3D Semantic Scene Reconstruction from Images

Ananth Kalyanasundaram
Technical University of Munich
ananth.kalyanasundaram@tum.de

Mreenav Shyam Deka
Technical University of Munich
mreenav.deka@tum.de

Abstract

3D semantic scene understanding is a challenging problem as one needs to focus on both scene understanding and object reconstruction. Given a single RGB image as input, the task would be to predict the room layout, reconstruct the 3D meshes of objects and predict semantics for the same. Consequently, combining these tasks in the 3D space can become computationally expensive very fast. We propose a novel non-learnable Shearing Layer, that can convert 2D feature maps to 3D feature maps in a computationally efficient manner, without the need for backprojection using depth information. Using Mesh R-CNN [4] as a baseline, we show that replacing the voxel head with our sheared decoder yields smoother and more accurate meshes.

1. Introduction

Over the last decade, we have seen big jumps in performance of neural networks in several 2D vision tasks such as object detection using bounding boxes, semantic and instance segmentation using masks. However the real world is three dimensional (3D) in nature and would need scene understanding to accurately represent the objects in this space.

We have seen works that have shown reasonable success in solving this problem. [3, 4, 13]. These papers propose networks that take RGB images as input, learn and extract important features using 2D convolutional neural networks [7, 9, 10, 14], backproject the images into voxel space and perform mesh refinement. The Total 3D Understanding paper [13] goes a step further to reconstruct the room layout, camera pose with bounding boxes, poses and meshes. Similarly, Dahnert et al. [3] proposed a method to unify the tasks of geometric reconstruction, 3D semantic segmentation, and 3D instance segmentation.

However, a crucial problem that could be noted from these papers [3, 13] is the computational complexity of computing the inverse of the intrinsics matrix on the fly to backproject the 2D images into voxel space. We note that this operation significantly increases the training and inference time.

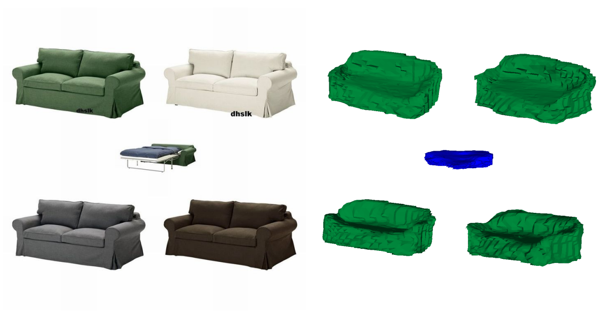

Figure 1. Left: Input Image, Right: Our predicted 3D mesh with semantics

Gkioxari et al. [4] uses a 2D convolutional layer where the number of feature maps are set to the number of rows thereby creating a single 3D map. However 2D convolutional layers may not be great at learning 3D feature maps due to the complex nature of the same. We note that a 3D decoder similar to the ones used in [3] are more accurate in reconstructing the objects better.

To alleviate these problems to a certain extent, we propose the Shearing layer. Our contributions can be summarized as follows : -

- We propose a novel non-learnable Shearing Layer which converts 2D feature maps directly into 3D feature maps.
- We show that our "sheared" decoder can learn to reconstruct meshes which are smoother and more accurate compared to the 2D decoder used by Gkioxari et al. [4]
- We also extend the MeshRCNN [4] codebase to allow for semantic reconstruction of meshes (as seen in 1) rather than just class agnostic ones.

2. Baseline - MeshRCNN

2.1. Voxel Branch

MeshRCNN [4] extends the architecture of MaskRCNN [6] by adding a voxel branch. The voxel branch consists of a decoder which uses 2D convolutional and transposed convolutional layers. The feature maps obtained from the ROIAlign layer of the MaskRCNN is taken as input for the decoder. The output of the layer is a single 3D feature map.

A process called cubify is used to replace each occupied voxel from the voxel occupancy grid into a triangle mesh using a threshold, in an efficient manner. The output is a watertight mesh whose topology depends on the voxel predictions.

2.2. Voxel Loss

Binary cross entropy loss between the predicted voxel occupancies and the ground truth voxel occupancies is minimized by training the voxel branch.

2.3. Mesh Refinement

The coarse mesh obtained from the voxel branch is further refined through a series of mesh refinement stages. First, the vertex alignment operation projects each vertex of the coarse mesh onto the image plane, which generates an initial image aligned feature vector. A series of graph convolutions [8] operating over the edges then adjust the vertex positions of the initial mesh. The vertex refinement stage predicts the offsets for each of the vertices which updates the mesh geometry while keeping the topology fixed. The vertex refinement output is further refined then by the next mesh refinement stages.

