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

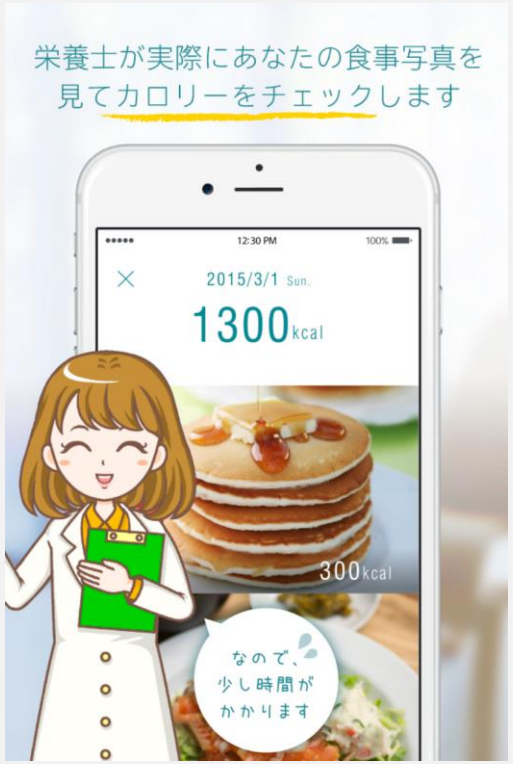
電気通信大学 情報理工学研究科

成冨志優, 柳井 啓司

はじめに

- 食事のカロリー量管理は、実需が高い。

- スマートフォン向けサービスなども存在



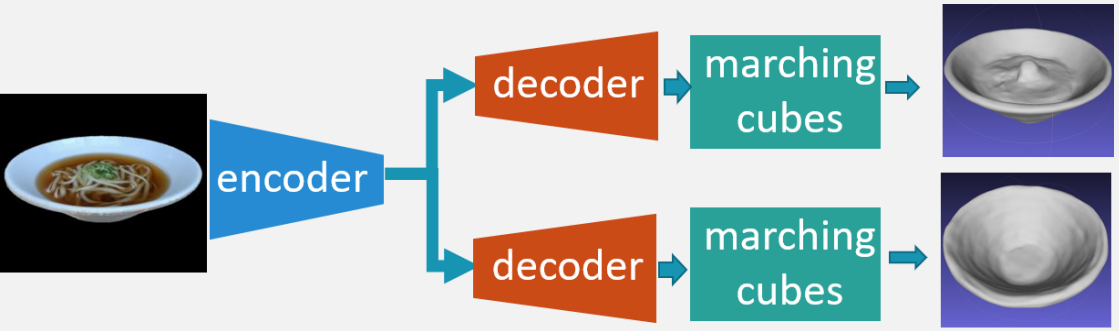
はじめに

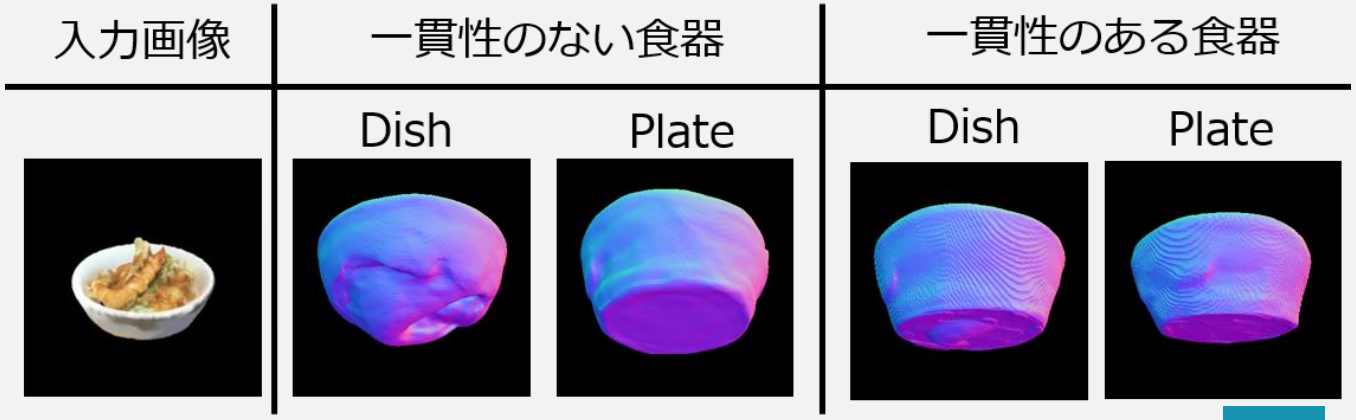
- カロリー量推定はマルチメディア界隈で研究が盛ん。
  - 2Dベースが多い
  - 深度ベース
  - センサーベース

はじめに

- 単一食事画像から
  食事(食品+食器)と食器の2つの三次元形状を別々に復元して食品の体積を推論

- 復元される2つの三次元形状の食器の一貫性も考慮



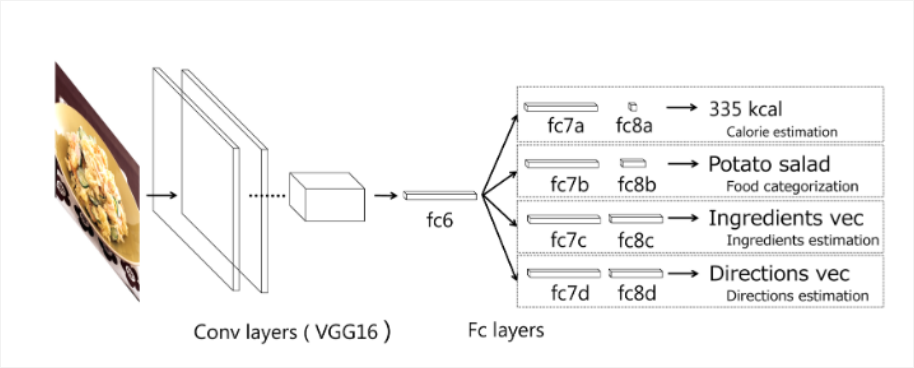


関連研究 : 食事のカロリー量推定

- 画像から直接カロリー量を回帰

  [Egeら, IEICE2018]

  [Egeら, ACM Multimedia Thematic ws]

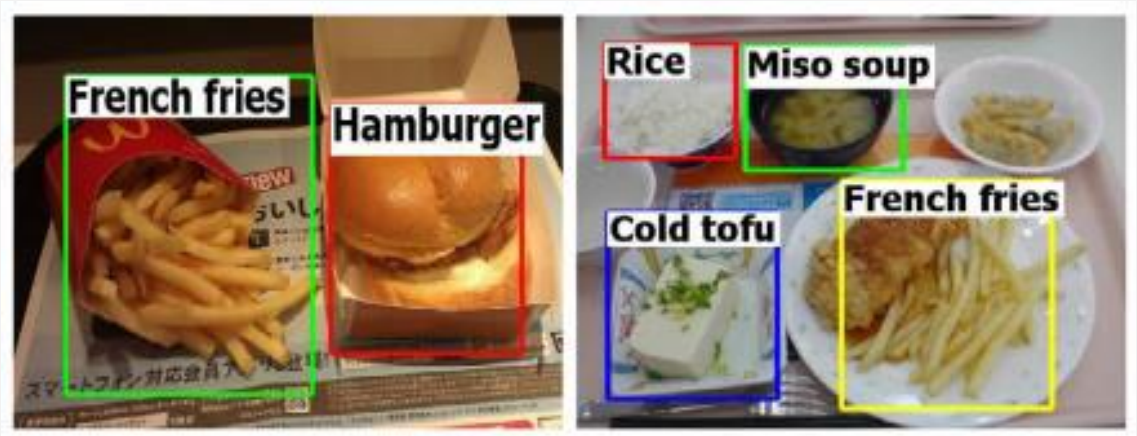


[Egeら, IEICE2018]から引用

- 検出/領域分割を用いる手法

  [Egeら,ACPR 2017]

  [Egeら,MADiMa 2018]



[Egeら,ACPR 2017]から引用

[Egeら, IEICE2018] Ege, T. et al. : Image-Based Food Calorie Estimation Using Recipe Information, *IEICE Transactions on Information and Systems*(2018)

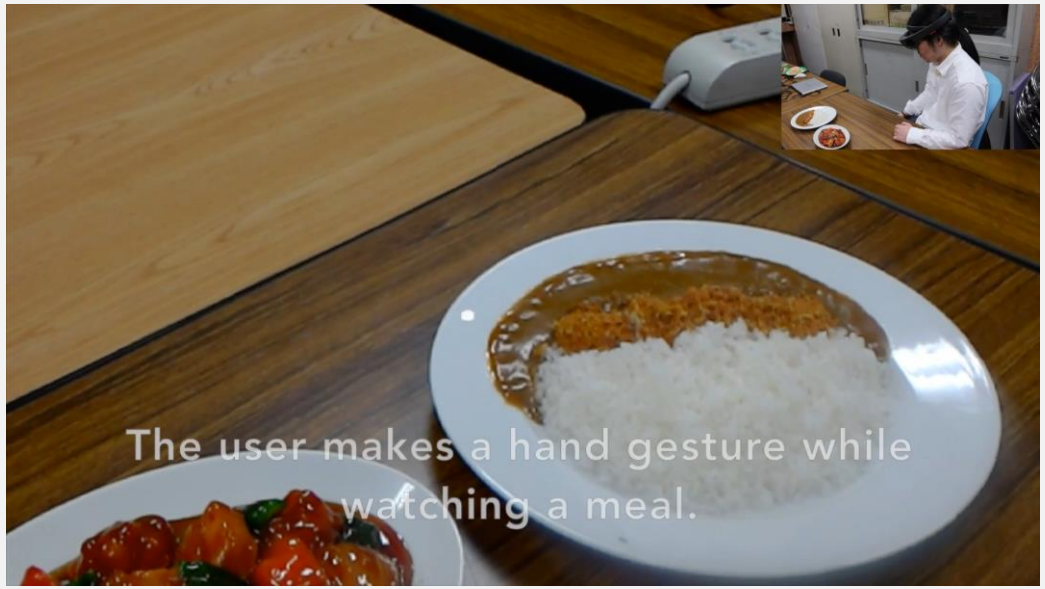
[Egeら, ACM Multimedia Thematic ws] Ege, T. et al. : Imag-Based Food Calorie Estimation Using Knowledge on Food Categories, Ingredients and Cooking Directions, *Proc. of ACM Multimedia Thematic Workshop* (2017).

[Egeら,ACPR 2017] Ege, T. et al. : Estimating Food Calories for Multiple-dish Food Photos, ACPR(2017).

[Egeら,MADiMa 2018]] Ege, T. et al. : Multi-task Learning of Dish Detection and Calorie Estimation, MaDiMa(2018).

