A paper review on how to reconstruct high quality and detail-rich 3D shapes from 2D images

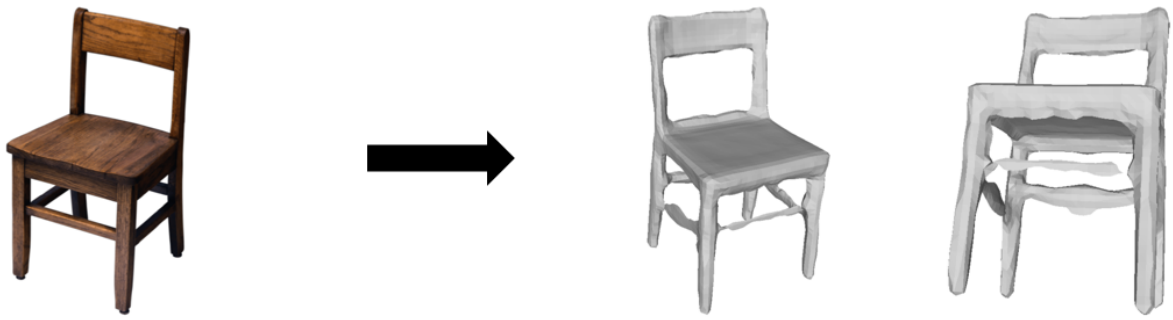

Figure 1: 3D shape reconstruction from a 2D image using DISN . Taken from [1]

While humans are quite good at recognizing objects and deriving their properties, for machines it is a rather complex task to recover a 3D shape from a single view. Since this capability is a core technology, necessary in a variety of fields, it is an important object of research in 3D computer vision.
Major progress has been achieved here, especially in the last few years through the introduction of deep learning. While most recent work already delivers quite decent results on recovering the overall shape, retrieving fine-grained details was not a major focus so far. In practice, this means small structures like holes have mostly been ignored in the reconstruction processes. To tackle this drawback Wang et al. presented “DISN: Deep Implicit Surface Network for High-quality single-view 3D Reconstruction” at the Conference on neural information processing systems (NeurIPS) 2019.

In their publication, a Neural Network is presented as being capable of reconstructing both a high-quality overall shape as well as fine-grained details. While this blog post is about presenting their work in a more understandable manner, the original paper, as well as the official code, can be found here.

How does contemporary research solve the problem of single-view 3D reconstruction?

Modern research shows that as of now deep learning is the state-of-the-art technique for single-view 3D reconstruction. However, apart from the Neural Network structures,

the approaches differ in the way 3D shapes are represented in the networks. Therefore, we can cluster the related work into two distinct representation methods:

- **explicit methods —** describe a 3D model as a solid using e.g. point clouds, voxels or meshes. The main advantage of such a method is its intuitiveness which also makes them easy to encode e.g. in a Neural Network. However, these methods suffer from limited resolution and/or fixed mesh topologies. Further, traditionally applied training losses like Earth-mover Distance (EMD) or Chamfer Distance (CD) only approximate the similarity of shape and are therefore not completly accurate. Examples that were compared to DISN are *AtlasNet*[2], *Pixel2Mesh*[3], and *3DN*[4]. While the first uses a set of parametric elements to generate 3D surfaces, the latter two reconstruct 3D shapes by deforming a given source mesh. For this, Pixel2Mesh uses a hardcoded Ellipsoid-mesh while 3DN expects the source Mesh as an input.

- **implicit methods —** in contrast, define a surface by using a volumetric scalar function. If the equation $F(X, Y, Z) = 0$ holds, then a point $P(X, Y, Z)$ is said to be on the surface. A common function F is the Signed-Distance-Function (SDF). An SDF maps a point P to a real value $s \in \mathbb{R}$ where the sign of s tells whether P is inside or outside of the 3D shape and the absolute value gives the distance of P to the isosurface. As this function is continuous, objects are represented with arbitrary resolution.

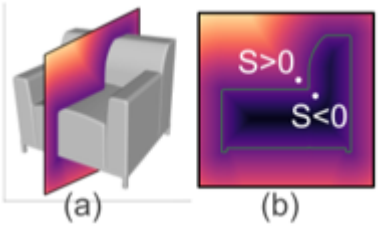

Figure 2: In (b) the SDF values of the rendered object in (a) are visualized. S is positive if outside and negative if inside. If S equals 0 ones knows that the corresponding point is part of the iso-surface. Taken from [1]

While in the here presented approach a SDF is predicted, in recent works like *IMNet*[5] or *OccNet*[6] a binary version of F is predicted – only telling whether a point is insider or outside. While none of these works has been capable of reconstructing fine-grained details, they have shown to be capable of avoiding the drawbacks of explicit methods.

A two-step approach

To achieve the goal of reconstructing both overall shape as well as fine-grained details, Wang et al. predict an SDF. They developed a feed-forward neural network that takes a single 2D image and a point in world coordinates $P(X, Y, Z)$ and returns the corresponding SDF value. Internally, this is done by using two consecutive networks: The first estimates the camera pose to map an object in world space to the image plane. Having this mapping a local feature extraction module is employed in the second (SDF predicting) network – additionally to a global feature encoder.

How is the camera pose estimated?

For camera pose estimation the authors use the general approach proposed by Insafutdinov and Dosovitskiy [7]. By using a Convolutional Neural Network several pose candidates are combined. However, their approach suffers from a large number of network parameters and a complex training procedure.

