GA-CSPN: generative adversarial monocular depth estimation with second-order convolutional spatial propagation network

Article *in* Journal of Electronic Imaging · August 2021

DOI: 10.1117/1.JEI.30.4.043019

Zhengyang Lu
Jiangnan University

GA-CSPN: generative adversarial monocular depth estimation with second-order convolutional spatial propagation network

Zhengyang Lu and Ying Chen*
Jiangnan University, Key Laboratory of Advanced Process Control for Light Industry,
Ministry of Education, Wuxi, China

Abstract. Monocular depth estimation, which provides a critical method for understanding 3D scene geometry, is an ill-posed problem. Recent research studies have achieved significant progress by reliable network architecture and optimized constraints, such as spatial propagation network and depth metrics. We propose an effective generative adversarial network for fast and accurate monocular depth estimation. Our approach demonstrates the feasibility of applying a dense-connected UNet for reducing information transmission loss and then fine-tuning the blur depth by the high-order convolutional spatial propagation network (CSPN) that used a modified loss function of discriminator. Furthermore, we modify the loss function of discriminator by adding the correlation loss that is used to measure the similarity of real and fake labels. Compared with the original CSPN, the high-order CSPN reduces the computation complexity and accelerates the convergence of the generator network by increasing the order of kernel, which emphasizes the weight of kernels in the update formula. With these modifications, our generative adversarial second-order convolutional spatial propagation network (GA-CSPN) achieves more accurate results against state-of-the-art methods in both indoor and outdoor scenes on Make3D, KITTI 2015, and NYUv2 datasets. © *2021 SPIE and IS&T* [DOI: 10.1117/1.JEI.30.4.043019]

Keywords: monocular depth estimation; generative adversarial network; convolutional spatial propagation network.

Paper 210199 received Apr. 15, 2021; accepted for publication Aug. 3, 2021; published online Aug. 17, 2021.

1 Introduction

Deep learning for computer vision has shown significant progress in depth estimation enabled by the availability of large-scale labeled datasets. Estimating high-quality depth from RGB images is a critical way for understanding 3D real-world scene because it could provide an inexpensive solution in self-driving cars, portable applications, and AR-compositing instead of LIDAR sensors. Therefore, for the solution to monocular depth estimation, which aims to estimate a relative depth from a single RGB image, several approaches have been proposed to construct a reliable pixel-to-pixel mapping. Though a single image cannot provide a reliable numerical solution on 2D-to-3D problem, data-driven prior for deep learning is a feasible method of monocular depth estimation.[1,2]

Classical model-based algorithm can predict depth from stereo images or from videos, limited by the rationality of the model. From the beginning of model-based depth estimation, fully supervised methods have shown themselves capable of fitting predictive models that exploit the relationship between RGB images and their corresponding depth, such as depth-transfer,[3] deep ordinal regression network (DORN),[4] and convolutional spatial propagation network (CSPN).[5] Liu et al.[6] proposed to learn the affinity through a deep CNN with spatial propagation networks (SPN), yielding better results compared with the traditional manually designed affinity. However, depth refinement commonly needed a local affinity matrix rather than a global one.

*Address all correspondence to Ying Chen, chenying@jiangnan.edu.cn

Therefore, to improve SPN, Cheng et al.[5] proposed CSPN, where the depths at all pixels are updated simultaneously within a local affinity.

More recently, self-supervised depth estimation aims to infer a dense depth image from a single RGB image. For the dense depth, several self-supervised approaches have shown that it is possible to train monocular depth estimation models using only synchronized stereo pairs or continuous image sequences, such as Monodepth2.[7] Then, generative adversarial networks (GANVO)[8] predicted 6-DoF pose camera motion and monocular depth map of the scene from unlabeled RGB image sequences, using deep convolutional generative adversarial networks (GANs).[9]

GAN optimizes the network parameters into more accurate value through the game strategy. CSPN[5] is proposed to learn the affinity through a CNN for refining the details of the depth map. To further improve the accuracy of depth estimation, it is a feasible way to combine the CSPN and GANs[9] because of their different optimization directions. Therefore, inspired by CSPN,[5] we construct an end-to-end CNN architecture based on the generative adversarial mechanism called generative adversarial second-order convolutional spatial propagation network (GA-CSPN). The modified propagation formula of CSPN accelerates the convergence without the expense of accuracy. Furthermore, we formulate the correlation loss for the discriminator that not only considers the adversarial loss but also focuses on the distance of discriminator label between fake and real. The GA-CSPN is superior to other existing methods both in accuracy and computational complexity.

Our work provides four contributions, as follows.

1. *Dense UNet for the generator:* Dense-connected UNet (Dense UNet), which introduced the multiple skip-connection mechanism into UNet,[10] is proposed for the generator of GANs. Different from original UNet, Dense UNet has complete information transmission connection between each convolution layer. For the depth estimation task, the skip-connection mechanism improves the recovery of low-dimensional texture features in depth maps, instead of being limited to information recovery of high-level perception features.
2. *Correlation loss for the discriminator:* The proposed correlation loss is applied in the discriminator network of GANs to improve the distribution of network parameter space. Different from common adversarial loss, the correlation loss not only considers the adversarial loss but also focuses on the correlation loss, which is the distance of discriminator label between fake and real. The modification of discriminator increases the discrimination performance of the discriminator and helps to separate the real images and the generated images in a novel way.
3. *High-order CSPN:* Second-order CSPN is a modified propagation formula of CSPN.[5] By increasing the order of kernel, which can emphasize the weight of kernels in the update processing, the formula of high-order CSPN accelerates the convergence under the same number of iterations than the original CSPN without the expense of accuracy.
4. *Performance:* Extensive experiments show that our GA-CSPN not only achieves better accuracy visual improvements against state-of-the-art methods on monocular depth estimation under both indoor and outdoor scenes but also can be deployed in a high speed.

2 Related Work

In the past few years, the deep convolutional neural network (CNN) has been applied to many fields in computer vision with great success. We briefly survey techniques related to our work, namely monocular depth estimation and deep learning method.

2.1 *Supervised Depth Estimation*

Most methods based on deep learning are fully supervised, requiring ground truth depth during training.

Depth estimation is the fundamental for understanding the stereoscopic of scenes from 2D images. From the beginning of depth estimation research, most works focused on geometry-based algorithms[11,12] that rely on point correspondences between images and triangulation.

A current common approach is to construct a depth map by learning non-linear multi-channel to single-channel mapping, implemented as a CNN work.[13]

Because the handcrafted features can only capture limited local information, the deep CNN features and Markov random fields features were used to incorporate the global relationship.[14,15] Another innovative way to use global information is the depth-transfer,[3] which used CNN method for style transfer.

To combine global and local cues, Eigen[1] employed two deep network stacks: one made a coarse global prediction based on the entire image, and another refined this prediction locally. A single multi-scale convolutional network architecture[2] was applied for three different computer vision tasks, namely depth prediction, normal surface estimation, and semantic labeling. Eigen[2] proposed another method, which captured many image details and refined predictions using a sequence of scales.

Furthermore, a deep convolutional neural field model[16] was presented for estimating depths from single monocular images, aiming to jointly explore the capacity of deep CNN and continuous conditional random field (CRF). Then, the deep structured learning scheme,[16] which learns the unary and pairwise potentials of continuous CRF in a unified deep CNN framework, was proposed by Liu.

DORN for monocular depth estimation[4] was proposed to discretize continuous depth into certain intervals and cast the depth network learning as an ordinal regression problem, and presented how to involve ordinal regression into a dense prediction task via deep CNNs.

To solve the problem of occluding contours, Ramamonjisoa and Lepetit[17] proposed SharpNet, a method that predicts an accurate depth map given a single input color image, with particular attention to the reconstruction of occluding contours.

For more effective guidance of densely encoded features to the desired depth prediction, Lee et al.[18] proposed a multi-scale local planar guidance for monocular depth estimation, which is called big to small (BTS), utilizes novel local planar guidance layers located at multiple stages in the decoding phase.

Most recently, Liu et al.[6] proposed to learn the affinity through a deep CNN with SPN, yielding better results compared with the traditional manually designed affinity. However, depth refinement commonly needed a local affinity matrix rather a global one. Therefore, to improve SPN, Cheng et al.[5] proposed CSPN, where the depths at all pixels are updated simultaneously within a local affinity.

2.2 *Self-Supervised Depth Estimation*

In addition to the ground truth depth, another alternative is to train depth estimation models using image reconstruction as supervision. Different from fully supervised method above, Godard et al.[19] constructed an unsupervised monocular depth estimation with left-right consistency called Monodepth. Monodepth built a deep network to predict stereo pairs and predicted the pixel disparities between pairs. Furthermore, Godard et al.[7] proposed a fully resolution multi-scale sampling method called Monodepth2 that reduces visual artifacts, and an auto-masking loss to ignore training pixels that violate camera motion assumptions.