関連研究 : 食事のカロリー量推定

- デバイスを用いて実寸を測定してカロリー量を推定
  [CalorieCaptorGlass, IEEE VR 2020]
  [AR DeepCalorieCam V2, VRST2018]
- RGB画像だけでは実スケールは分からない。



[CalorieCaptorGlass, IEEE VR 2020]から引用

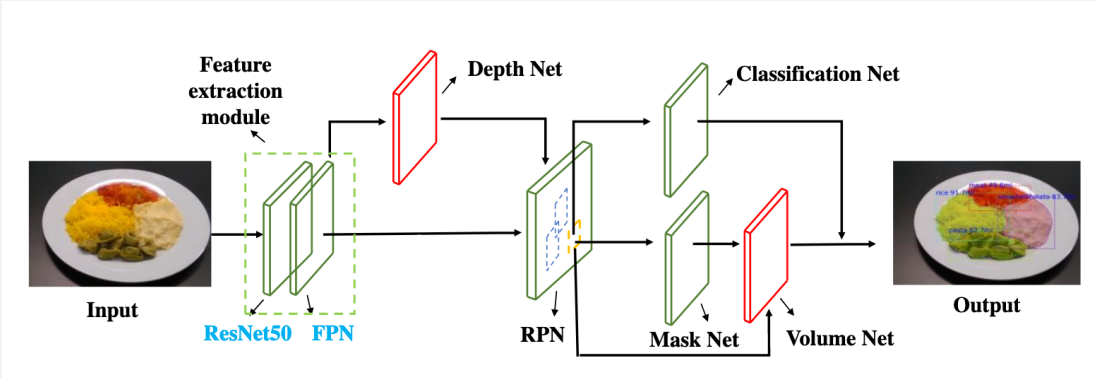
[AR DeepCalorieCam V2, VRST2018]から引用

[CalorieCaptorGlass, IEEE VR 2020] Naritomi, S. et al. : CalorieCaptorGlass: Food Calorie Estimation Based on Actual Size using HoloLens and Deep Learning, IEEE VR (2020)
[AR DeepCalorieCam V2, VRST2018] Tanno, R. et al. : AR DeepCalorieCam V2: food calorie estimation with CNN and AR-based actual size estimation, VRST (2018)

関連研究 : 食事のカロリー量推定

- 食事は3次元。故に、3Dベースが好ましい。
  - 深度画像を使うのが主流。
  - 深度カメラを利用[Andoら, MADiMa2019], CNNを使った深度推定を利用[Luら, MaDiMa2018]

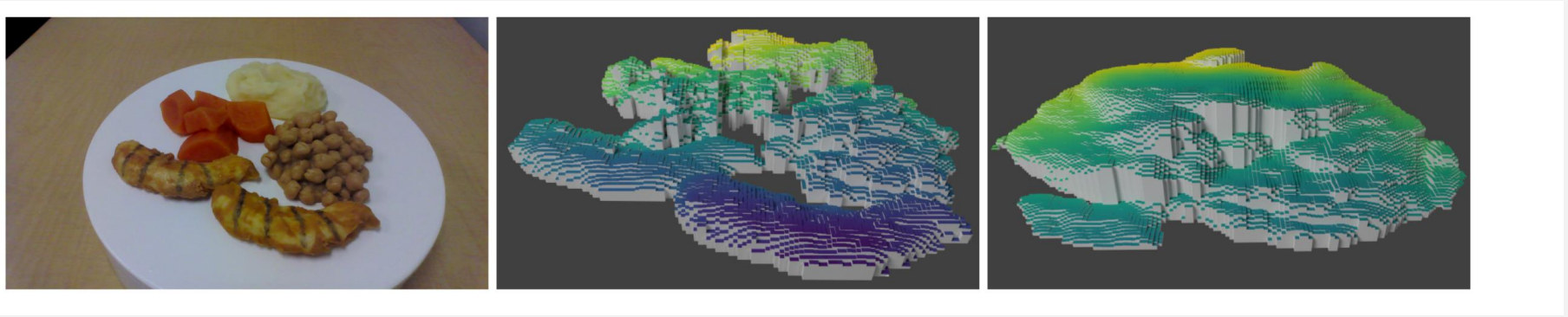
[Andoら, MADiMa2019]から引用



[Luら, MaDiMa2018]から引用

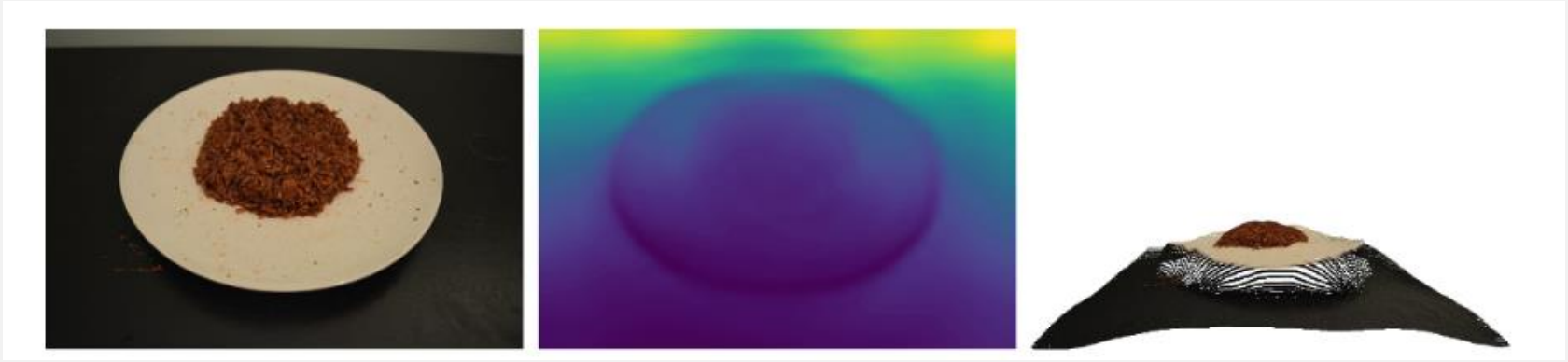
関連研究 : 食事のカロリー量推定

- 深度画像推定 -> ボクセルに変換[Im2Calories, ICCV 2015]



[Im2Calories, ICCV 2015]から引用

- 深度画像推定 -> 点群 -> Meshに変換[Alexandrosら, HCII2020]



[Alexandrosら, HCII2020]から引用

深度画像ベースの手法の問題点

- 全て、平らな食器の上に食品がある制約つき。
- 丼などに対応できない。

関連研究 : 単一画像からの三次元復元

- 単一画像からの三次元復元
  - 深層学習のパワーですごいスピードで進化

- 三次元形状は3つの表現に分類できる
  - ボクセル
  - 点群
  - メッシュ

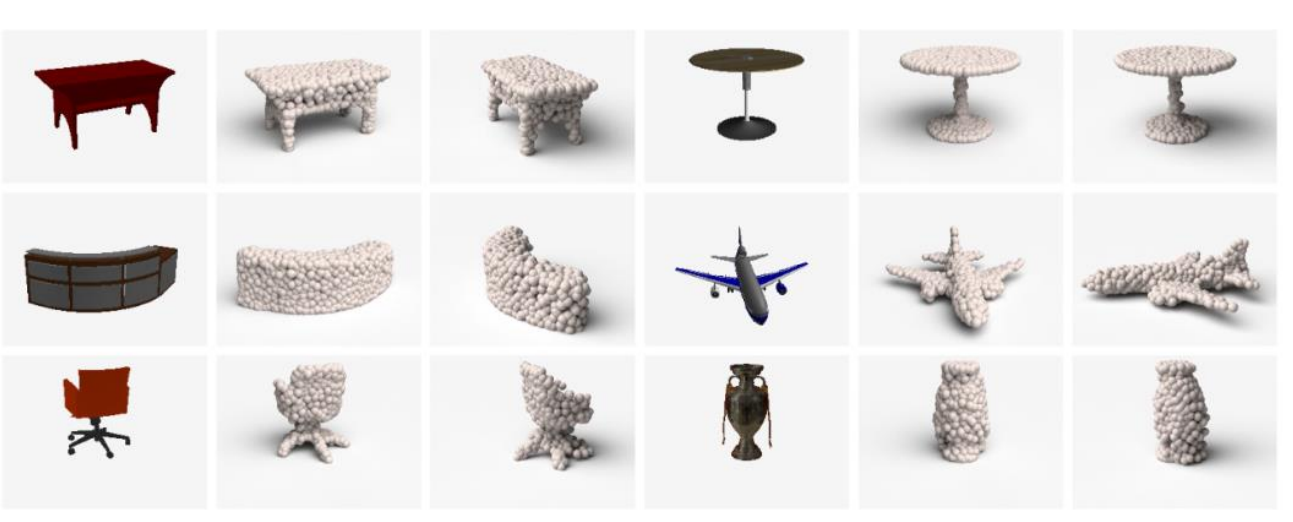
関連研究 : 単一画像からの三次元復元

- ボクセル[wuら, CVPR2015][3D-R2N2, CVPR2015]
  - ○シンプル
  - ○畳み込みとの相性良し。
  - ×メモリを非常に使う。
  - ×高解像度にするのが難しい。

[3D-R2N2, CVPR2015]から引用

関連研究 : 単一画像からの三次元復元

- 点群[Fanら, CVPR2017]
  - ○ メモリ効率
  - ✕ 点と点の接続がわからない。構造がわからない。
  - ✕ 点と点の接続には後処理が必要



