Architectural Neural Sketch: 3D Reconstruction from Single-View Sketch for Building Design

Alberto Tono
Stanford University
atono@stanford.edu

Ziang Liu
Stanford University
ziangliu@stanford.edu

Joseph Hoang Ngo
Stanford University
joseph72@stanford.edu

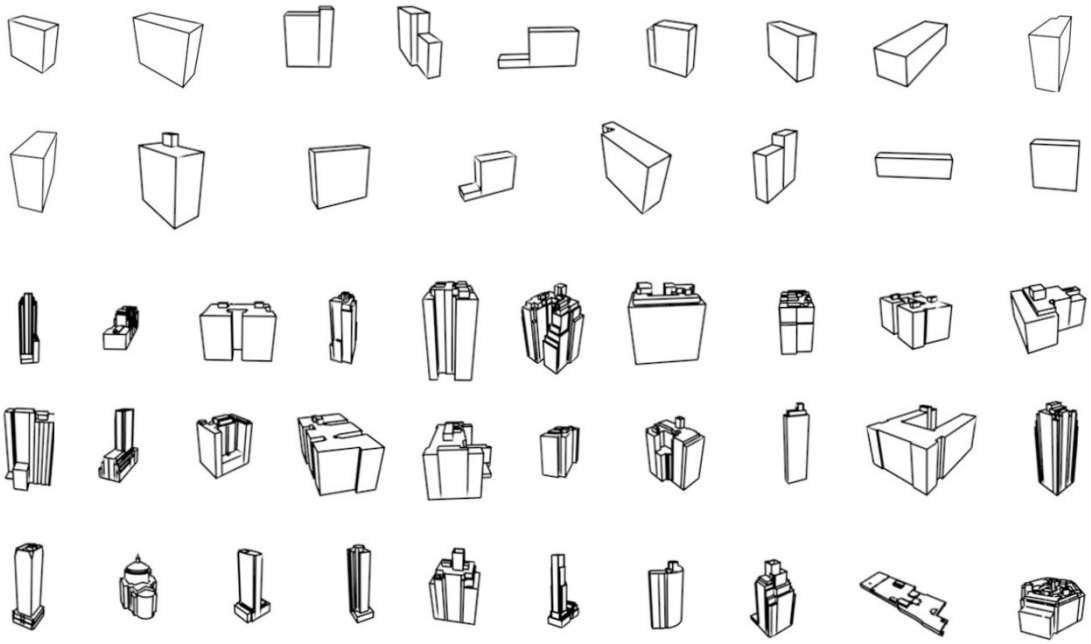

Figure 1. Dataset of synthetic sketches

Abstract

During the conceptual phases of a project, the ability to empower the creativity of all the stakeholders (architects, engineers, builders) together with real-time analysis, derived by a 3D model, is crucial for the design of high-performance buildings for a more sustainable built environment This research investigates translating single-view 2D architectural sketches of building envelopes into 3D models. This direction is crucial to allowing a real-time evaluations of the buildings performances and metrics. On one hand, we leverage traditional computer vision methods for detecting vanishing points. Because sketches are ambiguous, vary on the artistic style, and can be drawn from different perspectives occluding other parts, this approach alone won't be sufficient. Therefore, we use the 3D reconstruction provided by the detected vanishing points to validate the output of our deep learning algorithm based on the Occupancy Network. On the other hand, we employ implicit 3D geometry representation for 3D reconstruction from a single sketch, leveraging domain knowledge of the building envelopes. In this work, we provide a new dataset focused on 3D building envelope reconstruction from sketches, and demonstrate that our dataset poses a challenge to even state-of-the-art single-view 3D reconstruction methods.

1. Introduction

End-to-end training from initial sketches to 3d building would allow a more sustainable and effective decision process during conceptual design phases. This paper presents

the first dataset in that direction. Additionally, we provide code for dataset preparation, extensive analysis of the dataset based on [2], and an initial benchmark for sketch to 3D building reconstruction tasks. Our contribution is to provide a dataset that is challenging to both computer vision researchers focused on geometric deep learning and geometry processing tasks. The problem of sketch to 3d is ill-posed since there are issues about view, style variance, sparsity, and ambiguity between foreground and background [31]

2. Related work

Previous work for 3D sketch modeling and reconstruction used deep CNN to predict occupancy in a voxel grid [4] [26] [29] [28] While these methods allow an interactive editing process similar to [7], they are limited to known camera parameters. Modern approaches such as [33] use differentiable rendering techniques, volumetric rendering and other techniques to alleviate biases and noise produce by the ambiguities derived from the sketch. Since no previous work has been conducted in this specific topic, our work could open the door to a new research field that can bridge multidisciplinary fields.

