A winning solution to the Urban Scene Modeling competition

Kuo-Chin Lien
kuochin.lien@gmail.com

Abstract

This report describes one of the award-winning submissions to the 2024 S23DR challenge hosted by the Urban Scene Modeling workshop team. Targeting precise extraction of 3D wire frames from 2D images captured in urban scenes, our method explores both 2D and 3D processing techniques, and is ranked 3rd in both public as well as private test sets.

1. Introduction

This report is organized as follows: Section 1 first describes the nature of the problems and the technical decisions behind the proposed approach. Section 2 then presents findings during experiments and all corresponding modifications toward the final method.

1.1. Problem definition

HoHo dataset [1] is a recently released dataset, consisting of 4316 samples for training, 175 for validation, and 1072 as test sets in this urban scene competition. Each sample of the dataset is an urban house with information captured from at least 3 views. Each view has the following preprocessed data: gestalt segmentation, ade20K segmentation, depth map, and camera parameters. A reconstructed 3D point cloud is also provided for each scene. The competition is on designing an algorithm that can detect a set of predefined 3D keypoints (X, Y, Z) and edges (pairs of keypoints) for all test scenes within 2 hours of compute time. The evaluation metric is Wire Frame Edit Distance (WED), adopted from PC2WF [2].

1.2. Candidate methods

Considering the limited development time for the competition, a natural choice is building solutions on top of the multiview geometry pipeline: (i) detecting lines and keypoints on 2D gestalt images, subsequently also the connections between keypoints; (ii) using the corresponding depth information to project 2D keypoints to the 3D space; and (iii) aggregating 3D keypoints. This soon leads to a promising prototype highly ranked on the leaderboard, which will be described in detail in the next section.

One obvious alternative is formulating the problem as a 3D keypoint detection task, directly detecting 3D keypoints and connections from the input 3D pointcloud like a recent work PC2WF [2] does. The main advantage of such end-to-end approaches is that error propagation in the multiview geometry pipeline could be avoided. So the question comes to whether the quantity and quality of the 3D data can support.

At the first glance, the scale of the HoHo training set looks similar to what is used in PC2WF. In addition, the competition is focused on one single category of objects – urban house – that simplifies the complexity of modeling. However, the quality gap between HoHo's SfM pointclouds and CAD data used in PC2WF is a concern. This concern is later amplified after more HoHo pointclouds are investigated: HoHo pointclouds are not only relatively noisier, sparser, but also contain distracting environmental structures. The organizer provided 3D mesh information for the training set could large mitigate these problems, but building a reliable meshing algorithm for processing the unseen test data is not a trivial task, arguably making the total complexity and robustness back to a similar level to the multiview counterpart.

HoW-3D [3] presents another direction of extracting 3D wireframes. It first infers both visible and occluded keypoints from a single 2D image, build 2.5D wire frames, and then finally lifts them to 3D. The most appealing property of this direction is: keypoint detection and point-wise depth estimation could be jointly modeled and potentially help each other in a learning based neural network architecture. On the flip side, two main concerns are (a) Quality of input images: the ADE segmentation and gestalt segmentation are output of some other algorithms. Despite decent semantic quality, they may not carry sufficient information needed in the described task as raw images do. For example: absence of lighting/shading information can hurt the inference of depth and/or inference of the occluded keypoints. One may consider concatenating the depth map as a layer of input to the neural network to solve this problem; a caveat is that the provided depth maps sometimes contain irrelevant objects such as vehicles and trees that are not captured in the corresponding gestalt images, which can increase the difficulty of machine learning. (b) Complex structure of urban houses: man-made objects often encode symmetry, which has been a strong prior in computer vision and may be implicitly learned in neural nets as a cue. This is probably why How-3D authors had to select ”meaningful viewpoints” to render the CAD models for neural network training. In contrast to their manually curated training set, we do not have such a privilege in this competition. In fact, it is also unlikely that for every house only a single meaningful viewpoint is required to infer its full structure. Decorated structures such as balconies can either exist or not at the occluded sides of the house, whereas CAD models demonstrated in HoW-3D appear to have simpler and more predictable shapes. For this reason, more advanced model architectures that consider cues from multiple views are needed.

2. Implementation

With the initial analysis, we have chosen the multiview geometry pipeline to tackle the urban scene wire frame extraction problem. The following subsections describe implementation details of each step.

