PBCStereo: A Compressed Stereo Network with Pure Binary Convolutional Operations

Jiaxuan Cai[0000−0002−1787−5132], Zhi Qi*[0000−0003−4699−6440], Keqi Fu[0000−0001−6361−3865], Xulong Shi[0000−0002−5137−6038], Zan Li[0000−0002−0460−8393], Xuanyu Liu[0000−0001−8544−3430], and Hao Liu

Southeast University, Nanjing, China
{jiaxuan_cai,101011256,220201602,long,lz21,liuxy17,nicky_lh}@seu.edu.cn

Abstract. Although end-to-end stereo matching networks achieve great performance for disparity estimation, most of them require far too many floating-point operations to deploying on resource-constrained devices. To solve this problem, we propose PBCStereo, the first lightweight stereo network using pure binarized convolutional operations. The degradation of feature diversity, which is aggravated by binary deconvolution, is alleviated via our novel upsampling module (IBC). Furthermore, we propose an effective coding method, named BIL, for the insufficient binarization of the input layer. Based on IBC modules and BIL coding, all convolutional operations become binary in our stereo matching pipeline. PBCStereo gets 39× saving in OPs while achieving comparable accuracy on SceneFlow and KITTI datasets.

Keywords: Stereo Matching · Disparity Estimation · Binary Neural Network.

1 Introduction

Depth estimation plays an important role in complex computer vision tasks such as autonomous driving[10], augmented reality [1]and robot navigation[24]. Compared with the usage of expensive LiDARs, stereo matching algorithms that calculate the dense disparity from two input images provide a low-cost but equally accurate solution to depth estimation.

As the end-to-end CNNs are proposed for depth estimation, the accuracy of stereo matching has been greatly improved. At present, top 70 works on KITTI 2015 leaderboard have reached the overall three-pixel-error less than 2%[10]. While lots of effort has been carried out to improve the accuracy, the efficiency of stereo algorithms is still far from satisfactory. Because the large input image size and expensive convolutional operations consume such a large amount of floating point operations that the average FPS of those 70 works is only 1.2.

To cope with the challenge of heavy calculations in depth estimation, one strategy is adopting compact convolutional filters to replace standard convolution. For example, MABNet[32] and LWSN[30] have been proposed using depthwise separable convolution to reduce FLOPs. LWANet[9] designed pseudo 3D convolution to replace regular 3D convolution for aggregating the cost volume with less computational cost. However, these approaches are difficult to optimize memory access in hardware deployment due to the change of convolution mode and still use expensive floating-point operations. Once targeting at the applications on resource-constrained edge devices, a highly efficient stereo matching algorithm becomes more urgent.

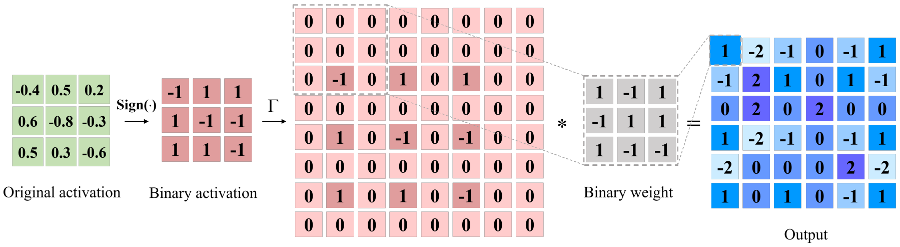
One effective approach to greatly reduce floating-point operations for resource-constrained devices is model binarization. Model binarization can achieve extreme compression ratio because both the weight and activation are represented by 1-bit[25] and floating point convolutions are replaced by XNOR and POPCOUNT operations[3]. Therefore, not only the cost of memory access, but also the amount of computation expense is greatly reduced. Binarization is also able to benefit stereo matching algorithms in terms of saving energy consumption for edge devices. StereoEngine[5] and StereoBit[6] made attempts to binarize the feature extraction module in stereo matching pipeline. Ignoring Stereobit's floating-point calculations in modules like aggregation and refinement, we find that its binary feature extraction module consumes 1.52G OPs, which is 2.7 times PBCStereo's total computational expense. There are two challenges for binarizing end-to-end stereo matching algorithms. Firstly, the degradation of feature diversity[31,16] is aggravated by binary deconvolution, which leads to the incomplete feature geometry as shown in Fig. 2. Secondly, the binarization of the input layer causes sharp decline of accuracy, resulting from the pixel distortion of binarized input images.

In this paper, we propose a stereo matching network, named PBCStereo, which replaces all floating point convolution with efficient binary operations. With 39× saving in OPs, the overall three-pixel-error of PBCStereo is 4.73% on KITTI 2015 benchmark. PBCStereo presents a reasonable balance between accuracy and energy-efficient computing, making depth estimation more suitable for deployment on embedded devices. The main contributions of this article are as follows:

- We design an embedded upsampling module to replace binary deconvolution, named IBC module, which uses interpolation and binary convolution to alleviate the phenomenon of feature homogenization.
- For the input layer, we propose a precision-preserving coding method named BIL to avoid the unary pixel distortion, so that all convolutional operations in our design become binarized without sacrificing performance.
- Based on the IBC module and BIL coding method, we design an efficient backbone PBCStereo. PBCStereo takes only 0.57G OPs with comparable accuracy for estimating the depth from a stereo pair at 512×256 resolution.