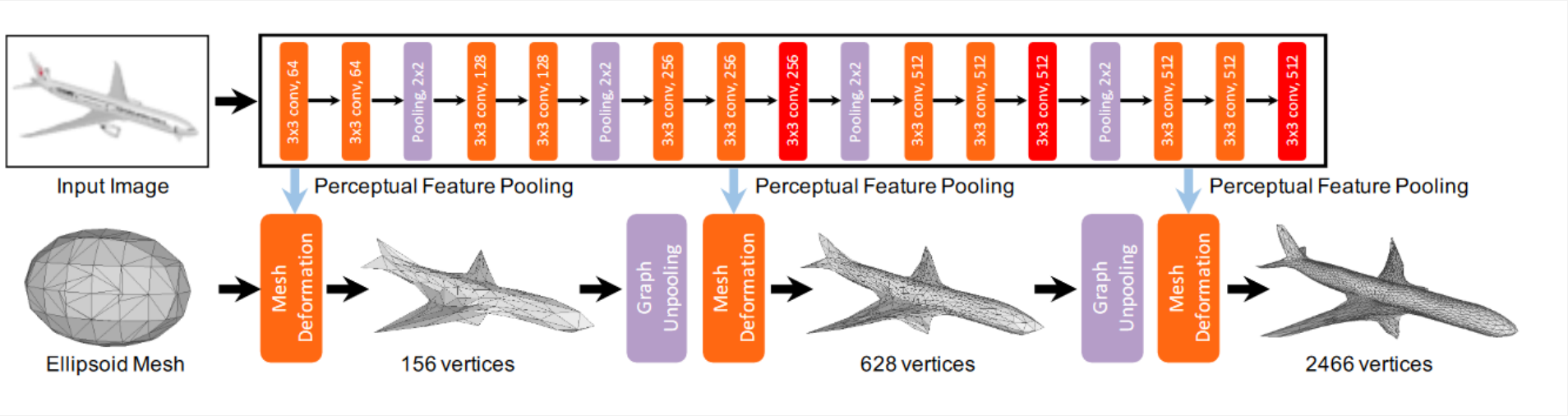
[Fanら, CVPR2017]から引用

関連研究 : 単一画像からの三次元復元

- メッシュ
  - ○ 点と点の接続, 面で構成される。 ← 形状がわかる。
  - ○ ボクセルよりメモリ効率が良い。

関連研究 : 単一画像からの三次元復元

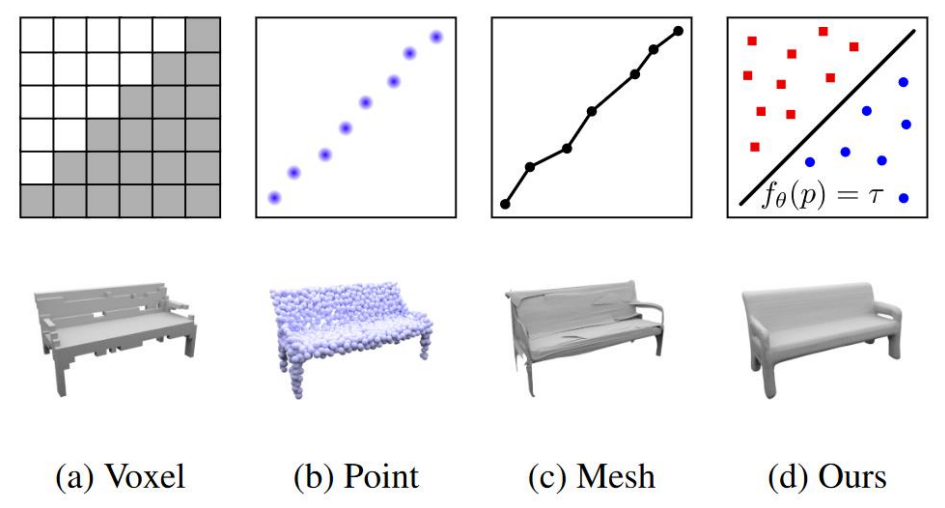
- メッシュテンプレートを利用[Pixel2mesh, ECCV2018]
  - 頂点の最適化



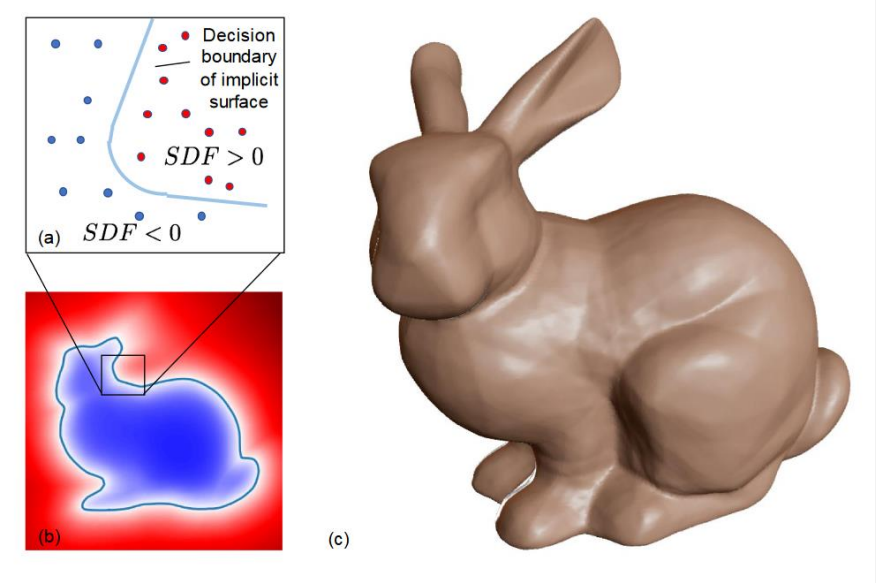
[Pixel2mesh, ECCV2018]から引用

関連研究 : 単一画像からの三次元復元

- Meshテンプレートを必要としない手法。
  - 占有率[Occupancy Networks, CVPR2019][PIFu, ICCV2019]
  - Signed distance field[DeepSDF, CVPR2019]



[Occupancy Networks, CVPR2019] から引用

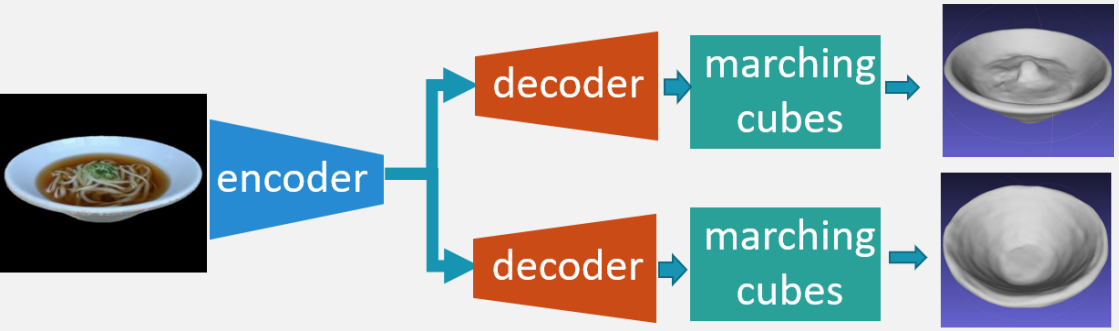


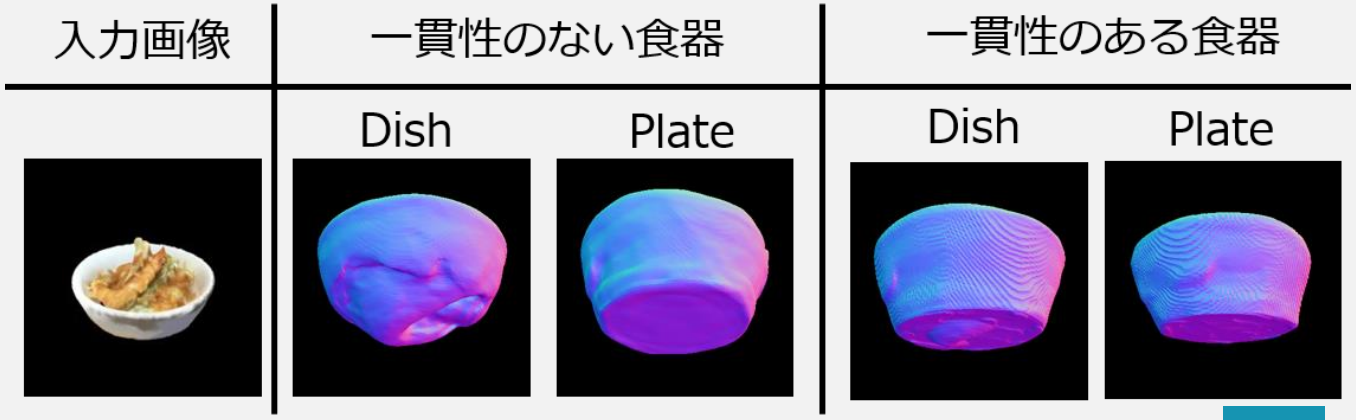
[DeepSDF, CVPR2019]から引用

提案手法(再掲)

- 単一食事画像から
  食事(食品+食器)と食器の2つの三次元形状を別々に復元
  して食品の体積を推論

- 復元される2つの三次元形状の食器の一貫性も考慮





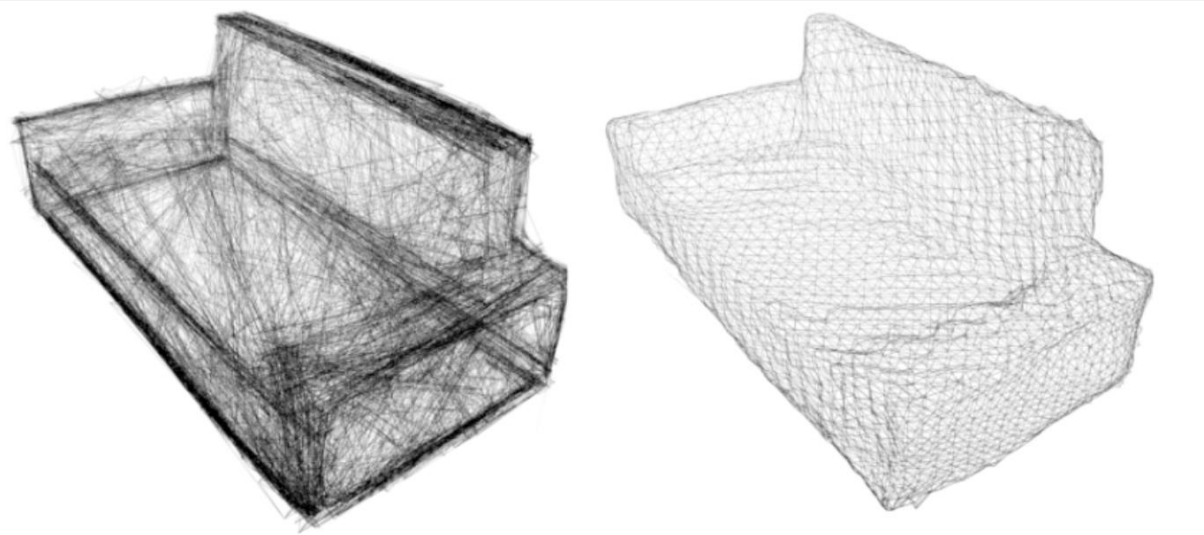
適切な三次元表現

- 食品の体積を求めたい。
  - ボクセル : ✕高解像度に向いてない
  - 点群: ✕ 点間の接続が分からない。体積が分からない。
  - メッシュ: ○ 高解像度かつ、簡単に体積が求められる。