2.4. Mesh Losses

MeshRCNN samples point-clouds P^{gt} and P^{i} from the ground truth mesh and the generated mesh respectively and defines the mesh loss of the i-th mesh refinement stage as the weighted average of the chamfer loss $\mathcal{L}_{cham}(P^i, P^{gt})$, the (absolute) normal loss $\mathcal{L}_{norm}(P^i, P^{gt})$, and the edge loss $\mathcal{L}_{edge}(V^i, E^i)$.

Chamfer distance:

$$\begin{aligned}\mathcal{L}_{cham}(P,Q) = |P|^{-1}\Sigma_{(p,q)\in\Lambda_{P,Q}}||p-q||^2 \\ +|Q|^{-1}\Sigma_{(q,p)\in\Lambda_{Q,P}}||q-p||^2\end{aligned} \quad (1)$$

(Absolute) Normal Distance:

$$\begin{aligned}\mathcal{L}_{norm}(P,Q) = -|P|^{-1}\Sigma_{(p,q)\in\Lambda_{P,Q}}|u_p \cdot u_q| \\ -|Q|^{-1}\Sigma_{(q,p)\in\Lambda_{Q,P}}|u_q \cdot u_p|\end{aligned} \quad (2)$$

Edge Loss:

$$\mathcal{L}_{edge}(V,E) = \frac{1}{|E|}\Sigma_{(v,v')\in E}||v-v'||^2 \quad (3)$$

where $E \subseteq V \times V$ are the edges of the predicted mesh.

The mesh refinement branch is trained to minimise the mean of the chamfer loss, the normal loss, and the edge loss between the ground truth mesh and the predicted mesh.

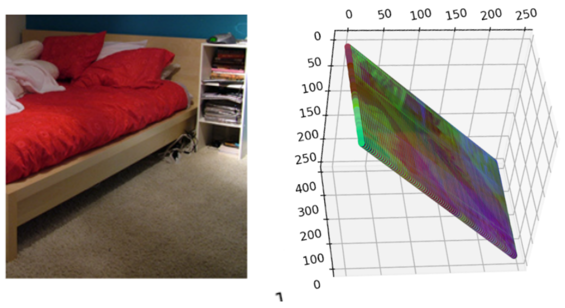

Figure 2. Left: Input Image, Right: 3D map of Sheared image

3. Our Method

3.1. Shearing Layer

We propose to create a non-learnable layer that would push each row of the previous layers feature maps into a separate channel while blacking out other pixels of the same channel, such that each row of the feature map is in a different channel. The output of the shearing layer is a diagonalized image in the 3D space (as seen in 2) .This would then be used as the input for the 3D convolution [16].

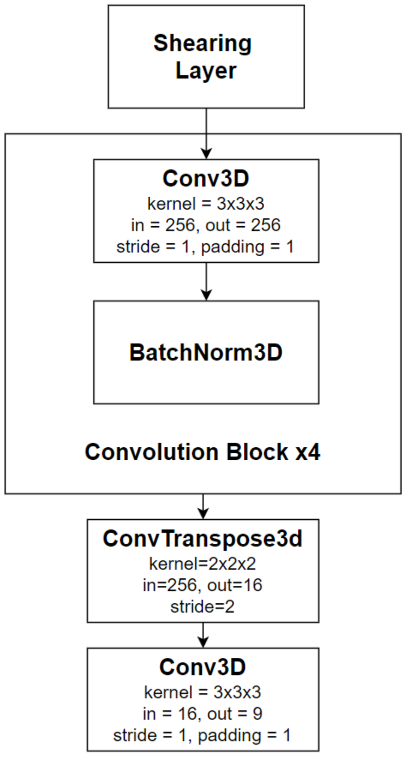

Figure 3. Architecture of the Sheared Decoder

3.2. Sheared Decoder

The sheared decoder consists of the Shearing layer followed by the 3D analogous of the 2D decoder used in the baseline as seen in 3. The final layer of the decoder is a 3D convolutional layer with number of classes as channels, thereby predicting a voxel occupancy grid for each detection and for each class. The sheared decoder is used to replace the voxel branch of the MeshRCNN. The sheared decoder is trained to minimize the same loss as proposed in the voxel branch of MeshRCNN.

4. Experiments

4.1. Training

We trained the baseline MeshRCNN and our proposed network on the Pix3D [15] dataset which consists of 10,069 images and 395 unique 3D models. We randomly split the data into a training set that consists of 7539 images and a test set consisting of 2530 images.

