Neural Markov Random Field for Stereo Matching

Supplementary Material

1. Implementation Details of Loss Functions

Superpixel-guided disparity downsample. To provide supervision signal for disparity proposal extraction at $1/8$ resolution, we introduce a superpixel-guided disparity map downsample function, which reduces each 8×8 disparity window to multiple modals. We divide the ground truth disparity map into non-overlapping 8×8 windows and perform an independent downsample for each window.

First, we over-segment the left image I^L into superpixels using the LSC method implemented in OpenCV[3]. As shown in Fig. 7, the superpixel effectively groups adjacent pixels while preserving local image structures, making it appropriate for reducing disparity values. Subsequently, each 8×8 window is decomposed into multiple segments utilizing the superpixel label map. We sort the segments based on their pixel count and compute the *median* disparity of each segment as the representative. To mitigate over-segmentation in the window, we employ a non-maximum suppression (NMS) on the representative disparity list. The suppression criterion is based on the difference between representative disparity values. If the absolute difference is less than 0.5 pixels, we merge the suppressed segment into the segment that suppresses it. After the merge step, we sort the segments again based on the pixel count and choose the median disparity of the top 4 segments as the downsample function output. If there exist fewer than 4 segments, we pad the output with null values.

Proposal loss. Once we have obtained the downsampled ground truth disparity modals, we use it to train our disparity proposal extraction network, as detailed in Sec. 3.5 of the paper. When computing the proposal loss in Eq. (7), we need to find the optimal bipartite matching between proposals and ground truth modals. For instance, consider a pixel on the coarse level with four ground truth disparity modals, namely $\{1.1, 1.8, \phi, \phi\}$, and four extracted proposals, namely $\{1.4, 10.2, 10.8, 11.2\}$. Ignoring null value ϕ, the optimal bipartite matching pairs consist of $(1.1, 1.4)$ and $(1.8, 10.2)$. However, we need to be careful with the close ground truth modals. In this case, the proposal 1.4 already captures the two close ground truth modals 1.1 and 1.8. Thus, the matching pair $(1.8, 10.2)$ is unnecessary and may induce negative impact on the training.

To address this, we perform an online non-maximum suppression (NMS) on ground truth modals when computing the proposal loss. First, we sort ground truth modals based on their proximity to the extracted proposal set. Proximity is measured by the minimum distance between the ground truth modal and all proposals. Then, we suppress the close ground truth modals using the threshold of 8 pixels. In continuation with the above example, our online NMS reduces the ground truth modals to $\{1.1, \phi, \phi, \phi\}$, and only one matching pair $(1.1, 1.4)$ is leveraged for proposal loss.

[3] https://opencv.org

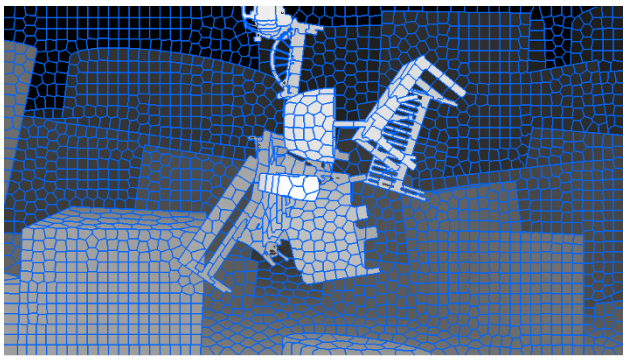

Figure 7. Superpixels overlaid on ground truth disparity map.

Initialization loss. Besides the proposal loss, an initialization loss is also employed to supervise label seeds to identify ground truth modals. As described in Sec. 3.3, label seeds are derived from a 3D cost volume $\mathbf{C}$, with

$$\mathbf{C}(i, j, z) = \langle \tilde{F}^L(i, j), \tilde{F}^R(i - z, j) \rangle. \tag{9}$$

We expect the initialization loss to penalize the discrepancy between ground truth modals and the 3D cost volume $\mathbf{C}$. To this end, we transform the ground truth modals of each pixel o into a probability distribution, $p^*(z) = \sum_k w_k \delta(z - z_k^*)$, where $\{z_k^*\}$ are the ground truth modals at pixel o and δ is the Dirac delta function. The mass weights $\{w_k\}$ are empirically set to $\{0.5, 0.3, 0.1, 0.1\}$ for the four sorted ground truth modals. This simple strategy performs well in all our experiments. Note that ground truth modals are given with subpixel precision, however, label seed extractioin happens with integer disparities. Therefore, we displace the probability mass as z_k^* to nearby integer disparities as

$$\tilde{p}^*(z) = \sum_k w_k(\lfloor z_k^* \rfloor + 1 - z_k^*)\delta(z - \lfloor z_k^* \rfloor) + w_k(z_k^* - \lfloor z_k^* \rfloor)\delta(z - \lfloor z_k^* \rfloor - 1). \tag{10}$$

We define the initialization loss to be the cross entropy between ground truth probability $\tilde{p}^*$ and softmax of 3D cost volume $\mathbf{C}$ along the z dimension, *i.e.*,

$$L^{\text{init}} = - \sum_{z \in [0, z_{\max}]} \tilde{p}^*(z) \cdot \log(\text{softmax}_z(\mathbf{C}(i, j, z))). \tag{11}$$

2. Additional Implementation Details

Local feature CNN. We use a similar backbone as RAFT-Stereo [30], which consists of a strided-2 stem and three residual blocks with strides 1, 2, 1, respectively. The network produces a feature map with 128 channels at $1/4$ input image resolution, which is then downsampled through average pooling with a stride of 2 and a kernel size of 2. We further pass the obtained $1/8$ resolution feature map and the original $1/4$ resolution feature map to a shared convolution layer with 256 channels.

