What Synthesis Is Missing: Depth Adaptation Integrated With Weak Supervision for Indoor Scene Parsing

Supplementary Material

Keng-Chi Liu, Yi-Ting Shen, Jan P. Klopp, Liang-Gee Chen
National Taiwan University
{calvin89029,dennis45677,kloppjp}@gmail.com, lgchen@ntu.edu.tw

Contents

1. Algorithm

Algorithm 1 gives the formal specification for the second integration step described in our paper.

Algorithm 1: Second Integration Step.

Result: Pseudo Labels y_{Pseudo}
$O_{\text{Scene Bounds}} = \{\text{Ceil}, \text{Floor}, \text{Wall}\}$
$O_{\text{Small}} = \{\text{Books}, \text{Paint}\}$
foreach *Category* $c \in C$ **do**
  $A_c = \sum_{\text{Pixel i}} \hat{y}_{\text{CAM},i,c} > \tau_{\text{CAM}}$
end
foreach *Contour* $\gamma_k^* \in \Gamma^*$ **do**
  // Add Maximum Likelihood candidate $P_k = \left\{\arg\max_{c \in C} \sum_{i \in \gamma_k^*} \hat{y}_{\text{CAM},c,i}\right\} \cup O_{\text{Small}}$
  // Compute confidence features
  $E_k = \emptyset$
  foreach *Category* $c \in P_k$ **do**
    $p_{k,c} = \max_{i \in \gamma_k^*} \hat{y}_{\text{CAM},i,c}$
    $r_{k,c} = \frac{\sum_{i \in \gamma_k^*} \hat{y}_{\text{CAM},i,c} > \tau_{\text{CAM}}}{\#\gamma_k^*}$
    // Check if thresholds are met
    switch $y_{\textit{Step 1},k}$ **do**
      case *is "Unknown"* **do**
        $\tau_p = \tau_{p,\text{Unknown}}$
        $\tau_r = \tau_{r,\text{Unknown}}$
      end
      case *is in* $O_{\textit{Scene Bounds}}$ **do**
        $\tau_p = \tau_{p,\text{Scene Bounds}}$
        $\tau_r = \tau_{r,\text{Scene Bounds}}$
      end
      otherwise do
        $\tau_p = \tau_{p,\text{Other}}$
        $\tau_r =, \tau_{r,\text{Other}}$
      end
    end
    if $p_{k,c} > \tau_p$ *and* $r_{k,c} > \tau_r$ **then**
      $E_k = E_k \cup \{c\}$
    end
  end
  if $E_k = \emptyset$ **then**
    $\forall i \in \gamma_k^* : y_{\text{Pseudo},i} = y_{\text{Step 1},k}$
  else
    $\forall i \in \gamma_k^* : y_{\text{Pseudo},i} = \arg\min_{c \in E_k} A_c$
  end
end

2. Unsupervised Depth Restoration and RANSAC Scale Recovery for Depth Adaptation

2.1. RANSAC Scale Recovery

During depth adaptation, to simulate sensor noise correctly, we apply min-max normalization to the depth map. Adaptation results and comparisons are shown in Fig.1.

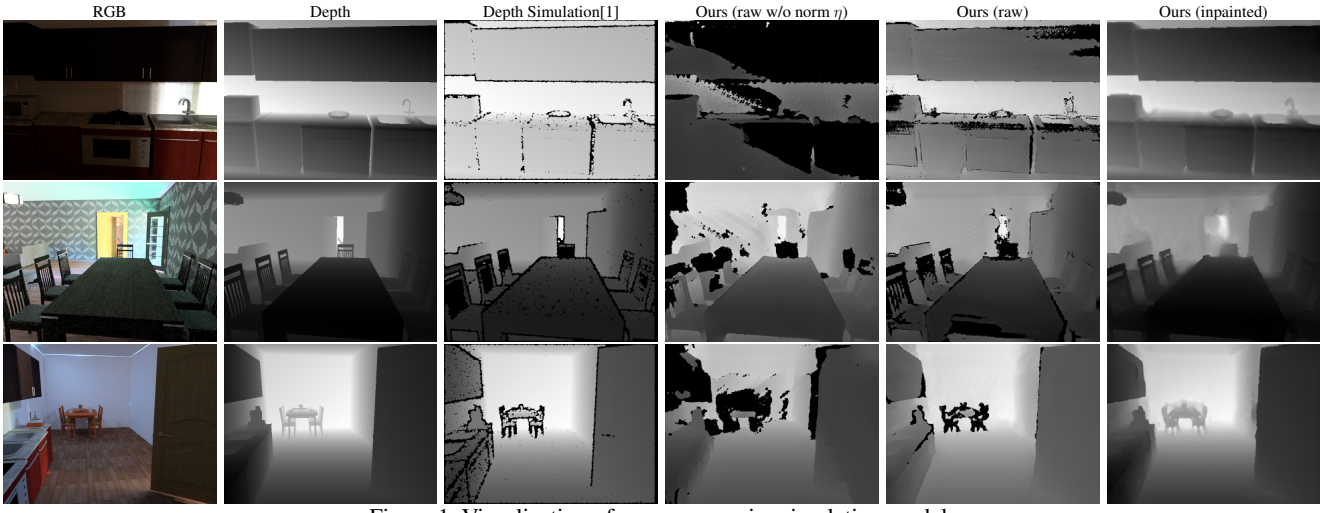


Figure 1. Visualization of our sensor noise simulation model.

The resulting depth map is correct up to an unknown scaling factor. To restore the depth map completely, besides utilizing the restoration model R which is able to transfer sensor depth map from the real to the synthetic, clean domain, performing tasks such as hole filling and denoising, scale recovery is indispensable. Scale is the relationship between different distances in the map and how these distances relate to the real world. By intuition, we can recover scale by simply aligning $[-1, 1]$ with original minimum and maximum in depth map. However, this way we would take only two extreme values, the closest and the farthest point, to estimate the scale. This introduces a large variance, hence we propose to align restored normalized depth prediction with sensor depth map via random sample consensus (RANSAC) [2] under the assumption that most values in the sensor depth map are correct. After pairing each normalized value in $[-1, 1]$ with the absolute value at the corresponding spatial position, RANSAC is performed to establish a relationship between the map units and real distance by estimating the best scaling factor γ and bias β that fit the images as shown in Fig.2 :

$$\gamma, \beta = RANSAC(R(\eta(x_{Real,D})), x_{Real,D}) \tag{1}$$

$$x_{Restore,D} = \gamma \times R(\eta(x_{Real,D})) + \beta. \tag{2}$$

2.2. Dataset

Since aligned sensor depth and ground truths are rare and expensive, we evaluate our model on a synthetic noise simulation dataset. The ICL-NUIM dataset [5] is a benchmark for the evaluation of visual odometry, 3D reconstruction and SLAM algorithms. It is a collection of handheld RGB-D camera sequences within synthetically generated environments. Care has been taken to simulate typically observed real-world artefacts in the synthetic imagery by modelling sensor noise in both RGB and depth data [5]. While this dataset is designed for the tasks mentioned above, we utilize it to evaluate the performance of depth restoration. We sample one image per 10 frames from the "Living Room lr kt2" set (89 frames in total) as our evaluation set.

