Pacific Graphics 2023
R. Chaine, Z. Deng, and M. H. Kim
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 7

Supplementary materials: Structure learning for 3D Point Cloud Generation from Single RGB Images

T. Ben Charrada [1], H.Laga [2] and H. Tabia [3]

[1] Reezocar, France. [2] Murdoch university, Australia. [3] Université Paris Saclay, France

1. Training Details

1.1. Training on ShapeNet

1.1.1. Teacher network.

We train the Teacher network using the Chamfer Distance as a loss function; see Equation (1) in the Main Manuscript. We adopt Adam [KB14] as an optimizer and use a learning rate of 10^{-4}. We use a batch size of 128 and train our network for 450 epochs. We noticed during our experiments that batch normalization [IS15] and Dropout [HSK*12] layers prevent the network from converging, thus we do not use them in the Teacher network. To avoid overfitting, we downsample the input point cloud and make use of the Chamfer Distance's robustness to noise. By downsampling the input point cloud, instance-specific details such as car spoilers, car bumpers, and table emboss will **(1)** very likely be removed from the down-sampled point cloud, and **(2)** will be treated as noise/outliers by the Chamfer Distance.

We sample 1024 points from each shape of the training set [CXG*16] of ShapeNet [CFG*15]. We follow the train/validation split of Mesh R-CNN [GMJ19], which results in $798,357$ training instances and $41,832$ validation instances. We train the network for 150 epochs with a fixed learning rate then start decaying the learning rate with a factor of 0.99 at the end of each epoch. The best performance on the validation set happened at the end of epoch 418. During training, the Teacher network takes as input 1024 ground-truth points and outputs 2025 points.

1.1.2. Student network.

To speed up the training while achieving visually-appealing 3D reconstructions, we train the Student network in two phases. In the first phase, we train the encoder part, *i.e.,* the VGG and the fully connected layers, to match the latent vector produced by the second stage of the Teacher network. We initialize the weights of the VGG network using ImageNet's pre-trained weights and adopt Adam [KB14] as an optimizer. We freeze the first five layers of the pre-trained VGG network as well as all the Batch Normalization layers [IS15]. We initialize and freeze the decoder weights with the weights of the decoder of the Teacher network. Then, we optimize it for 25 epochs using a learning rate of 10^{-4} and a weight decay of 10^{-7}. We use the loss function described in Equation (3) in the Main Manuscript. Then, we unfreeze the layers of the encoder and optimize the network for five more epochs.

In the second phase, we freeze the weights of the encoder and train the decoder for 30 epochs using the Latent Distance of Equation (1) in the Main Manuscript.

1.1.3. Refinement network.

We train the refinement network using the loss term of Equation (4) in the Main Manuscript using Adam as optimizer, a learning rate of 3×10^{-6}, and a weight decay of 10^{-6} for a total of five epochs. The refinement network deforms its input point cloud. As such, it outputs a point cloud. To train the refinement network in an adversarial manner, we define a discriminator network, which receives a point cloud as input and outputs a probability value, which represents the likelihood of the input belonging to the ground-truth distribution. The proposed discriminator network relies on PointNet [QSMG17] to extract point-based features. We extract a feature vector of size 1024, which is then processed by a cascade of 2 MLPs having 512 and 1 units, respectively. For the hidden layer, we apply batch normalization and ReLU as a non-linear activation function. For the prediction, we apply the softmax function. We train the discriminator network using the binary cross-entropy loss. For the first epoch, we regularly train both the Discriminator network and the Refinement network, *i.e.,* for each mini-batch, we optimize the Discriminator network and then immediately optimize the Refinement network. For the remaining four epochs, We train the Discriminator periodically using a period of five mini-batches while regularly training the Refinement network. For the Alpha complex algorithm, we use a filtering value of 0.03.

1.2. Training on Pix3D

1.2.1. Teacher network.

We fine-tune, for a total of 600 epochs, the Teacher network (Section 1.1), which was originally trained on ShapeNet [CFG*15]. We adopt the Chamfer Distance of Equation (1) in the Main Manuscript as a loss function and Adam [KB14] as an optimizer. We use a learning rate of 7×10^{-5}, which we decay by a factor of 0.99 after each epoch. We normalize the CAD objects of Pix3D to fit within a cube of 1m. We also center the models at the origin.

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

1.2.2. Student network.

Pix3D [SWZ*18] is composed of real images and slightly misaligned 3D CAD models. To overcome the misalignment issue, we train the Student network in two stages. In the first stage, we train, for 300 epochs, the Student network to map the input images to the embedding vectors of the Teacher network using the loss function of Equation (2) in the Main Manuscript. We adopt Adam [KB14] as an optimizer and use an initial learning rate of 7×10^{-5}. We initialize the weights of the VGG network using ImageNet's pre-trained weights. In the second stage, we apply augmentation on the input images. To take the misalignment between images and CAD models into consideration, we predict, for each image, its corresponding point cloud then align the prediction with the ground truth point clouds using the Iterative Closest Point (ICP) algorithm [BM92].