- Meshから体積を求めるための条件
  - 水密であること。
  - 自己交差がない事。

適切な出力表現

- 体積の条件を満たす表現
  - Meshテンプレート : 自己交差が頻発してしまう。
  - 占有率かSDF表現なら条件を満たせる。



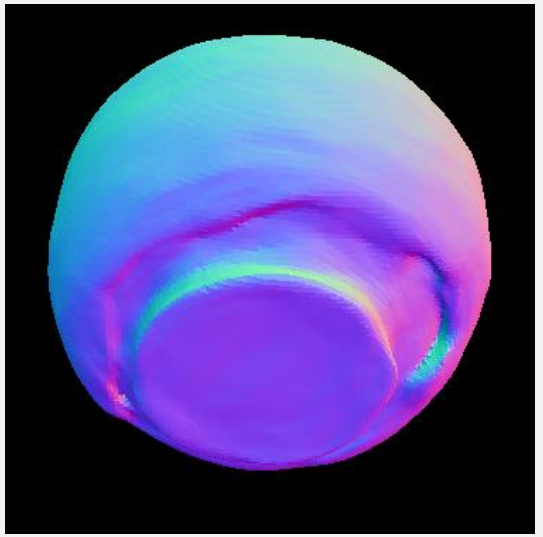
[Mesh R-CNN, ICCV2019] から引用

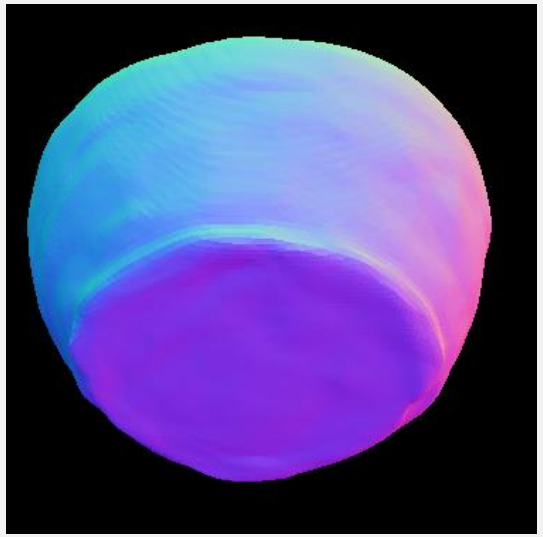
- 食器の一貫性
  - 占有率とSDF、どちらが最適か？

適切な出力表現

- 問題点
  - 食器の内側に含まれる点$p \in R^3$が、食事のメッシュの内側に含まれない
  - 个の逆は問題ない(食品領域は食器メッシュに含まれないため)

- 含まれる/ない、の問題。
  - 占有率が妥当。



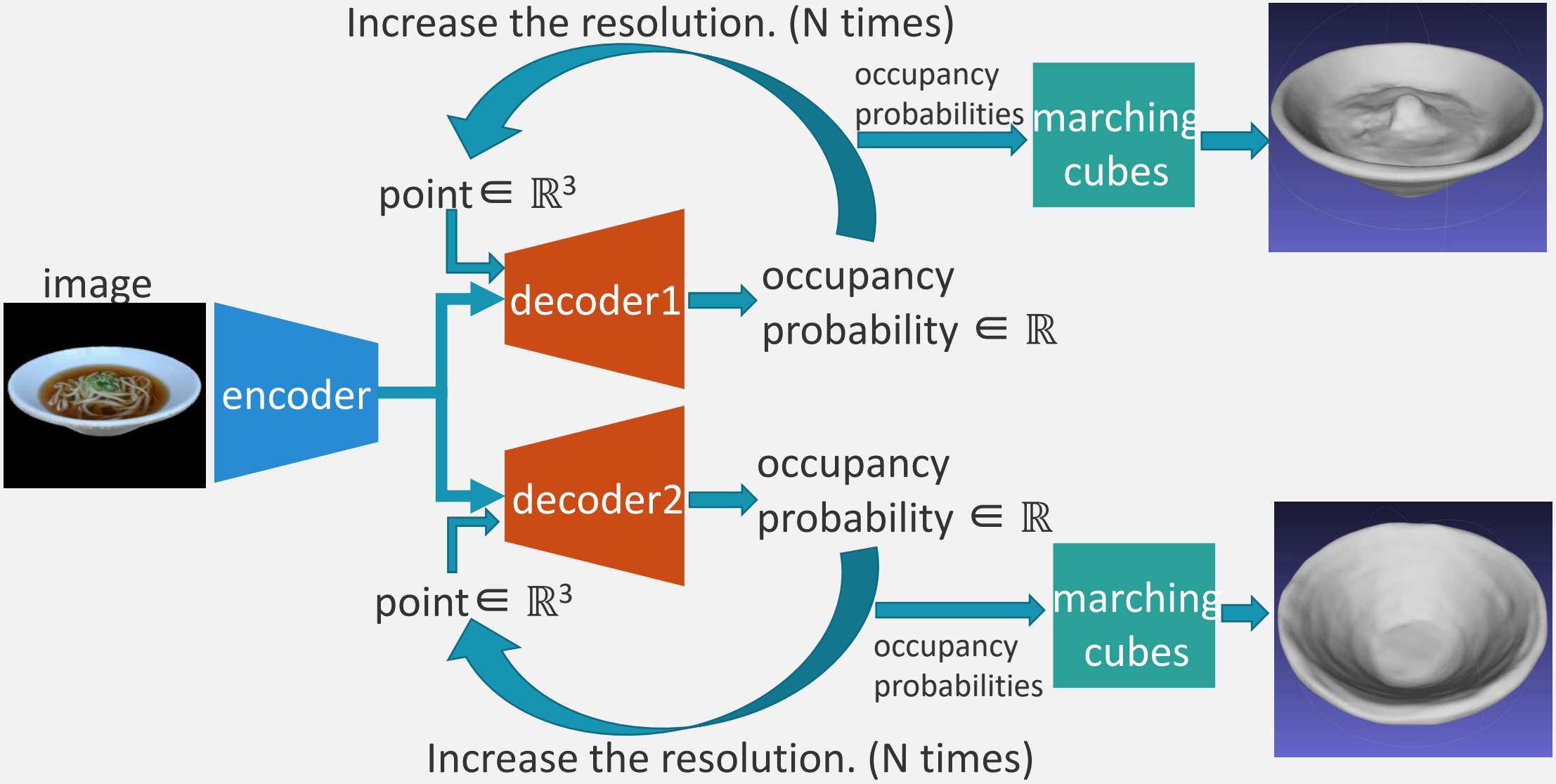


提案手法

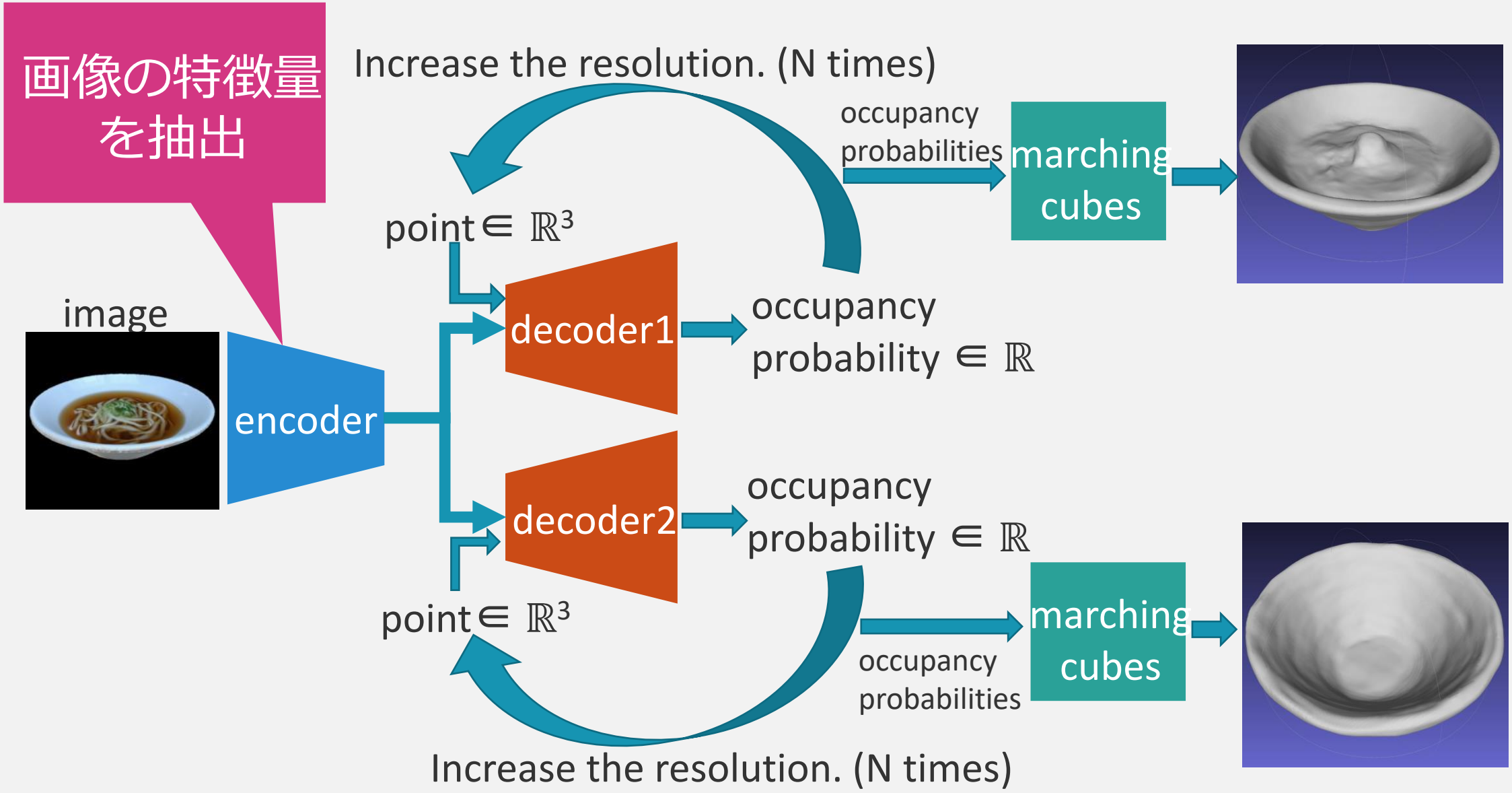
- Hungry Networksの提案
  - 単一の食事画像から、食事と食器の2つのMeshを復元
  - 占有率ベースの手法であるOccupancy Networks [17]を拡張