Neural message passing. The number of message passing blocks we use in label seeds propagation (N_p), MRF inference (N_i), and refinement (N_f) are 5, 10, 5 respectively. We use same settings for all experiments. The channels of embedding vectors in all message passing blocks are always 128. The neighborhood window size is 4×4 for refinement (Sec. 3.4), and 6×6 for neural MRF inference (Sec. 3.2). We also found that more message passing blocks and larger window size would bring slightly better accuracy with considerable computation overhead.

Observed label feature. The observed feature of a candidate label must integrate matching cues from both left and right views. Given the coarse level features $\tilde{F}^L$ and $\tilde{F}^R$, we compute the observed feature $\mathbf{x}_v$ of a candidate label positioned at $\mathbf{p}_v := (i, j, z)$ using a warping function w.r.t. $\tilde{F}^L(i, j)$ and $\tilde{F}^R(i - z, j)$ as

$$\begin{aligned} \mathbf{x}_v &= \text{MLP}\left(\mathbf{x}_v^{\text{concat}} || \mathbf{x}_v^{\text{corr}}\right) \\ \mathbf{x}_v^{\text{concat}} &= \gamma_1\left(\tilde{F}^L(i,j)\right) || \gamma_1\left(\tilde{F}^R(i-z,j)\right) \\ \mathbf{x}_{v,g}^{\text{corr}} &= \frac{N_g}{N_c}\langle \gamma_2\left(\tilde{F}_g^L(i,j)\right), \gamma_2\left(\tilde{F}_g^R(i-z,j)\right)\rangle, \end{aligned} \tag{12}$$

where $[\cdot||\cdot]$ denotes concatenation along the channel dimension. $\tilde{F}_g^L, \tilde{F}_g^R$ are g^{th} grouped features of F^L and $\tilde{F}^R$, which are evenly divided into N_g groups. N_c is the channel of coarse level features, and $\langle \cdot, \cdot \rangle$ denotes the inner product. γ_1 and γ_2 are normalization functions to make the terms $\mathbf{x}_v^{\text{concat}}$ and $\mathbf{x}_v^{\text{corr}}$ share similar data distribution. Both γ_1 and γ_2 consist of two linear layers, with instance normalization and activation function following the first linear layer. Since disparity z is a real number, we leverage bilinear interpolation when indexing feature map $\tilde{F}^R$. Our formulation is inspired by the success of GwcNet [17] and PCWNet [43].

In the refinement stage, we use the same warping function as Eq. (12), but w.r.t. fine level features $\hat{F}^L$ and $\hat{F}^R$.

Cross-shaped window attention. To efficiently capture long-range dependency for label seed message exchange, we employ the cross-shaped attention mechanism proposed in CSWin Transformer [13]. As illustrated in Fig. 8a, the interested label seed, positioned at $\mathbf{p}_v := (i, j, z_k)$, aggregates matching information from all other label seeds that share the same i or j coordinate. We follow the parallel multi-head grouping strategy and locally-enhanced positional encoding of CSWin Attention [13] when performing attentional aggregation. The initial matching feature ${}^{(0)}\mathbf{d}_v$ of a label seed v is expected to encode cost features and underlying disparity value, formally defined as:

$${}^{(0)}\mathbf{d}_v = \text{MLP}\Big(\gamma_3\left(L_z\left(\mathbf{C}(i,j,:)\right)\right) \parallel \text{PE}(z)\Big), \tag{13}$$

where $\mathbf{C}$ denotes the 3D cost volume computed in label seeds extraction using Eq. (9). The lookup operator L_z retrieves cost features from volume slice $\mathbf{C}(i, j, :)$ around integer disparity z for pixel (i, j), akin to RAFT-Stereo [30]. We apply a two-layer MLP called γ_3 to normalize the retrieved cost features before concatenating it with the sinusoidal positional encoding (PE) of disparity z.

We validate the design of cross-shaped window attention by comparing with the local window attention shown in Fig. 8b. The local window size is set to 8×8 to match the computation complexity of cross-shaped window attention on SceneFlow dataset [33]. The results are shown in Tab. 6. Due to its ability to capture long-range dependency, cross-shaped window attention performs better than local window attention for label seed propagation. However, we do not adopt the cross-shaped window attention in neural MRF inference and refinement, since it does not adapt to our proposed content-adaptive positional bias and position aggregation for different input resolutions.

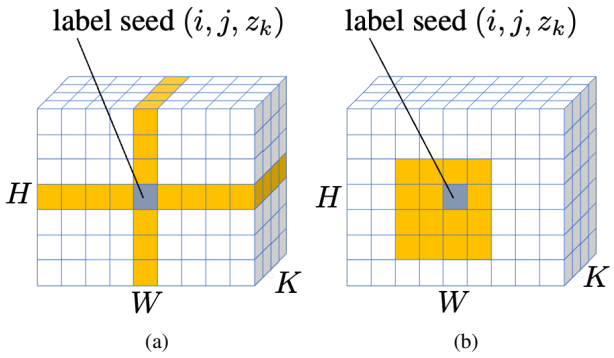

Figure 8. (a) Cross-shaped attention window arrangement, (b) local attention window arrangement.

	candidate labels		disparity estimation	
	3px [%]↑	8px [%]↑	EPE [px]↓	Bad 1.0 [%]↓
cross-shaped	99.29	99.77	0.45	4.50
local window	99.18	99.72	0.47	4.56

Table 6. Performance comparison between cross-shaped window attention and local window attention in label seed propagation.