2. Evaluation Protocols

The Chamfer Distance, which is widely used to evaluate the performance of 3D object reconstruction methods, is not a metric. In fact, it is affected by the scale of the objects and the number of points used to represent the 3D objects. Several protocols have been suggested to address this issue. To ensure a fair comparison, we adopt the following four evaluation protocols.

2.0.1. AttentionDPCR protocol [LXL*19].

This protocol estimates the Chamfer distance on unscaled ShapeNet [CFG*15] models. Both the ground truth and the reconstructed models are represented with $16,384$ points each.

2.0.2. 3D-LMNet protocol [MMAB18].

This protocol normalizes the CAD models so that they fit within a bounding box of unit length (of size 1m). It uses ground truth metadata to rotate the predicted points to the canonical view. The Iterative Closest Point (ICP) algorithm is used to properly align the predicted points and the ground-truth points. A resolution of 1024 points is used for both prediction and ground truth. The reconstruction error is reported in centimeters.

2.0.3. Pixel2Mesh protocol [WZL*18].

This protocol scales the ground truth and the reconstructed 3D objects by a factor of 0.57 to align them with their corresponding rendering. The Chamfer Distance and F_1 scores are then used to assess the quality of the reconstruction. The F_1 score is based on point accuracy. A predicted point is considered accurate if a ground truth point cloud is found within a sphere of a certain radius r. The induced measure is denoted by F_1^r, with $r \in \{0.0001, 0.0002\}$.

2.0.4. Mesh R-CNN protocol [GMJ19].

This protocol scales the ground truth and the reconstructions so that the longest edge is of length 10m. The Chamfer Distance is then evaluated on 10k points. The F_1 score is adopted as a point-accuracy measure and the Absolute Normal Consistency of Equation (1) is used to assess the smoothness of the reconstructed mesh. The F_1^r score uses the l_1 distance to estimate whether a point is found within a radius $r \in \{0.1, 0.3, 0.5\}$.

Table 1: Quantitative comparison on the ShapeNet benchmark [3] following DefTet [GCX*20] and SkeletonNet [THT*21] evaluation protocols. The lower the error the better is the result.

	DefTet protocol						SkeletonNet protocol		
Category	3D-R2N2	DeepMCube	Pixel2Mesh	DISN	DefTet	Ours	IMNet	SkeletonNet	Ours
Plane	2.26	4.80	1.52	1.52	1.49	**1.13**	1.459	**0.771**	0.818
Bench	2.00	7.58	1.62	1.96	1.77	**1.13**	2.020	1.037	**0.709**
Chair	2.83	7.01	2.64	2.51	2.39	**1.44**	1.485	1.138	**1.091**
Firearm	2.26	3.62	1.82	2.15	2.13	**1.15**	1.706	**0.685**	0.947
Table	2.17	6.10	2.20	1.78	1.68	**1.15**	2.540	1.718	**0.848**
Car	1.80	5.79	1.30	1.28	1.18	**1.03**	1.692	0.675	**0.588**
Cabinet	2.02	5.50	1.85	1.61	1.44	**1.24**	1.857	1.468	**0.851**
Couch	2.38	7.40	1.90	1.66	1.58	**1.41**	1.049	1.256	**0.972**
Lamp	4.33	6.39	2.91	3.49	3.53	**2.04**	5.450	2.540	**2.305**
Watercraft	2.69	5.23	2.01	2.29	2.26	**1.42**	2.318	**1.064**	1.085
Monitor	3.01	6.78	1.30	1.28	**1.18**	1.51	2.637	1.316	**1.162**
Speaker	2.94	6.73	2.67	2.21	2.03	**1.74**	3.486	2.446	**1.553**
Cellphone	1.78	6.48	1.59	1.55	1.34	**1.08**	1.088	1.127	**0.723**
Mean	2.50	6.11	2.04	2.04	1.95	**1.34**	2.214	1.326	**1.050**

The Absolute Normal Consistency between a reconstructed point cloud $\mathscr{P}$ and its corresponding ground truth $\mathscr{Q}$ is defined in terms of the angle between the unit normal vectors n_p and n_q at points $p \in \mathscr{P}$ and $q \in \mathscr{Q}$, respectively:

$$\begin{aligned} \mathrm{ANC}(\mathscr{P}, \mathscr{Q}) = & \frac{1}{|\mathscr{P}|} \sum_{p \in \mathscr{P}, q = \mathscr{N}(p)} |\mathbf{n_p} \cdot \mathbf{n}_q| \\ & + \frac{1}{|\mathscr{Q}|} \sum_{q \in \mathscr{Q}, p = \mathscr{N}(q)} |\mathbf{n_p} \cdot \mathbf{n}_q|. \end{aligned} \tag{1}$$

Here, $\cdot$ is the inner product of two vectors. The protocol also uses a subset of the original ShapeNet [CFG*15] test set, which contains objects of complex topological structures, referred to as holes test set. It is designed to assess the performance of methods on reconstructing meshes of arbitrary topological structures.