2.3. Evaluation Matrix

To evaluate the performance of our restoration, we follow related works and compare the results in terms of root-mean-squared error (RMSE). RMSE between the computed depth map $x_{Restore,D}(u, v)$ and the ground truths map $x_{Syn,D}(u, v)$ is defined as:

$$RMSE = (\frac{1}{N} \sum_{u,v} |x_{Restore,D}(u, v) - x_{Syn,D}(u, v)|^2)^{\frac{1}{2}}. \tag{3}$$

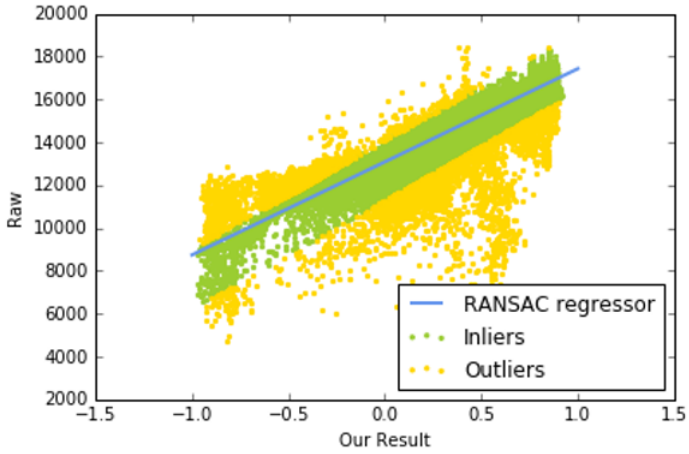


Figure 2. Scale alignment with RANSAC.

Additionally, $bad\,\tau$ indicates percentage of pixel whose absolute difference exceeds τ:

$$bad\,\tau = \frac{1}{N} \times \sum_{u,v} (|x_{Restore,D}(u,v) - x_{Syn,D}(u,v)| \geq \tau). \tag{4}$$

2.4. Ablation Study

Table 1 presents several ablation studies to show the advantages of our alignment and training method. Training the model without min-max normalization performs worse as discussed.

Table 1. Ablation studies for our alignment and training method. These results were obtained using the sampled 89 frames from the ICL-NUIM dataset.

	Model Normalization	Alignment	RMSE	Bad 2.5	Bad 5	Bad 10
Our	Yes	Minmax	10.57	72.16	53.19	29.09
Ours	No	RANSAC	3.69	23.59	7.52	2.16
Ours (Final)	**Yes**	**RANSAC**	**2.89**	**13.29**	**3.94**	**1.57**

2.5. Results

In this section, we compare our proposed method to other state-of-the-art depth restoration methods, i.e., guided filter [6], the joint static and dynamic filtering (SDF) method [4], mutual-structure filtering [10] and dynamic guidance learning [3]. Moreover, all depth maps are quantized from [0,65535] to [0,255], since some related works only accepted 8-bit depth maps. For a fair comparison, all methods are evaluated without tuning parameters on the ICL-NUIM dataset, kernel sizes for related works are adjusted to yield best results. We report results for quantizing after as well as before restoration. Quantization to $[0, 255]$ in Table 2 refers to applying quantization before restoration, i.e., process an 8-bit depth map, or applying restoration directly to the 16-bit raw depth map and quantize afterward. Obviously, the order of quantization and restoration has only effect. All results are reported in terms of RMSE. Fig. 3 gives visual examples of the restoration results on the ICL-NIUM dataset and Fig. 4 highlights where bad pixels occur.

Our approach outperforms state-of-the-art restoration methods on the ICL-NIUM dataset because we are able to restore depth map especially some large holes in consideration of the sensor noise model and process images using a larger receptive field. Most state-of-the-art approaches are not able to fill large holes in the data, leading to a higher bad pixel percentage for large τ and thereby to a higher RMSE. We have adjusted the kernel sizes of those traditional approaches to enlarge their receptive fields, however they were not designed for that purpose and are obviously not as powerful as neural networks. Nevertheless, one of the main issues for our method is that the shapes of objects and flat surfaces may sometimes be deformed

Table 2. Comparison of state-of-the-art depth restoration methods. These results were obtained using the sampled 89 frames on ICL-NUIM dataset.

	Quantization to [0,255] before	after	RMSE	Bad 2.5	Bad 5	Bad 10
Raw			5.19	7.61	5.72	3.75
SDF[4]	√		4.66	8.03	4.47	2.91
SDF[4]		√	4.63	7.40	4.33	2.86
Dynamic Guidance[3]	√		4.36	8.85	4.80	2.78
Mutual-Structure[10] (medfilter r=7)	√		3.89	7.77	3.54	2.01
Guided Filter[6] (r=15)	√		3.17	20.35	8.24	2.14
Guided Filter[6] (r=15)		√	3.19	20.33	7.03	1.92
Ours (Final)	√		2.92	11.84	4.16	1.60
Ours (Final)		√	**2.89**	13.29	3.91	1.57

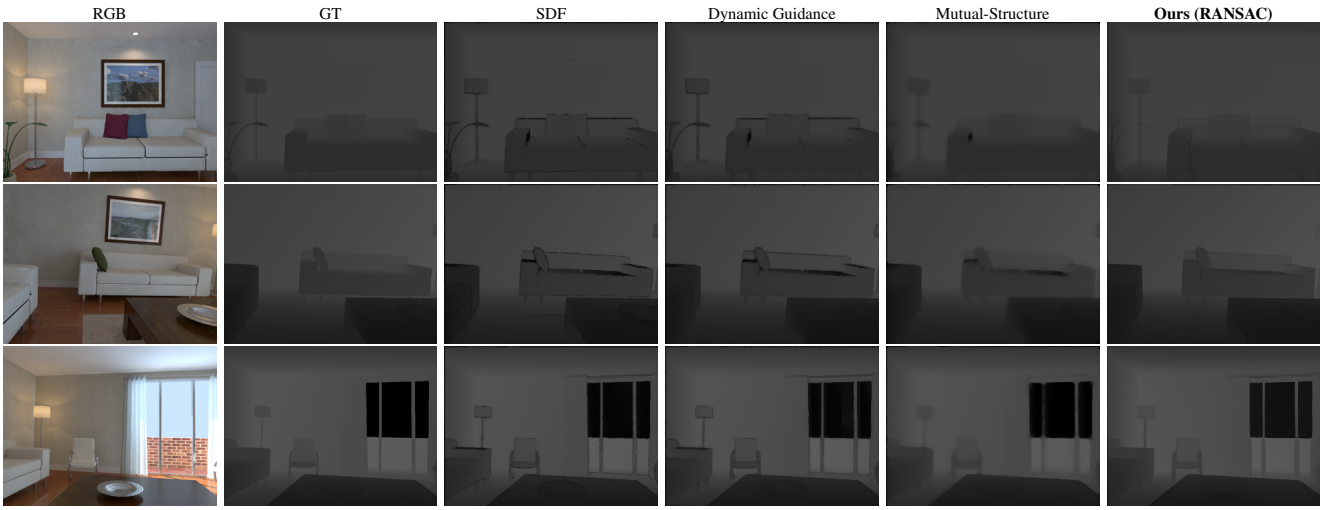


Figure 3. Visualization of state-of-the-art depth restoration methods.

