混合ガウス分布推定ネットワークを用いた単一画像からの３次元物体形状復元

山下 浩平[1,a)]　延原 章平[1,b)]　西野 恒[1,c)]

概要：本研究では，単一 RGB 画像から，物体の３次元形状を混合ガウス分布として推定するニューラルネットワークの導出を行う．従来のボクセルや点群による形状表現では目標形状との相違度は離散的なものであったが，提案手法では混合ガウス分布を用いることで目標３次元点群との相違度を連続的に求め，学習に用いることができる．具体的には，点群を真の分布からの標本であるとし，ネットワークの出力分布の標本点での負の対数尤度を損失関数として学習を行う．さらに，３次元混合ガウス分布を，平行透視投影を仮定して画像平面に解析的に投影する手法を提案し，これを利用した多視点幾何的な損失関数も学習に用いる．3D モデルから作成した画像，シルエット，点群を用いた実験を行い，提案手法の有効性を示す．

1. はじめに

画像に基づく被写体の３次元形状復元は，ロボティクスなどにおける幅広い応用が期待される．例えば，ロボットに備え付けられたカメラから周囲の物体の３次元形状を認識することができれば，ロボットがどのような行動をするべきか決定する手がかりとなる．周囲の３次元形状を把握することができれば，その情報を利用して逆にロボットの自己位置推定を行うこともできる [1], [2], [3]．そのため，画像から３次元形状復元を行う研究が様々なアプローチでなされてきた [4]．

特に近年，深層学習を用いて単一画像から物体形状を直接推定する様々な手法が提案されている [5], [6], [7]．これらでは，画像を入力，３次元形状を出力とするニューラルネットワークを作成する．ネットワークの学習には，3D モデルを用いて生成した画像やボクセルデータ，３次元点群などを用いる．大量のデータから画像や物体形状に関する規則性が自動的に学習され，画像からの形状復元が実現される．学習を行ったネットワークは，物体領域が既知であれば実画像からも３次元形状を復元することができる．学習時にデータ拡張などの工夫を行うことで，背景のある実画像からでも形状復元を行うことができる [5]．

これらの形状復元手法では，３次元形状をボクセルや点群，メッシュモデルなどで離散的に表現する．ボクセルは，空間を格子状に区切り，各格子について物体が存在するかという情報を保持することで形状を表現する．点群は，物体が存在する３次元位置に分布する点の集合として形状を表現する．メッシュモデルは，複数の頂点を結んで作られる多角形の面の集合として，物体の表面形状を表現する．これらの離散的な表現方法はコンピュータでデータとして扱う上での自然な表現であり，また，可視化も容易である．

しかし，物体のトラッキングなどの応用を考えた場合，これらは効率の良い表現方法なのだろうか？例えば，ボクセルや点群として表現された２つの形状について形状間の相違度を求める場合，値は本質的に離散的である．離散化の解像度が高ければこの問題は無視できるが，膨大な計算量が要求される．

このような観点から，本研究では図１のように３次元形状を混合ガウス分布として推定するニューラルネットワークを提案する．混合ガウス分布は複数のガウス分布の重み付け和として表現される確率密度分布であり，様々な密度

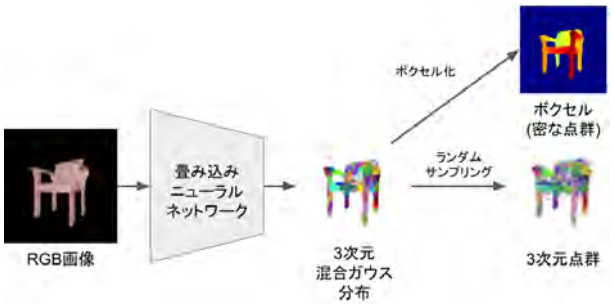

図 1　混合ガウス分布推定ネットワークを用いた単一画像からの３次元物体形状復元．推定した分布からは，ランダムサンプリングした点群や標本化・二値化したボクセルデータ (密な点群) を生成することができる．

[1] 京都大学大学院 情報学研究科
[a)] kyamashita@vision.ist.i.kyoto-u.ac.jp
[b)] nob@.i.kyoto-u.ac.jp
[c)] kon@.i.kyoto-u.ac.jp

分布を近似することができる．学習の際には，3次元形状に関する教師データである点群を，真の形状 (分布) からの標本とみなす．標本における推定分布の負の対数尤度を損失関数とし，推定分布が真の形状を近似するようにする．学習を経て得られる推定密度分布からは，必要に応じて点群やボクセルデータを生成することができる．

また，3次元の混合ガウス分布の画像平面への投影は，平行透視投影を仮定することで2次元の混合ガウス分布で近似することができる．本研究ではこれを利用し，多視点シルエット画像を教師として学習に用い，多視点幾何的な整合性を高める．

2. 関連研究

ニューラルネットワークによって画像から物体形状を推定する手法の多くでは，オートエンコーダのようなネットワーク構造が用いられている．これらでは，まず，エンコーダによって形状に相当する内部表現を画像から推定する．デコーダによって内部表現をさらに変換することで所望の形式で表現された出力形状を得る．Choy らは，ボクセルとして形状を推定するネットワーク構造を提案した [5]．Fan らは，3次元形状を効率よく表現できることからネットワークの出力を点群とした．そして，学習のための損失として目標点群との距離損失を用いた [6]．Jiang らは，Fan らと同様にネットワークの出力を点群としつつ，多視点幾何的な損失関数や敵対的学習法を用いることで推定精度を向上させた [7]．本研究ではこれらと同様に，3D モデルから作成した画像や点群を用いて学習を行うが，物体形状を連続的な混合ガウス分布として推定する．