2 BNN Preliminaries

Quantization for neural networks can reduce the bit width of data, effectively decreasing the power consumption of computation. Among the existing quantization techniques, binarization extremely compresses both the weight and activation to only 1-bit. BNN[7] is a pioneering work that first verified the feasibility of binarization on small datasets such as MNIST and CIFAR-10. Subsequently, a series of works were proposed. XNOR-NET[27] proposed floating point scaling factors acting on the channel dimension to recover the information loss, enhancing the top-1 accuracy to 51.2% on ImageNet for the first time. Bi-RealNet[19] and BinaryDenseNet[2] respectively found that there were some specific structures that could effectively reduce the negative impact of information loss in the process of binarization. ReactNet[18] changed the traditional Sign and PRelu functions to enable explicit learning of the distribution reshape and shift at near-zero extra cost, mitigating the accuracy gap between the binarized model and its full-precision counterpart. However, these methods developed on ImageNet classification are not readily transferable to depth estimation directly.



-0.4	0.5	0.2
0.6	-0.8	-0.3
0.5	0.3	-0.6

-1	1	1
1	-1	-1
1	1	-1

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	-1	0	1	0	1	0	0
0	0	0	0	0	0	0	0
0	1	0	-1	0	-1	0	0
0	0	0	0	0	0	0	0
0	1	0	1	0	-1	0	0
0	0	0	0	0	0	0	0

1	-1	1
-1	1	1
1	-1	-1

1	-2	-1	0	-1	1
-1	2	1	0	1	-1
0	2	0	2	0	0
1	-2	-1	0	-1	1
-2	0	0	0	2	-2
1	0	1	0	-1	1

Fig. 1. The binary deconvolution process of a 3 × 3 input with stride=2, padding=1 and output padding=1.

In BNNs, both weights and activations are restricted to -1 and 1. We define a binary convolutional layer as

$$O = X^b * W^b \tag{1}$$

O is the convolution output. X^b and W^b represent binary activations and binary weights. They are binarized through a sign function. Specifically,

$$x^b = sign(x^r) = \begin{cases} +1, & x^r \geq 0, \\ -1, & x^r < 0. \end{cases} \tag{2}$$

$$w^b = \frac{\|W^r\|_1}{n} sign(w^r) = \begin{cases} +\frac{\|W^r\|_1}{n}, & w^r \geq 0, \\ -\frac{\|W^r\|_1}{n}, & w^r < 0. \end{cases} \tag{3}$$

The superscripts b and r respectively refer to binary and real values. $\frac{\|W^r\|_1}{n}$ is a floating-point scaling factor proposed in XNOR-Net[27], meaning the average of absolute values of all weights.

Different from convolution, deconvolution adopts the expansion operator (Γ) of adding zero padding to both sizes of the input to improve the resolution, as shown in Fig. 1. Therefore, the binary deconvolutional layer can be rewritten as

$$X^{b^{'}} = \Gamma(X^b) \tag{4}$$

$$O = X^{b^{'}} * W^b \tag{5}$$

$X^{b^{'}}$ is a new input after the expansion operation.

3 Method

In this section, we first analyze how binary deconvolution pollutes feature quality. Based on this analysis, we propose IBC module to effectively improve the quality of upsampling in Section 3.1. Then we put forward a precision-preserving BIL coding method for the input layer in Section 3.2. Finally, we introduce the overall network design of PBCstereo in Section 3.3.

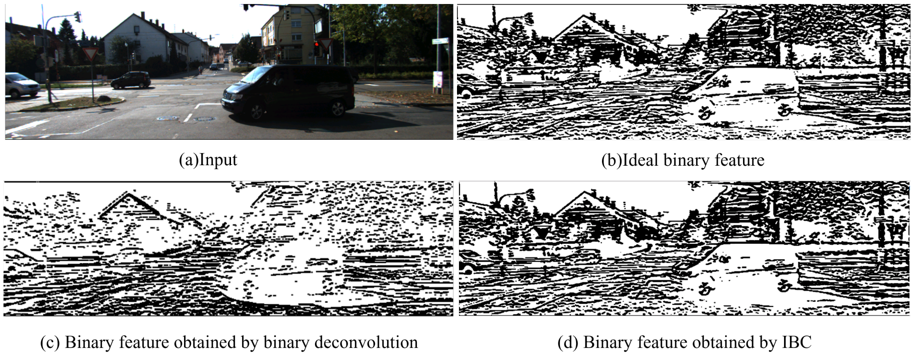

(a)Input
(b)Ideal binary feature
(c) Binary feature obtained by binary deconvolution
(d) Binary feature obtained by IBC

Fig. 2. Comparison of the feature obtained by different upsampling methods. (a) Input image. (b) The ideal binary feature is obtained by real-valued deconvolution and sign function. (c) The feature recovered by binary deconvolution is incomplete. (d) The result of IBC module.

3.1 IBC module

As shown in Fig. 2, the geometry obtained by binary deconvolution is incomplete, which indicates that there is a considerable information loss in the process of binary deconvolution. Given a binary activation and binary $k \times k \times C$ kernels, the output values could be an integer in the range of $-Ck^2$ to Ck^2. If there are p zeros in the $k \times k \times C$ convolution window, the range of output will be reduced to $p - Ck^2$ to $Ck^2 - p$. In other words, the operator (Γ) of adding zero padding dilutes the information in the feature map and makes features more difficult to be distinguished. So some effective structures are confused with the surrounding background and filtered out by *sign* function. Moreover, the smaller the number of channels, the more obvious this phenomenon is. For stereo matching algorithms, if the disparity map cannot maintain the same geometry as the original input, the final matching accuracy will be greatly affected.