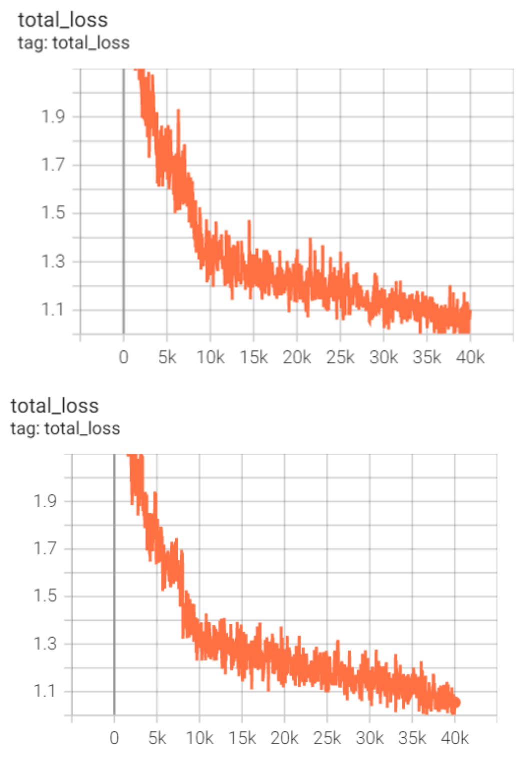

Figure 4. Top: Training loss curve for MeshRCNN, Bottom: Training loss curve for our network

The two networks were trained on a single NVIDIA RTX 3090 GPU. We used stochastic gradient descent with momentum as the optimizer for this problem. After linearly increasing the learning rate from 4×10^{-4} to 4×10^{-3} over the first 1000 iterations, we then decay it by a factor of 10 at 8000 and 36,000 iterations. The model is trained for a total of 40,000 iterations. Due to the memory limitations of a single GPU, we were restricted to a batch size of 4. The training curves for the baseline MeshRCNN and our proposed method can be seen in 4. We note that the gradients were more stable during the training of our network when compared to the baseline.

Table 1. Comparison of average AP^{box}, AP^{mask}, and AP^{mesh}

Architecture	AP^{box}	AP^{mask}	AP^{mesh}
MeshRCNN	**0.938**	**0.881**	0.420
MeshRCNN + Shear (Ours)	0.935	0.875	**0.455**

Table 2. Comparison of per category AP^{box}, AP^{mask}, and AP^{mesh}

Category	MeshRCNN	Ours
Bed	**0.449**	0.383
Bookcase	0.560	**0.607**
Chair	0.380	**0.429**
Desk	0.368	**0.378**
Misc	**0.1625**	0.136
Sofa	**0.716**	0.676
Table	0.546	**0.551**
Tool	0.116	**0.306**
Wardrobe	0.484	**0.621**

4.2. Evaluation

We reuse the metrics that are used for evaluation in the MeshRCNN [4] paper viz. AP^{box}, AP^{mask}, and AP^{mesh}. The first two are standard metrics used for object detection and segmentation challenges [11] at intersection-over-union (IoU) 0.5 respectively. AP^{mesh} is defined as the mean area under the precision-recall curves for each category at $F1^{0.3} > 0.5$. As Pix3D is not exhaustively annotated, the model may be penalised for correct predictions that correspond to non-annotated objects. To prevent this, AP^{mesh} is only calculated for predictions with box IoU > 0.3. The average AP^{box}, AP^{mask}, and AP^{mesh} are given in Table 1 while the per-category comparison can be seen in Table 2.

We note that the AP^{mesh} of MeshRCNN with the Shearing Layer is higher than that of the baseline MeshRCNN. The proposed method also outperforms the baseline MeshRCNN in most of the per-category AP^{mesh}, some by a significant margin.

4.3. Qualitative Comparison

Fig 5 shows us the comparison of the predictions made by both MeshRCNN and our network, given an input image. It can be clearly seen that the meshes generated by our network are smoother and more accurate. We can also see from Fig 5 that our model is able to discriminate bet-