3次元点群を教師とし，真の形状を密度分布として推定するというタスクは，標本点から真の密度分布を推定する密度推定と類似している．そのため，ニューラルネットワークによってある入力の下での条件付き密度分布を推定する手法を適用することができる．Bishop は，条件付き確率密度分布を混合分布として推定するニューラルネットワークを提案した [8]．彼は，混合分布の具体的な基底を等方なガウス分布とした．そして，出力層において適切な活性化関数を用いることにより，ネットワークの出力が分布のパラメータとして破綻しないようにした．また，Williams は，混合分布の基底を一般の多変量ガウス分布に拡張したネットワーク構造を提案した [9]．本研究ではこれらを単一画像からの3次元物体形状復元に応用する．そして，実験により有効性を示す．

3. 混合ガウス分布の性質

本節では，ネットワークの出力となる混合ガウス分布の性質，特に，パラメータが満たすべき制約条件や，真の分布との相違度を点群を用いて近似計算する方法について述べる．これらは3次元形状を密度分布として推定するネットワークの構造や学習法を考える上で重要である．また，3次元ガウス分布を平行透視投影を仮定して近似的に画像平面に投影する手法についても述べる．これにより，3次元点群を用いた損失関数だけでなく，シルエット画像を用いた多視点幾何的な損失関数も学習に用いることが可能となる．

3.1 混合ガウス分布

3次元のガウス分布は，密度関数が

$$\phi(\boldsymbol{x}|\boldsymbol{\mu},\Sigma) = \frac{1}{(2\pi)^{\frac{3}{2}}\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2} g(\boldsymbol{x}|\boldsymbol{\mu},\Sigma)\right) \tag{1}$$

で定義される．ただし，$\boldsymbol{\mu}$ は平均，Σ は共分散行列であり，$g(\boldsymbol{x}|\boldsymbol{\mu},\Sigma)$ は

$$g(\boldsymbol{x}|\boldsymbol{\mu},\Sigma) = (\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \tag{2}$$

である．また，Σ は正定値対称行列でなければならない．

混合ガウス分布は，複数のガウス分布を重み付けして足し合わせたものとなっており，密度関数は分布の数を N として，

$$f(\boldsymbol{x}|\{\pi_i\},\{\boldsymbol{\mu_i}\},\{\Sigma_i\}) = \sum_{i=1}^{N} \pi_i \phi(\boldsymbol{x}|\boldsymbol{\mu_i},\Sigma_i) \tag{3}$$

で定義される．ただし，π_i は各分布の重みであり，

$$\sum_{i=1}^{N} \pi_i = 1 \tag{4}$$

を満たす．混合ガウス分布は多峰的な分布を表現することができる．

3.2 真の分布のカルバック・ライブラー情報量

推定された分布と真の分布の近さを表すものとして，カルバック・ライブラー情報量がある．推定された密度分布 $q(\boldsymbol{x})$ に対する真の密度分布 $p(\boldsymbol{x})$ のカルバック・ライブラー情報量 $\mathrm{KL}(p||q)$ は

$$\mathrm{KL}(p||q) = \int p(\boldsymbol{x}) \left\{\log p(\boldsymbol{x}) - \log q(\boldsymbol{x})\right\} \mathrm{d}\boldsymbol{x} \tag{5}$$

である．この $\mathrm{KL}(p||q)$ を最小化する $q(\boldsymbol{x})$ によって $p(\boldsymbol{x})$ が近似される．しかし，目標分布 $p(\boldsymbol{x})$ そのものは未知であり，$\mathrm{KL}(p||q)$ を直接計算することはできない．そこで，$p(\boldsymbol{x})$ からサンプリングした点群 P を利用し，$\mathrm{KL}(p||q)$ を

$$\mathrm{KL}(p||q) \approx \frac{1}{|P|} \sum_{\boldsymbol{x}\in P} \left\{\log p(\boldsymbol{x}) - \log q(\boldsymbol{x})\right\} \tag{6}$$

と近似する．ただし $|P|$ は点群の点数を表す．$\sum_{\boldsymbol{x}\in P} \log p(\boldsymbol{x})$ は推定分布 $q(\boldsymbol{x})$ の変化に対して一定であるため，$\mathrm{KL}(p||q)$ の最小化は

$$L_{3D}(q(\boldsymbol{x}),P) = -\frac{1}{|P|} \sum_{\boldsymbol{x}\in P} \log q(\boldsymbol{x}) \tag{7}$$

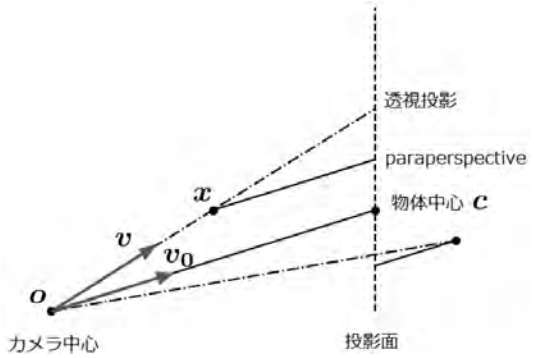

図 2 平行透視投影モデルの概念図．各地点の視線方向ベクトル $\boldsymbol{v}$ を物体中心における視線方向ベクトル $\boldsymbol{v_0}$ で近似する．

を最小化することと等価である．$q(\boldsymbol{x})$ が混合ガウス分布の場合，式 (3) を代入して，

$$L_{3D}(q(\boldsymbol{x}), P) = -\frac{1}{|P|}\sum_{\boldsymbol{x}\in P}\log\sum_{i=1}^{N}\pi_i\phi(\boldsymbol{x}|\boldsymbol{\mu_i}, \Sigma_i) \quad (8)$$