Almalioglu et al.[8] proposed an unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks (GANVO) that predicted 6-DoF pose camera motion and monocular depth map of the scene from unlabeled RGB image sequences, using deep convolutional GANs.[9]

3 Method

To use an efficient model to estimate the depth map from an unconstrained true-color image, we present a generative adversarial monocular depth estimation with second-order GA-CSPN for robust fast depth prediction. As shown in Fig. 1, a depth estimation block based on the Dense UNet is designed to fit the generator network in GAN architecture,[9] and the VGG classification[20] is chosen to be the discriminator network to improve the optimization result. As illustrated in Fig. 2, following the backbone of dense-connected UNet in the generator network,

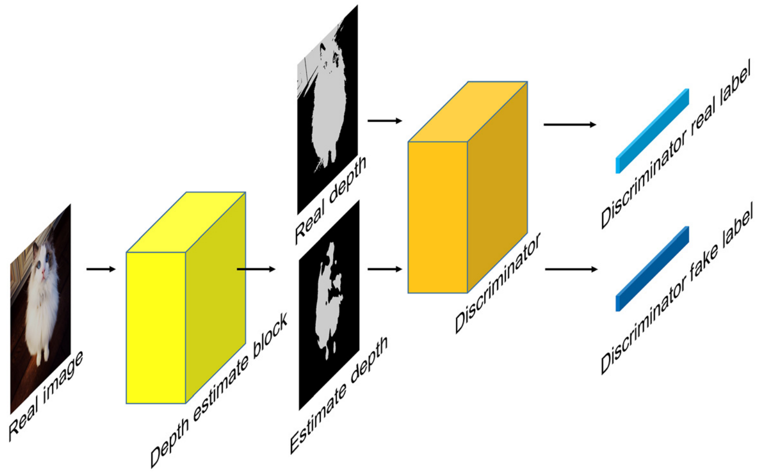


Fig. 1 The GAN architecture contains the depth estimation block and the discriminator block.

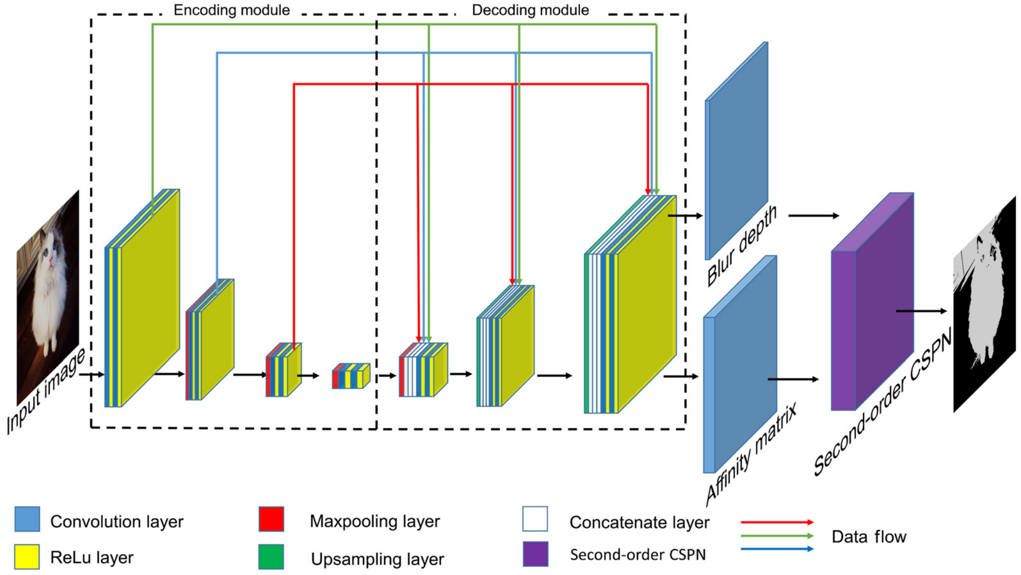


Fig. 2 The architecture of generator network contains a Dense UNet backbone and second-order CSPN enhancement.

we reformulate fast depth estimation as an anisotropic diffusion process called CSPN, and the diffusion tensor is learned from the given image through a deep CNN, which guides the refinement of the depth prediction. Compared with the original CSPN, the high-order CSPN accelerates the convergence of the network without the expense of computation complexity.

3.1 *Generative Adversarial Networks*

In our method, the GAN[9] is selected as the depth estimation architecture to encourage the network to favor solutions with more details and sharp edges.

First, the Dense UNet, designed to be the generator network, is applied to reconstruct depth maps from true-color images. Correspondingly, the generator network is adopted for generating images to deceive the discriminator network, thereby improving the accuracy of the classification network to distinguish generated images from ground truths. Then, from many feature classification networks, we select the VGG network[20] as the discriminator because of its deep network structure and reliable pre-trained model. In addition to the discriminator structure, a novel correlation loss is proposed for replacing the adversarial loss.

3.2 *Generator with Dense-Connected UNet*

The original UNet,[10] an end-to-end segmentation architecture, was chosen as the backbone of the GAN generator network. However, the original UNet had two shortages: the information loss from max-pooling operation in downsampling, and the other is the inadequate skip-connections between layers from the same depth.

To avoid the above problems caused by original UNet architecture, we propose a more effective architecture called dense-connected UNet (Dense UNet). Unlike original UNet,[10] dense UNet has multiple skip-connections between each layer from all depth, shown as data flow in Fig. 2. Because the skip-connection from multi-layers has different size, the data flow includes the operation of bilinear interpolation for upsampling and max-pooling for downsampling.

The multiple skip-connections are constructed for transferring feature maps from all depth of encoding blocks into every decoding block. Because the upsampling block combines all downsampling feature maps from all depth of encoding blocks instead of feature maps from the one-way downsampling path, theoretically the dense skip connection establishes a multi-path data transmission and reduces the information transmission loss.

The loss of generator L_G combines the MSE loss L_{MSE} with the adversarial loss L_{GAN} and shows below:

$$\begin{aligned} L_G &= L_{\mathrm{MSE}} + \lambda L_{\mathrm{GAN}} \\ L_{\mathrm{MSE}} &= \mathrm{MeanSquareError}(X_F, X_R) \\ L_{\mathrm{GAN}} &= \mathrm{BinaryCrossEntropy}(D(X_F), I) = -\log(\sigma(C(X_F))), \end{aligned} \tag{1}$$

where λ is the coefficient to balance different loss terms, σ is the sigmoid function, and $C(x)$ is the non-linear transformed discriminator output.

The MSE loss L_{MSE} is set to improve the accuracy of estimated results by reducing the distance between estimated depth X_F and real depth X_R. To further deceive the discriminator, the adversarial loss for generator L_{GAN} represents the distance between fake discriminator labels $D(X_F)$ and all-one labels I. All-one labels I are $1 \times N_o$ matrix and N_o is the number of discriminator features.

Accordingly, the integrated generator loss L_G combines the L_{MSE} with L_{GAN}.

3.3 *Correlation Loss of Discriminator*

Apart from correlation average GANs, our correlation loss of discriminator directly compared real discriminator labels and fake discriminator labels, which further distinguishes the estimated depth and real depth.

As shown in Fig. 1, the depth estimation module aims to predict the depth map based on the real image, and the discriminator aims to distinguish the estimated image and the ground truth. The estimated image X_F generated by the pre-trained depth estimation module is input to the discriminator to obtain fake discriminator labels $D(X_F)$ in the training process. Similarly, the real image X_R is input to the discriminator to obtain real discriminator labels $D(X_R)$.

$$D(X_R) = \sigma(C(X_R)) \to 1 D(X_F) = \sigma(C(X_F)) \to 0, \tag{2}$$

where σ is the sigmoid function and $C(x)$ is the non-linear transformed discriminator output.

From the above formulas, real label loss L_R and fake label loss L_F are shown below:

$$\begin{aligned} L_R &= \mathrm{BinaryCrossEntropy}(D(X_R), I) = -\log(\sigma(C(X_R))) \\ L_F &= \mathrm{BinaryCrossEntropy}(D(X_F), O) = -\log(1 - \sigma(C(X_F))), \end{aligned} \tag{3}$$

where I is an all-one matrix whose size is the same as real labels $D(X_R)$ and O is a matrix of zeros whose size is the same as fake labels $D(X_F)$.

The real label loss L_R measures the distance between real discriminator label $D(X_R)$ and absolute real label I. The fake label loss L_F measures the distance between fake discriminator label $D(X_R)$ and absolute fake label O.

