ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Vision
and Geometry Lab

Learning primal-dual network for semantic 3D reconstruction and completion

Semester Project

Hanxue Liang

Robotic, System and Control Msc.

Advisor: Ian Cherabier and Dr. Martin R. Oswald
Supervisor: Prof. Dr. Marc Pollefeys

July 10, 2019

Abstract

We explore the variational regularization method for semantic 3D reconstruction and completion, where we embed Primal-Dual energy minimization scheme into a neural network to learn the prior information between the semantic labels and the 3D geometry. Building on ideas from classical regularization theory and recent advances in deep learning, our network results in a fixed number of unrolled primal-dual optimization iterations, where both the regularizer and the updater operators are learned using a convolutional network with different weights across iterations. The network is end-to-end trainable and is embedded in multi-scale levels, which help to capture more complex semantic dependencies between labels and the geometry.

We present experimental results of such a learned updater and regularizer scheme on real and synthetic datasets, which demonstrates that even though it has more parameters compared with only-learn regularizer framework with shared weights across iterations, our network is able to provide more accurate and more complete reconstructions while requiring less than ten unrolled optimization iterations.

Contents

Chapter 1

Introduction

Understanding 3D environment is very important to many tasks spanning computer vision and computer graphics. In particular, to effectively navigate and interact with an environment, an understanding of the geometry and the semantic information of a scene is essential. Dense 3D reconstruction from images have long been explored, however, ambiguities arising from textureless or reflective regions, viewpoint changes and image noise render this task difficult to solve. In this case, a standard approach to address this problem is to regularize the solutions by introducing prior knowledge during reconstruction, among which, semantics and their interaction with 3D geometry could be highly beneficial. For example, building facades have to be vertical, the ground is often flat and horizontal. Our paper will further explore the method to incorporate semantic information to help improve reconstruction performance.

Due to the availability of reliable semantic image classification algorithm, a lot of progress have been made following the method of joint optimization of geometry and semantics in 3D. In recent years. [3, 4, 5] take depth maps and semantic segmentation images as input to jointly realize volumetric 3D reconstruction and semantic segmentation. [6] uses Wulff shapes as convex anisotropic regularizer to model the relationship between any two neighboring voxel labels. However, the prior used are hand-tuned and very simplistic, failing to fully capture the complex semantic and geometric dependencies of 3D world. [7] uses convolution operator to capture the semantic interactions and embeds variational regularization into a neural network by learning the regularizer function and iteratively optimizes it with shared weights.

Following [7, 8, 9], our work further explores the method of jointly segmenting and reconstructing 3D geometry and combines the advantage of variational approaches[3, 4] and deep learning. We still embed variational regularization into a neural network, resulting a learned iterative reconstruction scheme involving convolutional neural networks in both the reconstruction and data space. Our contributions are as follows:

• Compared with existing variational reconstruction methods [3, 4, 7], we add more flexibility to learned variational regularization framework by generalizing existing learned regularizer with different weights across optimization iterations so that different statistic distribution among la-

bels could be represented. We also learn the forward gradient operator in variational optimization scheme (and data representation) to accelerate optimization procedure. We use different weights across iterations. Even though it increases size of the parameter space, it notably improves reconstruction result and is able to get good reconstruction result in much less optimization iterations, resulting in a more efficient and more powerful model. Besides, formerly required manual and scene-dependent parameter tuning is no longer necessary since all meta-parameters could be learned implicitly.

• We introduce an U-shaped multi-scale optimization strategy, which could accelerate inference, increase the receptive field and allow long-distance information propagation.

• We use our model to handle both semantic 3D reconstruction and completion tasks and create incomplete scenes dataset based on SUNCG data to test our model. This incomplete dataset will be very important for training and evaluating further scene reconstruction and completion models.

Chapter 2

Related Work

There is a vast literature on semantic 3D reconstruction and here we sketch only a small subset of related literature which is closely related to our work.

Semantic 3D Reconstruction

Based on a collection of depth images (or equivalently densely sampled oriented 3D points), the methods proposed in [10, 11] essentially utilize the surface area as regularization prior, and obtain the final surface representation indirectly via volumetric optimization. The difference between them is that [10] employs a combinatorial graph-cut formulation while [11] utilizes a continuously inspired numerical scheme. Given a single RGB-D image, [12] proposes a conditional random field (CRF) for volumetric 3D labeling and solves the CRF using graph cuts. [3, 4] consider object-class specific shape priors and jointly solve 3D voxel space reconstruction and semantic segmentation in a multi-view setting.

All the employed priors utilized in these work are hand-crafted or not rich enough to capture the complex interactions and dependencies in 3D environment. [7] combines the advantage of variational semantic multi-view reconstruction with deep learning in an end-to-end trainable model, in which the regularizer is substituted by a convolution operator and learned from training data. While impressive semantic reconstruction results have been demonstrated, the network has shared parameters and needs about 50 optimization iterations to converge, which makes the whole model limited in its expressiveness and still cumbersome and time-consuming to train. Our work adds more flexibility to learned variational regularization frame-work by learning regularizer and update rule at the same time. Even though it increases size of the parameter space, it notably improves reconstruction result after less than ten optimization iterations, resulting in a more efficient and more powerful model.

Deep Learning in Variational Regularization A common method in solving an inverse problem is to minimize the missfit against data. Since the data usually will be affected by noise and missing part, a regularized missfiting function is constructed to handle it. Variational regularization has proven to be beneficial when dealing with noise and missing data. Many regularization functions have been studied and explored in the literature [13, 14, 15] in different context of computer

vision problems, for instance, [4, 16] show their advantage in 3D surface reconstruction. However, these designed regularizers are not able to fully capture the statistics of the data.