となる．$L_{3D}(q(\boldsymbol{x}), P)$ を損失関数として用いることで，ネットワークの出力分布を真の分布に近づけることができる．

3.3 ガウス分布の平行透視投影

一般に，カメラの幾何的特性は透視投影モデルで表される．透視投影モデルでは，カメラ中心と被写体の 3 次元位置を結ぶ直線 (視線方向) を考え，この直線と投影面の交わる位置を画像平面上における投影位置とする．しかし，透視投影は非線形な変換であり，3 次元ガウス分布の投影には複雑な計算を要する．そこで，透視投影の近似的なモデルとして平行透視投影を仮定する．これにより，3 次元ガウス分布の投影を 2 次元のガウス分布として解析的に求めることが可能となる．

図 2 に平行透視投影の概念図を示す．平行透視投影では，投影面が被写体の中心を通る面であると考える．そして，被写体の視線方向ベクトル $\boldsymbol{v}$ を，被写体の中心の視線方向ベクトル $\boldsymbol{v_0}$ で近似する．すると，3 次元点 $\boldsymbol{x}$ の投影面における位置 (u, v) を求める問題は

$$\boldsymbol{x} - \boldsymbol{c} = u\boldsymbol{a} + v\boldsymbol{b} + w\boldsymbol{v_0} \quad (9)$$

を満たす u，v，w を求める問題と等価となる．ただし $\boldsymbol{a}$ および $\boldsymbol{b}$ は画像の横および縦方向に対応する単位ベクトルであり，$\boldsymbol{c}$ は物体中心位置を表す．縦ベクトル $\boldsymbol{a}$，$\boldsymbol{b}$，$\boldsymbol{v_0}$ を並べた 3×3 行列を M と置くと，

$$\boldsymbol{x} - \boldsymbol{c} = M\begin{pmatrix} u & v & w \end{pmatrix}^{\mathrm{T}} \quad (10)$$

となり，u，v，w は

$$\begin{pmatrix} u & v & w \end{pmatrix}^{\mathrm{T}} = M^{-1}\boldsymbol{x} - M^{-1}\boldsymbol{c} \quad (11)$$

となる．この式は直交座標系から，$\boldsymbol{c}$ に関する平行透視投影に対応する 3 次元座標系への変換と考えられる．u および v は投影位置に相当し，w は投影面までの $\boldsymbol{v_0}$ 方向に見た距離を表す．

平行透視投影に相当する座標変換が得られたので，この変換を行ったときのガウス分布パラメータの変換を考える．式 (11) より，$\boldsymbol{y} = (u, v, w)^{\mathrm{T}}$ と置くと，

$$\boldsymbol{x} = M\boldsymbol{y} + \boldsymbol{c} \quad (12)$$

である．なお，物体中心 $\boldsymbol{c}$ はガウス分布の平均 $\boldsymbol{\mu}$ とする．式 (1) についてこの座標変換を行うことで，座標変換後のガウス分布のパラメータ

$$\boldsymbol{\mu}' = M^{-1}\boldsymbol{\mu} - \boldsymbol{c}, \quad (13)$$

$$\Sigma'^{-1} = M^{\mathrm{T}}\Sigma^{-1}M \quad (14)$$

を得る．そして，$\phi(\boldsymbol{x}|\boldsymbol{\mu}', \Sigma')$ を展開，整理して w 方向について周辺化することで，奥行き方向に周辺化された 2 次元ガウス分布のパラメータ

$$\boldsymbol{\mu}'' = \begin{pmatrix} \mu_x & \mu_y \end{pmatrix}^{\mathrm{T}}, \quad (15)$$

$$\Sigma''^{-1} = \begin{pmatrix} s_{00} - \frac{s_{02}^2}{s_{22}} & s_{01} - \frac{s_{12}s_{02}}{s_{22}} \\ s_{10} - \frac{s_{12}s_{02}}{s_{22}} & s_{11} - \frac{s_{12}^2}{s_{22}} \end{pmatrix} \quad (16)$$

を得る．ただし，$\boldsymbol{\mu}' = (\mu_x, \mu_y)^{\mathrm{T}}$，$\Sigma'^{-1} = \{s_{ij}\}$ である．

なお，ここまでの議論は投影面を物体中心とした場合であり，実際の画像座標へ変換するには更に 2 次元のアフィン変換を行う必要がある．そのため，式 (13)，(14) と同様の変換を 2 次元のガウス分布のパラメータ $\boldsymbol{\mu}'', \Sigma''^{-1}$ に対して行い，最終的なパラメータ $\boldsymbol{\mu}''', \Sigma'''^{-1}$ を得る．

これらを用いれば，3 次元の混合ガウス分布の透視投影を 2 次元の混合ガウス分布で近似することができ，

$$d(\boldsymbol{x}) = \sum_{i=1}^{M}\pi_i\phi_{2D}(\boldsymbol{x}|\boldsymbol{\mu}_i''', \Sigma_i''') \quad (17)$$

のように密度分布画像を作成することができる．なお，$\phi_{2D}(\boldsymbol{x})$ は 2 次元のガウス分布であり，

$$\phi_{2D}(\boldsymbol{x}|\boldsymbol{\mu}''', \Sigma''') = \frac{1}{(2\pi)\sqrt{|\Sigma'''|}}\exp\left(-\frac{1}{2}g(\boldsymbol{x}|\boldsymbol{\mu}''', \Sigma'')\right) \quad (18)$$

である．ここまでの一連の操作は全て微分可能であり，ニューラルネットワークの学習に用いることができる．

4. 混合ガウス分布推定ネットワーク

本節では，前節で述べた混合ガウス分布の性質をもとに，具体的なネットワーク構造や学習法を提案する．図 3 に学習方法の概要を示す．本研究では，単一 RGB 画像を入力とし，混合ガウス分布を出力するネットワークを，入力画像，3 次元点群，多視点シルエット画像の組からなる教師データを用いて学習し，単一画像からの 3 次元物体形状復元を実現する．4.1 節では，ネットワーク構造，特にネッ