- 3D shape consistency lossの提案
  - 食事と食器の三次元形状の、食器部分を一致させるための損失関数

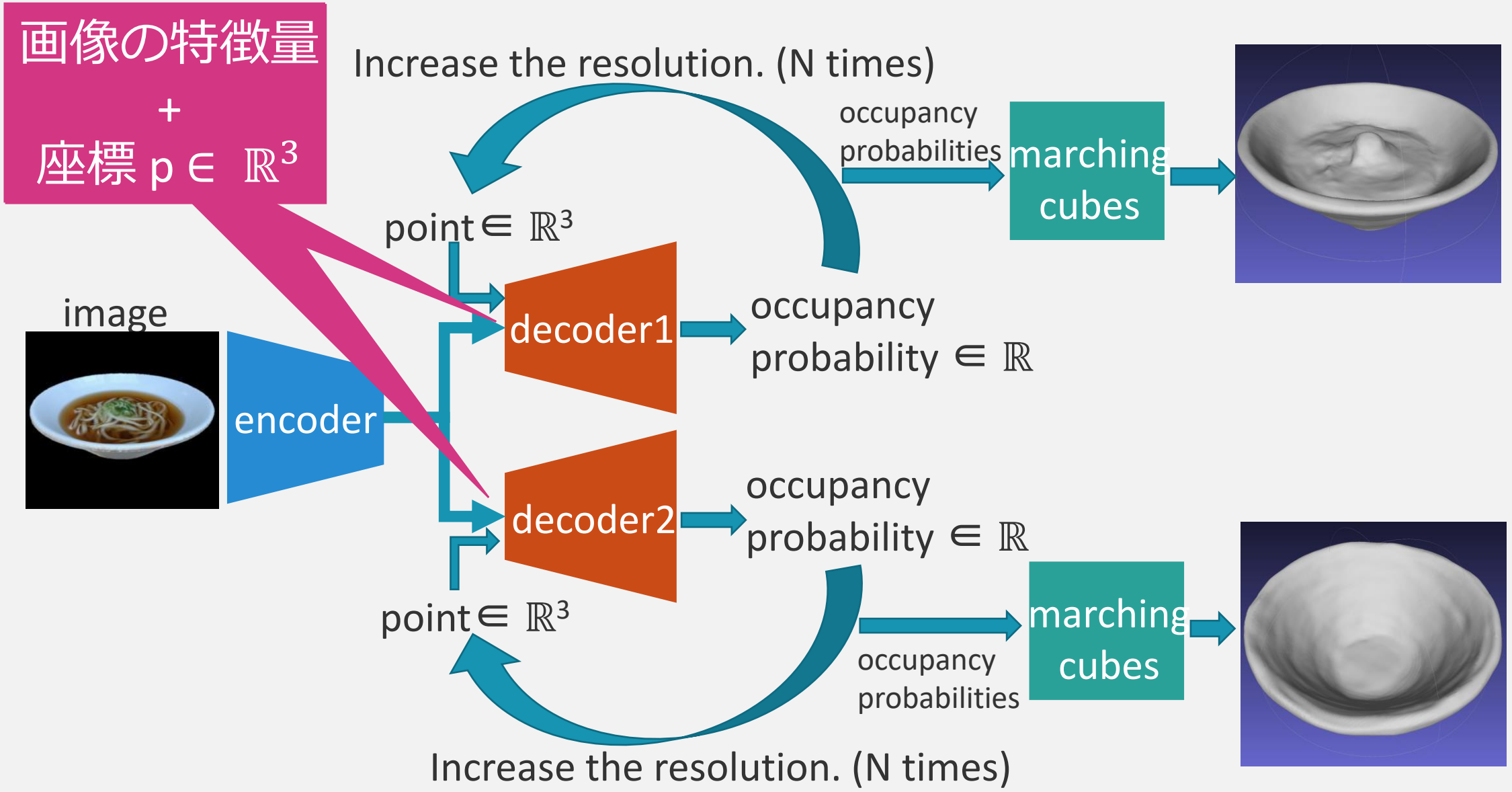
Hungry Networks



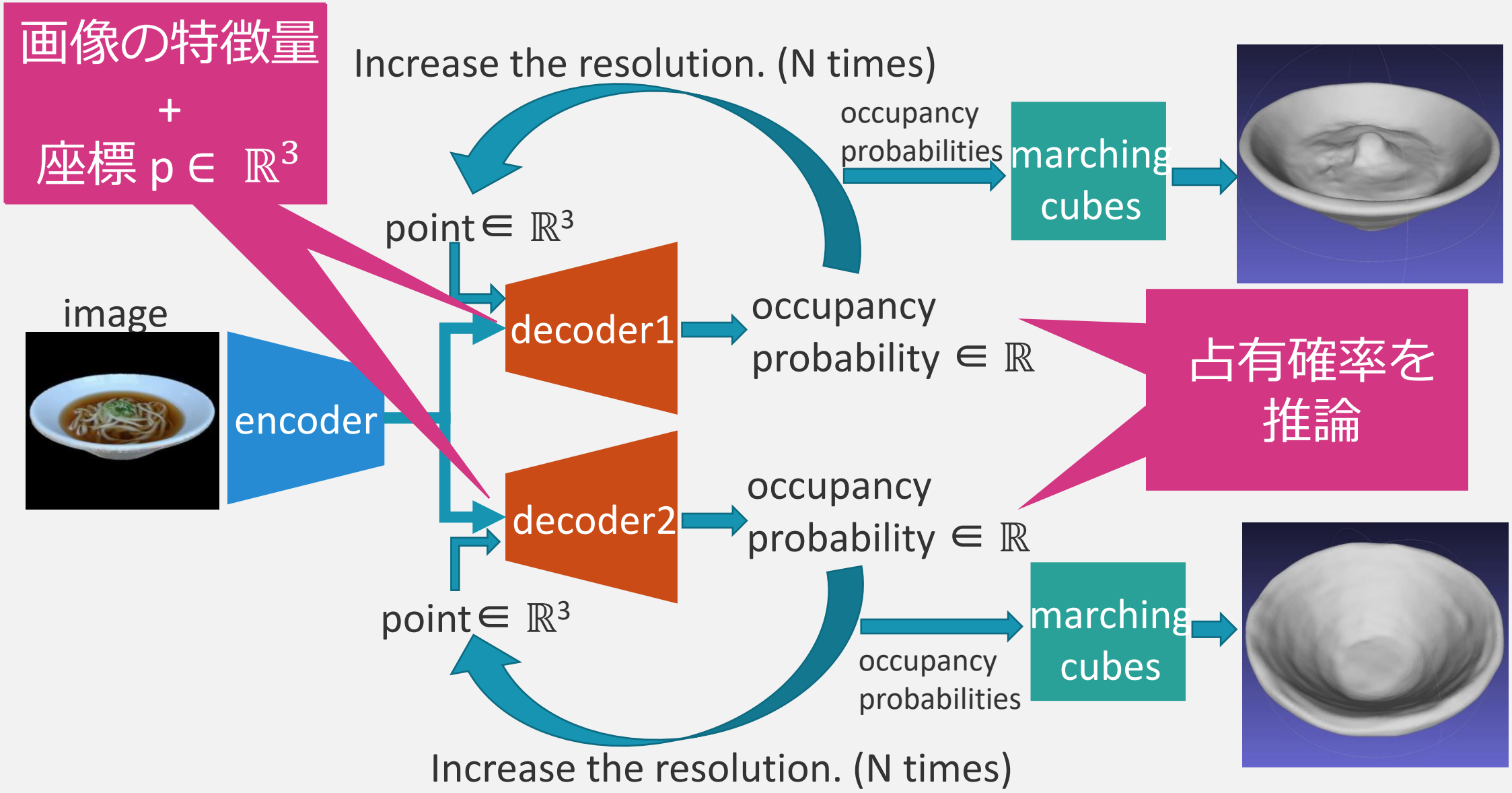
Hungry Networks



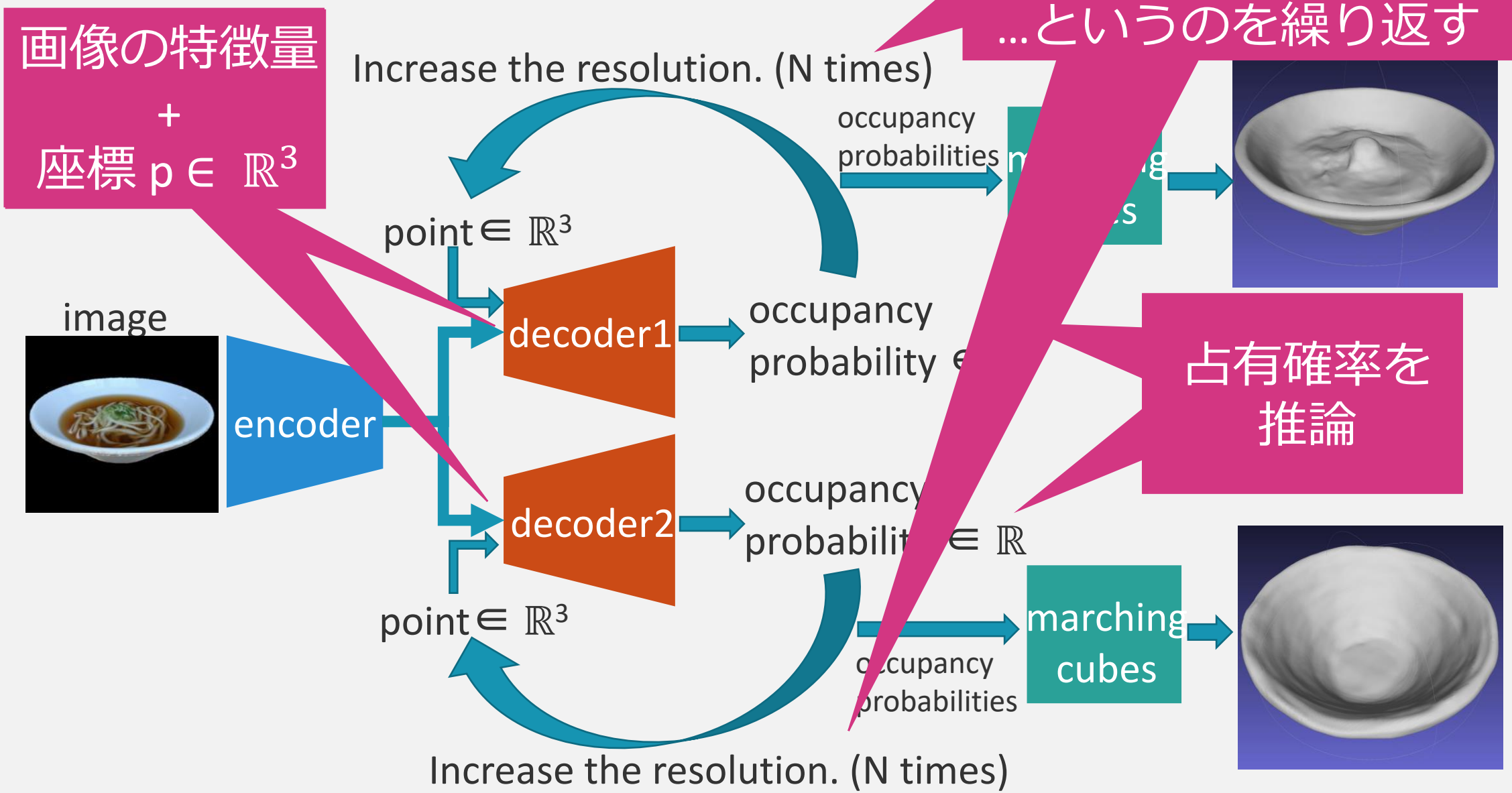
Hungry Networks



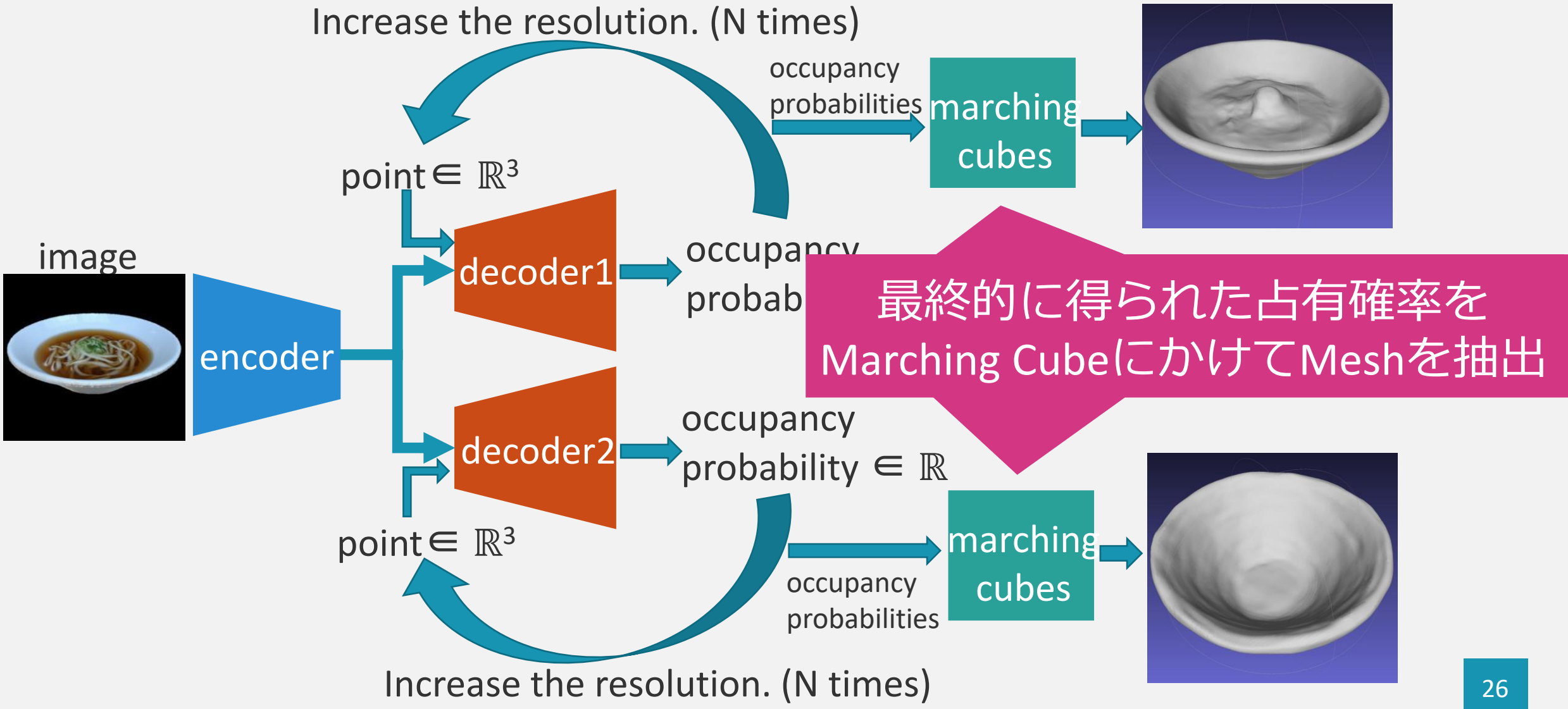
Hungry Networks



Hungry Networks



Hungry Networks



損失関数

- 学習する占有率は、実質2値分類
  - Binary cross entropy lossを利用。

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

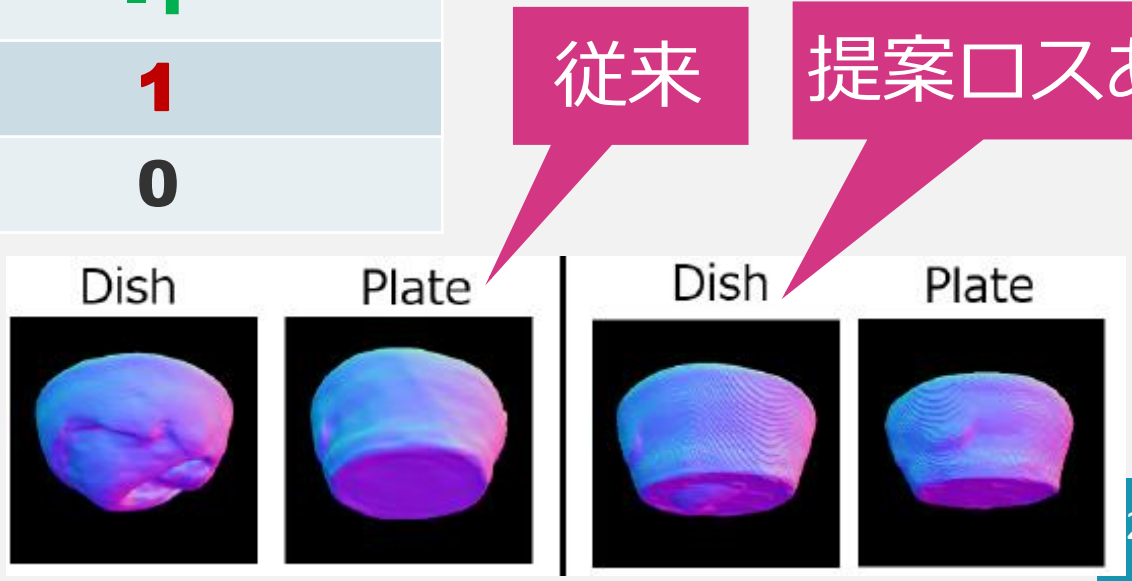
$p \in R^3$	入力のxyz座標
x	画像特徴ベクトル
$o(p) \in R$	点 p の占有率
$f_d(x,p) \in R$	占有率を出力するデコーダ

損失関数

- **3D shape consistency loss (新提案)**
  - 食器形状を一致させるためのロス

食事の占有率 $f_{d1}(x,p)$	食器の占有率 $f_{d2}(x,p)$	$f_{d2}(x,p) - f_{d1}(x,p)$
0	0	0
1	0	-1
0	1	1
1	1	0

$$\mathcal{L}_C(f_{d1}(p), f_{d2}(p)) = \max(f_{d2}(p) - f_{d1}(p)\ , 0)$$



