Semantically-Aware Aerial Reconstruction from Multi-Modal Data
Supplemental Material

Randi Cabezas Julian Straub John W. Fisher III
Massachusetts Institute of Technology
{rcabezas, jstraub, fisher}@csail.mit.edu

September 25, 2015

Abstract

This document is complementary to [2]; it contains additional and expanded results to the ones presented in there.

Contents

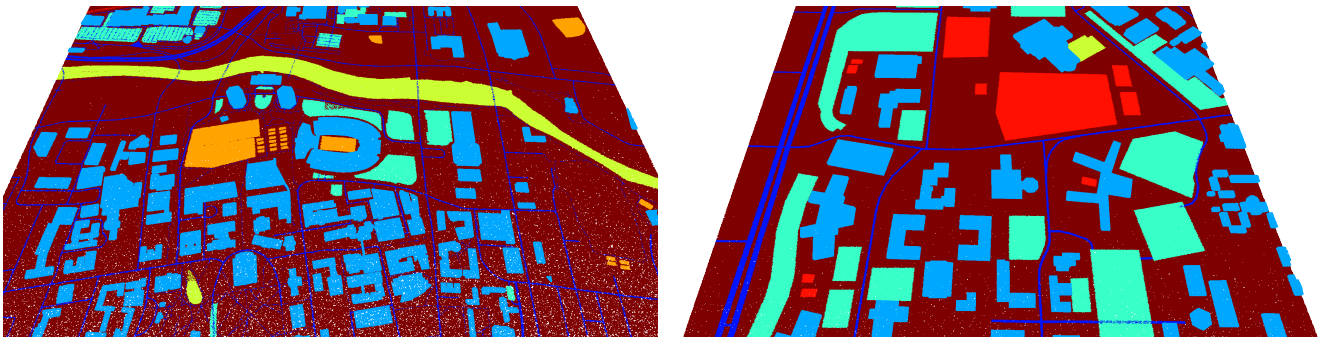

Figure 1: Real-scene semantic observations. *Left:* CLIF; *Right:* Lubbock. Semantic types: ground (maroon), building (light blue), recreation (orange), roads (dark blue), grass (light red), parking lots (teal), rail (royal blue), and water (yellow).

1. Real-Scenes

This section outlines results on scenes from Lubbock and CLIF (Ohio State University) [9] datasets obtained using the SAAR model.

1.1. Semantic Observation Generation

The semantic information for real scenes was obtained via OpenStreetMap (OSM) [5]. Specifically, areas surrounding the scenes were extracted using JOSM [8]. The map data was turned into 3D by using OSM2World [7]. OSM2World extrudes the 2D map data by parsing the OSM attributes and projecting them to 3D. Points were sampled on the 3D geometry to produce the semantic point clouds used as observations in the inference. The number of input points was controlled by varying sampling density from 0.1, 0.25, 0.5, 1, to 2 points per meter squared. See Fig. 1 for the extracted semantic point clouds of both CLIF and Lubbock scenes. Note that the point clouds do not require manual alignment as they are geo-referenced by construction. A minor complication arises since OSM does not have actual building elevation for most of the scene. This issue is addressed by our model in the primitive to semantic point association.

The CLIF scene contained seven semantic observations types: road, building, parking lot, water, recreational, rail and ground. The Lubbock scene contained seven slightly different semantic observations types: road, building, parking lot, water, grass, recreational and ground. It is important to note that neither scene contained trees, a major element in both scenes. As we will see in §1.2, SAAR is able to correctly cluster trees despite not having semantic observations of this type.

1.2. Semantic Meaning of Learned Clusters

As demonstrated with the synthetic cities in the paper, we can analyze the meaning of learned clusters by looking at the high probability semantic observations under each of the inferred semantic component distributions. The learned semantic distribution as well as the scene primitives (color-coded according to cluster assignment) are shown in Fig. 2 for MF-4 and TG-8 for the Lubbock scene. The underlying meaning of the clusters in MF-4 (top row of the figure) can be easily ascertained: the ground is represented by cluster one; buildings are represented by cluster two; cluster three consists of the road-network; cluster four is a mix of ground and buildings. The meaning of this last cluster is somewhat ambiguous; after checking the visual data we can see that despite being correctly clustered, most of the primitives in this area should be labeled as trees not buildings or ground. This miss-classification occurs because the semantic observations do not contain "tree" annotations. Importantly, the model is able to recover from this large amount of missing data and correctly cluster all of "tree" primitives as we can see from the visual evidence. However, because the model doesn't have semantic observations of type tree, the meaning of the cluster as learned by the model is incorrect.

The bottom row of Fig. 2 contains the TG-8 results. When compared to the MF model, we see that TG explains the same data but using different clusters. For example, buildings in TG-8 are represented by three clusters (numbers one, five, and seven) which roughly correspond to the sides, and rooftops as seen in the 3D scene. The two roof clusters (numbers five and seven) are likely due to different height attributes in the scene. This type of behavior where the structured prior enforces different properties (*e.g.*, different height profiles) is visible in all of our experiments.

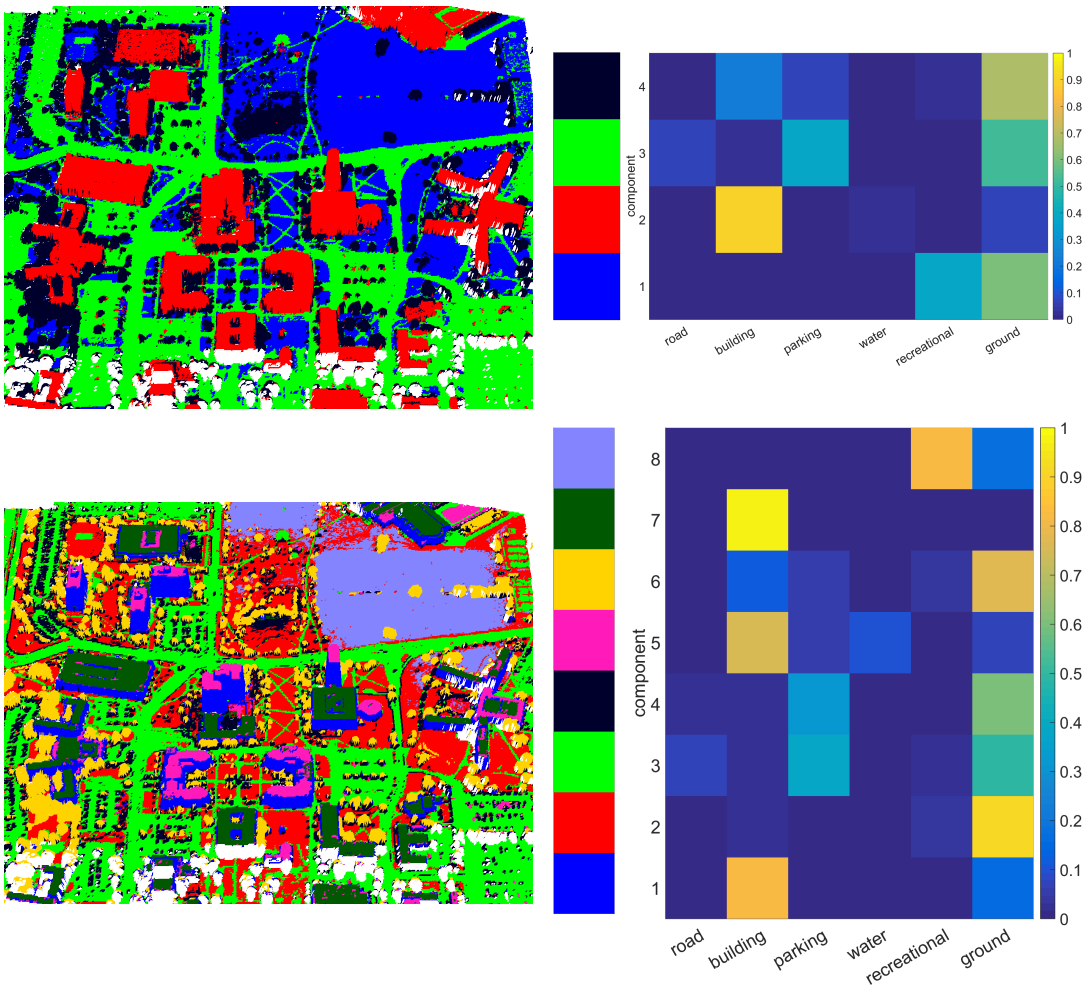

Figure 2: Semantic meaning of cluster for the Lubbock scene. *Rows:* different models: MF-4, TG-8. *Columns:* 3D scene colored according to cluster assignment; semantic mixture components φ^S. Note that the colorbar in the middle serves as a mapping between cluster number and color representing that cluster.