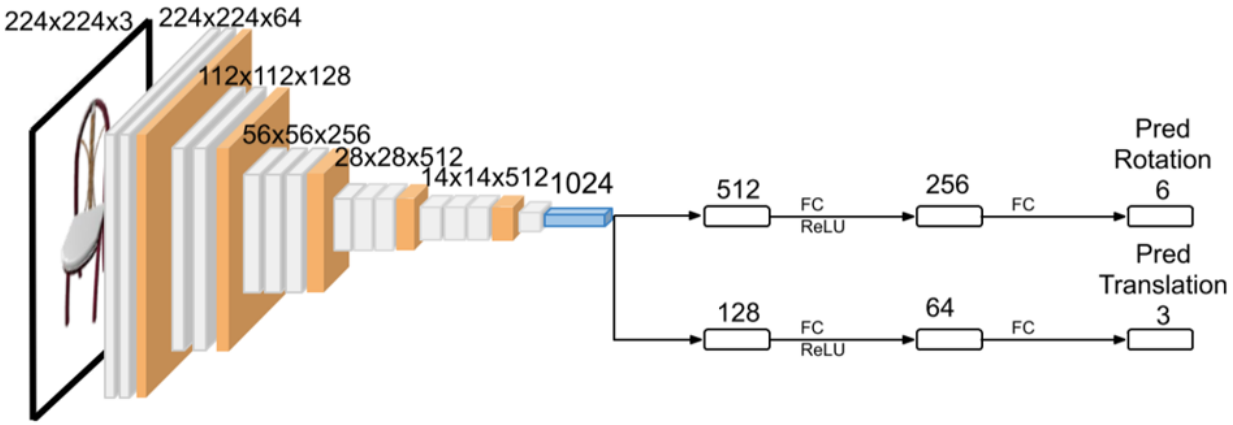

Figure 3: The camera pose estimation network. Taken from supplementary of [1]

To reduce these disadvantages, the authors of DISN make use of recent research results, that continuous representations are easier to regress for Neural Networks. Zhou et al. have shown that e.g. a 6D rotation representation $b = (b_x, b_y)$ where $b \in \mathbb{R}^6, b_x \in \mathbb{R}^3, b_y \in \mathbb{R}^3$ is continuous, while quaternions and Euler angles are not, and is, therefore, better suited for regression in neural networks. Once b is predicted, the rotation matrix $R = (R_x, R_y, R_z)^T \in \mathbb{R}^{(3x3)}$ is obtained with the following equations:

$$R_x = N(b_x),\ R_z = N(R_x \times b_y),\ and\ R_y = R_z \times R_x$$

with $N(\cdot)$ being the normalization function and '$\times$' the cross product. [8]
Translation $t \in \mathbb{R}^3$ is predicted directly.

Loss calcualtion for pose estimation

When training this network (see also figure 3), the authors use the ShapeNet Core dataset [9] , where all objects are within the same aligned model space, and the renderings provided by Choy et al. [10]. The model space of the original dataset is then set as the world space with all camera parameters in respect to. For regression, a given world space point cloud PC_w is transformed to camera space using predicted parameters and then compared to the camera space ground truth point cloud PC_{cam}. As a regression loss they use the Mean-squared-error (MSE) resulting in:

$$L_{cam} = \frac{\sum_{p \in PC_w} ||p_G - (Rp_w + t)||_2^2}{\sum_{p \in PC_w} 1}$$

How is the Signed Distance Function predicted?

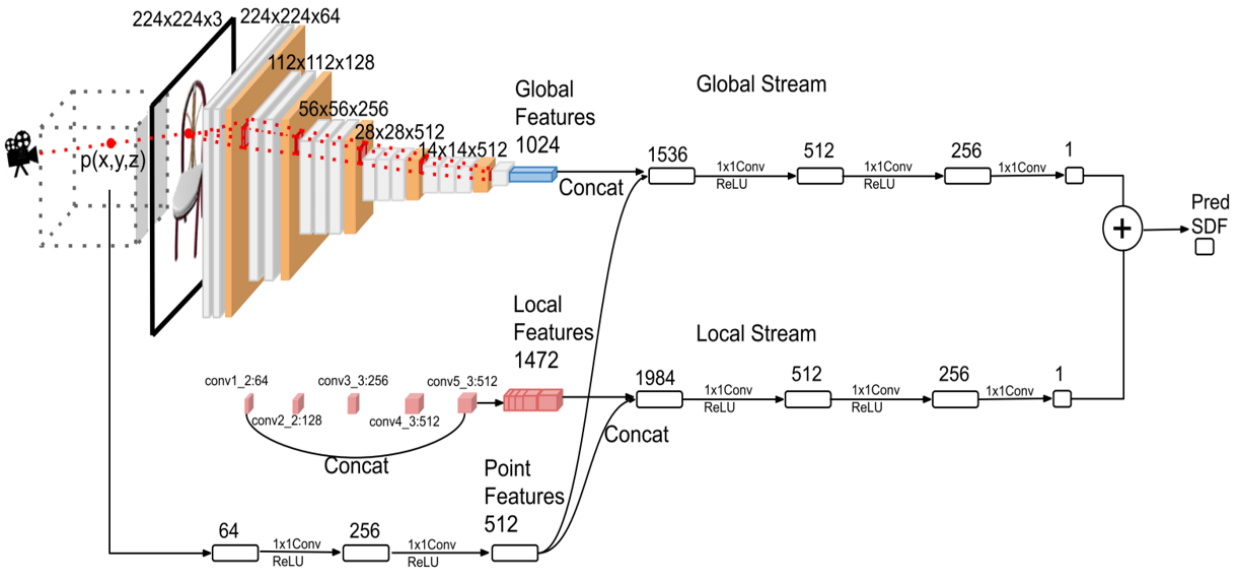

Figure 4: The SDF network model. Taken from [1]

