Vol.2020-CG-180 No.21
Vol.2020-DCC-26 No.21
Vol.2020-CVIM-223 No.21
2020/11/6

単一画像からの食事 (食器含む) と食器単体の三次元形状の同時復元を用いた食事領域の体積推定

成冨 志優[1,a)]　柳井 啓司[1,b)]

概要：食事のカロリー量管理は近年重要なトピックであり, 画像ベースの食品カロリー量推定に関するさまざまな方法とアプリケーションがマルチメディアコミュニティで公開されている. 食事のカロリー量を推定する既存の方法のほとんどは, 2D ベースの画像認識を用いている. 一方この論文では, より正確な推定のために, 3D ボリュームに基づいて推論を行うため, 単一食事画像から食事 (食器含む) と食器単体の三次元再構成を行い, 推定された食事と食器の 3D Mesh の食器部分の一貫性を保ちながら, 高精度に三次元形状を復元することに成功した. これを達成するために, 本論文では以下の貢献を行った. (1) 単一画像から２つの 3D Mesh を生成する, 新しいネットワークである “Hungry Networks” の作成. (2) 再構成される 2 つのモデルの食器部分の三次元形状の一貫性を保つための 3D shape consistency loss の導入. (3) 実際の食事と食器を 3D スキャンして作成された, 新しい 3D モデルのデータセットの作成.

キーワード：食事カロリー量推定, 画像認識, 三次元再構成

1. はじめに

食事管理のために, 食事のカロリー量を正確に推定するには, 可食部分の量を考慮する必要がある. マルチメディアコミュニティでは, 画像ベースの食事カロリー推定に関するさまざまな手法やアプリケーションが公開されている. 既存の食品のカロリー量を推定する方法のほとんどは画像認識を用いた 2D ベースのものである. 画像認識を用いたものには, カロリー量を直接回帰によって求める手法や [1], [2], 検出およびセグメンテーションを使用して 2D 領域サイズに基づいてカロリー量を推定するもの [3], [4] などが存在する. ただし, ほとんどの画像ベースの方法では, 食事の実際のサイズを推定することはできない. そのため, 正確な食事カロリー推定のために, サイズが既知の基準物体を用いる手法が一般的に用いられている. 最近のいくつかの研究では, 基準物体なしでで正確な実際の食事のサイズを推定するために AR/MR デバイスを用いているものも存在する [5], [6]. ただし, 実際の食事は 3D であるため, 2D ベースの方法によるカロリー推定の精度には限界がある. そのため 3D ベースの方法もこれまで検討されてきた. 深度推定 CNN を用いるものや [7], [8], 最近のスマートフォンに搭載された深度カメラを使用して単一の深度画像から食品の 3D ボリュームを推定するものが存在する [9]. しかしこれらの研究では, 食事は平らな食器の上にあると想定されていた. また, さまざまな視点から複数の深度画像が存在する場合, Fusion アルゴリズム [10], [11] を使用してより正確な形状を取得できる. しかし, 実際のアプリケーションとして使用するためには様々な角度から撮影しなければならず, 現実的ではない. そこで本研究では, 1 枚の 2D 画像から食事 (食器含む) と食器単体の両方を同時に三次元再構成するネットワークである “Hungry Networks” を提案する. 食事と食器の体積の差を利用することで, 一般的には入手困難な食品領域の体積を取得できる. また, 2 つの体積の差をより正確に推定するために, 2 つの出力される 3D モデルの食器部分を一致させるための新しい損失関数である 3D shape consistency loss を導入した. 3D スキャンして取得された Mesh データにはノイズと欠陥が含まれているため, 食事の体積と（食品と食器）と食器の体積（食品を含まない）の間の食器形状は完全には一致しないため, この新しい損失関数は有効である. なお, 食品のみの三次元データセットを作成することは困難であるため, 食品のみの三次元形状を直接推定を行わない. そのため, 単一の食品画像から直接食品の量を推定するのではなく, 食事と食器の三次元形状を同時に復元する.

[1] 電気通信大学大学院情報理工学研究科
Department of Informatics, The University of Electro-Communications
[a)] naritomi-s@mm.inf.uec.ac.jp
[b)] yanai@cs.uec.ac.jp