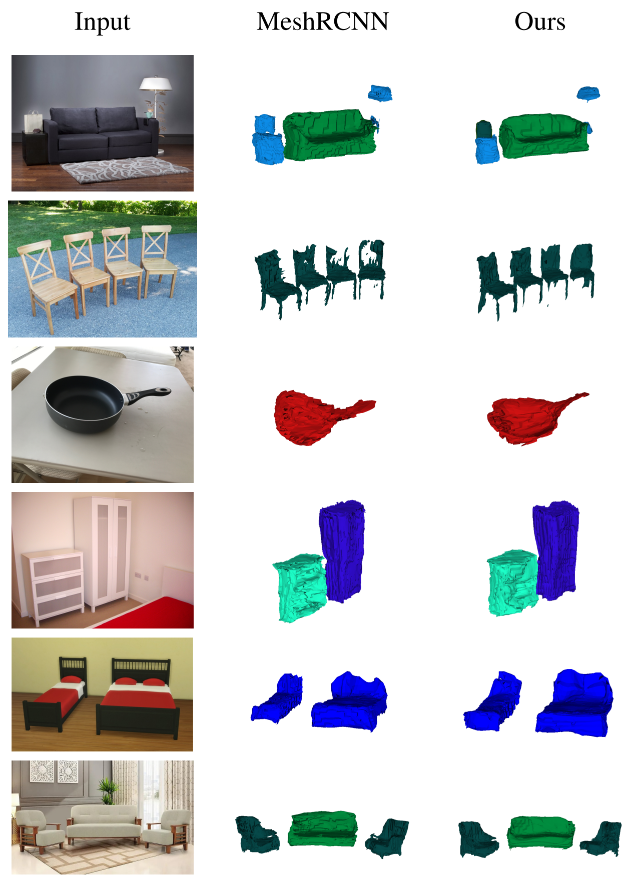

Figure 5. Qualitative comparison of reconstructed scenes by MeshRCNN and our method. From top to bottom: Sofa Scene, Chairs, Pan, Wardrobe and Bookcase, Beds, Previously Unseen Image [1]

ter between the semantics of different objects placed on top of each other compared to MeshRCNN. Our model is also able to yield a better geometry for difficult objects (such as Tools) in overall and more complete meshes. We can see that the overall AP^{mesh} score of the predictions is better compared to the baseline as seen from Table 1.

From Table 2, we observe that the baseline architecture beats our model in three classes by a small margin in AP^{mesh}. However our model yields a significantly higher AP^{mesh} in rest of the classes, sometimes almost thrice the AP^{mesh} of baseline metrics. This demonstrates the model's superior generalization capability compared to baseline MeshRCNN.

5. Limitations

As MeshRCNN does not use backprojection to project the 2D image to the 3D domain, we weren't able to experimentally demonstrate the computational efficiency of using the shearing layer. From preliminary testing, we see a 35x performance boost when we replace backprojection operation with the shearing layer. We also note that MeshRCNN does not predict the actual depth location of the objects centre. This creates scenes which may be inaccurate in the location of the objects with respect to each other leading to cluttered reconstructions.

6. Conclusion and Future Work

We observe that our model is able to learn more discriminative feature maps in the voxel space. This can be attributed to the sheared decoder, which uses 3D convolutions [16] as compared to the 2D convolutions used in the baseline. The shearing layer allows the voxel head to learn higher dimensional features that can reconstruct the scene in a more realistic sense.

Through the usage of the shearing layer, we are able to use learned 2D features maps in a 3D space. Therefore, information can be propagated from a 2D encoder into a 3D decoder and hence one could use sheared feature maps obtained from pretrained weights of popular encoders as input for the 3D decoder. As a result, one could look into developing GANs with 2D encoders and 3D decoders which will be a huge performance boosts compared to fully 3D GANs and V-Net. [2, 5, 12]

The MeshRCNN codebase makes class agnostic predictions. We modified the code to use the class labels while making the predictions and to generate meshes that contain semantic information in them directly. This removes the need for postprocessing in order to predict a semantic scene from the input RGB image.

We would like to explore the efficiency of the shearing layer by replacing the 2D-3D backprojection layer in the architecture proposed by Dahnert et al [3]. However, further research is required into the placement of the instances into the 3D grid after applying the shearing layer. Future work could also explore capturing better context between objects in a scene by adding attention in the form of transformers in the vertex alignment layers.

7. Acknowledgements

We would like to thank our supervisor, Dr. Yinyu Nie, for supporting us throughout the project and for the valuable discussions. We would also like to thank the TUM Visual Computing and AI group for loaning us the compute power required to train and evaluate our method.

References

[1] Berlynoak charger wooden sofa set. 4

[2] Marco Domenico Cirillo, David Abramian, and Anders Eklund. Vox2vox: 3d-gan for brain tumour segmentation. In *International MICCAI Brainlesion Workshop*, pages 274–284. Springer, 2020. 4

[3] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. *Advances in Neural Information Processing Systems*, 34:8282–8293, 2021. 1, 4

[4] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9785–9795, 2019. 1, 3

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 4

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[8] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[10] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3

[12] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016. 4

[13] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 55–64, 2020. 1

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[15] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2974–2983, 2018. 3

[16] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2, 4