The SDF prediction network consists of three encoders:

1. A simple VGG-16 Encoder that extracts global features from the 2D image.
2. A local feature extraction module. It uses the estimated camera pose to project the point $P \in \mathbb{R}^3$ onto a 2D location $q \in \mathbb{R}^2$ on the image plane. Having q in each feature map the corresponding part is extracted and then concatenated – resulting in the embedding vector. As not all feature maps equal the size of the input image, bilinear interpolation is used to resize the feature maps and extract the values.
3. A multilayer perceptron which maps the given point to a higher dimensional feature space. This is then concatenated to both the global and local features.

Having the global and local features encoded together with the higher dimensional query point, the two embedding vectors are then decoded separetly. This results then in

an SDF value for the overall shape for the former, and a *residual* SDF for the later. Combining them, trough simple summation, results in an SDF that in addition to an overall shape also recovers the in previous approaches missing details of an object.

Loss calculation for SDF prediction

For the loss calculation of the network, two things have to be taken into consideration. First, in contrast to e.g., IMNet one wants to recover a continous function and second, the network should concentrate on details near and inside the iso-surface. This, in consequence, then leads to a weighted loss function of SDF values being defined as:

$$L_{SDF} = \sum_{p} m|f(I,p) - SDF^{I}(p)|$$

$$m = \begin{cases} m_1, & SDF^{I}(p) > \delta \\ m_2, & \text{otherwise} \end{cases}$$

Evaluation of DISN

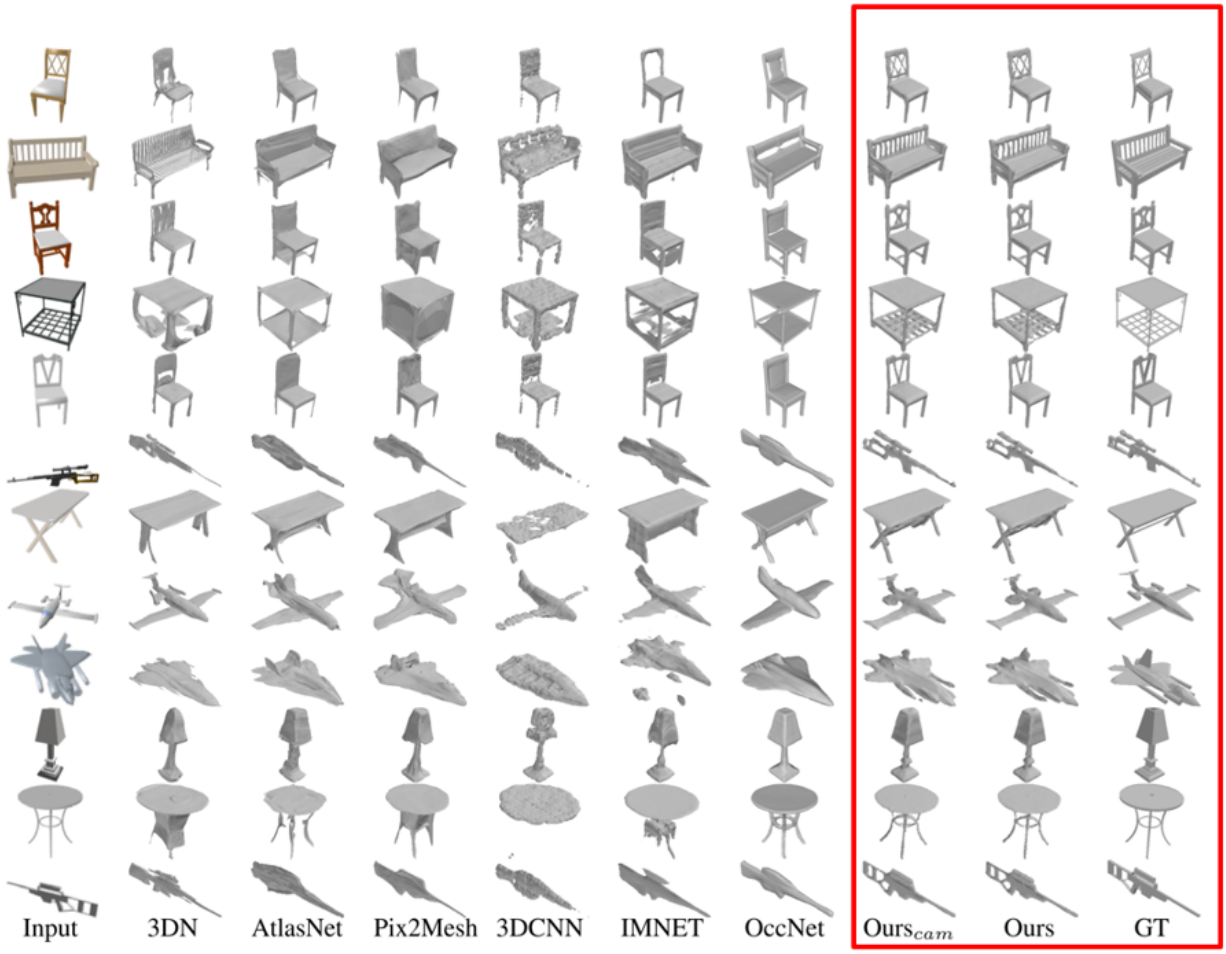

Figure 5: Single-view 3D reconstruction results of DISN and the other presented methods.

