3D Reconstruction from a Single RGB Image

Moustafa Elsharkawy
TUM
moustafa.elsharkawy@tum.de

Alexander Sheldrick
TUM
alexander.sheldrick@tum.de

Abstract

In this work, we show first quantitative and qualitative results proving the viability of IF-Nets for reconstruction of large, complex scenes. Furthermore, we extend IF-Nets with a depth regressor and differential voxelization to enable an end-to-end trainable 3D scene prediction network. The aim is combine their ability to complete partial, incomplete inputs to watertight meshes, while retaining local and global details, with the ability to generate those partial inputs autonomously from RGB images.

1. Introduction

Efforts involving 3D scene prediction date back to the 60s [7], but the subject remains a difficult and unfinished task. Indeed, "the inherent ambiguity in depth perception, the clutter and complexity of real-world environments make it still challenging to fully recover the scene context (both semantics and geometry) merely from a single image" [6]. Especially in recent years, the topic has received enormous interest from researchers in both computer vision and graphics communities, with many different approaches being explored simultaneously. Currently, implicit representations of 3D geometry and material properties have achieved state of the art results on staple data-sets such as Shapenet [1]. These methods are of special interest, as they promise to solve memory and resolution issues of discrete occupancy methods [2].

In this work, we are focusing on reconstructing scenes from a single image. We make the use of IF-Net [2] which focuses on shape completion from an incomplete 3D input. We explore the usage of IF-Net in the task of 3D reconstruction from images. We also explore its ability to work on complex scenes instead of simple shapes as was proposed in the paper.

2. Method

An overview of the pipeline can be seen in Fig. 1.
The task is to reconstruct 3D scenes from an rgb image. Our pipeline is a two-part architecture. The first part is a UNet [8] that learns to predicts a depth map from an rgb image input. The predicted depth map is voxelized into an incomplete occupancy grid. This grid is then fed into the second part, IF-Net [2], which learns to complete incomplete data using extra supervision as points sampled from the ground truth mesh along with their occupancy. We make use of differentiable voxelization proposed by [4] to train the pipeline end-to-end.

2.1. Depth regressor

Inspired by the work of [9], we use a UNet [8] as the depth estimator. The UNet has 8 downsampling and upsampling layers. UNet feature concatenation allows us to get a more precise depth estimation.

The feature concatenation in the upsamling layers puts a restriction on the input size. To use the full 8 layers of the UNet, we had to resize the rgb image input. First, the image is square padded with zeros and then resized to a size of 256×256. After the UNet predicts the depth, we resize it back to the original image size and discard the padding. Resizing it back is important as the camera intrinsic is tied to the image size.

Following this, a sigmoid layer and a renormalization step are applied so that the predicted depth values lies within the minimum and maximum values of our dataset.

2.2. Projection into 3D

Using the depth map and camera intrinsics, we project the 2D image points into 3D. Next, we need to discretize the 3D space into a voxelized occupancy grid. This step can be viewed as a step function where the occupancy of each voxel is determined by whether a point lies inside of it or not. As this process is not differentiable, we needed an alternative method to allow the gradient to flow back to the UNet and to train the full pipeline end-to-end.

We made use of the method in [4] to make this process differentiable. Instead of voxelizing points via their discrete position, we smooth their density over a region with a Gaussian distribution. The resulting voxelization is then the clipped sum of smoothed points evaluated at the voxel