トワークの出力が密度分布のパラメータとして破綻しないための出力層の構造について述べる．4.2 節では，3 次元点群および多視点シルエット画像を用いた具体的な損失関数について述べる．4.3 節では，ネットワークが出力する密度分布を点群やボクセルデータに変換し，可視化する手法について述べる．

4.1 ネットワーク構造

ネットワークは，画像を入力とし，混合ガウス分布のパラメータ $\{\pi_i\},\{\boldsymbol{\mu}_i\},\{\Sigma_i^{-1}\},\{(\det(\Sigma_i^{-1}))^{\frac{1}{2}}\}(i=1,2,...N)$ を出力とする．具体的な構造として，畳み込み層と全結合層で構成される畳み込みニューラルネットワークを用いる．畳み込み層は入力画像に矩形の重み行列を畳み込むことで特徴マップを抽出する構造となっており，近傍の画素ほど関連の強い画像データから効率よく特徴抽出を行うことができる．また，プーリングを伴う畳み込み層を縦続接続し，広い受容野を持つ特徴を学習することができる．一方，全結合層は，全ての入力ニューロンと出力ニューロンについて別々の重みを持つため学習の自由度が高い．そのため，非線形活性化を伴う全結合層を重ねることで，画像特徴から混合ガウス分布のパラメータへの複雑な変換を学習できる．よって，ネットワークは $128\times128\times3$ のテンソル (RGB 画像) を入力とし，

(1) conv(3, 32, 7)
(2) conv(32, 64, 5)
(3) conv(64, 128, 5)
(4) conv(128, 256, 3)
(5) conv(256, 256, 3)
(6) $4\times4\times256$ テンソルを 4096 次元ベクトルに変換
(7) fc(4096, 1024)
(8) fc(1024, 1024)
(9) fc(1024, (混合数)×10)
(10)分布のパラメータへの変換 (後述)

という処理を順に行うものとする．ただし，conv(入力チャンネル数, 出力チャンネル数, フィルタのサイズ) は畳み込み層，fc(入力次元数, 出力次元数) は全結合層を表す．そして，畳み込み層では

(1) 2 次元畳み込み
(2) Batch Normalization
(3) Leaky ReLU による非線形活性化
(4) 2×2 の max プーリング

を順に行う．また，全結合層 (隠れ層) の活性化関数にも Leaky ReLU を用いる．ただし，出力層については次節で述べる活性化関数を用いる．

混合ガウス分布のパラメータをニューラルネットワークによって推定する場合，出力パラメータは各々に関する制約条件を常に満たしている必要がある．さもなければ学習中にパラメータが不正な値となり，学習が破綻する．そのため，Bishop や Williams のモデル [8], [9] のように，出力層の構造を工夫する．ガウス分布の平均 $\boldsymbol{\mu}_i$ は実数全てをとりうるから，対応する出力層の重み付け和 $\boldsymbol{a}_{\boldsymbol{\mu}_i}$ について活性化は行わない．ガウス分布の混合の重み π_i は正でありかつ総和が 1 である必要がある．そのため，対応する出力層の重み付け和 a_{π_i} は，softmax 関数による活性化

$$\pi_i=\frac{\exp(a_{\pi_i})}{\sum_i\exp(a_{\pi_i})} \tag{19}$$

を行う．共分散行列の逆行列 Σ_i^{-1} については正定値対称性が満たされる必要があるため，ネットワークの出力をこの行列とはせずに，以下のように考える．正定値対称行列である Σ_i^{-1} は下三角行列

$$L=\begin{pmatrix} l_{00} & 0 & 0\\ l_{10} & l_{11} & 0\\ l_{20} & l_{21} & l_{22}\end{pmatrix} \tag{20}$$

を用いて

$$\Sigma_i^{-1}=LL^{\mathrm{T}} \tag{21}$$

とコレスキー分解される．一方，このような L から求められる Σ_i^{-1} は必ず正定値対称となる. この性質を利用し，ネットワークは Σ_i^{-1} の代わりに下三角行列 L を推定する．ただし，対角成分 l_{ii} については正の値となるよう指数関数による活性化を行い，それ以外の要素については活性化を行わない．これにより，

$$\det(\Sigma_i^{-1})=\det(LL^{\mathrm{T}})=\det(L)\det(L^{\mathrm{T}}) \tag{22}$$

から，

$$\sqrt{\det(\Sigma_i^{-1})}=l_{00}l_{11}l_{22} \tag{23}$$

と求められる．

4.2 損失関数

学習には，まず，目標 3 次元点群と真の分布の近似的な相違度である 3 次元幾何損失を用いる．これは，目標点群からランダムにサンプルした点群 P に対して式 (8) の損失を求めるものである．この損失関数によってネットワークの出力分布 $q(\boldsymbol{x})$ が真の密度分布に近づくように学習される．

また，平行透視投影によって密度分布 $q(\boldsymbol{x})$ を任意の視点に投影した密度マップ $d_j(\boldsymbol{x})$ を作成できるため，K 枚の目標シルエット画像 $s_j(\boldsymbol{x})$ と整合するように多視点幾何損失を定義し，学習に用いる．本研究では，2 種類の多視点幾何損失を提案し，その効果について検討する．

まず，密度分布がシルエット内に収まっているかを評価する損失 L_d を考える．ネットワークの出力密度分布からサンプルした点が目標シルエット $s(\boldsymbol{x})$ 内に収まっている確率は $\sum_{\boldsymbol{x}}d(\boldsymbol{x})s(\boldsymbol{x})$ である．そこで，この確率の負の対数