This novel approach in the specific field related to building envelopes tries to fill a gap in the literature, and open a new research field at the intersection of architecture, engineering, and construction. Previous applications didn't use deep learning approaches but tried to leverage specific mathematical reconstruction of each stroke, leaving to the designers full responsibility of the 3D reconstruction. Those interfaces also required multiple edits from different views.

3. Dataset

Our dataset uses synthetic sketches, conscious of the limitations highlighted by previous work [33] [25] [32]. It uses a small portion of RealCity3D dataset, that we normalized to a canonical pose to facilitate the training procedure. Our dataset is represented by signed distance functions representations (SDF), voxels (32x32), [normals, depth maps and point cloud] in a canonical pose, adjusting all the orientations of the buildings, tracking the north position, [scaling factors, camera calibrations and poses.]

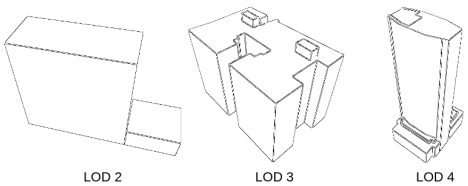

Figure 2. Dataset division based on the Level Of Details (LOD)

3.1. LOD Division

Our dataset is composed of 47 buildings categorized based on their sketch complexity following a simple rule: LOD 2 for simple extrusions, LOD 3 with more complex structure but without curves or slopes, and LOD 4 for buildings that present curve lines (circle, semi-circle) or uncommon geometry for a building, such as a hexagon. Currently, we have 17 models with LOD 2 , 20 models with LOD 3, and 10 models with LOD 4, and an example can be visualized in Fig 2 we could also follow the classification of LOD defined by [1].

3.2. Synthetic Sketch representation

The sketches have been obtained after rendering the initial polygonal meshes [15] in Blender, combining a Lambertian representation of the building with a Suggestive Contours filter to highlight the edges. Previous work demonstrated better performance because after converting these renders to synthetic sketches using a canny edge filter, the main edges and lines are preserved. During the rendering phase, we avoid the creation of additional lines caused by the shadows. Finally, we moved the camera around the object (instead of moving the object) extracting 24 camera extrinsic and intrinsic parameters R, T, and K. We made sure that our light source is aligned with the camera position and that the building had the identical appearance values (luminance) independent by the view point, that pass the photo-consistency Test (Lambertian).

* The part related to the Dataset creation is shared with **CS 348N**.

3.3. Canonical Pose

Two main objectives to consider during the initial conceptual phase of the design are location and orientation. Location is influenced by the geographical, cultural, and geopolitical context in which the building is located, providing a more holistic view of the project. Therefore, we would like to invite future researchers to pursue these essential extrinsic considerations and research directions for this specific research. We focused mainly on more intrinsic characteristics, such as orientation. The orientation drives the energy efficiency of the buildings and the indoor comfort of the occupants amongst other characteristics of the building. Taking advantage of natural daylight impacts the heating/cooling systems increasing the savings and human wellbeing. With these considerations and the intent to spark future research on this matter, we create our dataset tracking both location and orientation of these buildings. Each model in our dataset is associated with their exact position on Earth following the epsg:4326 convention according to the GeoDataFrame [6] coordinate reference system (CRS) [17]. For this specific experiment, we consider only

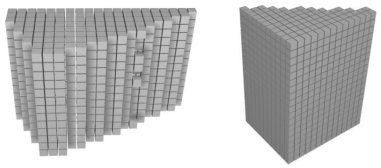

Figure 3. Voxel Artifacts

47 buildings divided into three levels of detail (LOD) as specified above.

We apply two main strategies to orient the building to a uniform canonical pose to avoid creating artifacts in the voxelization as presented in the Fig. 3.

First, we capture the main axis of the building, provided by the longer facade. Although simple and straightforward, this could be a limitation in the future, since it doesn't apply to all building topology, especially more complex ones or buildings with "L" shapes or triangular shapes as shown in Fig. 3. We projected all the building lines to the XY ground plane to find the axis direction. Although we could not extract the longest edge due to the noisy nature of this approach, we sampled points from these edges, and determined the major axis. We could also exploit the fact that buildings, for the most part, develop straight facades, and use the building footprints directly. We decided to rotate the model around the vertical Z-axis until the central axis is parallel to the Y-axis. In this way, the primary building orientation is aligned with the voxel grid. Finally, we record the rotation angle and its association with the building.