The ground truth is denoted by 'GT'. Taken from [1]

In order to evaluate whether the goal of reconstructing high-quality fine-grained 3D shapes has been achieved extensive evaluation and comparison against the previously mentioned methods [2-6] have to been done. To do this, qualitative (visual), as well as quantitative evaluation results on single-view 3D reconstruction, are provided. Additionally, the performance of the adapted camera pose estimation is examined against the original approach. In the last step Ablation studies have been conducted showing again qualitative as well as quantitative results.

Experimental setup

This includes the Dataset as well as the preparation training and testing implementation.

Dataset

For the experiments the ShapeNet Core **[...]** dataset was used. According to the official website it "is a subset of the full ShapeNet dataset with single clean 3D models and manually verified category and alignment annotations. It covers 55 common object categories with about 51,300 unique 3D models." However, to make the evaluation comparable, the official training/test split on 13 object categories is used. Furthermore, to obtain 2D images the renderings of Choy et al. [10] are employed. This is quite good work as most of the aforementioned other approaches [2-4,6] employed the same settings for their evaluation.

As an additional contribution they rendered a new 2D dataset that contains 5 degrees of freedom (DoF) at a Resolution of 224×224 *– pairing each image "with a depth image, a normal map and an albedo image provided by blender" as well. While this is certainly an improvement, when compared to Choy et al., who only provide 3 DoF and a resolution of* 137×137*, it is not used in the official evaluation"*

Data Preparation

In the data preparation step, two things must be done: Firstly, ground truth data for camera pose estimation is needed. To achieve this, the rendering of Choy et al. is used. The renderings provide different viewpoints of the objects in the main data set together with annotation of their transformation from world to camera space. Secondly, SDF ground truth data has to be generated. Following the approaches of [11,12] this is done by an SDF grid resolution of 256^3. But, as one is mostly interested in SDF values close to the iso-surface it is not necessary to train on all $256^3 = 16,777,216$ values. To

reduce this number, Monte Carlo sampling under Gaussian distribution $\mathbb{N}(0,0.1) is used to choose 2048 grid points for training.

Training and Testing

In the training procedure, the two networks (camera pose estimation and SDF prediction) are trained individually, using ground truth camera parameters for the latter . As hyperparameters using Adam optimizer the following values are chosen:

$$m_1 = 4,\ m_2 = 1,\ \delta = 0.001$$
$$\alpha = 0.0001,\ \text{batch size} = 16$$

Convergence takes 50 epochs.

Afterward – for testing – the estimated camera parameters are employed. However, as presented later they also show results with ground truth parameters.

Quantitative Evaluation

The improvements, in utilizing not only an implicit method but also a local feature extraction module, are measured by four commonly used metrics:

1. **Earth Mover's Distance (EMD)** is the minimum amount work that has to be done to match two distribution x and y – in this case prediction and ground truth. Normally, x and y have to be normalized, however, as we are comparing two distributions of equal weights this is no issue. The work itself is calculated using the L2-norm resulting in:

   $$EMD(PC, PC_T) = min_{\phi:PC->PC_T} \sum_{p \in PC} ||p - \phi(p)||_2$$

   For EMD counts, the smaller the better.

2. **Chamfer Distance (CD)** calculates the matching distance to the nearest feature in both ways, from $PCtoPC_T$ as well as the other way round. Here the distance is calculated as the *squared* L2-Norm leading to the following equation:

   $$CD(PC, PC_T) = \sum_{p_1 \in PC} min_{p_2 \in PC_T} ||p_1 - p_2||_2^2$$

   For CD counts, the smaller the better.

3. **Intersection over Union (IoU)** is a ratio, measuring how much overlap is present between two distributions or in this case voxelized meshes. The general formula is given by $IoU = \frac{Intersection}{Union}$. A drawback here is that the authors do not mention how they calculated the values of Intersection and Union. Nonetheless, as the same metric is applied to each method, comparability exists.
   For IoU counts, the higher the better.

4. **F-score** gives a percentage of how much of the object is reconstructed correctly. For its calculation, we need two measures: Precision and Recall.
   The former describes a ratio between all **predicted points** with a distance to the closest ground truth point smaller than a threshold t and all generated points, while the later similarly describes a ratio between all **ground truth points** with a distance to the closest predicted point smaller than t and all ground truth points. Having precision and recall, the F-score is calculated by:

$$\text{F-score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

   Here the principle 'the higher the better' counts true, while a smaller threshold describes a bigger similarity.

EMD, CD & IoU

The results for EMD, CD, and IoU are presented in the following table 1. Each metric has been calculated for each object category. For OccNET[6], however, due to a scale mismatch, only IoU is evaluated which is scale-invariant.

