Hungry Networks: 3D Mesh Reconstruction of a Dish and a Plate from a Single Dish Image for Estimating Food Volume

Shu Naritomi Keiji Yanai
The University of Electro-Communications, Tokyo, Japan
naritomi-s@mm.inf.uec.ac.jp

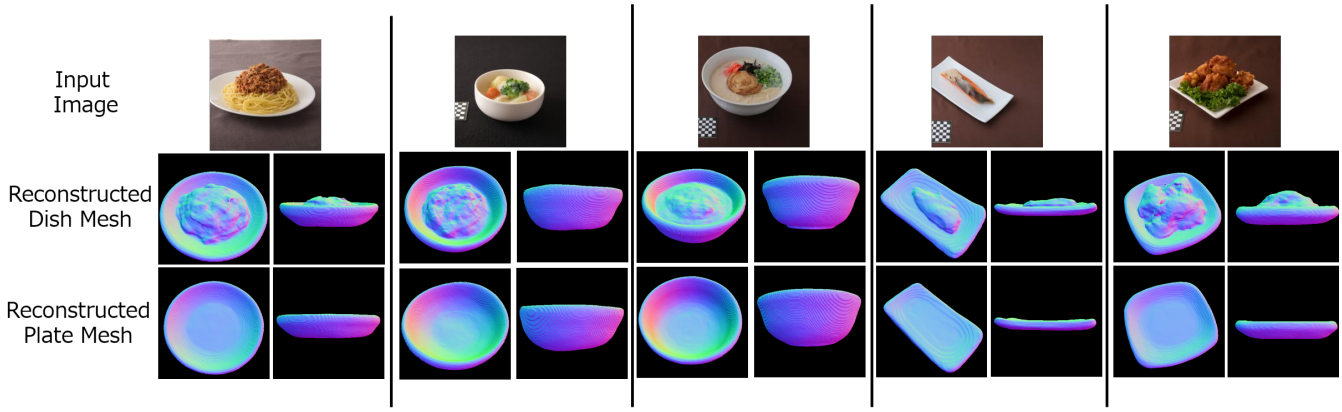

Figure 1: 3D reconstruction results from real food photos with ResNet18, $\lambda_3 = 20$ (w/ 3D consistency loss) and backgrounds.

ABSTRACT

Dietary calorie management has been an important topic in recent years, and various methods and applications on image-based food calorie estimation have been published in the multimedia community. Most of the existing methods of estimating food calorie amounts use 2D-based image recognition. On the other hand, in this paper, we would like to make inferences based on 3D volume for more accurate estimation. We performed 3D reconstruction of a dish (food and plate) and a plate (without foods), from a single image. We succeeded in restoring the 3D shape with high accuracy while maintaining the consistency between a plate part of an estimated 3D dish and an estimated 3D plate. To achieve this, the following contributions were made in this paper. (1) Proposal of "Hungry Networks," a new network that generates two kinds of 3D volumes from a single image. (2) Introduction of 3D shape consistency loss that matches the shapes of the plate parts of the two reconstructed models. (3) Creating a new dataset of 3D food models that are 3D scanned of actual foods and plates. We also conducted an experiment to infer the volume of only the food region from the difference of the two reconstructed volumes. As a result, it was shown that the introduced new loss function not only matches the 3D shape of the plate, but also contributes to obtaining the volume with higher accuracy. Although there are some existing studies that consider 3D shapes of foods, this is the first study to generate a 3D mesh volume from a single dish image.

KEYWORDS

food volume estimation, 3D reconstruction from a single image, food image recognition

ACM Reference Format:
Shu Naritomi Keiji Yanai. 2021. Hungry Networks: 3D Mesh Reconstruction of a Dish and a Plate from a Single Dish Image for Estimating Food Volume. In *ACM Multimedia Asia (MMAsia '20), March 7–9, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3444685.3446275

1 INTRODUCTION

It is necessary to consider the amount of food for accurate estimation of the amounts of food calories for dietary management. Various methods and applications on image-based food calorie estimation have been published in the multimedia community. Most of the existing methods of estimating food calorie amounts use 2D-based image recognition. Some methods infer calorie amounts directly with regression [6, 7], while the others estimate calorie amounts based on 2D area sizes using detection and segmentation methods [5, 8]. However, most of the image-based methods cannot estimate the actual size of foods. Then, size-known reference objects were commonly used for accurate food calorie estimation. Recently, some works use AR/MR devices to estimate accurate actual food size without a reference object [21, 30].