Vol.2020-CG-180 No.21
Vol.2020-DCC-26 No.21
Vol.2020-CVIM-223 No.21
2020/11/6

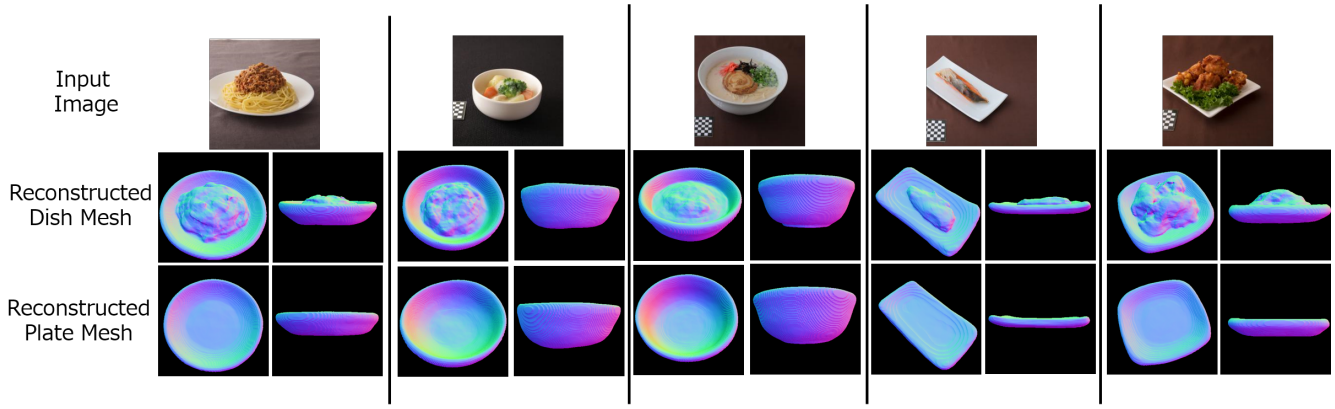

図 1 本物の食事画像からの三次元再構成の結果. ResNet18, $\lambda_3 = 20$, 背景付きのレンダリング画像で学習

一部の既存の食事データセットには深度画像が含まれているが, 食品の完全な 3D 形状 (Mesh) を含むものは存在しない. そのため, 今回の研究では市販の 3D スキャナーで食事と食器を 3D スキャナでスキャンし, 3D Mesh を含む食事データセットを作成した. 対応する食事の画像は, スキャンされた 3D モデルをレンダリングすることによって作成した. また, レンダリングされた画像で学習したネットワークが実際の食事の画像を正しく三次元再構築できるかどうかを実験した. 本論文での貢献は次のとおり.

- 単一画像から 2 つのモデルを生成する, 新しいネットワークである Hungry Networks の作成.
- 再構成される 2 つのモデルの食器部分の形状を一致させる 3D shape consistency loss の導入.
- 実際の食事と食器を 3D スキャンした 3D モデルの新しいデータセットの作成.

2. Related work

2.1 3D shape reconstruction from a single image

画像から三次元形状を再構成する手法は大きく分類すると 3 つある. ボクセル, 点群, Mesh のどの 3 次元表現で再構成するかである. ボクセルを出力する手法 [12], [13] は GPU のメモリを非常に使うため, 低い解像度でしか再構成ができない. またボクセル表現で高い解像度の出力を得ようとする場合, 実装が非常に複雑になる. 点群の出力表現 [14] はただ単に点の集合を出力するため, プログラム上で再構成した物体の形状を得るためには点同士の接続を別途計算しなければならない. Mesh 表現での出力は Mesh テンプレートを使用する手法 [15] や, Geometry Image を用いるもの [16], Mesh テンプレートを必要としない Occupancy ベースの手法 [17], [18] や SDF 表現の手法 [19] などがある. Mesh 表現は点とそれらの接続エッジと面から構成されるので, ボクセルに比べメモリ効率よく高解像度にできる. また点群と違い点同士の接続情報もあるので形状も取れるなど, Mesh には利点が多い.

2.2 Food recognition considering 3D shapes

三次元形状, ないし体積を考慮した食事に関する研究を紹介する. どの研究も最終的な目標はカロリー量や成分量を推定するために行われている. Chen らの手法 [20] では深度センサを用いて深度画像を撮影し, 食事のカロリー量を推定を行っている. また古典的な複数視点によるカメラ行列を推定することで三次元形状を復元する手法として, Puri ら [21] の手法や, DietCam [22] などが存在する. 近年では CNN を用いた研究が発展している. Lu ら.[8] は深度画像をニューラルネットワークを用いて生成し, 生成した深度画像から食事の量を推論しようとした. Im2calorie[7] ではカラー画像からボクセル形式で三次元形状を推定し, カロリー量推定を行おうとしている.

3. 手法

本研究では単一の食事画像から三次元形状を復元を行う. 三次元復元の手法は主にボクセル, ポイントクラウド, Mesh の 3 つに分類できるが, この研究では Mesh を対象に研究を行う. 最終的には単一食事画像から食事付き食器と食器のみの三次元形状を復元し, その体積の差分を求めてカロリー量を推定することを目標としているためだ. 最終的な目標のためには, ボクセルの場合は高解像度に行わなければならないが, これは計算資源に対するコストが非常に高い. また点群の場合は表面の形状を後処理で接続しなければ体積を求めるのには使えない. なので最初から接続が考慮された Mesh 表現で復元することが望ましい. その中でも, 体積を簡単に考慮できるようにするため, 水密かつ自己交差の無い Mesh を生成する手法が適している. また, 本研究では生成される 2 つの Mesh の皿の部分の一貫性を保てなければいけない. つまり, 本研究では, 生成する Mesh に対して, 以下の条件を満たすような設計を目標とした.

- 生成された Mesh が水密であり, ポリゴンに重複がない (自己交差がない) こと

Vol.2020-CG-180 No.21
Vol.2020-DCC-26 No.21
Vol.2020-CVIM-223 No.21
2020/11/6

- 生成された 2 つの Mesh の食器の形状に一貫性があること

3.1 要件を満たす三次元表現

一つめの条件である, 出力された Mesh が水密であり, 自己交差がないという制約とても重要である. なぜなら, Mesh が水密かつ, 自己交差のなく, 各面 $f \in Faces$ が面の表から見た時に反時計回りに $(v1, v2, v3)$ から構成される時, 以下の式 1 で比較的簡単に体積を求めることができるからである.

$$V = \sum_{f \in Faces} \det \begin{vmatrix} v1 & v2 & v3 \end{vmatrix} \qquad (1)$$

なので, まずひとつめの条件である, 水密で自己交差が無い Mesh を生成するためには, テンプレート Mesh を用いる方法を用いることは難しい. なぜなら簡単に自己交差を起こしてしまうからである. それに対して, Marching cubes を用いて Mesh を抽出する手法 [17], [18], [19], [23] では, 水密かつ, 自己交差のない Mesh を生成することができる.