1.3. Manhattan Frame Orientations

In this section we expand on the orientation clustering results shown in the paper. SAAR clusters surface normals in the correct space, the 3D unit-Sphere, $\mathbb{S}^2$. Fig. 3, shows the learned orientation clusters for the Lubbock scene using MF-4. The figure shows pixels in the three input images according to the learned clusters. The visual evidence in all three images confirms the underlying meaning of the clusters shown in Fig. 2, *i.e.*, ground, buildings, roads, and vegetation respectively.

The fourth row of Fig. 3 shows the surface normals of the primitives assigned to each cluster in the unit-sphere (the spheres has been oriented so that the scene up-direction is out of the page). Several observations can be made from this row; first, we can see that most clusters are centered around the up direction and have distinct and tight clusters around it, for example the ground cluster (number one). Second, dominant scene directions can be easily seen from the clustering: namely north and south, east and west, and north-east and south-west as seen by looking down onto the scene in the building and road clusters. Third, we can see that the vegetation cluster does not have this orientation preference that the other three clusters have. Intuitively this makes sense as trees do not follow any specific pattern.

The last row of Fig. 3 shows the assignment of surface normals to the Manhattan frames of each learned cluster; learned frames are plotted as lines colored according to direction (blue, red, green), surface normals are colored according to their assignment to each direction. From the figure we can see that the points are almost always assigned to the closest frame. Furthermore, we can see that some frames are explaining multiple directions, *e.g.*, the green direction in cluster two.

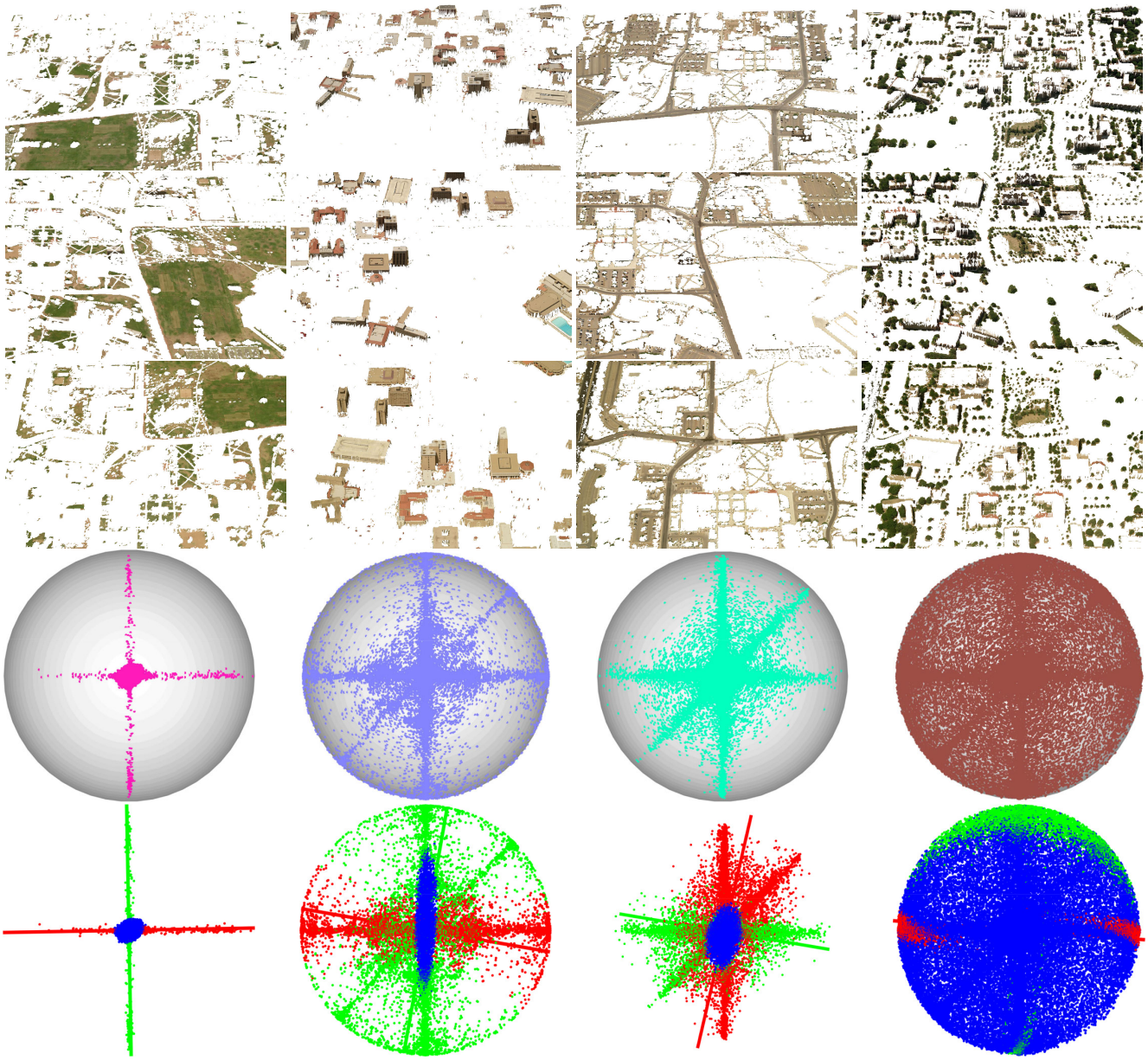

Figure 3: SAAR MF-4 model, Lubbock Scene. *Columns:* Learned semantic clusters: *ground, buildings, roads, and vegetation*. *Rows:* 3D scene as viewed by each of the three input images; cluster orientations; and MF axis assignment (color corresponds to frame assignment). Spheres oriented so that scene up direction points out of the page.

2. Handling and Predicting Missing Data

One of the main advantages of probabilistic modeling is the ability to easily handle noisy and missing data. We saw an example of how the proposed model handles noisy data in the prior section; here we show that the model can infer cluster assignments when a portion of data is missing as well as its ability to predict the missing data. Specifically, we will learn cluster assignment and cluster posterior parameters using all the visible data for a scene: appearance, geometry, and semantic observations. Visibility here, refers to triangles that have image evidence associated with them. The learned cluster parameters will then be used to predict the cluster assignment of non-visible scene primitives using their location, orientation and semantic observations. Once the non-visible primitives are assigned to a cluster, we can predict their appearance by