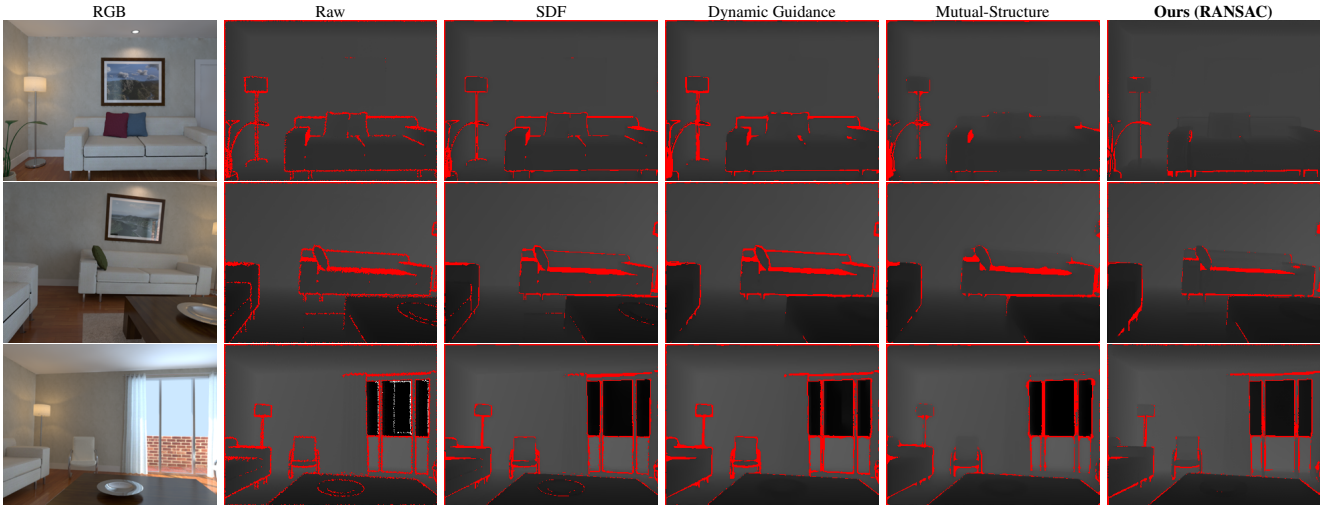


Figure 4. Bad pixel vizualization of state-of-the-art depth restoration methods given $\tau = 5$.

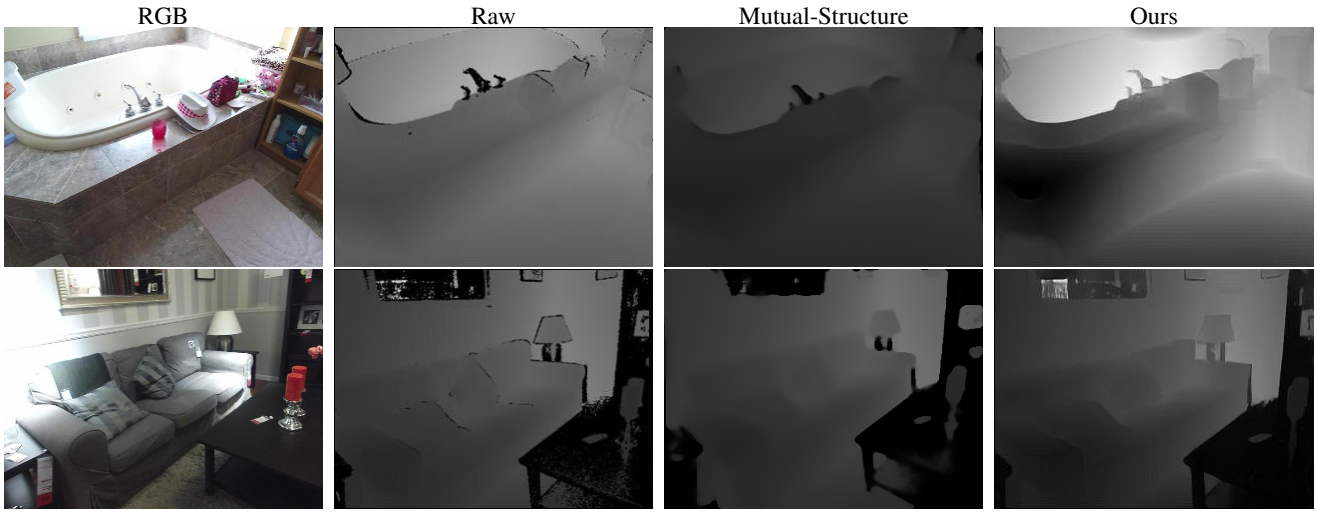


Figure 5. Depth restoration on real dataset (without scale alignment yet). It is shown that our approach is more capable of handling large hole filling.

as our method was trained without paired image translation or guidance by RGB images. Hence, worse bad pixel percentage is obtained when τ is set to a small value. Fig.5 shows our restoration result on real data from SUN RGB-D dataset [11].

3. Confidence Threshold for Adapted Scene Parsing Network

As mentioned in the proposed integration mechanism, to remove results with low confidence from the synthetic-to-real adaptation in the first integration step, we apply a Softmax and a threshold $\tau_{Adapted}$ to the output of $SP_{ada}(X_{Real,D})$. Fig.6 supports the intuition that global accuracy (GA) and mIoU are positively correlated with Softmax's confidence. Since small thresholds lead to imprecise results and large threshold results in a low coverage ratio, $\tau_{Adapted} = 0.6$ was chosen by experiment. After adopting a winner-take-all mechanism based on the histogram of categories within contours to correct wrong or uncertain regions, the high confidence result may effectively act as pseudo-ground truth within our learning process.

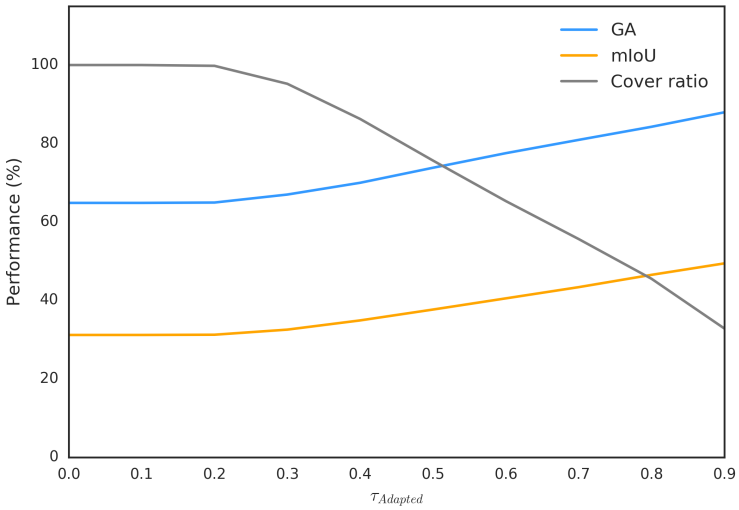


Figure 6. Cover ratio versus Performance for Adapted Scene Parsing Model SP_{ada} over different confidence thresholds $\tau_{Adapted}$. Global Accuracy (GA) and mIoU are evaluated only over those data points that exceed the given threshold.