Besides, a version of DISN is evaluated using the ground truth camera parameters denoted '$Ours$' whereas the normal version using predicted camera parameters is denoted as '$Ours_{cam}$'.

		plane	bench	box	car	chair	display	lamp	speaker	rifle	sofa	table	phone	boat	Mean
EMD	AtlasNet	3.39	3.22	3.36	3.72	3.86	3.12	5.29	3.75	3.35	3.14	3.98	3.19	4.39	3.67
	Pixel2mesh	2.98	2.58	3.44	3.43	3.52	2.92	5.15	3.56	3.04	2.70	3.52	2.66	3.94	3.34
	3DN	3.30	2.98	3.21	3.28	4.45	3.91	**3.99**	4.47	2.78	3.31	3.94	2.70	3.92	3.56
	IMNET	2.90	2.80	3.14	2.73	3.01	2.81	5.85	3.80	2.65	2.71	3.39	2.14	2.75	3.13
	3D CNN	3.36	2.90	3.06	**2.52**	3.01	2.85	4.73	**3.35**	2.71	**2.60**	**3.09**	2.10	**2.67**	3.00
	Ours_{cam}	**2.67**	**2.48**	**3.04**	2.67	**2.67**	**2.73**	4.38	3.47	**2.30**	2.62	3.11	**2.06**	2.77	**2.84**
	Ours	2.45	2.41	2.99	2.52	2.62	2.63	4.11	3.37	1.93	2.55	3.07	2.00	2.55	2.71
CD	AtlasNet	**5.98**	6.98	13.76	17.04	13.21	7.18	38.21	15.96	4.59	8.29	18.08	6.35	15.85	13.19
	Pixel2mesh	6.10	**6.20**	12.11	13.45	11.13	**6.39**	31.41	**14.52**	4.51	**6.54**	15.61	6.04	12.66	11.28
	3DN	6.75	7.96	**8.34**	7.09	17.53	8.35	**12.79**	17.28	**3.26**	8.27	14.05	**5.18**	10.20	**9.77**
	IMNET	12.65	15.10	11.39	8.86	11.27	13.77	63.84	21.83	8.73	10.30	17.82	7.06	13.25	16.61
	3D CNN	10.47	10.94	10.40	**5.26**	11.15	11.78	35.97	17.97	6.80	9.76	**13.35**	6.30	**9.80**	12.30
	Ours_{cam}	9.96	8.98	10.19	5.39	**7.71**	10.23	25.76	17.90	5.58	9.16	13.59	6.40	11.91	10.98
	Ours	9.01	8.32	9.98	4.92	7.54	9.58	22.73	16.70	4.36	8.71	13.29	6.21	10.87	10.17
IoU	AtlasNet	39.2	34.2	20.7	22.0	25.7	36.4	21.3	23.2	45.3	27.9	23.3	42.5	28.1	30.0
	Pixel2mesh	51.5	40.7	43.4	50.1	40.2	55.9	29.1	52.3	50.9	60.0	31.2	69.4	40.1	47.3
	3DN	54.3	39.8	49.4	59.4	34.4	47.2	35.4	45.3	57.6	60.7	31.3	71.4	46.4	48.7
	IMNET	55.4	49.5	51.5	74.5	52.2	56.2	29.6	52.6	52.3	64.1	45.0	70.9	56.6	54.6
	3D CNN	50.6	44.3	52.3	**76.9**	52.6	51.5	36.2	58.0	50.5	**67.2**	50.3	70.9	**57.4**	55.3
	OccNet	54.7	45.2	**73.2**	73.1	50.2	47.9	**37.0**	**65.3**	45.8	67.1	**50.6**	70.9	52.1	56.4
	Ours_{cam}	**57.5**	**52.9**	52.3	74.3	**54.3**	**56.4**	34.7	54.9	**59.2**	65.9	47.9	**72.9**	55.9	**57.0**
	Ours	61.7	54.2	53.1	77.0	54.9	57.7	39.7	55.9	68.0	67.1	48.9	73.6	60.2	59.4

Table 1: Quantitative results on ShapeNet Core for the above presented methods. Metrics are CD ($\times 0.001$), EMD ($\times 100$) and IoU(%). CD and EMD are computed on 2048 points. Taken from [1]

The quantitative results show that, on average, DISN is superior using EMD and IoU. In CD it is only beaten by 3DN [4]. Nevertheless, as explained above this method requires further information in the form of a source mesh.

F-score

The F-score results are shown in Table 2. One can see that, apart from a threshold of 20%, DISN again is superior to the other methods. Something to point out here is that, especially for low thresholds, DISN is superior to the other methods by up to 1.5% at a threshold of 0.5% and up to 3.2% at a 1% threshold. As this difference is constantly declining when the threshold reaches higher than 2%, one can see strong indications that especially fine-grained details that correlate with small distances/threshold values are improved, while the overall shape produces similar values.