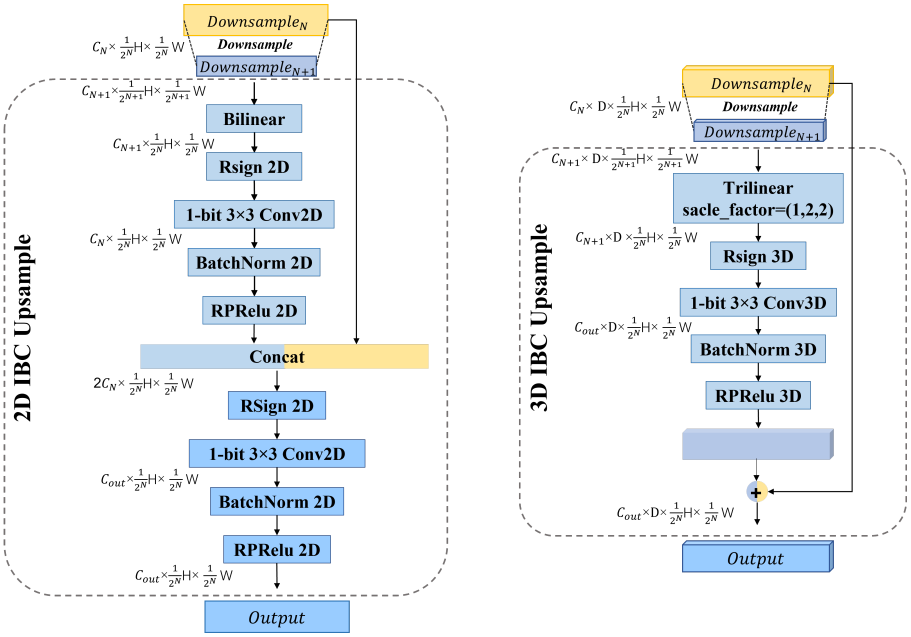

$Downsample_N$
Downsample
$Downsample_{N+1}$
$C_N\times \frac{1}{2^N}H\times \frac{1}{2^N}W$
$C_{N+1}\times\frac{1}{2^{N+1}}H\times \frac{1}{2^{N+1}}W$
Bilinear
$C_{N+1}\times\frac{1}{2^N}H\times \frac{1}{2^N}W$
Rsign 2D
1-bit 3×3 Conv2D
$C_N\times \frac{1}{2^N}H\times \frac{1}{2^N}W$
BatchNorm 2D
RPRelu 2D
Concat
$2C_N\times \frac{1}{2^N}H\times \frac{1}{2^N}W$
RSign 2D
1-bit 3×3 Conv2D
$C_{out}\times\frac{1}{2^N}H\times \frac{1}{2^N}W$
BatchNorm 2D
RPRelu 2D
$C_{out}\times\frac{1}{2^N}H\times \frac{1}{2^N}W$
Output
2D IBC Upsample
$Downsample_N$
Downsample
$Downsample_{N+1}$
$C_N\times D\times\frac{1}{2^N}H\times \frac{1}{2^N}W$
$C_{N+1}\times D\times\frac{1}{2^{N+1}}H\times \frac{1}{2^{N+1}}W$
Trilinear
sacle_factor=(1,2,2)
$C_{N+1}\times D\times\frac{1}{2^N}H\times \frac{1}{2^N}W$
Rsign 3D
1-bit 3×3 Conv3D
$C_{out}\times D\times\frac{1}{2^N}H\times \frac{1}{2^N}W$
BatchNorm 3D
RPRelu 3D
+
$C_{out}\times D\times\frac{1}{2^N}H\times \frac{1}{2^N}W$
Output
3D IBC Upsample

Fig. 3. The proposed IBC module, which respectively replaces 2D binary deconvolution and 3D binary deconvolution. The transformation of data dimension is marked.

For stereo matching algorithms, there are generally two forms of deconvolution, 2D and 3D. In order to improve the feature quality of upsampling and prevent the information from being diluted, we design 2D IBC and 3D IBC module respectively as shown in Fig. 3. Inspired by ReactNet[18], we choose *RSign* as the sign function and *RPrelu* as the activation function. We design the 2D IBC module in two stages. The first stage completes the expansion of the image size and the conversion of channel numbers. We first adopt bilinear interpolation to double the resolution of the input feature map $Downsample_{N+1}$, where the subscript $N+1$ denotes the number of downsampling. Then, 1-bit convolutional layer changes the number of channels from C_{N+1} to C_N, the same as $Downsample_N$. In the second stage, we exploit context information to make up for the information loss due to binarization. To achieve this purpose, we concatenate $Downsample_N$ with the output in the first stage. Therefore, the shallow and deep features of the network will be effectively fused. Concatenating also makes use of more feature maps to alleviate the phenomenon of feature homogenization, increasing the output range to $-2C_N k^2$ to $2C_N k^2$. Meanwhile, all convolutional operations in IBC module are binarized, which is conducive to hardware acceleration. In addition to the 2D IBC module, we also implement an efficient 3D IBC module in a similar way, as shown in Fig. 3. The difference is that 3D deconvolution is generally applied to the part of cost aggregation. The resolution of cost aggregation is much smaller than that of feature extraction. 3D deconvolution also has one more dimension than 2D deconvolution, resulting in a larger range and more calculations after binarization. Therefore, instead of concatenating to build a larger feature map, we use the residual connection summation to make up for the information loss.

