


RESEARCH

Open Access



# An image-guided network for depth edge enhancement

Kuan-Ting Lee, En-Rwei Liu, Jar-Ferr Yang\*  and Li Hong

\*Correspondence:  
jefyang@mail.ncku.edu.tw  
Department of Electrical  
Engineering, Institute  
of Computer  
and Communication  
Engineering, National Cheng  
Kung University, Tainan City,  
Taiwan

## Abstract

With the rapid development of 3D coding and display technologies, numerous applications are emerging to target human immersive entertainments. To achieve a prime 3D visual experience, high accuracy depth maps play a crucial role. However, depth maps retrieved from most devices still suffer inaccuracies at object boundaries. Therefore, a depth enhancement system is usually needed to correct the error. Recent developments by applying deep learning to deep enhancement have shown their promising improvement. In this paper, we propose a deep depth enhancement network system that effectively corrects the inaccurate depth using color images as a guide. The proposed network contains both depth and image branches, where we combine a new set of features from the image branch with those from the depth branch. Experimental results show that the proposed system achieves a better depth correction performance than state of the art advanced networks. The ablation study reveals that the proposed loss functions in use of image information can enhance depth map accuracy effectively.

**Keywords:** Depth map, Deep convolutional neural network, Image-guided depth enhancement

## 1 Introduction

With the rapid development of 3D media and display technologies, 3D multimedia immersive services are being explored to address new demands. For example, 3D virtual reality has been widely used in various fields, such as medicine, games, and education. In recent years, an increasing number of 3D movies are made commercially available in cinemas. With display technology improvement, there is high expectation that the glassed 3D displays to watch stereo 3D videos will be ultimately replaced by the naked-eye 3D displays, without wearing any glasses.

Supporting naked-eye 3D displays brings the need to have multiple view images available. In depth image-based rendering (DIBR) [1], 3D information is represented by the color image frame and its corresponding depth map. Pixel-based perspective multiple view images are then generated effectively based on information from the color image frame and depth map. The multi-view can help retrieve the image successfully [2]. To achieve a high-quality and comfortable 3D viewing experience, it is crucial to have an accurate depth map.

Depth maps are commonly acquired either by depth sensors [3] or by stereo matching pairs [4, 5]. However, the depth map generated by depth sensors is often noisy, missing depth values and tends to misalign with the object boundaries in color image. The depth map estimated by stereo matching methods often contains errors in occlusion and flat regions. Recently, much work has been done to predict depth maps based on 2D videos, such as using monocular depth estimation networks [6–8] or creating key depth frames manually followed with depth map interpolation [9] afterward. The generated depth map suffers either blurred edges as shown in Fig. 1 or other inevitable errors caused by manual notations along object boundaries. To achieve high-quality accurate depth maps, a depth enhancement system is needed to remove the noise and correct the depth errors.

Without loss of generality, a typical scenario that a limited quality depth map containing noise and errors is retrieved from a pair of stereo color images is considered. Multiple approaches have been proposed to further enhance the depth map quality by exploring the pixel correlation and structure relationship between the color image and depth map. However, when the color image contains highly textured objects, the enhancement from correlation calculation often causes texture-like artifacts in depth map. Traditional depth enhancement approaches apply adaptive filters to smooth or remedy the noisy depth maps. Gaussian filtering-based methods [10] estimate the missing depth values using the known neighboring depth values. Joint bilateral filtering (JBF)-based methods [11, 12] as an extension of bilateral filtering [13], apply information from color frame to gain more accurate depth map enhancement. However, these frameworks usually generate blurry depth results since they cannot train the systems to focus on the area around object boundaries.

Recently, deep learning has been introduced in various image processing applications, such as image super-resolution [14, 15], image restoration [16], image denoising [17, 18], and depth denoising by graph [19] networks. The denoising and enhancement convolutional neural network (DE-CNN) [20] and image-guided method [21] also have been adopted for depth enhancement.

In this paper, we propose an end-to-end framework for depth enhancement with the inputs of color frame and noisy depth map and the output of the enhanced depth map. The rest of the paper is organized as follows. In Sect. 2, we briefly review related work, including deep residual convolution neural network (DRECNN) [22], residual dense network [23], and focal loss [24]. In Sect. 3, we describe in details the proposed

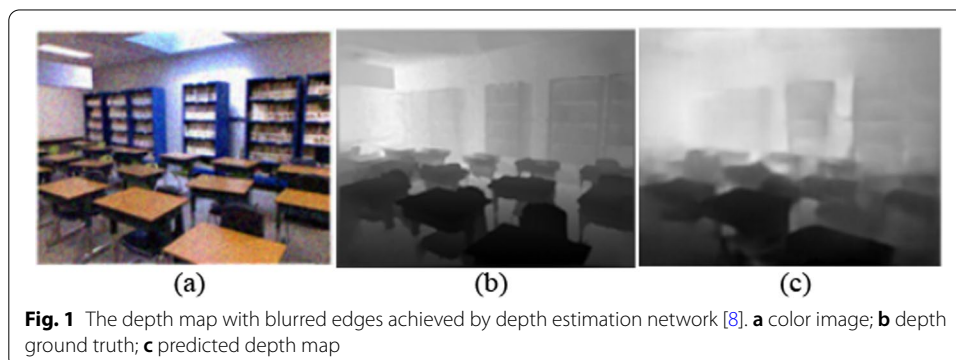


image-guided depth enhancement (IGDE) system. In Sect. 4, we present the experimental results. Finally, we draw the conclusions in Sect. 5.