Threshold(%)	0.5%	1%	2%	5%	10%	20%
3DCNN	0.064	0.295	0.691	0.935	0.984	0.997
IMNet	0.063	0.286	0.673	0.922	0.977	0.995
DISN gt cam	0.079	0.327	0.718	0.943	0.984	0.996
DISN est cam	0.070	0.307	0.700	0.940	0.986	0.998

Table 2: F-score results. Taken from [1]

Camera Pose Estimation results

If camera pose estimation has improved – by using a continuous higher-dimensional parameter representation – is tested by applying two metrics:

1. d_{3D} measures the mean distance between a point cloud transformed with the predicted values and the ground truth point cloud.
2. d_{2D} is the average 2D reprojection error. Generally, such reprojection error is calculated by projecting a 3D point $\hat{X}$ onto the image point using the predicted parameters resulting in the 2D point $\hat{x}$. A reprojection error then equals the euclidean distance $d(x, \hat{x})$ where x is the ground truth projection. Moreover, this reprojection error is measured in pixels.

The results of these metrics for pose estimation are depicted in Table 3. They show that quantitative improvement in the process of pose estimation is measurable compared to the original approach of [7]. More importantly, when analyzing table 1, less difference between the mean results of '$Ours_{cam}$' and '$Ours$' than between '$Ours_{cam}$'. and most other reconstruction approaches can be seen.

	Original	DISN
d_{3D}	0.073	**0.047**
d_{2D}	4.86	**2.95**

Table 3: Quantitative results of camera pose estimation . Taken from [1]

Qualitative Evaluation

As the quantitative results already indicate, qualitative results which are depicted in figure 5 show that DISN fulfills its goal of obtaining fine-grained details. For example, when looking at the first chair sample it is the only method capable of recovering not only the holes in the back part but nearly the exact pattern. This stands in contrast to the other methods where some are capable of adding holes but most only return a dense surface.

Ablation studies

To further, test the effectiveness and robustness of the approach Ablation studies are conducted. Originally, ablation means to surgical remove organs or other human material from the body. In the context of deep learning, the term ablation studies was adopted to name a process where one removes different pieces of a network to gain a better understanding of how the network behaves.

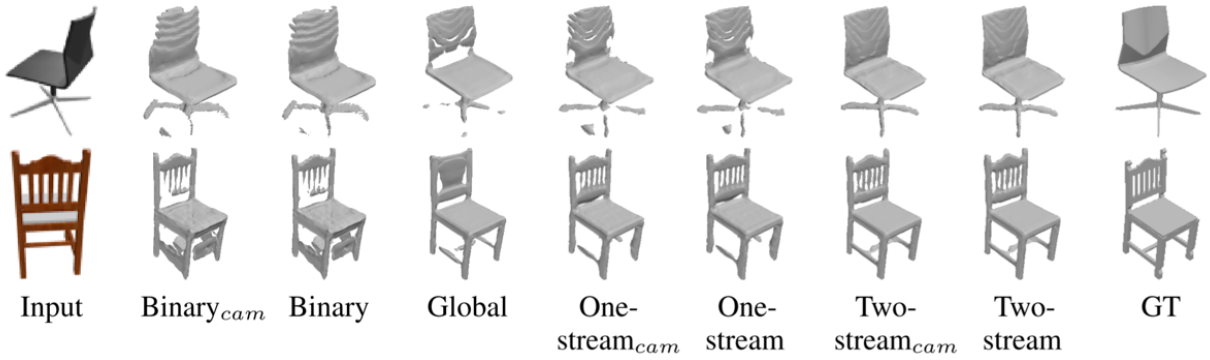

Figure 6: Qualitative results for the category 'chair' when employing the different ablation studies. Taken from [1]

For DISN the authors wanted to find out what impact the camera pose estimation, local feature extraction, and different network architectures have. This impact is evaluated by applying the same qualitative and quantitative measures as above and results are presented in figure 6 for the former and table 4 for the latter.

Camera Pose estimation

As the local feature extraction module is directly dependant on the pose estimation, one wants to know how big this influence is and whether the further improvement is crucial here. Especially, as the evaluation of the pose estimation network (see also table 3) has shown that an average reprojection error of 2.95 pixels is introduced to the network. After conducting the ablation studies one can see (depicted in table 1) that constantly better results can be achieved if true camera parameters are used. However, as already stated this might not be a crucial issue as the difference in metrics between estimated and true values is lower than the difference to most other approaches.

When looking at figure 5 and 6 one can further, conclude that there are small differences but the overall aim is still met.

Binary Classifications

Novel about the here presented approach is also the prediction of concrete SDF values instead of a simple inside/outside classification. However, to further investigate the effectiveness of this different approach the same network structure is employed but

trained with a softmax cross-entropy loss. With this classifier, the output of a point is the probability of being inside or outside the iso-surface.

While quantitative results are slightly worse than the original proposed DISN they are – compared to the other methods – still a lot better using the chair category. Nevertheless, when comparing the concrete visual results in figure 6 quality loss is obvious. Both presented chairs are not completely solid compared to ground truth and the continuous prediction of an SDF.

Removal of Local Feature Extraction Module

The authors of the DISN paper presents their additional local feature extraction module as their key finding for 3D reconstruction. Therefore, to validate its effectiveness its is completely removed from the network and the results in table 4 and figure 6 are labeled as Global. Surprisingly, while IoU is nearly equal, EMD and CD are better compared to the Binary studies – this is unexpected as this implies a stronger impact of the classifier change than the one of adding local feature extraction. Notwithstanding, the qualitative results of the chair lose a lot of detail and seem to be more similar to IMNET or OCCNET than to DISN.