After adjusting the orientation, we scaled and centered all the buildings to facilitate the training and provide a better data normalization.

After aligning the building to the Y axis and saving the rotation, we extract our non-photorealistic render(NPR) representation of the building and convert to a synthetic sketch. We keep the building fixed and place the camera in 24 randomly sampled poses and orientations, carefully considering the lighting condition and avoiding the creation of shadows as mentioned above.

For the first experiment related to the computer vision part that exploit the vanishing points for an approximate 3d reconstruction, we manually selected the most expressive and simple sketch representing the building. This emulates how a designer would sketch a building in a quick way for the initial massing studies (overall shape of the building).

We analyze the potential issues that our dataset normalization process can resolve as follows:

- Outliers in the dataset (circular buildings, hexagonal shapes)
- Missing information. Scaling loses the information on building size and only preserves the relative ratio. For buildings that have a central atrium, with our multi-view system, we won't be able to capture their internal configuration due to the occlusion.
- Possible noise in the dataset, mitigated by aligning the models to a canonical pose.
- Inter-class variation. We divide the buildings in 3 main LOD categories to reduce this variation, and minimizes model variance within each class.

Algorithm 1: Track orientation with respect to north

Input: 3d model
Output: 3d model oriented to X,Y
$index - model$ = 0;
repeat
  project vertex to XY ground plane ;
  regress the major axis of the ellipse;
  $maxis$ = x_{min},y_{min} - x_{max},y_{max};
  for *vertex* **do**
    if *x,y* <= x_i,y_i ;
    findMajorAxis();
    if x_i *and* y_i *!= max and min;*
    ignore;
    then
      $i \leftarrow i + 1$
    end
    else
      update x_i, y_i value;
    end
    $i \leftarrow i + 1$
  end
until *each edge/vertex in the 3d model*;

3.4. Voxel Representation

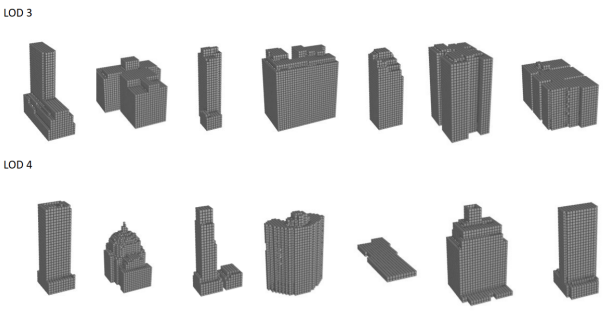

Figure 4. Batch sample of Voxel 32 LOD3-4

3.5. SDF Representation

Signed-distance functions are set in a metric space that determines the distance of a given point x from the surface of a shape. The sign indicates if the point x is inside or outside the shape. While in shapeGAN [12], voxel and implicit functions used both signed distance, we mainly focus on the binary representation of the implicit field required by OccNet. We generate the SDF point clouds using the package $mesh\_to\_sdf$.

DeepSDF [16] sampled 500,000 points from watertight shapes of synthetic objects similar to our approach, and split the points based on the sign of the distance. The sampling strategy used by [16] targeted points near the object surface. A sampling approach that better captures fine details has been used later by [13] where they used a weighted sampling based on the number of points nearby; they emphasize the importance of small-scale details, which we currently do not need at this stage.

As in DeepSDF, our building meshes are not all watertight. To obtain shells, they positioned 100 cameras homogeneously around the object as in Fig 5, and densely sampled points on the surface contour for each camera by back-projecting the depth pixels from the render, with surface normals oriented towards the camera. This process is prone to errors when the object is not watertight, and the normal of the triangle mesh is visible on both sides.

Adapting the approach in Occupancy Network, we define a shape with points from the voxel grid.

$$o : \mathbb{R}^3 \rightarrow \{0, 1\}$$

We extract the SDF from this voxel grid occupancy function. So the series of points identify the object's surface with the voxel value in which the surface is contained. We scale the voxel representation to [0,1] to align with Occupancy Network.

$$f_\theta : \mathbb{R}^3 \times \mathcal{X} \rightarrow [0, 1]$$

To determine if a point lies in the interior of a mesh (e.g., for measuring IoU), the mesh needs to be watertight. OccNet used the code provided by [21] , which performs TSDF-fusion on random depth renderings of the object, to create watertight versions of the meshes. They centered and rescaled all meshes so that they are aligned with the voxelizations from [3]. The 3D bounding box of the mesh is centered at 0, and its longest edge has a length of 1. They sampled 100k points offline in the unit cube centered at 0 with additional slight padding of 0.05 on both sides and determined if the points lie inside or outside the watertight mesh. To this end, they counted the number of triangles that a ray that starts at the given point and is parallel to the z-axis intersects. If this number is even, the point lies outside the mesh. Otherwise, it lies inside. We adapt the same approach to preprocess our dataset.

