Supplementary Material for VolETA++: A Fully Automatic Depth-Free Radiance-Field Framework for Food 3D Reconstruction and Volume Estimation

Ahmad AlMughrabi
Universitat de Barcelona, Spain
ahmad.almughrabi@ub.edu

Ricardo Marques*
Grup de Tecnologies Interactives (GTI),
Universitat Pompeu Fabra (UPF), Spain
ricardo.marques@upf.edu

Petia Radeva*
Universitat de Barcelona, Spain
Institut de Neurociències, Barcelona
petia.ivanova@ub.edu

1. Broader Impact and Relevance

While our work centers on food volume estimation, the concept of establishing a scaling factor to metrically scale 3D meshes is broadly relevant beyond this specific application. Our proposed method is applicable to several domains, including medical imaging (e.g., scaling volumetric CT/MRI reconstructions for accurate diagnostics [7]), augmented and virtual reality (e.g., ensuring correct proportions of virtual objects within real-world settings [2]), manufacturing (e.g., rescaling 3D-scanned items for industrial quality control [9]), and cultural heritage preservation (e.g., accurately recreating and digitizing artifacts [8]).

By enhancing the accuracy of 3D reconstructions and providing a practical metric calibration strategy, our research addresses a core challenge in volumetric estimation and metric-aware neural rendering. This makes it relevant to the broader research community engaged in neural rendering, 3D scene understanding, and real-world mesh reconstruction applications.

2. Keyframe Selection

To enhance the input quality for NeRF and reduce reconstruction artifacts, we implement a keyframe selection process tailored to multi-view video input. When multiple sequences are available from various cameras or from a single moving camera, we begin by subsampling to reduce redundancy and accelerate processing. Specifically, we extract every k^{th} frame from each video, generating a reduced set X', denoted as $X' = S(X_i, k)$. This subsampling decreases the probability of duplicate or highly similar frames that can destabilize training and reduce robustness [1, 11]. By retaining only essential frames, we improve NeRF efficiency while preserving meaningful scene variation [1].

Next, we address defocus blur, a common issue in real-world video capture that degrades reconstruction precision. Compression artifacts (e.g., JPEG) or motion blur can corrupt frames and lead to inconsistencies in the reconstructed scene. To mitigate this, we assess image sharpness in the frequency domain using the Fast Fourier Transform (FFT) and a blur threshold h_b [4]. Frames whose sharpness falls below this threshold are discarded, ensuring that only distinct and well-defined images contribute to NeRF training. This filtering step reduces reconstruction errors and improves the visual quality of synthesized views [1].

Finally, we further refine X' by removing nearly identical frames produced by slow camera motion. We employ a perceptual hashing function, P_{Hash} [10], to compute compact visual descriptors for each frame, and organize them in a BK-tree [3] for efficient nearest-neighbor search. Using the Hamming distance, we detect and discard redundant frames whose hash distance falls below a threshold τ. This results in a final optimized set X''' that maintains scene diversity while limiting excessive overlap [5]. By carefully balancing frame selection, our approach improves NeRF's ability to reconstruct accurate and photorealistic 3D scenes [1].

*Equal supervision.

3. Data Selection

We begin our approach by considering a set of RGB images from the dataset, denoted as $\mathcal{I} = \{\mathcal{I}_i\}_{i=1}^{n}$, where n is the total number of frames.

From this full set of images, we apply keyframe selection to obtain a subset $\{\mathcal{I}_j^K\}_{j=1}^{k} \subseteq \{\mathcal{I}_i\}_{i=1}^{n}$ of informative frames. We detect and remove duplicates [5] and blurry images [4] to ensure high-quality input.

Blurry image removal proceeds by convolving the input image $\mathcal{I}_i(x, y)$ with a Gaussian kernel $G_\sigma(x, y)$, where σ is the standard deviation of the Gaussian. This convolution yields a blurred image

$$\mathcal{I}_{i,b}(x, y) = \mathcal{I}_i(x, y) * G_\sigma(x, y).$$

We then apply the Fast Fourier Transform (FFT) to transform the blurred image into the frequency domain,

$$\hat{\mathcal{I}}_{i,b}(u, v) = \mathcal{F}\{\mathcal{I}_{i,b}(x, y)\},$$

where u and v are the frequency coordinates. High-frequency components are analyzed and attenuated to characterize blur. Finally, an inverse FFT reconstructs a deblurred image

$$\mathcal{I}_{i,d}(x, y) = \mathcal{F}^{-1}\{\hat{\mathcal{I}}_{i,d}(u, v)\},$$

and frames classified as too blurry according to this analysis are removed.