With the advances of deep learning, many works combining the benefit of variational method and deep learning start to emerge. [8] integrates variational method directly into neural networks for depth super-resolution, [17, 18] explore this strategy for image denoising. Vogel et al. [9] combines the structure of regular energy optimization techniques with the flexibility of deep learning to adapt to the statistics of the input data. They learn both the datacost and regularizer for subsequent energy minimization. [19, 20] account for a forward operator in a deep neural network, where in former work a gradient operator has been replaced with convolutional neural networks (CNN), while in latter work, they substitute a proximal operator with CNN. In these works, the optimization steps are unrolled and embedded as layers into a neural network. Our work is inspired by these ideas and tailored to multi-label semantic 3D reconstruction problem.

Chapter 3

Methods

Our method is based on the traditional variational approach to volumetric 3D reconstruction [3, 4] minimizing the energy

$$\min_u \int_\Omega \Big(\underbrace{\phi_x(u)}_{regularization} + \underbrace{\langle f, u \rangle}_{data\ \ fidelity} \Big)\ dx$$

$$\text{subject to } \forall x \in \Omega : \sum\nolimits_l u_l(x) = 1 \tag{1}$$

to find the best regularized labeling function $u : \Omega \to [0,1]^{|L|}$ that assigns each point in space a probability for each label $l \in L$, where L is the set of labels we consider and Ω is the 3D voxel grid. Constraint(1) makes sure that the sum of probability across all labels $l \in L$ at every point $x \in \Omega$ is 1. $f : \Omega \to \mathbb{R}^{|L|}$ is the input to our model and is usually modeled as truncated signed distance function (TSDF). In our model, it not only aggregates the noisy measurement of likely surface location, but also corresponding labelling. Since the input f is affected by noise, outliers and missing data, a regularization term $\phi_x(u)$ is added to the energy function to make the reconstruction problem well-posed.

The simplest choice for regularization is the total variation (TV) norm [21] $\phi_x(u) = \lambda \|\nabla u(x)\|_2$ which is to minimize gradient on the surface area of a 3D shape. The work of [7] generalizes the gradient operator in the regularizer to the general matrix W, so the regularization term is in the form $\phi_x(u) = \|Wu\|_2$. To minimize the Eq.(1), primal dual hybrid gradient (PDHG) algorithm [22] is adopted which is a good way to solve non-smooth convex optimization problem. We first write (1) in its primal dual form, i.e. we introduce the dual variable ξ and replace the TV-norm by its conjugate [22]. We also relax the constraints in Eq.(1) by introducing the Lagrangian variable v. Then the corresponding discretized saddle point problem is written as:

$$\min_u \max_{\|\xi\|_\infty \leqslant 1} E_{saddlepoint} = \min_u \max_{\|\xi\|_\infty \leqslant 1} \langle \nabla u, \xi \rangle + \langle f, u \rangle + v(\sum\nolimits_l u_l - 1) \tag{2}$$

This saddle-point problem can be solved by alternating gradient ascent and descent steps on the primal and dual variables. In particular, we select the algorithm proposed in [22] and minimize the

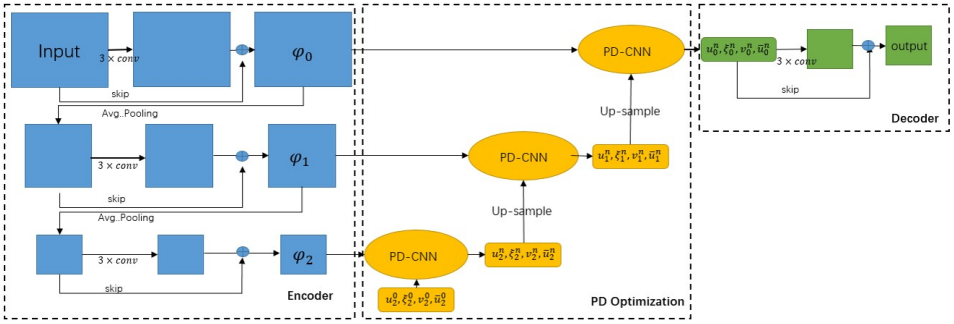

Figure 3.1: Proposed network architecture. Our network is known as U-shaped network. The whole network is composed of three stage: Feature encoding, Primal-Dual optimization and Probability decoding. This graph show an example of multi-scale optimization with three scales, while the optimization steps in each scale should be fixed.

energy with the following update equations:

$$1.v^{t+1} = v^t + \sigma(\sum_l \overline{u}_l^t - 1) \qquad 3.u^{t+1} = \Pi_{[0,1]}[u^t - \tau(\nabla^* \xi^{t+1} + f + v^{t+1})]$$
$$2.\xi^{t+1} = \Pi_{\|\cdot\|\leqslant 1}[\xi^t + \sigma \nabla \overline{u}^t] \qquad 4.\overline{u}^{t+1} = u^{t+1} + \theta(u^{t+1} - u^t)$$

Here $\Pi_{[0,1]}$ denotes the projection into the interval $[0, 1]$ and $\Pi_{\|\cdot\|\leqslant 1}$ denotes projection into space with euclidean norm less than 1, ∇^* is the adjoint of ∇. Dual variable ξ has the same dimension as regularizer term ∇u.

The work [7] transfers the whole primal-dual optimization scheme into a neural network where matrix W is learned from data and is shared across iterations. Our work extends [7] by introducing a U-shaped network which is able to learn multi-scale interactions between labels. We also add more flexibility to the network in several ways by generalizing the gradient operator in primal-dual optimization scheme with neural network and using different interaction weights between iterations. In the following, we will give more detail about our network structure and the way to adding more flexibility to the learned reconstruction scheme inspired by PDHG.