2.1. Gestalt segmentation

Most of our development attempts are around the gestalt segmentation because of its rich semantic information. A gestalt segmentation is generated by a domain specific model which encodes key semantics including edges and vertices into a set of predefined colors. With proper color filtering, pixel clustering, and line fitting on a gestalt image, these 2D keypoints and connection information can be extracted for a view. These can be easily done by OpenCV built-in functions. One needs to be careful on the selection of color thresholds. The default 0.5 is too conservative that can result in broken line segments which are hard to process later. We chose 10 as the default threshold, that allows more aggressive feature extraction and largely improves the accuracy. This yet can lead to some drawback: false detection on irrelevant pixels that accidentally carry similar colors to the target semantic (e.g., due to JPEG compression artifact). One prominent case is false detection of *flashing_end_point*, coded in purple, can happen around the border of *concrete* and *unclassified*, which are coded in blue and red respectively. These two categories unfortunately occupy big portions of pixels in the dataset, thus also their confusing mutation. To mitigate this problem, we narrow the color masking range to 5 for *flashing_end_point*. Morphological cleaning could be considered to remove this kind of false detection. Not that our final submission did not include all categories on the gestalt image such as *soffit*, mainly due to the development time constraint.

Another image processing technique we performed is connected component analysis to retain only the largest non-background structure in a gestalt image. This is inspired by the observation that some urban houses are very close to neighbors, so their gestalt images capture semantics located on nearby irrelevant house structures. There are a few cases where the target house cannot form a single connected component, but fortunately, the broken parts are never located around the critical keypoints and edges. Noticeable improvement on WED is observed after applying this gestalt image clean up.

Extraction of edge connection can be built upon gestalt color thresholding results. A reference method is finding the left most pixel and right most pixel in a cluster of edge pixels and checking if they are close enough to two keypoints. This method fails when two edges intersect and thus all pixels belonging to two edges get mixed in one single cluster. We developed another algorithm to avoid this problem: exhaustive hypothesis evaluation on all pairs of keypoints. First we perform Bitwise OR to combine all feature masks corresponding to all keypoint/edge of interest. This resulting binary mask serves as a connectivity mask that we will use to evaluate each connection hypothesis. The idea is that a line segment can be built to connect two keypoints if enough number of sample pixels in between the two keypoints agree. We chose 80% as the threshold to accommodate occlusions that can break edges on the connectivity mask: e.g., a ridge connection can be broken due to occlusion by a chimney. We found this new method successfully improved accuracy of 2D connection, but interestingly the ultimate WED can suffer. This observation leads us to the investigation on the quality of depth estimation.

2.2. Pointcloud back projection to replace monocular depth estimation

The first investigation is on the provided scale estimation coefficient 2.5. In the very first few scenes of the first batch of training set, it can be observed that 2.5 does not always look the best. However, with insufficient knowledge to this parameter and visualization on more training data, we still decided to continue using this coefficient.

Given the hint that the provided Colmap pointcloud has high quality than the monocular depth algorithm, we attempted to project 3D pointcloud to each 2D camera view and see if the resulting depth map leads to better final keypoint/wireframe detection. Our experiments help determine the following best practice: (1) this depth information can appear very sparse on the high resolution image plane, so some expansion is needed when searching for a keypoint's depth. (2) it is best to increase the search range progressively. We defined 6 scales to search, from small to large. (3) There can be multiple candidate depth values in the range of search. We chose the $\mathbf{min}()$ operation to determine the depth value. (4) What is the role of the default depth map from monocular depth estimation? We turned out only use it to assign depth values when the largest search region cannot even find any depth information on the pointcloud derived depth map.

2.3. Prioritizing vertices over connections

Even with much improved depth information to lift 2D detection to 3D, it is found still challenging to get vertices and edges correctly associated with the ground truth. As long as a false 3D vertex is somehow placed close to a ground truth vertex, there is a chance that the real vertex detection got hijacked during the association step and consequently its perfect edge connections. This missed assignment can even propagate: the real vertex now may hijack another vertex during association. This observation suggests that the development effort should be on vertices rather than edges because even perfect edge prediction can cause high WED penalties if vertex prediction is not robust. We decided to only submit vertex prediction, since the organizer modified the evaluation rule and empty edges are acceptable. Built upon this, we attempted to identify a type of edges that can only improve WED. Unfortunately empty edge still performs the best throughout the competition.

2.4. Filtering in 3D

Knowing there is still room to improve on 3D estimation, we performed below techniques to filter 3D pointclouds as well as vertices.

2.4.1 Vertex filtering

There was one idea of pruning non-connected vertices. This step turns out unnecessary since the rule no longer enforce connected vertices. We also found that this step often introduces higher WED, which is not favorable. We at a time used dataset distribution to set some condition to trigger this step. However, our final submissions show that higher score is achieved if completing removing this step.

Pruning vertices far away from others in 3D can be useful when some outliers appear due to noises from any steps of the pipeline. We implemented a pruning strategy to remove an isolated vertex if there is no nearby vertex to support it. While this minimal method has nothing to do if two erroneous vertices stay together, we still see improvement of scores on test and training sets.

Pruning unrealistically tall or short vertices has been used in the experiment. While statistically it brings in improvement on both training and public test sets, it is disabled in the final submission as we found not all provided 3D pointcloud are normalized. Advanced ground level estimation algorithm could be developed in the future to enable this filter again.