2.0.5. SkeletonNet protocol [THT*21]

This protocol estimates the Chamfer distance on unscaled ShapeNet [CFG*15] models. Both the ground truth and the reconstructed models are represented with $10,000$ points each.

2.0.6. DefTet protocol [GCX*20]

This protocol estimates the Chamfer distance on unscaled ShapeNet [CFG*15] models. Both the ground truth and the reconstructed models are represented with 100K points each.

3. Implicit functions

Explicit volumetric representations suffer from problems such as limited resolution. Implicit functions [MON*19, CZ19, XWC*19, CLW21] alleviate the memory footprint problem. An implicit function receives the coordinates of a 3D point and outputs its corresponding occupancy or signed distance value. At inference time, implicit functions are capable of generating 3D reconstructions at a user-selected resolution. Additionally, a deep learning network that generates implicit functions does not operate directly on voxels, which results in a low memory footprint. However, these methods require time-consuming post-processing operations to identify the iso-surface and extract the 3D mesh, *e.g.,* by using the Marching Cubes algorithm.

In Table 1, we extend the performance comparison to include implicit function-based methods. We compare our method

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

Table 2: In this ablation study, we analyze the effect of the different components of our framework. We report the Chamfer Distance computed between 1024 un-scaled ShapeNet [CFG*15] ground truth points and 2048 reconstructed points. All values should be multiplied by 10^{-3}. The ablation of the Teacher network is performed after training for 15 epochs.

Student	No Teacher	ResNet50	VGG11	No LD	Full
	4.031	2.635	1.572	2.19	**1.338**
Teacher	No reg	Batch norm	Dropout	-	Full
	1.471	1.693	1.535	-	**1.169**

to DISN [XWC*19], DeepMarchCubes [LDG18], and IM-Net [CZ19]. As one can see, the proposed method outperforms DISN [XWC*19], which is the most accurate implicit function-based method among the considered ones, by 38%. We also compare our method to SkeletonNet [THT*21], which relies on an intermediate volumetric representation to reconstruct the surface of the 3D object. Our method is 20% more accurate than SkeletonNet [THT*21]. Finally, we compare the performance of our proposed method to DefTet [GCX*20], which predicts the occupancy of a tetrahedral grid. As seen in Table 1, the proposed method outperforms DefTet [GCX*20] by 31%.

4. Evaluation on additional images

Figure 1 provides additional results, which could not fit in the Main Manuscript. In this figure, we show the input images (first column) followed by the reconstruction results obtained using our method. We also show the ground truth (last two columns) for comparison. We can clearly see that our reconstructions exhibit a nice visual aspect compared to the ground truth points; see for example rows 2, 3, and 4. The surface extracted points, such as ground truth points showcased in rows 1 and 2 of Figure 1, are randomly sampled and have no structure. Our proposed framework relies on the annotations of the Teacher network and a novel loss function to learn to reconstruct structured points.

In Figure 3, we present further examples demonstrating our Teacher's capacity to reconstruct detailed structures. The teacher network reconstructs a structured, or parametrized, point cloud (Columns 2 and 4) from an input point cloud (Columns 1 and 3). The Teacher network was trained using the Chamfer distance and as a result, the reconstructions shown in Figure 3 present some minor noise.

5. Ablation Study

We undertake an ablation study to assess the contribution of each of the components of the proposed framework.

Training methodology. To assess the importance of the proposed Student-Teacher training methodology, we train our Student network without the Teacher annotations. In this case, we train the Student network to reconstruct point clouds in an end-to-end manner using the Chamfer Distance as a loss function (Equation 1 in the main paper). As illustrated in Table 2, the student network fails to converge without the annotations of the Teacher network. This is not a surprise as it explains why most state-of-the-art methods rely on more complex architectures to solve the single view-based 3D object reconstruction problem.

Latent distance. We train our student network with the Chamfer Distance instead of the newly defined loss function. As seen in Table 2, the ablated version of our model performs poorly compared to the full model. The Chamfer Distance does not establish a one-to-one correspondence between points. On the other hand, our latent loss addresses this limitation by using a deep feature extractor that generates a global feature vector. It then compares points in the latent space. From this observation, we conclude that our proposed Latent Distance is more accurate in comparing point clouds.

Refinement. We evaluate the performance of our proposed method without the refinement stage and report the results in Table 2 in the main paper. The proposed Refinement network results in a performance increase, in terms of CD, of more than 20%.

Regularization. We down-sample the ground-truth points to avoid over-fitting our training set. Traditional regularization methods include batch normalization [IS15] and Dropout [HSK*12]. We train our Teacher network using various regularization methods and report the results in Table 2. Our novel regularization method results in a faster convergence compared to the traditional methods that are commonly used in 3D point cloud reconstruction.

Image feature extractor. We adopt VGG19 [SZ14] to extract deep features from the input RGB images. We compare the performance of various feature extractors and report their respective results in Table 2.