### 2 Related work

The DRECNN is a typical framework that performs depth enhancement well. It learns the underlying correlation between depth map and color image first, and then applies the learned correlation to enhance the quality of the depth map. As shown in Fig. 2, the DRECNN architecture is divided into depth branch, intensity branch, and fusion module. The depth and intensity branches have the same structure, consisting of one set of convolution and ReLU layers and seven sets of convolution, batch normalization, and ReLU layers to retrieve the depth and intensity feature maps. The fusion module applies eleven sets of convolution, batch normalization, and ReLU layers and a convolution layer to retrieve fusing coefficient maps. By referring to the concept of image-guided filter [21], the filter output  $\tilde{I}_i$  is given as

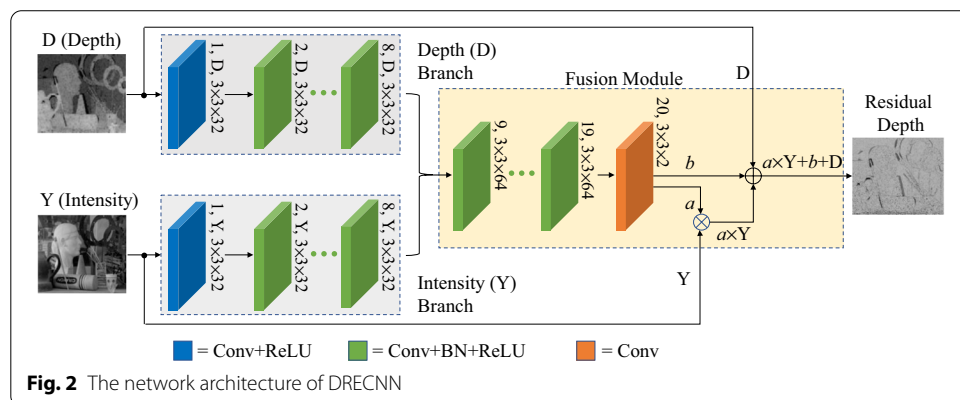
$$\tilde{I}_i = a_k I_i + b_k, \forall i \in \omega_k, \tag{1}$$

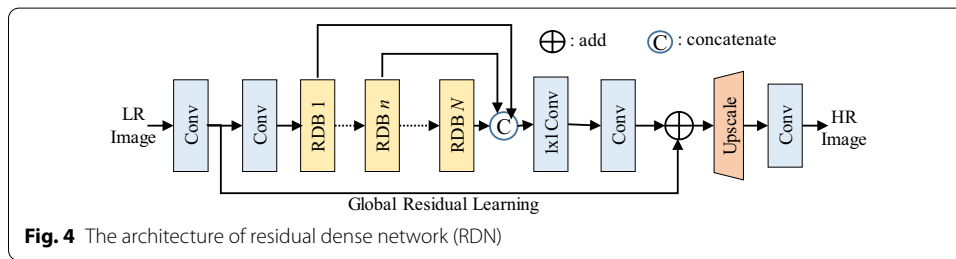
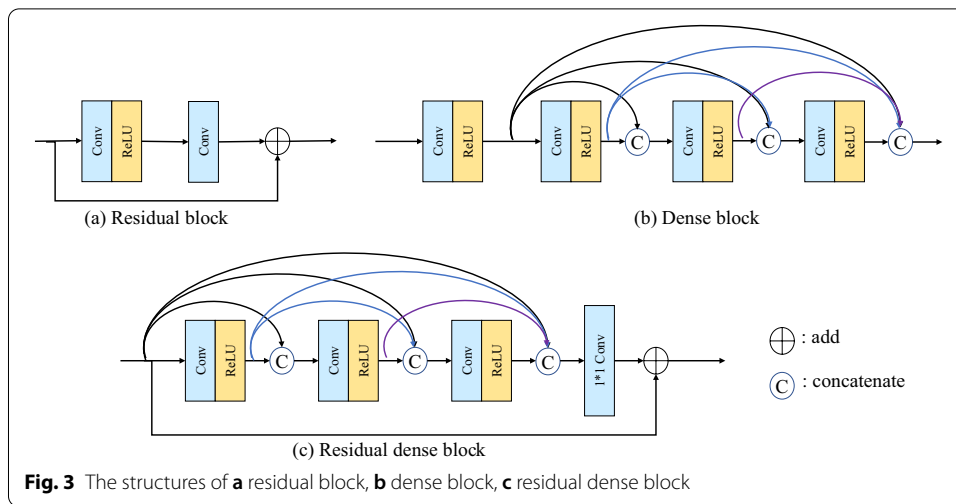
where  $I$  is the input guidance image,  $a_k$  and  $b_k$  are the linear coefficients assumed to be constant in the window  $\omega_k$ . Extending this concept, the DRECNN with the fusion module retrieves the pixel-level fusing coefficient maps  $a$  and  $b$ . As shown in Fig. 2, the residual depth map can be obtained by

$$\Delta\tilde{D} = a \times Y + b + D, \tag{2}$$

where  $Y$  is the luminance of color image and  $D$  is the depth map. The DRECNN effectively improves depth enhancement and avoids overfitting problem by finding a linear model supervised by the ground-truth label.

After AlexNet [25] was proposed, the state-of-the-art CNN architectures commonly adopt large number of layers. However, by simply increasing convolutional layers, better results are not guaranteed due to the gradient vanishing problem. Using batch normalization could solve partially the problem of gradient vanishing. ResNet [23], which utilizes the residual blocks by adding the original input to the output with a shortcut connection, effectively solves the degradation problem caused by increasing the network layers. Since then, the residual blocks are modified to build various high performance





networks. The EDSR [26] removes the batch normalization to boost the convergence speed. DenseNet [27] achieves similar results with much smaller number of parameters. SRDenseNet [28] applies DenseNet to solve image super-resolution effectively. Figure 3 shows the structures of the residual block, dense block, and residual dense block.

For image super-resolution, the architecture of residual dense network (RDN) [15] as shown in Fig. 4 is composed of multiple residual dense blocks. The features generated by previous convolution layers are concatenated to  $1 \times 1$  convolution layer to reduce the number of channels. The RDN consists of  $N$  residual dense blocks (RDBs) which transfer the low-resolution (LR) image to the high-resolution (HR) image. The RDN preserves the details of the LR image and performs the suitable image corrections to obtain the HR image.