損失関数

- **3D shape consistency loss (新提案)**
  - 食器形状を一致させるためのロス

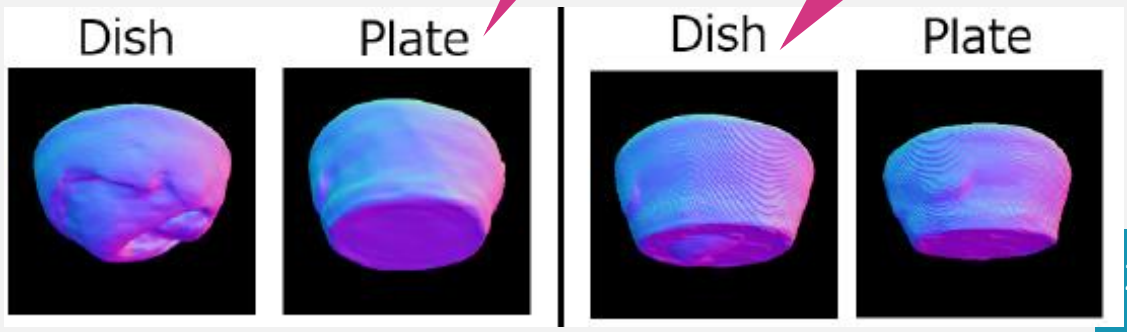
食事の占有率 $f_{d1}(x,p)$	食器の占有率 $f_{d2}(x,p)$	$f_{d2}(x,p) - f_{d1}(x,p)$
0	0	0
1	0	-1
0	1	1
1	1	0

1が問題あるパターン。

$$\mathcal{L}_C(f_{d1}(p), f_{d2}(p)) = \max(f_{d2}(p) - f_{d1}(p)\ , 0)$$

従来

提案ロスあり



損失関数

- **3D shape consistency loss (新提案)**
  - 食器形状を一致させるためのロス

食事の占有率 $f_{d1}(x,p)$	食器の占有率 $f_{d2}(x,p)$	$f_{d2}(x,p) - f_{d1}(x,p)$
0	0	0
1	0	-1
0	1	1
1	1	0

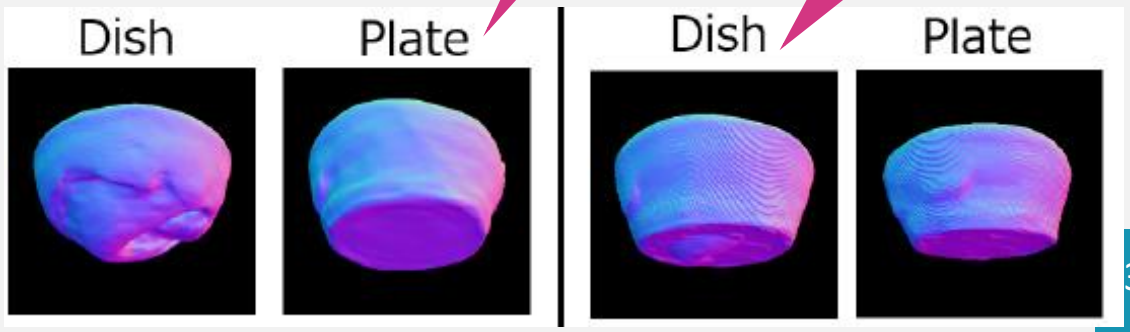
-1は問題ないパターン

1が問題あるパターン。

$$\mathcal{L}_C(f_{d1}(p), f_{d2}(p)) = \max(f_{d2}(p) - f_{d1}(p)\ , 0)$$

従来

提案ロスあり



損失関数

- バッチ毎の損失関数

$$
\begin{aligned}
x_i &= f_e(I_i) \\
y1_{i,j} &= f_{d1}(x_i, p_{i,j}) \\
y2_{i,j} &= f_{d2}(x_i, p_{i,j})
\end{aligned}
$$

