Supplementary
Bi3D: Stereo Depth Estimation via Binary Classifications

Abhishek Badki[1,2] Alejandro Troccoli[1] Kihwan Kim[1] Jan Kautz[1] Pradeep Sen[2] Orazio Gallo[1]

[1]NVIDIA [2]University of California, Santa Barbara

1. Additional Details on Training

Below we provide more details for the network architecture used in our paper:

FeatNet. The input images for our feature extraction network are normalized using mean and standard deviation of 0.5 for each color channel. This network is based on the feature extraction network of PSMNet [1]. We first apply a convolutional layer with a stride of 3 to get downsampled features. This is followed by two convolutional layers, each with a stride of 1. We use 3×3 kernels, a feature-size of 32, and a ReLU activation. This is followed by two residual blocks as proposed in PSMNet [1] each with a dilation of 2, a stride of 1 and a feature-size of 32. This is followed by the SPP module and the final fusion operation as explained in PSMNet [1] to generate a 32-channel feature map for the input image. We do not use any batch normalization layers in our training.

SegNet. SegNet architecture takes as input concatenated left-image features and warped right-image features and generates a binary segmentation confidence map. SegNet is a 2D encoder-decoder with skip-connections. The basic block of the encoder is composed of a convolutional layer that downsamples the features with a stride of 2 followed by another convolutional layer with a stride of 1. We use 3×3 kernels. We repeat this block 5 times in the encoder. The feature-sizes for these blocks are 128, 256, 512, 512 and 512 respectively. The basic block of the decoder is composed of a deconvolutional layer with 4×4 kernels and a stride of 2. This is followed by a convolutional layer with 3×3 kernels and a stride of 1. This block is repeated 5 times to generate the output at the same resolution of the input. The feature-sizes for these blocks are 512, 512, 256, 128 and 64 respectively. For all our layers we use a LeakyReLU activation with slope of 0.1. We do not use a batch normalization layer in our network. We have a final convolutional layer with 3×3 kernels and without any activation to generate the output of SegNet. Applying sigmoid to this output generates the binary segmentation confidence map.

RegNet. SegNet is applied independently for each input plane to generate the corresponding binary segmentation maps when we apply sigmoid operation. RegNet is a 3D encoder-decoder architecture with residual connections and is based on GC-Net [2]. The outputs of the SegNet corresponding to all input planes are concatenated to form an input 3D volume. RegNet refines this volume using input left images features from FeatNet as a guide. Note that this architecture does not take the warped right image features. We first pre-process the input 3D volume using a convolutional layer with 3×3 kernels. We use a feature-size of 16, a stride of 1, and a ReLU activation for this step. Then we concatenate the left image features with the features of each confidence map to generate an input volume with a feature-size of 48. This serves as input to a 3D encoder-decoder architecture. This architecture is same as the one proposed in Section 3.3 in GC-Net [2]. However, we use only half the features as used in the original GC-Net [2] architecture and don't use any batch normalization layers in our architecture. The output of this network is a refined volume at the same resolution as input. Applying sigmoid operation gives us the refined binary segmentation confidence volume.

DispRefine. Our disparity refinement network uses the left-image as a guide to refine the disparity map computed using area under the curve operation on binary segmentation confidence volume. We use the network proposed in StereoNet [3] for this purpose. However, we don't use any batch normalization layers in our network.

SegRefine. Our SegRefine network refines the upsampled output of the SegNet using the left-image as a guide. We first apply three convolutional layers each with 3×3 kernels with a stride of 1 on the input left-image. We use a feature-size of 16 for these layers. We apply ReLU activation on the first two layers. The third layer does not have any activation and gives us left-image features at the resolution of input left-image. Note that these features need to be computed only once per stereo pair. We then concatenate these feature maps with the up-sampled output of SegNet and estimate the refined output by applying a single convolutional layer with a 3×3 kernel. Applying sigmoid operation to the output of this layer gives us the binary segmentation confidence map at the resolution of input image.

2. Supplementary video

We explain the adaptive depth estimation application through video demonstration in our supplementary video. We also further discuss use of area under the curve (AUC) formulation for disparity regression in the supplementary video. Please refer to the supplementary video for a complete overview of our work.

References

[1] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[2] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[3] Sameh Khamis, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi. StereoNet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1