One motivation behind these pruning strategies is again the distracting 3D structures in the surroundings. Even though the 2D distraction on gestalt has been removed, these 3D distraction can contribute significant noise during lifting 2D estimations to 3D. Thus, in addition to pruning vertices, we also attempt to clean up 3D pointcloud before using it.

2.4.2 Pointcloud filtering

The main idea is using DBSCAN clustering to remove irrelevant 3D points. Unfortunately all submitted variants ran out of server compute time before yielding results. It is perhaps because highly parallelized program together with python package version discrepancy that use up system resource on the server but not in my local development. The final submission applies OPTICS instead of DBSCAN to complete the evaluation. Some lessons we learned at this step include: (1) in addition to the main structure, including the noise cluster predicted by DBSCAN and OPTICS is not preferred. The trade off here is cleaner surroundings vs. more information on some sparsely-reconstructed part of the house e.g., roof; and it turns out that in the HoHo dataset removing surrounding noise leads to higher gain over preserving sparse points. This suggests that more advanced point cloud cleaning method could further improve the result in the future. (2) How to define the main structure? It turns out that the number of member point is the right prior to apply. Most of the scenes have the house pointcloud sit close to (0,0,0), but it is not always the case.

2.5. ADE segmentation

ADE segmentation potentially carries complementary information to the above pipeline. For example, the well aligned ADE and Monocular Depth maps could help extract 3D information that is even better than Colmap. We leave this as future work.

2.6. Ablation

As time did not permit, we did not conduct a formal ablation study. For future reference, below is a rough timeline highlighting a few milestones.

- First submission, $WED = 2.5$, ranked 2nd; This is the baseline model with a wide color thresholding range on gestalt, disabled *prune_not_connected_vertex*, and taking $\mathbf{mean}()$ operation to determine a stable depth value from a local region.
- $WED = 2.3$, ranked 3rd; introduced pointcloud derived depth; introduced two search ranges on depth maps; replaced $\mathbf{mean}()$ with $\mathbf{min}()$. Conditioned 3D vertex pruning mechanisms and more aggressive 3D vertex merging.

- $WED = 2.0$, ranked 1st; Despite densely predicted connections, only submitted at most 2 connections per scene.
- Last day, $WED = 1.99$, ranked 1st-3rd; gestalt image clean up. Completely no connections.
- Final submission, $WED = 1.96$, ranked 3rd: Pointcloud clean up.

2.7. Learning vertex and connection

As discussed earlier, directly predicting 3D vertex and connection does not seem to be a viable path in this competition. However, there could be a chance to improve some of the steps in our pipeline with machine learning methods. One attempt we have made is training a HAWP [4] on the HoHo dataset. HAWP is designed to construct 2D wire frames for 2D images. We see the potential that HAWP could yield robust 2D wire frame and visible keypoints estimation, and later be extended to 3D estimation.

To generate training data for HAWP, we back projected ground truth vertices to 2D gestalt image planes and only retained those visible keypoints from the camera perspective. Rather than completely switching the gear, this step actually helps visualize the distribution of the training set, which was useful during we validated behaviors & tuned parameters for the main pipeline. Thanks to the light architecture of HAWP, 30 epochs of training or finetuning on a batch of HoHo set can complete within an hour in a Colab T4 environment and show promising progress. The vertex predictions became focused on the HoHo keypoints as apposed to the author's pretrained model yielding junction predictions at all (window, door, floor...) corners. Connection prediction is less ideal; considerable numbers of false alarms can be observed.

This line of exploration did not continue as we soon realized (with improving WED in mind) that the development priority should not be on 2D estimation, and it is risky to proceed to 3D estimation with yet imperfect 2D under an all-in-one neural network architecture. We encourage future work to investigate this intriguing topic on large real world datasets such as HoHo.

References

[1] Jack Langerman, Caner Korkmaz, Hanzhi Chen, Daoyi Gao, Ilke Demir, Dmytro Mishkin, and Tolga Birdal. S23dr competition at 1st workshop on urban scene modeling @ cvpr 2024. https://huggingface.co/usm3d, 2024. 1

[2] Yujia Liu, Stefano D'Aronco, Konrad Schindler, and Jan Dirk Wegner. Pc2wf: 3d wireframe reconstruction from raw point clouds. *arXiv preprint arXiv:2103.02766*, 2021. 1

[3] Wenchao Ma, Bin Tan, Nan Xue, Tianfu Wu, Xianwei Zheng, and Gui-Song Xia. How-3d: Holistic 3d wireframe perception from a single image. In *International Conference on 3D Vision*, 2022. 1

[4] Nan Xue, Tianfu Wu, Song Bai, Fu-Dong Wang, Gui-Song Xia, Liangpei Zhang, and Philip H. S. Torr. Holistically-attracted wireframe parsing: From supervised to self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):14727–14744, 2023. 4