3.1 U-shaped Network structure

Our network structure for semantic 3D reconstruction is depicted in Fig.3.1. Its input consists of a 3D volume of truncated signed distance functions(TSDFs) aggregated by a set of semantically labeled depth maps. More specifically, we follow [4] and utilize depth maps from stereo and corresponding semantic image segmentation to incorporate labelling evidence to TSDF. In the same way as traditional TSDF fusion, we trace rays from every pixel in each depth map to determine which voxels are occupied or empty. However, instead of using a fixed additive cost, we scale it

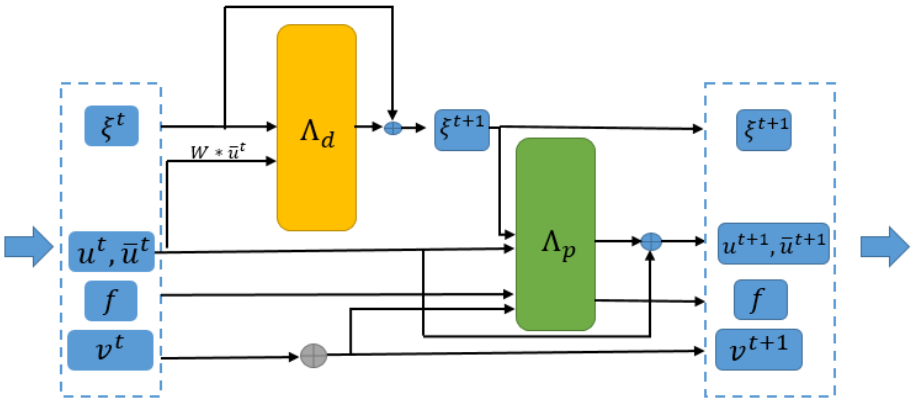

Figure 3.2: This figure illustrates one iteration of primal-dual optimization for our model with learned updater. Orange and green boxes denote convolution operators. They are both composed of two pairs of conv-relu operations followed by a final convolution without activation. Grey circle is update v^t by adding $\sigma(\sum_l \bar{u}_l^t - 1)$ to obtain v^{t+1}

using the semantic score at the corresponding pixel. So this semantic TSDFs should have dimension of $h \times w \times d \times L$, where h, w, d are the height,width and depth of the space and L is the number of labels including empty space. The output of the network is a volumetric semantic 3D reconstruction where each voxel have one of the semantic labels or the empty space label.

Inspired by [7, 9],our network consists of three main stages:

(1) An encoding stage to generate the feature representation

(2) Optimization stage to realize our primal-dual network (PD-CNN)

(3) Decoding stage to get semantic result for each voxel

These stages are embedded into a multi-scale framework. For primal-dual optimization stage, we use U-shaped network, where the principle idea is to work on a low resolution version of the problem first and use the output as the initialization of the next high level. We use simple down- and up-sample operators between levels, average-pooling and nearest-neighbour interpolation. This multi-scale design allows for (i) modeling semantic interactions at different scales (ii) propagating information over larger distance during inference (iii) reconstructing semantic result for each voxel in less optimization iterations. We initialize our variable at the coarsest resolution ($u_L^0, \bar{u}_L^0$ in Fig.3.1) with uniform distribution among the labels.

3.2 Feature Encoding

In the encoding stage, operating from fine to coarse, a feature representation, φ, is obtained from our semantic TSDFs input, and will be used later at the respective resolution in our iterative primal-

dual convolution network (PD-CNN). The encoding stage serves several purposes: first, it helps to reduce low-level noise in the input; second, it extracts feature from the input and gives more flexibility to the network. More specifically, we apply a residual function [23] composed of M pairs of conv-relu operations followed by a final convolution without activation. The result is then added back to the input feature vector. This so called identity skip connection could help to mitigate effects of vanishing gradients and allows to train very deep architecture [23]. The encoded input will then be down-sampled with average pooling to the next stage and be further processed within next encoding stage.

3.3 Learned regularizor

Inspired by the work [7], we first explore the variational optimization approach for semantic 3D reconstruction and by constructing a learned regularizor, we tend to extract semantic interaction feature from the input data and help to provide a smoother and more complete reconstruction result. Algo. 1 and 2 compare our proposed learned-regularizer PD-CNN reconstruction and the original 3D reconstruction method with TV-L1 as regularizer.

Algorithm 1 Primal-Dual algorithm for 3D reconstruction

Given: σ,$\tau > 0$, given f, initialize $u^0, \xi^0, v^0, \overline{u}^0$
for $t = 0$ to $n - 1$ **do**
  $v^{t+1} = v^t + \sigma(\sum_l \overline{u}_l^t - 1)$
  $\xi^{t+1} = \Pi_{\|\cdot\| \leqslant 1}[\xi^t + \sigma \nabla \overline{u}^t]$
  $u^{t+1} = \Pi_{[0,1]}[u^t - \tau(\nabla^* \xi^{t+1} + f + v^{t+1})]$
  $\overline{u}^{t+1} = u^{t+1} + (u^{t+1} - u^t)$
end for
return u^n;

Algorithm 2 Learn Regularizer PD-CNN for 3D reconstruction

Given f, initialize $u^0, \xi^0, v^0, \overline{u}^0$
for $t = 0$ to $n - 1$ **do**
  $v^{t+1} = v^t + s_\sigma^t(\sum_l \overline{u}_l^t - 1)$
  $\xi^{t+1} = \Pi_{\|\cdot\| \leqslant 1}[\xi^t + \mathbf{s}_\sigma^\mathbf{t} \mathbf{W}^\mathbf{t} \overline{u}^t]$
  $u^{t+1} = \Pi_{[0,1]}[u^t - \mathbf{s}_\tau^\mathbf{t}((\mathbf{W}^\mathbf{t})^* \xi^{t+1} + \mathbf{C}^\mathbf{t} \varphi + v^{t+1})]$
  $\overline{u}^{t+1} = u^{t+1} + \theta(u^{t+1} - u^t)$
end for
return u^n;