なので, ネットワークの出力表現としては, 占有率や Signed Distance Field (SDF) を用いることが望ましい. 次に, 2 つめの条件である生成された食事と食器の 2 つの Mesh の食器の形状に一貫性があることについての設計を考える. なぜこの一貫性を考慮しないといけないかというと, 学習に用いる実際の食事の 3D データには, ノイズや欠損が当然含まれるため, 食器の形状が一致しないことがよくあるためである. そこで, この問題に対処するにあたり, 占有率か SDF がどちらかが良いかを考える. 占有率は, ある点 $p \in \mathbb{R}^3$ が Mesh の内側にあるか, 外側にあるか, という表現 , SDF は Mesh の表面からどれだけ離れているか, という表現である. 食器の一貫性に関するこの問題は, より噛み砕いて考えると, 食器と出力されている Mesh の内部に含まれる点 p が, 食事として出力される Mesh の内側に含まれていない事にある. つまり, この問題に自然に対処するためには, SDF より占有率を用いる方が適切である.

そこで, 本研究では, Occupancy Networks [17] に基づく占有ベースの手法を用いて, 単一の食事画像から食事と食器の両方の Mesh を復元するネットワークである “HungryNetworks” を提案する. さらに, 復元された 2 つの Mesh の三次元形状を一致させるさせるために, 新しい損失関数である 3D shape consistency loss を提案する. なお, 本論文での食事とは, 図 1 に示すように, 食器と食器の上の食べ物の組み合わたものを意味することに注意する必要がある.

3.2 Hungry Networks

提案手法である Hungry Networks の概要図を図 2 に示した. ネットワークは一つのエンコーダと 2 つのエンコーダから構成されている. エンコーダーは ResNet などの学習

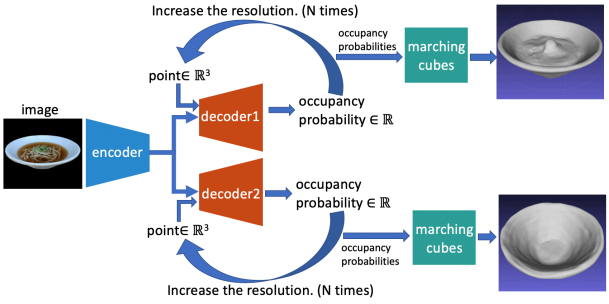

図 2 “Hungry Networks” の概要

済みのネットワークを用いて, 食事画像の特徴量を抽出. エンコーダのネットワークの最終レイヤには Global Average Pooling を用いて 1 次元のベクトルにする. デコーダの入力には画像の特徴量と, 3 次元の点である $p \in \mathbb{R}^3$ を用いる. デコーダーは, 食事（食品部分と食器部分を含む）と食器の占有率をそれぞれ出力する. 占有率は, 各三次元点が Mesh の外側にあるか内側にあるか Mesh を 0/1 のバイナリで表現されている. 図 2 の Decoder-1 は食事の 3D Mesh を生成するための占有率を学習し, Decoder-2 は食器についての占有率をそれぞれ学習する. Mesh の生成の手法は Occupancy Networks [17] と同様のアルゴリズムを用いる. まず最初に,$32 \times 32 \times 32$ の初期解像度で占有確率を推論する. 次に生成したい物体の境界部分のみの解像度を高めて再度占有率を推論し, 高い解像度で物体の境界部分のみの占有確率を再度求める. 解像度を一度上げるごとにグリッドを 8 分割し, $32 \times 32 \times 32 \Rightarrow 64 \times 64 \times 64 \Rightarrow 128 \times 128 \times 128$ のように解像度を上げる. ボクセル表現と違い, 高い解像度でも全ての点を求めず, オブジェクトの境界面のみ段階的に解像度を上げていくため, メモリ効率が非常によい. そうして高い解像度で得られた occupancy filde を Marching cubes アルゴリズム [24] の入力とし, 等値面を Mesh として抽出する. このアルゴリズムでは水密で自己交差のない Mesh が必ず生成できるため, 本研究の一つめの目標を達成することができる.

3.3 学習

ネットワークの学習方法について説明する. $p \in \mathbb{R}^3$ を入力の点, X を入力画像の特徴量とし, 食事と食器のデコーダーネットワークは, それぞれ $f_{d1}(x, p)$ と $f_{d2}(x, p)$ と表す. また点 p に対応する占有率を $o \in \mathbb{R}$ とする. 今回の占有率の学習は, Mesh の内側にあるか, 外側にあるか (それぞれ 1 と 0 に対応する) の二値分類問題と等価である. なので, 占有率を学習するための損失関数は式 2 のようになる. 2 値分類に帰着するため, 損失関数にはバイナリークロスエントロピーロスを用いる.

$$\mathcal{L}_{\mathcal{O}}(f_d(x, p), o(p)) = \mathcal{L}_{bce}(f_d(x, p), o(p)) \qquad (2)$$

次に出力される 2 つの Mesh の食器部分を一致させるた

表 1　占有率のとり方のパターン

dish occupancy ($f_{d1}(p)$)	plate occupancy ($f_{d2}(p)$)	$f_{d2}(p) - f_{d1}(p)$
0	0	0
1	0	-1
0	1	1
1	1	0