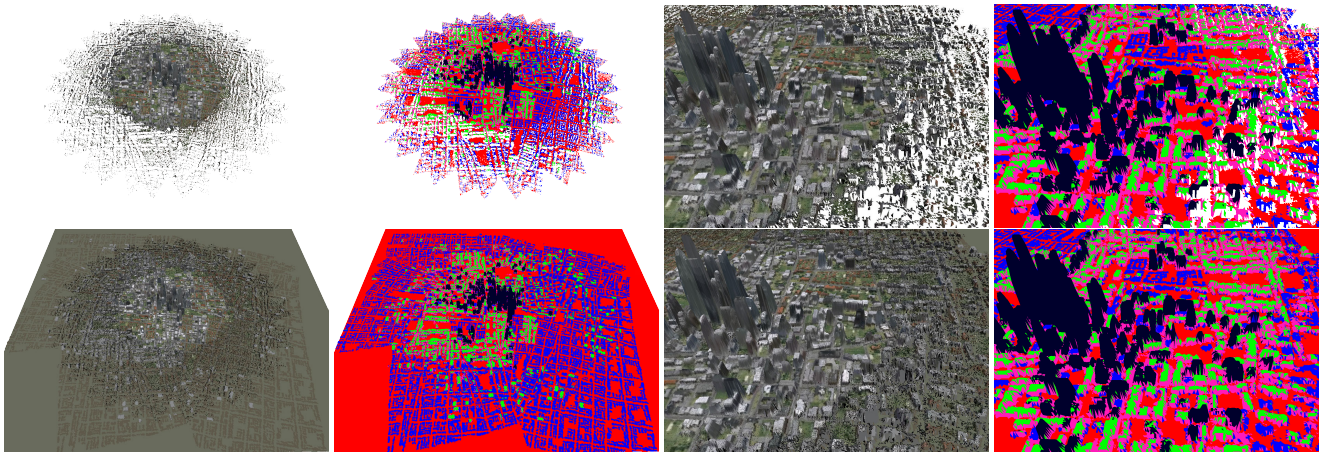

Figure 4: SAAR missing data cluster assignment and data prediction (City1-1). *Top:* two views of the visible primitives' appearance and cluster assignment. *Bottom:* predicted appearance for non-visible primitives and cluster assignment. SAAR can use any subset of available data to infer labels for each primitive. Given these labels, the missing data modalities can be predicted by sampling from the posterior of the respective component. Notice how the cluster assignments match in both visible and non-visible regions.

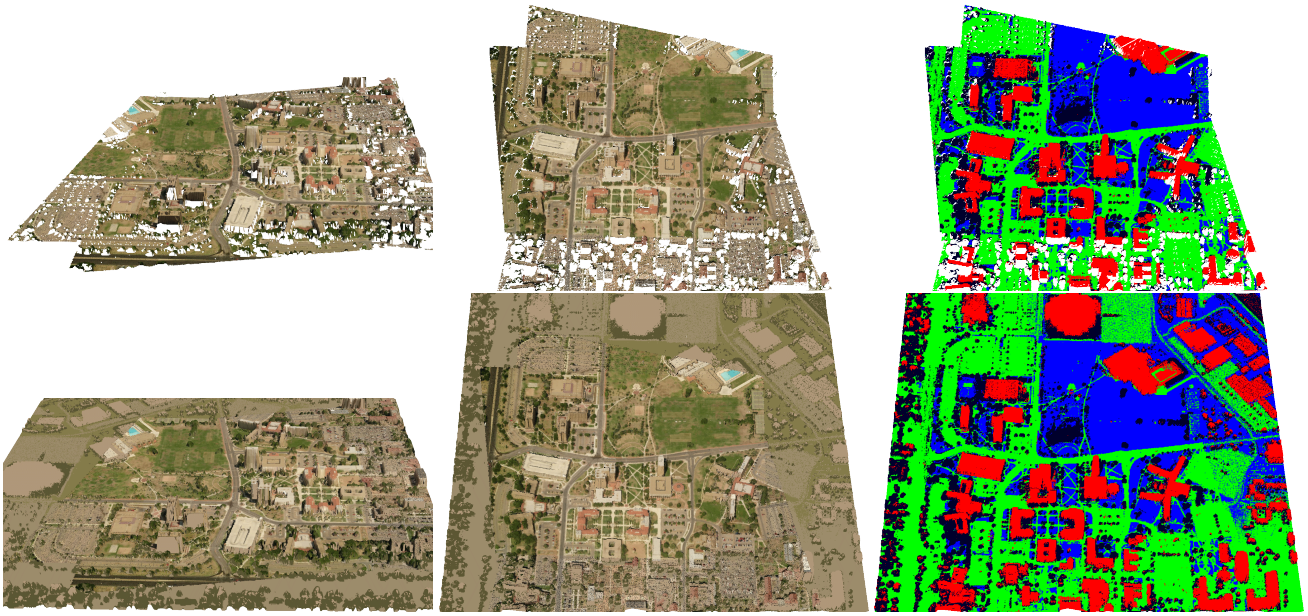

Figure 5: SAAR missing data cluster assignment and data prediction (Lubbock). *Top:* two views of the visible primitives' appearance and cluster assignment. *Bottom:* predicted appearance for non-visible primitives and cluster assignment.

sampling from the corresponding appearance component posterior. The procedure outlined above was performed for synthetic and real scenes, where the mean of the appearance posterior was used instead of sampling. This change was performed for ease of visualization. The paper contained results for the City3 and CLIF scenes; here we include results on City1-1 (MF-5) and Lubbock (MF-4), Figs. 4 and 5 respectively.

Figure 4 shows the results on the City1-1 scene. As can be seen in the figure, the assignment of non-visible primitives to clusters is consistent with the visible regions. This is a result of the high quality semantic observations used in this reconstruction. On the other hand the predicted appearance is not very detailed. This is expected due to the small number of appearance clusters (five) used. As a result only main color trends are learned. Figure 5 shows the results on the Lubbock

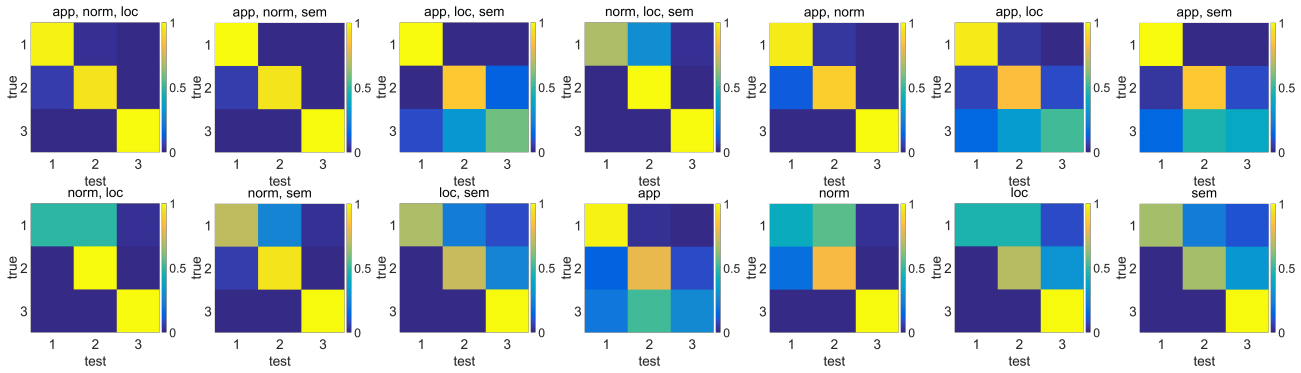

Figure 6: ToyCity2 label validation confusion matrix, TG-3. Each matrix compares the resulting labelling obtained using the specified modalities and the labelling obtained using all available information for 1000 visible primitives. In general as modalities are added the results improve.

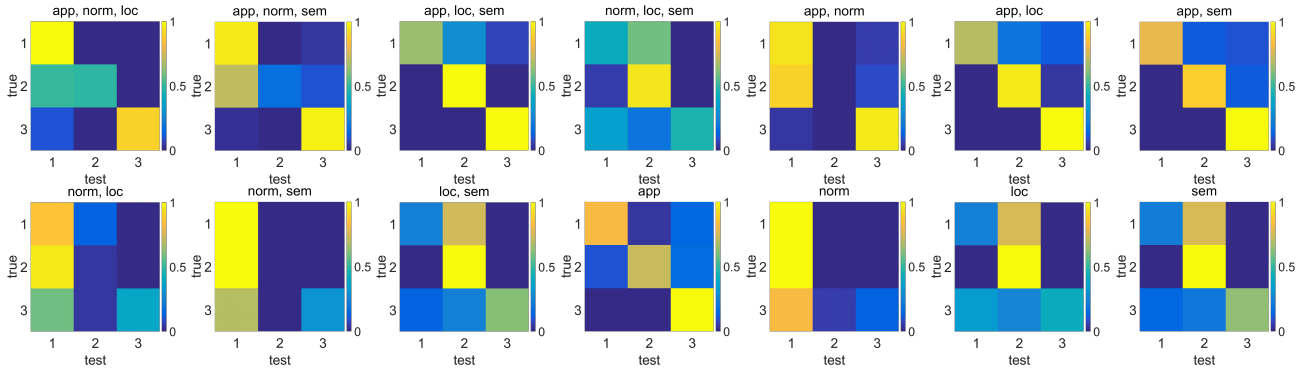

Figure 7: ToyCity2 label validation confusion matrix, MF-3. Each matrix compares the resulting labelling obtained using the specified modalities and the labelling obtained using all available information for 1000 visible primitives.

scene. From the figure we can see that the labeling outside the visible support is consistent with that of the visible region. Some miss-assignments occur on flat regions where the available data is somewhat ambiguous, *e.g.*, right portion of the scene (as seen in the bottom right figure). The predicted appearance for the scene is consistent with the observed scene color. As with City1-1, the small number of appearance clusters used (four) produces homogeneous appearances.