On the remaining deblurred images $\mathcal{I}_{i,d}$, Near-Image Similarity [5] is applied using perceptual hashing. Each deblurred image is converted to a binary hash code $H(\mathcal{I}_{i,d})$, capturing its visual structure. The Hamming distance d_H between hash codes $H(\mathcal{I}_{i,d})$ and $H(\mathcal{I}_{j,d})$ for $i \neq j$ is computed as:

$$d_H(H(\mathcal{I}_{i,d}), H(\mathcal{I}_{j,d})) = \sum_{k=1}^{L} \delta(H_i[k], H_j[k]), \quad (1)$$

where $\delta(\cdot, \cdot)$ is the bitwise disparity function and L is the hash length. Two images are considered similar if $d_H(\cdot, \cdot) \leq \tau$, with τ a predefined threshold. This process retains overlapping regions that are informative for 3D reconstruction while discarding near-duplicates and blurry frames, resulting in the refined keyframe set $\mathcal{I}^K$, as illustrated in Fig. 2(a) of the main paper.

4. NeuS2: Radiance Field

Since the radiance field backbone is a central component in our framework, Fig. 1 shows the NeuS2 architecture in more detail.

NeuS2 combines a signed distance function (SDF) and a radiance field with visibility-aware rendering to produce high-fidelity meshes, particularly advantageous for the detailed geometries found in food objects.

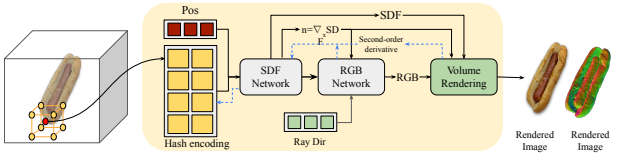


Figure 1. NeuS2 static scene reconstruction architecture used as our radiance-field backbone.

5. Scaling Factor: Distance Matrix

The pipeline for estimating the scaling factor at the coordinate level is illustrated in Fig. 4 (main paper). Our approach is based on matching checkerboard corner projections between 2D keyframes and 3D points from the dense reconstruction.

Using the dense model from PixSfM [6], we retrieve camera poses and the dense point cloud from selected keyframes $\{\mathcal{I}_j^K\}_{j=1}^{k}$. For each keyframe $\mathcal{I}_j^K$ with extrinsic parameters $[R/t]_j$, we first threshold the image at a predefined intensity level φ:

$$I_j^\varphi = \mathbb{I}(\mathcal{I}_j \geq \varphi). \quad (2)$$

We then perform connected component labeling (CCL) on the thresholded image I_j^φ. The connected components $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_m$ are obtained by grouping adjacent pixels into regions. Formally, each component $\mathcal{O}_i$ satisfies:

$$\forall v_a, v_b \in \mathcal{O}_i,\ \exists\, \text{path } p \subseteq \mathcal{O}_i \text{ such that } v_a \leftrightarrow v_b, \quad (3)$$

where v_a and v_b are pixels in the same component and p is a path of adjacent pixels connecting them.

We select the two largest components, denoted $\mathcal{O}_{\max_1}$ and $\mathcal{O}_{\max_2}$, and compute their convex hulls $H_{\max_1}$ and $H_{\max_2}$. Corners are detected within each convex hull region.

Using the intrinsic calibration matrix and the extrinsic parameters $[R/t]_j$, we project the 3D point cloud onto the 2D image plane of keyframe $\mathcal{I}_j^K$. For each detected corner at pixel coordinate x_i^j, we find the nearest corresponding 3D point P_i^j in the projected point cloud. The 3D point is $P_i^j = (P_{i_x}^j, P_{i_y}^j, P_{i_z}^j)$.