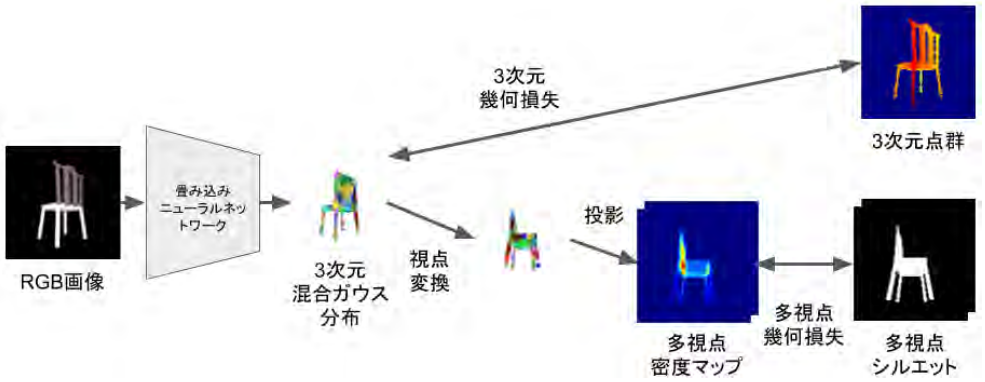

図 3 ネットワークの学習の概要．ネットワークは画像を入力として，混合ガウス分布のパラメータを推定する．学習時には 3 次元点群を教師とした 3 次元幾何損失および多視点シルエット画像を教師とした多視点幾何損失を利用する．

$$L_d(d(\boldsymbol{x}), s(\boldsymbol{x})) = -\log\left[\sum_{\boldsymbol{x}} d(\boldsymbol{x})s(\boldsymbol{x})\right] \tag{24}$$

を損失関数とすることで，推定密度分布が目標シルエット内に収まるように学習を行う．

この損失 L_d は，推定分布が目標シルエットからはみ出すことに対する制約にはなっているが，シルエット内で分布がどのような形となるかに関する制約とはなっていない．そのため，シルエット内の特定の部位だけ極端に密度が高くなるかもしれない．そこで，密度分布から擬似的にシルエット画像 $\hat{s}(\boldsymbol{x})$ を作成し，目標シルエット画像との相違度 L_{sil} を損失として用いる．具体的には，2 次元密度分布 $d(\boldsymbol{x})$ からランダムに M 点サンプリングしたとき，ある画素 x に少なくとも 1 つ以上の点が存在する確率 $\hat{s}(\boldsymbol{x})$ は

$$\hat{s}(\boldsymbol{x}) = 1 - \{1 - d(\boldsymbol{x})\}^M \tag{25}$$

である．この $\hat{s}(\boldsymbol{x})$ を密度分布から生成された点群のシルエットと考え，平均二乗誤差

$$L_{sil}(\hat{s}(\boldsymbol{x}), s(\boldsymbol{x})) = \sum_{\boldsymbol{x}} \{\hat{s}(\boldsymbol{x}) - s(\boldsymbol{x})\}^2 \tag{26}$$

をシルエットに関する損失とする．なお，$\hat{s}(\boldsymbol{x})$ を実際に学習や評価に使う際の M は，実験に用いた画像の画素数に合わせ $M = 128^2$ とした．

4.3 推定形状の可視化

ネットワークによって推定された 3 次元密度分布は点群やボクセルデータに変換し，可視化することができる．本研究では，2 種類の方法で密度分布の可視化を行う．

まず，推定された密度分布から擬似的にランダムサンプリングを行う方法が考えられる．具体的には，

(1) 重み π_i に応じた割合で混合ガウス分布からガウス分布をランダムに選択する．
(2) 選択されたガウス分布から 1 点疑似的にランダムサンプリングを行う．

という処理を必要な点数だけ繰り返し行う．この方法は分布の定義に即した自然な方法であり，点数以外のハイパーパラメータもない．また，各点が混合分布のどのクラスからサンプリングされたものかを調べ，点群を混合分布の各クラスごとに領域分割することができる．

次に，標本化と閾値処理により二値的なボクセルデータに変換する方法が挙げられる．この方法では，空間を格子状に区切り，各格子の中心における密度を計算する．そして，密度が一定の閾値を超えた場合，そこに物体が存在するとする (ボクセル値を 1 とする)．また，ボクセル値が 1 である地点について，混合分布の各クラスの密度 $\pi_i\phi(\boldsymbol{x}|\boldsymbol{\mu}_i, \Sigma_i)$ を求め，その値が最も大きいクラスに応じてラベル付けを行うことで，得られたボクセルデータを混合分布の各クラスで領域分割することができる．この処理を行うには閾値を手動で与える必要がある．分布の鋭さに対して標本化の幅が小さ過ぎても大き過ぎても正確な形状が得られない．また，計算オーダーが解像度の 3 乗となる．しかし，等間隔かつ密な点群を得ることができる．なお，評価実験では，ボクセル化の閾値は k-means による 2 クラス分類を行い決定する．

5. 評価実験

提案手法について椅子の画像を例にとり評価実験を行った．実験には，3D モデルデータセットである ShapeNet[10] から作成したデータを用いた．ShapeNet には 3D メッシュモデルや高解像度のボクセルデータが含まれている．本研究では 3D メッシュモデルを Blender を用いてレンダリングし，RGB 画像およびシルエット画像を生成した．また，高解像度のボクセルデータは等間隔の点群とみなすことができ，そこからのサンプリングを形状に相当する密度分布からのサンプリングと近似することができる．これらを用いて提案手法に関する学習を行い，混合分布の混合数および多視点幾何損失について評価を行った．