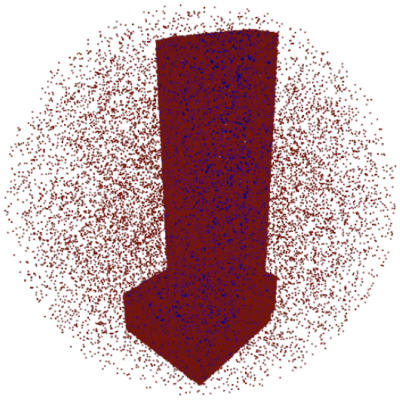

Figure 5. SDF visualization. Blue points correspond to internal and red to external samples. Credit to mesh_to_sdf. During the training of OccNet we did not used the mesh to sdf version since our implicit function is based on the binary classification of whether the point is inside or outside the shape, and we are not interested the specific distance of the points during training. We generate SDF for our dataset to spark further research in this direction.

4. Baselines

The main experiments are related to monocular sketch-based 3D building reconstruction and single view reconstruction (SVR) for sketch modeling applications in the architectural, engineering, and construction (AEC) space.

The first methodology employs the ability to detect vanishing points (VP) [20] [23] in real-time during the sketching process to construct a 3d reconstruction of the building [27] [5] [22] [24] [34]

The second approach is based on implicit representation [26], [18] and [8] [16] [30] differentiable rendering [11] [28] to optimize the final mesh reconstruction. While approaches such as OccNet [14] struggled at generalization and to reproduce fine details, this application could benefit from it, since we aim to reconstruct the overall volume and not the exact details during the initial stages. Hence, for this project we choose OccNet [14] as our main evaluation baseline.

5. Vanishing Points

This research has been inspired by the recent development in lifting sketches to 3D to allow new viewpoints and more refined scaffolding structures [5]. Currently it is possible to detect vanishing points from a sketch in real-time [27] since it is tailored for situations when the camera parameters are unknown and leverage Hough Transform specifically for building exteriors. It finds a series of vanishing

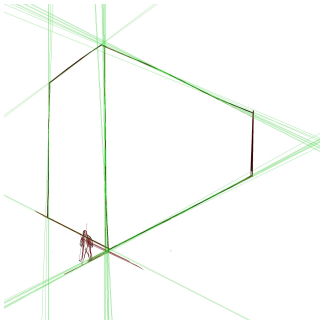

Figure 6. Line detection with OpenCV using Hough Transform before applying voting scheme.

points (VPs) by continuously ignoring inliers of any recognized VPs from minimal sample sets until the stop criterion is reached.

While work in the past has been focused mainly on interior spaces [24] [34] that mainly exploit the normal map of the environment, we could use a similar approach by adding an initial styleGAN [10] layer that convert sketches to normal maps representations. In [24] they were able to combine robust vanishing point and line detection methods with a deep CNN. But as similar to LayoutNet they mainly exploit indoor features, such as corners, boundaries, and interior size and proportion difficult to be exploited for building envelopes. Since most of those codes have not been released we decided to approach with more common baselines. [20] and [22].

Our method need to be robust to noise since hand-drawings inherit human imperfections in the act of sketching. We applied an hough transformation with polar coordinates, since it is bounded and more robust to vertical lines. After that we applied a voting scheme to detect possible vanishing points.

The two pairs of parallel lines in 3D have directions d_1 and d_2, and are associated with the points at infinity $x_{1,\infty}$ and $x_{2,\infty}$. Let v_1 and v_2 be the corresponding vanishing points. Then, we find that the angle θ between d_1 and d_2 is given by using the cosine rule: where $\omega = (KK^T)^{-1}$.

In 3D we can compute its associated vanishing line ℓ_{horiz} and its normal $K^T\ell_{\text{horiz}}$. Therefore, we can determine the angle θ between two planes by computing the angle between each of the planes' normal vectors n_1 and n_2. We derive the angle θ between two planes with vanishing lines ℓ_1 and ℓ_2 respectively:

$$\begin{aligned}\cos\theta &= \frac{n_1 \cdot n_2}{\|n_1\|\|n_2\|} \\ &= \frac{\ell_1^T \omega^{-1} \ell_2}{\sqrt{\ell_1^T \omega^{-1} \ell_1}\sqrt{\ell_2^T \omega^{-1} \ell_2}}\end{aligned} \tag{1}$$

But recall that ω depends on the camera matrix K, which is potentially unknown at this time. We know that K has 5 degrees of freedom and that $v_1 \omega v_2 = 0$ provides only one constraint, so we do not have enough information to calculate K. Then we know that $v_1 \omega v_2 = v_1 \omega v_3 = v_2 \omega v_3 = 0$. And if we make the assumption that the camera has zero-skew and square pixels (since it is a sketch, and there is no camera), then we know that ω.