めの損失関数である 3D shape consistency loss を導入する．まず, 食事と食器上の対応する点の占有率の組み合わせの可能なパターンを表 1 に示しめした．対応する点での両方のモデルの占有率が同じである場合は問題ない．また, 食事の占有率が 1 で, 食器の占有率が 0 の条件は, 食事モデルの食品部分に対応しているため, 問題ない．しかし, 食事の占有率が 0 で, 食器の占有率が 1 である条件は, 食事の 3D モデルと食器の 3D モデルの食器部分が一致していないことを意味するため, 問題であるから, 解決する必要がある．学習中にペナルティを適用したいのは, 食事の占有率が 0 で, 食器の占有率が 1 の場合のみであり, これは表 1 で $f_{d2}(p) - f_{d1}(p)$ が 1 の場合に対応する．なので, $\max(f_{d2}(p) - f_{d1}(p), 0)$ を最小化するよう学習を行う．そのため損失関数として以下の式を定義する．

$$\mathcal{L}_{\mathcal{C}}(f_{d1}(p), f_{d2}(p)) = \max(f_{d2}(p) - f_{d1}(p)\ , 0) \qquad (3)$$

これを “3D shape consistency loss” と呼ぶことにする．

以上の 2 つの式 (2),(3) を纏めて, 学習全体のミニバッチごとのロス $\mathcal{L}_{\mathcal{B}}$ を定める．ここで, $\mathcal{B}$ はサンプリングしたミニバッチ, I_i はバッチの i 番目の画像であり, バッチ i 番目から全部で K 個の点をサンプリングし, サンプリングした j 番目の点を $p_{i,j}$, f_e は画像特徴量を出力するエンコーダ, f_{d1}, f_{d2} はそれぞれ食事と食器の占有率を出力するデコーダであるとする．

$$x_i = f_e(I_i) \qquad (4)$$

$$y1_{i,j} = f_{d1}(x_i, p_{i,j}) \qquad (5)$$

$$y2_{i,j} = f_{d2}(x_i, p_{i,j}) \qquad (6)$$

$$\mathcal{L}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\sum_{j=1}^{K}\Bigg(\lambda_1\mathcal{L}_{\mathcal{O}}(y1_{i,j}, o1_i(p_{i,j})) + \lambda_2\mathcal{L}_{\mathcal{O}}(y2_{i,j}, o2_i(p_{i,j})) + \lambda_3\mathcal{L}_{\mathcal{C}}(y1_{i,j}, y2_{i,j})\Bigg) \qquad (7)$$

4. データセットの構築

既存の食事データセットにはカラー画像の他に深度画像を含むものは存在するが [25], 3D Mesh の食事のデータセットは存在しない．そこで本研究のため, 新しく食事の 3D Mesh を含むデータセットを作成した．今回作成したデータセットは食事の 3D モデル 240 個, 食器のモデル 38 個で構成されている．モデルの作成には, “Structure Sensor” と呼ばれる市販の 3D スキャナ, 及び専用の 3D スキャンアプリケーションを使用した．異なる食事に対しても同じ食器を利用し撮影したため, 食事のモデルに対して食器のモデルが少なくなっている．

4.1 スキャンしたデータを学習可能にする手順

近年ニューラルネットワークの学習によく使用される多くの 3D モデルデータセットは, 基本的に 3D モデリングソフトを使用して人の手で作成されたデータで構成されている．しかし, 本研究で作成されたデータセットは, 市販の 3D スキャナで実際のオブジェクトをスキャンして作成した．スキャナから出力される Mesh にはノイズや欠陥が含まれているため, ニューラルネットワークの学習にそのまま使用することはできない．そのため, スキャンしたモデルを学習可能にするために幾つかの前処理が必要である．また, 本研究では, 食事と食器の 2 つのモデルの位置をあわせない限り, 同じ座標での占有率を比較することを前提とした 3D shape consistency loss を用いることができない．スキャンによって作成された 3D モデルには 5 つの問題が存在した．

(1) 3D モデルの中心が原点と一致しない
(2) 水密でない
(3) サイズが統一されていない
(4) ノイズが含まれている
(5) 食事の Mesh の食器部分が食器の Mesh の座標と一致していない

4.1.1 3D モデルを原点に合わせる

スキャンされたモデルは 3D 空間のどこかに配置されており, 場所は統一されていない．そのため, まず最初にすべての 3D モデルを中心が 3D 空間の原点になるよう, 位置合わせを行う．

4.1.2 Mesh の欠陥の補完

本研究用いるネットワークでは占有率を推論するように設計している．そのためモデルが水密なモデルでないと, モデルの外側か内側かの定義ができないため訓練データを作成できない．しかしスキャンした 3D モデルには図 3 の左上の図のようにノイズや欠陥が含まれるため, これらを水密なモデルにする必要がある．

穴が空いた 3D モデルを埋めるアルゴリズムはいくつか存在する [26], [27], [28]．本研究では Poisson Surface Reconstruction [27], [28] を用いた．しかしこのアルゴリズムをそのままモデルに適用はできない場合がある．そのままモデルに適応した場合の結果を図 3 に示した．モデルに Poisson Surface Reconstruction を適用した場合, モデルに

Vol.2020-CG-180 No.21
Vol.2020-DCC-26 No.21
Vol.2020-CVIM-223 No.21
2020/11/6

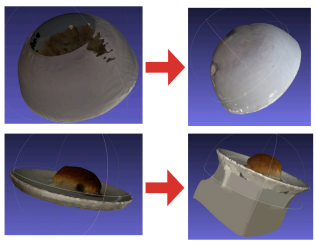

図 3 Poisson Surface Reconstruction の結果．上の行のような小さな穴はうまく補完されるが、大きな穴があるモデルでは失敗してしまう．