In addition to the traditional adversarial loss for discriminator in GAN, the special correlation loss is proposed to measure the distance between $D(X_F)$ and $D(X_R)$.

$$\begin{aligned} L_{FR} &= \text{BinaryCrossEntropy}(D(X_F), D(X_R)) = -\sigma(C(X_R)\log(\sigma(C(X_F))) \\ &\quad - (1-\sigma(C(X_R))\log(1-\sigma(C(X_F))) \\ L_{RF} &= \text{BinaryCrossEntropy}(D(X_R), D(X_F)) = -\sigma(C(X_F)\log(\sigma(C(X_R))) \\ &\quad - (1-\sigma(C(X_F))\log(1-\sigma(C(X_R))), \end{aligned} \tag{4}$$

where L_{FR} is the loss between fake discriminator label $D(X_F)$ and real discriminator label $D(X_R)$. Equally, L_{RF} shows the binary cross entropy loss between real discriminator label $D(X_R)$ and fake discriminator label $D(X_F)$. Both L_{FR} and L_{RF} are applied in the correlation loss due to the asymmetric of binary cross entropy formula.

$$L_{\text{adv}} = L_R + L_F \quad L_{\text{rel}} = L_{RF} + L_{FR}. \tag{5}$$

As shown above, in addition to the adversarial loss L_{adv}, which combines L_R with L_F, the special correlation loss L_{rel} considers the L_{FR} and L_{RF}.

By combining the special correlation loss and the adversarial loss, the general correlation loss of discriminator L_D shows below:

$$L_D = \sigma(L_{\text{adv}} - L_{\text{rel}}) + L_{\text{reg}} = \sigma(L_R + L_F - L_{FR} - L_{RF}) + L_{\text{reg}}. \tag{6}$$

In the general correlation loss of discriminator L_D, the sigmoid function σ is added to constrain the value since the loss function must be positive. L_{reg}, the regularization term, aims to prevent the overfitting of discriminator.

The schematic diagram of the adversarial loss for discriminator and the proposed correlation loss is illustrated in Fig. 3, which indicates the relationship between each label. The arrow in the figure is used to indicate the direction of optimization direction. Back to the original motivation, the L_F is set to make fake discriminator label $D(X_F)$ close to the absolute fake label O. Similarly, the L_R is set to make real discriminator label $D(X_R)$ close to the absolute real label I. The L_{RF} and L_{FR} are set to increase the distance between $D(X_R)$ and $D(X_F)$. Therefore, in general correlation loss, the L_{FR} and L_{RF}, which is the direct distance between $D(X_F)$ and $D(X_R)$, should be maximized. Meanwhile, the L_F and L_R should be minimized.

Figure 3(a) shows an initial condition of discriminator label before optimization. As in situations shown in Fig. 3(b), the constrain of the adversarial loss L_{adv} pushes $D(X_F)$ and $D(X_R)$ to label O and label I, respectively, which results in reduced distance between $D(X_F)$ and $D(X_R)$. Following Fig. 3(c), L_F, L_R, L_{RF}, and L_{FR} can all be optimized with combined constraints of L_{adv} and L_{rel}.

The modification of discriminator increases the performance of discriminator and separates the real images and the generated images effectively.

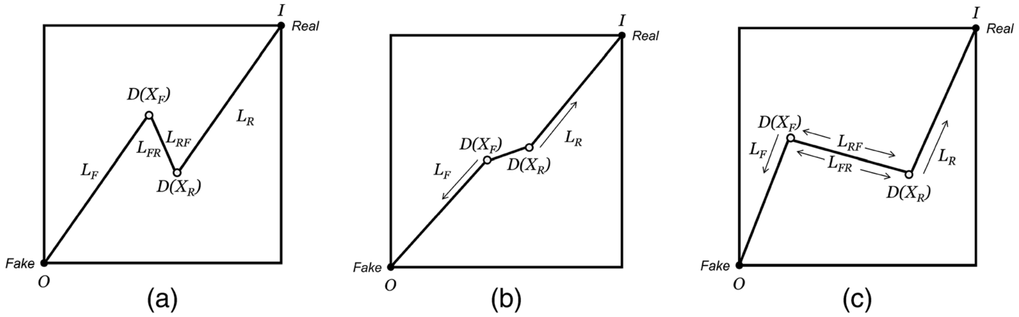


Fig. 3 Schematic diagram of the discriminator label optimized by adversarial loss and correlation loss: (a) initial condition, (b) optimized by L_{adv}, and (c) optimized by L_{adv} and L_{rel}.

Algorithm 1 GAN with high-order CSPN.

```
Require: $G(x)$: Generator network; $D(x)$: Discriminator network; $S$: Source; $X$: Depth;
Ensure: optimal $G(x)$ and $D(x)$
Randomly initial weight of $G(x)$ and $D(x)$;
for $k$ iterations do
    Compute the output of generator network;
        Estimated depth: $X_F = G(S)$
    Compute the label of discriminator network;
        Fake discriminator label: $D(X_F)$
        Real discriminator label: $D(X_R)$
    Compute the loss of generator network:
        $L_G = L_{\mathrm{MSE}} + \lambda L_{\mathrm{GAN}}$
    Compute the loss of discriminator network:
        $L_{\mathrm{adv}} = L_R + L_F$
        $L_{\mathrm{rel}} = L_{RF} + L_{FR}$
        $L_D = \sigma(L_{\mathrm{adv}} - L_{\mathrm{rel}})$
    Update the generator network by adaptive moment estimation optimizer:
        $\theta_G \coloneqq \arg\min_{\theta_G} L(G)$
    Update the discriminator network by stochastic gradient descent optimizer:
        $\theta_D \coloneqq \arg\min_{\theta_D} L(D)$
end for
```

Algorithm 1 explains the generative adversarial learning strategy with high-order CSPN. As shown in Algorithm 1, the generator network of GAN with high-order CSPN is optimized by adaptive moment estimation optimizer and the discriminator network is optimized by stochastic gradient descent optimizer.

3.4 *High-Order Convolutional Spatial Propagation Network*

A high-order CSPN[5] is proposed for accelerating the propagation process by emphasizing the weight of kernels in the update formula.

In CSPN, an affinity matrix is used to organize the mutual similarities between a set of images, which can be interpreted as the probability that two points are related. Therefore, the depth estimate can be further refined by estimating the affinity matrix.

The output of the estimate backbone, Dense UNet, and the image $X \in \mathbb{R}^{m\times n}$ is a blur depth map $D_0 \in \mathbb{R}^{m\times n}$. To obtain the clear depth map D_N, the blur depth map needs to be updated within N iterations.

The original CSPN, which is the first-order CSPN, is a specific form of the extended CSPN. The extended CSPN includes the first-order CSPN and high-order CSPN. In extended CSPN, D_0 is a blur observation of the hidden space $H \in \mathbb{R}^{m\times n\times c}$, which is the ground truth. Therefore, through the iteration of the propagation formula, the blur depth D_N will be approximate to the hidden space H. The convolutional update formula with a kernel size of k for each time step t is as shown below:

$$
\begin{aligned}
H_{i,j,t+1} &= \sum_{a,b=-(k-1)/2}^{(k-1)/2} k_{i,j}(a,b) \odot H_{i-a,j-b,t} \\
\text{where, } \; k_{i,j}(a,b) &= \frac{\hat{k}_{i,j}^{N}(a,b)}{\sum_{a,b,a,b\neq 0}(\hat{k}_{i,j}^{N}(a,b))} \\
k_{i,j}(0,0) &= 1 - \sum_{a,b,a,b\neq 0} k_{i,j}^{N}(a,b),
\end{aligned} \tag{7}
$$

where the transformation kernel $\hat{k}_{i,j} \in \mathbb{R}^{k\times k\times c}$ is the output from the affinity matrix. The kernel size k is usually set as an odd number so that the computational context surrounding pixel (i, j) is symmetric. $\odot$ is the element-wise product.

The update formula is derived from the original CSPN when N is 1. Different from CSPN, which directly uses the weighted sum of the kernel, the high-order CSPN increases the order of each kernel as N, which emphasizes the weight of kernels in the update formula and accelerates the convergence of the results.

Then propagation formula is reformed as a diffusion evolution by vectorizing the feature map H to $H \in \mathbb{R}^{mn\times c}$, as in CSPN.[5]

$$
H^{t+1} = \begin{bmatrix} 1-\lambda_{0,0} & k_{0,0}(1,0) & \cdots & 0 \\ k_{1,0}(-1,0) & 1-\lambda_{1,0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \cdots & \cdots & 1-\lambda_{m,n} \end{bmatrix} = GH^{t}, \tag{8}
$$

where $\lambda_{i,j} = \sum_{a,b} k_{i,j}^{N}(a,b)$ and G is a $mn \times mn$ transformation matrix. The reason why we use convolutional operation is that it can be directly implemented by image vectorization in PyTorch. Thus, the diffusion process is derived as a partial different equation:

$$
H_{t+1} = GH_{t} = (I - D + A)H_{t} \quad H_{t+1} - H_{t} = -(D - A)H_{t} \quad \partial_{t}H_{t+1} = -LH_{t}, \tag{9}
$$

where L is the Laplacian matrix, D is the diagonal matrix of $\lambda_{i,j}$, and A is the affinity matrix, which is the off-diagonal part of G.

4 Experiment

In this section, we describe the implementation details, the datasets information, and benchmark performance of our experiments. The GA-CSPN outperforms other exist monocular depth estimation methods and can be deployed as a high-speed application.