Since we can only know ω up to scale, we can solve it and then we can use Cholesky decomposition to compute K. Performing the initial camera calibration. Once K is known, then we can reconstruct the 3D geometry of the scene. [23]

First, we apply Hough Transform to a grey canny version of the input sketch image in order to detect edge lines. However, the output is very noisy, having many lines for each edge due to ambiguities in the sketched line. To reduce noise, we filtered out non-vanishing lines, rejecting lines of slope near (within a θ degree difference) to 0 degree or 90 degree and storing the others. We also only take the 25 longest lines to lower the amount of noise.

After noise reduction, we use current methods of vanishing point detection which applies a grid over the image's intersections and the grid cell with the most intersections contains a vanishing point. However, this method has trouble detecting the two horizontal vanishing points at the same time, since the 4 vanishing lines have two intersections that are just corners of the cube and two intersections that are vanishing points. To solve this, we calculate the "left" vanishing point by only considering intersections with an x coordinate within the range range of 0 to .45 * image width; the same is done for the "right" vanishing point but with an x range of .55 * image width to image width. Since we need 3 orthogonal vanishing points to solve to the camera matrix, we can either assume that the 3rd vertical vanishing point is at infinity or apply this detection algorithm in the y axis, use the image height, and find the "top" or "bottom" vanishing points Fig. 7.

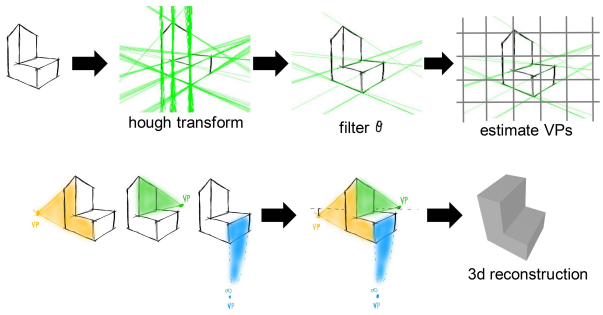

Figure 7. Vanishing point detection

Method	e_{rec}	IoU_{points}	IoU_{voxels}
Pretrained	664.303	0.033	0.007
Building Only	**307.564**	0.485	**0.037**
Pretrained + Building	444.027	**0.501**	0.036

Table 1. Metric evaluation of reconstruction results of our three models

6. Evaluations

To leverage the full representation power of OccNet, we train the model under three settings: 1) pre-trained on ShapeNet 2) trained on our dataset from scratch 3) pre-trained on ShapeNet and fine-tuned on our dataset. For settings 2 and 3, we train on our dataset with synthetic sketches and ground truth voxel as input for 100k steps on a Nvidia RTX 2080S GPU. We then evaluate the performance of the three models using both synthetic sketch and real sketch. Since we are most interested in the ability of the model to reconstruct 3D structure from sketches unseen during training, in this part of the evaluation we only sample from the test set. Using one randomly selected sketch per model as the input, we compare the model performance with 6 synthetic sketches and 6 real sketches evenly distributed amongst three LOD classes, and report the results in Table 2.

Overall, directly applying a model pretrained on ShapeNet results in ill-shaped or near-empty mesh, demonstrating that our dataset poses a new problem that can't be solved directly with off-the-shelf pretrained models. Both of the second and the third method works well in reconstruction, but due to the limitation of our relatively small dataset, the reconstructed meshes are highly dependant on the training set distribution, and shows poor generalization capabilities. The model pretrained on ShapeNet and fine-tuned on our dataset is able to recover finer detail and more accurate dimensions than the model trained only on our dataset, thus showing promising future application of transfer learning to improve the reconstruction accuracy.

Similar to OccNet, we select classic **quantitative** metrics including IoU [19] and 3D Intersection over Union (Voxel IoU) [14] for evaluation. Furthermore we will leverage the initial reconstruction method based on vanishing points [9] as an additional metric evaluating the performance of the learned implicit representation. Table 1 shows quantitative comparison of model performance. The model trained only on our building dataset gives lower reconstruction error and voxel IoU, but we suspect the reason is overfitting to the training set. As shown in Table 2, the model trained with only synthetic building sketches has worse performance on real sketch reconstruction tasks.