However, the accuracy of the calorie estimation by 2D-based methods is limited due to the 3D nature of real foods. 3D-based methods have been explored so far as well. Some works tried to estimate 3D volume of foods from a single image using depth estimation CNNs [18, 20] and using a depth camera mounted on a recent smartphone [1]. In these studies, the meal was assumed to

MMAsia '20, March 7–9, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8308-0/21/03…$15.00
https://doi.org/10.1145/3444685.3446275

be on a plate on a flat surface. If multiple depth images are collected from various viewpoints, more accurate shape can be obtained using a fusion algorithm [22, 23, 33]. However, this is not realistic when used in a real situation. This is because it is not easy to take multiple depth images for a single dish. Therefore, in this work, we propose "Hungry Networks," which is a network for simultaneous 3D reconstruction of both a dish and a plate from a single 2D image. By using the difference between the estimated volumes of a dish and a plate, we can obtain only the food volume, which is difficult to obtain in general. To estimate the difference of two volumes, we introduce 3D shape consistency loss, which is a new loss function for matching the plate parts of the two output models.

Note that we do not estimate 3D food-only volumes directly, since it is difficult to create a dataset containing 3D food-only volumes. Because 3D scanned volume data contains noise and defects, the shapes of plate parts between a dish volume (containing foods and a plate) and a plate volume (containing no foods) does not completely match. Therefore, we estimate volumes of a dish and a plate simultaneously instead of estimating a food volume directly from a single food image.

Although some existing dietary datasets contain depth images, none contain a complete 3D shape (mesh) of foods. Therefore, in this work, we captured dishes and plates with a 3D scanner, and created a 3D mesh food data set. The corresponding dish image was created by rendering a scanned 3D model. We also experimented with whether the model learned from the rendered image can be reconstructed from the actual dish image. The contributions in this paper are as follows:

- Proposing "Hungry Networks," a new network that generates two models from a single image.
- Introduction of 3D shape consistency loss that matches the shape of the plate part of the two reconstructed 3D models.
- Creating a new dataset of 3D models with 3D scans of real food and plate.

2 RELATED WORK

2.1 3D shape reconstruction from a single image

There are three major methods for reconstructing a 3D shape from a single image regarding 3D representation: voxel-based, point-cloud-based, and mesh-based.

The methods for estimating voxels [4, 31, 34] uses the memory of the GPU very much. Therefore, it can be reconstructed only at a low resolution. When trying to get high resolution output in a voxel representation, the implementation becomes very complex. Since the output representation of a point cloud [10] simply outputs a set of points, the connection between points must be calculated separately in order to obtain the shape of the object. Outputs in a mesh representation mainly includes a method that uses a Mesh template [13, 27, 32], a method that dynamically creates a Mesh template [12], a method that uses geometry image [24], an occupancy expression-based method [19, 28, 29], and a Signed Distance Field (SDF) based method [25]. Since the mesh representation consists of points and their connecting edges and faces, it can be memory-efficient and high-resolution compared to voxels. Since the mesh representation consists of points and their connected edges and surfaces unlike the point-cloud representation, it is more memory efficient than voxels and has higher resolution than voxels.

2.2 Food recognition considering 3D shapes

In this section, we review works on diets that consider 3D shape or volume. The ultimate goal of each study is to estimate the amounts of calories and ingredients. In Chen et al. [3], a depth sensor is used to take a depth image to estimate the amount of calories in a food. Some methods such as Puri et al. [26] and DietCam [16] obtained a 3D shape by estimating a classical camera matrix from multiple viewpoints. In recent years, CNN-based has been acitively explored. Lu et al.[18] generated a depth image using a neural network and tried to infer the amount of food from the generated depth image. Im2calories[20] is trying to estimate the calorific value by estimating the 3D shape in voxel representation from a color image.