In deep learning CNNs, it is important to design loss functions in order to train the target CNN network. The loss functions with mean square error (MSE) and mean absolute error (MAE) are often used in regression problems, while the cross-entropy is used in classification problems. To improve speed and direction of network convergence, special loss functions has been proposed. Taking the binary classification problem, the cross-entropy loss is given as

$$CE(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1 - p) & \text{otherwise} \end{cases} \quad (3)$$

where  $y$  with  $\{+1, -1\}$  denotes the label and  $p$  represents the probability that the predicted sample belongs to 1. Adding the cross-entropy of all samples, we can find the loss of the network. To correct the imbalance of binary samples, the focus loss used in RetinaNet [24] is suggested as

$$FL(p_t) = -\alpha(1 - p)^\gamma \log(p), \tag{4}$$

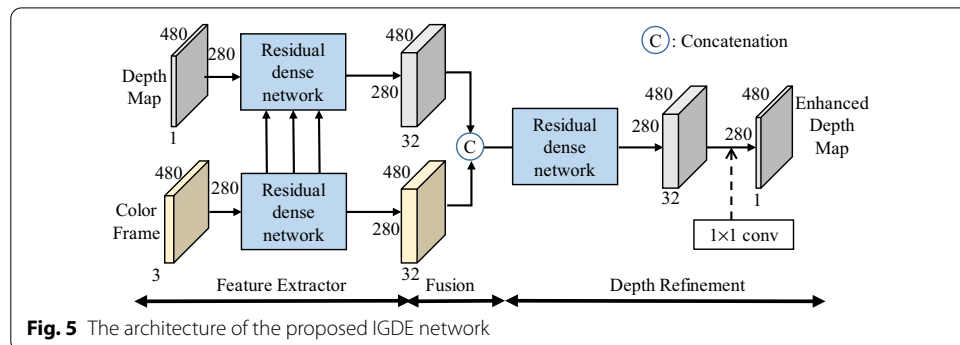
where  $\gamma > 0$  is a focusing parameter to reduce the relative loss for well-classified examples with  $p > 0.5$  and  $\alpha$  is the shared weight to control positive and negative samples. Comparing to two-stage object detectors, faster RCNN [29] and RFCN [30], the focal loss can improve one-stage object detectors, YOLO [31] and SSD [32] to obtain higher performance. The one-stage detector has too much difference in the number of positive and negative samples during training,  $\alpha$  is used to reduce the influence of negative samples with  $(1 - p_t)^\gamma$  modulating factor. The modulating factor reduces the weight of easy-to-classify samples to ensure the network pay more attention to difficult-to-classify samples. The effectiveness of focal loss has been proven in many advanced networks.

### 3 The methods

The guided image filter [21] uses the correlation between color and depth maps to enhance the noisy depth map. However, images with complex textures often degrade the depth map with ghosting textures. Learning-based methods mitigate the strong influence from image texture. However, the enhanced depth maps often contain inaccurate depth at object boundaries. To address this issue, we proposed an end-to-end depth map enhancement system that focuses mainly on correction of the depth edges.

#### 3.1 The IGDE network

The proposed image-guided depth enhancement (IGDE) network, as shown in Fig. 5, consists of two feature extractors, one fusion module, and one depth refinement module. It is noted task-adaptive attention [33] and multi-feature fusing [34] can help increase the performances of image captioning and recognition, respectively. We employ the residual dense network (RDN) as the backbone of the feature extractor and depth refinement module. We extract features from the image and depth frames, and concatenate them together as the fused feature. The fused feature is sent to the depth refinement module to obtain the enhanced depth map.



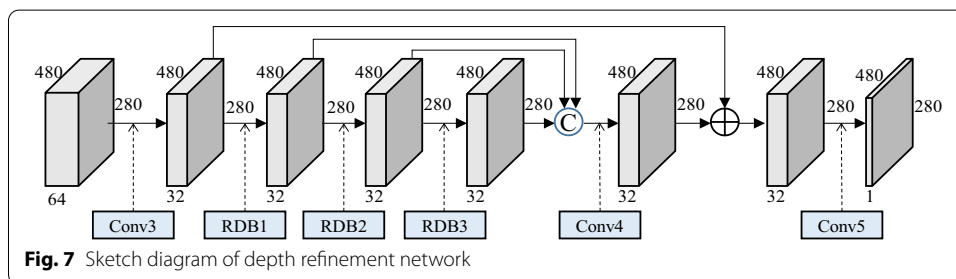
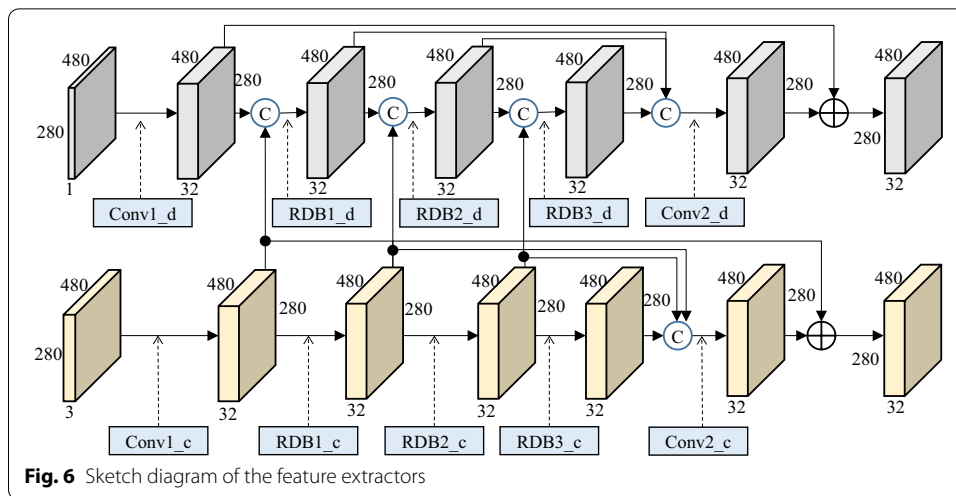


Figure 6 shows the detailed structure of the feature extractor. In the early stages of the network, the low-level features of the image frame are concatenated into the low-level features of the depth map. In simulation section, we will describe a better number of layers for the concatenations of low-level image features.