Table 3 shows the reconstructed meshes of our best model on sketches drawn from the training set and the test set. OccNet is able to encode the building models with fine details, but when provided with unseen sketches as input, OccNet predicts mesh very similar to the meshes in the training set, even though the input sketches are visibly different to a human eye. This shows that although state-of-the-art implicit representation methods performs well with encoding meshes in the training set, they suffer from poor generalization to previously unseen input drawn from a distribution different from that of the training set.

Finally, in the future, it will possible to evaluate the reconstruction also in a **qualitative** manner, using perceptual studies for this specific application, the 3D reconstruction should serve for real-time analysis, so experts could confirm if the output mesh could be used for real-time analysis and it could provide reliable results. This qualitative approach will focus on the learned model's ability to capture the real volume of the buildings on an analytical level, in order to allow engineering analysis on the output mesh such as computational fluid dynamic (CDF), Structural Analysis, Radiance and so on. Since this tasks require specific domain knowledge it will be conducted by a small number of experts in the future.

7. Limitations

The limitations in this work are related to three main aspects.

The first limitation lies in the dataset creation and the way we designed the algorithm to properly orient the buildings with the voxel grid to avoid artifacts. Currently we perform the rotation with the angle drawn from the major axis of the building to the y axis, but unfortunately this is not applicable to buildings with square or circular footprints.

Secondly, we are using only synthetic sketches during training, without considering limitations related to the sketching style that different designers have, or the different perspective views that could be used.

Lastly our vanishing point detection is limited to straight shapes, this wouldn't work on buildings with curved facades. Furthermore, currently the speed of our reconstruction is not real-time.

We believe that this work could open the door to new research directions in fields such as statistics, civil environmental engineering and computer science. Our new dataset makes it possible to study confounding factors such as the north orientation of the buildings and how it is influencing their shapes. A potential future research work could be done in the way these buildings are classified, and find a way to consistently classify them based on their shapes, more than their level of details. So it would be possible to cluster buildings with more complex shapes such as an L shape or C shape.