Alpha shape. The alpha complex algorithm generates meshes of high resolution as seen in Table 3. We use the alpha shape algorithm (*i.e.,* we only keep the boundary of the alpha complex reconstruction) to generate meshes of lower resolutions. This results in more consistent normal vectors; see normal consistency in Table 3, and a lower reconstruction accuracy; see the CD and F_1 scores in Table 3.

Structure. The Teacher network folds a 2D grid to reconstruct 3D objects. As stated by FoldingNet [YFST18], this operation establishes a mapping from a 2D regular domain to a 3D point cloud. In this paper, we make the following hypothesis: a mapping from a regular 2D grid to a 3D point cloud engenders structure. The framework proposed in this paper relies on this hypothesis to propose a lightweight model for single-view 3D point cloud reconstruction. As this structure generated by the Teacher network is regularized but not supervised, the annotations of the Teacher network are pseudo-labels. To validate the hypothesis, we compare the ground-truth point clouds to the point clouds reconstructed using the Teacher network. As one can see in Figure 2, the flat surfaces of the Teacher's reconstructions have grid-like properties, *i.e.,* they look uniformly distributed.

Noise. To assess the sensitivity to noise of the proposed LD and CD, we add random noise to the reconstructed points. First, we reconstruct using the Student network 10K points. Then, using a

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

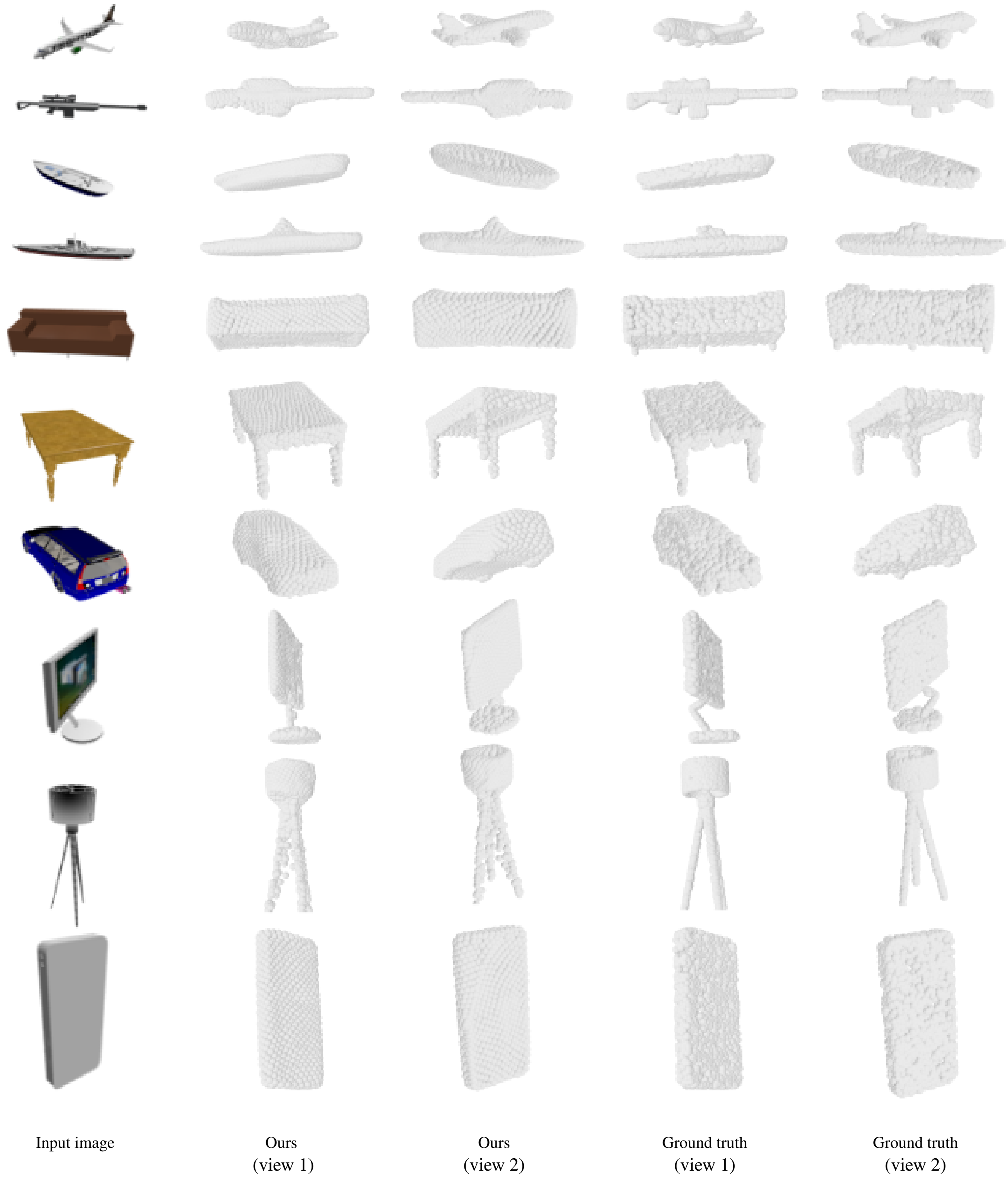


Figure 1: Additional on the ShapeNet Benchmark [CFG*15].