4.1 *Implementation Details*

In our method, the generator network is trained by Adam optimizer[21] with $\beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The learning rate of generator network optimizer is set to 10^{-3} initially and decreases to half every 100 epochs. The weights of VGG network[20] in the feature extraction layers for discrimination are initialized with models pre-trained on the ImageNet dataset.[22] Correspondingly, the discriminator network is trained by stochastic gradient descent optimizer.[23] The learning rate of discriminator network optimizer is set to 10^{-3} in the whole training process. In the training parameter set, the batch of data is set to 1. The PyTorch implement of our method is trained with one RTX 2080 GPU, and this project is proposed online.

4.2 *Datasets*

4.2.1 *Make3D*

The Make3D dataset[14,24] contains 534 outdoor images, 400 images for training, and 134 images for testing. The dataset has images with the resolution of 2272 × 1704 and depth maps with 55 × 305. Same as the pre-processing of most previous methods, we resize the resolution of all the images to 512 × 512 and depth maps to 512 × 512. Same as previous works, the depth map $C1$, which ranges from 0 to 80 m, and $C2$, which ranges from 0 to 70 m, are reported by three commonly used evaluation metrics.[3]

4.2.2 *KITTI 2015*

The KITTI depth prediction dataset[25] consists of 42,949 training frames, 1000 validation samples, and 500 test samples, annotated with sparse point clouds. Same as the pre-processing of most previous methods, the input images are cropped to 352 × 1216 during training and testing. Following previous works, the depth map is evaluated by REL, SqREL, RMS, $\mathrm{RMS}_{\mathrm{log}}$, and three accuracy metrics with different thresholds.

4.2.3 *NYU depth v2*

The NYU Depth v2 dataset[26] contains 464 indoor video scenes captured by a Microsoft Kinect camera. GA-CSPN is trained on all images from 249 training scenes and tested on 694 image testing scenes. To speed up training, images are resized into 512 × 512 while the depth map is resized into 512 × 512. Following previous works, the depth map, which ranges from 0 to 80 m, is evaluated by REL, RMS, $\mathrm{RMS}_{\mathrm{log}}$, and three accuracy metrics with different thresholds.

4.3 *Evaluation Method*

Following previous works, the depth map is evaluated by absolute relative error (REL), root mean square error (RMS), root mean square logarithmic error ($\mathrm{RMS}_{\mathrm{log}}$),[3] and three accuracy metrics with different thresholds (1.25, 1.25^2, and 1.25^3).

4.3.1 *Absolute relative error*

$$\mathrm{REL} = \frac{1}{n}\sum\left|\frac{y_{\mathrm{pred}} - y_{\mathrm{gt}}}{y_{\mathrm{gt}}}\right|, \tag{10}$$

where y_{pred} is the depth estimated by network and y_{gt} is the ground truth.

4.3.2 *Square relative error*

$$\mathrm{SqREL} = \frac{1}{n}\sum\left(\frac{y_{\mathrm{pred}} - y_{\mathrm{gt}}}{y_{\mathrm{gt}}}\right)^2. \tag{11}$$

4.3.3 *Root mean square error*

$$\mathrm{RMS} = \sqrt{\frac{1}{n}\sum(y_{\mathrm{pred}} - y_{\mathrm{gt}})^2}. \tag{12}$$

4.3.4 *Root mean square logarithmic error*

$$\mathrm{RMS}_{\log} = \sqrt{\frac{1}{n}\sum(\log(y_{\mathrm{pred}}) - \log(y_{\mathrm{gt}}))^2}. \tag{13}$$

4.3.5 *Accuracy*

Percentage of depth y where the ratio of estimated and ground truth depth is less than a threshold:

$$\max\left(\frac{y_{\mathrm{pred}}}{y_{\mathrm{gt}}}, \frac{y_{\mathrm{gt}}}{y_{\mathrm{pred}}}\right) = \delta < \text{threshold}. \tag{14}$$

4.4 *Ablation Experiment*

4.4.1 *Hyper-parameter setting*

In the generator part, the loss of generator L_G combines the L_{MSE} with L_{GAN}. Therefore, the weight of L_{GAN} is a critical hyper-parameter that requires to be determined. Figure 4 shows the accuracy of GA-CSPN with UNet under different λ values. It can be observed from Fig. 4 that the accuracy reaches the optimal value in RMS and REL when λ equals 10^{-3}. Thus, λ is set as $1e^{-3}$ in following experiments.

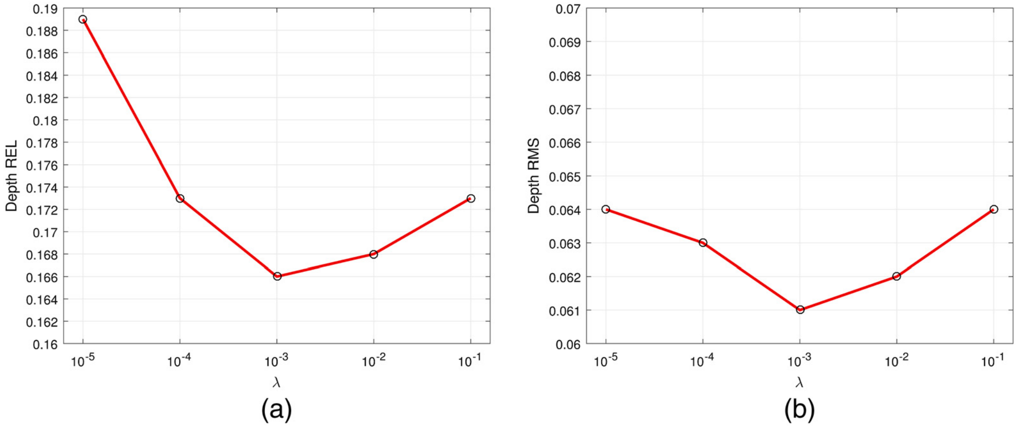
4.4.2 *Order selection*

Figure 5 shows the accuracy of CSPN and GA-CSPN under different orders. CSPN is a simple and effective convolutional spatial propagation network with the backbone of UNet, and GA-CSPN is the generative adversarial convolutional spatial propagation network with the backbone of UNet.

From Fig. 5, with the increase of iteration number, the absolute relative errors of high order CSPN are reduced to 0.1810, and the RMSs are reduced to 0.6460.

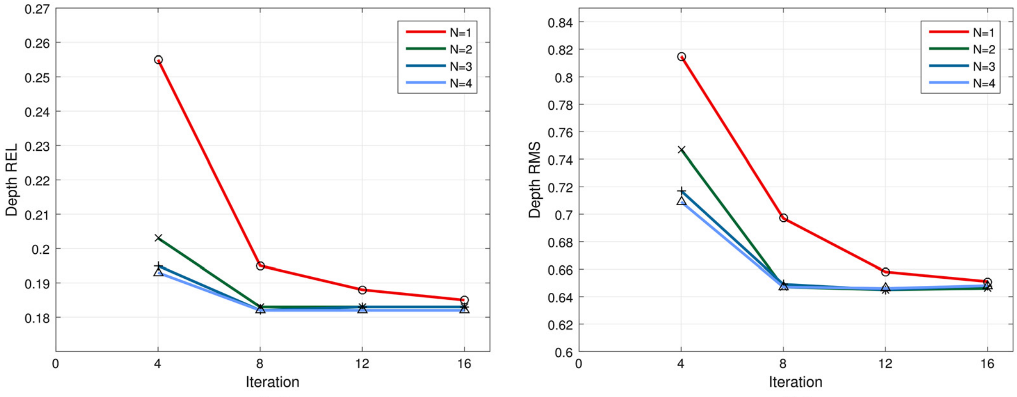
With the iteration increase from 4 to 8, the improvement in accuracy of CSPN comes from the iterative process that fails to converge when iteration is four. The original CSPN where $N = 1$ reached the convergence after 16 iterations, whereas the high-order CSPN where $N \geq 2$ starts convergence after 8 iterations and maintains a stable value after that. In above experiments, it can be seen that extended CSPN has better convergence performance than CSPN.

The running times of extended CSPN with different orders are provided in Table 1 when iteration is 8. As can be seen in Table 1, the high-order CSPN where $N \geq 2$ has almost the



(a) (b)

Fig. 4 Accuracy comparison of λ on NYUv2 dataset: (a) REL and (b) RMS.



(a)

(b)

Fig. 5 Accuracy comparison of different order number on NYUv2 dataset: (a) REL and (b) RMS.

Table 1 Running time comparison with orders and image sizes.

Order	512 × 512	800 × 800	1024 × 1024
1	139.035 ms	405.268 ms	864.160 ms
2	139.192 ms	405.933 ms	870.152 ms
3	139.308 ms	406.705 ms	879.517 ms
4	139.624 ms	407.081 ms	893.802 ms

same running speed, whose time only increases less than 0.2% in each order in CSPN. In other words, the complexity of the overall network is almost unchanged when the order increases.