2.1. Label Validation on Synthetic City Dataset

A natural question that arises from the results presented in the prior section is how does the categorization results compare when using a subset of modalities to using all available modalities. This key question cannot be answered for non-visible primitives as we lack visual evidence but it can be examined for visible primitives. In order to answer this question we selected a set of one-thousand visible primitives as a validation set and computed the resulting cluster assignment for all combinations of modalities. We treated the labelling obtained using all data (*i.e.*, appearance, location, orientation and semantic observations) as the *correct* label and compared all other results to it. Figs. 6-9 show the confusion matrices obtained for TG-3, MF-3, TG-4 and MF-4 respectively on the TopyCity2 scene.

In general, Figs. 6-9 show that as more modalities are added the models do a better job at predicting the correct cluster. For example consider normal data and semantic data on TG-3, Fig. 6, individually label one is estimated correctly only about 50% of the time (fifth and seventh column on the bottom row). If both modalities are used, the correct estimate jumps to approximately 65% (second column on the bottom row). If in addition, we add appearance we can see that we obtain all correct answers on label one (second column on the top row). Similar behavior can easily be seen in TG-4. For both TG-3 and TG-4 the combination of appearance, orientation and location; or appearance orientation and semantic seem to produce better results. The MF models, MF-3 and MF-4, seem to follow the same trend as TG models, however, the gains are not as clear from the figures, specially on MF-3 where even using three modalities the model's performance is somewhat lacking.

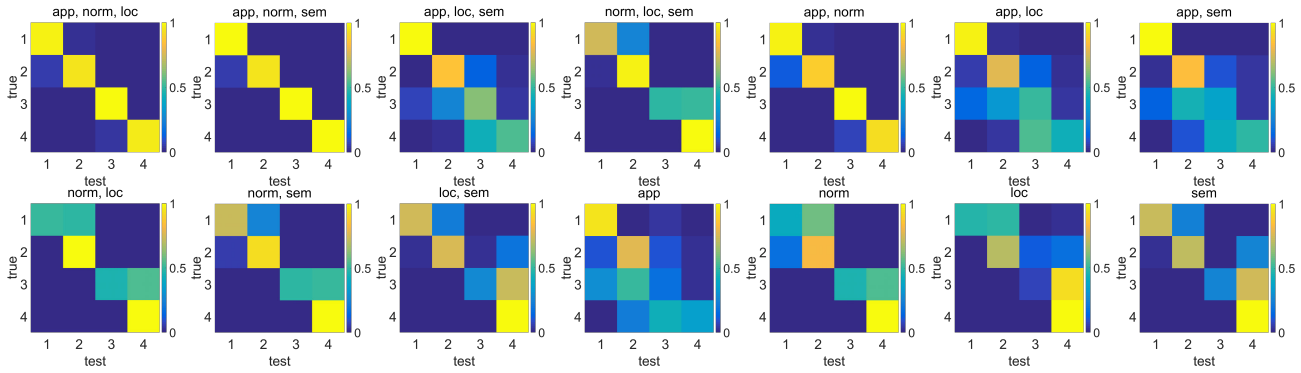

Figure 8: ToyCity2 label validation confusion matrix, TG-4. Each matrix compares the resulting labelling obtained using the specified modalities and the labelling obtained using all available information for 1000 visible primitives. In general as modalities are added the results improve.

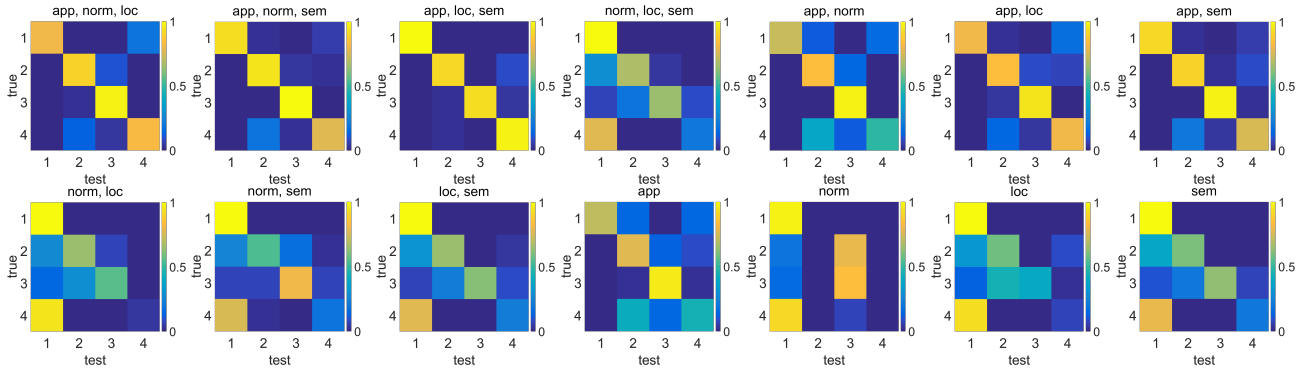

Figure 9: ToyCity2 label validation confusion matrix, MF-4. Each matrix compares the resulting labelling obtained using the specified modalities and the labelling obtained using all available information for 1000 visible primitives.

3. Synthetic City Overview

The Synthetic City (SynthCity) consist of five synthetic cities depicted in Fig. 10. SynthCity was created using CityEngine [4]. CityEngine uses procedural modeling to build scenes in a pseudo-random manner. The typical city generation process begins by randomly generating the road-network. Based on this network the ground is tessellated to produce "lots". These lots are then subdivided and assigned one of eight possible categories. Category specific buildings are generated in each lot by randomly selecting geometry and appearance elements as outline in the build rules.

Each city in the SynthCity dataset consists of eight types of scene elements: sidewalk, roads, office buildings, apartment buildings, parking lots, high-rise buildings, green-space, and trees. Fig. 10 outlines the proportions of these labels for each of the five generated cities. The label proportions are an indicator of the scene type. All labels were maintained in the 10-20% range to roughly balance the scene. The proportion of green space is exaggerated due to the fact that this category is used as a filler around the cities.

3.1. Fly-By Image Generation

A set of custom scripts (provided with the dataset) allow the simulation of arbitrary flight paths over the generated SynthCity cities. The scripts allow the collection of both textured and labeled scenes. For simplicity, all experiments in the paper were performed using the circular flybys. These consisted of a single loop, orbiting around the center of each city with the view direction pointed at the center of the city. Aerial oblique images were obtained every 20° or 30°, yielding 18 or 14 images for each city. The size of each image is 2048x1536, the collection angle of view was set to 54.5°. For faster processing we down-sample the images to 512x384. See Fig. 11 for sample fly-by and Fig. 12 for close-up of the cities.