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

$f_e(I_i)$ 画像を入力に特徴量を出力するエンコーダ

I_i バッチ中のi番目の画像

$\mathcal{B}$ ミニバッチ

データセット

- 食事の3D Meshを含むデータセットは存在しない。
  - 新しくデータセットを構築

- 食事の3Dモデルに240個、食器のモデルに38個用意。




スキャンデータの問題点

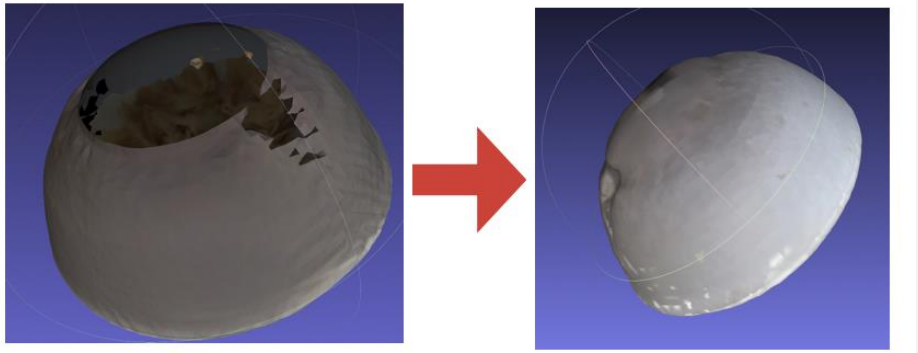
- スキャナが出力したMeshはそのまま学習できない。
- 問題点
  - (1)原点とモデル中心が一致しない
  - (2)水密ではない
  - (3)サイズが統一されてない
  - (4)ノイズを含んでいる
  - (5)食事と食器のMeshの食器部分の座標がずれてる。

スキャンデータの問題点

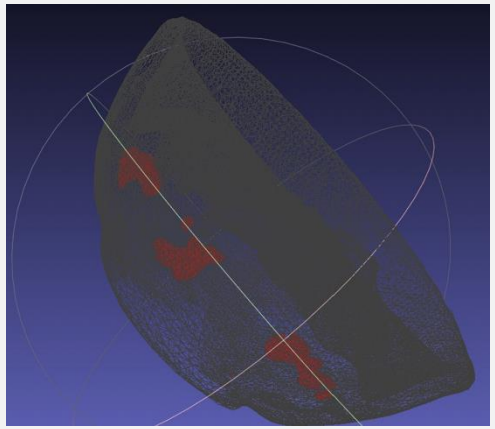
- スキャナが出力したMeshはそのまま学習できない。
- 問題点
  - (1)原点とモデル中心が一致しない
  - (2)水密ではない
  - (3)サイズが統一されてない
  - (4)ノイズを含んでいる
  - (5)食事と食器のMeshの食器部分の座標がずれてる。

スキャンデータの修正

- (2)水密ではない
  - スキャナが撮影したモデルは床に接していた面が欠落
  - Poisson Surface Reconstruction[27,28]で補完

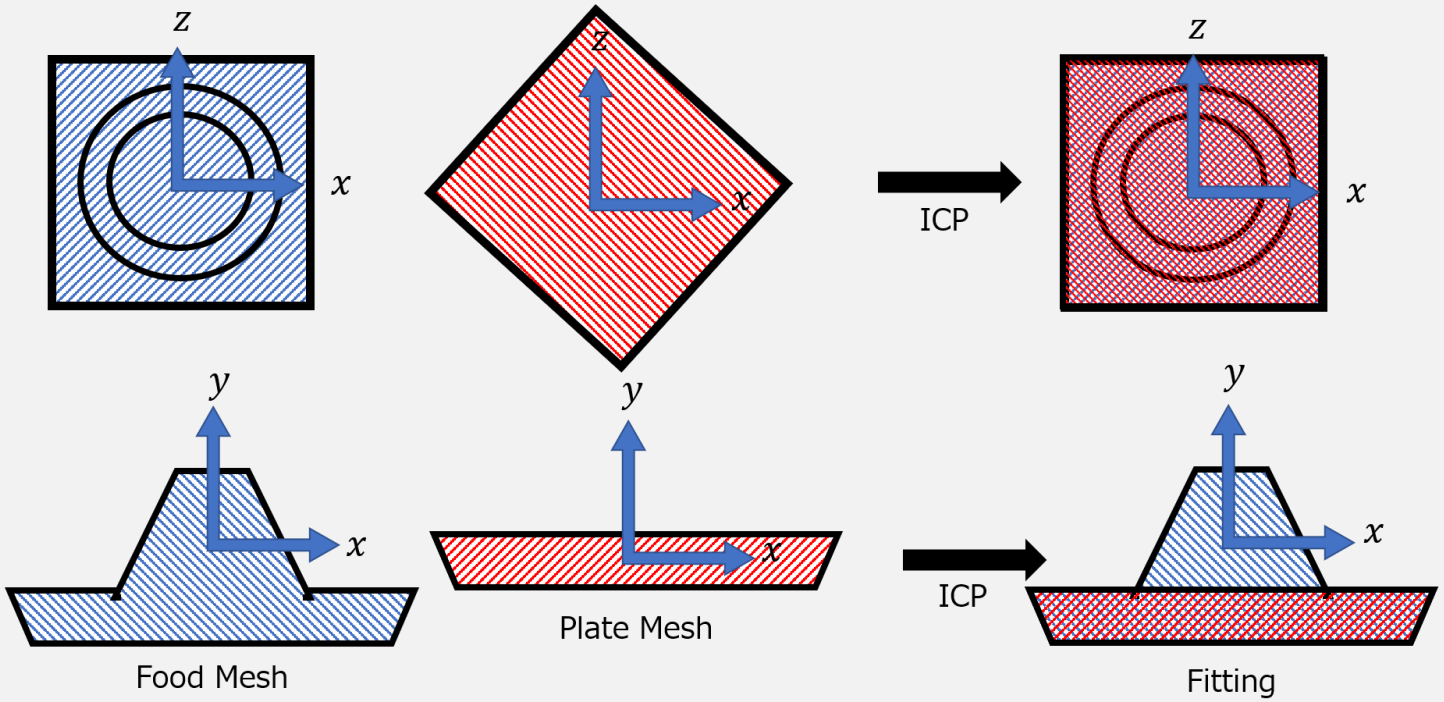


- (4)ノイズを含んでいる
  - TSDF Fusion[10]を用いて排除



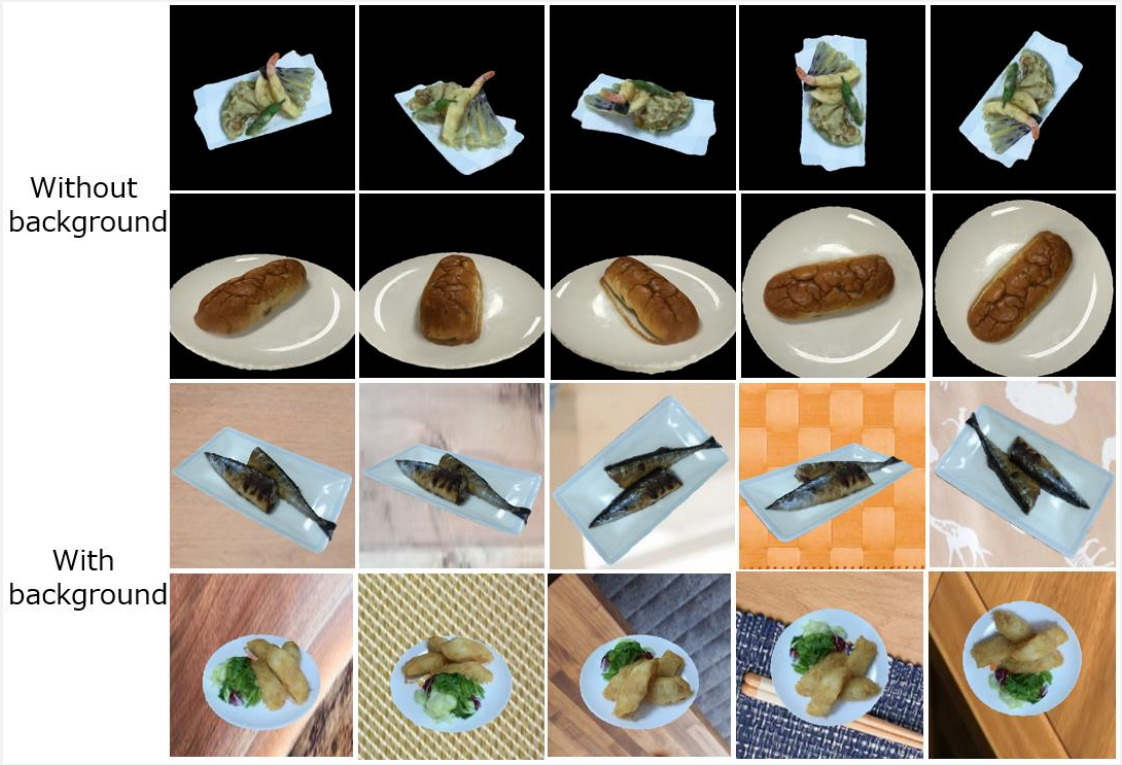
スキャンデータの修正

- (5)食事と食器のMeshの食器部分の座標がずれてる。
  - ICP(Iterative closest point)を利用して食事と食器のMeshの位置を合わせる。
  - 3D shape consistency lossが使えるようになる。



学習用画像の作成

- 3D-R2N2[13]と同様にblenderを用いてレンダリング。
- モデル単体のレンダリングと、背景を合成した2パターン用意。



実験

- 実験項目は3つ。
  - (1) 3D shape consistency loss の重みを3パターン
  - (2) バックボーンネットワークを3パターン
  - (3) 学習画像を背景なし/ありの2パターン