3 METHOD

In this work, we restore 3D shape from a single food image. In general, regarding 3D representation, 3D reconstruction methods can be classified into three main types: voxels, point clouds, and meshes. In this work, we focus on mesh. This is because we are trying to reconstruct the 3D shape of dish and plate only from a single dish image, and to obtain the volume of only the food area from their difference. To achieve the same goal, voxels must be done in high resolution, which is very costly for computational resources. Moreover, in the case of a point cloud, it cannot be used to obtain the volume unless the surface shapes are connected by post-processing. Therefore, it is desirable to restore with a mesh representation that considers the connection from the beginning. Among them, the method of generating a watertight and self-intersecting mesh is suitable so that the volume can be easily considered. Moreover, in this work, the plate parts in the two generated mesh models must be consistent. In other words, the goal was to design the generated mesh so that the following conditions were met.

- The generated mesh is watertight and contains no self-intersection.
- Consistency exists in the plate parts of the 3D mesh models of a dish and a plate.

3.1 Representation that meet the requirements

The first condition, the constraint that the output mesh is watertight and contains no self-intersection, is very important. Because when the mesh is watertight and without self-intersection, each face $f \in Faces$ is composed of $(v1, v2, v3)$ counter-clockwise when viewed from the surface of the face. This is because the volume of a given model, V, can be calculated relatively easily by the following equation:

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

To fulfill the first condition, it is difficult to use the method using the template mesh, because it easily causes self-intersection. On the other hand, the method of extracting meshes using marching cubes [19, 25, 28, 29] can generate meshes that are watertight and have no self-intersection. Therefore, it is desirable to use the occupancy or Signed Distance Field (SDF) as the output representation of the network.

Next, we consider the design for the second condition, the consistent shape of the plate of the two generated meshes. The reason why this consistency must be taken into consideration is that the 3D data of the actual dish used for training naturally contains noise and defects, so that the shapes of the dishes often do not match

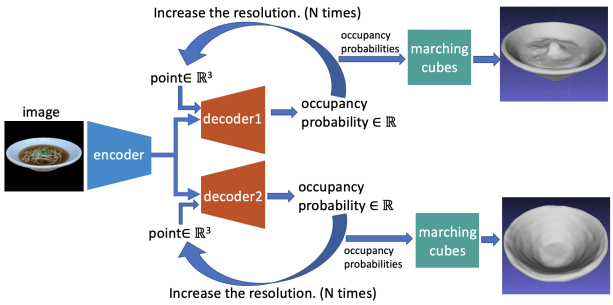

Figure 2: The overview of "Hungry Networks."

perfect. Therefore, in dealing with this problem, we consider which is better, the occupancy or SDF. Occupancy is the representation, $o \in \mathbb{R}$, whether it is inside or outside the mesh. SDF is the representation, $s \in \mathbb{R}$, how far away from the surface of the mesh. The problem with this plate consistency is that the point, $p \in \mathbb{R}^3$, that was contained inside the plate mesh is not contained inside the food mesh. In other words, in order to deal with this problem naturally, it is better to use occupancy. Therefore, in this work, we propose "Hungry Networks," which is a network that restores mesh representation of both a dish and a plate from a single image using an occupancy-based method based on Occupancy Networks [19]. In addition, we propose a new loss function, 3D shape consistency loss, so that the plate parts of two estimated mesh-based models become closer. Note that "a dish" in this paper means a combination of a plate and foods on the plate as shown in Figure 1.

3.2 Hungry Networks

A schematic diagram of the proposed network, Hungry Networks, is shown in Figure 2. The network has one encoder and two decoders.

The encoder extracts features of a dish image, which consists of a pre-trained backbone network such as ResNet. The final output layer of the encoder is a global average pooling layer to make the output of the encoder a vector which represents an image feature. Image features and 3D points, $x \in \mathbb{R}^3$, are used as decoder inputs. The decoders output the occupancy for a dish (containing a food part and a plate part) and a plate, respectively. The occupancy represents if each of 3D point is inside the mesh or outside the mesh with 1/0 binary values.

In Figure 2, Decoder-1 learns occupancy for generating a 3D mesh model of a dish, and Decoder-2 learns occupancy for generating a 3D mesh model of a plate. The generation algorithm is based on Occupancy Networks [19]. First, we infer the occupancy at the initial resolution of 32×32×32. Next, the occupancy is inferred again by increasing the resolution of only the boundary portion of the object to be generated, and the occupancy of only the boundary portion of the object is obtained again at higher resolution. In each iteration, we increase the resolution, divide the grid into eight parts, and increase the resolution as $32\times32\times32 \Rightarrow 64\times64\times64 \Rightarrow 128\times128\times128$. Unlike voxel representation, high resolution does not require all points, and only the boundary portion of the object is gradually increased in resolution. So memory efficiency is very good. By applying the marching cubes algorithm [17] to the occupancy field obtained in high resolution, the iso-surface is extracted as a mesh. Since this algorithm can always generate a 3D mesh model that is watertight and has no self-intersection, the first requirement of generated mesh is achieved.