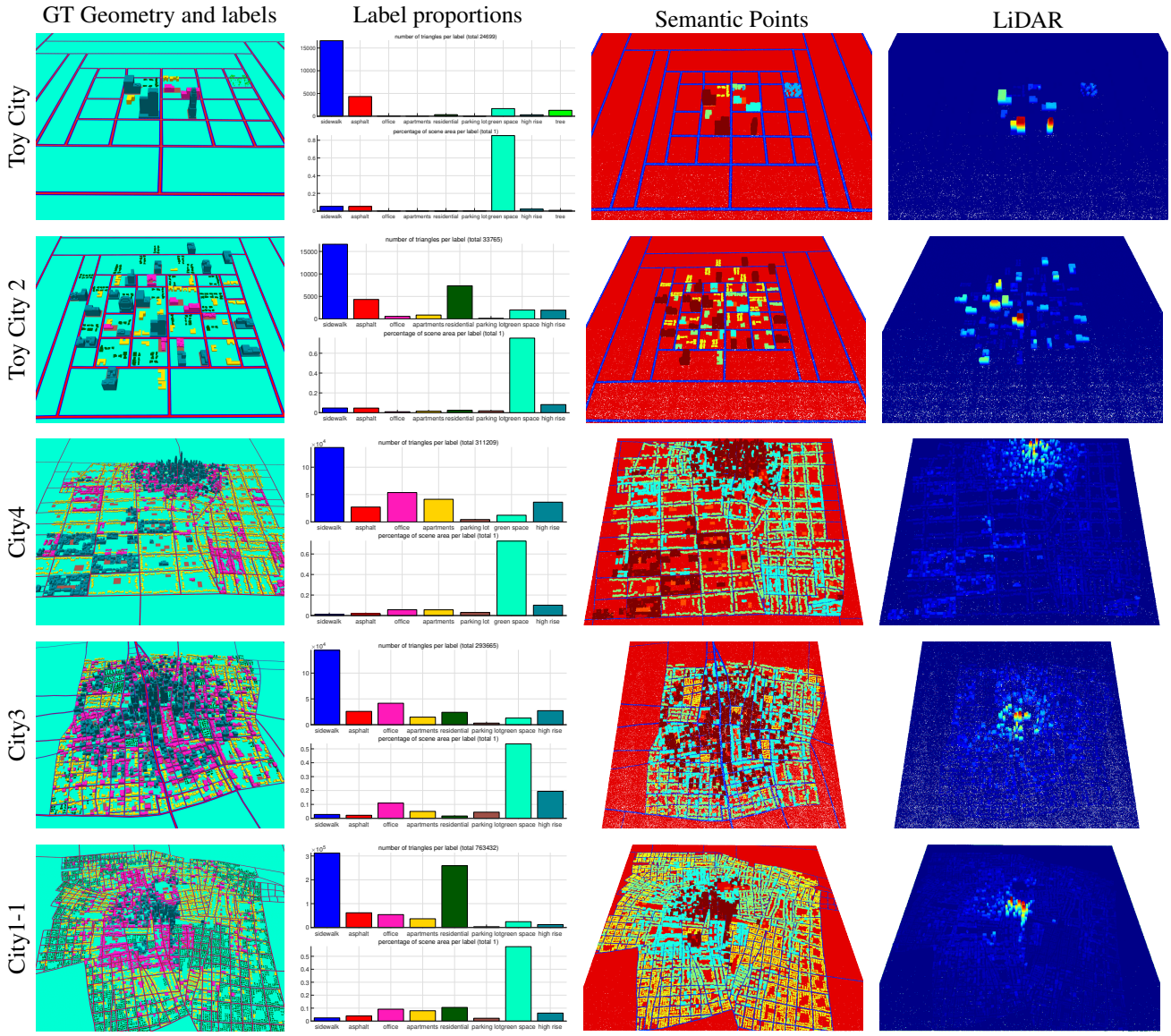

Figure 10: Synthetic City Overview. *Rows:* the five cities in SynthCity. *Columns:* ground-truth geometry and semantic label; distribution of semantic labels as a function of number of triangles and area; semantic point cloud, color-coded according to label (1.5M points each); synthetic LiDAR (1.5M points each).

3.2. LiDAR Generation

LiDAR was generated by simulating a LiDAR sensor mounted on an aircraft flying over the city. The data was generated by following a single-direction lawn-mower pattern, *i.e.*, only collecting data when north-bound. The north-bound section of the flyby was divide into 144 time instances for sufficient coverage of the scene (Fig. 11). The projection of every pixel into the 3D scene was recorded for all time instances. All 3D points were aggregated to obtain a candidate set of noiseless 3D points. Independent and identically distributed zero mean Gaussian noise with a standard deviation of 19 cm was added to each point [6]. In order to obtain the final LiDAR data, the aggregated list of points was subsampled to produce the final noisy data. The noisy data varied in size ranging from 200K-5M points depending on the scene. Generated LiDAR data for each of the SynthCity cities can be seen in the last column of Fig. 10.

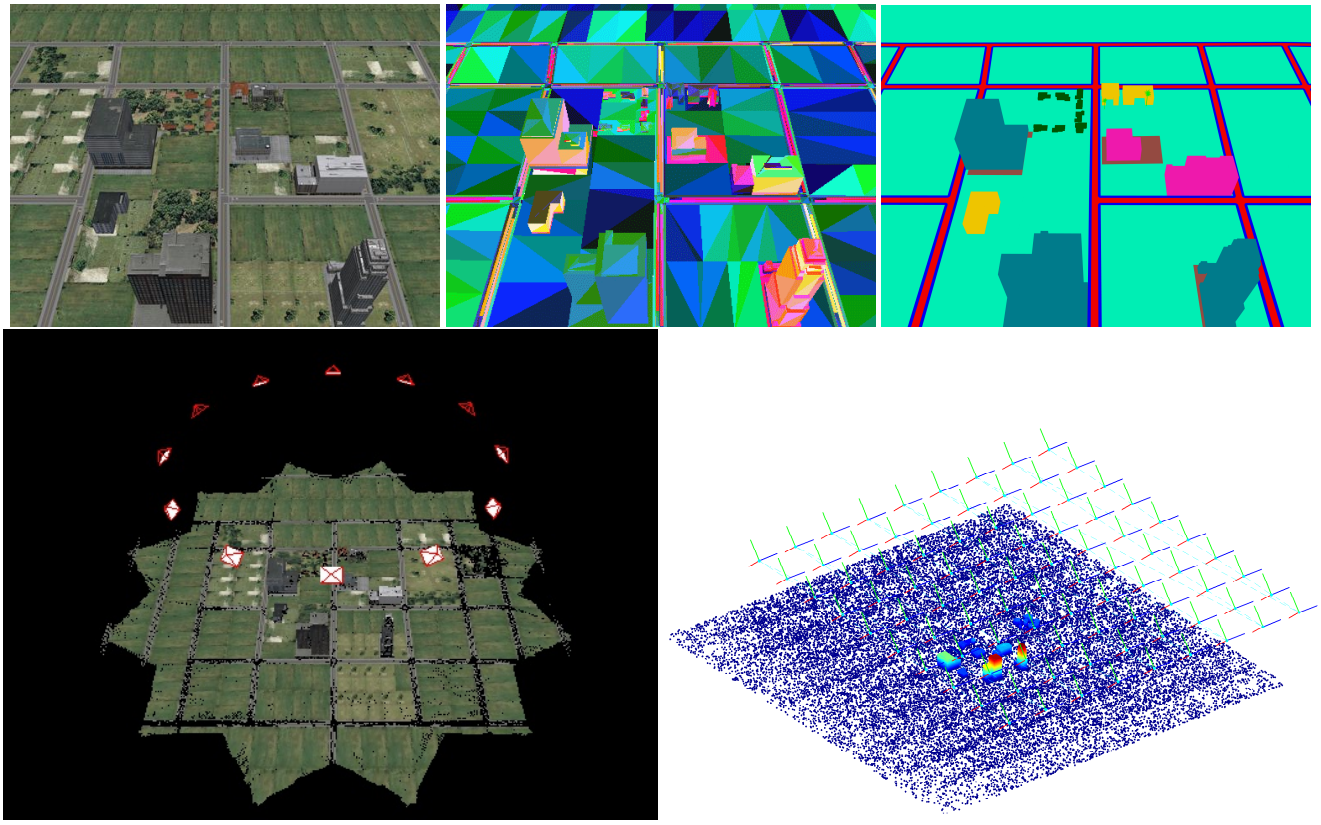

Figure 11: Synthetic City dataset details. *Left-to-right:* Sample image; ground truth geometry and labels; sample scene and camera locations, generated LiDAR.