Combining Fig. 5 and Table 1, the original CSPN where $N = 1$ does not converge when iteration is 8, but high-order CSPN converges in the same condition. More vital information obtained from Fig. 5 and Table 1 is that the second-order is enough to reach the optimal value, so 2 is chosen as the final order number. Consequently, following high-order GA-CSPN is referred to as second-order GA-CSPN.

4.4.3 *Ablation results*

To further illustrate the effectiveness of strategies of the method, ablation experiments for extended CSPN are taken, whose results are shown in Table 2. In following experiments, the base model is the original CSPN with the backbone of Unet.

From Table 2, it can be observed that each module has an explicit improvement on model performance. Major accuracy improvements come from the generative adversarial mechanism and the Dense UNet backbone. The REL is decreased about 0.3 by generative adversarial mechanism and about 0.470 by Dense UNet backbone. The best combination is applying four modules because all modification modules have improvements in accuracy. Furthermore, the combination of all modules outperforms the others and reduce the REL to 0.1130. Further analysis for each module is shown below.

1. *Generative adversarial mechanism:* Because the discriminator only participates in training processing and does not participate in test deployment, the generative adversarial mechanism significantly improves the accuracy by about 15% without increasing computational complexity.
2. *Dense UNet:* By replacing the original UNet with Dense UNet backbone, the REL reduced around 0.047 in CSPN and 0.037 in GA-CSPN with the expense of around 33 ms per frame.

Table 2 Ablation experiments on NYUv2 dataset.

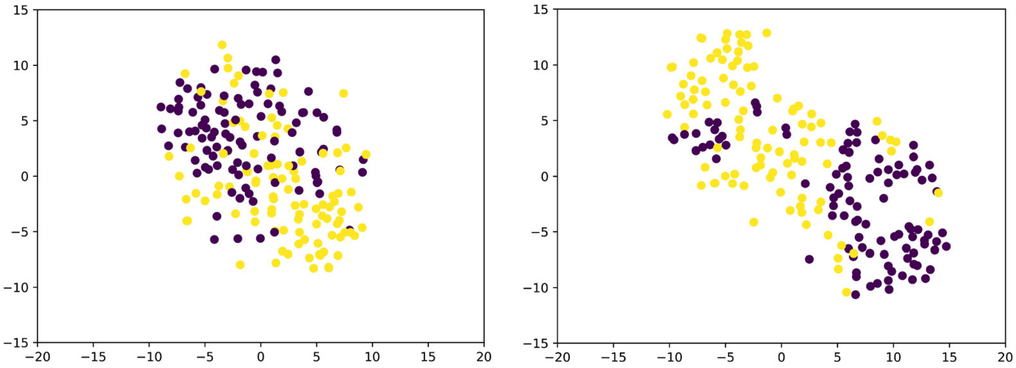
GANs	Dense Unet	Second-order	Correlation loss	Running time (ms)	REL	$\mathrm{RMS}_{\log}$	RMS
			—	139.035	0.1950	0.0640	0.6970
		√	—	139.192	0.1830	0.0600	0.6890
	√		—	171.627	0.1480	0.0560	0.5900
	√	√	—	171.862	0.1350	0.0520	0.5810
√				139.011	0.1660	0.0610	0.6320
√		√		139.175	0.1530	0.0580	0.6230
√			√	139.021	0.1520	0.0570	0.6020
√	√		√	171.678	0.1320	0.0540	0.5480
√	√	√		171.864	0.1260	0.0520	0.5390
√	√	√	√	171.853	0.1130	0.0490	0.5080

3. *Second-order CSPN:* The second-order CSPN has a reduction of around 0.12 on REL without the expense of running time.
4. *Correlation loss:* Without increasing running time, the correlation loss of GA-CSPN outperforms the adversarial loss and reduced the REL of 0.013.

Compared with the original GAN loss, correlation loss directly separates the discriminator fake label and discriminator real label. To demonstrate the effectiveness of correlation loss, T-distributed stochastic neighbor embedding (T-SNE),[27] a machine learning algorithm for dimensionality reduction, is applied to observe the distribution of labels output by the discriminator.

In Fig. 6, yellow points represent the dimensionality reduction label of the real image, and purple points represent the fake image. The results of the above experiments indicate that the discriminator that is optimized by the correlation loss separated the label of the fake image from the whole label more precise and effective than adversarial loss.

From the above experiments, it can be observed that second-order GA-CSPN with correlation loss is the best combination for monocular depth estimation.



(a) (b)

Fig. 6 T-SNE distribution of real and fake labels output by the discriminator: (b) adversarial loss and (b) correlation loss.

4.5 *Comparison with State-of-the-Art Method*

4.5.1 *Qualitative comparison*

Some qualitative comparison with four methods namely depth-transfer,[3] DORN,[4] CSPN,[5] and GA-CSPN are shown in Fig. 7, from which it can be observed that the proposed GA-CSPN well reconstructs local depth details and yields an outstanding visual performance in Make3D dataset.[24]

Due to the low-resolution ground truth of depth maps, not all pixels in high-resolution images can be correspond to the pixels in depth maps. In the Make3D dataset, it can be observed from Fig. 7 that the neighbor pixels in 2D images are tend to be close in depth in all existing methods. For example, the depth boundary between the leaves and the sky is not sharp enough because they are spatially closed in 2D images. Superior to the other methods, the proposed GA-CSPN effectively reconstructs these depths in outdoor scenes and successfully rebuilds the distinct boundary between each object.

Figure 8 shows the constructed depth maps and the corresponding point cloud sets with different methods in NYUv2 indoor dataset. It is noticeable that the proposed GA-CSPN reconstructs local large gradient area well and achieves a more precise result.

Similarly, in Fig. 8, we observe that the neighbor pixels in 2D images are tend to be close in the constructed 3D point cloud in previous works, and it causes the shape distortion of the table

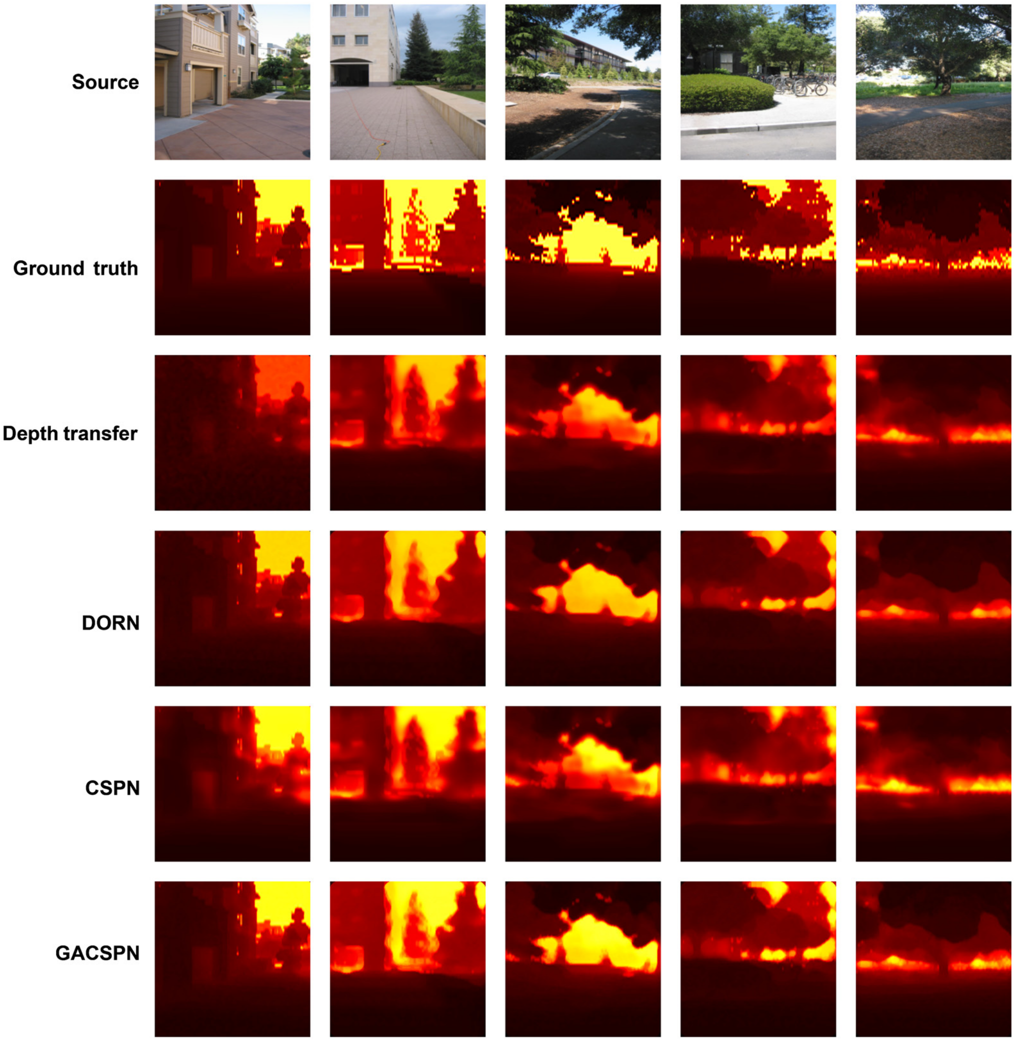


Fig. 7 Qualitative comparison on Make3D dataset.

