{conn}}, \lambda_{\mathrm{ps}}) =$

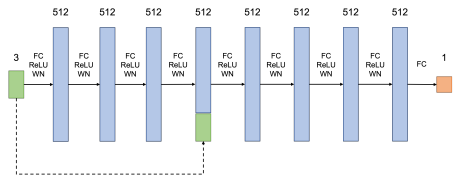


図 4 f_θ のネットワーク構造.
Fig. 4 Network architecture of f_θ.

$(1, 10)$ とした．L_{conn} 式 (7) において，$K_1 = 32$ とした．また，各 death d_i に対応する長さ l_i の辺の端点 $v_{1,i}$， $v_{2,i}$ の各近傍において，距離 $\delta_i = 0.5l_i$ 以内の点を用いて $V_{1,i}$, $V_{2,i}$ を作成した．L_{ps} 式 (8) において，$K_2 = 1024$ とした．

学習中のゼロレベルセットからのサンプリング（3.3 章）のためにマーチングキューブ法を適用する際は，$[-1, 1]^3 \subset \mathbb{R}^3$ に含まれる点群に対して，$[-1.1, 1.1]^3$ の範囲で解像度 128^3 のグリッドを用いた．サンプリングした点に加えるノイズの標準偏差 σ はグリッドサイズの 0.2 倍とした．ゼロレベルセットからのサンプリング点数は 32,768 点とした．

4.4.3 訓練設定

ベース学習は，Adam optimizer を学習率 0.0001, $(\beta_1, \beta_2) = (0.9, 0.999)$ で用いて，15,000 エポック行なった．

追加学習では，以下のように選択したベース学習の最良モデルの重みを初期値とした：ベース学習の過程で，正解点群 20,000 点と学習中のモデル用いて得られたメッシュからサンプリングされた 20,000 点との間の chamfer discrepancy の値を定期的に計算し，その値が最小のモデルをベース学習の最良モデルとする．

追加学習では，Adam optimizer を学習率 0.00005, $(\beta_1, \beta_2) = (0.9, 0.999)$ で用いて，500 エポック行なった．ただし，追加学習が失敗したもの（288 件中 1 件）については，学習率を 0.00001 として改めて追加学習を行なった．

損失関数の計算の際に設定する分布 D_X からのサンプリングは，SAL [2] の公開実装に従いデータ点の密度を考慮して行なった: X（20,000 点）からの一様分布，X の各点を中心に 50 番目の最近傍点の距離を標準偏差とする正規分布と X の各点を中心に標準偏差 1.0 の正規分布の混合分布，の 2 つの分布から，各エポック 16,384 点ずつサンプリングした．

4.5 評価指標及び評価方法

評価には，NN で近似した SDF からマーチングキューブ法 [12] で作成される各物体のメッシュを用いる．マーチングキューブ法を適用する際は，$[-1.1, 1.1]^3$ の範囲で解像度 128^3 のグリッドを用いた．

評価指標として，平均 chamfer discrepancy（mCD）及び， [15] で使用されている平均連結成分数（mCC）と連結率（CR）とを用いた．Chamfer discrepancy (式 (4)) は点群間の差異を測り，表面再構成のタスクの評価で一般的に用いられる指標である．各物体の正解データ点 X_i に対して，作成されたメッシュからランダムにサンプリングした点群 Y_i を用いて，$d(X_i, Y_i)$ を計算する．スケールは元データのサイズに戻して計算した．正解のデータ点数，メッシュからサンプリングした点数はともに 100,000 点である．平均連結成分数（mCC）は得られた再構成表面の連結成分のデータセットにおける平均であり，連結率（CR）はデータセットの中で連結成分が一つになる割合であり，ともに連結性に関する正当性を評価する．i 番目のデータに対して作成されたメッシュの連結成分数を c_i とすると，mCC，CR は各々以下で計算される：

$$\text{mCC} = \frac{1}{N}\sum_{i=1}^{N} c_i, \ \text{CR} = \frac{1}{N}\big|\{i \mid c_i = 1, i = 1, 2, \ldots, N\}\big|.$$

ここで，N はデータ数，$|\cdot|$ は集合の要素数である．

ベース学習のモデル選択の際は，学習時に用いる 20,000 点で評価した chamfer discrepancy が最小のモデルを選択した．追加学習のモデル選択の際は，学習時に用いる 20,000 点で評価した chamfer discrepancy が最小のモデル（ただし，初期値のモデルは除く）を選択した．

表 1 定量評価．B：ベース学習，FT：追加学習．
Table 1 Quantitative results. B: base learning, FT: additional learning.