In our PD-CNN all linear operators in primal-dual optimization are replaced by convolution operators. For instance, the learned regularizer operator W is equivalent to a linear map $\hat{W}$:

$\mathbf{R}^{N^1\times\cdots N^{\mathrm{d}}} \rightarrow \mathbf{R}^{N^1\times\cdots N^{\mathrm{d}}\times d}$, to replace the forward differences encoded by ∇. C is defined in a similar manner. [7] learns a regularizer operator, but we are able to add more flexibility to the network in several ways. First, all the weights in the network is iteration dependent, expressed by superscript n. In [7], they only learn parameters for gradient operator in regularizer and share the same parameters for W among all iterations, but we argue that during the process of iterative optimization, the interactions between labels could have different statistical distribution, so that using different parameters could help to improve the representation capacity of the model. This increases the size of the parameter space but it also notably improves reconstruction result. In our semantic 3D reconstruction network, we use a six dimension matrix $W \in \mathbf{R}^{2\times2\times2\times|L|\times|L|\times3}$, where $2\times2\times2$ encodes forward-backward differences in three spacial dimension, $|L|\times|L|$ encodes the interactions between semantic labels and 3 encodes the gradient in three spatial dimension. When $W=\nabla$, we obtain a standard TV regularizer. Second, in the update of the proximal variable u, our generated feature maps replace our input f, for which we learn a convolution operator C per iteration. Further, instead of explicitly choosing step length σ and τ, which is troublesome in our reconstruction task, we let the network to learn them from data and use different step size(s_σ^t, s_τ^t) per iteration, which could accelerate the process of reconstruction. The last way to add more flexibility to is utilize more channels during the computation than the optimization would suggest. In our semantic 3D reconstruction network, we donate multiple dimensions to ξ and φ, but fix our primal variable u to the same dimension as input data,$i.e.$ $\varphi \in \mathbb{R}^{h\times w\times d\times q}$ and $\xi \in \mathbb{R}^{h\times w\times d\times 3\cdot q}$ for some $q \geqslant |L|$ and $u \in \mathbb{R}^{h\times w\times d\times|L|}$. In this way, we are able to use more channels to represent features encoded in datacost f by $C\varphi$ and improve the representation capacity of capturing the interaction between semantic labels. It should be noted that the dimension of W in regularization term should be adjusted to $W \in \mathbf{R}^{2\times2\times2\times|L|\times3\cdot q}$ accordingly. We note that this learned Primal-Dual algorithm is minimizing (in every step) different energy function.

3.4 Learned updater operator in optimization

Let's reconsider the optimization problem in Eq.(2). In the scheme of primal-dual optimization, the algorithm alternatively applies gradient ascent and descent of saddle point energy $E_{saddlepoint}$ with respect to dual variable ξ and primal variable u ($E_{saddlepoint} = \langle Wu,\xi\rangle + \langle f,u\rangle + v(\sum_l u_l - 1)$ in our problem), that is:

$$\xi^{t+1} = \Pi_{\|\cdot\|\leqslant1}[\xi^t + \sigma W\overline{u}^t] = \Pi_{\|\cdot\|\leqslant1}[\xi^t + \sigma(\nabla_\xi E_{saddlepoint})(\xi^t)] \qquad \text{for } \xi \in T$$
$$u^{t+1} = \Pi_{[0,1]}[u^t - \tau(W^*\xi^{t+1} + f + v^{t+1})] = \Pi_{[0,1]}[u^t - \tau(\nabla_u E_{saddlepoint})(u^t)] \qquad \text{for } u \in U$$

U is $\mathbb{R}^{h\times w\times d\times L}$ and T is $\mathbb{R}^{h\times w\times d\times 3\cdot q}$ in our problem. Since during each iteration, the gradient operator $\nabla_u E_{saddlepoint} : U \rightarrow U; \nabla_\xi E_{saddlepoint} : T \rightarrow T$ requires knowledge from semantic 3D signal ($i.e.u^t$), it is natural to try to learn the update rules from training data while utilizing knowledge about gradient operator $W\overline{u}, Q^*\xi, f, v$ [19, 20]. That is trying to learn an appropriately

selected parameter $\theta_d, \theta_p \in \mathbb{Z}$ such that

$$\Lambda_{\theta_d}(\xi^t, W\overline{u}^t) \approx \xi^t + \nabla_\xi E_{saddlepoint}(\xi^t)$$
$$\Lambda_{\theta_p}(u^t, W^*\xi^{t+1}, f, v^{t+1}) \approx u^t - \nabla_u E_{saddlepoint}(u^t)$$

where $\Lambda_{\theta_d}, \Lambda_{\theta_p}$ is our introduced updating operator. By learning the process of gradient descent from training data, we also expect the optimization scheme could converge in less iterations. So the following modifications to the learned PD-CNN can be done:

• Instead of explicitly enforce the updating of the form $\xi^t + \mathbf{s}_\sigma^\mathbf{n}\mathbf{W}^\mathbf{n}\overline{u}^t$ and $u^t - \mathbf{s}_\tau^\mathbf{t}((\mathbf{W}^\mathbf{t})^*\xi^{t+1} + \mathbf{C}^\mathbf{t}\varphi + v^{t+1})$, we allow the network to learn how to combine the previous update with the result of the operator evaluation. During the updating of primal variable u, we use $\Lambda_{\theta_p}(u^t, W^*\xi^{t+1}, f + v^{t+1})$.

• Instead of enforcing the over-relaxation $\overline{u}^{t+1} = u^{t+1} + \theta(u^{t+1} - u^t)$, we let the network to freely learn in which point the forward operator should be evaluated.