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

Table 3: Reconstruction error on ShapeNet using the scale-invariant protocol of Mesh R-CNN. We compare to the state-of-the-art and to an ablated model of Mesh R-CNN.

	Full Test Set							Holes Test Set						
	CD ↓	Normal ↑	$F_1^{0.1}$ ↑	$F_1^{0.3}$ ↑	$F_1^{0.5}$ ↑	$\lvert V\rvert$	$\lvert F\rvert$	CD ↓	Normal ↑	$F_1^{0.1}$ ↑	$F_1^{0.3}$ ↑	$F_1^{0.5}$ ↑	$\lvert V\rvert$	$\lvert F\rvert$
Pixel2Mesh	0.265	0.729	29.9	76.2	89.0	2466 ±0	4928 ±0	0.273	**0.733**	30.8	76.5	88.9	2466 ±0	4928 ±0
Mesh R-CNN (Best)	0.133	0.729	38.8	86.8	95.1	1899 ±928	3800 ±1861	0.130	0.725	41.7	86.7	94.9	2291 ±903	4595 ±1814
Mesh R-CNN (Pretty)	0.171	0.713	35.1	82.6	93.2	1896 ±928	3795 ±1861	0.171	0.700	37.1	82.4	92.7	2292 ±902	4598 ±1812
Ours (Alpha shape)	0.114	**0.731**	41.3	88.3	96.0	2120 ±502	7729 ±2008	0.116	0.705	41.4	87.7	95.7	2120 ±502	7729 ±2008
Ours	**0.108**	0.611	**44.5**	**89.2**	**96.3**	3161 ±569	31311 ±7455	**0.108**	0.588	**46.5**	**88.9**	**96.0**	3165 ±576	30887 ±6537

selection probability of 2%, we randomly select points from the reconstructed 10k points and deform them by 10cm. Finally, we report the CD and the LD between the noisy reconstructions and the ground truth points.

We recall that the Chamfer distance between point sets $\mathscr{P}$ and $\mathscr{Q}$ is given by:

$$\mathrm{CD}(\mathscr{P},\mathscr{Q}) = \frac{1}{|\mathscr{P}|}\sum_{p\in\mathscr{P}}\min_{q\in\mathscr{Q}}\|p-q\|_2^2 + \frac{1}{|\mathscr{Q}|}\sum_{q\in\mathscr{Q}}\min_{p\in\mathscr{P}}\|p-q\|_2^2, \qquad (2)$$

After attaining a certain precision, the CD becomes insensitive to subtle changes. To demonstrate this, we can select one point i_{old} from set $\mathscr{P}_{\text{old}}$ and deform it by offset in any direction ($i = i_{\text{old}} +$ offset). The resulting CD between the new point set $\mathscr{P}$ and the target $\mathscr{Q}$ is then:

$$\begin{aligned}\mathrm{CD}(\mathscr{P},\mathscr{Q}) = \frac{1}{|\mathscr{P}|}\sum_{p\in\mathscr{P},p\neq i}\min_{q\in\mathscr{Q}}\|p-q\|_2^2 + \min_{q\in\mathscr{Q}}\|i-q\|_2^2 + \\ \frac{1}{|\mathscr{Q}|}\sum_{q\in\mathscr{Q}}\min_{p\in\mathscr{P}}\|p-q\|_2^2,\end{aligned} \qquad (3)$$

We note $q_{\text{old}} = \arg\min_{q\in\mathscr{Q}}\|i_{old}-q\|_2^2$. We know that:

$$\begin{aligned}\min_{q\in\mathscr{Q}}\|i-q\|_2^2 \leq \|i-q_{\text{old}}\|_2^2 = \\ \|i_{\text{old}}+\text{offset}-q_{\text{old}}\|_2^2.\end{aligned} \qquad (4)$$

$$\begin{aligned}\|i_{\text{old}}+\text{offset}-q_{\text{old}}\|_2^2 = \|i_{\text{old}}-q_{\text{old}}\|_2^2 + \\ \|\text{offset}\|_2^2 + 2\times(i_{\text{old}}-q_{\text{old}},\text{offset}).\end{aligned} \qquad (5)$$

Applying Cauchy-Swatrz with $x = i_{\text{old}} - q$ and $y = $ offset, we obtain:

$$(i_{\text{old}}-q_{\text{old}},\text{offset}) \leq \|i_{\text{old}}-q_{\text{old}}\|_2\|\text{offset}\|_2. \qquad (6)$$

Therefore:

$$\begin{aligned}\|i_{\text{old}}+\text{offset}-q_{\text{old}}\|_2^2 = \|i_{\text{old}}-q_{\text{old}}\|_2^2 + \\ \|\text{offset}\|_2^2 + 2\times\|i_{\text{old}}-q_{\text{old}}\|_2\|\text{offset}\|_2.\end{aligned} \qquad (7)$$