モデル (損失関数項)	mCD[10^{-5}](↓)	mCC(↓)	CR(↑)
B(L_{base})	2.458	1.990	0.677
FT(L_{base})	2.382	1.917	0.670
FT(L_{base}, L_{conn})	25.20	**1.208**	0.833
FT(L_{base}, L_{ps})	**0.9956**	1.660	0.698
FT(L_{base}, L_{conn}, L_{ps})	1.579	1.240	**0.847**

4.6 結果

4.6.1 定量評価及び定性評価

表 1 に aiplane クラスの定量評価を，ベース学習及び追加学習において損失関数項の組み合わせを変えた場合と合わせて示す．損失関数項を新たに追加しない場合 FT(L_{base}) では，mCD・mCC で改善が見られる．mCC が最良（最小）のモデルは，連結性の損失関数項のみを新たに追加したモデル FT(L_{base}, L_{conn}) であった．このモデルでは，mCD のオーダーが変わり再構成表面の質が下がることが確認された．mCD が最良（最小）のモデルは，点群を比較する損失関数項のみを新たに追加したモデル FT(L_{base}, L_{ps}) であった．点群を比較する損失関数項のみを入れたこのモデルの場合も，FT(L_{base}) に比べて連結性の指標がより改善した．提案手法である，L_{total} を使ったモデル FT(L_{base}, L_{conn}, L_{ps}) では，CR が最良（最大）になった．このモデルでは，平均的に連結成分数を減らしつつ，chamfer discrepancy で測られる表面としての質を大きく損なわない，バランスの良い追加学習をできていると言える．

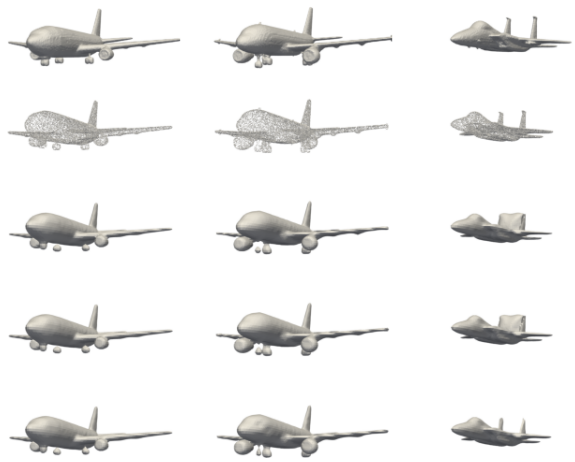


図 5 定性比較：良い例．上から，元メッシュ，学習点群，B(L_{base})，FT(L_{base})，FT(L_{base}, L_{conn}, L_{ps})．

Fig. 5 Qualitative comparison: good cases. From top to bottom, ground truth mesh, learning point cloud, B(L_{base}), FT(L_{base}), FT(L_{base}, L_{conn}, L_{ps}).

図 6 定性比較：悪い例．上から，元メッシュ，学習点群，B(L_{base})，FT(L_{base})，FT(L_{base}, L_{conn}, L_{ps})．

Fig. 6 Qualitative comparison: bad cases. From top to bottom, ground truth mesh, learning point cloud, B(L_{base}), FT(L_{base}), FT(L_{base}, L_{conn}, L_{ps}).