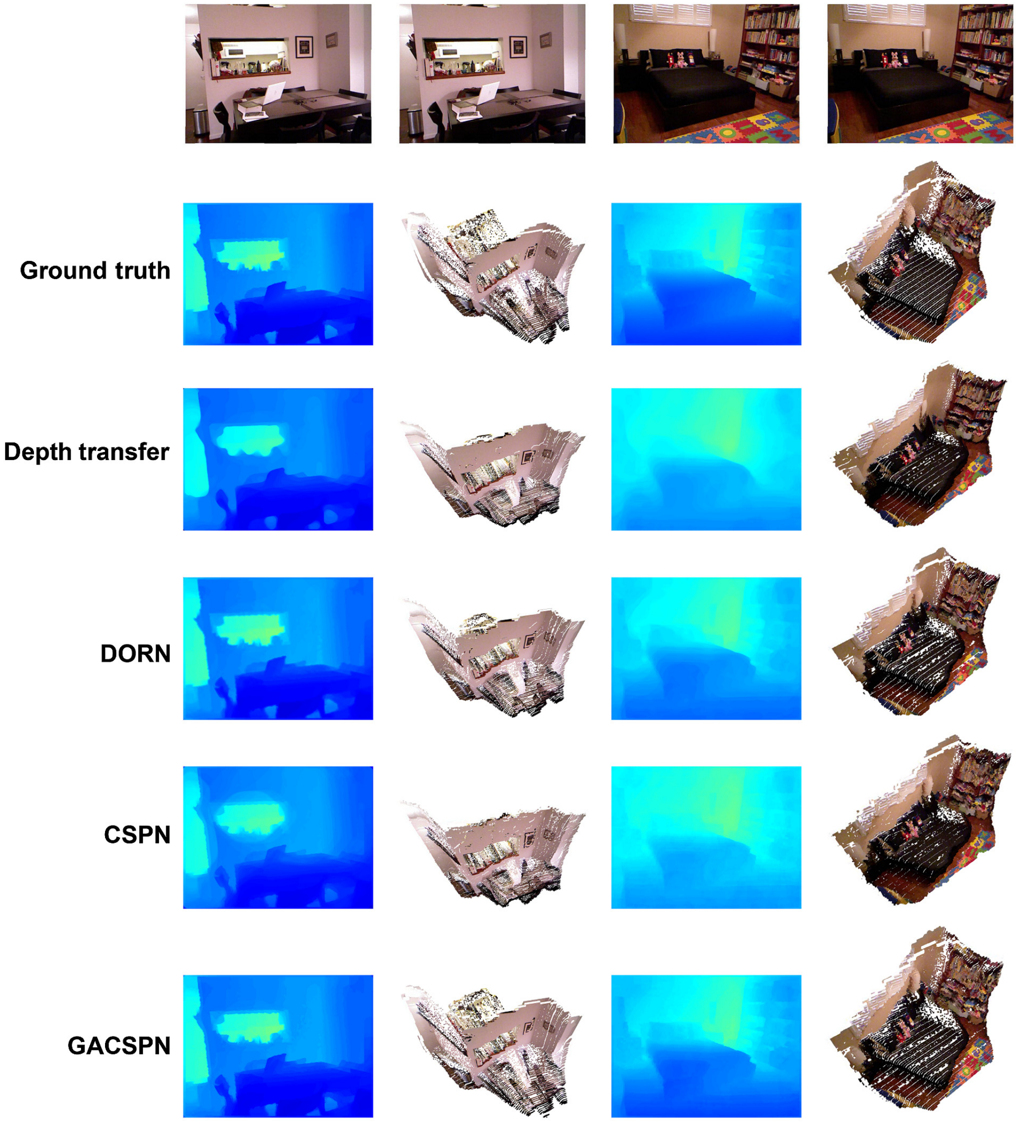


Fig. 8 Qualitative comparison on NYUv2 dataset.

and bed in most methods. The point cloud reconstructions of GA-CSPN maintain the shape of the furniture and resist the uneven lighting from one side.

For depth estimation algorithms, the performances on the KITTI 2015 dataset are the critical benchmark. Because of the large amounts of video sequences in KITTI 2015 dataset and the specific lane scenes, the reconstruction results of model-based methods perform outstanding, which is shown in Fig. 9. Focused on the reconstruction of small objects, we can see that the depth of tree trunks and signposts are ignored or replaced by the background in previous works, whereas the proposed GA-CSPN not only successfully reconstructs a smooth lane but also keeps a high accuracy on small objects.

4.5.2 *Quantitative comparsion*

As shown in Table 3, GA-CSPN outperforms most other methods on Make3D dataset.[24] The REL, $\mathrm{RMS}_{\log}$, and RMS of GA-CSPN all achieve the lowest errors in depth map C1. Meanwhile, the REL and $\mathrm{RMS}_{\log}$ of GA-CSPN both achieve the lowest errors, and the RMS achieves the second-lowest error in depth map C2. Since Make3D is an outdoor dataset, which sets in a specific area, the results in Table 3 illustrate that GA-CSPN performs extraordinarily in outdoor streets and building scenes.

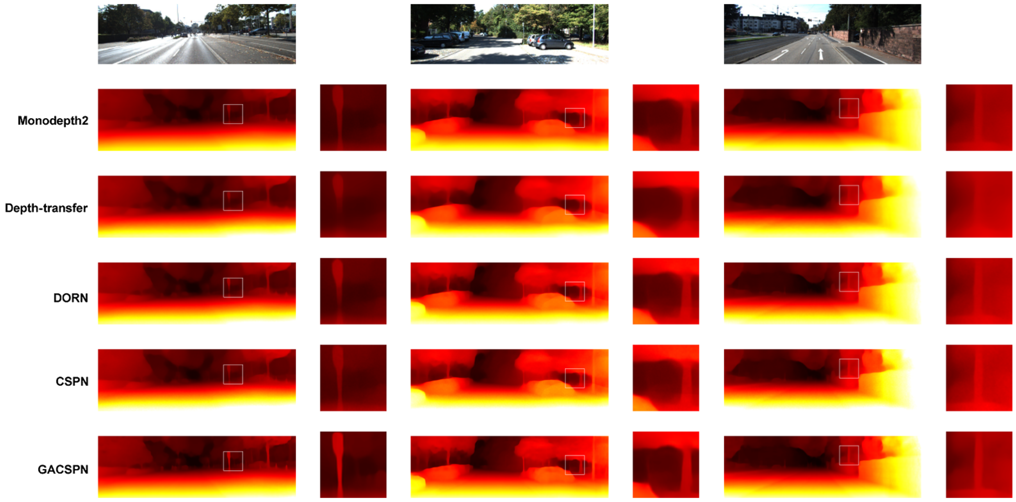


Fig. 9 Qualitative comparison on KITTI 2015 dataset.

Table 3 Comparison results on Make3D dataset.

	C1			C2		
Method	REL	RMS_{log}	RMS	REL	RMS_{log}	RMS
Make3D[14]	—	—	—	0.3700	0.1870	—
Liu et al.[28]	—	—	—	0.3790	0.1480	—
DepthTransfer[3]	0.3550	0.1270	9.2000	0.3610	0.1480	15.1000
Liu et al.[15]	0.3350	0.1370	9.4900	0.3380	0.1340	12.6000
Li et al.[29]	0.2780	0.0920	7.1200	0.2790	0.1020	10.2700
Liu et al.[16]	0.2870	0.1090	7.3600	0.2870	0.1220	14.0900
Roy et al.[30]	—	—	—	0.2600	0.1190	12.4000
Laina et al.[31]	0.1760	0.0720	4.4600	—	—	—
LRC-Deep3D[32]	1.0000	2.5270	19.1100	—	—	—
Monodepth[19]	0.4430	0.1560	11.5130	—	—	—
Kuznietsov et al.[33]	0.4210	0.1900	8.2400	—	—	—
MS-CRF[34]	0.1840	0.0650	4.3800	0.1980	—	8.5600
DORN[4]	**0.1570**	**0.0620**	**3.9700**	**0.1620**	**0.0670**	*7.3200*
CSPN[5]	0.2360	0.0800	6.0480	0.2370	0.1090	12.0110
GA-CSPN	*0.1540*	*0.0580*	*3.7880*	*0.1470*	*0.0580*	**7.8600**

Note: Bold values denote the best score, and italic values denote the second best score.