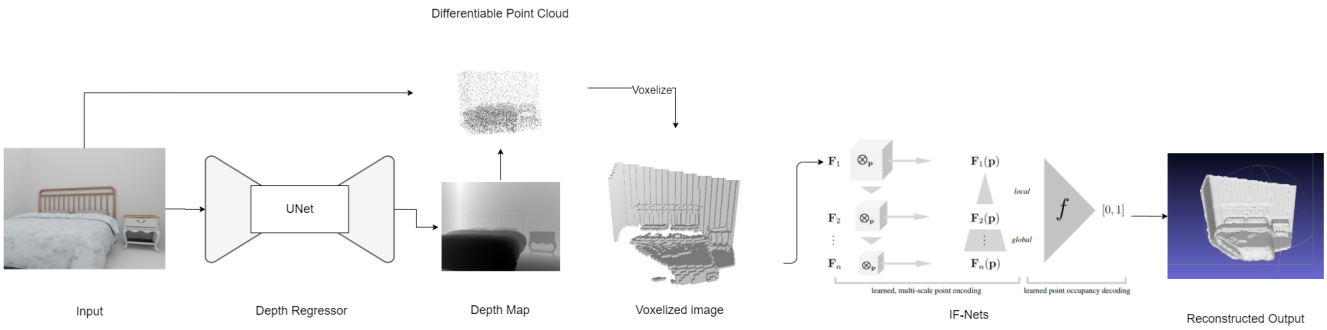

Figure 1. Overview of the full pipeline

grid points.

Assume the point cloud is a set of N points $P = \{\mathbf{x}_i\}_{i=1}^{N}$, each including the point position $\mathbf{x}_i = (x_i, y_i, z_i)$. To allow the gradient to flow, we apply Gaussian densities f_i to each point $\mathbf{x}_i$. The occupancy function of a point in 3D space is a clipped sum of the individual per-point functions:

$$o(\mathbf{x}) = clip(\sum_{i=1}^{N} f_i(\mathbf{x}), [0,1]), \tag{1}$$

$$f_i(\mathbf{x}) = c_i \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x_i^{'}})^{\mathbf{T}} \mathbf{\Sigma_i^{-1}} (\mathbf{x} - \mathbf{x_i^{'}})\right)$$

Where $\Sigma_i = diag(\sigma_1^2, \sigma_2^2, \sigma_3^2) \in \mathbb{R}^{3\times3}$ is a diagonal covariance matrix of the Gaussian and c_i defines the scale of each one. We use the occupancy function $o(\mathbf{x})$ to obtain the discretized grid $\mathbf{X} \in \mathbb{R}^{N\times N\times N}$, where $N \in \mathbb{N}$ denotes the required grid resolution. The grid values are now in the range $[0, 1]$ instead of 0's and 1's.

As for the implementation details, first we apply trilinear interpolation on all points to place them on a grid. Then, we a apply 3D convolution over the grid with Gaussian kernels. The 3D convolutions are in the form of three 1D convolutions along each axis for improved efficiency. The restriction of this implementation is that the covariance Σ_i is fixed for all functions f_i. We also implemented the covariance parameters $\{\sigma_i^2\}_{i=1}^{3}$ to be learnable as it was shown in [4] to perform better than hand-tuned.

2.3. Implicit functions

We use the IF-Net architecture proposed by [2]. The goal of this network is to learn a function that, given the voxelized grid $\mathbf{X}$ and a point $\mathbf{p}$, determines the occupancy of the point.

Shape Encoding: We apply subsequent 3D convolutions on the voxelized grid each of which is followed by a down scaling layer. We concatenate the features at each level to obtain a set of features at multiple scales. These features encode the global and local structure of the scene. The encoder is denoted as

$$g(\mathbf{X}) := \mathbf{F}_1, .., \mathbf{F}_n, \tag{2}$$

where the feature grid at stage k $\mathbf{F}_k \in \mathcal{F}_k^{K\times K\times K}$, of decreasing resolution $K = \frac{N}{2^{k-1}}$, and variable channel dimensionality $F_k \in \mathbb{N}$ at each stage $\mathcal{F}_k \subset \mathbb{R}^{F_k}$.

Shape Decoding: We extract the features at the location of the query point $\mathbf{p}$ and at its neighborhood defined by a distance d along the Cartesian axes:

$$\{\mathbf{p} + a.\mathbf{e}_i.d \in \mathbb{R}^3 | a \in \{1, 0, -1\}, i \in \{1, 2, 3\}\}, \tag{3}$$

where $d \in \mathbb{R}$ is a hyper-parameter that defines the distance to $\mathbf{p}$ and $\mathbf{e} \in \mathbb{R}^3$ is the $i-th$ Cartesian axis unit vector.