• Instead of using the same updator in each iteration, we allow them to differ. This could again improve reconstruction quality even though increase parameter space.

The above modification result in a new algorithm which is outlined in algorithm 3. To define

Algorithm 3 Learn Updater and Regularizer PD-CNN for 3D reconstruction

1: Given f, initialize $u^0, \xi^0, v^0, \overline{u}^0$
2: **for** $t = 0$ to $n - 1$ **do**
3: $\quad v^{t+1} = v^t + s_\sigma^t(\sum_l \overline{u}_l^t - 1)$
4: $\quad \xi^{t+1} = \Pi_{\|\cdot\| \leqslant 1}[\Lambda_{\theta_d^t}(\xi^t, W^t\overline{u}^t)]$
5: $\quad [u^{t+1}, \overline{u}^{t+1}] = \Pi_{[0,1]}[\Lambda_{\theta_p^t}(u^t, \overline{u}^t, (W^t)^*\xi^{t+1}, C^t\varphi + v^{t+1})]$
6: **end for**
7: **return** u^n;

the class of updator that are parametrised by θ, we follow the work of [19] and use convolution network architectures to learn the updater in Algo.3, where each input is followed by two pairs of conv-relu operations followed by a final convolution without activation. The reason for choosing convolutional neural network is that it has the advantage of being efficient to implement and having much fewer parameters than fully connected network. Meanwhile, our design of multi-level network structure could account for global dependencies. One iteration of primal dual optimization in Algo.3 is depicted in Fig.3.2.

3.5 Probability Decoder

After PD-CNN optimization iterations, we will decode the obtained solution. The main reason here is to increase contrast, enabling stronger decision on final labeling and thereby improve accuracy. The optimized primal variable will be feed into a residual unit with two pairs of conv-relu operation followed by a final convolution with softmax activation for normalization.

3.6 Parameter sharing and training

The primal dual algorithm has no restriction on the size of input data f. And since our network is fully convolutional [24], we can apply the network to arbitrarily sized input data. Even though we have no restriction on input data, the number of iteration in each PD-CNN level and the depth of our multi-scale framework is limited. Since in different levels, the network could extract features from different scales, we use different parameters across different levels. Under each level, we use different weights across iterations. This treatment provides more freedom to the network. For the model of learned regularizer and updater at the same time (Algo. 3), we would use same weights across iterations in each level, since different weights introduce too many parameters and the model becomes difficult to converge. During training, we will specify a loss at each resolution level, penalizing the difference to down-sampled ground-truth. This could help to stabilize the training in its early stages.

Chapter 4

Experiments and Results

Our evaluation is split into two distinct parts. First, we analyze the capability of our network on challenging indoor semantic 3D reconstruction tasks; Second, we validate our approach in semantic 3D completion tasks. In each task, we evaluate and compare the performance of our model with generalized learned regularizer and updater. The model is evaluated on two challenging datasets: ScanNet Dataset and SUNCG Dataset.

Datasets and Training procedure In our experiments, the input 3d scans are represented as multi-label TSDF for datacost aggregation [4] encoded in volumetric grids. We integrate the provided depth maps and semantic segmentation using TSDF fusion based on provided camera poses to establish voxelized groundtruth datacost. The provided datacost encodes strong evidence and will be used as input to our model. The first dataset is recently released ScanNet Dataset [1], comprising 1513 scenes with fine-grain semantic labeling. We adopt the NYU [25] labeling with 40 classes. Using a voxel resolution of 5cm, the largest scenes have a size of around 400^3 voxels. For our evaluation, we generate data costs by integrating every 50th frames. The second dataset is synthetic scans of SUNCG dataset [2] with 38 classes mainly for the task of semantic 3D completion. We use a voxel resoluton of 2.5cm size in case that surfaces of some objects are too thin to be shown in voxelized representation. It should be noted that for SUNCG dataset, after getting the groudtruth datacost for a scene, we will create some empty holes and empty balls on the datacost to get incomplete datacost. For the voxels belonging to the empty hole and balls, datacost can no longer provides any semantic and depth information and we want our model to learn the segmentation result for that voxel from its neighbouring voxels. Since we have the segmentation groundtruth for all voxels, we can use this incomplete datacost to evaluate our model for 3D completion task.

Since our network is fully convolutional, we can optimized arbitrarily sized scenes both during inference and training. While due to the increased memory requirements during back-propagation, we train on fixed-size, random crops of dimension 24^3 with a batch size of 4 and exponential decayed learning rate starting from $5 * 10^{-4}$. We also perform data augmentation by randomly rotating and flipping around the gravity axis. For ScanNet Dataset, We train our model on 231 scenes and test it on 77 scenes. For SUNCG dataset, we train the model on 160 scenes and test on

Table 4.1: Comparison of semantic 3D reconstruction accuracy on Scannet[1]

Method	num. of iterations	freespace	occupied space	semantic accuracy
Input Data	-	39.1	99.7	68.4
TV-L1	50	71.0	91.4	87.8
TV-L1	500	86.4	92.3	88.5
Learned Variational [7]	50	**96.6**	94.4	91.5
Ours				
(share weights across iter. on each level)	9	54.9	58.5	24.3
(share weights across iter. on each level)	24	56.4	59.5	26.5
(share weights across iter. on each level)	45	94.5	97.8	94.8
Ours				
(different weights across iter. on each level)	9	93.6	97.3	95.5
(different weights across iter. on each level, 3× num. of channel)	9	95.1	**98.5**	96.5
(different weights across iter. on each level)	24	94.8	97.6	96.3
(different weights across iter. on each level)	45	95.7	98.1	**96.7**
Ours with learned updater				
(share weights across iter. on each level)	45	94.9	97.3	95.5

35 scenes.