学習・評価のためのデータセットは以下の手順で作成し

た．まず，メッシュおよび点群 (surface) の 3D モデルについて，バウンディングボックスの中心が 3 次元座標系の原点となるよう 3 次元座標系を定めた．次に，Blender を用いてメッシュモデルを撮影した 128×128 の RGB 画像およびシルエット画像を各モデル 100 枚ずつ作成した．このとき，カメラ位置は原点から一定の距離のランダムな位置であるとする．また，カメラの中心軸は原点を通るように撮影するものとする．照明はカメラと同方向からシーン全体を均等に照らす HEMI ライトを用いた．データセットは 3D モデルごとに学習用データ，検証用データ，評価用データに 4:1:1 に分割し，実験に用いた (モデル A は学習用，モデル B は評価用というように分割した)．なお，学習時には，点群は入力画像の視点に合わせて回転させた．つまり，ネットワークが視点に応じて回転された物体形状を推定するように学習を行った．

また，評価には以下の 4 種類の方法で作成したデータを用いる．

- hemi：学習データと同じ条件で作成
- env：照明を環境光 (物体周囲の全方位から照らす) としたもの
- notex：3D モデル表面のテクスチャをなくし，全面白色としたもの
- env&notex：3D モデルのテクスチャをなくし，照明を環境光としたもの

これにより，物体の見えの変化に対する提案手法の頑健性を検討する．

5.1 評価指標

推定された 3 次元密度分布 $q(\boldsymbol{x})$ の評価には評価データにおける 3 次元幾何損失 L_{3D} を用いる．3 次元幾何損失は真の分布 $p(\boldsymbol{x})$ と推定分布 $q(\boldsymbol{x})$ のカルバック・ライブラー情報量 $\mathrm{KL}(p||q)$ のうち $q(\boldsymbol{x})$ に依存する項について近似的に求めたものであり，その値は形状の近さを表しているわけではない．しかし，評価データを固定すれば $q(\boldsymbol{x})$ に依存しない項は一定となり，2 つの推定結果 $q_1(\boldsymbol{x})$ と $q_2(\boldsymbol{x})$ の 3 次元幾何損失の差は $\mathrm{KL}(p||q_1)$ と $\mathrm{KL}(p||q_2)$ の差であると考えることができる．

また，推定密度分布に関する多視点幾何的な評価指標として，

$$\mathrm{IoU}(\hat{s}'(\boldsymbol{x}), s(\boldsymbol{x})) = \frac{\sum \hat{s}'(\boldsymbol{x}) \wedge s(\boldsymbol{x})}{\sum \hat{s}'(\boldsymbol{x}) \vee s(\boldsymbol{x})} \tag{27}$$

を用いる．$\hat{s}'(\boldsymbol{x})$ は疑似シルエット $\hat{s}(\boldsymbol{x})$ を 0.5 を境界として二値化したものであり．この指標は密度分布から生成された疑似シルエットと真のシルエットが近いほど大きくなる．入力視点について求めたものを IoU_{in} とし，多視点シルエットについて求めたものを IoU_{mv} とする．

表 1 混合数と 3 次元幾何損失の関係．各行が評価データのレンダリング設定に対応し，値は評価データにおける 3 次元幾何損失である．混合数が多いほど損失が小さくなっている．

混合数 N:	16	32	64	128	256	512
hemi	-2.65	-2.72	-2.78	-2.81	-2.84	-2.85
env	-2.61	-2.68	-2.74	-2.77	-2.80	-2.81
notex	-2.65	-2.72	-2.77	-2.81	-2.84	-2.85
env&notex	-2.61	-2.68	-2.68	-2.73	-2.79	-2.80

5.2 混合数と復元形状の関係

ネットワークが推定する混合ガウス分布の混合数について，比較実験により評価を行った．混合数はネットワークのパラメータであるが，この混合数 N を $N = 8, 16, 32, 64, 128, 256, 512$ としたネットワークについてそれぞれ 3 次元幾何損失を用いて学習を行い，評価データにおける損失の大きさを比較した．

学習アルゴリズムは Adam とし，バッチサイズを 64，学習率 $\alpha = 0.0001$ とした．また，3 次元幾何損失 L_{3D} の計算に用いる目標点群の点数は 8192 とした．そして，学習のエポック数は 40 回とし，検証用データにおける損失が最低となったエポック時の重みを最終的な学習結果とした．

復元結果を図 4 に示す．混合数が多い場合は，多くのガウス分布の混合によって椅子の足のような物体形状の比較的詳細な部分まで再現されている．一方，混合数が少ない場合は詳細形状は失われているものの，物体の概形が復元されている．3 次元幾何損失や疑似シルエットの IoU についても，表 1，2 のように混合数 512 の場合に最も良い値となっている．このことから，実験を行った範囲では混合数が多いほど物体形状をよく近似していると言える．ただし，混合数が多い場合，形状の効率のよい表現という意味は失われる．例えば，式 (8) を用いて推定密度分布と 3 次元点群の類似度を計算した場合，その計算オーダーは (混合数)×(点数) となり，混合数に比例する．また，混合数 512 の場合の復元結果を多視点から見たものを図 5 に示す．入力画像において視認不能な部分についても形状が復元されている．