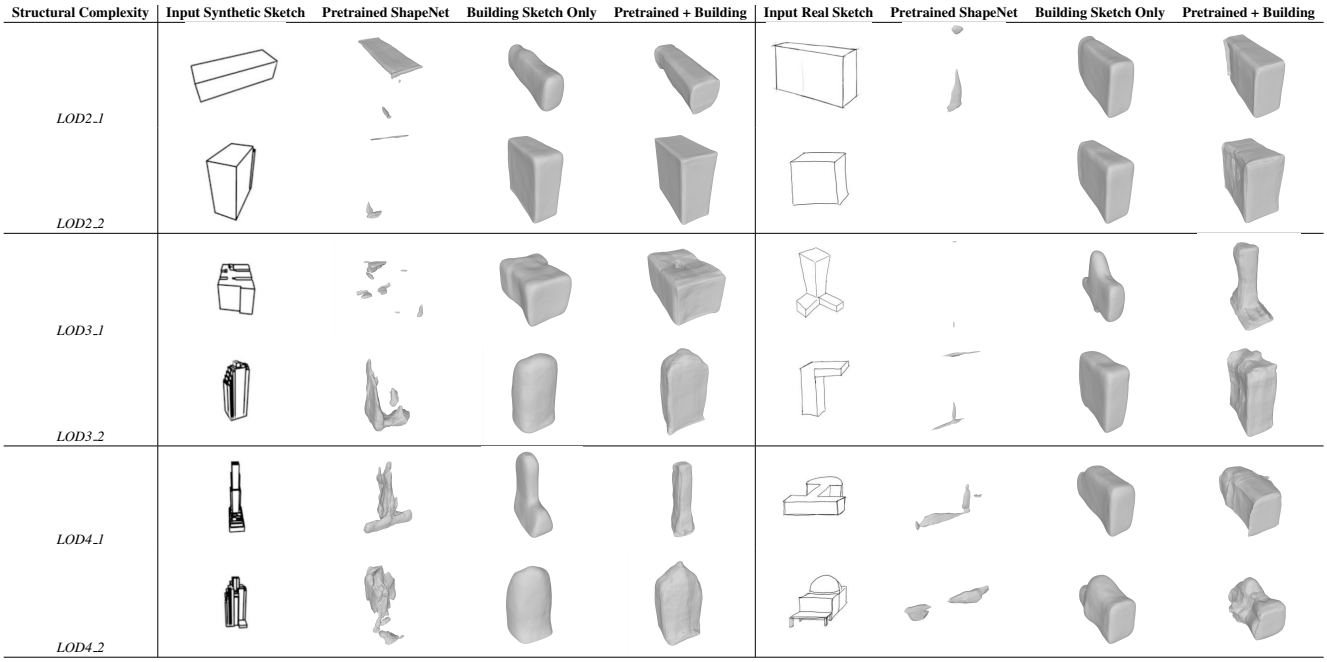

Table 2. 3D mesh reconstruction results from three models. Row index represents structural complexity (LOD). As complexity increases, the performance of the model degrades. The right most column shows the reconstruction results on real 3D sketches, demonstrating that our baseline provides non-trivial yet non-satisfactory results.

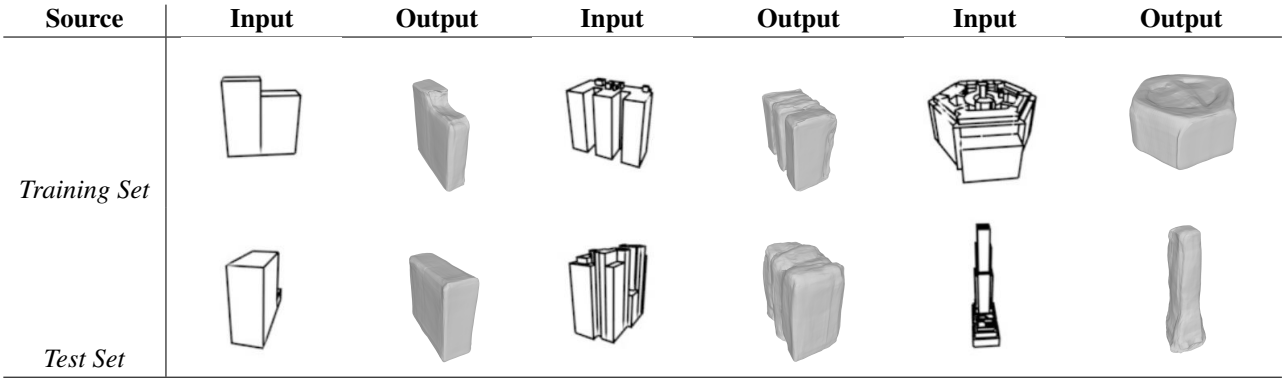

Table 3. Reconstruction using sketches from training set and test set, showing that our method recognizes the general structure from the sketches but is overfitting to the training set.

8. Conclusion

During this research, we were able to successfully translate a single sketch of a building to a 3D representation independently from the topology of the building, with non-trivial detail and flexibility. This revolutionary approach has been possible thanks to the creation of a specific building dataset developed during this research. Furthermore it will enable designers, architects to have an extra layer to evaluate their design ideas. The output mesh is watertight and it allows you to easily query the volume.

The result is not expected to be perfect but to provide an initial guide and data-driven approach related to the conceptual design phases. During this research we aimed to validate the output mesh with a 3D reconstruction from vanishing point detection techniques. While the mesh generated with the CV technique from the sketch is not accurate due to the noise in the sketch, we could take an image of the output mesh produced by our deep learning approach and compare it with the input sketch. Finally we hope to see more hybrid approaches that could take advantage of the deterministic and rigid property of CV approaches with the stochastic and flexible deep learning methodologies.

For future work, we plant to augment the dataset with

auxiliary information such as a reference to a person, or a specific note related to the dimensions of the façade, specifying width, eight, length, making it possible to run adequate real-time analysis such as cost predictions, occupancy test fitting, computational fluid dynamic analysis, view analysis, and sustainability analysis among others. We also plan to expand the size of our dataset and run more extensive baseline evaluations with more state-of-the-art 3D implicit representation methods.

Acknowledgements

This work has been supported and supervised by Andrey Kurenkov who provided great feedback and recommendations. We thank the inputs and suggestions from Jeannette Bohg, JunYoung Gwak and other TAs in CS231A at Stanford.