At the end of the network, we convert the fused feature into the enhanced depth map with the depth refinement module. Here, we use the same residual dense network as the backbone of the depth refinement module. The features obtained by the residual dense network are restored to a depth map by a  $1 \times 1$  convolution. The detailed architecture of the depth refinement module is shown in Fig. 7. All the convolutions used in the module utilize ReLU to prevent the network output from being too linear.

### 3.2 Loss functions

To train the proposed IGDE network, we refine the noisy depth values at object boundaries to be consistent with the image frame. Typically, depth loss function calculates the depth loss from all pixels in depth map. Since the number of object boundaries pixels is much lower than that of the pixels of the whole image, the impact of errors at object boundaries is often compromised. We proposed to add a special depth focal loss by assigning lower weights for pixels with smaller depth deviations and higher weights for pixels with larger deviation. In addition, we also design a Sobel loss to emphasize the depth deviation at object boundaries. The total loss function, including depth loss  $L_{depth}$ , depth focal loss  $L_{focal}$ , and Sobel loss  $L_{sobel}$ , becomes

$$L(d^*, d, M_{\text{sobel}}) = \rho L_{\text{depth}}(d^*, d) + \mu L_{\text{focal}}(d^*, d) + \lambda L_{\text{sobel}}(d^*, d, M_{\text{sobel}}), \quad (5)$$

where  $d$  and  $d^*$  are predicted depth value and the corresponding ground truth, respectively.  $M_{\text{sobel}}$  is the mask that focuses on the object boundaries, where  $\rho$ ,  $\mu$ , and  $\lambda$  are the weighting factors of the losses. We set all of them to 1. The details of each loss function are described as follows.

### 3.3 Depth loss

To minimize the difference between predicted depth maps  $d$  and the corresponding ground truth  $d^*$ , we use the L1 loss as

$$L_{\text{depth}}(d^*, d) = \frac{1}{HW} \sum_{i,j}^{H,W} |d_{i,j}^* - d_{i,j}|, \quad (6)$$

where  $i$  and  $j$  denote the pixel indices, and  $H$  and  $W$  are the height and width of depth maps, respectively.

### 3.4 Depth focal loss

For refining the noisy depth map, the error depth pixels only occupy a small portion of the depth map. We aim to train the network to focus more on the error pixels than the correct ones. Hence, we suggest the depth focal loss as

$$L_{\text{focal}}(d^*, d) = \frac{1}{HW} \sum_{i,j}^{H,W} -\alpha (1 - e_{i,j})^\gamma \log(e_{i,j}) \quad (7)$$

with

$$e_{i,j} = 1 - \frac{|d^* - d|}{255}, \quad (8)$$

where  $\alpha$  and  $\gamma$  are the shared weight and focusing parameter. Currently, we set the values of  $\alpha$  to 0.25 and  $\gamma$  to 2. Similar to the focal loss [24],  $e_{i,j}$  can be treated as the probability that the correct predicted depth is 1 in most cases. In (8), the ratio of the difference between the predicted and ground truth values to the maximum depth value 255 exhibits similar characteristics of positive and negative samples for depth focal loss. With the depth focal loss, our network emphasizes more on the pixels with a higher ratio of errors in order to make the training results more accurate.

### 3.5 Sobel loss

Since the depth map contains the errors mostly near object boundaries, we design a Sobel loss to ensure the network to focus more on areas close to object boundaries. The Sobel loss is expressed as

$$L_{\text{sobel}}(d^*, d, M_{\text{sobel}}) = \frac{1}{HW} \sum_{i,j} \beta \cdot |d_{i,j}^* - d_{i,j}| \times M_{\text{sobel}}(i, j) + (1 - \beta) \cdot |d_{i,j}^* - d_{i,j}| \times (1 - M_{\text{sobel}}(i, j)), \tag{9}$$

where  $M_{\text{sobel}}$  is a depth edge mask and  $\beta$  is the parameter used to control the importance of the edge area. We set  $M_{\text{sobel}}$  to 1 for pixels close to object boundaries and set it to 0 for the rest of pixels. Here, we choose  $\beta = 0.9$ .

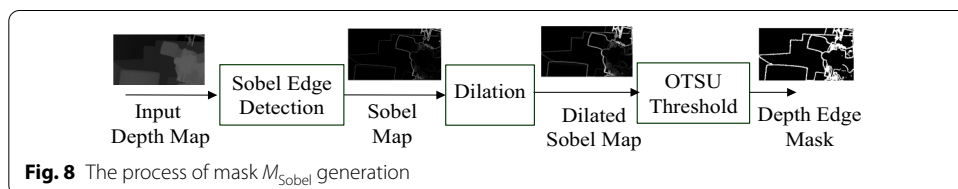
To compute the depth edge mask, as shown in Fig. 8, we first perform Sobel edge detection to the input depth map to obtain Sobel edges. Then, the detected edges are expanded by dilation operator. Finally, we apply the OTSU thresholding method to compute the depth edge mask. In (9), the depth error pixels in the region of depth edge mask will be weighted by 0.9, while those outside the depth edge mask will be weighted by 0.1. Of course, we can use  $\beta$  to adjust the weights of the area near the depth edge with respect to the rest of the area.