Now if $\|\text{offset}\|_2$is larger than the error $\|i_{\text{old}}-q_{\text{old}}\|_2$, and by combining Equations (3) and (7), we obtain:

$$\mathrm{CD}(\mathscr{P},\mathscr{Q}) \leq \mathrm{CD}(\mathscr{P}_{old},\mathscr{Q}) + \frac{3}{|\mathscr{P}|}\|\text{offset}\|_2^2. \qquad (8)$$

In our case, $|\mathscr{P}| = 10\text{K}$, offset $= 0.01$, and the average CD at training stage ($\mathrm{CD}(\mathscr{P}_{\text{old}},\mathscr{Q})$) is around 3×10^{-4}. Therefore, after offsetting the reconstruction, the new CD would, at most, be higher by 3×10^{-8}, which is insignificant compared to $\mathrm{CD}(\mathscr{P}_{\text{old}},\mathscr{Q})$.

Resampling. We compare the qualitative aspect of our reconstructions before and after the resampling. As seen in Figure 2, points that are resampled look more similar, in terms of patterns, to the ground-truth points than the raw reconstructions. Additionally, our raw reconstructions look more similar to the Teacher annotations than to the ground-truth points. This is expected since we use the Teacher's annotations to train the Student network. As one can see in Figure 2, both the Teacher annotations and the raw reconstructions have grid-like patterns. These patterns are more observable on flat surfaces such as the tabletop of the second row of Figure 2.

Teacher. We report the quantitative performance of the Teacher network in Table 2 of the main paper. The Teacher network has an F_1^τ score that is 8% higher than the refined network. We note that the Teacher network auto-encodes 3D point clouds and has a drastically easier task than the Student network.

We also provide a qualitative evaluation in Figure 2. The Teacher network was trained using the Chamfer Distance. Therefore, it has some noisy reconstructions as seen in the second and third rows of Figure 2.

6. Implementation of the training pipeline

Algorithm 1 outlines the training pipeline of the Student Network. For each mini-batch, the training function takes as input the ground truth points $\mathscr{P}$ and the input image I. At first, we re-annotate (rearrange/structure) the points using the Teacher network, *i.e.,* $\mathscr{P}$ undergoes a full encoding-decoding cycle. Second, we extract a latent descriptor of the annotated points using the Teacher's encoder. We then generate an estimation of the latent space using the Student encoder and decode it to obtain its respective reconstruction. We encode the student reconstruction using the Teacher encoder. This allows us to compute the Latent Distance which compares the latent descriptor of the annotation to that of the reconstruction. Finally, we compute the total loss as stated in Equation 3 in the main manuscript.