4.1 Semantic 3D reconstruction on ScanNet [1]

We evaluate our model on semantic 3d reconstruction performance on our real-world dataset ScanNet [1]. The input to our model is previously explained multi-label TSDF. The objective is to recover the high fidelity groundtruth given the weak datacosts as input.

Table 4.1 summaries the quantitative results for a reconstruction extracted from the input data cost, a multi-label TV-L1, Learned variational method [7], a variant of algorithm 2 with shared weights across iterations, a variant of algorithm 2 with different weights across iterations, a variant of algorithm 3 with shared weights across iterations. We draw the following conclusion: First, running TV-L1 both for the same number of iterations as our model and for an order of magnitude more iterations will produce worse result than our model; Second, our model with shared weights across iterations gets better reconstruction result than learned variational [7] method, where the difference between them is the design of multi-scale optimization structure. Our U-shaped structure proves its good capacity in capturing semantic dependencies on multi scales by producing better occupied space and semantic accuracy in less number of iterations. Third, our learned regularizer method with different weights across iterations produces the highest accuracy in occupied space and semantic segmentation. It validates our argument that during the process of iterative optimization, different weights could help to further capture the interactions between labels. And also increasing the number of channel for dual variable will also improve the performance. Last, the method of learned regularizer and updater simutaneouly with shared weights across iterations on each level produces better semantic reconstruction result than the model only learning regularizer, which demonstrates the new updater's better reconstruction capacity. We also test with the variant of algorithm 3 with different weighs across weights and find it difficult for the model to converge.

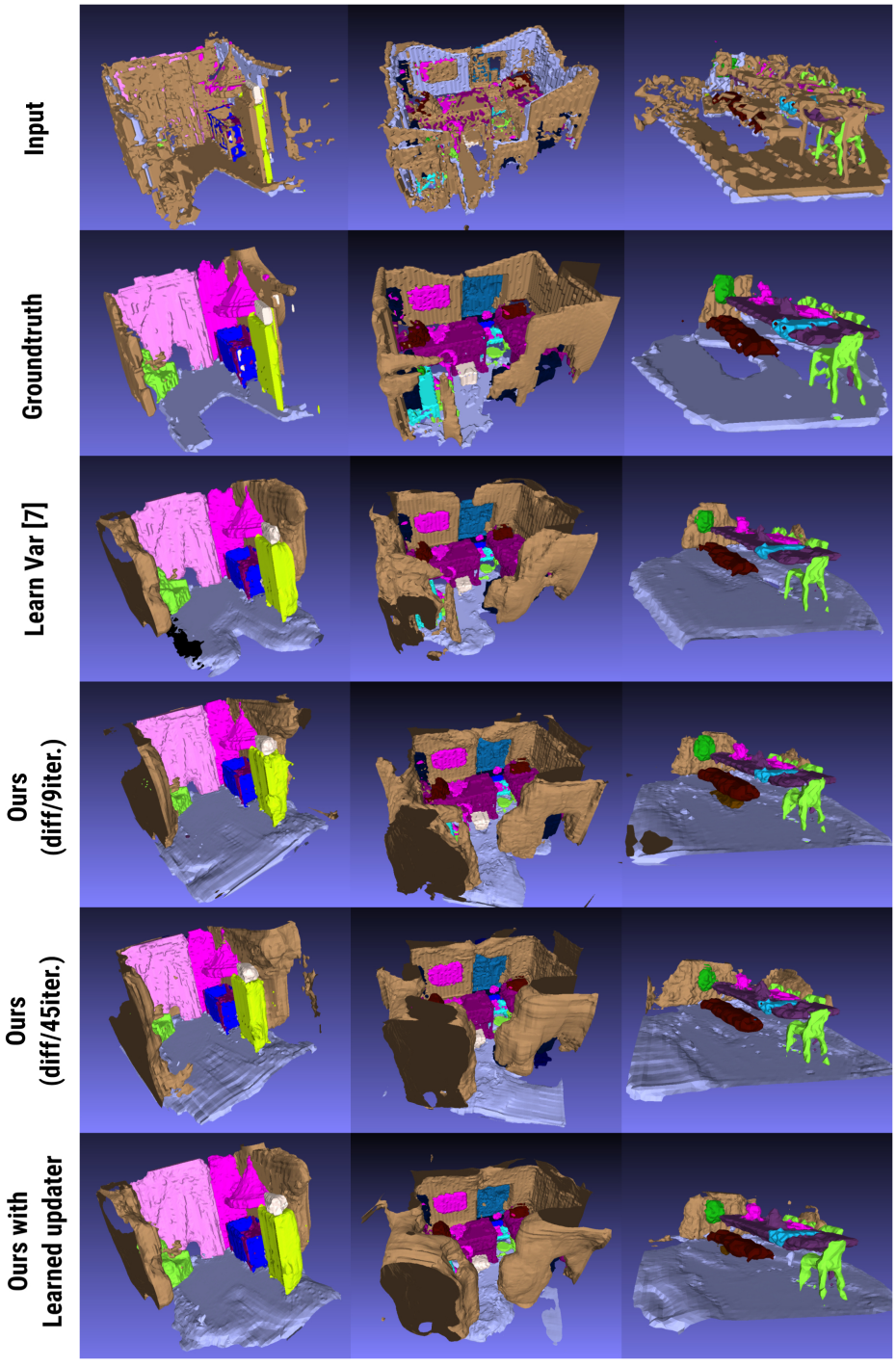

Figure 4.1: Qualitative reconstruction result on ScanNet [1]

Table 4.2: Comparison of semantic 3D completion accuracy on SUNCG[2]

Method	num. of iterations	freespace	occupied space	semantic accuracy
Input Data	-	87.9	26.8	23.0
Learned Variational [7]	50	67.7	88.1	68.0
Ours (share weights across iter. on each level)	45	66.9	92.8	70.8
Ours (different weights across iter. on each level)	9	67.9	89.9	72.6
Ours (different weights across iter. on each level)	45	**72.5**	**94.1**	**73.1**
Ours with Updater (share weights across iter. on each level)	45	67.3	91.5	72.2

The reason is that there are too many parameters in the model and also due to limited number of training scenes, we find it difficult for the model's parameters to converge to a local minimum.