#### 4 Results and discussions

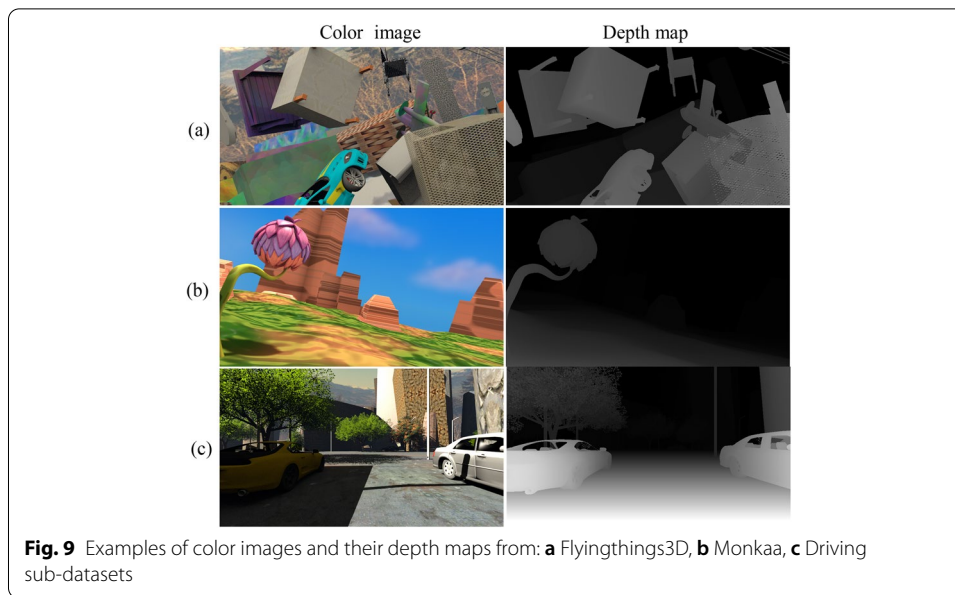
The proposed IGDE system is implemented in Python 3.7, CUDA 10.2, cuDNN 7.6.5, and Tensorflow-GPU 1.15.0 learning function library. For hardware infrastructure, we use the personal computer with Intel Core i7-9700 k CPU 3.6 GHz-4.9 GHz, 32 GB 3200 MHz RAM. NVIDIA Geforce RTX 2080Ti 11G GPU is used to accelerate the training process of the proposed IGDE system.

We evaluate the proposed IGDE system on the Scene Flow dataset [26], which is a large-scale synthetic dataset containing Flyingthings3D, Monkaa, and Driving sub-datasets. Some selected examples of datasets are shown in Fig. 9. Compared to other datasets, it has more accurate ground-truth depth maps since they are generated by virtual images. The images in the dataset are divided into 70,908 training images and 8740 testing images with  $H = 540$  and  $W = 960$ . We crop the image to  $H = 512$  and  $W = 960$  in the proposed network.

To simulate depth maps with erroneous edges, we randomly inflate or reduce depth values at object boundaries in all depth maps. We then take the simulated depth maps and the corresponding color maps as input to the proposed system. During training, images with a batch size of 2 were randomly cropped to size  $H = 280$  and  $W = 480$ . To improve the prediction accuracy, we normalized the input images by dividing them by 255. We trained our network with a learning rate of 0.0001 for 50 epochs.







#### 4.1 Visualization performance of the network

Figure 10 shows the visual results of testing on the Flyingthings3D sub-dataset. The depth map refined by the proposed IGDE system performs well at object boundaries. Comparing the error map before and after refinement, the number of error points has been significantly reduced.

To test the effectiveness of the proposed network, the trained network is directly applied to Middlebury dataset. Figure 11a and b, respectively, shows four original natural images and their corresponding ground-truth depth maps with unknown holes, which are treated as noisy depth maps. After simple extension of known depth values from the bottom vertically and the enhanced process by the proposed IGDE system, Figure 11c and d show the enhanced depth maps and the error depth maps, respectively. For those unknown holes, for natural images, we do not know the exact depth value. The subjective quality as the graphic image becomes impossible. However, the refined depth maps by the proposed IGDE system show the reasonably good objective quality. The IGDE system can enhance the depth maps of natural images successfully. For detailed evaluation of the performances, we present numerical comparisons with other methods in the next sub-section.

#### 4.2 Comparisons with quality measures

We compare the performance of the proposed IGDE system with three depth refinement networks, namely, denoising and enhancement CNN (DE – CNN) [20], deep residual enhancement CNN (DRECNN) [22], and depth enhancement network with color-based prediction network (DEN+CBPN) [35]. The DE – CNN with single branch concatenates the depth map and color image as the input, while the proposed IGDE, DRECNN, and DEN+CBPN systems with two branches fuse the depth and image features in different approaches. Without ground truth values, the quality measure can also use no-reference measure [36]. With the ground-truth depth maps,