These extracted encodings of the point $\mathbf{p}$ $\mathbf{F}_1(\mathbf{p}), ..., \mathbf{F}_n(\mathbf{p})$ are fed into a fully connected network $f(.)$ to determine the occupancy of the point:

$$f(\mathbf{F}_1(\mathbf{p}), ..., \mathbf{F}_n(\mathbf{p})) : \mathcal{F}_1 \times ... \times \mathcal{F}_n \mapsto [0, 1] \tag{4}$$

2.4. Method Training

During training, If-Net is fed points from two sources: **Ground truth mesh sampling:** As a preprocessing step, transform the ground truth distance field into a mesh using the marching cubes algorithm. We make this mesh watertight meaning that it's a continuous surface devoid of holes. Then, we sample points from the mesh with a predefined variance and determine the occupancy of each. These points serve as extra supervision to allow the IF-Net to train better.

Point cloud: For each point in the point cloud P obtained from the projection layer, we compute the occupancy using the ground truth mesh. Using all the points in the point cloud P is redundant and greatly increased our computation and memory usage. Hence, we sample a subset of these points randomly for supervision.

2.5. Method Inference

During testing, the IF-Net is evaluated on points on a grid of the desired resolution. We can obtain high resolution results as we can query the IF-Net at an arbitrary resolution. Then, using the marching cubes algorithm, the resulting occupancy grid can be transformed into a mesh.

3. Experiments

So far, IF-Net has shown great results for shape completion of single objects and people. This work explores it's potential for single view 3D-scene reconstruction of complex indoor scenes. We prove 1) its feasibility of completing large scenes with a GT depth map, that is being differentially voxelized to create the necessary inputs, and 2) results, merits and limitations of generating these depth maps online via depth regressor.

3.1. Experimental setup

Data: Our data-set, 3D Front [3], consists of synthetic indoor scenes, rendered into RGB images in blender. This provides us with ground truth depthmaps, ground truth distance fields and camera intrinsic. As a pre-processing step, we extract singular GT meshes of the scenes from the distance fields via the marching cubes algorithm. Next, these meshes are made watertight using the procedure from IF-Net and OCC-Net [5]. Next, we sample 3D points in space at a normally distributed distance from the surface of these meshes, and since these meshes are watertight we calculate their corresponding occupancy values $[0, 1]$. In the feasibility experiments, we reduce the original input voxelgrid (z = 139, y = 104, x = 112) to half scale (70, 52, 56) before meshing to more quickly explore hyper-parameter configurations and resulting mesh quality. After having performed an ablation study for the half scale inputs, we evaluate the results of our fully end-to-end trained network on full scale. The half scale data-set consists of 703 training samples, 118 validation and 118 test samples. The full scale studies train on 2753 samples and evaluate on 291 val and 291 test samples.

Metrics: We follow the lead of the authors of IF-Net to measure the reconstruction quality quantitatively and evaluate the meshes with these three established metrics: *volumetric intersection over union* (IoU) measuring the spatial overlap of the volumes (higher is better), *Chamfer-L_2 distance*, to measure the accuracy and completeness of the surface (lower is better), and *normal consistency* measuring the accuracy and completeness of the shape normals (higher is better).

Baseline: The main goal is to investigate feasibility of the proposed method. We are not aware of any other works for 3D Scene reconstruction using 3D Front [3], hence we will compare the proposed metrics with the original results of IF-Net on the ShapeNet [1] data-set to quantitatively evaluate the mesh quality of the *viability*, *pretrained-UNet* and *end-to-end* experiments.

3.2. Configuration

Hyperparameters: The investigated hyper-parameters like batch-size, learning rate and smoothing kernel-size had a minor effect on mesh quality when reasonably configured, we therefore decided on a learning rate of 2e-4, the largest batch-size that fits the gpu and a Kernel-size of 3.