図 5 に，定性的に良い結果の例を示す．ベース学習及び FT(L_{base}) で途切れていた飛行機の脚の部分が，FT(L_{base}, L_{conn}, L_{ps}) では繋がっている（1, 2 列目）．また，膜のような部分が解消されている例（3 列目）もあり L_{ps} の項も効いていると考えられる．図 6 に，定性的に課題がある結果の例を示す．これらの例では，確かに連結成分数が減っているものの細い部分（胴体と羽を結ぶ棒, プロペラの先端や車輪）の再構成がうまくいっていない．

4.6.2 連結性の損失関数項による違い

ここでは，本手法で導入した連結性の損失関数項 L_{conn} と，0 次のパーシステント図のライフタイムのみを使って計算される連結性の損失関数項の効果を比較する．そこで，式 (1) の損失関数 L_{total} において L_{conn} を取り替えた，以下の損失関数を用いて追加学習した場合と比べた：

表 2 連結性の損失関数項の違いによる結果比較

Table 2 Comparison of results for different loss function terms of connectivity

損失関数 ($\lambda_{\text{conn}}, \lambda_{\text{ps}}$)	mCD[10^{-5}](↓)	mCC(↓)	CR(↑)
$L'_{\text{total}}(1, 10)$	0.9884	1.656	0.698
$L'_{\text{total}}(10, 10)$	0.9887	1.698	0.712
$L_{\text{total}}(0, 10)$	0.9956	1.660	0.698
$L_{\text{total}}(1, 10)$	1.579	1.240	0.847

$$L'_{\text{total}} := L_{\text{base}} + \lambda_{\text{conn}} L'_{\text{conn}} + \lambda_{\text{ps}} L_{\text{ps}},\ L'_{\text{conn}} = \sum_{i=2}^{K_1} |d_i - b_i|. \tag{11}$$

表 2 に定量指標の比較を示す．L'_{conn} では，λ_{conn} を大きくしても必ずしも mCC，CR がともに改善するとは限らなかった．また，L_{total} で連結性の項を入れない $(\lambda_{\text{conn}}, \lambda_{\text{ps}}) = (0, 10)$ の結果と平均的にはほぼ同程度の結果となった．厳密な比較にはより実験が必要と考えらえるが，L_{conn} の方が L'_{conn} よりも強く連結性の制約をかけられる傾向があると考えられる．

5. 議論

本研究における課題の一つは計算速度であり，現状ではベース学習に連結性の制約を入れることは現実的ではなく追加学習における手法として提案を行った．NVIDIA RTX A6000 GPU 1 つを用いて最大 40 スレッドで実行した場合，ベース学習は 15,000 エポックに対して 1 時間程度かかる一方，提案手法の追加学習は 500 エポックで 1 時間 30 分程度かかった．特に，パーシステントホモロジーの計算時間において，近似などを用いた高速化の工夫（例えば，[18]）が求められる．調整可能なハイパーパラメータが多く調整が難しい点も課題である．特に，連結性の項の重み係数を大きくした場合に，再構成表面の質が落ちてしまう現象が見られるなど，重み係数の選択は結果に大きな影響を与える．連結性の制約を入れつつもより強く質を担保するために，点群比較の損失関数項に用いる点群間の距離の選択や別の損失関数項の導入も検討要素である．

6. 結論

本研究では，NN による陰関数表現を用いた SDF による一つの連結物体の表面再構成おいて，パーシステントホモロジーを応用した連結性の制約を与える損失関数項と，点群同士を比較する損失関数項を導入し，追加学習する方法を提案した．連結制約を実現しつつ表面としての質を大きく損なわないバランスの良い追加学習をできていることが，定量的・定性的に確認された．また，連結成分の分断

箇所の近傍点を利用することで従来よりも効率的に連結性の制約をかけることが可能になった．

より高次元のトポロジカルな制約を用いた制約，データセットからの形状空間の学習への提案手法の適用は，興味深い今後の方向性である．

謝辞 本研究を行うにあたり，株式会社 ALBERT の山内隆太郎氏には，実装のアドバイス及び手法・草稿に関するコメントを頂いた．また，同社の松林達史氏には継続的な励ましと草稿へのコメントを頂いた．両氏に感謝する．

参考文献

[1] Atzmon, M., Haim, N., Yariv, L., Israelov, O., Maron, H. and Lipman, Y.: Controlling neural level sets, *Advances in Neural Information Processing Systems*, pp. 2032–2041 (2019).

