1 Datasets

**NYUDv2** NYUDv2 [2] is an indoor RGB-D semantic segmentation dataset. It contains 1449 RGB-D images with pixel-wise labels. And it provides depth maps captured by Kinect and the corresponding camera intrinsic parameters for all images. We follow the 40-class setting and the standard split which consists of 795 training images and 654 testing images.

**Cityscapes** Cityscapes [1] is an urban scene understanding dataset that contains outdoor scene in different cities. The dataset has 5,000 stereo frames, each frame containing an 2048 × 1024 RGB image, a disparity map, a set of camera parameters, and a fine-annotated 19-category ground truth label map. There are 2,979 images in training set, 500 images in validation set and 1,525 images in test set. We use camera parameters and disparity maps to calculate depth maps. The quality of depth data in this dataset is not as good as NYUDv2, and the scenes have wider ranges and more complicated structures.

2 More experiments of the initialization of introduced parameters

In Table 1, we compare more different initialization settings of the introduced parameters $a_k$ and $t$. When we change the initialization settings, the performance may slightly drop, but still outperforms the baseline and 2.5D convolution. This validates the effectiveness and the robustness of the learnable depth receptive field in our method. Small initialization values of $a_k$ seems to have relatively obvious harm on the performance. We suppose that it is because in this case the receptive fields of different kernels are largely overlapped at the initial state, and it brings difficulty for learning.
Table 1. Results of different initialization of $a_k$ and $t$. The backbone model is ResNet-50

<table>
<thead>
<tr>
<th>Method</th>
<th>$a_k$</th>
<th>$t$</th>
<th>mIoU(%)</th>
<th>pixel Acc(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>-</td>
<td>-</td>
<td>44.56</td>
<td>73.01</td>
</tr>
<tr>
<td>2.5D[3]</td>
<td>-</td>
<td>-</td>
<td>48.23</td>
<td>75.73</td>
</tr>
<tr>
<td>Malleable 2.5D</td>
<td>$[-4, -2, 0, 2, 4]$</td>
<td>1</td>
<td>48.66</td>
<td>75.94</td>
</tr>
<tr>
<td>Malleable 2.5D</td>
<td>$[-2, -1, 0, 1, 2]$</td>
<td>1</td>
<td>48.80</td>
<td>76.03</td>
</tr>
<tr>
<td>Malleable 2.5D</td>
<td>$[-1, -0.5, 0, 0.5, 1]$</td>
<td>1</td>
<td>48.42</td>
<td>75.78</td>
</tr>
</tbody>
</table>

3 Images of Assigning Functions $h_k$ and $g_k$

To give a clear illustration of the assigning functions $h_k$ and $g_k$, we present several real-case images of $h_k$ and $g_k$ in Fig. 1. We draw the images of the initialization state of $h_k$ and $g_k$, and we also draw the images of $h_k$ and $g_k$ in a model trained on NYUDv2.

4 Comparisons of Depth Receptive Functions

In Fig. 2 and Fig. 3, we compare depth receptive fields of depth-aware, 2.5D and malleable 2.5D convolution in NYUDv2 and Cityscapes. In NYUDv2, pixels are closer in 3D space than those in Cityscapes since they are respectively indoor and outdoor scenes. Therefore, it is intuitive that convolutions should have larger depth receptive fields on Cityscapes than NYUDv2. From the figures we can see that malleable 2.5D convolution indeed learns wider depth receptive fields for Cityscapes, while depth-aware and 2.5D convolutions cannot automatically fit different environments.

5 Effects of Kernel Rebalancing

In Fig. 4, we present the effects of kernel rebalancing. We show the kernel rebalancing results of all four malleable 2.5D convolutions in the model trained on NYUDv2. As we know, in earlier stages, local geometric features play a more important role, and in later stages, the importance of capturing context increases. The rebalancing parameters in early stages only fix part of the imbalance problem and keep the two further kernels decayed compared to the center kernel, which makes the convolution sensitive to local depth changes. When comes to the later stages, the rebalancing parameters tend to balance the kernels well and therefore let the convolution able to handle long-distance relations.
Fig. 1. Images of $h_k$ and $g_k$. (a) and (b) are the initialization of $h_k$ and $g_k$. And the rest subfigures are $h_k$-s and $g_k$-s of each malleable 2.5D convolution at different ResNet stages after training.
Fig. 2. Comparison of depth receptive field functions $g_k$ on NYUDv2 where the depth $d(c_i) = 1m$. We compare depth-aware, 2.5D and malleable 2.5D convolution at each stages of ResNet after training. Note that we scale the y-axis to see better. The overall scale does not affect output results because of batch normalizations.
Fig. 3. Comparison of depth receptive field functions $g_k$ on Cityscapes where the depth $d(c_i) = 20m$. We compare depth-aware, 2.5D and malleable 2.5D convolution at each stages of ResNet after training. Note that we scale the y-axis to see better. The overall scale does not affect output results because of batch normalizations.
Fig. 4. The ratio of pixels assigned to each kernel, before and after rebalance. We count the sum of $g_k$ and $s_k \cdot g_k$ for each kernel across the whole NYUDv2 dataset and calculate the ratio.
6 Visualization of feature maps

In Fig. 5, we visualize the feature maps generated by different kernels of a malleable 2.5D convolution. We save the output feature maps of the malleable 2.5D convolution in res2 stage of a trained ResNet-101-based model, and select two channels to draw figures. Generally, the three kernels respectively handle pixels that are "in front of", "around the same depth with" and "behind" the center pixel of a local receptive field. From the feature maps, we can see that the three kernels indeed learn different relations and can activate accordingly.

**Fig. 5.** Visualization of the feature maps generated by different kernels in a malleable 2.5D convolution. We draw 2 feature maps for each kernel and each input image. The feature maps of "kernel 1" are determined by what is in front of the center pixel of a local receptive field. The feature maps of "kernel 2" are determined by what is around the same depth with the center pixel of a local receptive field. The feature maps of "kernel 3" are determined by what is behind the center pixel of a local receptive field.
7 Network Structures

In Fig. 6, we present the network structures we use in NYUDv2 and Cityscapes respectively. We adopt ResNet-based DeepLabv3+ as our baseline network. To evaluate the effect of our method, we replace the 3 × 3 convolution with a malleable 2.5D convolution in the first residual unit in each stage of the ResNet. For the NYUDv2 dataset, we adopt a multi-stage merging block on the backbone network. And for Cityscapes, we keep the original DeepLabv3+ structure.

References

Fig. 6. Network structures