4. More Visualizations of our Proposed Method

Visualizations of the integration mechanism are shown in Fig.7. In addition, more comparison examples are shown in Fig. 8, 9, and 10. Lastly, we have claimed that the performance of our proposed method may be underestimated due to some ground truth labeling errors. Examples supporting this are shown in Fig. 11.

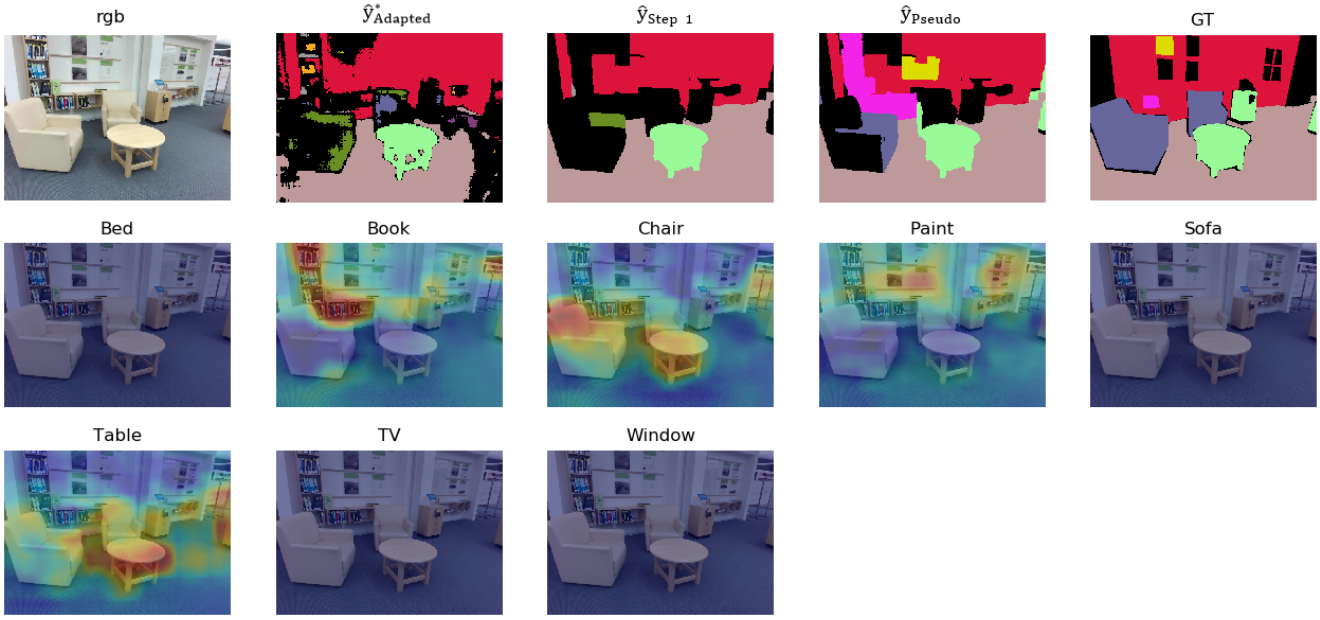


Figure 7. Visualization of each stage of the proposed integration mechanism along with heat maps for several categories. Note that the background for ground truth (GT) is incorrectly annotated as it's missing most parts of the book shelf, while our own method captures those parts.