Table 1: occupancy table

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

3.3 Training

We explain how to train the network. $p \in \mathbb{R}^3$ is the input point, x is the feature vector of the input image, and the decoder network for the dish and the plate are represented as $f_{d1}(x, p)$ and $f_{d2}(x, p)$, respectively. In addition, the occupancy of training data is represented by $o(p) \in \mathbb{R}$ corresponding to the point p. Training of occupancy is equivalent to the binary classification problem of whether the point is inside or outside the mesh surface. Then, the loss function for learning the occupancy is represented in Eq.2. Binary cross entropy loss is used for the loss function because it results in binary classification.

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

Next, we introduce a 3D shape consistency loss to match the plate parts of both the output mesh models to each other. First, the possible patterns of the combination of occupancy of the corresponding points on two mesh models are shown in Table 1. When the occupancy of both models at the corresponding point is the same, it is in the desirable condition. In addition, the condition where the occupancy of the dish is 1 and the occupancy of the plate is 0 is no problem, since such a point corresponds to a part of the food part of the dish model. On the other hand, the condition where the occupancy of the dish is 0 and the occupancy of the plate is 1 is problematic, since this means that inconsistency happens between the plate model and the dish model, which should be resolved. Penalties were applied during training only if the dish occupancy is 0 and the plate occupancy is 1, which corresponds to the condition where $f_{d2}(p) - f_{d1}(p)$ equals 1 as shown in Table 1. So, $\max(f_{d2}(p) - f_{d1}(p), 0)$ is used as a loss function to be minimize. We will call this "3D shape consistency loss" (hereinafter, this is called "3D consistency loss").

$$\mathcal{L}_C(f_{d1}(p), f_{d2}(p)) = \max(f_{d2}(p) - f_{d1}(p), 0) \tag{3}$$

The above two formulas (Eq.2, Eq.3) are put together to determine the loss $\mathcal{L}_{\mathcal{B}}$ for each mini-batch of the entire learning. Here, $\mathcal{B}$ is the sampled mini-batch, I_i is the i-th image of the batch, and K points in total from the i-th batch are sampled, and $p_{i,j}$ represents the sampled j-th point of the i-th image. It is assumed that f_e is the encoder that output image features, and f_{d1} and f_{d2} are decoder outputs that output food and plate occupancy rates, respectively.

$$x_i = f_e(I_i) \tag{4}$$
$$y1_{i,j} = f_{d1}(x_i, p_{i,j}) \tag{5}$$
$$y2_{i,j} = f_{d2}(x_i, p_{i,j}) \tag{6}$$

$$\mathcal{L}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\sum_{j=1}^{K}\Big(\lambda_1\mathcal{L}_O(y1_{i,j}, o1_i(p_{i,j})) + \lambda_2\mathcal{L}_O(y2_{i,j}, o2_i(p_{i,j})) + \lambda_3\mathcal{L}_C(y1_{i,j}, y2_{i,j})\Big) \tag{7}$$

4 DATASET CONSTRUCTION

Some existing dietary datasets contain color and depth images [11]. However, no dietary dataset contains 3D Mesh models of foods. Therefore, for this work, we had to create a new 3D dietary dataset. We created the dataset consisting of 240 3D models of foods and 38 models of plates. To create the models, we used a commercially available 3D sensor called "Structure Sensor" and a dedicated 3D scanning application. Since the same plate was used for different dishes, the number of plate models is smaller than that of dish models.

4.1 Steps to make scanned data learnable

Many 3D model datasets, which are often used for training neural networks in recent years, are basically composed of data created by humans using 3D modeling software. However, the dataset created for learning in this work was constructed by scanning a real object with a commercially available 3D scanner. Since the mesh output by the scanner contains noise and defects, it cannot be used as it is for training of neural networks. Therefore, some pre-processings are needed to perform to make the scanned models ready for training. Additionally, in this work, unless the two models of food and plate are fitted, the 3D consistency loss, which is premised on comparing the occupancy rates at the same coordinates, cannot be used. There are five problems with the 3D models created by scanning.