3.2 Binarizing Input Layer

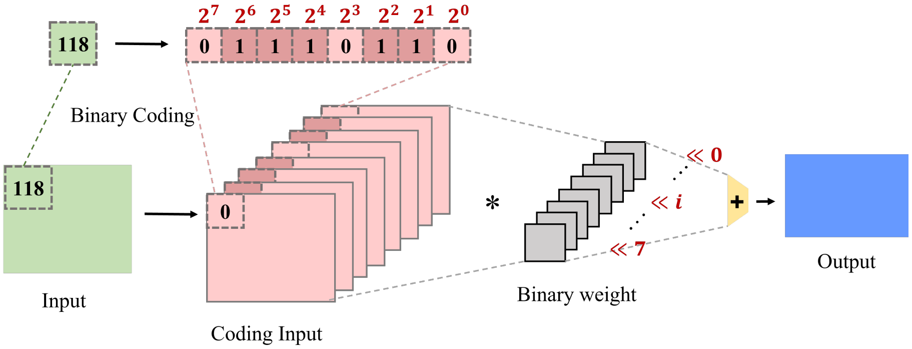


Fig. 4. The proposed BIL coding method. The binary vector after encoding denotes to the magnitude of the pixel.

In order to preserve the accuracy, most of binary neural networks use real-valued activations and weights in an input layer. Therefore, a floating point convolution engine is needed without reusing operations of XNOR and POPCOUNT[35]. There are two reasons for the sharp decline of accuracy after binarizing an input layer. The first reason is the lack of input channels. The input of CNNs contains generally RGB three channels, and some may even be a single channel grayscale image. Such a limited number of input channel is insufficient for binarization. The other reason is the information loss of binarized activations. For vision tasks, the value of the input image is between 0-255. Whether the input is normalized or not, directly turning an input image into a unary feature map results in a considerable loss of information. In this paper, we figure out a BIL coding method that binarizes the activations to a larger channel-wise size without much information loss. As shown in Fig. 4, we first encode the input activation to expand the number of channels. We encode a value between 0-255 by an eight-dimensional binary vector, e.g., translating 118 to 01110110. This increases the number of input channels from one dimension to 8 dimensions. Accordingly, we apply the weight factor 2^i to the corresponding i_{th} channel without retraining. On hardware, this process can be performed efficiently by shifting registers.

3.3 PBCStereo Overview

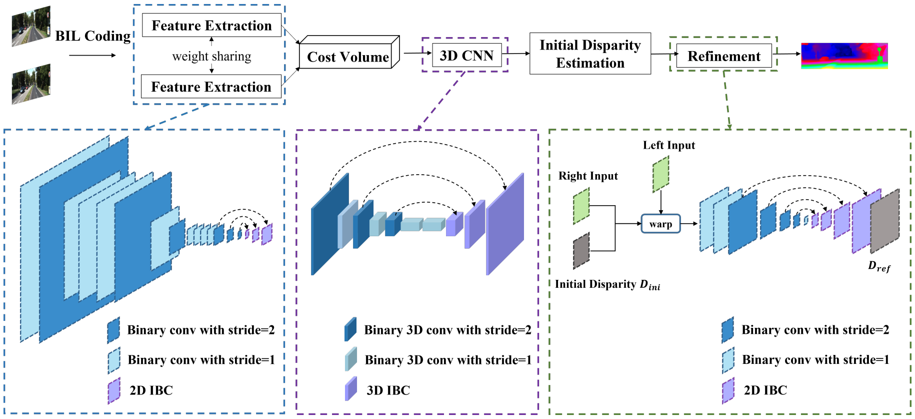

BIL Coding
Feature Extraction
weight sharing
Feature Extraction
Cost Volume
3D CNN
Initial Disparity Estimation
Refinement
Binary conv with stride=2
Binary conv with stride=1
2D IBC
Binary 3D conv with stride=2
Binary 3D conv with stride=1
3D IBC
Left Input
Right Input
warp
Initial Disparity D_{ini}
D_{ref}
Binary conv with stride=2
Binary conv with stride=1
2D IBC

Fig. 5. Architecture overview of PBCStereo.

Based on IBC modules and BIL coding method, we propose a binary end-to-end stereo matching network PBCStereo, as shown in Fig. 5. At a light cost of calculations, PBCStereo receives a reasonable accuracy through increasing the discrimination between features. PBCStereo is the first lightweight stereo network that all convolutional operations are binarized. The pipeline of PBCStereo consists of four steps: feature extraction, cost volume aggregation, initial disparity computation and disparity refinement. We will introduce these modules in detail in the following. For ease of illustration, we define H and W as the height and weight of the input image, and D standing for the maximum disparity.

Feature Extraction. We adopt a Siamese network to extract the image features. The left and right input feature share the weights. For the first three layers, we set the convolution kernel size to 7×7 , 5×5 and 3×3 respectively. The purpose of using larger convolution kernels is to build a larger receptive field and prevent the accuracy decline after BIL encoding. Then, the original input image will be quickly downsampled to $1/64$ resolution after a group of six convolutional layers with strides of 2. Next, features are restored to $1/8$ resolution through upsampling via three 2D IBC modules. Finally, we concatenate 384 feature maps at $1/8$ resolution together to generate a compact cost volume, which greatly saves computations for subsequent networks.

Cost Volume. After feature extraction, the left and right feature maps are both with the size of $384 \times \frac{H}{8} \times \frac{W}{8}$. Therefore, the corresponding maximum disparity should also be adjusted to $\frac{D}{8}$ here. Then, the cost volume is constructed by group-wise correlation[11]. 384 feature channels are divided into 48 groups, and the left feature group is cross-correlated with the corresponding right feature group over all disparity levels. At last, we get a 4D cost volume of $\frac{D}{8} \times \frac{H}{8} \times \frac{W}{8} \times 48$. For this 4D cost volume, we design a 3D binary convolution network via 3D IBC module to complete cost aggregation. Instead of stacking hourglass architectures in PSMNet[4], we only use a U-Net-like structure to obtain an initial guess of disparity D_{ini} at a smaller computational cost, and left the correction of it to the step of disparity refinement.