The Euclidean distance between two 3D points P_i^j and P_ℓ^j is computed as:

$$D(P_i^j, P_\ell^j) = \sqrt{(P_{i_x}^j - P_{\ell_x}^j)^2 + (P_{i_y}^j - P_{\ell_y}^j)^2 + (P_{i_z}^j - P_{\ell_z}^j)^2}. \quad (4)$$

If we detect n checkerboard corner points in keyframe j, we construct the $n \times n$ symmetric distance matrix D^j, where each entry $D_{i\ell}^j$ corresponds to the distance between

points P_i^j and P_ℓ^j:

$$D^j = \begin{pmatrix} 0 & D(P_1^j, P_2^j) & \cdots & D(P_1^j, P_n^j) \\ D(P_2^j, P_1^j) & 0 & \cdots & D(P_2^j, P_n^j) \\ \vdots & \vdots & \ddots & \vdots \\ D(P_n^j, P_1^j) & \cdots & D(P_n^j, P_{n-1}^j) & 0 \end{pmatrix} \tag{5}$$

From each row of D^j, we compute the minimum non-diagonal value to obtain a vector d^j of minimal neighbor distances:

$$d_i^j = \min_{\ell \neq i} D_{i\ell}^j. \tag{6}$$

This vector captures the approximate physical spacing between adjacent checkerboard corners in 3D.

Finally, we compute the scaling factor as:

$$\text{scale} = \frac{\ell_{\text{square}}}{\text{med}(d^j)}, \tag{7}$$

where ℓ_{square} is the known physical side length of a single checkerboard square, and $\text{med}(d^j)$ is the median of d^j. This scaling factor is used to convert the unitless NeRF reconstruction into metric units (meters) before volume computation.

6. VolETA++ Results: Additional Analyses

Fig. 3 illustrates mesh registration between our generated mesh and the ground truth using ICP. Fig. 4 shows PixSfM results after keyframe selection on MTF and BlendedMVS, demonstrating its ability to refine camera poses and generate dense point clouds under free motion and diverse topologies.

Tables 1 and 2 provide extended quantitative results, including Chamfer distances with and without transformation metrics. Fig. 5 shows the failure cases corresponding to Table 1 (main paper) for scenes 4, 5, and 7, where NeuS2 struggles to reconstruct occluded lower parts of the food objects.

7. Datasets (Extended)

For convenience, we reproduce here the MetaFood3D dataset statistics used in our experiments.

8. Implementation Settings (Details)

The binary threshold for the checkerboard segmentation is a sensitive hyperparameter: small changes significantly affect corner detection quality. We evaluated $\varphi \in \{190, 200, 210, 220, 230, 240\}$ (Fig. 6) and selected $\varphi = 240$ based on visual inspection of segmentation quality.

We set $l_{\text{real}} = 0.012$ m, corresponding to the physical edge length of each checkerboard square. For NeuS2, we use scale 0.15 for all scenes. Other implementation details

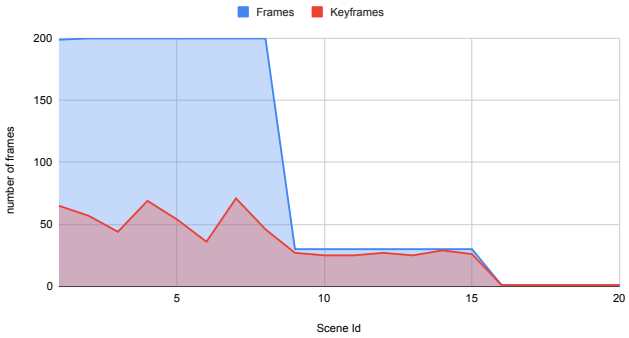


Figure 2. Quantitative analysis of the number of frames before and after keyframe selection. Our approach uses only 34.8% of the frames.

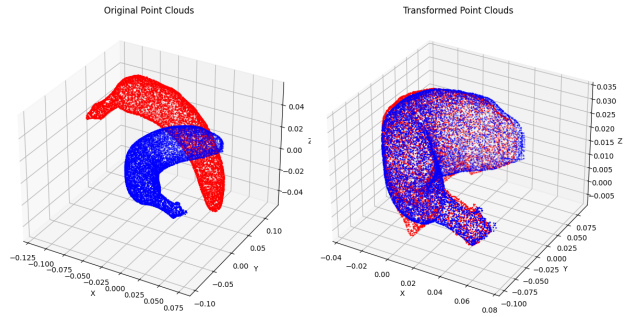
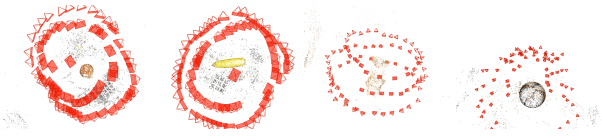


Figure 3. ICP mesh registration between our generated mesh and the ground truth for the *banana_2* scene. Unregistered meshes (left) and registered meshes (right). Our point cloud is red, ground truth is blue.