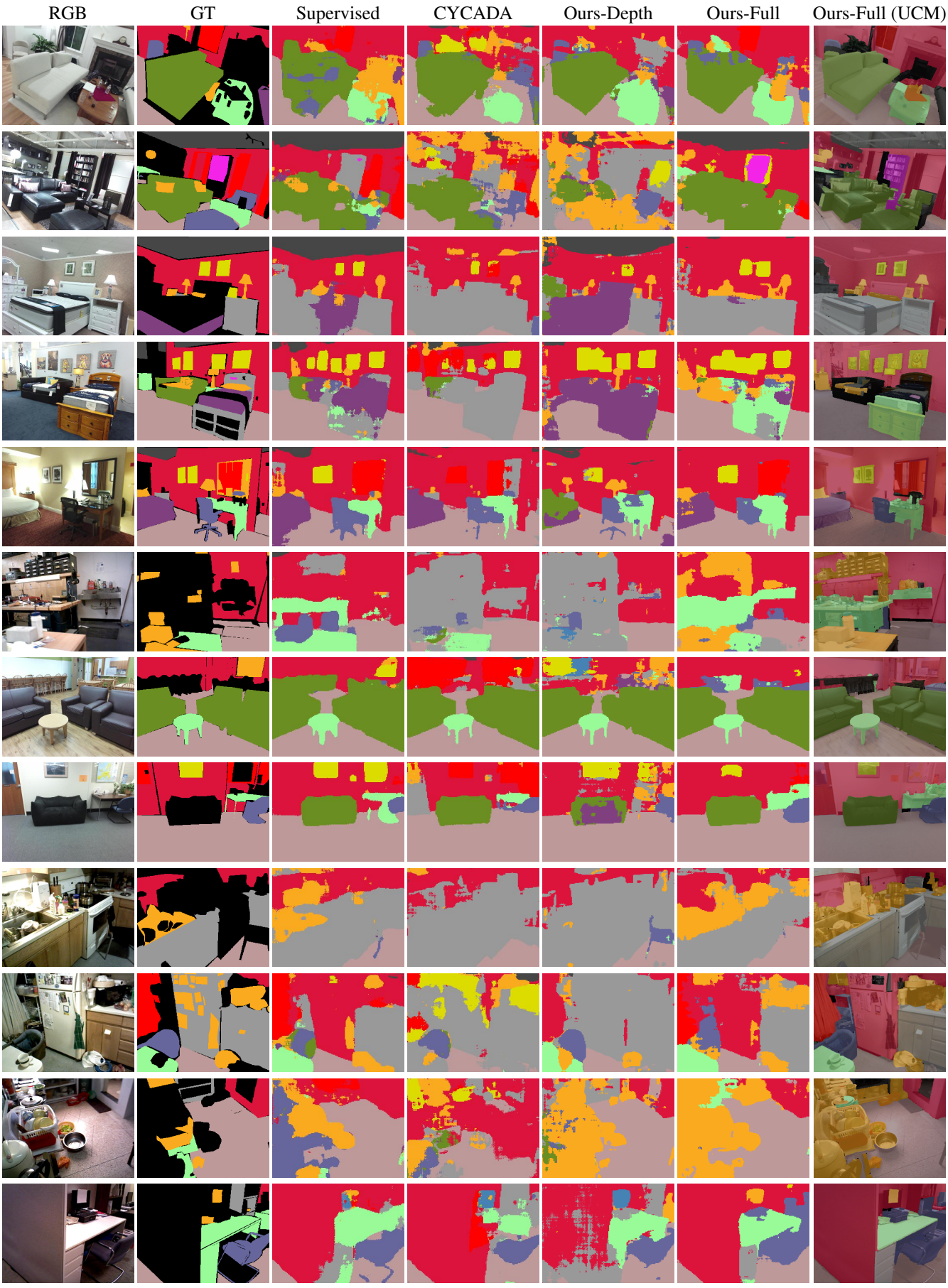


Figure 8. Visualization of results of our proposed method and comparisons to supervised and other transfer learning approaches. Last column is shown by overlaying RGB images with our predictions.

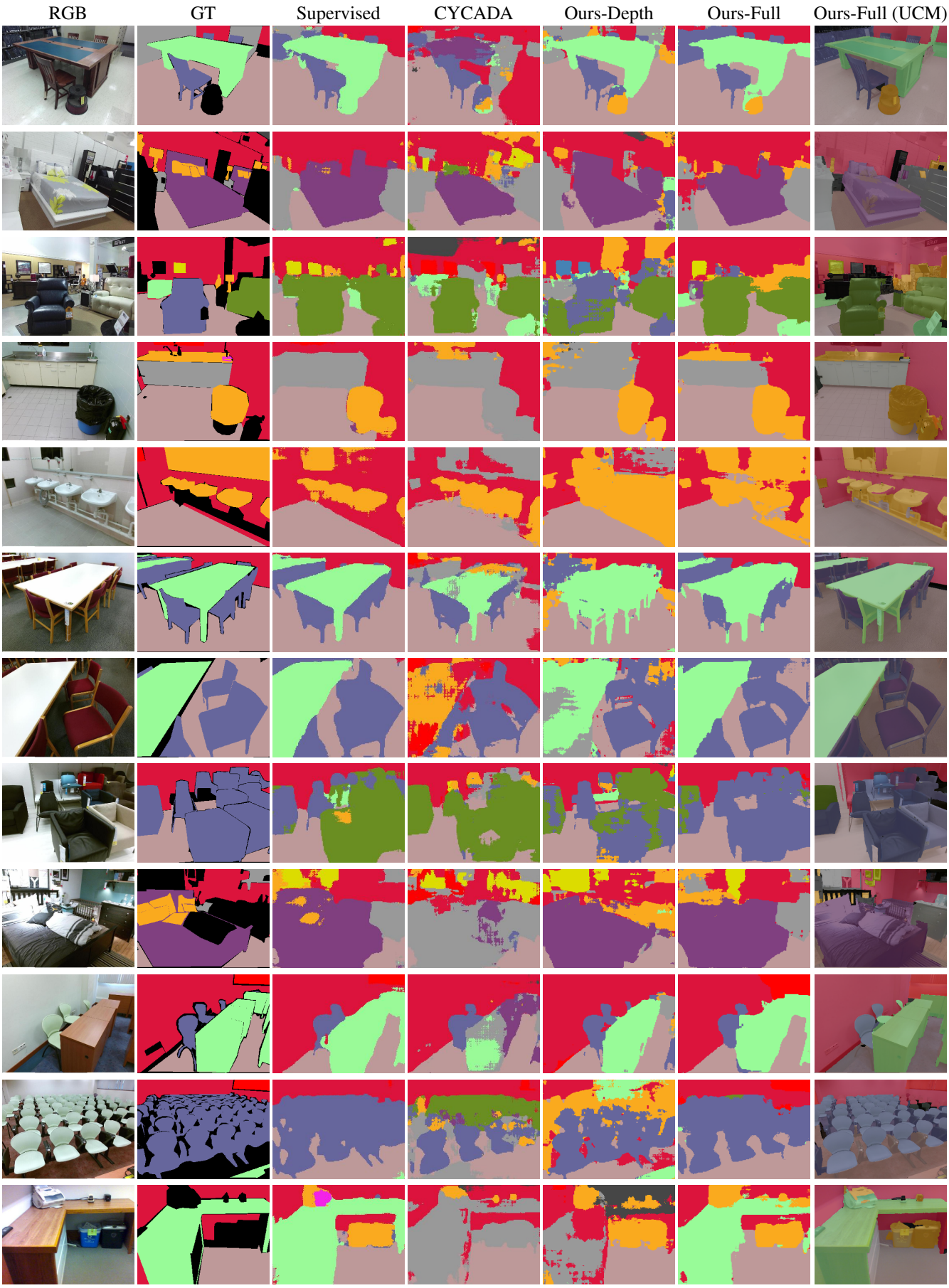


Figure 9. Visualization of results of our proposed method and comparisons to supervised and other transfer learning approaches. Last column is shown by overlaying RGB images with our predictions.

Figure 10. Visualization of results of our proposed method and comparisons to supervised and other transfer learning approaches. Last column is shown by overlaying RGB images with our predictions.