[2] Atzmon, M. and Lipman, Y.: SAL: Sign Agnostic Learning of Shapes From Raw Data, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[3] Atzmon, M. and Lipman, Y.: SALD: Sign Agnostic Learning with Derivatives, *International Conference on Learning Representations* (2021).

[4] Baorui, M., Zhizhong, H., Yu-shen, L. and Matthias, Z.: Neural-Pull: Learning Signed Distance Functions from Point Clouds by Learning to Pull Space onto Surfaces, *International Conference on Machine Learning (ICML)* (2021).

[5] Brüel-Gabrielsson, R., Ganapathi-Subramanian, V., Skraba, P. and Guibas, L. J.: Topology-Aware Surface Reconstruction for Point Clouds, *COMPUTER GRAPHICS FORUM* (2020).

[6] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L. and Yu, F.: ShapeNet: An Information-Rich 3D Model Repository, Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015).

[7] Chen, Z. and Zhang, H.: Learning Implicit Fields for Generative Shape Modeling, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

[8] Edelsbrunner, H. and Harer, J.: *Computational Topology - an Introduction.*, American Mathematical Society (2010).

[9] Gabrielsson, R. B., Nelson, B. J., Dwaraknath, A. and Skraba, P.: A Topology Layer for Machine Learning, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, Vol. 108, PMLR, pp. 1553–1563 (2020).

[10] Gameiro, M., Hiraoka, Y. and Obayashi, I.: Continuation of point clouds via persistence diagrams, *Physica D: Nonlinear Phenomena*, Vol. 334, pp. 118–132 (online), DOI: https://doi.org/10.1016/j.physd.2015.11.011 (2016).

[11] Gropp, A., Yariv, L., Haim, N., Atzmon, M. and Lipman, Y.: Implicit Geometric Regularization for Learning Shapes, *Proceedings of Machine Learning and Systems 2020*, pp. 3569–3579 (2020).

[12] Lewiner, T., Lopes, H., Vieira, A. W. and Tavares, G.: Efficient Implementation of Marching Cubes' Cases with Topological Guarantees, *Journal of Graphics Tools*, Vol. 8, No. 2, pp. 1–15 (online), DOI: 10.1080/10867651.2003.10487582 (2003).

[13] Liu, M., Zhang, X. and Su, H.: Meshing Point Clouds with Predicted Intrinsic-Extrinsic Ratio Guidance, *Computer Vision – ECCV 2020*, Springer International Publishing, pp. 68–84 (online), DOI: 10.1007/978-3-030-58598-3_5 (2020).

[14] Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S. and Geiger, A.: Occupancy Networks: Learning 3D Reconstruction in Function Space, *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2019).

[15] Mezghanni, M., Boulkenafed, M., Lieutier, A. and Ovsjanikov, M.: Physically-Aware Generative Network for 3D Shape Modeling, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9330–9341 (2021).

[16] Nigmetov, A., Krishnapriyan, A. S., Sanderson, N. and Morozov, D.: Topological Regularization via Persistence-Sensitive Optimization (2020).

[17] Poulenard, A., Skraba, P. and Ovsjanikov, M.: Topological Function Optimization for Continuous Shape Matching, *Computer Graphics Forum*, Vol. 37 (2018).

[18] Solomon, E., Wagner, A. and Bendich, P.: A Fast and Robust Method for Global Topological Functional Optimization, *AISTATS* (2021).

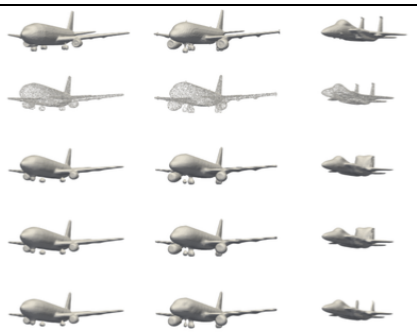
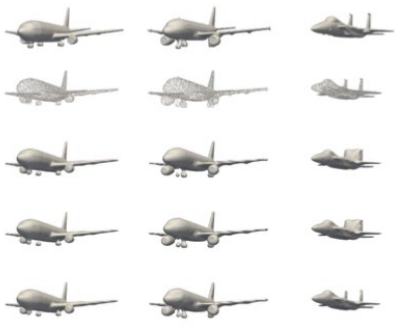
[19] Wang, Z., Wang, P., Dong, Q., Gao, J., Chen, S., Xin, S., Tu, C. and Wang, W.: Neural-IMLS: Learning Implicit Moving Least-Squares for Surface Reconstruction from Unoriented Point Clouds (2021).