Different Network structures

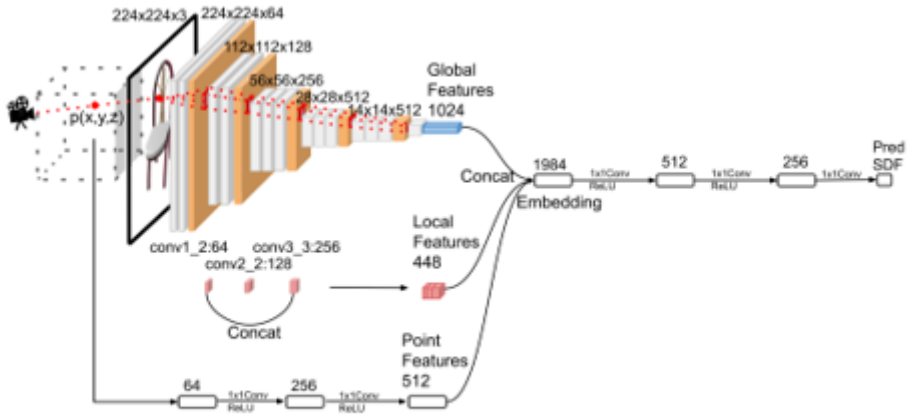

Figure III: DISN One-stream Network.

Figure 7: The network architecture "One-stream" employed for ablation studies. Taken from [1]

As their last ablation study, the authors create a second SDF prediction network called One-stream (see also figure 7). Different from the actual proposed one, only one decoder is used and both global and local features are concatenated. The only slightly inferior quantitative as well as qualitative results of one-stream vs two-stream (which is the proposed approach) show that DISN can be implemented by different network

structures.

Camera Pose	Binary ground truth | estimated	Global n/a	One-stream ground truth | estimated	Two-stream ground truth | estimated
EMD	2.88 | 2.99	2.75 | n/a	2.71 | 2.74	**2.62 | 2.65**
CD	8.27 | 8.80	7.64 | n/a	7.86 | 8.30	**7.55 | 7.63**
IoU	54.9 | 53.5	54.8 | n/a	53.6 | 53.5	**55.3 | 53.9**

Table 4: Quantitative results for the category 'chair' when employing the different ablation studies. Taken from [1]

Additional Work

To further, show the capabilities of DISN three applications are employed and briefly depicted:

1. **Shape interpolation** – generates plausible shapes between two different key-objects, using interpolation. Figure 8 shows that when interpolation is applied to both global and local features, a gradual transformation is possible.

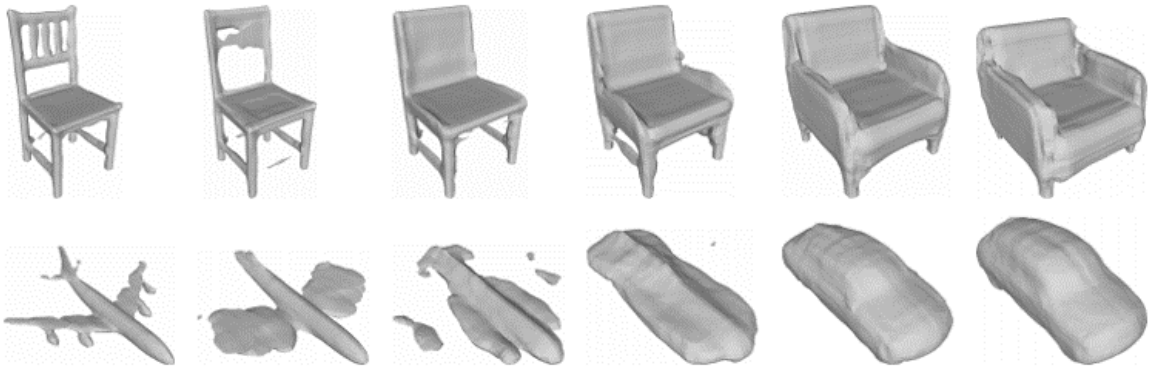

Figure 8: Shape interpolation results. Taken from [1]

2. **Online product images as input** – are possible candidates for a real application. As the Network is trained on rendered images, this experiment gives a first clue on how applicable DISN is for other domains. Despite that the reconstruction results (see also figure 9) do not look as nice as the ones of the official test set, they still return plausible predictions.

Figure 9: Results on single-view 3D reconstruction with online product images. Top are the inputs and bottom the rendered outputs. Taken from [1]

3. **Multi-view reconstruction** – makes use of more information, in form of multiple view inputs, to improve the reconstruction process. For this, the global and local features are encoded for each image separately and then concatenated in the corresponding embedding vector. After applying a max-pooling operation on this embedding vector it is fed two the decoder. As figure 10 shows training such an extended architecture with two additional views can result in better predictions.