(1) The center of the model does not coincide with the origin.
(2) Not watertight.
(3) The size is not unified.
(4) Containing noise.
(5) The coordinates of the plate parts of a dish mesh and a corresponding plate mesh do not match to each other.

4.1.1 Aligning 3D models to the origin. Scanned models were located anywhere in the 3D space, and the locations are not unified. Then, first, we move to all the 3D models so that the center of the model is aligned to the origin of the 3D space.

4.1.2 Complementing mesh defects. Since the network used in this work is designed to infer the occupancy, all the 3D model should be watertight. However, the scanned models sometime have holes as shown in the left column of Figure 3. We have to complementing holes to make the models watertight.

There exist several algorithms to fill the perforated model [2, 14, 15]. In this work, we use Poisson Surface Reconstruction [14, 15]. However, this algorithm cannot be applied to the model as it is. Figure 3 shows the results when the Poisson Surface Reconstruction is applied to the scanned 3D model as it is. If the defects present in the model are small to some extent, the surface is complemented relatively neatly on Figure 3. However, as shown at the bottom of Figure 3, there were many models whose surfaces were not complemented as expected. The main expected reason why Poisson Surface Reconstruction does not create the surface is that the mesh deficiency is too large. This is because the scanned 3D model cannot scan the surface in contact with the floor. Therefore, before applying Poisson Surface Reconstruction, we created and applied an simple algorithm to fill the holes in the ground plane to some extent to fill the holes. After filling in the defects on the surface of the model, the size of the model was normalized to -0.5 to 0.5 and the size was unified.

4.1.3 Removing noise. Next, we address the issue of noise remaining in the model. Such noise was dealt with by reconfiguring the mesh again using TSDF Fusion. TSDF Fusion refers to a part of the method proposed by Kinect Fusion [23]. Using this method, the noise inside the model was completely removed.

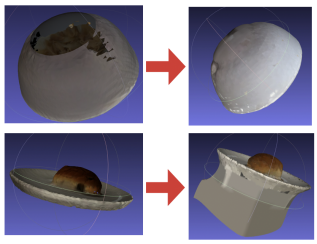

Figure 3: Poisson surface as a result of reconstruction. Although small holes in the ground plane are complemented well, large holes are not complemented correctly.

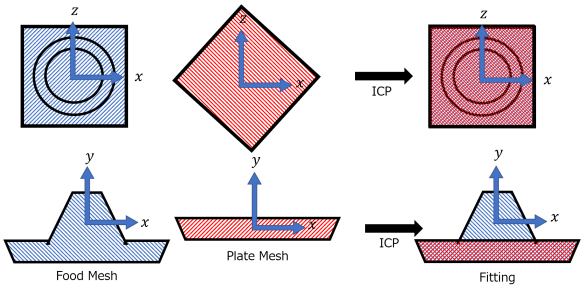

Figure 4: Use ICP (Iterative Closest Point) to match the coordinates of the plate part of the dish and the plate mesh.

4.1.4 Fitting. Finally, we align the plate parts of the two mesh models, a dish mesh model (containing both food and plate parts) and a plate mesh model (containing only a plate part). In the 3D consistency loss, we assumed that the plate parts of the corresponding dish and plate models are aligned. Therefore, we use ICP (Iterative Closest Point) to fit the dish and plate as shown in Figure 4.

4.2 Generating input images by rendering

In this work, we rendered images for learning using software called blender, similar to 3DR2N2 [4]. 25 images were rendered for each model, taken from various angles. The images created by 3DR2N2 only shows only the models, and they contain no background. However, food photos taken in actual situations always contain backgrounds. In this work, we collected textures of various types of tables and tablecloths from the Web as the background of the rendered dish images, and created composite images. Figure 5 shows the image created by rendering. The top two lines are just rendered, and the bottom two lines are a composite of the background.

5 EXPERIMENTS

We made experiments with the proposed model, "Hungry Networks", on the following conditions: (1) we set three values as the 3D consistency loss weight ($\lambda 3$ in Eq.7), (2) we use three different backbone networks, and (3) we train the model with rendered dish images with/without backgrounds. To train the proposed network, we used 216 models for training and 24 models for evaluation among 240 models in the constructed dataset. The hyperparameters, $\lambda 1$, $\lambda 2$, were fixed at 1, and only $\lambda 3$ was changed in the experiments. We used Adam as an optimizer.