We also compare the influence of different iterative numbers on reconstruction result. For learned regularizer with shared weights, number of iterations have great influence and it should take around 50 iterations for the primal-dual optimization procedure to get good reconstruction result, the same phenomenon happens for learned variational method [7]. Whereas, for our model with different weights across iterations, even though we only iterate for less than ten steps, it could provide a decent reconstruction result. And our model also proves to be much faster during inference task than [7], which is a good advantage for time critical applications. Figure 4.1 shows qualitative results for selected scenes. We could notice that our method provides the result which can be more pleasing and complete than groundtruth for training, especially for the part of floors and walls. We attribute this to the fact that our model could learn labels interactions jointly and is able to apply it to simple instance.

4.2 Semantic 3D completion on SUNCG [2]

We also evaluate our model on semantic 3d completion performance on our synthetic dataset SUNCG [2]. Several reasons make this dataset more challenging than Scannet Dataset [2]. First is that we use only ten frames for each scenes to generate data costs, which means the information we obtained from input is quite limited. And another processing is that the input to our model is previously explained incomplete multi-label TSDF, where we create some empty holes and empty balls on the datacost to test our completion results. The object is to reconstruct the semantic information for voxels where we have no datacost information by inferring from its neighbours' datacost. Table 4.2 shows the quantitative result for SUNCG completion and qualitative result is shown in Fig. 4.2.

First, even though given the input data with only around 25 percent occupied and semantic accuracy, our model could provide more complete reconstruction result by improving occupied accuracy to around 90 percent and semantic accuracy to more than 70 percent due to its good in-

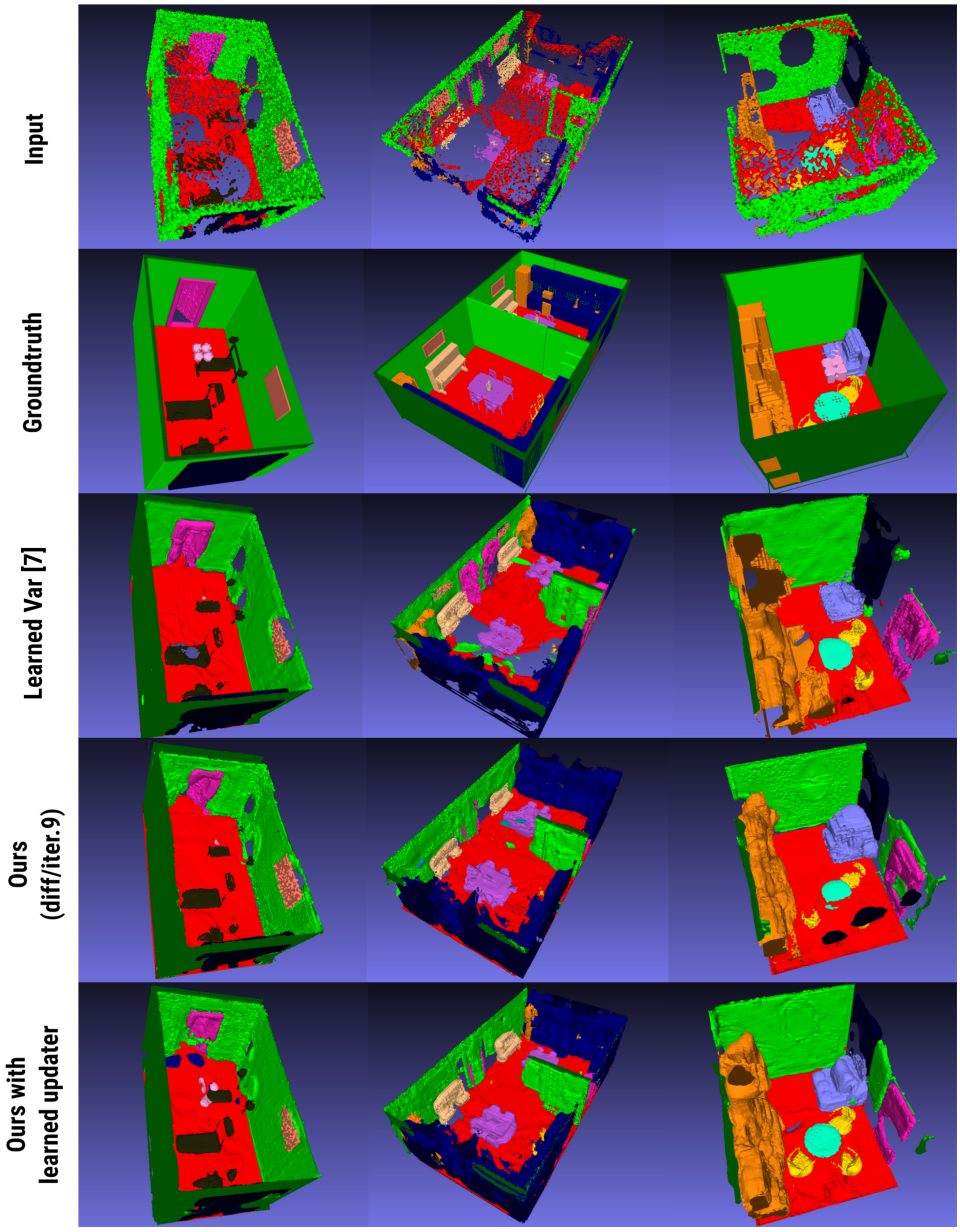

Figure 4.2: Qualitative reconstruction result on SUNCG [2]