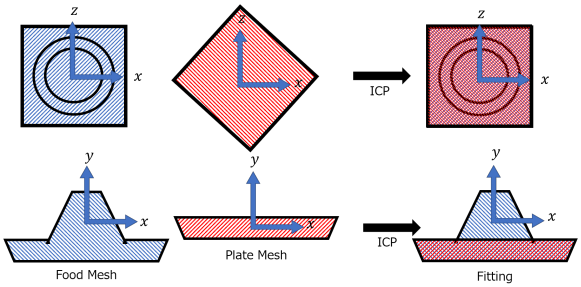

図 4 ICP (Iterative Closest Point) を用いて，食事の Mesh と食器の Mesh の食器部分の座標合わせを行う．

存在する欠損がある程度小さい場合は図 3 の上の図のように，比較的綺麗に面が補完される．しかし図 3 の下の図のように，期待した通りには面が作成されないモデルも多数存在した．Poisson Surface Reconstruction が期待するような面を作成しない主な理由として，欠損が大きすぎるという点である．スキャンした 3D モデルは床との接地面が全て欠損している．そのため平皿のような床との接地面積が広いものは，期待どおりに面が埋まらない．そこで Poisson Surface Reconstruction を適応する前に，ある程度接地面の穴を埋めるアルゴリズムを作成し適用することで穴を埋めた．以上の処理によって，モデルの表面の欠損を埋めた後，モデルのサイズを-0.5〜0.5 に正規化してサイズを統一した．

4.1.3 ノイズの除去

次にモデル内部にノイズが残る問題に対処する．このようなノイズに対して，TSDF Fusion を用いて再び Mesh を再構成することで対処した．TSDF Fusion とは Kinect Fusion [10] で提案された手法の一部を指す．この手法を用いて，モデル内部にあるノイズを完全に取り除いた．

4.1.4 位置合わせ

最後に，食事 (食品と食器を含む) と食器の 2 つの Mesh の位置合わせを行う．今回の学習の 3D shape consistency loss では，皿の領域が同じ領域にあることを前提としている．そこで本研究では図 4 のように，ICP(Iterative Closest Point) を用いて食器と食事の 2 つの Mesh のフィッティングを行い，食器領域の位置をあわせた．

4.2 レンダリングによる入力画像の生成

本研究では 3DR2N2 [13] 同様に blender というソフト

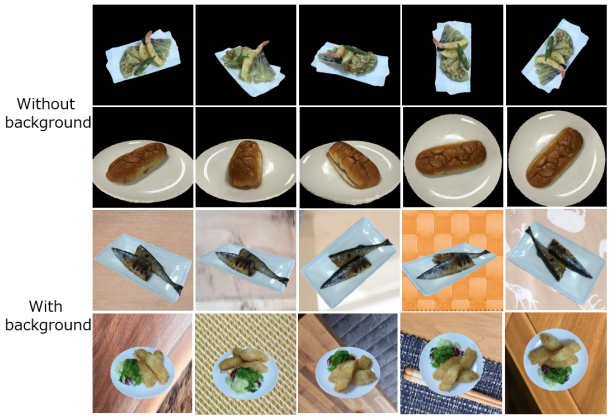

図 5 背景なし/背景ありの学習用にレンダリングされた画像

を用いて学習用の画像を生成した．各モデルあたり 25 枚，様々な角度から撮影された画像を生成した．3DR2N2 で作成された画像は 3D モデルが写っているだけで，背景などは存在しない．しかし，実際のアプリケーションで用いる際に背景が存在しないなどということはない．そこで本研究では生成した食事画像の背景に，web から様々な種類のテーブルやテーブルクロスのテクスチャを背景画像として収集し，合成画像を作成した．図 5 の上の 2 行はモデルがレンダリングされただけの画像で，下 2 行がレンダリングされた画像に背景が合成されたものである．

5. 実験

提案したモデルである “Hungry Networks” を使用して，以下の条件で実験を行った．(1)3D shape consistency loss の重みである式 7 中の λ_3 を 3 通りの値で学習，(2)3 つの異なるバックボーンネットワークで学習，(3) 背景を合成されたレンダリング画像と，合成されていない画像で学習．新しく構築したデータセットのうちから 216 モデルを提案したネットワークの学習に，24 モデルを評価に利用した．ハイパーパラメータである λ_1, λ_2 は 1 で固定し，λ_3 のみを変更して実験を行った．Optimaizer には Adam を利用した．

5.1 Metrics

定量的評価には，Volumetric IoU, Chamber L1 distance, plate consistency, および Volume error を使用する．Volume error は，食事のカロリー量の推定に直接関係しているため，本研究では最も重要である．

Volume IoU は，生成された Mesh と ground-truth Mesh の間の union と intersection の商として定義される．これは，Mesh の境界ボックスの内側から 100,000 点をランダムにサンプリングし，ポイントが内側か外側かを推測することによって計算される．

Chamfer L1 distance は，出力された Mesh 上の点から，ground-truth の Mesh 上の点までの最近傍点までの距離と，ground-truth の Mesh 上から，出力された Mesh 上の点までの最近傍点までの距離との平均で計算される．点のとり

Vol.2020-CG-180 No.21
Vol.2020-DCC-26 No.21
Vol.2020-CVIM-223 No.21
2020/11/6

方は, それぞれの Mesh の表面上から 10 万点サンプリングし, KD-Tree を用いて最近傍法を探索して実行した. これは [17], [29] と同様の計算方法である.

plate consistency score は, 生成された食器の Mesh 上の点から, 生成された食事の Mesh 上の点に対する最近傍法点の距離の平均である. このスコアは生成された 2 つの Mesh の食器の形状がずれていれば, 大きく, 食器の形状が同じであれば小さくなる指標である.