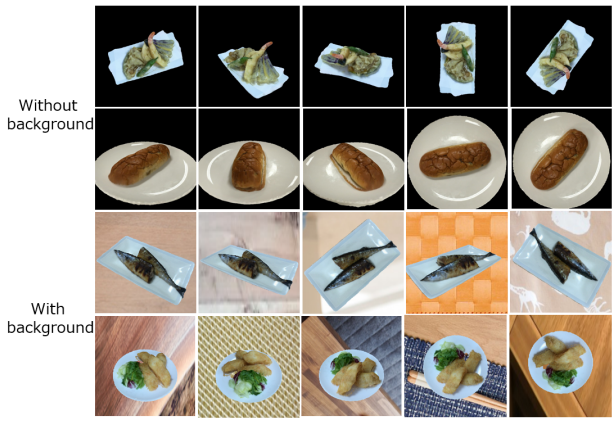

Figure 5: Images rendered for training without/with backgrounds.

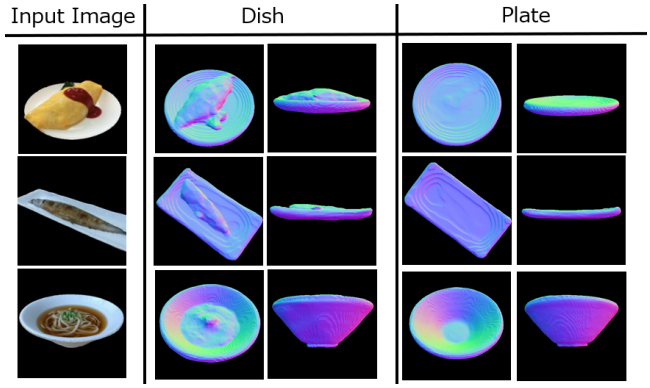

Figure 6: The estimated volumes of both dishes and plates with the model trained from non-background images with a ResNet18-based encoder and $\lambda 3 = 20$.

5.1 Metrics

For quantitative evaluation, we use Volumetric IoU, Chamber L1 distance, plate consistency, and volume error. Volume error is the most important in this work, since it directly connected to estimation of food calorie amounts.

Volume IoU is defined as the quotient of the union volume between the estimated mesh and the ground-truth mesh and the volume of their intersection. It is calculated by randomly sampling 100,000 points from the inside of the bounding box of the mesh and inferring whether the points are inside or outside.

Chamfer L1 distance is calculated from the average of the two indicators. One is the mean distance from points on the generated mesh to the nearest neighbor points on the ground-truth mesh. The other is opposite. 100,000 points were sampled from the surface of each mesh, and the nearest neighbor points were searched using KD-Tree as the previous works [9, 19].

The plate consistency is the mean distance from points on the generated plate mesh to the nearest neighbor points on the generated food mesh. This value indicates how different the plate part of the dish volume is from the plate volume.

The food volume is obtained by subtracting the plate volume from the dish volume. To evaluate estimation of the food volume, we calculate ground-truth food volumes by removing the plate parts from the corresponding dish mesh manually. Since it is very time-consuming, we created ground-truth 3D food mesh models on only 24 evaluation models. Volume error is the mean distance from the inferred volume of the food region to the ground-truth food volume.

On IoU, the higher value is better, while on the other metrics, the Chamfer L1 distance, plate consistency and the volume error, the lower values are better.

5.2 Quantitative evaluation

First, we investigated the effect of $\lambda 3$ on evaluation. The encoder was based on ResNet34, and training images without backgrounds were used for training and evaluation. We made experiments with 0, 20 and 50 for $\lambda 3$. Note that 0 means we did not used the 3D consistency loss. The results are shown in Table 2. As a result, it indicates that the volume error is greatly reduced when 3D shape consistency loss is used. On the other hand, the 3D meshes of dishes and plates were estimated the most accurately without the 3D consistency loss. However, as shown in Figure 7, in case of no 3D consistency loss invisible parts of the dish volume and the plate volume were differently reconstructed. Since both the dish decoder and the plate decoder were optimized independently using only independent occupancy loss functions to each other, individual evaluation tends to become better and integrated evaluation such as volume error tends to become worse.