(a) burger (7) (b) corn (4) (c) angel (d) ball

Figure 4. PixSfM results after keyframe selection on MTF and BlendedMVS, showing refined camera poses and dense point clouds that handle free motion and diverse geometries.

(keyframe selection, Hamming threshold, mesh cleaning) are described in the main paper.

References

[1] Ahmad AlMughrabi, Umair Haroon, Ricardo Marques, and Petia Radeva. Pre-nerf 360: Enriching unbounded appearances for neural radiance fields. *arXiv preprint arXiv:2303.12234*, 2023. 1

[2] RT Azuma. A survey of augmented reality. presence: teleoperators and virtual environments, 6 (4), 355-385, 1997. 1

[3] Walter A. Burkhard and Robert M. Keller. Some approaches to best-match file searching. *Communications of the ACM*, 16(4):230–236, 1973. 1

Table 1. Quantitative comparison of our approach with ground truth on the MTF dataset. We report absolute volume error (cm^3), APE (in %), and Chamfer distance ($\times 10^{-3}$) with and without transformation metrics. Volumes are in cm^3. Methods marked with * denote baselines used in this paper.

Id	Abs. volume error (cm^3) ↓				GT Vol.	APE (%) ↓				Chamfer $\times 10^{-3}$ (w/o t.m.) ↓				Chamfer $\times 10^{-3}$ (w/ t.m.) ↓			
	i-NGP*	NeuS2	SuGar	NeuAng.		i-NGP*	NeuS2	SuGar	NeuAng.	i-NGP*	NeuS2	SuGar	NeuAng.	i-NGP*	NeuS2	SuGar	NeuAng.
1	5.53	1.53	1.53	1.53	38.53	14.352	3.971	3.97	3.97	1.454	0.085	0.477	0.463	0.006	0.0016	0.005	0.016
2	36.36	1.880	11.64	15.64	280.36	12.969	0.671	4.15	5.58	1.275	0.111	0.618	0.479	0.009	0.0061	0.011	0.02
3	209.33	4.011	116.33	102.67	249.67	83.843	1.607	46.59	41.12	0.113	0.173	0.618	0.113	0.012	0.0046	0.016	1.417
4	32.13	25.414	50.13	154.13	295.13	10.887	8.611	16.99	52.22	0.124	0.061	0.469	0.351	0.007	0.0015	0.007	0.007
5	83.58	24.962	39.58	110.58	392.58	21.290	6.359	10.08	28.17	0.106	0.102	0.555	0.458	0.006	0.0025	0.011	0.008
6	19.44	5.908	5.44	31.44	218.44	8.899	2.704	2.49	14.39	0.109	0.151	0.699	0.569	0.099	0.0017	0.009	0.176
7	89.77	28.815	179.77	237.77	368.77	24.343	7.814	48.75	64.48	0.018	0.067	0.691	0.548	0.008	0.0027	0.01	0.033
8	21.13	7.649	359.87	73.13	173.13	12.205	4.418	207.86	42.24	0.037	0.152	0.78	0.383	0.008	0.0019	0.004	0.007
9	39.74	3.112	29.74	131.74	232.74	17.075	1.337	12.78	56.60	0.061	0.16	0.441	0.28	0.009	0.0026	0.008	0.007
10	49.09	6.851	26.09	103.09	163.09	30.100	4.201	16.00	63.21	0.093	0.138	0.469	0.417	0.006	0.0021	0.013	0.008
11	53.18	1.82	1.82	85.18	85.18	62.432	2.137	2.14	100.00	0.095	0.151	0.637	0.513	0.008	0.0033	1.877	0.029
13	81.28	5.960	36.28	171.28	308.28	26.366	1.933	11.77	55.56	0.197	0.148	0.61	0.513	0.009	0.0012	0.009	0.008
14	118.83	48.759	140.83	269.83	589.83	20.146	8.267	23.88	45.75	1.733	0.09	0.537	0.378	0.01	0.0041	0.011	0.011
Mean	64.57	**12.82**	76.85	114.46	–	26.531	**4.156**	31.34	44.10	0.417	**0.122**	0.585	0.420	0.015	**0.00276**	0.1532	0.134
Std	54.33	**14.58**	102.28	79.95	–	22.064	**2.781**	55.186	26.622	0.6192	**0.0379**	0.1042	0.1240	0.0253	**0.0014**	0.5180	0.3881

Table 2. Quantitative comparison of Chamfer distance ($\times 10^{-3}$) with and without transformation metrics on the BlendedMVS dataset. Methods marked with * denote baselines.