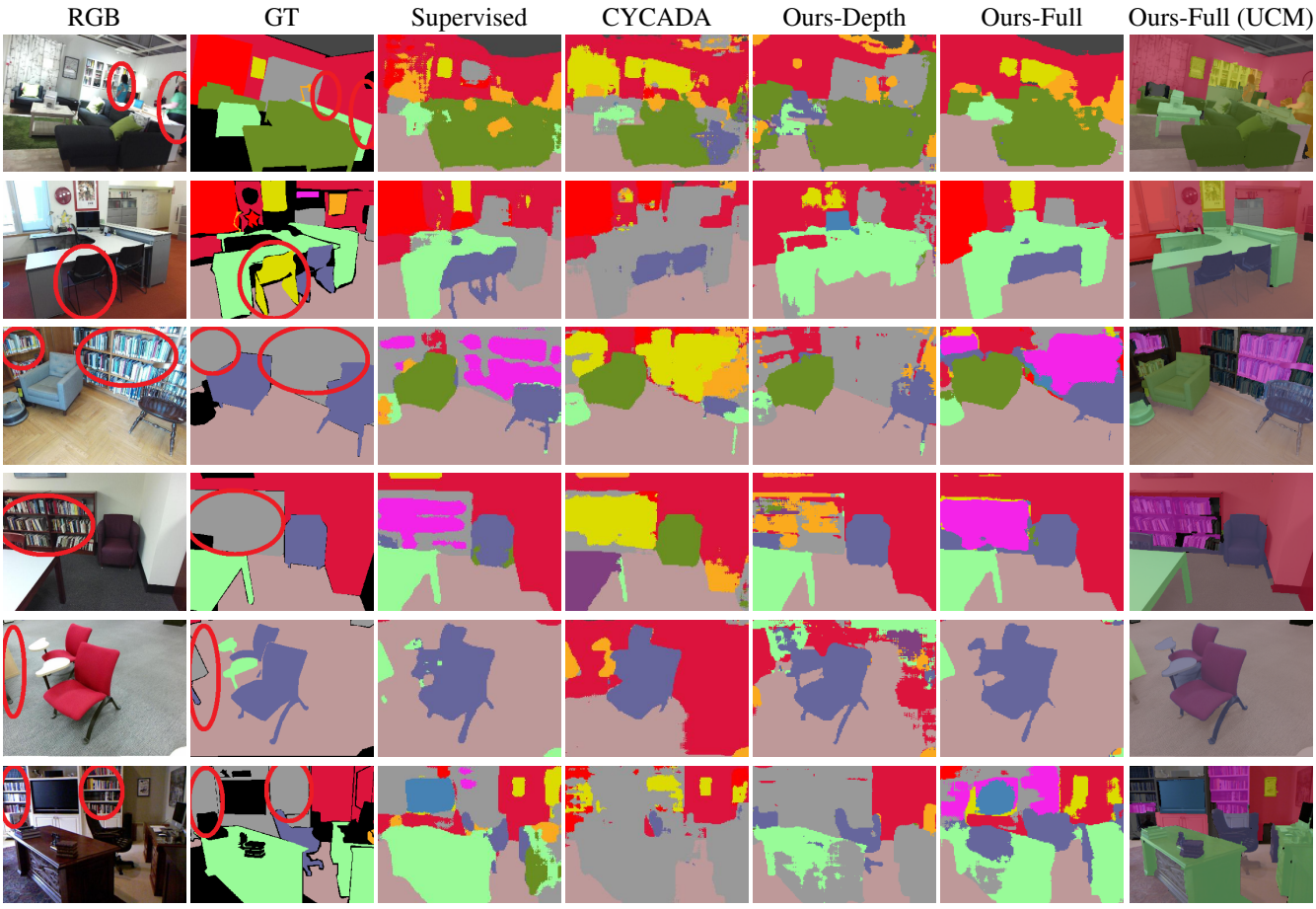


Figure 11. Examples with incorrect or missing ground truth. Last column is shown by overlaying RGB images with our predictions.

5. Computational Complexity Reduction

Reducing computational complexity is important to facilitate mobile scene parsing applications, for example for autonomous agents. We apply weight and activation quantization to our network to evaluate its applicability in environments with little computational resources.

5.1. Weight Quantization

In order to evaluate the impact of low bit-width allocations on SUN RGB-D dataset, we implement different quantization methods originally developed for classification. Furthermore, the bit-width decay training procedure from [12] (based on [13]) for the scene parsing task is implemented and compared. In the first step, activation is held at full precision and weight quantization is explored, i.e., $(W, A) = (k_w, 32)$. All models are fine-tuned from full precision models and trained with same hyperparameters. The results are reported in Fig. 12. We observe that the performance of linear quantization degenerates significantly with decreasing bit-width, which corresponds to our intuition. While the 8-bit model performs almost the same as the full-precision model, decreasing bit-width from 4-bit to less incurs a sharp performance drop. Binarized methods including BNN [7], XNOR [9], DoReFa [13] on the other hand reduce this drop to 3.4% of mIoU when compared to full precision model. We chose to use ternary quantization to further decrease the performance loss. Extending from TTQ [8] yielded an mIOU of 41.73%, achieving almost the same performance as TTQ with only a single scaling factor and 1.4% of performance loss compared to the 8-bit model. $TTQ(1st\ layer)$ indicates that ternarizing all layers inclusive of the first layer results in degradation. [13] and [12] achieve mIOU of 40.19% and 41.07% respectively, which further discussion will be given later.

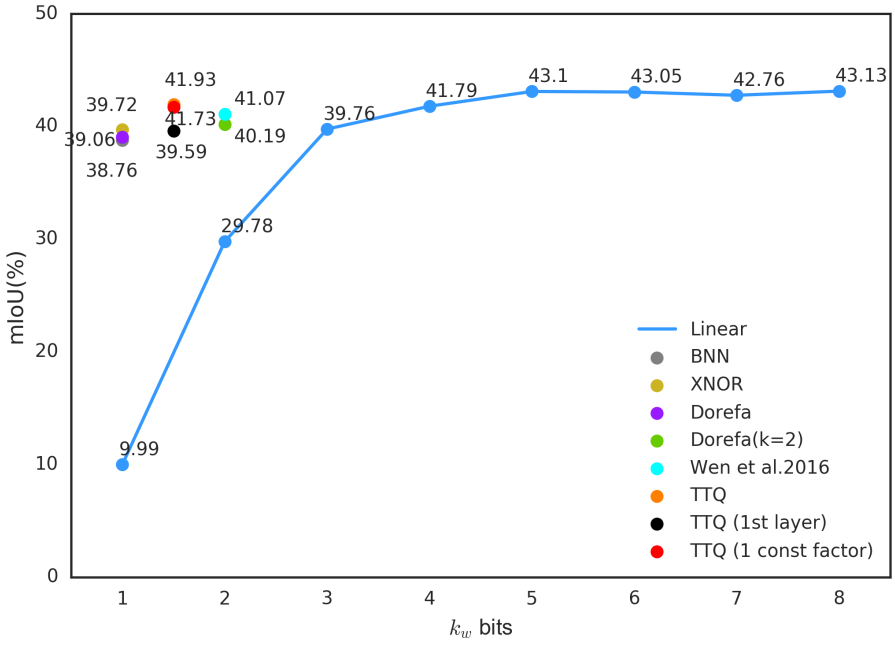


Figure 12. Comparison of weight quantization given activation is held at full precision (32-bit). Ternary weights are marked as 1.5-bit.

5.2. Activation Quantization

Note that activations, different from weights, which are known beforehand, are unbounded and may have significantly larger ranges. In order to constrain a convolutional neural network to have ternary weights and low bit-width activations, we need to retrain the network and enforce the activations to lie in the $[0, R]$ through the clipping function. $R = 4$ is chosen

based on our experimental results.

$$Clip(z) = \begin{cases} 0, & \text{for } z \leq 0 \\ z, & \text{for } 0 < z < R \\ R, & \text{for } z \geq R \end{cases} \tag{5}$$

Thereafter, linear N_{code} bits codebook quantization is applied by equally dividing the range into 2^N sections.

$$X_Q = \frac{R}{2^{bits} - 1} \times round((2^{bits} - 1)\frac{Clip(X, 0, R)}{R} + 0.5) \tag{6}$$

The weights are held at full precision and activation quantization is explored by turn as demonstrated in Fig. 13. The methods include of linear quantization, logarithmic quantization and codebook quantization with different clipping ranges R, i.e. $(W, A, R) = (32, k_a, R)$. It is observed that for a wider clipping range, accuracy can be maintained for less quantization, $k_a = 8$ for instance. However, low bit quantization such as $k_a = 2$, large R may lead to significant loss. Hence, different quantization methods and (k_a, R) combinations were evaluated around 3 and 4 bits. By experiment, codebook quantization maintains a higher accuracy for a low bit-width since codebook quantizations are applied by quantizing the index so as to preserve higher bits for weights. To sum up, $R = 4$ and $k_a = 4$ bits is able to achieve performance without loss.