[20] Wu, R., Zhuang, Y., Xu, K., Zhang, H. and Chen, B.: PQ-NET: A Generative Part Seq2Seq Network for 3D Shapes, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[21] 平岡裕章：タンパク質構造とトポロジー: パーシステントホモロジー群入門，共立出版 (2013).

正誤表

下記の箇所に誤りがございました．お詫びして訂正いたします．

訂正箇所	誤	正
5 ページ 3.4.1 章 式(7)	$L_{\mathrm{conn}} := \sum_{i=2}^{K_1} d(V_{i,1}, V_{i,2}).$	$L_{\mathrm{conn}} := \sum_{i=2}^{K_1+1} d(V_{i,1}, V_{i,2}).$
5 ページ 3.4.1 章 最終文内	…，ライフタイムの上位K_1個の death に対応する…	…，ライフタイムの最上位を除く上位K_1個の death に対応する…
5 ページ 4.4.2 章 1 文目内	…の重みは，$(\lambda_{\mathrm{conn}}, \lambda_{\mathrm{ps}}) = (1,10)$とした．	…の重みは，$(\lambda_{\mathrm{conn}}, \lambda_{\mathrm{ps}}) = (0.5,10)$とした．
6 ページ 4.6.1 章 表 1	**表 1** 定量評価．B：ベース学習，FT：追加学習． **Table 1** Quantitative results. B: base learning, FT: additional learning. モデル (損失関数項) / mCD[10^{-5}](↓) / mCC(↓) / CR(↑) B(L_{base}) / 2.458 / 1.990 / 0.677 FT(L_{base}) / 2.382 / 1.917 / 0.670 FT(L_{base}, L_{conn}) / 25.20 / **1.208** / 0.833 FT(L_{base}, L_{ps}) / **0.9956** / 1.660 / 0.698 FT(L_{base}, L_{conn}, L_{ps}) / 1.579 / 1.240 / **0.847**	**表 1** 定量評価．B：ベース学習，FT：追加学習． **Table 1** Quantitative results. B: base learning, FT: additional learning. モデル (損失関数項) / mCD[10^{-5}](↓) / mCC(↓) / CR(↑) B(L_{base}) / 2.458 / 1.990 / 0.677 FT(L_{base}) / 2.382 / 1.917 / 0.670 FT(L_{base}, L_{conn}) / 24.25 / 1.212 / 0.833 FT(L_{base}, L_{ps}) / **0.9956** / 1.660 / 0.698 FT(L_{base}, L_{conn}, L_{ps}) / 1.560 / **1.181** / **0.872**
6 ページ 4.6.1 章 3・4 文目	mCCが最良（最小）のモデルは，連結性の損失関数項のみを新たに追加したモデル FT(L_{base}，L_{conn})であった．このモデルでは，mCD のオーダーが変わり再構成表面の質が下がることが確認された．	連結性の損失関数項のみを新たに追加したモデル FT(L_{base}，L_{conn})では，mCDのオーダーが変わり再構成表面の質が下がることが確認された．
6 ページ 4.6.1 章 7 文目	提案手法である，L_{total}を使ったモデル FT(L_{base}，L_{conn}，L_{ps})では，CRが最良（最大）になった．	提案手法である，L_{total}を使ったモデル FT(L_{base}，L_{conn}，L_{ps})では，mCCが最良（最小）・CRが最良（最大）になった．
7 ページ 4.6.1 章 図 5	 **図 5** 定性比較：良い例．上から，元メッシュ，学習点群，B(L_{base})，FT(L_{base})，FT(L_{base}, L_{conn}, L_{ps})． **Fig. 5** Qualitative comparison: good cases. From top to bottom, ground truth mesh, learning point cloud, B(L_{base}), FT(L_{base}), FT(L_{base}, L_{conn}, L_{ps}).	 **図 5** 定性比較：良い例．上から，元メッシュ，学習点群，B(L_{base})，FT(L_{base})，FT(L_{base}, L_{conn}, L_{ps})． **Fig. 5** Qualitative comparison: good cases. From top to bottom, ground truth mesh, learning point cloud, B(L_{base}), FT(L_{base}), FT(L_{base}, L_{conn}, L_{ps}).