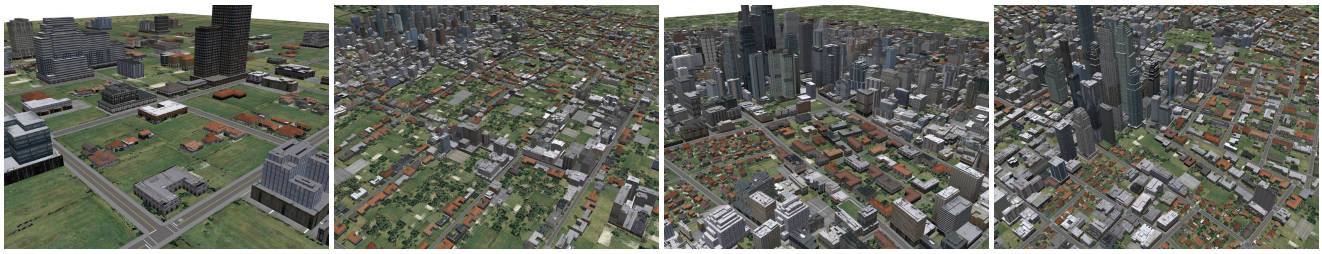

Figure 12: Zoomed in views for four out of the five cities in the SynthCity dataset (all but ToyCity).

3.3. Semantic Data Generation

The semantic data followed a similar generation step as the LiDAR for the location component, where we obtained measurements by simulating an aircraft flying over the city. In addition to the location, we also maintained the semantic attribute of the ground-truth primitive that generated the 3D point. The final measurement contained the noisy location and noisy semantic category; categorical noise was added by randomly changing the label of a subset of the points (subsets ranged from 0,2,5,10% of the points). The variable semantic noise was chosen to more accurately match the synthetic conditions to real scenes. Similar to the LiDAR the final noisy data was generated by randomly selecting 200K-5M points. The semantic point clouds for each of the generated cities can be seen in second to last column of Fig. 10.

4. Synthetic-Scenes

This section outlines expands on the results provided under the SAAR model on the SynthCity dataset.

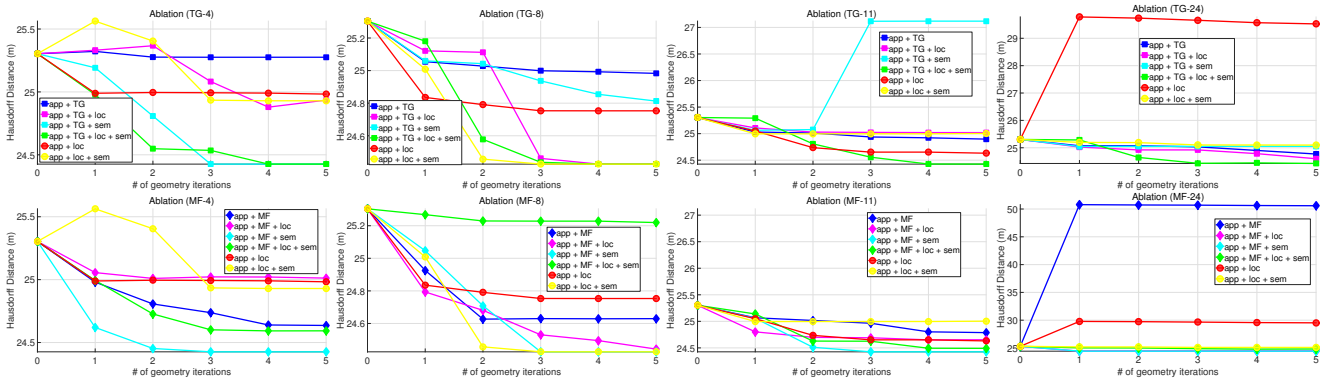

Figure 13: Hausdorff distance of ToyCity2 ablation study.

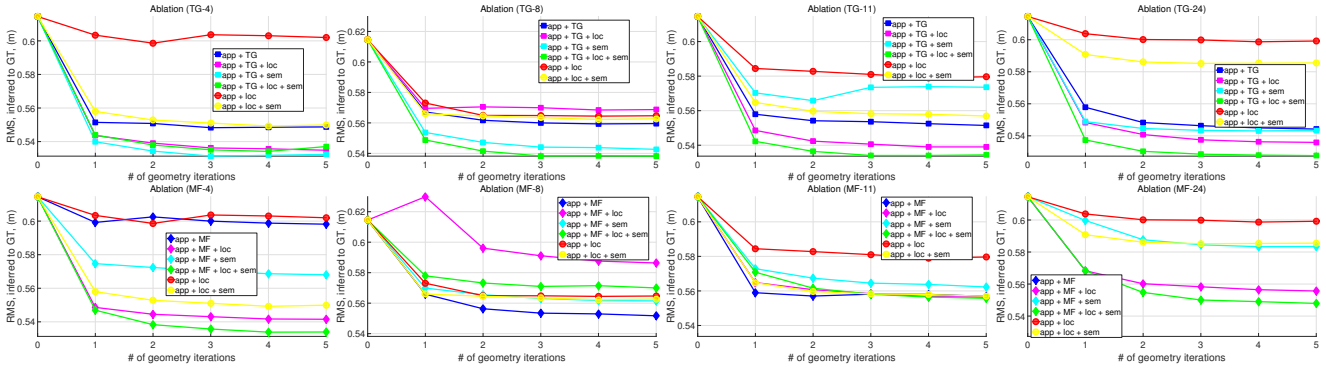

Figure 14: RMS distance of ToyCity2 ablation study.

4.1. Ablation Study - Extra Comparisons

The main benefit of SynthCity dataset is that it allows quantitative geometry comparisons of ground-truth and inferred geometry. Figures 13 and 14, show the Hausdorff, and root mean square (RMS) distances between inferred and true mesh. These distances are shown for the ToyCity2 scene as a function of number of geometry updates for a variety of modalities. A geometry update consist of updating the vertex positions of each visible primitive once. To compute the distances we used [3]. We note that Hausdorff distance includes the distance from ground-truth to inferred geometry and from inferred to ground-truth, while RMS only includes distances from the inferred to the true mesh. We found the distance from ground-truth to inferred to be of little use since the ground-truth geometry contains triangles inside buildings, which are not observable under our model and hence are not inferred. By measuring the distance from the inferred to the ground-truth such triangles are automatically left out of the error metric.

Figures 13 and 14 further reinforce the results presented in [2], namely that as modalities are added the reconstruction error decreases. Furthermore, the figures can also be used to compare the reconstruction accuracy of the MF and the TG models. Surprisingly, the reconstruction under the TG-based model yields more accurate results than under the MF-based model (in all but one configuration). We hypothesize that this is due to less degrees of freedom in the model for appearance, semantics and 3D location. Recall that each of the N_S MF has six TGs, for a total of $6N_S$ clusters to represent the orientation, but only N_S clusters for the appearance, semantics and 3D location.

For ease of visualization Fig. 15 provides qualitative results of the ablation study. The figure further highlights that as the number of modalities available increases the resulting clustering becomes more stable and consistent. Furthermore as the number of clusters increases the reconstruction becomes more detailed, allowing for more fine-grained categorization across modalities, *e.g.*, a cluster representing "roofs" is split into several "roof" clusters but now at different heights.