Disparity Refinement. In order to further make up for the information loss due to the binarization, we design the disparity refinement module as shown in the Fig. 5. Inspired by iResNet[15], the right image is fused with the initial disparity at 1/2 resolution to generate a new synthesized left image. Next, we calculate the difference between the left input image and the synthesized left image, and use this difference to estimate the residual disparity D_{ref} through disparity refinement sub-network. The summation of D_{ini} and D_{ref} is considered as the final disparity.

Disparity Regression. We use *softargmin* proposed in GCNet[13] for disparity regression as,

$$p_d = softmax(-c_d) \tag{6}$$

$$\hat{d} = \sum_{d=0}^{D} d \times p_d \tag{7}$$

p_d is the probability of each disparity d calculated from the softmax of cost volume c_d. $\hat{d}$ denotes the predicted disparity.

Loss function. If the distributions of binary neural networks are more similar to that of real-valued networks, the performance will be improved[18]. Inspired by knowledge distillation[12], we take the real-valued network as the teacher network to guide the distribution of binary student network. However, we don't employ layer-wise distillation here, because the optimization of BNNs is challenging[25,17], and layer by layer distillation will further make the model difficult to converge. So we choose c_d and D_{ref} for distillation. The loss function of PBCStereo is defined as,

$$L(d, \hat{d}) = \alpha \cdot \frac{1}{N} \sum_{i=1}^{N} smooth_{L_1}(d_i - \hat{d}_i) + \frac{(1-\alpha)}{2} \cdot (L_p + L_D) \tag{8}$$

in which

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2, & |x| < 0.5, \\ |x| - 0.5, & |x| \geq 0.5. \end{cases} \tag{9}$$

$$L_p = \frac{1}{ND} \sum_{j=1}^{N} \sum_{d_i=0}^{D} \left| c_{d_i}^{R} - c_{d_i}^{B} \right| \tag{10}$$

$$L_D = \frac{1}{N}\sum_{i=1}^{N}\left|D_{ref}^{R} - D_{ref}^{B}\right| \tag{11}$$

N is the total number of labeled pixels. Superscript R and B denotes the real-valued network and binary network respectively.

4 Experiments

To evaluate the performance of PBCStereo, we conduct experiments on SceneFlow, KITTI 2012 and KITTI 2015. The datasets and the experiment settings are introduced in Section 4.1. Furthermore, we perform ablation studies to validate the effectiveness of the proposed IBC module and binary encoding method for the input layer in Section 4.2. Finally, we compare PBCStereo with other published stereo matching algorithms in Section 4.3.

4.1 Experiment Details

Datasets SceneFlow[22] is a large dataset rendered from various synthetic sequences, including FlyingThings3D, Driving and Monkaa. There are 39824 stereo pairs of size 960×540 with dense ground-truth disparity maps. We further divide the whole dataset into 35454 training images and 4370 testing images. The end-point error (EPE) is used as the evaluation metric.

Unlike SceneFlow, KITTI 2012[10] and KITTI 2015[23] are both real-world datasets with street views of size 1240×376, using LiDAR to collect ground-truth disparity maps. KITTI 2012 provides 194 stereo pairs for training and 195 stereo pairs for testing. End-point error in non-occluded areas (Noc) and in total (All) are used as the evaluation metric. For KITTI 2015, it consists of 200 training scenes and 200 testing scenes. The percentage of stereo disparity outliers (D1) is reported.

OPs Calculation We count the binary operations (BOPs) following [27,19]. The floating point operations(FLOPs) caused by the BatchNorm are also listed. The total operations(OPs) is calculated as $OPs = \frac{BOPs}{64} + FLOPs$. In this paper, we take the input size of 512×256 as the standard for data analysis.

Implementation Details Our PBCStereo is implemented using PyTorch on NVIDIA RTX 2080Ti GPU. We train our models by Adam optimizer with β_1=0.9 and β_2=0.999.

The maximum disparity D is set to 192 and the batch size is set to 12. The input is a grayscale image without normalization. During training, we randomly crop the input to size 512×256 for data augmentation. For SceneFlow, we adjust the learning rate with the cosine annealing[20], setting the maximum value of 0.001. We train our network for 50 epochs in total on SceneFlow. For KITTI, we finetune our network for 400 epochs based on the model which is pretrained on SceneFlow. α is set to 0.8. Moreover, we repeat training for three times to submit the final model with the best performance.

4.2 Ablation Study

Ablation study for IBC Module To validate the effectiveness of IBC modules, we first replace it with ordinary binary deconvolution in our PBCStereo model. After the replacement, the end-point error decreases from 1.83 to 1.71 for SceneFlow. For KITTI 2015, the validation error decreases from 3.79 to 3.40. The evaluation results of our real-valued teacher network are also listed in Table 1. With the help of IBC modules, PBCstereo achieves 39× reduction in OPs with an acceptable cost in accuracy compared with the real-valued version. In Fig.6, qualitative results on Middlebury 2014 and ETH3D show that our IBC module produces more distinct object boundaries than binary deconvolution.