References

[1] Filip Biljecki, Hugo Ledoux, and Jantien Stoter. An improved lod specification for 3d building models. *Computers Environment and Urban Systems*, 59:25–37, 09 2016. 2

[2] 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. 12 2015. 2

[3] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 4

[4] Johanna Delanoy, Mathieu Aubry, Phillip Isola, Alexei A. Efros, and Adrien Bousseau. 3d sketching using multi-view deep volumetric prediction. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(1):1–22, Jul 2018. 2

[5] Yulia Gryaditskaya, Felix Hähnlein, Chenxi Liu, Alla Sheffer, and Adrien Bousseau. Lifting freehand concept sketches into 3d. *ACM Transactions on Graphics (SIGGRAPH Asia Conference Proceedings)*, 2020. 4

[6] Gerhard Gröger and Lutz Plümer. Citygml – interoperable semantic 3d city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71:12–33, 2012. 2

[7] Benoit Guillard, Edoardo Remelli, Pierre Yvernay, and Pascal Fua. Sketch2mesh: Reconstructing and editing 3d shapes from sketches. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13023–13032, October 2021. 2

[8] Benoit Guillard, Edoardo Remelli, Pierre Yvernay, and Pascal Fua. Sketch2mesh: Reconstructing and editing 3d shapes from sketches. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13023–13032, October 2021. 4

[9] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Geometric context from a single image. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 - Volume 01*, ICCV '05, page 654–661, USA, 2005. IEEE Computer Society. 6

[10] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018. 5

[11] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *CoRR*, abs/2006.12057, 2020. 4

[12] Marian Kleineberg, Matthias Fey, and Frank Weichert. Adversarial generation of continuous implicit shape representations. *CoRR*, abs/2002.00349, 2020. 4

[13] Manyi Li and Hao Zhang. D^2im-net: Learning detail disentangled implicit fields from single images. *CoRR*, abs/2012.06650, 2020. 4

[14] 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. 4, 6

[15] Charlie Nash, Yaroslav Ganin, S. M. Ali Eslami, and Peter W. Battaglia. Polygen: An autoregressive generative model of 3d meshes. *CoRR*, abs/2002.10880, 2020. 2

[16] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. *CoRR*, abs/1901.05103, 2019. 4

[17] PROJ contributors. *PROJ coordinate transformation software library*. Open Source Geospatial Foundation, 2021. 2

[18] Edoardo Remelli, Artem Lukoianov, Stephan R. Richter, Benoît Guillard, Timur M. Bagautdinov, Pierre Baqué, and Pascal Fua. Meshsdf: Differentiable iso-surface extraction. *CoRR*, abs/2006.03997, 2020. 4

[19] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 6

[20] Carsten Rother. A new approach to vanishing point detection in architectural environments. *Image and Vision Computing*, 20(9):647–655, 2002. 4, 5

[21] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 4

[22] Richard Szeliski. *Computer Vision - Algorithms and Applications, Second Edition*. Texts in Computer Science. Springer, 2022. 4, 5

[23] Jean-Philippe Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1250–1257, 2009. 4, 5

[24] R. Wang, D. Geraghty, K. Matzen, R. Szeliski, and J. Frahm. Vplnet: Deep single view normal estimation with vanishing

points and lines. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 686–695, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society. 4, 5

[25] Zeyu Wang, Sherry Qiu, Nicole Feng, Holly Rushmeier, Leonard McMillan, and Julie Dorsey. Tracing versus freehand for evaluating computer-generated drawings. *ACM Trans. Graph.*, 40(4), Aug. 2021. 2

[26] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, William T Freeman, and Joshua B Tenenbaum. MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In *Advances In Neural Information Processing Systems*, 2017. 2, 4

[27] Jianping Wu, Liang Zhang, Ye Liu, and Ke Chen. Real-time vanishing point detector integrating under-parameterized ransac and hough transform. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3732–3741, October 2021. 4

[28] Nan Xiang, Ruibin Wang, Tao Jiang, Li Wang, Yanran Li, Xiaosong Yang, and Jianjun Zhang. Sketch-based modeling with a differentiable renderer. *Computer Animation and Virtual Worlds*, 31(4-5):e1939, 2020. 2, 4

[29] Peng Xu. Deep learning for free-hand sketch: A survey. *CoRR*, abs/2001.02600, 2020. 2

[30] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images, 2020. 4

[31] Honggang Zhang Yi-Zhe Song Yue Zhong, Yulia Gryaditskaya. A study of deep single sketch-based modeling: View/style invariance, sparsity and latent space disentanglement. 2022. 2

[32] Yue Zhong, Yulia Gryaditskaya, Honggang Zhang, and Yi-Zhe Song. Deep sketch-based modeling: Tips and tricks. *2020 International Conference on 3D Vision (3DV)*, Nov 2020. 2

[33] Yue Zhong, Yonggang Qi, Yulia Gryaditskaya, Honggang Zhang, and Yi-Zhe Song. Towards Practical Sketch-based 3D Shape Generation: The Role of Professional Sketches. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020. 2

[34] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single RGB image. *CoRR*, abs/1803.08999, 2018. 4, 5