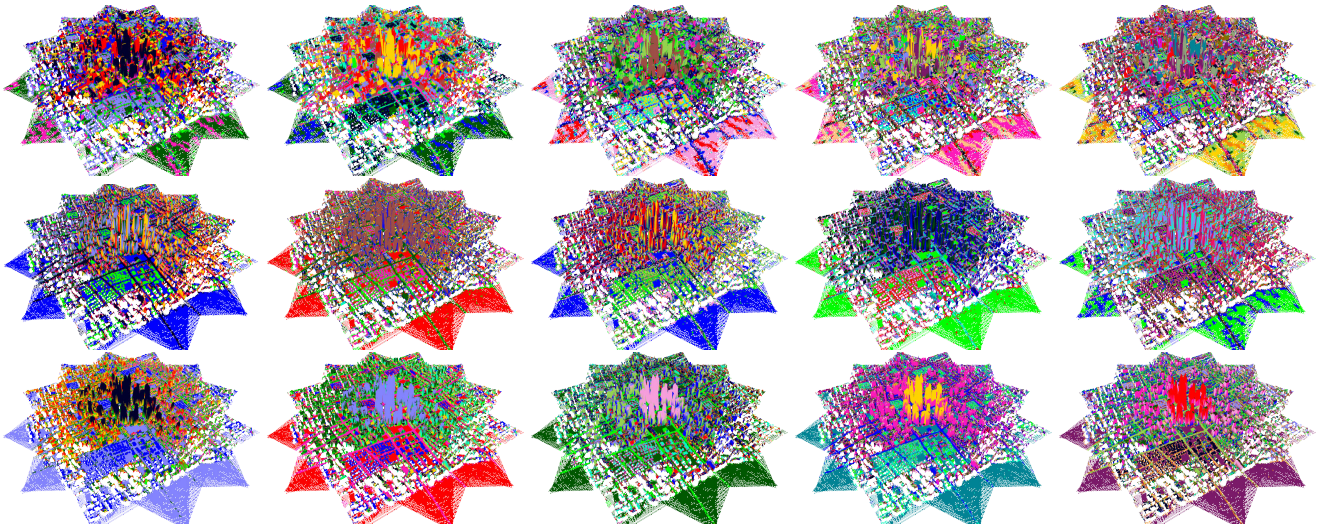

Figure 15: Learned categories for SynthCity City3 assuming the TG model for primitive orientations: *Columns:* Number of clusters: 8,10,14,18,22. *Rows:* Modalities used in the inference: appearance-only; appearance+orientations; appearance+orientations+location. Note the increasingly fine-grained categorization from left to right as well as the more and more meaningful labeling towards the bottom.

4.2. Semantic Meaning of Learned Clusters

As shown in [2] the semantic observations in SAAR can be used to attribute meaning to the learned clusters. One possible method of doing this is by looking at learned semantic component distribution φ^S, for each of the learned clusters. The collection of high probability semantic observations under each cluster forms the inferred meaning of the cluster. Figure 16 shows the semantic component distribution for two different values of N_S and orientation model: TG-8, and TG-9. To aid visualizations, two views of the scene are provided: an overlaid of cluster assignments in an input image, and the cluster assignments in 3D. Note that the mapping of component number to color is given by the colorbar in the center of the figure.

Fig. 16 shows the eight and nine cluster results. The figure shows similar behavior as that of the MF-3 and MF-4 model presented in the paper. The main difference seems to be repeated cluster assignment, *e.g.*, clusters four to six of TG-8 (third row). These clusters have similar semantics interpretation, yet the model chose separated them. This is an indicator that while their semantic meaning is similar, some other aspect of the data is a differentiating factor. In the case of clusters four to six, it is the orientations: the surface normal direction of the primitives in question point forward, up and left (with respect to the view in the first column). The meaning of clusters in TG-8 roughly correspond to open-space (1) , residential (2, 4-6), road-network (3), and commercial (7-8). Similarly, the meaning of clusters in TG-9 correspond to open-space (1,7), residential (2,8), road-network (5), and commercial (3-4, 6, 9).

4.3. Initialization Comparison

SAAR is sensitive to geometry initialization as is [1]. To quantify how much of an impact initialization can have in the model we compared the results of running the model on five different initial configurations. That is, we created five different initial triangulations, 64K, 91K, 140K, 231K, and 279K for the ToyCity2 scene. We then ran SAAR using various configurations of TG and MF models. Importantly, the only difference between the configurations is the number of initial triangles, all other parameters remained the same. The mean distance between inferred and ground-truth geometry for each of the triangulations can be seen in Fig. 17.

Several observations can be made from Fig. 17. First, independently of initialization, the general trends are all the same, *i.e.*, SAAR always outperforms [1]; and the main improvement in accuracy always occur within the first five geometry updates. Second, as the initialization changes only the initial separation between estimated and inferred change, *i.e.*, as initialization improves the initial separation between estimated and ground-truth decreases. Third, the improvement provided by SAAR decreases as the quality of initialization increases. For example, in the 64K triangulation the geometry improvements is approximately 0.04 meters, while its only 0.015 meters for the 279K initialization. This last point implies that the model is sensitive to initialization, thus producing different results for different initializations.

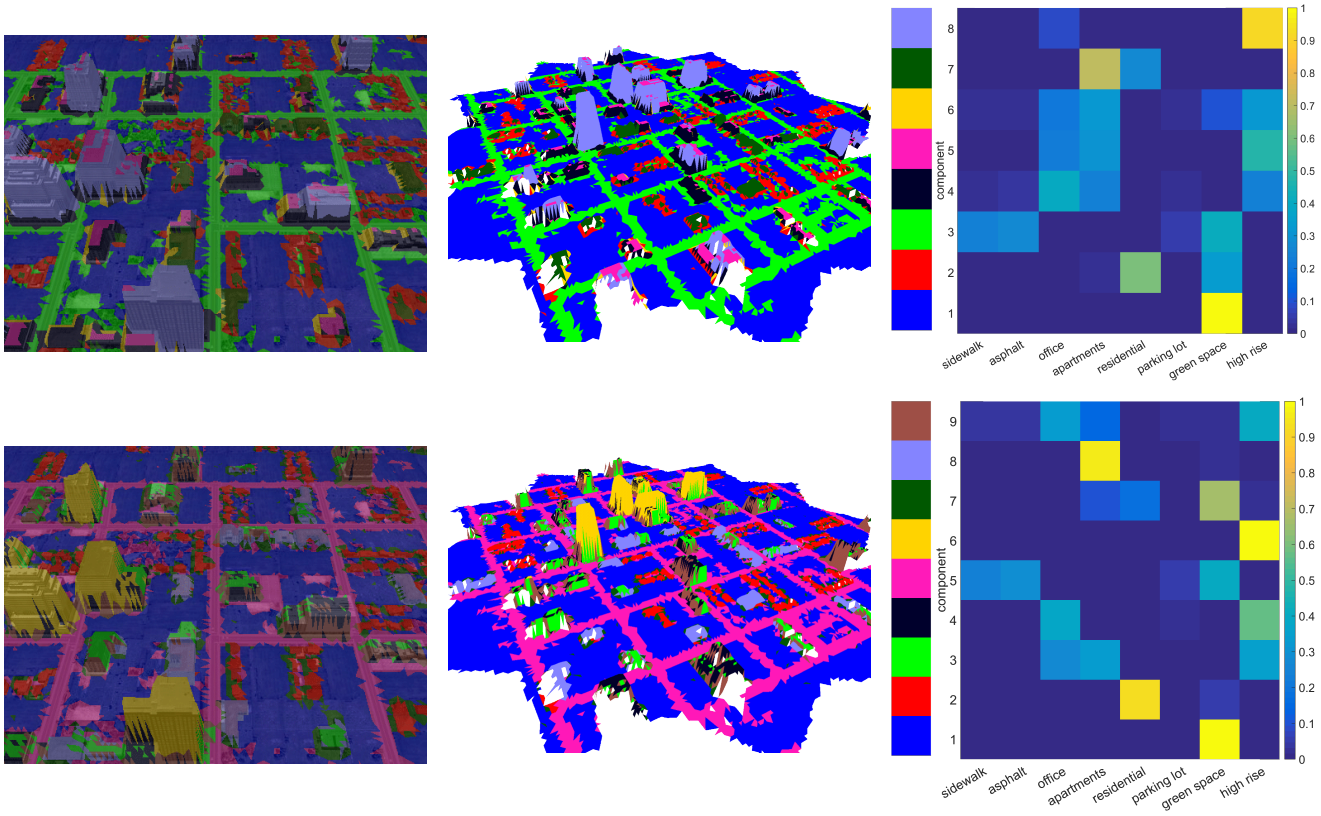

Figure 16: Semantic meaning of clusters for ToyCity2 from SynthCity. *Rows:* different models: TG-8, and TG-9. *Columns:* overlaid cluster assignment on input appearance; 3D scene with cluster assignment; semantic mixture components φ^S. Note that the colorbar in the middle serves as a mapping between cluster number and color representing that cluster.