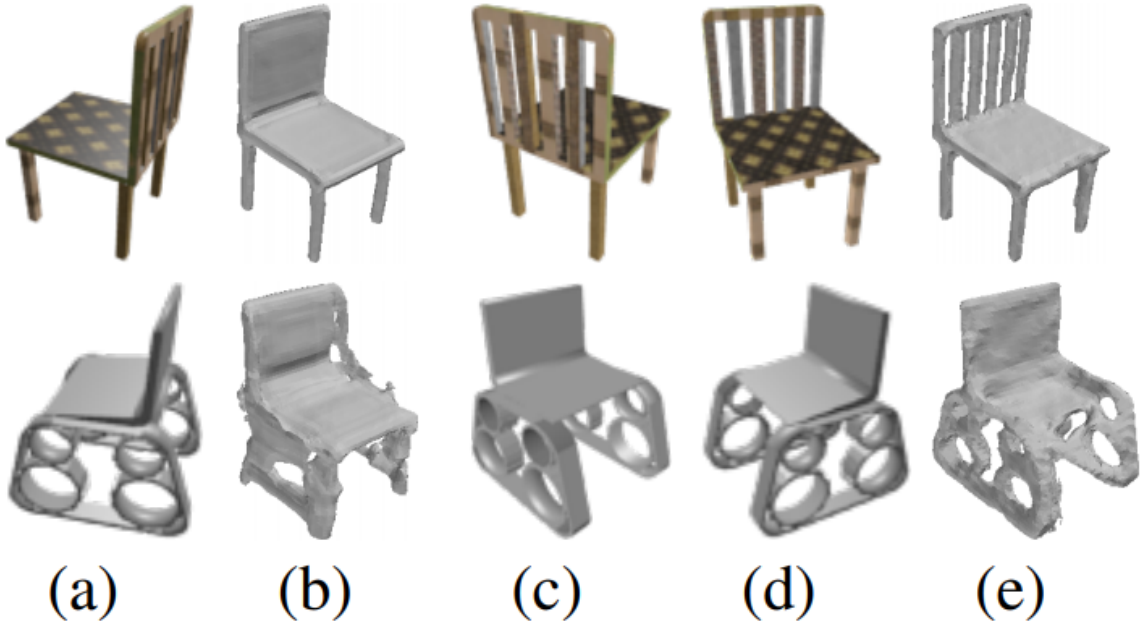

Figure 10: Multi-view reconstruction results. (b) shows the result from single-view input (a), while (e) is predicted using (a), (c) and (d)… Taken from [1]

Conclusion

In this blog post, current challenges in single-view 3D reconstruction were discussed and a novel approach to also recover fine-grained details presented. Further, the advantages of implicit 3D surface representation methods were highlighted and the feasibility of directly predicting a Signed-Distance-Function (SDF) tested.

Below, first, the concluding thoughts of the authors of DISN are presented followed by my perception of the presented work.

Author's conclusion

In their conclusion Wang et al. [1] state that their work provides two main contributions:

1. Using a local feature extraction module, it is possible to recover fine-grained details in single-view 3D reconstruction. While a lot of work delivered good results on recovering overall shape, to the best of the author's knowledge DISN is the first work of such capability.
2. Not only are implicit methods feasible, but also in most cases superior to explicit methods due to their flexibility to generate topology-variant 3D meshes. This has been proven using qualitative and quantitative evaluation. Further, their approach of predicting continuous SDF values outperforms current work using only a binary classifier. In addition to comparison, this has also been tested using ablation studies.

However, as the networks are only trained on rendered images they can only take images with a clear background as input. To tackle this limitation future work should include texture prediction using a differentiable renderer as proposed by [13].

My perspective

My overall opinion of DISN is very positive. The extensive evaluation seems to prove the conclusion of the paper's authors. They have not only compared their work equally to several highly rated other papers but also employed the settings of the test set up as similar as possible. Further, they conducted extensive ablation studies to prove the effectiveness of their newly introduced modules. However, there are four things to criticize:

1. The quantitative metrics have not been explained in detail, IoU not at all. It would be better if they at least reference them to accepted scientific work, to assure that the measures are not manipulated for their work benefit.
2. As evaluated above in F-score DISN provides the best results. However, the quality of the comparisons suffers from the fact, that other than to DISN F-score was only applied to 3DCNN and IMNET.
3. They have added several additional applications, creating a perception of how awesome their work ist. However, no concrete implementation details have been published nor extensive evaluation. Therefore, in my opinion, this should be more seen as a possible outlook for the future than a proposal of what it is capable of doing.
4. OccNet seems to provide the next best comparable research for single-view reconstruction. However, they did only compare IoU which is nearly equal and found no other quantitative metric. Further, in the supplementary extensive qualitative results are presented – considering all other methods except OccNet.

References

[1] Weiyue Wang, Qiangeng Xu, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. DISN: Deep Implicit Surface Network for High-quality Single-view 3D Reconstruction. In NeurIPS, 2019

[2] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In CVPR, 2018.

[3] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. arXiv preprint arXiv:1804.01654, 2018.

[4] Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 3dn: 3d deformation network. In CVPR, 2019.

[5] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. arXiv preprint arXiv:1812.02822, 2018.

[6] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In CVPR, 2019.

[7] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In NeurIPS, 2018.

[8] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. arXiv preprint arXiv:1812.07035, 2018.

[9] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. arxiv, 2015.

[10] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In ECCV, 2016.

[11] Hongyi Xu and Jernej Barbic. Signed distance fields for polygon soup meshes. In ˇ Proceedings of Graphics Interface 2014, pages 35–41. Canadian Information Processing Society, 2014.

[12] Fun Shing Sin, Daniel Schroeder, and Jernej Barbic. Vega: non-linear fem deformable object ˇ simulator. In Computer Graphics Forum, volume 32, pages 36–48. Wiley Online Library, 2013.

[13] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In CVPR, 2018.