ference ability. Second, compared with Learned Variational [7], our model with shared weights across iterations again demonstrates its good capacity in capturing semantic dependencies on multi scales by providing better reconstruction results in terms of occupied and semantic accuracy. Third, when we use different weights across iterations, the model can give reasonable construction performance with less than ten iterations. However, when we iterate ten iterations on learned variational [7] model with shared weights across iterations, the reconstruction performance is worse and the whole primal-dual optimization scheme fail to provide reasonable result within such limited iteration times. We note that different weights across iterations could not only capture more interactive relations between labels, but can also help to give accelerations on reconstructions. Furthermore, same as what happens to Scannet Dataset [1], our model with different weights across iterations shows the highest occupied and semantic accuracy. Finally, compared with our model of learned regularizer with shared weights, our model with learned regularizer and updater is able to provide better freespace accuracy and semantic accuracy, so the new updating method do help to improve the reconstruction performance. Observing the qualitative result in fig 4.2, we find that our models are able to complete the empty space of the walls and floors in input data. For our model with different weights across iterations, optimizing only 9 iterations is able to provide a more complete and accurate scene than [7].

Chapter 5

Conclusion

Building on ideas from classical regularization theory and recent advances in deep learning, our network results in a fixed number of unrolled primal-dual optimization iterations, where both the regularizer and the updater component are learned using a convolutional network. Compared with [7], several modifications are made and help to improve reconstruction performance and efficiency: first, we introduce U-shaped structure to the model, which is able to capture interactive relations between labels and geometry on different scales; second, using different weights across iterations helps to represent different statistic distribution among labels during optimization and accelerate primal-dual optimization procedure to get good result with less than ten iterations, resulting in a more powerful and efficient model; third, by learning updater operators, our model could accelerate optimization procedure and produce more accurate and complete reconstruction and completion results. We test our models on two challenging datasets: Scannet Dataset[1] and Suncg dataset [2], and they demonstrate good potential in both 3D reconstruction and 3D completion tasks by providing more smooth and more complete result given noisy and incomplete input data costs.

However, there are still several limitations to our method: The first is that for our model with learned updater, if we use different weights across iterations, there are too many parameters in the model which makes the model difficult to converge or very sensitive to converge to local minimum, resulting a bad reconstruction result. Other methods directly working on data fidelity function with fewer parameters might also work to extend this term to more complex function space. Second limitation is that since the input of our model is a semantic TSDF fused with a set of depth maps with corresponding 2D semantic segmentation, our model depends on the quality of 2D segmentation result.

Bibliography

[1] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. pages 5828–5839, 2017.

[2] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. pages 1746–1754, 2017.

[3] Christian Häne, Christopher Zach, Andrea Cohen, and Marc Pollefeys. Dense semantic 3d reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1730–1743, 2016.

[4] Christian Hane, Christopher Zach, Andrea Cohen, Roland Angst, and Marc Pollefeys. Joint 3d scene reconstruction and class segmentation. pages 97–104, 2013.

[5] Christian Hane, Nikolay Savinov, and Marc Pollefeys. Class specific 3d object shape priors using surface normals. pages 652–659, 2014.

[6] Selim Esedolu and Stanley J Osher. Decomposition of images by the anisotropic rudin-osher-fatemi model. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(12):1609–1626, 2004.

[7] Ian Cherabier, Johannes L Schonberger, Martin R Oswald, Marc Pollefeys, and Andreas Geiger. Learning priors for semantic 3d reconstruction. pages 314–330, 2018.

[8] Gernot Riegler, Matthias Rüther, and Horst Bischof. Atgv-net: Accurate depth super-resolution. pages 268–284, 2016.

[9] Christoph Vogel and Thomas Pock. A primal dual network for low-level vision problems. pages 189–202, 2017.

[10] Victor Lempitsky and Yuri Boykov. Global optimization for shape fitting. pages 1–8, 2007.

[11] Christopher Zach, Thomas Pock, and Horst Bischof. A globally optimal algorithm for robust tv-l 1 range image integration. pages 1–8, 2007.

[12] Byung-soo Kim, Pushmeet Kohli, and Silvio Savarese. 3d scene understanding by voxel-crf. pages 1425–1432, 2013.

[13] Selim Esedolu and Stanley J Osher. Decomposition of images by the anisotropic rudin-osher-fatemi model. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(12):1609–1626, 2004.

[14] Carl Olsson, Martin Byröd, Niels Chr Overgaard, and Fredrik Kahl. Extending continuous cuts: Anisotropic metrics and expansion moves. pages 405–412, 2009.

[15] Christopher Zach, Liang Shan, and Marc Niethammer. Globally optimal finsler active contours. pages 552–561, 2009.

[16] Kalin Kolev, Maria Klodt, Thomas Brox, and Daniel Cremers. Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 84(1):80–96, 2009.

[17] Erich Kobler, Teresa Klatzer, Kerstin Hammernik, and Thomas Pock. Variational networks: connecting variational methods and deep learning. pages 281–293, 2017.

[18] Tim Meinhardt, Michael Moller, Caner Hazirbas, and Daniel Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. pages 1781–1790, 2017.

[19] Jonas Adler and Ozan Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017.

[20] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.

[21] Tony F Chan, Selim Esedoglu, and Mila Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM journal on applied mathematics*, 66(5):1632–1648, 2006.

[22] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 2016.

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. pages 3431–3440, 2015.

[25] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. pages 746–760, 2012.