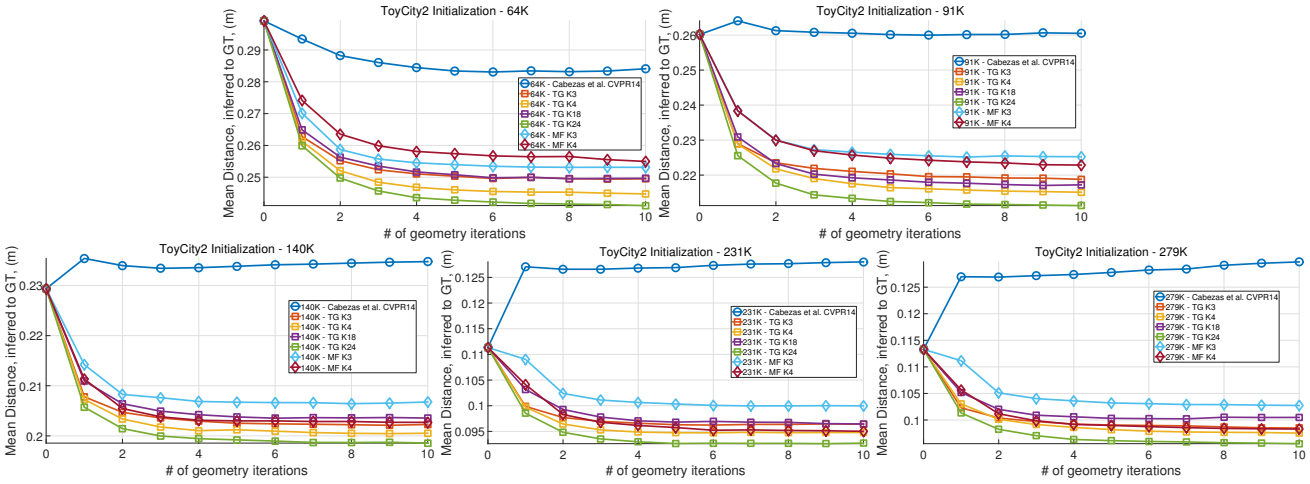

Figure 17: Mean distance between inferred and ground-truth geometry for various ToyCity2 initial configuration.

5. Gibbs Sampler Details

This section details the derivation of the Gibbs sampler posteriors for the structured prior model over geometry, appearance and semantic labels. Specifically, the label posterior (Eq. (7) of the paper) and the parameter posterior (Eq. (8) of the paper) are derived in full detail.

Label posterior: The posterior distribution for a label Z_m of primitive m given the mixture parameters is

$$\begin{aligned} p(Z_m = k|\mathbf{Z}_{\setminus m}, \mathbf{G}, \mathbf{A}, \mathbf{S}, \pi, \varphi^G, \varphi^A, \varphi^S) &\propto p(Z_m, \mathbf{Z}_{\setminus m}, \mathbf{G}, \mathbf{A}, \mathbf{S}, \pi, \varphi^G, \varphi^A, \varphi^S) \\ &= p(\mathbf{Z}_{\setminus m}, \mathbf{G}, \mathbf{A}, \mathbf{S}, \varphi^G, \varphi^A, \varphi^S|Z_m, \pi)p(Z_m|\pi)p(\pi) \\ &= p(\mathbf{Z}_{\setminus m}, \mathbf{G}_{\setminus m}, \mathbf{A}_{\setminus m}, \mathbf{S}_{\setminus m}, |Z_m, \pi)p(A_m, G_m, S_m, \varphi^G, \varphi^A, \varphi^S|Z_m, \pi)p(Z_m|\pi)p(\pi) \\ &\propto p(A_m, G_m, S_m, \varphi^G, \varphi^A, \varphi^S|Z_m)p(Z_m|\pi) \\ &= p(G_m, \varphi^G|Z_m)p(A_m, \varphi^A|Z_m)p(S_m, \varphi^S|Z_m)\pi_k \\ &\propto p(G_m|\varphi^G_k)p(A_m|\varphi^A_k)p(S_m|\varphi^S_k)\pi_k \,. \end{aligned} \tag{1}$$

The steps above are as follows: we first factor the conditional on the right-hand-side to be proportional to the joint probability distribution (on the left-hand-side), from which we condition on the label $Z_m = k$. Subsequently, we can factorize the data term into terms that depend on Z_m and terms that don't in order to further simplify. The last two lines reveal follow from the conditional independencies of the individual components given Z_m and simplify the notation to index the k^{th} cluster directly.

Parameter posteriors: The posterior distributions over the parameters of the mixture model factors into the different data modes as

$$\begin{aligned} p(\pi, \varphi^G, \varphi^A, \varphi^S|\mathbf{Z}, \mathbf{G}, \mathbf{A}, \mathbf{S}) &= p(\pi, \varphi^G, \varphi^A, \varphi^S, \mathbf{Z}, \mathbf{G}, \mathbf{A}, \mathbf{S}) \\ &\propto p(\pi, \varphi^G, \varphi^A, \varphi^S, \mathbf{G}, \mathbf{A}, \mathbf{S}|\mathbf{Z})p(\mathbf{Z}) \\ &= p(\pi|\mathbf{Z})p(\mathbf{A}, \varphi^A|\mathbf{Z})p(\mathbf{G}, \varphi^G|\mathbf{Z})p(\mathbf{S}, \varphi^S|\mathbf{Z})p(\mathbf{Z}) \\ &\propto p(\pi|\mathbf{Z})p(\varphi^A|\mathbf{A}, \mathbf{Z})p(\varphi^G|\mathbf{G}, \mathbf{Z})p(\varphi^S|\mathbf{S}, \mathbf{Z}) \\ &\propto p(\pi|\mathbf{Z}) \prod_{k=1}^{N_S} p(\varphi^A_k|A_{\mathcal{I}_k})p(\varphi^G_k|G_{\mathcal{I}_k})p(\varphi^S_k|S_{\mathcal{I}_k}) \,, \end{aligned} \tag{2}$$

where we use the indicator set $\mathcal{I}_k = \{m : Z_m = k\}$ to collect all data that is assigned to cluster k. The steps above are as follows: we first expand the conditional to be proportional to the joint probability distribution. The model can be simplified by realizing that the individual cluster components are conditionally independent on $\mathbf{Z}$. Using the indicator set $\mathcal{I}_k$ to aggregate the data assigned to cluster k yields the final expression. As stated in the paper, SAAR exploits conjugacy to efficiently sample the posterior parameters after computing the sufficient statistics.

References

[1] R. Cabezas, O. Freifeld, G. Rosman, and J. W. Fisher III. Aerial Reconstructions via Probabilistic Data Fusion. *CVPR*, 2014.

[2] R. Cabezas, J. Straub, and J. W. Fisher III. Semantically-Aware Aerial Reconstruction from Multi-Modal Data. *ICCV*, 2015.

[3] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: measuring error on simplified surfaces. In *Computer Graphics Forum*, volume 17. Wiley Online Library, 1998.

[4] Esri. CityEngine. http://www.esri.com/software/cityengine.

[5] M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE*, 7(4), 2008.

[6] M. E. Hodgson and P. Bresnahan. Accuracy of Airborne Lidar-Derived Elevation : Empirical Assessment and Error Budget. *Photogrammetric Engineering Remote Sensing*, 70(3), 2004.

[7] OSM2World. Osm2world. http://osm2world.org/.

[8] D. Stcker. Josm. http://josm.openstreetmap.de/.

[9] US Air Force. Columbus Large Image Format Dataset 2007. https://www.sdms.afrl.af.mil/index.php?collection=clif2007.