5.3 多視点幾何損失の効果

多視点幾何損失の有無による復元結果の違いについて評価を行った．まず，3 次元幾何損失 L_{3D}(式 (8)) のみで学習する場合をベースライン手法 (3D) とした．そして，ランダムに選択した 2 視点の密度画像に関する損失 L_d(式 (24)) を学習に用い，損失関数全体で

$$L_{total} = L_{3D} + L_d \tag{28}$$

とした場合を手法 dmap とした．また，密度画像から生成した疑似シルエット画像に関する損失 L_{sil} を用い

$$L_{total} = L_{3D} + L_{sil} \tag{29}$$

とした場合を手法 sil とした．これらの手法について混合

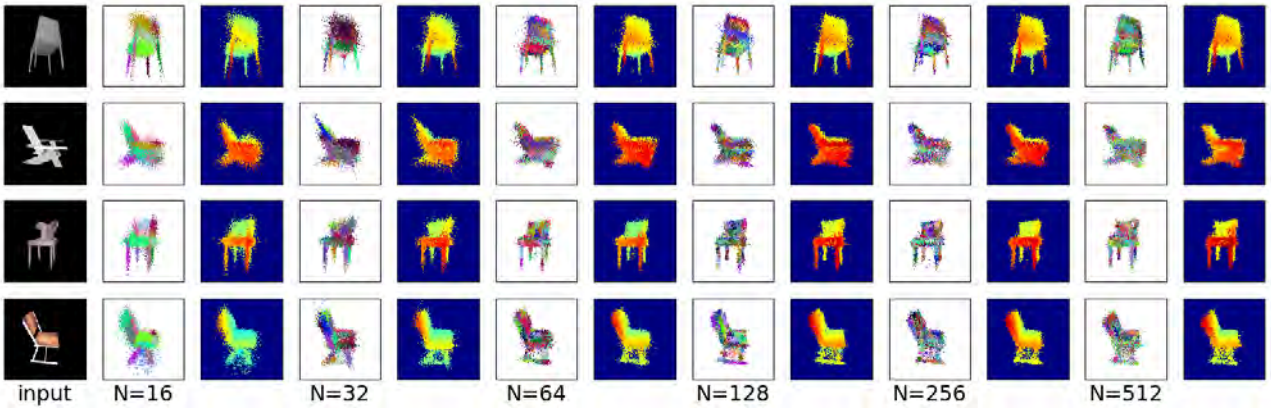

図 4　混合数と復元結果の関係．左が入力画像であり，右に行くほど混合数の多い場合の復元結果を表す．点群として可視化しており，左が混合クラスによる色付け，右が深度による色付けを行ったものである．混合数 N が多い場合，比較的詳細な形状まで復元されている．混合数が少ない場合は詳細形状は失われているものの，概形は復元されている．

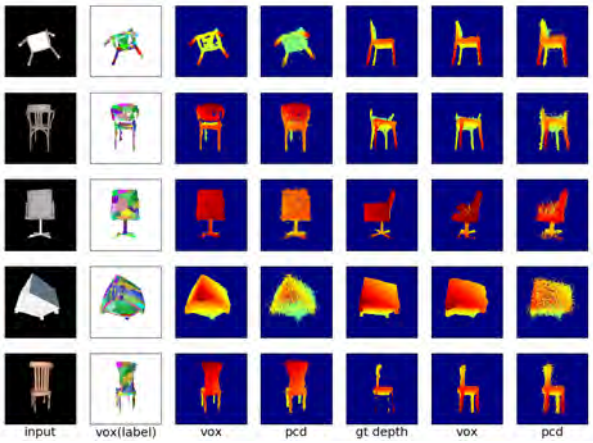

図 5　混合数 512 の場合の多視点から見た復元結果．input は入力画像，vox(label) はボクセル化した結果を混合分布のクラスにより色付けしたもの，vox・pcd はボクセル化・点群化した形状の深度を可視化したもの，gt depth は真の深度である．入力画像において視認不能な部分についても形状が復元されている．

表 2　推定密度分布から生成された疑似シルエットと正解シルエットの IoU．混合数 16 の場合でも IoU0.7 程度の精度でシルエットが推定されている．

多視点 (%)						
混合数:	16	32	64	128	256	512
hemi	69.5	72.2	73.9	75.0	75.4	75.7
env	69.4	71.9	73.7	74.7	75.1	75.3
notex	71.6	74.4	76.2	77.3	77.7	77.9
env&notex	71.5	74.1	75.9	77.0	77.5	77.7
入力視点 (%)						
混合数:	16	32	64	128	256	512
hemi	71.3	74.6	76.6	77.6	78.0	78.4
env	71.3	74.4	76.5	77.5	77.8	78.3
notex	73.5	76.9	79.0	80.0	80.4	80.8
env&notex	73.6	76.8	78.9	79.9	80.3	80.8

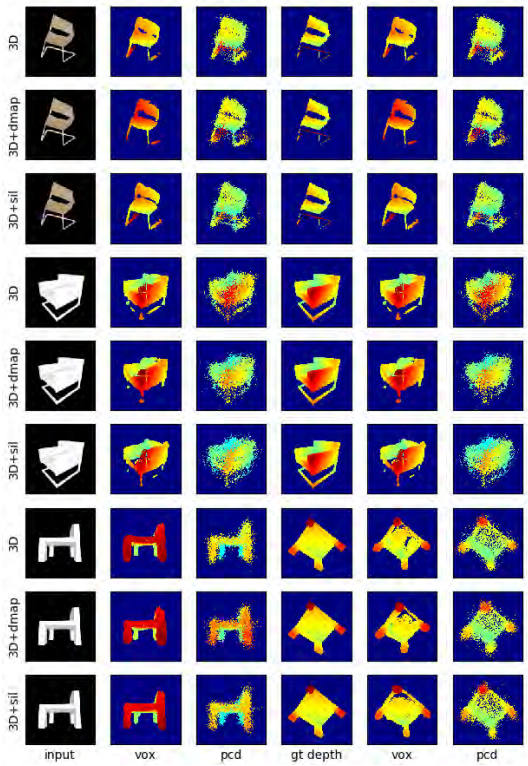

図 6　多視点幾何損失の有無による復元結果の比較．vox，pcd はボクセル化・点群化したした結果の深度であり，gt depth は真の深度である．復元結果について大きな差は見られない．