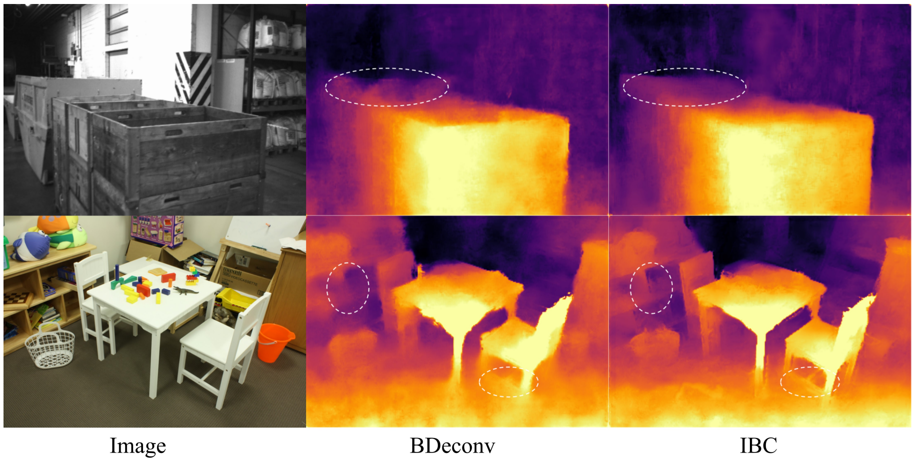


Fig. 6. Qualitative results on Middlebury 2014 and ETH3D. Note that the results are generated by our SceneFlow trained model without any fine-tuned training.

Table 1. Ablation study results on SceneFlow and KITTI 2015. The input layers are all real-valued here. We evaluate the EPE both on SceneFlow validation set and test set. We also compute error on KITTI 2015 validation set. The last column show the total amount of operations. BDeconv stands for the binary deconvolution.

Upsampling Method	Network setting		SceneFlow		KITTI 2015	OPs
	Bit-width of Input Layer	Bit-width of Other Layers	EPE (val)	EPE (test)	Val Err (%)	
Deconv	32/32	32/32	1.13	1.13	2.62	22.31G
BDeconv	32/32	1/1	1.90	1.83	3.79	0.65G
IBC	32/32	1/1	1.76	1.71	3.40	0.66G

Further, we change the specific structure of the IBC module through experiments to analyze why our design can bring performance improvement. Each time we change just one of the following items of IBC module, *i.e.*, removal of skip-connection, 4× the resolution of feature map through interpolation, and joining the aggregated $Downsample_{N-1}$ with $Downsample_{N}$. Testing results of these modified models in Table2 reveal that none of them has achieved the accuracy as good as the base module. Based on the observation, we conclude that skip-connection is the crucial structure in IBC module to enrich the feature map representation, because it not only expands the channel dimension, but also assembles context information from more layers. Although aggregating $Downsample_{N-1}$ with $Downsample_{N}$ achieves comparable accuracy with the proposed IBC module, this structure is at the cost of increasing model size and OPs.

Table 2. Evaluation of different structural settings of IBC module on SceneFlow, comparing to our base module shown in Fig.3.

Settings	ΔModel Size	ΔOPs	ΔEPE
without skip-connection	-97KB	-28M	-0.67
4× interpolation	-127KB	+91M	-0.10
$Downsample_{N-1}$	+40KB	+80M	-0.03

In addition, we also binarize two popular stereo networks of BGNET[33] and DeepPruner[8] through the method of ReactNet[18]. All convolutional layers except the input layer are binarized. For validation, we embed our IBC module to replace the binary deconvolution. As demonstrated by the quantitative results on SceneFlow in Table 3, the binary BGNET and DeepPruner with IBC modules present better performance than those with ordinary binary deconvolution.

Table 3. Ablation study results of IBC modules embedded into BGNET and DeepPruner on SceneFlow.

Method	Bit-width of Input Layer	Bit-width of Other Layers	Upsample	EPE
BGNET[33]	32/32	1/1	BDeconv	2.31
		1/1	IBC	2.23
DeepPrunerFast[8]	32/32	1/1	BDeconv	10.98
		1/1	IBC	10.30

Ablation study for BIL To validate the effectiveness of the proposed BIL coding method, we adopt four methods to binarize the input layer differently in PBCStereo. Dorefa-net[36], IRNet[26] and ReactNet[18] adjust the data distribution of the input layer in different ways, but the lack of input channels leads to accuracy drop. FracBNN[35] uses thermometer encoding to expand input dimension for the input layer. Although it has achieved good performance on CIFAR-10 dataset, thermometer encoding presents nonnegligible information loss in the tasks of pixel-level stereo matching. Compared with these methods, our BIL coding is quite efficient in this case, as shown in Table 4.

Table 4. Evaluation of different methods binarizing the input layer on SceneFlow. ΔEPE denotes the error caused by binarizing the input layer, comparing to our base model.

Method	Bit-width of Input Layer	EPE	ΔEPE
Base	32/32	1.71	-
Dorefa-net[36]	1/1	10.52	-8.81
IRNet[26]	1/1	7.82	-6.11
ReactNet[18]	1/1	6.22	-4.51
FracBNN(R=8)[35]	1/1	2.58	-0.87
Ours	1/1	1.84	-0.13

Ablation study for Loss Weight We use the real-valued teacher model to guide the training of binary student model. The loss function consists of three parts, and the hyperparameter α controls the contribution of three parts to the final loss. As shown in Table 5, we conducted experiments with different values of α. We set $\alpha = 1$ as the baseline without distillation. When $\alpha = 0.8$, our model achieves the best performance with 3.43% error on KITTI 2015 validation set. We adopt the best model and submit the results to KITTI.

Table 5. Influence of the hyperparameter α on KITTI 2015 validation set.