Volume error は, 推論された食品領域の体積と, 正しい食事領域の体積の L1 距離の平均である. 食品の体積は, 食事の体積から食器の体積を差し引くことによって得られる. ground-truth になる食品領域の体積は, 手動で食事の Mesh から食器の Mesh を取り除き, 整形することで作成した. これは非常に時間がかかるため, 評価用の 24 個のモデルのみを対象に行った. で ground-truth の 3D 食品 モデルを作成した.

IoU は高いほうがよく, Chamfer L1 distance, Plate consistency score, Volume error は小さいほうが良い指標である.

5.2 定量評価

まず最初に, λ_3 の影響を評価した. λ_3 の影響のみを評価するため, エンコーダのバックボーンは ResNet34, 学習に用いる画像を背景を合成していないものに固定して学習を行った. 実験では, λ_3 の値を 0, 20, 50 の 3 つの値で実験を行った. なお, λ_3 が 0 ということは, 3D shape consistency loss を用いない事と等価である. 結果を表 2 に示した. その結果, 3D shape consictency loss を用いると Volume error のスコアに大きく貢献していることが判明した. 一方, 食事と食器の IoU や Chamfer L1 distance などスコアは, 3D shape consictency loss を使用しない場合が最も精度が良いという結果になった. しかし, 図 7 に示すように, 3D shape consictency loss を用いない場合, 食事の Mesh と食器の Mesh に皿の領域が異なって再構成されてしまっていた. 食事用デコーダ, 食器用デコーダともに, 互いに独立した Binary cross entropy で最適化しているため, IoU などの個別の評価が良くなり, 体積誤差などの統合評価が悪くなる傾向がある.

次に表から, 体積の推定がもっとも正確であった $\lambda_3 = 20$ を用いて, エンコーダのバックボーンを ResNet18, ResNet34, ResNet50 に変えて背景の合成されていない画像で学習し, それぞれで評価した. 結果を 3 に示した. 結果としては, ResNet18 と ResNet50 の差は非常に小さいものの, 食品の体積誤差は ResNet50 が最も正確であった.

次の実験では, 学習画像に背景を合成するのとしないのとで, どれだけ精度に影響するかを評価した. バックボーンには ResNet18 と ResNet50, λ_3 は 20 を用いた. 結果を表 4 に示した. 学習画像に背景が合成されている場合のほうが, Volume error と plate consistency score が最も良好な結果が得られた.

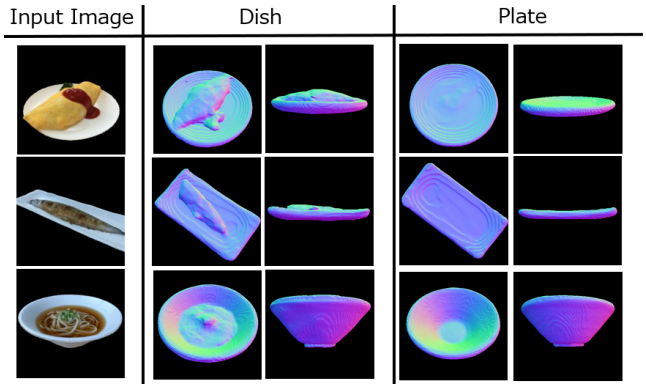

図 6 食事と食器の三次元形状の推論結果. バックボーンに ResNet18, $\lambda_3 = 20$, 背景無しの画像で学習したモデルを利用.

5.3 定性評価

図 6 は $\lambda_3 = 20$, ResNet18 で背景なしの画像で学習されたネットワークで単一画像から推論された食事と食器の Mesh の結果である. 食事と食器の 3D Mesh の両方ともが, 画像に対して正しく推論されている事がわかる. また入力の食事画像には食器だけではなく食品部分も含まれているが, 食器のデコーダが正しく食器のみを復元していることも確認できる. 加えて, 食事と食器の Mesh の双方の食器部分の形状がほとんど同一であることであることが見て取れる.

図 7 に 3D shape consistency loss を用いた場合と, 用いてない場合での結果を比較している. 3D shape consistency loss を用いない場合, 復元された食事と食器の三次元形状は大きく異なっていることがある. これは主に, 学習データに用いている 3D モデルの性質によるものである. スキャンされた 3D Mesh データは, 地面に接触している部分にノイズや欠陥が発生しやすい傾向がある. したがって, モデルを下から見た場合, 2 つの Mesh の生成結果は大幅に異なる場合がある. 一方, 3D shape consistency loss を使用した場合, 両方の Mesh 形状の食器領域に一貫性が維持できていることがわかる.

図 1 には ResNet18, $\lambda_3 = 20$, 背景を合成した画像で学習したネットワークを用いて, レンダリング画像ではない本物の食事写真を入力に三次元再構成した結果を示した. 本物の食事の写真はネットワークの学習に用いられていないが, 背景をレンダリング画像に合成した画像で学習されたネットワークは, 本物の食事の写真にも適用することができた. 四角い平皿, 四角い平皿, 丸い平皿, お椀など様々な種類のお皿が, それぞれ形や高さが大きく異なるものの, 正常に再構成されていることがわかる.

表 2 λ_3 を 3 パターン実験して評価。バックボーンには ResNet34, 背景なしの画像で学習.