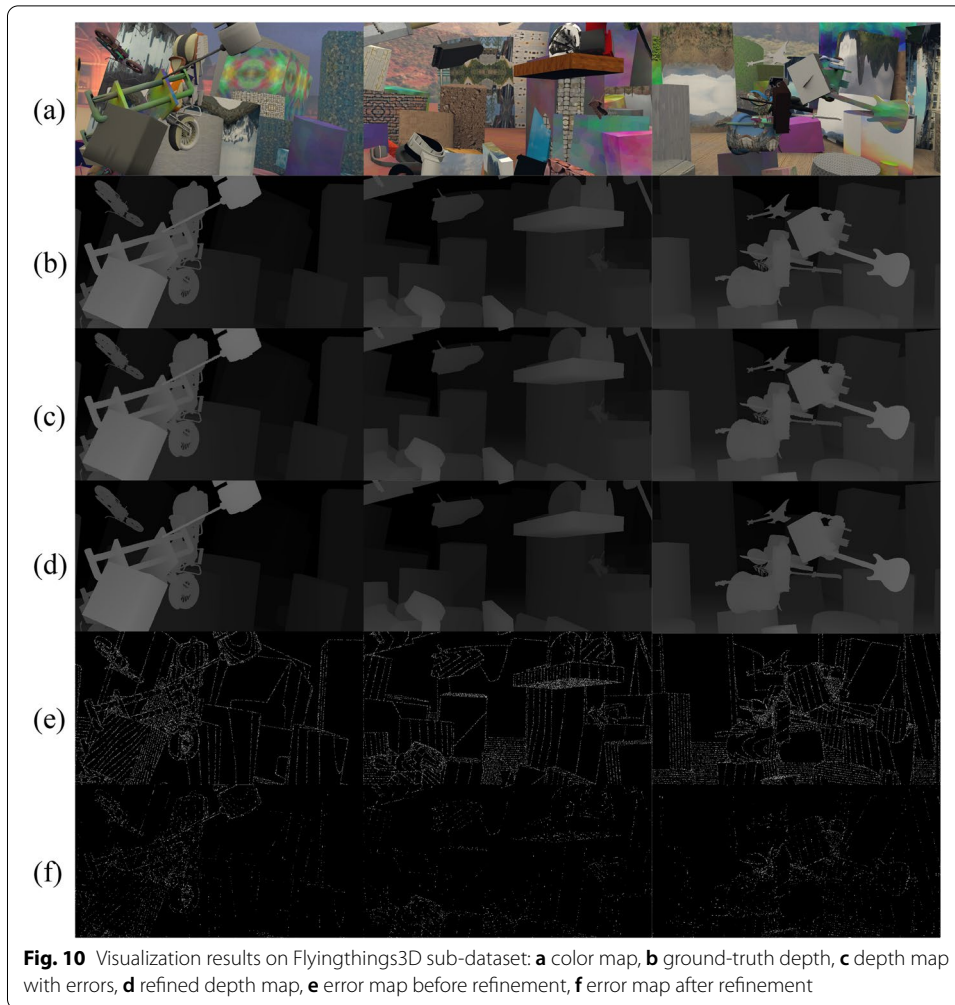


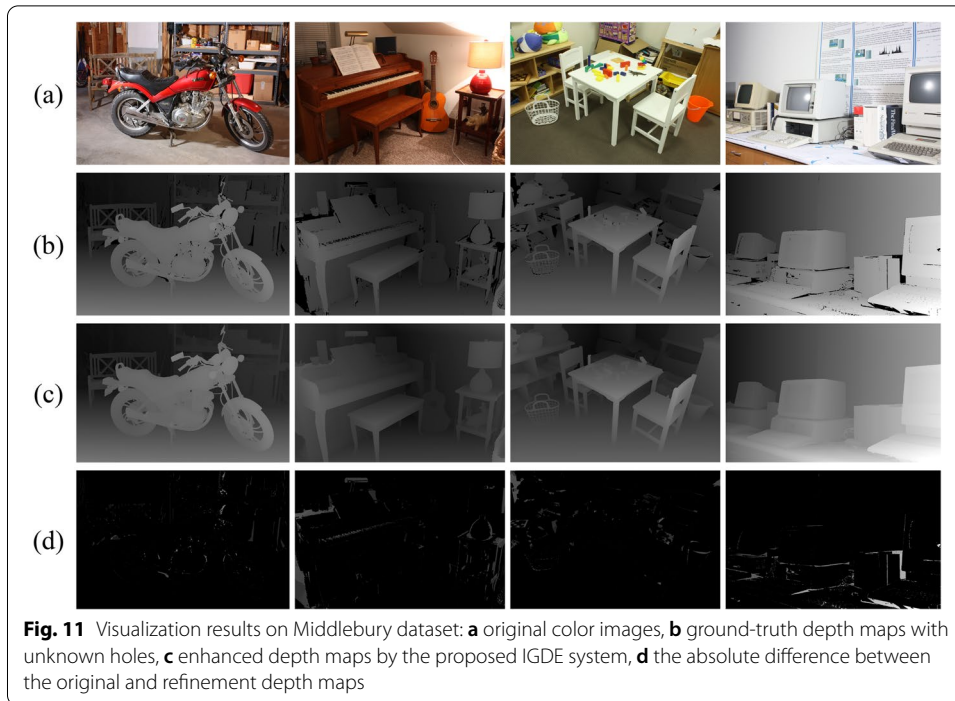
Table 1 shows the comparison results on Scene Flow testing set [37]. We use common quality measures, such as PSNR in dB, SSIM, and RMSE, to evaluate the performance of all networks. In addition, we also, respectively, calculate the  $PSNR_t$  and  $PSNR_f$  of correct and error depth pixels of the prediction results. Table 1 shows that the proposed IGDE achieves the best results, which are marked with bold face. Hereafter, in Tables 2 and 3, we also marked the best results with bold face.

The PSNR,  $PSNR_e$ , and  $PSNR_f$  in dBs are defined as follows:

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX^2}{MSE_{all}} \right) = 10 \cdot \log_{10} \left( \frac{255^2}{MSE_{all}} \right), \tag{10}$$

$$PSNR_t = 10 \cdot \log_{10} \left( \frac{255^2}{MSE_t} \right), \tag{11}$$

$$PSNR_f = 10 \cdot \log_{10} \left( \frac{255^2}{MSE_f} \right), \tag{12}$$



**Table 1** Performance comparisons in scene flow testing set

Measures Model	PSNR (dB)	PSNR <sub>t</sub> (dB)	PSNR <sub>r</sub> (dB)	SSIM	RMSE
Input depth	36.6835	None	21.1025	0.99999589	0.0146703
DE-CNN [20]	42.1632	45.1183	29.1516	0.99999824	0.0078065
DRECNN [22]	45.9433	49.6217	32.3469	0.99999935	0.0050490
DEN + CBPN [35]	37.1230	None	21.1025	0.99999581	0.0139462
Proposed	<b>48.1056</b>	<b>52.9595</b>	<b>33.7814</b>	<b>0.99999962</b>	<b>0.0039363</b>

where  $MSE_{all}$ ,  $MSE_p$ , and  $MSE_f$ , respectively, denote the mean square error of all, correct, and incorrect depth pixels. The structural similarity (SSIM) measure is defined as

$$SSIM(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y), \tag{13}$$

where  $l(x, y)$ ,  $c(x, y)$ , and  $s(x, y)$  denote the luminance, contrast, and structure measures of  $x$  and  $y$ , which are, respectively, defined as

$$l(x, y) = \frac{2u_x u_y + C_1}{u_x^2 + u_y^2 + C_1}, c(x, y) = \frac{2\sigma_x \sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3}, \tag{14}$$

where  $u_x$  and  $u_y$  are the averages of  $x$  and  $y$ ;  $\sigma_x$  and  $\sigma_y$  represent the standard deviations of  $x$  and  $y$ , respectively;  $\sigma_{xy}$  denotes the covariance of  $x$  and  $y$ ; and  $C_1$ ,  $C_2$ , and  $C_3$  are constants to stabilize the division with a weak denominator. The RMSE is defined as