On KITTI dataset, the results in Table 4 show that our GA-CSPN outperforms the existing state-of-the-art depth supervision methods in RMS, RMS_{log}, and three accuracy metrics with different thresholds. As expected, state-of-the-art accuracy metrics are increased into 95.1% in $\delta < 1.25$, 98.9% in $\delta < 1.25^2$, and 99.6% in $\delta < 1.25^3$.

From Table 5, which shows the comparison results on NYUv2 dataset, GA-CSPN achieves best scores in the REL, RMS_{log}, and three accuracy metrics with different thresholds.

Table 4 Comparison results on KITTI dataset.

Method	Train	REL	SqREL	RMS	$\text{RMS}_{\log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
GeoNet[35]	S	0.1490	1.0600	5.5670	0.2260	0.7960	0.9350	0.9750
DDVO[36]	S	0.1510	1.2570	5.5830	0.2280	0.8100	0.9360	0.9740
DF-Net[37]	S	0.1500	1.1240	5.5070	0.2230	0.8060	0.9330	0.9730
Monodepth[19]	S	0.1330	1.1420	5.5330	0.2300	0.8300	0.9360	0.9700
Monodepth2[7]	S	0.1060	0.8060	4.6300	0.1930	0.8760	0.9580	0.9800
Eigen et al.[2]	D	0.2030	1.5480	6.3070	0.2820	0.7020	0.8900	0.8900
Liu et al.[16]	D	0.2010	1.5840	6.4710	0.2730	0.6800	0.8980	0.9670
AdaDepth[38]	D	0.1670	1.2570	5.5780	0.2370	0.7710	0.9220	0.9710
Kuznietsov et al.[33]	D	0.1130	0.7410	4.6210	0.1890	0.8620	0.9600	0.9860
DVSO[39]	D	0.0970	0.7340	4.4420	0.1870	0.8880	0.9580	0.9800
SVSM FT[40]	D	0.0940	0.6260	4.2520	0.1770	0.8910	0.9650	0.9840
Guo et al.[41]	D	0.0960	0.6410	4.0950	0.1680	0.8920	0.9670	0.9860
DORN[4]	D	**0.0720**	**0.3070**	*2.7270*	*0.1200*	*0.9320*	*0.9840*	*0.9940*
FeatDepth[42]	D	0.1040	0.7290	4.4810	0.1790	0.8930	0.9650	0.9840
Sumanta et al.[43]	D	0.1040	0.7250	4.4040	0.1790	0.8920	0.9660	0.9840
AdvDepth[44]	D	0.1196	0.8890	4.3290	0.1920	0.8650	0.9430	0.9890
CSPN[5]	D	0.1750	1.3620	5.9770	0.2450	0.7600	0.9140	0.9690
GA-CSPN	D	*0.0720*	*0.3250*	**2.7020**	**0.1160**	**0.9510**	**0.9890**	**0.9960**

Note: D, depth supervision; S, self-supervised stereo supervision.
Note: Bold values denote the best score, and italic values denote the second best score.

Meanwhile, the proposed GA-CSPN achieves the second-best scores in RMS. Significant progress has been made by GA-CSPN that increasing the state-of-the-art score into 87.1% in $\delta < 1.25$, 97.9% in $\delta < 1.25^2$, and 99.3% in $\delta < 1.25^3$.

It can be observed from Table 6 that the parameter number of GA-CSPN is much less than other state-of-the-art methods. Compared to the original CSPN, the parameter number of GA-CSPN is only increased by 24.6%.

Considering applications of depth estimation, running time is also a key factor to evaluate algorithms. In Fig. 10, running time versus accuracy of different methods is illustrated.

From Fig. 10, it can be seen that although DORN[4] and BTS[18] achieve similar accuracy with GA-CSPN, they suffer much more computation cost than GA-CSPN. Taking both accuracy and running time capability into the evaluation consideration, the performance of GA-CSPN is superior to these state-of-the-arts.

4.5.3 *Robustness analysis*

To explore the robustness of GA-CSPN, we test some fully supervised deep-learning methods on the NYUv2 dataset with Gaussian noise. Gaussian noise is statistical noise having a probability density function equal to that of the normal distribution, which is also known as the Gaussian distribution. The probability density function p of a Gaussian random variable z is given by

$$p_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}, \tag{15}$$

Table 5 Comparison results on NYUv2 dataset.

Method	REL	$\mathrm{RMS}_{\log}$	RMS	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Make3D[14]	0.3490	—	1.2140	0.4470	0.7450	0.8970
DepthTransfer[3]	0.3500	0.1310	1.2000	—	—	—
Liu et al.[15]	0.3350	0.1270	1.0600	—	—	—
Ladicky et al.[45]	—	—	—	0.5420	0.8290	0.9410
Li et al.[29]	0.2320	0.0940	0.8210	0.6210	0.8860	0.9680
Wang et al.[13]	0.2200	—	0.8240	0.6050	0.8900	0.9700
Roy et al.[30]	0.1870	—	0.7440	—	—	—
Liu et al.[16]	0.2130	0.0870	0.7590	0.6500	0.9060	0.9760
Eigen et al.[2]	0.1580	—	0.6410	0.7690	0.9500	0.9880
Chakrabarti et al.[46]	0.1490	—	0.6200	0.8060	0.9580	0.9870
Laina et al.[31]	0.1270	0.0550	0.5730	0.8110	0.9530	0.9880
Li et al.[47]	0.1430	0.0630	0.6350	0.7880	0.9580	0.9910
MS-CRF[34]	0.1210	0.0520	0.5860	0.8110	0.9540	0.9870
DORN[4]	0.1150	0.0510	0.5090	0.8280	0.9650	0.9900
SharpNet[17]	0.1390	**0.0470**	0.5020	0.8360	0.9660	0.9900
BTS[18]	**0.1100**	**0.0470**	*0.3920*	**0.8450**	**0.9730**	**0.9920**
CSPN[5]	0.1950	0.0640	0.6970	0.7620	0.9250	0.9720
GA-CSPN	*0.1090*	*0.0450*	**0.4030**	*0.8710*	*0.9790*	*0.9930*

Note: Bold values denote the best score, and italic values denote the second best score.

Table 6 Comparison of parameter number and running time on NYUv2 dataset.

Method	#Parameters	Running time (ms)
GA-CSPN	21.3M	171.853
CSPN[5]	17.1M	139.698
BTS[18]	47.0M	860.589
DORN[4]	51.1M	276.255

where z represents the pixel value, μ represents the mean value, and σ represents its standard deviation.

For pre-processing, the original images are added with random Gaussian noise and images under different noise variances are shown in Fig. 11. Var represents the variance of Gaussian noise in Fig. 11. It can be observed from Fig. 11, the slight noise, whose variance $\le 10^{-3}$, is unperceivable for human vision.

To validate the robustness of exist deep-learning methods, we evaluate the performance of CSPN, GA-CSPN, SharpNet, and DORN on Gaussian noise additive dataset.

As shown in Fig. 12, the NYUv2 dataset with gradual increase of Gaussian noise is evaluated by three accuracy metrics with different thresholds. Though the slight noise is unperceivable for

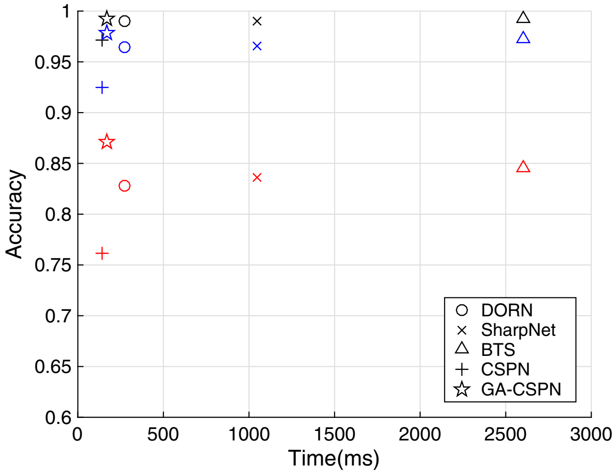


Fig. 10 A comparison between accuracy and time on NYUv2 dataset. Red marks the accuracy of $\delta < 1.25$. Blue marks the accuracy of $\delta < 1.25^2$. Black marks the accuracy of $\delta < 1.25^3$.

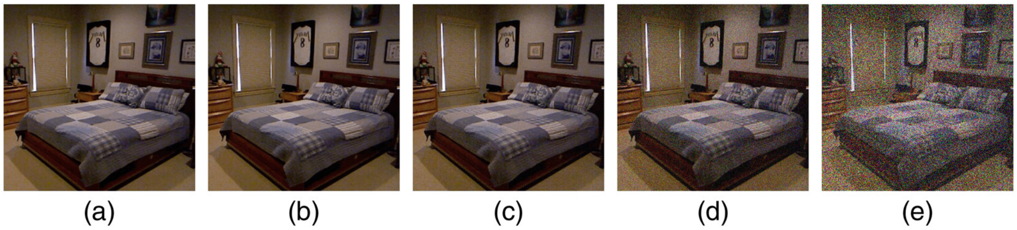

(a) (b) (c) (d) (e)

Fig. 11 Gaussian noise additive examples on NYUv2 dataset: (a) source, (b) $\mathrm{Var} = 10^{-4}$, (c) $\mathrm{Var} = 10^{-3}$, (d) $\mathrm{Var} = 10^{-2}$, and (e) $\mathrm{Var} = 10^{-1}$.