λ_3	IoU (dish)	IoU (plate)	Chamfer L1 (dish)	Chamfer L1 (plate)	**plate consistency**	**Volume error**
0	**0.624**	**0.621**	**0.0189**	0.0186	0.0256	0.0252
20	0.550	0.607	0.0262	**0.0182**	0.0168	**0.0155**
50	0.542	0.610	0.0260	0.0209	**0.0152**	0.0161

表 3 バックボーンのネットワークを,ResNet18, ResNet34, ResNet50 の 3 パターンで実験.
すべて $\lambda_3 = 20$, 背景無しの画像で学習.

encoder	IoU (dish)	IoU (plate)	Chamfer L1 (dish)	Chamfer L1 (plate)	Plate consistency score	Volume error
ResNet 18	0.560	**0.634**	0.0265	0.0193	**0.0146**	0.0150
ResNet 34	0.550	0.607	0.0262	**0.0182**	0.0168	0.0155
ResNet 50	**0.564**	0.617	**0.0251**	0.0186	0.0148	**0.0147**

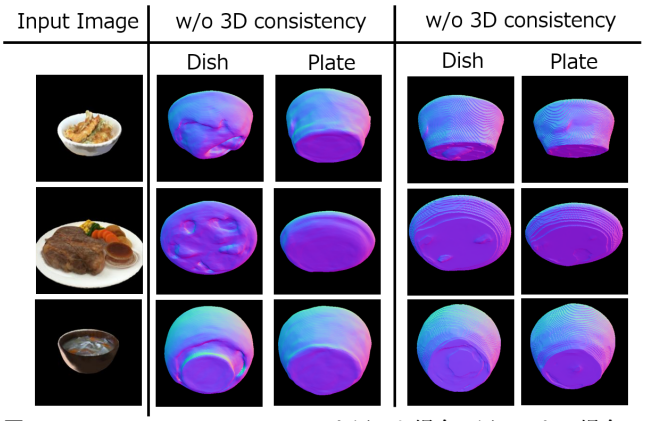

図 7 3D shape consistency loss を用いた場合と用いてない場合での生成結果の比較. バックボーンに ResNet18, $\lambda_3 = 20/0$, 背景無しの画像で学習したモデルを利用

5.4 3D shape consistency loss に関する議論

3D shape consistency loss を導入することは, いくつかのメリットとデメリットの両方の側面が存在する. デメリットとしては, 一般的な評価指標である Chamfer distance などを用いた評価では, 悪くなるということだ. 3D shape consistency loss は ground-truth に近づけるための loss では無いからだ. メリットとしては, 3D スキャナの生成した Mesh のノイズを吸収できる, という点である. 学習データとして用いるデータを廉価な 3D スキャナを用いて作成する場合, それは欠損であったり, 膨らんだりといったノイズがどうしても含まれてしまう. ShapeNet などのデータセットは, 3D モデリングソフトを用いて人が丁寧に作成したモデルであり, 非常に整っているが, 3D スキャナを用いて Mesh を作成した場合は当然そうはいかない. そのため, 食事と食器, 実際には同じ食器であるはずなのに, スキャナの精度により少し違った形状を含んでいる. 食品部分の体積を求めるために, 食事の Mesh の体積から食器の Mesh の体積を引いても食器部分のノイズに影響されて増減してしまう. そこで, 3D shape consistency loss を導入し, 食器形状を一致させることで, 体積を求める精度が上がった.

6. Conclusions

研究では食事の 3D Mesh の食事データセットを作成し, 単一食事画像から食事と食器の三次元形状復元を実現する Hungry Networks を作成した. 学習には食事と食器の Mesh の食器部分の一貫性を保つため新しいロスである 3D shape consistency loss を導入した. 学習にはレンダリング食事画像とレンダリング食事画像と背景画像を合成した画像を用い, 高精度に三次元形状を復元できることを示した. また 3D shape consistency loss を導入することで, 2 つの Mesh の食器部分の一貫性を保ち復元することに成功し, それが食事領域の体積の推定に貢献していることを示した. また, 背景画像を合成した食事画像で学習したネットワークは, リアルの食事画像を入力にしても正しく再構成できることをしめした. 今後の課題としては, 現在の三次元形状復元は, 正規化された空間の中で行われており, 実際の大きさを考慮できていない. カロリー量推定のためには, 実際の大きさを考慮できなければいけない. そこで, AR デバイスの環境認識機能や, 深度画像, 基準物体などを用いて実寸を考慮した三次元形状復元を行い, 正確なカロリー量推定につなげたいと考えている.

参考文献

[1] Ege, T. and Yanai, K.: Image-Based Food Calorie Estimation Using Recipe Information, *IEICE Transactions on Information and Systems*, Vol. E101-D, No. 5, pp. 1333–1341 (2018).
[2] Ege, T. and Yanai, K.: Imag-Based Food Calorie Estimation Using Knowledge on Food Categories, Ingredients and Cooking Directions, *Proc. of ACM Multimedia Thematic Workshop* (2017).
[3] Ege, T. and Yanai, K.: Estimating Food Calories for Multiple-dish Food Photos, *Proc. of Asian Conference on Pattern Recognition* (2017).
[4] Ege, T. and Yanai, K.: Multi-task Learning of Dish Detection and Calorie Estimation, *Proc. of IJCAI and ECAI Workshop on Multimedia Assisted Dietary Management* (2018).
[5] Naritomi, S. and Yanai, K.: CalorieCaptorGlass: Food Calorie Estimation Based on Actual Size using HoloLens and Deep Learning, *Proc. of IEEE Conference on Virtual Reality and 3D User Interfaces* (2020).
[6] Tanno, R., Ege, T. and Yanai, K.: AR DeepCalorieCam V2: food calorie estimation with CNN and AR-based

表 4 背景付きの画像となしの画像で学習した結果。バックボーンに ResNet18/50, $\lambda_3 = 20$ で学習.

encoder	background	IoU (dish)	IoU (plate)	Chamfer L1 (dish)	Chamfer L1 (plate)	Plate consistency score	Volume error
ResNet 18	none	0.560	0.634	0.0265	0.0193	**0.0146**	0.0150
ResNet 50	none	0.564	0.617	**0.0251**	0.0186	0.0148	0.0147
ResNet 18	yes	**0.565**	**0.645**	0.0254	**0.0173**	**0.0146**	**0.0146**
ResNet 50	yes	0.558	0.628	0.0252	**0.0173**	0.0157	0.0157

actual size estimation, *Proc. of the 24th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–2 (2018).