Loss weight		KITTI 2015 Val Error(%)
α	(1-α)/2	
1	0	3.74
0.9	0.05	3.49
0.8	0.1	3.43
0.7	0.15	3.55
0.6	0.2	3.52
0	0.5	4.04

4.3 Evaluations on Benchmarks

We evaluate PBCStereo on SceneFlow and KITTI benchmark against competing algorithms to prove the effectiveness of our model. As shown in Table 6 and Table

7, PBCStereo exhibit a good balance between accuracy and computational cost. Pure binary convolutional operations makes PBCStereo take only 0.57G OPs for depth estimation. It even outperforms some real-valued networks such as StereoNet[14] and LWANet[9]. The qualitative results of KITTI 2015 and KITTI 2012 are shown in Fig. 7.

Table 6. Quantitative evaluation results on KITTI 2015 benchmark and SceneFlow benchmark. For KITTI 2015, we report the percentage of pixels with end-point error more than three pixels, including background regions(D1-bg), foreground regions(D1-fg) and all regions(D1-all). Note that the model size is the number of bytes required to store the parameters in the trained model.

Method	Kitti 2015			SceneFlow	Model Size	OPs
	D1-bg	D1-fg	D1-all	EPE		
PSMNET[4]	1.86	5.62	2.32	1.09	20.4MB	257.0G
AANet[34]	1.99	5.39	2.55	0.87	15.6MB	159.70G
Content-CNN[21]	3.73	8.58	4.54	-	1.3MB	157.30G
MADnet[29]	3.75	9.20	4.66	-	14.5MB	55.66G
SGM-Net[28]	2.66	8.64	3.66	-	450KB	28.0G
DispNet[22]	4.32	4.41	4.34	1.68	168M	17.83G
StereoNet[14]	4.30	7.45	4.83	1.10	1.41MB	14.08G
BGNet[33]	2.07	4.74	2.51	1.17	11.5MB	13.58G
LWANet[9]	4.28	-	4.94	-	401KB	7.03G
MABNet_tiny[32]	3.04	8.07	3.88	1.66	188KB	6.60G
StereoBit[6]	3.50	-	4.57	-	-	-
Ours(1bit)	4.22	7.28	4.73	1.84	653KB	**0.57G**

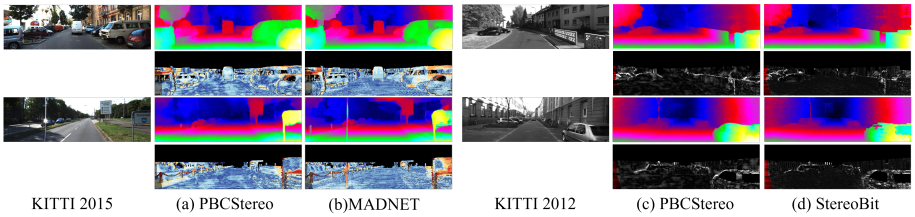


Fig. 7. Qualitative results on KITTI benchmark. On KITTI 2015, we compare our evaluation results with MADNET[29]. On KITTI 2012, we compare our evaluation results with StereoBit[6].

Table 7. Quantitative evaluation results on KITTI 2012. We report the percentage of pixels with end-point error more than two and three pixels, including non-occluded regions(-noc) and all regions(-all). The last column refers to the ratio of OPs of the corresponding algorithm to that of our method.

Method	Kitti 2012						OPs Saving
	2-noc	2-all	3-noc	3-all	EPE(noc)	EPE(all)	
PSMNET[4]	2.44	3.01	1.49	1.89	0.5	0.6	451×
AANet[34]	2.90	3.60	1.91	2.42	0.5	0.6	280×
ContentCNN[21]	4.98	6.51	3.07	4.29	0.8	1.0	276×
SGM-Net[28]	3.60	5.15	2.29	3.50	0.7	0.9	49×
DispNet[22]	7.38	8.11	4.11	4.65	0.9	1.0	31×
StereoNet[14]	4.91	6.02	-	-	0.8	0.9	25×
BGNet[33]	3.13	3.69	1.77	2.15	0.6	0.6	24×
MABNet_tiny[32]	4.45	5.27	2.71	3.31	0.7	0.8	12×
StereoBit[6]	-	-	3.56	4.98	-	-	-
Ours(1bit)	7.32	8.16	3.85	4.46	0.9	1.0	-

5 Conclusion

In this paper, we propose the first compressed stereo network using pure binarized convolutional operations. Our PBCStereo gets 39× saving in OPs with comparable accuracy. To achieve the performance, we propose the IBC module to replace binary deconvolution, improving the upsampling quality for stereo matching. Moreover, we implement BIL encoding in the input layer to resist the usually severe information loss due to the binarization. We are looking forward to realizing PBCStereo in hardware-software co-design, deploying our algorithms on edge devices.

Acknowledgement This work was supported by National Key R&D Program of China with Grant No.2022YFB4400600, and the Fundamental Research Funds for the Central Universities with Grant No.2242022k30052.