<table>
<tr><td>7 ページ
4.6.1 章
図 6</td><td> 
図 6　定性比較：悪い例．上から，元メッシュ，学習点群，B(L_{base})，FT(L_{base})，FT(L_{base}, L_{conn}, L_{ps})．
Fig. 6　Qualitative comparison: bad cases. From top to bottom, ground truth mesh, learning point cloud, B(L_{base}), FT(L_{base}), FT(L_{base}, L_{conn}, L_{ps}).</td><td> 
図 6　定性比較：不十分な例．上から，元メッシュ，学習点群，B(L_{base})，FT(L_{base})，FT(L_{base}, L_{conn}, L_{ps})．
Fig. 6　Qualitative comparison: insufficient cases. From top to bottom, ground truth mesh, learning point cloud, B(L_{base}), FT(L_{base}), FT(L_{base}, L_{conn}, L_{ps}).</td></tr>
<tr><td>7 ページ
4.6.2 章
式(11)内</td><td>$$L'_{\text{conn}} = \sum_{i=2}^{K_1} |d_i - b_i|.$$</td><td>$$L'_{\text{conn}} = \sum_{i=2}^{K_1+1} |d_i - b_i|^2.$$</td></tr>
<tr><td>7 ページ
4.6.2 章
表 2</td><td>表 2　連結性の損失関数項の違いによる結果比較
Table 2　Comparison of results for different loss function terms of connectivity
損失関数 ($\lambda_{\text{conn}}, \lambda_{\text{ps}}$)	mCD[10^{-5}](↓)	mCC(↓)	CR(↑)
$L'_{\text{total}}(1, 10)$	0.9884	1.656	0.698
$L'_{\text{total}}(10, 10)$	0.9887	1.698	0.712
$L_{\text{total}}(0, 10)$	0.9956	1.660	0.698
$L_{\text{total}}(1, 10)$	1.579	1.240	0.847 |</td><td>表 2　連結性の損失関数項の違いによる結果比較 Table 2　Comparison of results for different loss function terms of connectivity
損失関数 ($\lambda_{\text{conn}}, \lambda_{\text{ps}}$)	mCD[10^{-5}](↓)	mCC(↓)	CR(↑)
$L'_{\text{total}}(1, 10)$	1.543	1.229	0.840
$L_{\text{total}}(0.5, 10)$	1.560	1.181	0.872
$L'_{\text{total}}(2, 10)$	2.080	1.115	0.906
$L_{\text{total}}(1, 10)$	2.039	1.111	0.924 |</td></tr>
<tr><td>7 ページ
4.6.2 章
4・5・6 文目</td><td>L'_{conn}では，λ'_{conn}を大きくしても必ずしもmCC，CRがともに改善するとは限らなかった．また，L'_{total}で連結性の項を入れない$(\lambda_{\text{conn}}, \lambda_{\text{ps}}) = (0,10)$の結果と平均的にはほぼ同程度の結果となった．厳密な比較にはより実験が必要と考えらえるが，L_{conn}の方がL'_{conn}よりも強く連結性の制約をかけられる傾向があると考えられる．</td><td>L'_{conn}に比べ，L_{conn}は点集合の組の双方向からの距離を計算しているため，L'_{total}ではλ_{conn}が2倍の場合を比較対象とした．L_{total}の方がL'_{total}より強く連結性の制約をつけている傾向はあるものの，違いの有無を結論づけるためにはより実験が必要と考えられる．</td></tr>
<tr><td>7 ページ
6 章
3 文目</td><td>また，連結成分の分断箇所の近傍点を利用することで従来よりも効率的に連結性の制約をかけることが可能になった．</td><td>(削除)</td></tr>
</table>