数 $N = 64$ の場合に前述のアルゴリズムで学習を行い，評価データにおける復元性能を比較した．

図 6 に復元結果を示す．また，表 3 に L_{3D}，表 4 に IoU_{in} および IoU_{mv} の結果を示す．復元形状を可視化した図には大きな差異は見られないものの，定量的には手法 dmap の多視点整合性 (IoU) がベースライン手法に比べ向上がみられる．よって，損失 L_d には多視点幾何的な整合性を高める作用があるといえる．

表 3　多視点幾何損失の有無と L_{3D} の関係．手法による差はほぼ見られないが，手法 sil の損失が比較的小さい．

	3D	dmap	sil
hemi	-2.78	-2.77	-2.78
env	-2.74	-2.73	-2.75
notex	-2.77	-2.77	-2.78
env&notex	-2.73	-2.73	-2.74

表 4　多視点幾何損失の有無による IoU の変化．手法 dmap が最も IoU が高い．

	多視点 (%)			入力視点 (%)		
	3D	dmap	sil	3D	dmap	sil
hemi	73.9	74.7	73.3	76.6	77.2	75.8
env	73.7	74.6	73.1	76.5	77.2	75.7
notex	76.2	76.9	75.5	79.0	79.5	78.2
env&notex	75.9	76.7	75.3	78.9	79.6	78.2

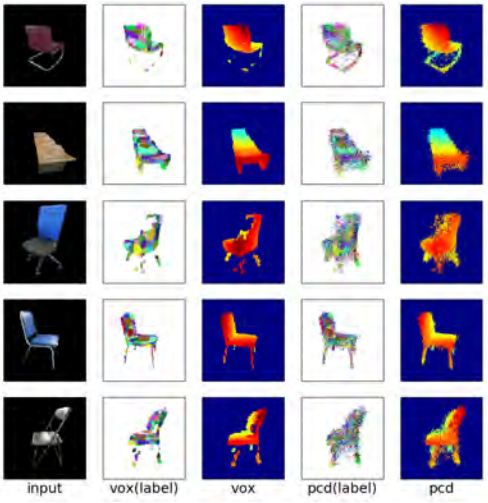

図 7　アノテーション付きの実画像からの 3 次元形状復元結果．左から，入力画像，ボクセルによる可視化 (ラベル，深度)，点群による可視化 (ラベル，深度) である．上から 3 番目の画像以外は概ね 3 次元形状が復元されている．

5.4　実画像からの復元

ネットワークの凡化性能の評価のため，手動でアノテーションを行った実画像をネットワークに入力し，形状復元を行った．

結果を図 7 に示す．3 次元的に誤っているものもあるが，概ね正しい 3 次元形状が復元されている．また，入力視点から見たシルエットに関してはほぼ一致している．

6.　まとめ

画像から混合ガウス分布として 3 次元形状を復元するネットワークおよびその学習法の提案を行った．3D モデルから作成したデータセットを用いて実験を行い，レンダリング条件の異なる評価用画像やアノテーション付きの実画像についても 3 次元形状が復元されることを示した．また，3 次元混合ガウス分布の投影を 2 次元混合ガウス分布として近似的に求める手法を提案した．そして，これを用いた多視点幾何的な損失関数により，推定形状の多視点幾何的な整合性が向上することを示した．

今後の課題としては，まず，実画像への適応が考えられる．実画像の場合，正確なアノテーションが得られず，カメラと物体の位置関係も様々である．そのため，教師データを実データに近づける工夫が必要と考えられる．また，混合ガウス分布によって表現された形状を用いて撮影物体のトラッキングや自己位置推定を行うことが応用として考えられる．

謝辞

本研究の一部は JSPS 科研費 17K20143 の助成を受けたものです．

参考文献

[1] Klein, G. and Murray, D.: Parallel Tracking and Mapping for Small AR Workspaces, *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan (2007).

[2] Newcombe, R. A., Lovegrove, S. J. and Davison, A. J.: DTAM: Dense tracking and mapping in real-time, *2011 International Conference on Computer Vision (ICCV)*, pp. 2320–2327 (online), DOI: 10.1109/ICCV.2011.6126513 (2011).

[3] Mur-Artal, R., Montiel, J. M. M. and Tardós, J. D.: ORB-SLAM: A Versatile and Accurate Monocular SLAM System, *IEEE Transactions on Robotics*, Vol. 31, No. 5, pp. 1147–1163 (online), DOI: 10.1109/TRO.2015.2463671 (2015).

[4] Häming, K. and Peters, G.: The structure-from-motion reconstruction pipeline - a survey with focus on short image sequences, *Kybernetika*, Vol. 46, No. 5, pp. 926–937 (online), available from ⟨http://eudml.org/doc/197165⟩ (2010).

[5] Choy, C. B., Xu, D., Gwak, J., Chen, K. and Savarese, S.: 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction, *Proceedings of the European Conference on Computer Vision (ECCV)* (2016).

[6] Fan, H., Su, H. and Guibas, L. J.: A Point Set Generation Network for 3D Object Reconstruction From a Single Image, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

[7] Jiang, L., Shi, S., Qi, X. and Jia, J.: GAL: Geometric Adversarial Loss for Single-View 3D-Object Reconstruction, *The European Conference on Computer Vision (ECCV)* (2018).

[8] Bishop, C. M.: Mixture density networks, Technical report (1994).

[9] Williams, P.: Using Neural Networks to Model Conditional Multivariate Densities, *Neural computation*, Vol. 8, pp. 843–54 (online), DOI: 10.1162/neco.1996.8.4.843 (1996).

[10] 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).