Smoothing covariance and kernel size: Interestingly, using an isotropic covariance $\sigma^2 I$, the network learned σ converged to just above 0.5 regardless of kernel-size, which we interpret as the uncertainty of point positions of the order of half a voxel in size.

Bit precision: We also investigated the effects of mixed and full precision training. We noted that full precision training converged more quickly to lower validation losses. Due to a known PyTorch Bug, 'torch.nn.functional.grid_sample' (which IF-Net uses frequently) runs roughly 10x faster in 32bit than in 16bit, which led us to abandon mixed precision training for the full training.

Training speed: Due to limited resources and computation power, the training was very slow. To speed up the training without compromising the results, we implemented scaling of the occupancy grid that is fed as input to the IF-Net. Additionally, the meshing and point sampling step needed to be run over the data for each different scale. Second, we only used a subset of the point cloud P which greatly reduced our memory usage. With these modifications, we were able to speed up the training by 10 times which allowed us to experiment faster for our results.

3.3. Results

The task of single view 3D scene reconstruction is especially challenging as we aim to complete partial inputs dimensions to a full 3D mesh. These partial inputs are autonomously generated via the projection of a predicted depth map into object space via camera intrinsic. In addition, these partial inputs are the approximate surface points of complex surfaces, with added uncertainty from the depth regressor. From this projected partial one-sided point cloud input the network will have to infer complete 3D objects with appropriate volumes.

As inaccuracies in the depth prediction will severely hamper inference accuracy, we first prove feasibility by training on the differentially voxelized GT depth map as a first benchmark for inference. This feasibility test also serves as a first exploration of suitable hyper-parameters for the training of the full network later. For these experiments we investigate the problem on a resized data set $(z, y, x) = (70, 52, 56)$, where z denotes the viewing direc-

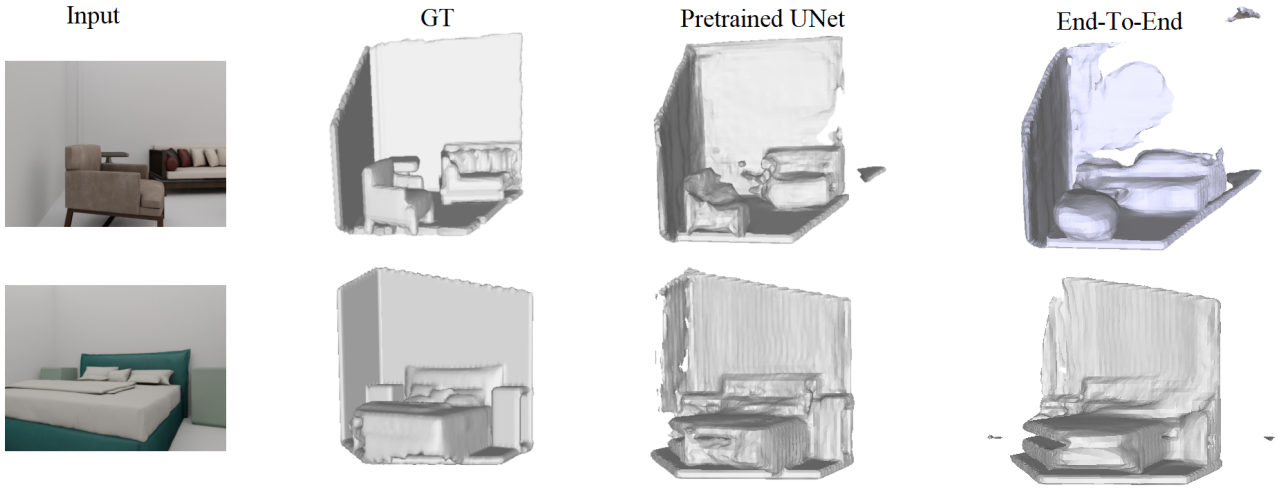

Figure 2. Exemplary inference results of previously unseen images. Depth regressor + differential voxelization create inputs for IF-Net. Pretraining the depth regressor helps faster convergence. End-to-End is trained without depth supervision

tion. The network testing feasibility uses resized data and is trained 703 training samples. We qualitatively and quantitatively evaluate the quality of predicted scene meshes on singular views of the scene with an RGB image. Quantitative results for voxelization from GT + IF-Net inference, Pretrained Unet and End-to-End trained network are shown in Table 1.