Id	Chamfer $\times 10^{-3}$ (w/o t.m.) ↓				Chamfer $\times 10^{-3}$ (w/ t.m.) ↓			
	i-NGP*	NeuS2	SuGar	NeuAng.	i-NGP*	NeuS2	SuGar	NeuAng.
angel	1.588	1.715	2.042	1.186	0.141	0.233	0.041	0.040
ball	0.679	0.894	3.293	4.234	0.244	0.335	0.065	0.118
ball_rock	62.79	63.32	58.97	57.97	0.377	1.034	0.648	0.351
beer	52.89	52.92	47.98	47.19	53.54	0.301	0.238	0.069
bread	1.039	1.071	5.038	5.939	0.163	0.232	0.088	0.065
bule	1561	1561	1553	1553	11.163	0.847	0.557	8.275
clock	0.334	0.484	0.984	0.444	0.022	0.020	0.050	0.098
egg	0.797	1.240	2.637	2.575	0.120	0.107	0.062	0.260
face	1.122	1.508	4.990	5.841	0.271	0.233	0.240	0.703
fox	12.92	12.93	11.68	12.97	0.513	0.401	0.166	3.222
herc	93.59	92.97	90.39	91.84	74.04	1.758	0.207	82.268
ironic_bule	1.137	1.133	2.023	0.960	0.148	0.193	0.567	0.769
ironic_pot	72.67	73.86	71.77	68.03	1.555	1.058	76.433	2.404
lion	0.857	1.120	2.719	1.468	0.109	0.080	0.066	0.050
outdoor	0.483	6.739	2.171	0.339	0.275	6.739	0.151	0.019
plant	0.787	0.927	3.544	4.849	0.045	0.035	0.037	0.560
sweet	0.684	0.731	7.738	9.701	0.097	1.003	0.128	0.056
wolf	0.866	0.935	5.986	6.648	0.048	0.041	0.037	0.589
woman	0.695	0.928	1.935	1.075	0.235	0.169	0.040	0.065
wood	296	296	288	288	269	1.749	1.985	0.638
Mean	**108**	108.6	108.2	108.3	20.6	**0.828**	5.031	4.090
Std	339.95	339.82	337.60	337.61	60.14	1.46	16.60	17.82
Sum	**2163**	2173	2166	2167	413	**16.56**	100.62	81.805
Rel.	+0.04	–	+0.65	+0.65	+4032%	–	+1041%	+1124%

[4] Kanjar De and V Masilamani. Image sharpness measure for blurred images in frequency domain. *Procedia Engineering*, 64:149–158, 2013. 1, 2

[5] Tanuj Jain, Christopher Lennan, Zubin John, and Dat Tran. Imagededup. https://github.com/idealo/imagededup, 2019. 1, 2

[6] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5987–5997, 2021. 2

[7] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017. 1

[8] Fabio Remondino. Heritage recording and 3d modeling with photogrammetry and 3d scanning. *Remote sensing*, 3(6): 1104–1138, 2011. 1

[9] Peter Rogelj, Stanislav Kovačič, and James C Gee. Point similarity measures for non-rigid registration of multi-modal data. *Computer vision and image understanding*, 92(1):112–140, 2003. 1

[10] Christoph Zauner. Implementation and benchmarking of perceptual image hash functions. 2010. 1

[11] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4480–4488, 2016. 1

Table 3. Quantitative comparison of our approach with ground truth using MTF and BlendedMVS. We report MAPE for volume and Chamfer distance with and without transformations (mean and std).