**Table 2** Evaluation results of the proposed loss functions

Measures Model	PSNR (dB)	PSNR <sub>t</sub> (dB)	PSNR <sub>f</sub> (dB)	SSIM	RMSE
w/o depth focal loss w/o Sobel loss	47.5933	52.8712	33.0588	0.999999578	0.004178
w/ depth focal loss w/o Sobel loss	47.8342	53.5258	33.1803	0.999999602	0.004060
w/o depth focal loss w/ Sobel loss	47.5974	52.5389	33.2153	0.999999596	0.004178
w/ depth focal loss w/ Sobel loss	<b>48.1057</b>	<b>52.9595</b>	<b>33.7814</b>	<b>0.999999625</b>	<b>0.003936</b>

**Table 3** Evaluation results of reducing layers of color map features

Measures Model	PSNR (dB)	PSNR <sub>t</sub> (dB)	PSNR <sub>f</sub> (dB)	SSIM	RMSE
w/o color map information	47.7598	<b>53.6426</b>	33.0325	0.999999603	0.0040952
w/ one layer of color map information	47.4954	52.3334	33.1743	0.999999583	0.0042278
w/ two layers of color map information	47.154	51.9130	32.8571	0.999999537	0.0043897
w/ three layers of color map information	<b>48.1056</b>	52.9595	<b>33.7814</b>	<b>0.999999625</b>	<b>0.0039363</b>

$$\text{RMSE} = \left( \frac{1}{N} \sum_{i=1}^N (d_i^* - d_i)^2 \right)^{\frac{1}{2}}, \quad (15)$$

where  $N$  denotes the total number of pixels for prediction result,  $d_i^*$  and  $d_i$  indicate the  $i$ th ground-truth depth value map and the  $i$ th predicted depth value, respectively. We implemented networks of the three selected depth refinement methods due to their source codes are not available. We used our training configuration to train their networks. Based on comparison results, the proposed IGDE system achieves the best performance.

### 4.3 Ablation study

We evaluated the performance of the proposed IGDE system with different settings. First, we train the IGDE network with different sets of loss functions to prove that depth focal loss and Sobel loss make the network predict better. The prediction results with or without adding the proposed loss functions are shown in Table 2. The comparison results show that the two proposed loss functions clearly help achieve better results.

To demonstrate the effectiveness of concatenating three layers of low-level features of the image branch to those of the depth branch, we also try to reduce the number of concatenating layers of low-level features. Since the deeper layers of color image frame are more important to the depth map, we try to reduce the shallow layers of color map information to the depth branch. The comparison results are shown in Table 3. The results show that concatenating three different layers of color map information to the depth branch generates the best prediction results.

## 5 Conclusion

In this paper, we propose an image-guided depth enhancement system that extracts the features of color images to enhance the depth values of object boundaries through the residual dense network. To enable the network to focus more on enhancing the depth value of object boundaries, we propose Sobel loss to increase the weight of object edges. Regarding the concept of focal loss used in object detection, we further propose depth focal loss to improve the performance of network prediction. In addition, the inclusion of color information to the first half of the depth branch shows benefits for depth map restoration. We simulate the situation where the depth values of the object boundaries are intentionally mismatched to the color map in order to create a training dataset on Scene Flow dataset. Using this dataset to train and compare with other advanced methods, the proposed IGDE system obtains the best prediction results from multiple data. Finally, the ablation study shows that each function proposed in this paper effectively improves prediction results.

### Abbreviations

3D: Three dimension; DIBR: Depth image-based rendering; 2D: Two dimension; JBF: Joint bilateral filtering; SAD: Summation of absolute differences; DECNN: Denoising and enhancement convolutional neural network; DRECNN: Deep residual convolution neural network; IGDE: Image-guided depth enhancement; ReLU: Rectified linear unit; AlexNet: Alex network; CNN: Convolution neural network; EDSR: Enhanced deep residual network; DenseNet: Densely network; SRDenseNet: Super-resolution DenseNet; RDN: Residual dense network; RDB: Residual dense block; LR: Low resolution; HR: High resolution; MSE: Mean square error; MAE: Mean absolute error; RCNN: Region-based convolutional neural networks; RFCN: Region-based fully convolutional network; YOLO: You only look once; SSD: Single shot multibox detector; L1: 1 Norm; CUDA: Compute unified device architecture; cuDNN: CUDA<sup>®</sup> deep neural network; GPU: Graph processing unit; PSNR: Peak signal-to-noise ratio; dB: Decibel; SSIM: Structural similarity; RMSE: Root mean square error.

### Author contributions

All the authors have made contributions to the current work. KT LEE devised the image processing study plan, participated in the proposed system, and drafted the manuscript. ER Liu carried out software simulations, conducted the experiment, and collected the data. JF Yang and L Hong conceived of the study, and participated in its design and coordination, and helped modify the manuscript. All the authors read and approved the final manuscript.

### Funding

This work was partially supported by the Ministry of Science and Technology under Grant MOST 109-2218-E-006-032, 110-2218-E-006-025-MBK and Qualcomm, USA under Grant SOW#NAT-435536.

### Availability of data and materials

The color images with corresponding depth maps are obtained from Scene Flow dataset [26]. The datasets generated for the current study are available from the corresponding author on reasonable request.

## Declarations

### Competing interests

The authors declare that they have no competing financial interests and all simulations were completed in National Cheng Kung University.