Note that testing still occurs on previously unseen data, and that local and global details are inferred by the IF-Net. From these first tests, Fig. 2, we determined that fine details like chair and table geometry can in principle be inferred by this architecture, although the quality clearly degrades with the inaccuracy of depth prediction. Note here that the MSE for the per pixel depth estimation was roughly 1 voxel on average (5 cm) for the pre-trained network, and that the end-to-end trained network was trained without depth supervision. We conclude that the latter has meaningless depth-loss, as the network did not learn a pixel-wise depth value but rather an occupancy encoding. The pre-trained depth regressor is supervised with ground truth depth data during training, to improve the accuracy of depth prediction on unseen data and to accelerate convergence. However, as shown, the network can be trained end-to-end without depth supervision with sufficient accuracy, although the trade-off seems to be speed of convergence and loss of details.

We continue to train models with and without explicit depth supervision and will investigate per-dimension variable smoothing kernel-size next. These first results indicate that the main deficit of the here presented single view scene-reconstruction network are inaccuracies in the generation of the partial inputs (the voxilization of projected point-cloud), and that occlusions and out of bounds predictions hurt their quantitative performance, as frontal views of simple scenes seem to yield the best results during evaluation.

	IoU ↑	Chamfer -L_2 ↓	Normal -Consistency ↑
IF-Net**	**0.73**	**0.00002**	**0.91**
Viability*	0.48	0.0007	0.82
Pretrained UNet	0.43	0.0065	0.82
End-to-End***	0.40	0.0094	0.77

Table 1. Results for different network sizes and depth accuracy. *Viability uses resized inputs and meshes. **IF-Net self-reported evaluation on ShapeNet on similar gridsize. End-to-End was trained without pre-training or depth supervision. Viablity Pretrained with depth-supervision

4. Discussion and Conclusion

In this work we successfully applied IF-Nets for the task of single view 3D reconstruction. Their ability to infer coherent local and global details from partial, one sided inputs, enables the joint depth and mesh prediction from a single, low resolution RGB image input. We present first quantitative and qualitative results and strongly believe, that these, due to time and computational resource limits, not fully optimized experiments, are motivation for further research. We will continue to investigate the quantitative performance of models with and without depth supervision and different hyper-parameter settings. First experiments indicate, that the uncertainty of the network learned σ can have different values for different dimensions, and that these are especially relevant for unsupervised training. It seems furthermore, the depth uncertainty is greater than the other dimensions and we will therefore move to perform ablation studies for varying smoothing coefficients and kernel sizes for multiple dimensions to capture this additional point uncertainty.

References

[1] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing 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. *CoRR*, abs/1512.03012, 2015. 1, 3

[2] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion, 2020. 1, 2

[3] Huan Fu, Bowen Cai, Lin Gao, Lingxiao Zhang, Cao Li, Zengqi Xun, Chengyue Sun, Yiyun Fei, Yu Zheng, Ying Li, Yi Liu, Peng Liu, Lin Ma, Le Weng, Xiaohang Hu, Xin Ma, Qian Qian, Rongfei Jia, Binqiang Zhao, and Hao Zhang. 3d-front: 3d furnished rooms with layouts and semantics, 2020. 3

[4] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. *CoRR*, abs/1810.09381, 2018. 1, 2

[5] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[6] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image, 2020. 1

[7] Lawrence Roberts. *Machine Perception of Three-Dimensional Solids*. 01 1963. 1

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 1

[9] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. *CoRR*, abs/1912.08804, 2019. 1