実験

- 評価指標
  - Volumetric IoU

    3d volumeの重なり具合
  - Chamber L1 distance

    生成されたMeshの表面か真値のMesh表面までの距離と
    その逆の平均
  - plate consistency

    食器の一貫性のスコア
  - Volume error

    食事の体積の誤差

実験 : 定量評価

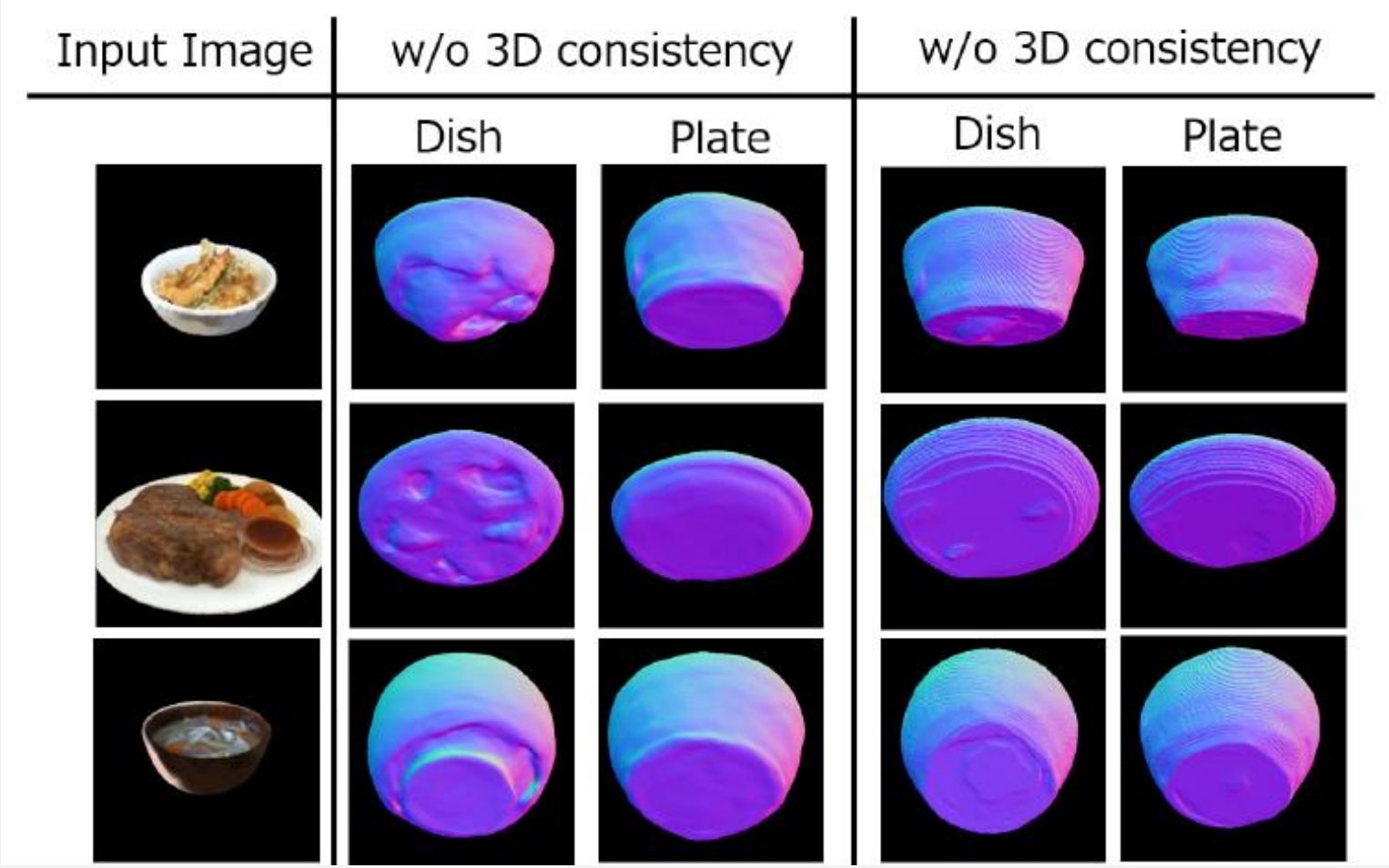
- (1) 3D shape consistency loss の重み

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

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

実験 : 定量評価

- (1) 3D shape consistency loss の重み



実験 : 定量評価

- (2) バックボーンネットワークを3パターン

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**

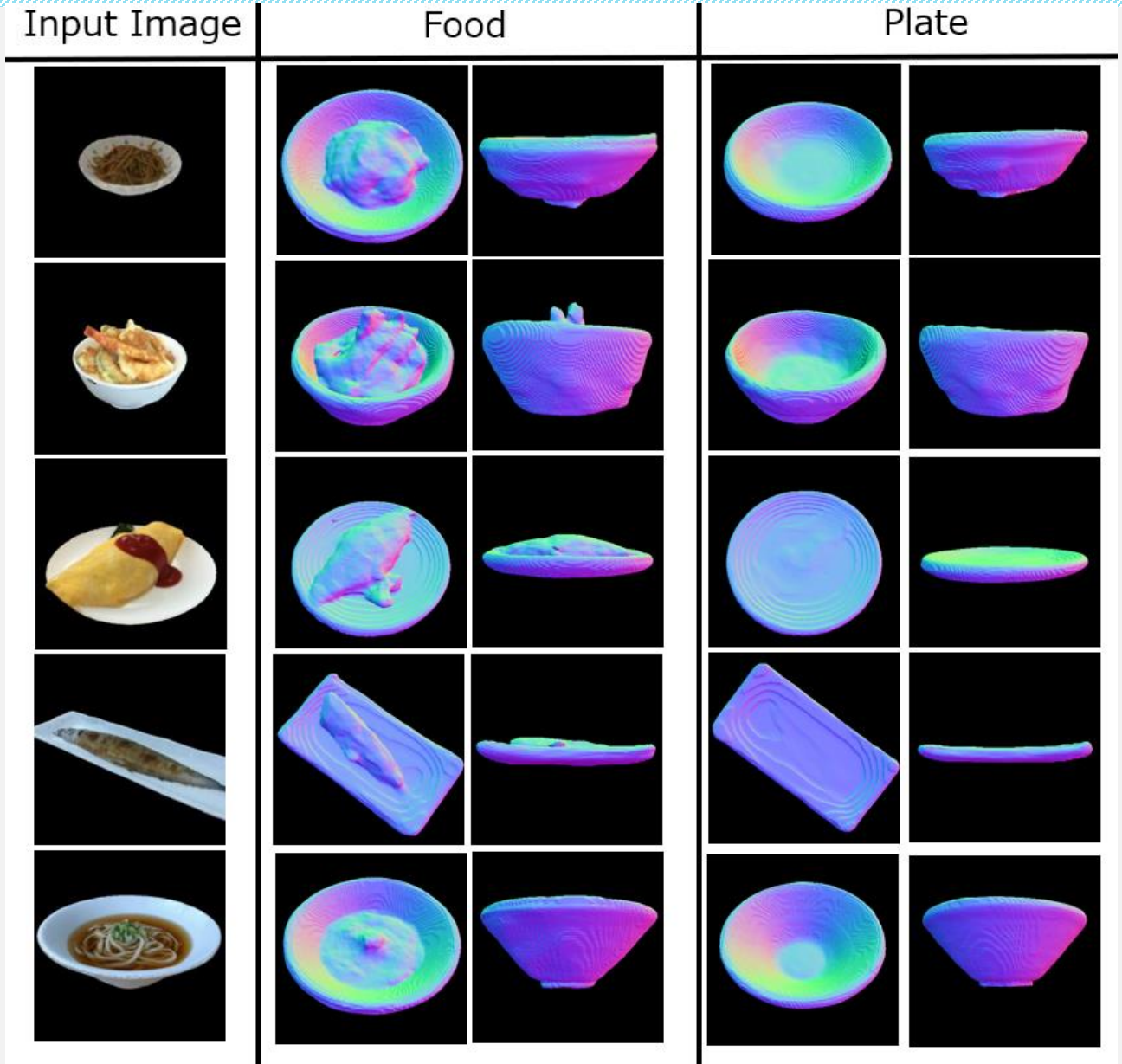
実験 : 定量評価

- (3) 学習画像を背景なし/ありの2パターン
  - 背景つき画像 + ResNet18 + λ_3=20 が最も精度がよい。

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

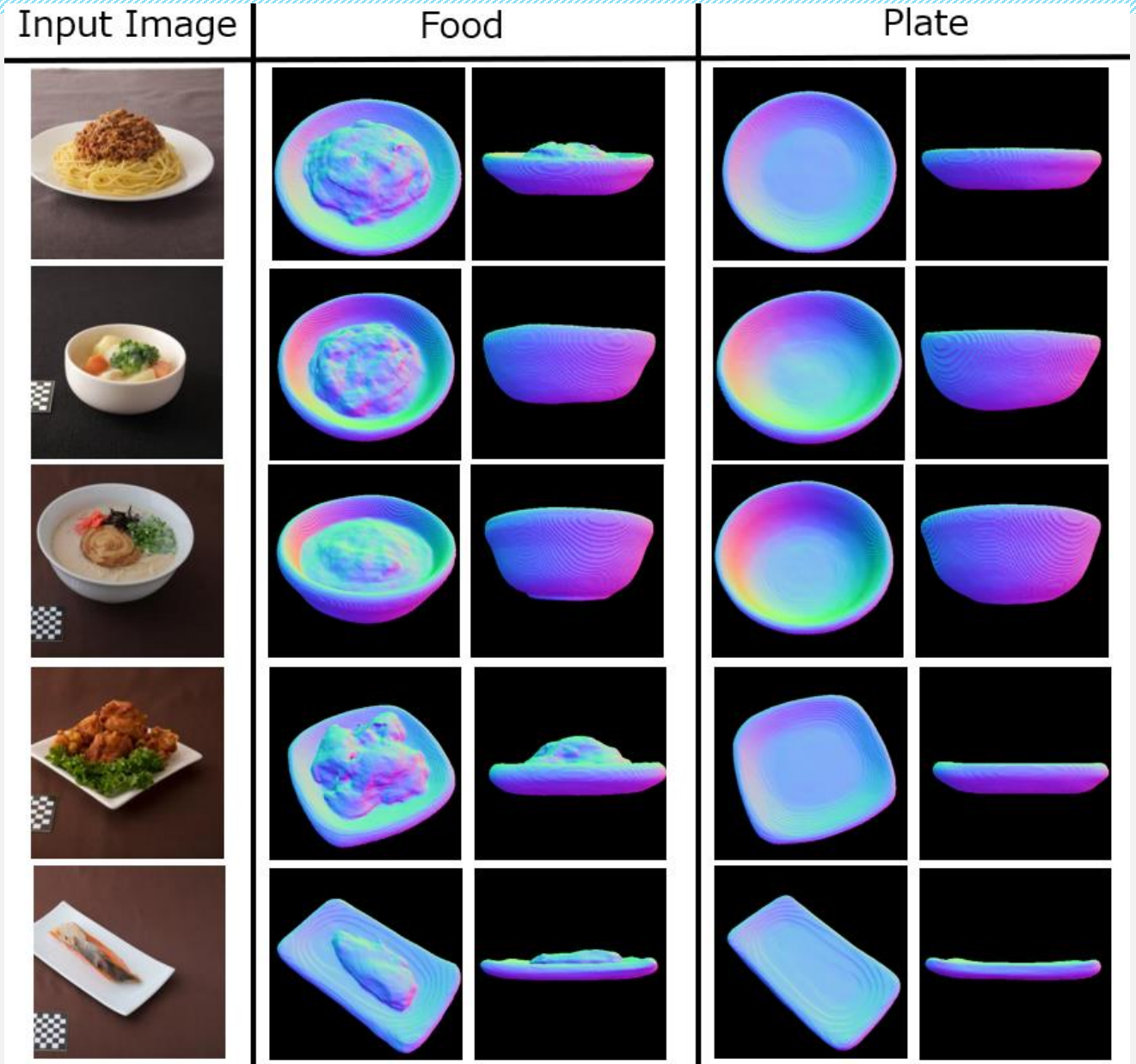
実験 : 定性評価

- ResNet18
- λ_3=20
- 背景無し



実験 : 定性評価

- ResNet18
- λ_3=20
- 背景あり



おわりに

- Hungry Networksの作成
  - 単一食事画像から食事と食器の三次元形状を復元

- 3D shape consistency lossの提案
  - 生成される2つの三次元形状の食器形状が一致。
  - 体積推定の精度に貢献

- 学習のための3D食事データセットの作成
  - 実際の食事画像に対応できる事を示した。