Table 3 shows the results using different backbone networks, ResNet18, ResNet34, and ResNet50 using non-background images and $\lambda 3 = 20$, which achieved the most accurate estimation regarding food volume error in the previous experiments. As a result, ResNet50 was the most accurate in food volume error, although the difference between ResNet18 and ResNet50 was very small.

In the next experiments, we evaluated how much accuracy was affected by backgrounds of training images. We used $\lambda 3 = 20$ with ResNet18 and ResNet50 as backbones. With backgrounds in the training images, we achieved the best results regarding the volume error and the plate consistency.

5.3 Qualitative evaluation

Figure 6 shows the estimated 3D meshes of both dishes and plates with $\lambda 3 = 20$, ResNet18 and training images without background. The 3D meshes of both the dishes and the plates were correctly estimated for the corresponding images. Although the input images contained foods, the plate decoder, which was trained with only 3D plate models, estimated only the plate parts. In addition, we can see that most of the plate parts of the dish meshes were identical to the plate meshes.

Figure 7 shows comparative results between the results with the 3D consistency loss and the ones without the 3D consistency loss. In case of no plate consistency loss, the reconstructed 3D shapes of dishes and plates are different from each other, especially on the invisible parts such as the bottom parts of the plates from the input images. This mainly comes from the nature of the training 3D data. The scanned 3D mesh data is prone to noise and defects in the parts in contact with the grounds. Therefore, when viewed from the bottom, the results of generating two volumes may differ significantly. On the other hands, when the 3D consistency loss was used, it can be seen that the plate parts were consistent in both volumes.

Figure 1 shows the results with real food photos as input images in the same training condition as the previous experiments.

Table 2: The evaluation results with three kinds of $\lambda 3$ using ResNet34 and non-background images.

$\lambda 3$	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

Table 3: The evaluation results with three kinds of backbones, ResNet18, ResNet34, and ResNet50 with $\lambda 3 = 20$ and non-background images.

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**

Table 4: The evaluation results with training images with/without backgrounds with $\lambda 3 = 20$ and ResNet18/50 backbones.

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

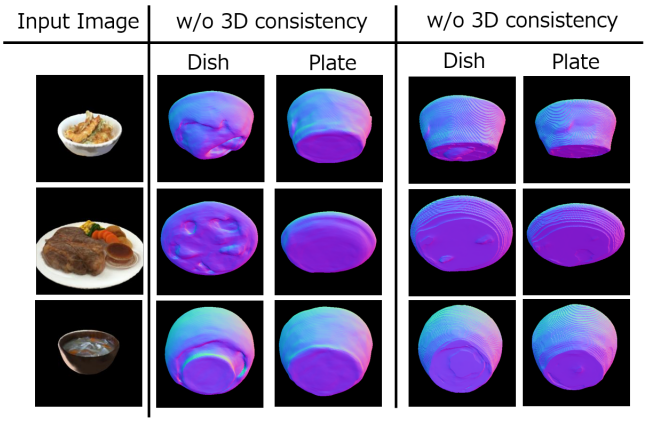

Figure 7: Comparative results with/without the 3D consistency loss. Note that training condition is the same as Fig.6.

Although real images of actual foods were not used for network training, the trained model was able to reconstruct 3D volumes of the dishes and the plates. Various kinds of the plates such as square flat plates, rectangular plates, round flat plates, and bowls were successfully reconstructed, although the shape and the height of each of the plates were different greatly.

5.4 Discussion on 3D shape consistency loss

We found that introducing 3D shape consistency loss brought both advantages and disadvantages. The disadvantage is that evaluation using a general evaluation metrics on each of the dish and the plate such as Chamfer distance will be worse. This is because 3D shape consistency loss is not a loss to become closer to ground-truth. The advantage is that it can absorb the noise of the mesh generated by the 3D scanner. Since the dataset was created using an inexpensive 3D scanner, it inevitably contains noise such as defects and swelling. Data sets such as ShapeNet are models carefully created by humans using 3D modeling software and were very well-organized. However, this is not the case when creating 3D real object datasets using a 3D scanner. Therefore, the dishes and the corresponding plates, which should contain the same plate, contain slightly different shapes due to the accuracy of the scanner. Even if the volume size of the plate model is subtracted from the volume size of the dish model in order to obtain the volume size of the food portion, it will not become idential to the volume size of the actual foods due to the noise of the plate portion. Therefore, by introducing 3D shape consistency loss and matching the plate shapes, the accuracy of calculating of the 3D volume size has improved.