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

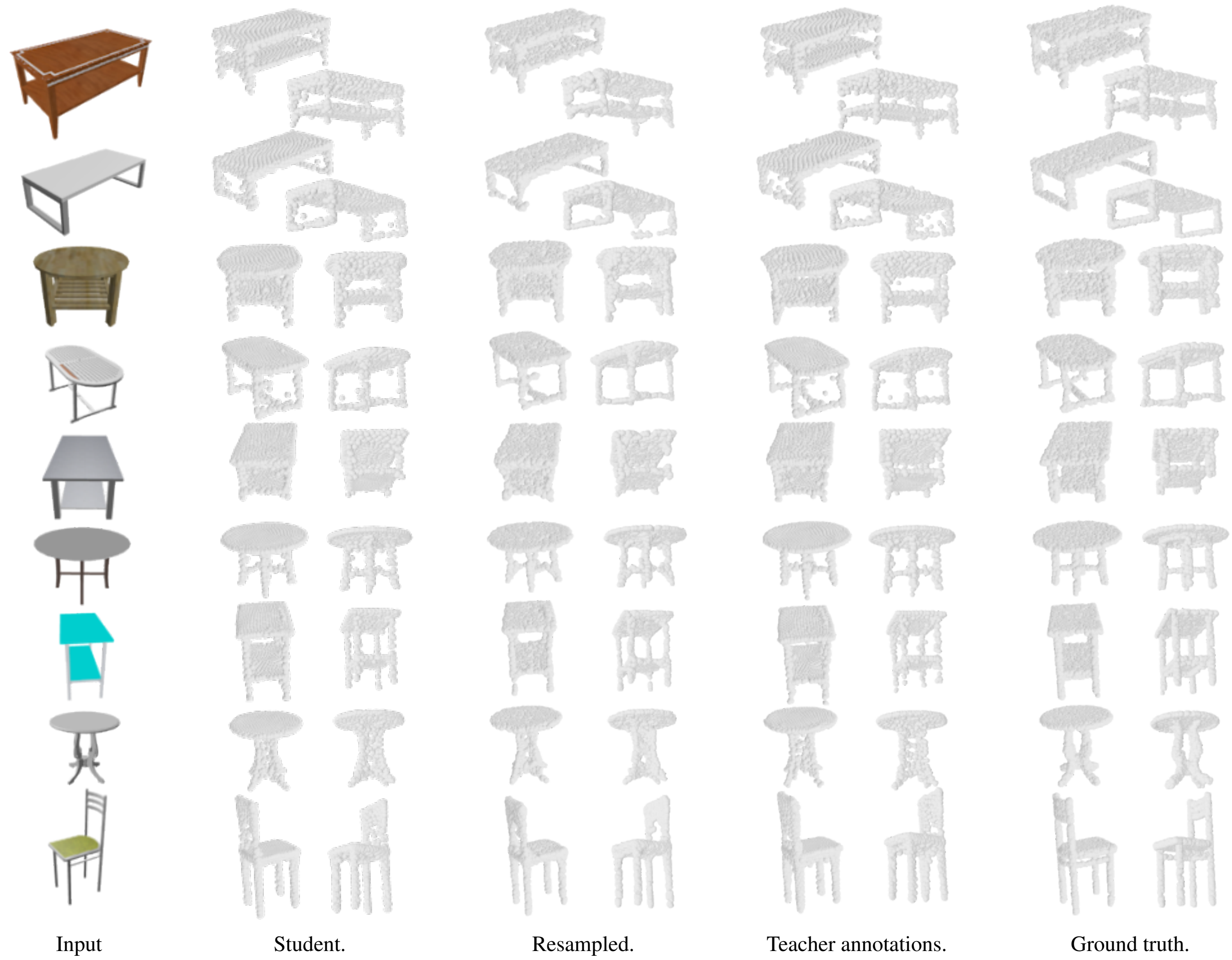


Figure 2: Qualitative comparison, using ShapeNet [CFG*15], of the visual aspect of our reconstructions. We show the the reconstructions of our method before (raw) and after refinement, refined points, the annotations of our Teacher network, and, we compare them to the ground truth points.

Algorithm 1 The proposed training algorithm of the student network.

```
Require: Ground truth points $\mathscr{P}$, Input image $I$.
function TRAIN($\mathscr{P}$,$I$)
    $Annotations \leftarrow$ Teacher($\mathscr{P}$)
    $Latent\_gt \leftarrow$ Teacher.Enc($Annotations$)
    $Latent\_s \leftarrow$ Student.Enc($I$)
    $Reconstruction \leftarrow$ Student.Dec($latent_s$)
    $Latent\_gen \leftarrow$ Teacher.Enc($Reconstruction$)
    $Latent\_loss \leftarrow$ MSE($Latent\_gen$,$Latent\_gt$)
    $Loss \leftarrow Latent\_loss +$
                MSE($Latent\_s$,$Latent\_gt$)
end function
```

References

[BM92] Besl P. J., McKay N. D.: Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures* (1992), vol. 1611, International Society for Optics and Photonics, pp. 586–607. 2

[CFG*15] Chang A. X., Funkhouser T., Guibas L., Hanrahan P., Huang Q., Li Z., Savarese S., Savva M., Song S., Su H., et al.: Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015). 1, 2, 3, 4, 6

[CLW21] Chen Y., Liu S., Wang X.: Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 8628–8638. 2

[CXG*16] Choy C. B., Xu D., Gwak J., Chen K., Savarese S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision* (2016), Springer, pp. 628–644. 1

[CZ19] Chen Z., Zhang H.: Learning implicit fields for generative shape modeling. In *IEEE CVPR* (2019), pp. 5939–5948. 2, 3

[GCX*20] Gao J., Chen W., Xiang T., Jacobson A., McGuire M., Fidler S.: Learning deformable tetrahedral meshes for 3d reconstruction. *Advances In Neural Information Processing Systems 33* (2020), 9936–9947. 2, 3

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

[GMJ19] Gkioxari G., Malik J., Johnson J.: Mesh r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 9785–9795. 1, 2

[HSK*12] Hinton G. E., Srivastava N., Krizhevsky A., Sutskever I., Salakhutdinov R. R.: Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580* (2012). 1, 3

[IS15] Ioffe S., Szegedy C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (2015), PMLR, pp. 448–456. 1, 3

[KB14] Kingma D. P., Ba J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 1, 2

[LDG18] Liao Y., Donné S., Geiger A.: Deep Marching Cubes: Learning Explicit Surface Representations. In *IEEE CVPR* (2018), pp. 2916–2925. 3

[LXL*19] Lu Q., Xiao M., Lu Y., Yuan X., Yu Y.: Attention-based dense point cloud reconstruction from a single image. *IEEE Access 7* (2019), 137420–137431. 2

[MMAB18] Mandikal P., Murthy N., Agarwal M., Babu R. V.: 3D-LMNet: Latent Embedding Matching for Accurate and Diverse 3D Point Cloud Reconstruction from a Single Image. *BMVC* (2018), 662–674. 2

[MON*19] Mescheder L., Oechsle M., Niemeyer M., Nowozin S., Geiger A.: Occupancy Networks: Learning 3D Reconstruction in Function Space. *IEEE CVPR* (2019). 2

[QSMG17] Qi C. R., Su H., Mo K., Guibas L. J.: PointNet: Deep learning on point sets for 3d classification and segmentation. In *IEEE CVPR* (2017), pp. 652–660. 1