Method	MTF					BlendedMVS			
	MAPE ↓	Chamfer w/o t.m. ↓		Chamfer w/ t.m. ↓		Chamfer w/o t.m. ↓		Chamfer w/ t.m. ↓	
		Std	Mean	Std	Mean	Std	Mean	Std	Mean
i-NGP	26.53	0.656	0.416	0.025	0.0150	339.95	**108.15**	60.14	20.61
NeuS2	**4.26**	**0.038**	**0.095**	**0.004**	**0.0028**	339.82	108.62	**1.46**	**0.83**
SuGar	31.34	0.104	0.585	0.518	0.1534	337.60	108.34	16.60	4.09
NeuAng.	44.10	0.080	0.448	0.388	0.1345	**337.61**	108.21	17.82	5.03

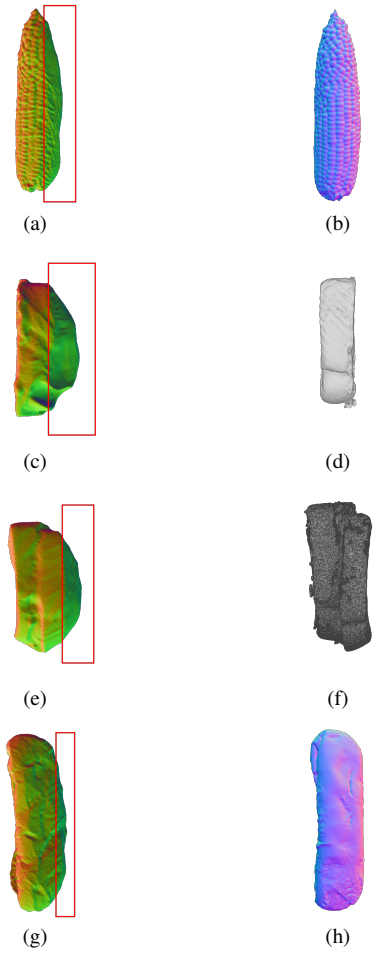
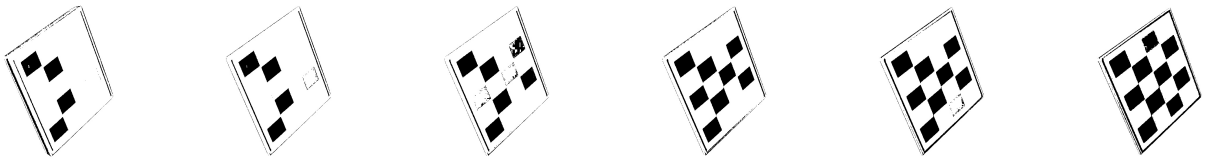


Figure 5. Failure cases: NeuS2 reconstructions (left) versus ground truth (right). NeuS2 fails to reconstruct unseen lower parts of some food items (highlighted in red in the original figures), which affects volume estimation in these challenging scenes.

Table 4. MetaFood3D dataset details and number of images per scene. We do not use one-shot data in our experiments.

L	ID	Food name	# Images
E	1	Strawberry	199
	2	Cinnamon bun	200
	3	Pork rib	200
	4	Corn	200
	5	French toast	200
	6	Sandwich	200
	7	Burger	200
	8	Cake	200
	9	Blueberry muffin	30
M	10	Banana	30
	11	Salmon	30
	12	Steak	30
	13	Burrito	30
	14	Hotdog	30
	15	Chicken nugget	30
H	16	Everything bagel	1
	17	Croissant	1
	18	Shrimp	1
	19	Waffle	1
	20	Pizza	1
			1620



(a) $\varphi = 190$ (b) $\varphi = 200$ (c) $\varphi = 210$ (d) $\varphi = 220$ (e) $\varphi = 230$ (f) $\varphi = 240$

Figure 6. Threshold analysis for the binary image threshold φ on the test set. We empirically found $\varphi = 240$ to provide the most reliable checkerboard segmentation.

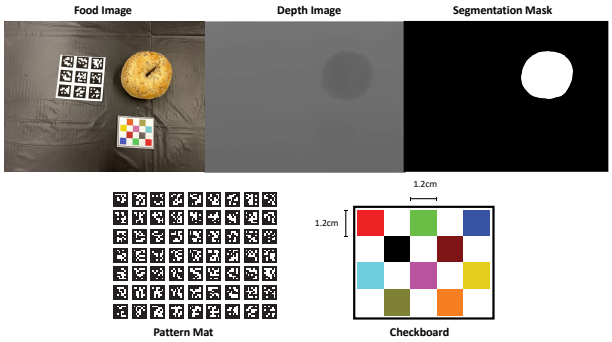


Figure 7. Example MetaFood3D scene: RGB image, depth image, segmentation mask, pattern matrix, and checkerboard.