6 CONCLUSIONS

In this work, we proposed "Hungry Networks" that enabled 3D shape reconstruction of dishes and plates from a single food image. For training, we introduced a new loss, 3D shape consistency loss, in order to maintain the consistency between the plate part of the dish and the plate. In addition, for experiments, we created a dataset consisting of 3D mesh models of dishes. By the experiments, it was shown that 3D shapes could be reconstructed with high accuracy by using rendered images of dishes and composite rendered images of backgrounds for training. In addition, by introducing 3D shape consistency loss, we succeeded in maintaining and restoring the consistency of the plate parts of the two meshes, which contributed to the estimation of the volume of the dietary area. It was shown that the network learned from the dish images obtained by synthesizing the background image can be correctly reconstructed even if the real dish image is input as well.

As a future task, the current 3D shape restoration is performed in a normalized space, and the actual size cannot be taken into consideration. In order to estimate the amount of calories, it is necessary to be able to consider the actual size. Therefore, we would like to use the environment recognition function of the AR device, RGB-D depth images, reference objects and so on to perform 3D shape restoration considering the actual size, which will lead to accurate estimation of the amounts of food calorie intake.

REFERENCES

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

[2] F. Calakli and G. Taubin. 2011. SSD: Smooth signed distance surface reconstruction. In *Computer Graphics Forum*, Vol. 30. 1993–2002.

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

[4] C. B. Choy, Danfei. Xu, J. Gwak, K. Chen, and S. Savarese. 2016. 3D-R2N2: A unified approach for single and multi-view 3d object reconstruction. In *Proc. of European Conference on Computer Vision*. 628–644.

[5] T. Ege and K. Yanai. 2017. Estimating Food Calories for Multiple-dish Food Photos. In *Proc. of Asian Conference on Pattern Recognition*.

[6] T. Ege and K. Yanai. 2017. Imag-Based Food Calorie Estimation Using Knowledge on Food Categories, Ingredients and Cooking Directions. In *Proc. of ACM Multimedia Thematic Workshop*.

[7] T. Ege and K Yanai. 2018. Image-Based Food Calorie Estimation Using Recipe Information. *IEICE Transactions on Information and Systems* E101-D, 5 (2018), 1333–1341.

[8] T. Ege and K Yanai. 2018. Multi-task Learning of Dish Detection and Calorie Estimation. In *Proc. of IJCAI and ECAI Workshop on Multimedia Assisted Dietary Management*.

[9] Haoqiang Fan, Hao Su, and Leonidas Guibas. 2017. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In *Proc. of IEEE Computer Vision and Pattern Recognition*.

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

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

[12] G. Gkioxari, J. Malik, and J. Johnson. 2019. Mesh R-CNN. In *Proc. of IEEE International Conference on Computer Vision*. 9785–9795.

[13] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik. 2018. Learning category-specific mesh reconstruction from image collections. In *Proc. of European Conference on Computer Vision*. 371–386.

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

[15] M. Kazhdan and H. Hoppe. 2013. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)* 32, 3 (2013), 1–13.

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

[17] W. E. Lorensen and H. E. Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *ACM siggraph computer graphics* 21, 4 (1987), 163–169.

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

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

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

[21] S. Naritomi and K. Yanai. 2020. CalorieCaptorGlass: Food Calorie Estimation Based on Actual Size using HoloLens and Deep Learning. In *Proc. of IEEE Conference on Virtual Reality and 3D User Interfaces*.

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

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

[24] Albert P., Jordi S., Gary P. T. C., Alberto S., and Francesc M. 2019. 3DPeople: Modeling the Geometry of Dressed Humans. In *Proc. of IEEE International Conference on Computer Vision*.

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

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

[27] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black. 2018. Generating 3D faces using convolutional mesh autoencoders. In *Proc. of European Conference on Computer Vision*. 704–720.

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

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

[30] R. Tanno, T. Ege, and K. Yanai. 2018. AR DeepCalorieCam V2: food calorie estimation with CNN and AR-based actual size estimation. In *Proc. of the 24th ACM Symposium on Virtual Reality Software and Technology*. 1–2.

[31] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik. 2017. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proc. of IEEE Computer Vision and Pattern Recognition*. 2626–2634.

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

[33] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger. 2016. ElasticFusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research* 35, 14 (2016), 1697–1716.

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