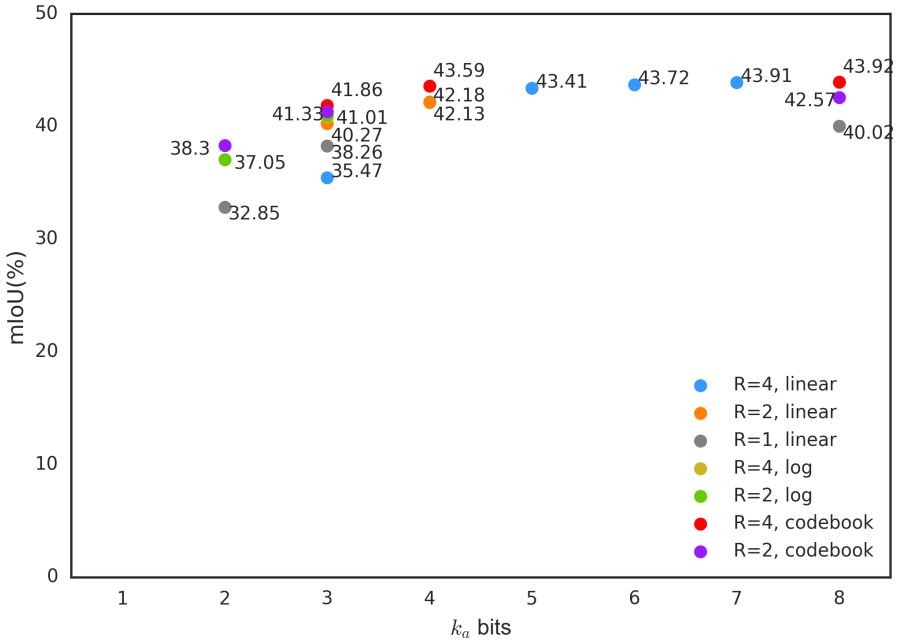
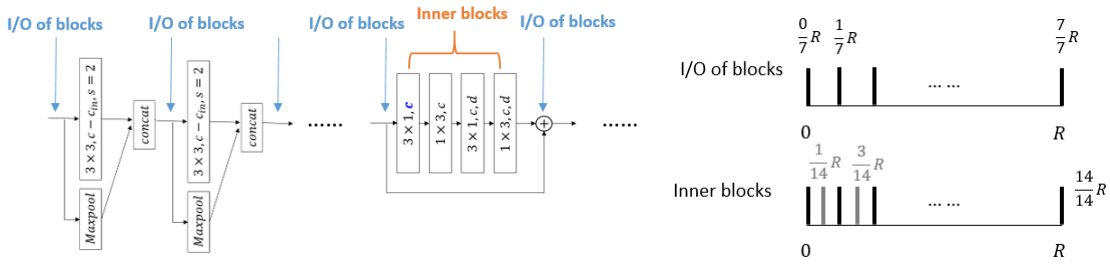


Figure 13. Comparison of activation quantization given weights maintain full precision.

5.3. Activation Quantization for Ternary Weights

In accordance with the experiments above, a serious mIOU drop is observed between 3-bit and 4-bit. We further experiment on non-uniform bit width. As residual connections often cost more bandwidth and dilated convolutions play a critical role, we quantize the I/O of each block of layers to 3 bits while others remain at 4 bits (Fig. 14(a)). Nevertheless, the computation of these convolutional layers is still carried out using the actual values. We transform the computation from those actual values into the codebook domain, including the learned parameters in batch normalization layers. We can thereby accomplish the calculation by merely accumulating the index of the codebook. Moreover, to align activations with different bit-width in different layers, we choose the size of the codebook $N_{code} = 8$ for 3 bits and $N_{code} = 15$ (instead of 2^4) for others (Fig. 14(b)). By combining this process with weight ternarization, the overall proposed quantization procedure can be



(a) Illustration of I/O of blocks and inner blocks. (b) Bit-width alignment for 3- and 4-bit.

Figure 14. Bit-width alignment for I/O of blocks and inner blocks.

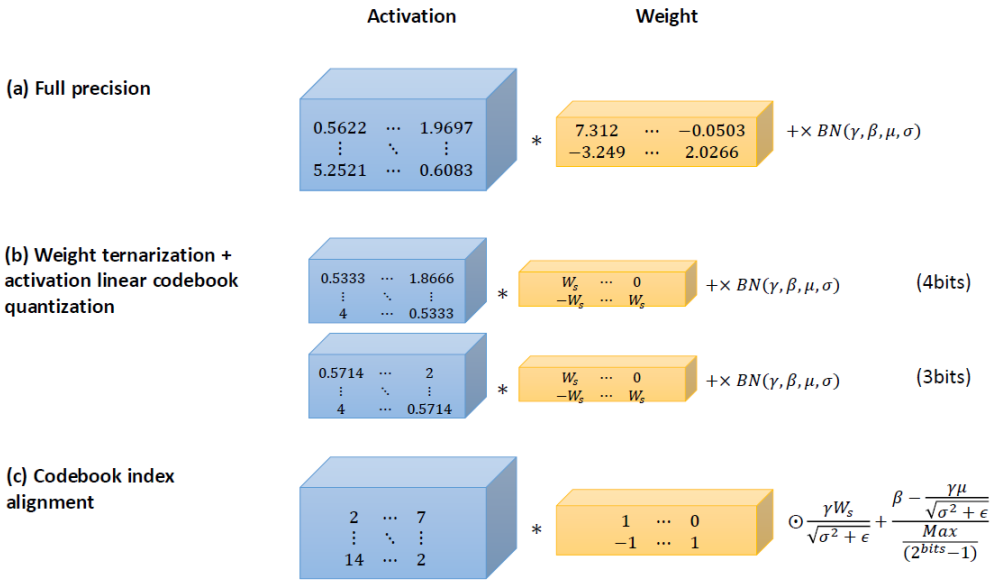


Figure 15. Overall quantization procedure.

expressed with exact $4-$bit addition followed by a smaller number of multiplications in the batch normalization as shown in Fig. 15.

Finally, we evaluate non-uniform activation quantization on the weight ternarization network as shown in Fig. 16, i.e. $(W, A_{\text{i/o}}, A_{\text{inner}}, R) = (1.5, k_o, k_i, 4)$. "Uniform" denotes that all layers are quantized with same bit-width. It is observed that the performance drop occurs at $k_i = k_o = 3$. Hence, we attempt to quantize the I/O of each block and the inner layers at different bit-widths. As mentioned, with residual connection often costing more bandwidth and storage, we quantize I/O to lower bits width and others to higher. "$k_i = 4$(aligned)" is our final result, which aligns activations with 3-bit and 4-bit in different layers, which yields an mIOU of 41.57%, nearly lossless when compared to the full precision activation network.