References

1. Abu Alhaija, H., Mustikovela, S.K., Mescheder, L., Geiger, A., Rother, C.: Augmented reality meets computer vision: Efficient data generation for urban driving scenes. International Journal of Computer Vision **126**(9), 961–972 (2018) 1
2. Bethge, J., Yang, H., Bornstein, M., Meinel, C.: Binarydensenet: developing an architecture for binary neural networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019) 2
3. Bulat, A., Tzimiropoulos, G.: Xnor-net++: Improved binary neural networks. arXiv preprint arXiv:1909.13863 (2019) 1
4. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5410–5418 (2018) 3.3, 6, 7
5. Chen, G., Ling, Y., He, T., Meng, H., He, S., Zhang, Y., Huang, K.: Stereoengine: An fpga-based accelerator for real-time high-quality stereo estimation with binary neural network. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **39**(11), 4179–4190 (2020) 1
6. Chen, G., Meng, H., Liang, Y., Huang, K.: Gpu-accelerated real-time stereo estimation with binary neural network. IEEE Transactions on Parallel and Distributed Systems **31**(12), 2896–2907 (2020) 1, 6, 7, 7
7. Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. arXiv preprint arXiv:1602.02830 (2016) 2
8. Duggal, S., Wang, S., Ma, W.C., Hu, R., Urtasun, R.: Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4384–4393 (2019) 4.2, 3
9. Gan, W., Wong, P.K., Yu, G., Zhao, R., Vong, C.M.: Light-weight network for real-time adaptive stereo depth estimation. Neurocomputing **441**, 118–127 (2021) 1, 4.3, 6
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3354–3361. IEEE (2012) 1, 4.1
11. Guo, X., Yang, K., Yang, W., Wang, X., Li, H.: Group-wise correlation stereo network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3273–3282 (2019) 3.3
12. Hinton, G., Vinyals, O., Dean, J., et al.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 **2**(7) (2015) 3.3
13. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. In: Proceedings of the IEEE international conference on computer vision. pp. 66–75 (2017) 3.3
14. Khamis, S., Fanello, S., Rhemann, C., Kowdle, A., Valentin, J., Izadi, S.: Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 573–590 (2018) 4.3, 6, 7
15. Liang, Z., Feng, Y., Guo, Y., Liu, H., Chen, W., Qiao, L., Zhou, L., Zhang, J.: Learning for disparity estimation through feature constancy. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2811–2820 (2018) 3.3

16. Liu, C., Ding, W., Xia, X., Zhang, B., Gu, J., Liu, J., Ji, R., Doermann, D.: Circulant binary convolutional networks: Enhancing the performance of 1-bit dcnns with circulant back propagation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2691–2699 (2019) 1
17. Liu, Z., Shen, Z., Li, S., Helwegen, K., Huang, D., Cheng, K.T.: How do adam and training strategies help bnns optimization. In: International Conference on Machine Learning. pp. 6936–6946. PMLR (2021) 3.3
18. Liu, Z., Shen, Z., Savvides, M., Cheng, K.T.: Reactnet: Towards precise binary neural network with generalized activation functions. In: European Conference on Computer Vision. pp. 143–159. Springer (2020) 2, 3.1, 3.3, 4.2, 4.2, 4
19. Liu, Z., Wu, B., Luo, W., Yang, X., Liu, W., Cheng, K.T.: Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In: Proceedings of the European conference on computer vision (ECCV). pp. 722–737 (2018) 2, 4.1
20. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016) 4.1
21. Luo, W., Schwing, A.G., Urtasun, R.: Efficient deep learning for stereo matching. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5695–5703 (2016) 6, 7
22. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4040–4048 (2016) 4.1, 6, 7
23. Menze, M., Heipke, C., Geiger, A.: Joint 3d estimation of vehicles and scene flow. ISPRS annals of the photogrammetry, remote sensing and spatial information sciences **2**, 427 (2015) 4.1
24. Murray, D., Little, J.J.: Using real-time stereo vision for mobile robot navigation. autonomous robots **8**(2), 161–171 (2000) 1
25. Qin, H., Gong, R., Liu, X., Bai, X., Song, J., Sebe, N.: Binary neural networks: A survey. Pattern Recognition **105**, 107281 (2020) 1, 3.3
26. Qin, H., Gong, R., Liu, X., Shen, M., Wei, Z., Yu, F., Song, J.: Forward and backward information retention for accurate binary neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2250–2259 (2020) 4.2, 4
27. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In: European conference on computer vision. pp. 525–542. Springer (2016) 2, 2, 4.1
28. Seki, A., Pollefeys, M.: Sgm-nets: Semi-global matching with neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 231–240 (2017) 6, 7
29. Tonioni, A., Tosi, F., Poggi, M., Mattoccia, S., Stefano, L.D.: Real-time self-adaptive deep stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 195–204 (2019) 6, 7
30. Wang, J., Duan, Z., Mei, K., Zhou, H., Tong, C.: A light-weight stereo matching network with color guidance refinement. In: International Conference on Cognitive Systems and Signal Processing. pp. 481–495. Springer (2020) 1
31. Xie, B., Liang, Y., Song, L.: Diverse neural network learns true target functions. In: Artificial Intelligence and Statistics. pp. 1216–1224. PMLR (2017) 1
32. Xing, J., Qi, Z., Dong, J., Cai, J., Liu, H.: Mabnet: A lightweight stereo network based on multibranch adjustable bottleneck module. In: European Conference on Computer Vision. pp. 340–356. Springer (2020) 1, 6, 7

33. Xu, B., Xu, Y., Yang, X., Jia, W., Guo, Y.: Bilateral grid learning for stereo matching networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12497–12506 (2021) 4.2, 3, 6, 7
34. Xu, H., Zhang, J.: Aanet: Adaptive aggregation network for efficient stereo matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1959–1968 (2020) 6, 7
35. Zhang, Y., Pan, J., Liu, X., Chen, H., Chen, D., Zhang, Z.: Fracbnn: Accurate and fpga-efficient binary neural networks with fractional activations. In: The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. pp. 171–182 (2021) 3.2, 4.2, 4
36. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160 (2016) 4.2, 4