[SWZ*18] Sun X., Wu J., Zhang X., Zhang Z., Zhang C., Xue T., Tenenbaum J. B., Freeman W. T.: Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 2974–2983. 2

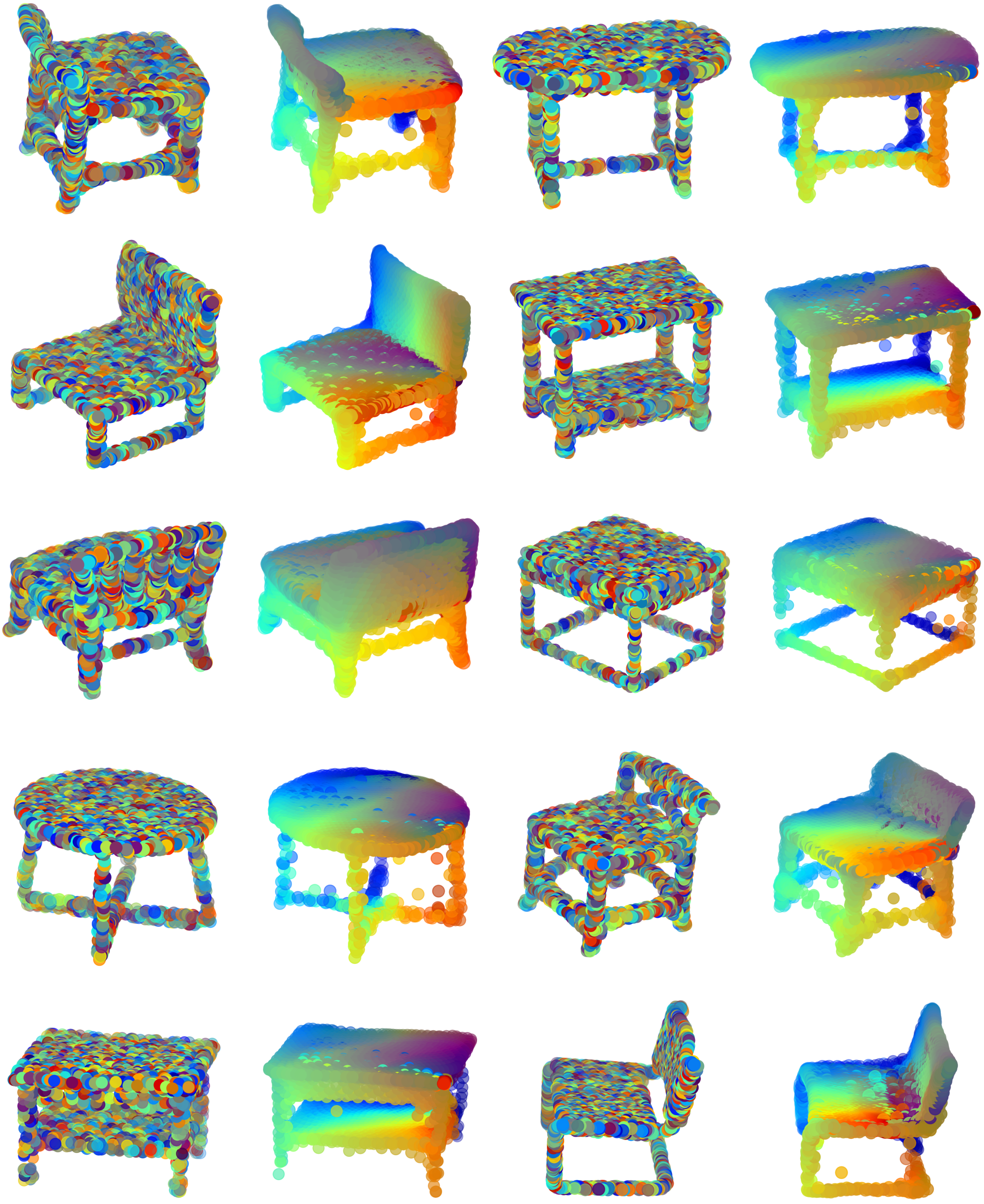
[SZ14] Simonyan K., Zisserman A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014). 3

[THT*21] Tang J., Han X., Tan M., Tong X., Jia K.: Skeletonnet: A topology-preserving solution for learning mesh reconstruction of object surfaces from rgb images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021). 2, 3

[WZL*18] Wang N., Zhang Y., Li Z., Fu Y., Liu W., Jiang Y.-G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 52–67. 2

[XWC*19] Xu Q., Wang W., Ceylan D., Mech R., Neumann U.: Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *Advances in Neural Information Processing Systems 32* (2019). 2, 3

[YFST18] Yang Y., Feng C., Shen Y., Tian D.: Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 206–215. 3



Input. Reconstruction. Input. Reconstruction.

Figure 3: Additional examples highlighting our Teacher's capacity to reconstruct detailed structures. The Teacher network reconstructs a structured, or parametrized, point cloud (Reconstruction) from an input point cloud.

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.