5.4. Results

Table 3 shows our class-wise quantization results. It can be observed that most performance degeneration occurs in categories that are more difficult to classify. In SUN RGB-D, we observe that for classes which are less confident in pseudo ground truth such as "books" and "object", our result is worse than its 32- bit counterpart (50.26 % and 11.6 % relative mIoU loss, respectively). The observation corresponds to our intuition that a low bit-width quantized network is usually less powerful and thus harder to tolerate faults on training data. Visualizations of our quantization results (with UCM alignment) are shown in Fig. 17.

From a computing resource perspective, the results are shown in Table 4. Although 1.8% mIoU drop occurs, activation memory bandwidth is reduced $8.2\times$. Furthermore, since 53.24% of parameters is zero, only 1.8G 4-bit additions are required and the memory consumption of parameters is reduced by $22\times$ after Huffman coding. Table 5 compares our result to state-of-the-art scene parsing quantization works. 41.07% mIOU is achieved by applying bit-width decay by fine tuning k-bit

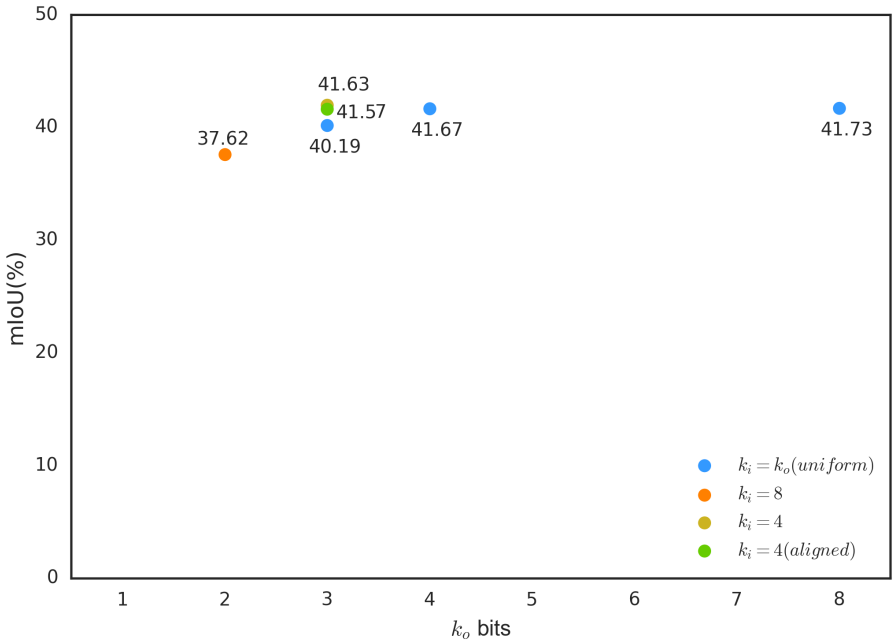


Figure 16. Comparison of both weight ternarization and activation quantization with $R = 4$.

Table 3. Comparison of models with and without quantization.

Input	bed	books	ceil	chair	floor	furn.	objs.	paint	sofa	table	tv	wall	window	**mIOU**	**mIoU (w/o book & obj)**
Full Precision	52.06	23.52	50.03	49.44	81.00	36.39	25.17	28.09	44.64	47.88	19.68	69.69	38.25	**43.53**	**47.01**
Full Precision +UCM Refinement	54.07	21.94	47.54	50.37	81.10	36.56	24.75	30.67	46.23	49.15	17.76	70.19	39.00	**43.80**	**47.51**
Ternary weight + (3,4) bits aligned activation	49.65	11.70	52.90	46.31	78.33	36.89	22.25	27.74	43.22	46.70	19.79	68.66	36.29	**41.57**	**46.04**
Ternary weight + (3,4) bits aligned activation +UCM Refinement	53.62	7.66	52.82	47.57	79.23	37.27	21.48	31.03	45.54	47.75	18.88	69.86	36.82	**42.27**	**47.31**

network from $(k + 1)$-bit network for setting decay rate to 1 which may preserve a better performance compared to DoReFa but is time-consuming. It is shown that our result outperforms them in performance as well as training time consumption with even less bits.

Table 4. Result of model complexity along with required parameter memory and activation memory bandwidth after quantization.

	mIOU	Parameter	Activation	Operation
Full Precision	43.53	7.82 MB	85.88 MB	3.91 GMAC (32 bits)
Ours	41.57	365.6 KB (+huffman)	10.51 MB	1.8G ADD+23.3M MUL (4 bits fixed point)

Figure 17. Visualization of our quantization results (with UCM alignment) on SUN RGB-D test set.

Table 5. Comparison of our method and state-of-the-art scene parsing quantization methods.

	mIoU	Quantize Epoch	bit-width (W-A)	size compressed (W-A)
Full Precision	43.53	-	32-32	1×-1×
DoReFa [13]	40.19	100	2-32	16×-N/A
Wen et al. 2016 [12]	41.07	240	2-32	16×-N/A
Ours	41.57	70	1.5-3.5	21.9×-8.2×

References

[1] Jeannette Bohg, Javier Romero, Alexander Herzog, and Stefan Schaal. Robot arm pose estimation through pixel-wise part classification. In *2014 IEEE International Conference on Robotics and Automation, ICRA*, pages 3143–3150, 2014.

[2] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[3] Shuhang Gu, Wangmeng Zuo, Shi Guo, Yunjin Chen, Chongyu Chen, and Lei Zhang. Learning dynamic guidance for depth image enhancement. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 712–721, 2017.

[4] Bumsub Ham, Minsu Cho, and Jean Ponce. Robust image filtering using joint static and dynamic guidance. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4823–4831, 2015.

[5] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J. Davison. A benchmark for RGB-D visual odometry, 3d reconstruction and SLAM. In *2014 IEEE International Conference on Robotics and Automation, ICRA*, pages 1524–1531, 2014.

[6] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(6):1397–1409, 2013.

[7] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, pages 4107–4115, 2016.

[8] Fengfu Li and Bin Liu. Ternary weight networks. *CoRR*, abs/1605.04711, 2016.

[9] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pages 525–542, 2016.

[10] Xiaoyong Shen, Chao Zhou, Li Xu, and Jiaya Jia. Mutual-structure for joint filtering. *International Journal of Computer Vision*, 125(1-3):19–33, 2017.

[11] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 567–576, 2015.

[12] He Wen, Shuchang Zhou, Zhe Liang, Yuxiang Zhang, Dieqiao Feng, Xinyu Zhou, and Cong Yao. Training bit fully convolutional network for fast semantic segmentation. *CoRR*, abs/1612.00212, 2016.

[13] Shuchang Zhou, Zekun Ni, Xinyu Zhou, He Wen, Yuxin Wu, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *CoRR*, abs/1606.06160, 2016.