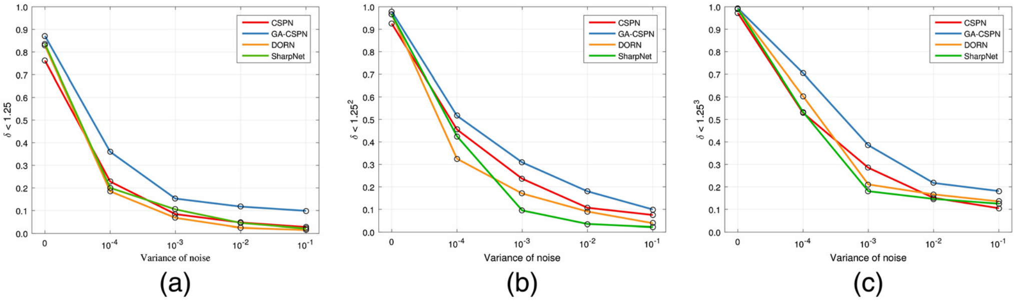


Fig. 12 Gaussian noise additive model results on NYUv2 dataset: (a), (b), and (c).

human vision, it greatly reduces the accuracy of depth estimation in most deep-learning methods. Compared with CSPN and DORN under Gaussian noise circumstance, the accuracy of GA-CSPN decreases much less.

From above experiments, our method is an accurate end-to-end depth estimation approach that is more robust than most traditional fully supervised methods. Although the accuracy of the GA-CSPN decreases in three noise datasets, it has achieved significant progress compared to previous methods.

5 Conclusion

We presented a generative adversarial CSPN with a correlation loss, achieving state-of-the-art depth prediction. We introduced three contributions. (1) A correlation loss to enhance the discrimination performance and thus to distinguish the estimated depths precisely. (2) Dense UNet, the backbone of the generator network, building complete information transmission connections between each convolution layer and improving depth estimation accuracy. (3) An efficient coarse-to-fine depth propagation named high-order CSPN. Experiments on Make3D,[14] KITTI 2015,[25] and NYUv2[26] datasets indicated the superiority of the proposed methods over state-of-the-arts.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61573168).

References

1. D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," arXiv:1406.2283 (2014).
2. D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vision*, pp. 2650–2658 (2015).
3. K. Karsch, C. Liu, and S. B. Kang, "Depth transfer: depth extraction from video using non-parametric sampling," *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(11), 2144–2158 (2014).
4. H. Fu et al., "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2002–2011 (2018).
5. X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," *Lect. Notes Comput. Sci.* **11220**, 103–119 (2018).
6. S. Liu et al., "Learning affinity via spatial propagation networks," arXiv:1710.01020 (2017).
7. C. Godard et al., "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vision*, pp. 3828–3838 (2019).
8. Y. Almalioglu et al., "Ganvo: unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks," in *Int. Conf. Rob. and Autom.*, IEEE, pp. 5474–5480 (2019).
9. I. J. Goodfellow et al., "Generative adversarial networks," arXiv:1406.2661 (2014).
10. O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," *Lect. Notes Comput. Sci.* **9351**, 234–241 (2015).
11. J. Flynn et al., "Deep stereo: learning to predict new views from the world's imagery," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 5515–5524 (2016).
12. D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.* **47**(1), 7–42 (2002).
13. P. Wang et al., "Towards unified depth and semantic prediction from a single image," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2800–2809 (2015).
14. A. Saxena, M. Sun, and A. Y. Ng, "Make3D: learning 3D scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(5), 824–840 (2009).
15. M. Liu, M. Salzmann, and X. He, "Discrete-continuous depth estimation from a single image," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 716–723 (2014).
16. F. Liu et al., "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(10), 2024–2039 (2016).
17. M. Ramamonjisoa and V. Lepetit, "Sharpnet: fast and accurate recovery of occluding contours in monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vision Workshops* (2019).
18. J. H. Lee et al., "From big to small: Multi-scale local planar guidance for monocular depth estimation," arXiv:1907.10326 (2019).
19. C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 270–279 (2017).

20. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv:1409.1556 (2014).
21. D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” arXiv:1412.6980 (2014).
22. J. Deng et al., “Imagenet: a large-scale hierarchical image database,” in *IEEE Conf. Comput. Vision and Pattern Recognit.*, IEEE, pp. 248–255 (2009).
23. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” Technical Report, California University of San Diego La Jolla Inst. for Cognitive Science (1985).
24. A. Saxena et al., “Learning depth from single monocular images,” in *Proc. 18th Int. Conf. Neural Inf. Process. Syst.*, pp. 1–8 (2005).
25. A. Geiger et al., “Vision meets robotics: the KITTI dataset,” *Int. J. Rob. Res.* **32**(11), 1231–1237 (2013).
26. N. Silberman et al., “Indoor segmentation and support inference from RGBD images,” *Lect. Notes Comput. Sci.* **7576**, 746–760 (2012).
27. L. Van der Maaten and G. Hinton, “Visualizing data using T-SNE,” *J. Mach. Learn. Res.* **9**(11), 2579–2605 (2008).
28. B. Liu, S. Gould, and D. Koller, “Single image depth estimation from predicted semantic labels,” in *IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, IEEE, pp. 1253–1260 (2010).
29. B. Li et al., “Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1119–1127 (2015).
30. A. Roy and S. Todorovic, “Monocular depth estimation using neural regression forest,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 5506–5514 (2016).
31. I. Laina et al., “Deeper depth prediction with fully convolutional residual networks,” in *Fourth Int. Conf. 3D Vision*, IEEE, pp. 239–248 (2016).
32. J. Xie, R. Girshick, and A. Farhadi, “Deep3D: fully automatic 2D-to-3D video conversion with deep convolutional neural networks,” *Lect. Notes Comput. Sci.* **9908**, 842–857 (2016).
33. Y. Kuznietsov, J. Stuckler, and B. Leibe, “Semi-supervised deep learning for monocular depth map prediction,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 6647–6655 (2017).
34. D. Xu et al., “Multi-scale continuous CRFs as sequential deep networks for monocular depth estimation,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 5354–5362 (2017).
35. Z. Yin and J. Shi, “Geonet: unsupervised learning of dense depth, optical flow and camera pose,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1983–1992 (2018).
36. C. Wang et al., “Learning depth from monocular videos using direct methods,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2022–2030 (2018).
37. Y. Zou, Z. Luo, and J.-B. Huang, “Df-net: unsupervised joint learning of depth and flow using cross-task consistency,” *Lect. Notes Comput. Sci.* **11209**, 36–53 (2018).
38. J. N. Kundu et al., “Adadepth: unsupervised content congruent adaptation for depth estimation,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2656–2665 (2018).
39. N. Yang et al., “Deep virtual stereo odometry: leveraging deep depth prediction for monocular direct sparse odometry,” *Lect. Notes Comput. Sci.* **11212**, 817–833 (2018).
40. Y. Luo et al., “Single view stereo matching,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 155–163 (2018).
41. X. Guo et al., “Learning monocular depth by distilling cross-domain stereo networks,” *Lect. Notes Comput. Sci.* **12364**, 484–500 (2018).
42. C. Shu et al., “Feature-metric loss for self-supervised learning of depth and egomotion,” *Lect. Notes Comput. Sci.* **12364**, 572–588 (2020).
43. S. Bhattacharyya et al., “Efficient unsupervised monocular depth estimation using attention guided generative adversarial network,” *J. Real-Time Image Process.* 1–12 (2021).
44. K. Li et al., “Adv-depth: self-supervised monocular depth estimation with an adversarial loss,” *IEEE Signal Process Lett.* **28**, 638–642 (2021).

45. L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 89–96 (2014).
46. A. Chakrabarti, J. Shao, and G. Shakhnarovich, "Depth from a single image by harmonizing overcomplete local network predictions," arXiv:1605.07081 (2016).
47. J. Li, R. Klein, and A. Yao, "A two-streamed network for estimating fine-scaled depth maps from single RGB images," in *Proc. IEEE Int. Conf. Comput. Vision*, pp. 3372–3380 (2017).

Zhengyang Lu received her MSc degree in electrical and electronic engineering from the University of Surrey, Guilford, United Kingdom, in 2018. Currently, he is a PhD candidate with Jiangnan University. His current research interests include depth estimation, photometric stereo, and SLAM.

Ying Chen received her PhD in control science and engineering from Xi'an Jiaotong University in 2005. She is a professor in the Department of Information Technology at Jiangnan University. Her research activities are focused on computer vision, pattern recognition, and information fusion.