Received: 7 October 2021 Accepted: 22 March 2022

Published online: 15 April 2022

## References

1. S.C. Chan, H.Y. Shum, K.T. Ng, Image-based rendering and synthesis—technological advances and challenges. *IEEE Signal Process. Mag.* **24**(6), 22–33 (2007). <https://doi.org/10.1109/Msp.2007.905702>
2. C. Yan, B. Gong, Y. Wei, Y. Gao, Deep multi-view enhancement hashing for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(4), 1445–1451 (2021). <https://doi.org/10.1109/TPAMI.2020.2975798>
3. K. Tang, L. Shi, S. Guo, S. Pan, H. Xing, S. Su, P. Guo, Z. Chen and Y. He, "Vision locating method based RGB-D camera for amphibious spherical robots", in *IEEE International Conference on Mechatronics and Automation (ICMA)*, (2017)
4. H.M. Zhu, J.H. Yin, D. Yuan, SVCV: segmentation volume combined with cost volume for stereo matching. *IET Comput. Vision* **11**(8), 733–743 (2017). <https://doi.org/10.1049/iet-cvi.2016.0446>

5. N.Y.C. Chang, T.H. Tsai, B.H. Hsu, Y.C. Chen, T.S. Chang, Algorithm and architecture of disparity estimation with minicensus adaptive support weight. *IEEE Trans. Circuits Syst. Video Technol.* **20**(6), 792–805 (2010). <https://doi.org/10.1109/Tcsvt.2010.2045814>
6. A. Roy and S. Todorovic, "Monocular depth estimation using neural regression forest", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016)
7. C. Godard, O. Mac Aodha and G.J. Brostow, "Unsupervised monocular depth estimation with left-right consistency", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2017)
8. S. Kumari, R.R. Jha, A. Bhavsar and A. Nigam, "Autodepth: Single image depth map estimation via residual cnn encoder-decoder and stacked hourglass", in *IEEE International Conference on Image Processing (ICIP)*, (2019)
9. H.-M. Wang, C.-H. Huang, J.-F. Yang, Block-based depth maps interpolation for efficient multiview content generation. *IEEE Trans. Circuits Syst. Video Technol.* **21**(12), 1847–1858 (2011)
10. K.R. Vijayanagar, M. Loghman and J. Kim, "Refinement of depth maps generated by low-cost depth sensors", in *International SoC Design Conference (ISOCC)*, (2012)
11. O.P. Gangwal and B. Djapic, "Real-time implementation of depth map post-processing for 3D-TV in dedicated hardware", in *Digest of Technical Papers International Conference on Consumer Electronics (ICCE)*, (2010)
12. J. Kopf, M.F. Cohen, D. Lischinski, M. Uyttendaele, Joint bilateral upsampling. *ACM Trans. Graph. (ToG)* **26**(3), 96 (2007)
13. C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images", in *Sixth International Conference on Computer Vision* (IEEE Cat. No. 98CH36271), (1998)
14. C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(2), 295–307 (2015)
15. Y. Zhang, Y. Tian, Y. Kong, B. Zhong and Y. Fu, "Residual dense network for image super-resolution", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018)
16. Y.-T. Zhou, R. Chellappa, A. Vaid, B.K. Jenkins, Image restoration using a neural network. *IEEE Trans. Acoust. Speech Signal Process.* **36**(7), 1141–1151 (1988)
17. K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a Gaussian Denoiser: residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.* **26**(7), 3142–3155 (2017)
18. H.C. Burger, C.J. Schuler and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?", in *IEEE Conference on Computer Vision and Pattern Recognition*, (2012)
19. C. Yan, Z. Li, Y. Zhang, Y. Liu, X. Ji, Y. Zhang, Depth image denoising using nuclear norm and learning graph model. *ACM Trans. Multimed. Comput. Commun. Appl.* **16**(4), 1–17 (2020). <https://doi.org/10.1145/3404374>
20. X. Zhang and R. Wu, "Fast depth image denoising and enhancement using a deep convolutional network", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (2016)
21. K. He, J. Sun, X. Tang, Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(6), 1397–1409 (2012)
22. J. Zhu, J. Zhang, Y. Cao and Z. Wang, "Image guided depth enhancement via deep fusion and local linear regularization", in *IEEE International Conference on Image Processing (ICIP)*, (2017)
23. K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016)
24. T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal loss for dense object detection", in *Proceedings of the IEEE International Conference on Computer Vision*, (2017)
25. A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks. *Adv. Neural. Inf. Process. Syst.* **25**, 1097–1105 (2012)
26. B. Lim, S. Son, H. Kim, S. Nah and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, (2017)
27. G. Huang, Z. Liu, L. Van Der Maaten and K.Q. Weinberger, "Densely connected convolutional networks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017
28. T. Tong, G. Li, X. Liu and Q. Gao, "Image super-resolution using dense skip connections", in *Proceedings of the IEEE International Conference on Computer Vision*, (2017)
29. S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks. *Adv. Neural. Inf. Process. Syst.* **28**, 91–99 (2015)
30. J. Dai, Y. Li, K. He and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks", in *Advances in Neural Information Processing Systems*, (2016)
31. J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016)
32. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A.C. Berg, "SSD: Single shot multibox detector", in *European Conference on Computer Vision*, (2016)
33. C. Yan, Y. Hao, L. Li, J. Yin, A. Liu, Z. Mao, Z. Chen, X. Gao, Task-adaptive attention for image captioning. *IEEE Trans. Circuits Syst. Video Technol.* **32**(1), 43–51 (2022). <https://doi.org/10.1109/TCSVT.2021.3067449>
34. C. Yan, L. Meng, L. Li, J. Zhang, J. Yin, J. Zhang, Z. Wang, B. Zheng, Age-invariant face recognition by multi-feature fusion and decomposition with self-attention. *ACM Trans. Multimed. Comput. Commun. Appl.* **18**(1), 1–18 (2022). <https://doi.org/10.1145/3472810>
35. W. Zhou, X. Li and D. Reynolds, "Guided deep network for depth map super-resolution: How much can color help?", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (2017)
36. C. Yan, T. Teng, Y. Liu, Y. Zhang, H. Wang, X. Ji, Precise no-reference image quality evaluation based on distortion identification. *ACM Trans. Multimed. Comput. Commun. Appl.* **17**(3), 1–21 (2021). <https://doi.org/10.1145/3468872>
37. N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy and T. Brox, "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*, (2016)

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.