[7] Meyers, A., Johnston, N., Rathod, V., Korattikara, A., Gorban, A., Silberman, N., Guadarrama, S., Papandreou, G., Huang, J. and Murphy, K. P.: Im2Calories: towards an automated mobile vision food diary, *Proc. of the IEEE International Conference on Computer Vision*, pp. 1233–1241 (2015).

[8] Lu, Y., Allegra, D., Anthimopoulos, M., Stanco, F., Farinella, G. M. and Mougiakakou, S.: A multi-task learning approach for meal assessment, *Proc. of the Joint Workshop on Multimedia for Cooking and Eating Activities and Multimedia Assisted Dietary Management*, pp. 46–52 (2018).

[9] Ando, Y., Ege, T., Cho, J. and Yanai, K.: DepthCalorieCam: A Mobile Application for Volume-Based FoodCalorie Estimation using Depth Cameras, *Proc. of the 5th International Workshop on Multimedia Assisted Dietary Management*, pp. 76–81 (2019).

[10] Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S. and Fitzgibbon, A.: KinectFusion: Real-time dense surface mapping and tracking, *Proc. of 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 127–136 (2011).

[11] Newcombe, R. A., Fox, D. and Seitz, S. M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 343–352 (2015).

[12] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. and Xiao, J.: 3D ShapeNets: A deep representation for volumetric shapes, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 1912–1920 (2015).

[13] 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, *Proc. of European Conference on Computer Vision*, pp. 628–644 (2016).

[14] Fan, H., Su, H. and Guibas, L. J.: A Point Set Generation Network for 3D Object Reconstruction from a Single Image, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 605–613 (2017).

[15] Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W. and Jiang, Y. G.: Pixel2mesh: Generating 3d mesh models from single rgb images, *Proc. of European Conference on Computer Vision*, pp. 52–67 (2018).

[16] P., A., S., J., C., G. P. T., S., A. and M., F.: 3DPeople: Modeling the Geometry of Dressed Humans, *Proc. of IEEE International Conference on Computer Vision* (2019).

[17] Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S. and Geiger, A.: Occupancy Networks: Learning 3d reconstruction in function space, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 4460–4470 (2019).

[18] Saito, S., Simon, T., Saragih, J. and Joo, H.: PIFuHD: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization, *Proc. of IEEE Computer Vision and Pattern Recognition* (2020).

[19] Park, J. J., Florence, P., Straub, J., Newcombe, R. and Lovegrove, S.: DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation, *Proc. of IEEE Computer Vision and Pattern Recognition* (2019).

[20] Chen, M. Y., Yang, Y. H., Ho, C. J., Wang, S. H., Liu, S. M., Chang, E., Yeh, C. H. and Ouhyoung, M.: Automatic chinese food identification and quantity estimation, *Proc. of SIGGRAPH Asia 2012 Technical Briefs*, pp. 1–4 (2012).

[21] Puri, M., Zhiwei Zhu, Yu, Q., Divakaran, A. and Sawhney, H.: Recognition and volume estimation of food intake using a mobile device, *2009 Workshop on Applications of Computer Vision (WACV)*, pp. 1–8 (2009).

[22] Kong, F. and Tan, J.: DietCam: Regular Shape Food Recognition with a Camera Phone, *2011 International Conference on Body Sensor Networks*, pp. 127–132 (2011).

[23] Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A. and Li, H.: PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization, *Proc. of IEEE International Conference on Computer Vision* (2019).

[24] Lorensen, W. E. and Cline, H. E.: Marching cubes: A high resolution 3D surface construction algorithm, *ACM siggraph computer graphics*, Vol. 21, No. 4, pp. 163–169 (1987).

[25] Ferdinand, C. P., Schlecht, S., Ettlinger, F., Grun, F., Heinle, C., Tatavatry, S., Ahmadi, S. A., Diepold, K. and Menze, B. H.: Diabetes60-Inferring Bread Units From Food Images Using Fully Convolutional Neural Networks, *Proc. of the IEEE International Conference on Computer Vision Workshops*, pp. 1526–1535 (2017).

[26] Calakli, F. and Taubin, G.: SSD: Smooth signed distance surface reconstruction, *Computer Graphics Forum*, Vol. 30, No. 7, pp. 1993–2002 (2011).

[27] Kazhdan, M., Bolitho, M. and Hoppe, H.: Poisson surface reconstruction, *Proc. of the fourth Eurographics symposium on Geometry processing*, Vol. 7 (2006).

[28] Kazhdan, M. and Hoppe, H.: Screened poisson surface reconstruction, *ACM Transactions on Graphics (ToG)*, Vol. 32, No. 3, pp. 1–13 (2013).

[29] Fan, H., Su, H. and Guibas, L.: A Point Set Generation Network for 3D Object Reconstruction from a Single Image, *Proc. of IEEE Computer Vision and Pattern Recognition* (2017).