Direction-aware Spatial Context Features for Shadow Detection and Removal

Xiaowei Hu, *Student Member, IEEE*, Chi-Wing Fu, *Member, IEEE,*
Lei Zhu, Jing Qin, *Member, IEEE,* and Pheng-Ann Heng, *Senior Member, IEEE*

Abstract—Shadow detection and shadow removal are fundamental and challenging tasks, requiring an understanding of the global image semantics. This paper presents a novel deep neural network design for shadow detection and removal by analyzing the image context in a direction-aware manner. To achieve this, we first formulate the direction-aware attention mechanism in a spatial recurrent neural network (RNN) by introducing attention weights when aggregating spatial context features in the RNN. By learning these weights through training, we can recover direction-aware spatial context (DSC) for detecting and removing shadows. This design is developed into the DSC module and embedded in a convolutional neural network (CNN) to learn the DSC features in different levels. Moreover, we design a weighted cross entropy loss to make effective the training for shadow detection and further adopt the network for shadow removal by using a Euclidean loss function and formulating a color transfer function to address the color and luminosity inconsistency in the training pairs. We employ two shadow detection benchmark datasets and two shadow removal benchmark datasets, and perform various experiments to evaluate our method. Experimental results show that our method clearly outperforms state-of-the-art methods for both shadow detection and shadow removal.

Index Terms—Shadow detection, shadow removal, spatial context features, deep neural network.

1 Introduction

Shadow is a monocular visual cue for perceiving depth and geometry. On the one hand, knowing the shadow location allows us to obtain the lighting direction [2], camera parameters [3] and scene geometry [4], [5]. On the other hand, the presence of shadows could, however, deteriorate the performance of many computer vision tasks, e.g., object detection and tracking [6], [7]. Hence, shadow detection and shadow removal have long been fundamental problems in computer vision research.

Early approaches detect and remove shadows by developing physical models to analyze the statistics of color and illumination [7], [8], [9], [10], [11], [12], [13], [14], [15]. However, these approaches are built on assumptions that may not be physically correct [16]. To distill the knowledge from real images, the data-driven approach learns and understands shadows by using hand-crafted features [17], [18], [19], [20], [21], or by learning the features using deep neural networks [16], [22], [23], [24], [25], [26]. While the state-of-the-art methods are already able to detect shadows with an accuracy of 87% to 90% [19], [23] and recover most shadow regions [25], [26], they may misunderstand black objects as shadows and produce various artifacts; see Sections 4 & 5 for quantitative and qualitative comparison results.

Understanding shadows requires exploiting the global image semantics, as shown very recently by V. Nguyen *et al.* [24] for shadow detection and L. Qu *et al.* [25] for shadow removal. To improve the understanding, we propose to analyze the image contexts in a *direction-aware* manner. Taking region A in Figure 1 as an example, comparing it with regions B and C, region B would give a stronger indication (than region C) that A is a shadow. Hence, spatial contexts in different directions would give different amount of contributions in suggesting the presence of shadows.

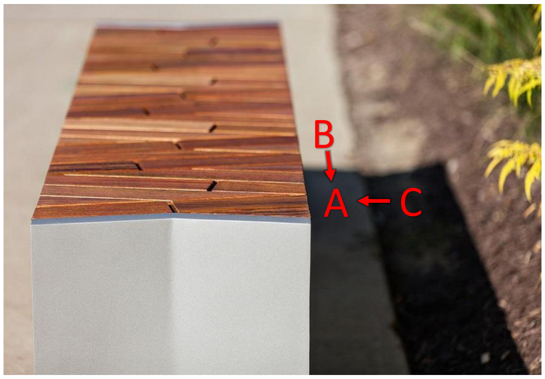


Fig. 1: In this example image, region B would give a stronger indication that A is a shadow compared to region C. This motivates us to analyze the global image context in a direction-aware manner for detecting and removing shadows.

To take directional variance into account when reasoning the image/spatial contexts, we first design a network module called the *direction-aware spatial context* (DSC) module, or *DSC module* for short, by adopting a spatial recurrent neural network (RNN) to aggregate spatial contexts in four principal directions, and by formulating the direction-aware attention mechanism in the RNN to learn attention weights for each direction. Then, we embed multiple copies of this DSC module in a convolutional neural network to learn the DSC features in different layers (scales), and combine the DSC features with the convolutional features

- *X. Hu is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. E-mail: xwhu@cse.cuhk.edu.hk.*
- *L. Zhu and J. Qin are with Centre for Smart Health, School of Nursing, The Hong Kong Polytechnic University.*
- *C.-W. Fu and P.-A. Heng are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong and Guangdong Provincial Key Laboratory of Computer Vision and Virtual Reality Technology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China.*
- *A preliminary version of this work was accepted for presentation in CVPR 2018 [1]. The source code is publicly available at https://xw-hu.github.io/.*

to predict a shadow mask for each layer. After that, we fuse the predictions at different layers into the final shadow detection result with the weighted cross entropy loss to optimize the network.

To further adopt and train the network for shadow removal, we replace shadow masks with shadow-free images as the ground truths, and use a Euclidean loss between the training pairs (images with and without shadows) to predict the shadow-free images. In addition, due to variations in camera exposure and environmental lighting, the training pairs may have inconsistent colors and luminosity; such inconsistency can be observed in existing shadow removal datasets such as SRD [25] and ISTD [26]. To this end, we formulate a transfer function to adjust the shadow-free ground truths and use the adjusted ground truths to train the network, so that our shadow removal network can produce shadow-free images that are more faithful to the input test images.

We summarize the major contributions of this work below:

- First, we design a novel attention mechanism in a spatial RNN and construct the DSC module to learn the spatial contexts in a direction-aware manner.
- Second, we develop a new network for shadow detection by adopting multiple DSC modules to learn the direction-aware spatial contexts in different layers and by designing a weighted cross entropy loss to balance the detection accuracy in shadow and non-shadow regions.
- Third, we further adopt the network for shadow removal by formulating a Euclidean loss and training the network with color-compensated shadow-free images, which are produced through a color transfer function.
- Last, we evaluate our method on several benchmark datasets on shadow detection and shadow removal, and compare it with the state-of-the-art methods. Experimental results show that our network clearly outperforms previous methods for both tasks by a significant margin; see Sections 4 & 5 for quantitative and qualitative comparison results.

2 Related Work

In this section, we focus on discussing works on single-image shadow detection and removal rather than trying to be exhaustive.

Shadow detection. Traditionally, single-image shadow detection methods [8], [9], [10] exploit physical models of illumination and color. This approach, however, tends to produce satisfactory results only for wide dynamic range images [18], [24]. Another approach learns shadow properties using hand-crafted features based on annotated shadow images. It first describes image regions by feature descriptors and then classifies the regions into shadow and non-shadow regions. Features like color [18], [27], [28], texture [19], [27], [28], edge [17], [18], [19] and T-junction [18] are commonly used for shadow detection followed by classifiers like decision tree [18], [19] and SVM [17], [27], [28]. However, since hand-crafted features have limited capability in describing shadows, this approach often fails for complex cases.

Convolutional neural network (CNN) is recently demonstrated to be a very powerful tool to learn features for detecting shadows, with results clearly outperforming previous approaches. Khan et al. [22] used multiple CNNs to learn features in super pixels and along object boundaries, and fed the output features to a conditional random field to locate shadows. Shen et al. [30] presented a deep structured shadow edge detector and employed structured labels to improve the local consistency of the predicted shadow map. Vicente et al. [23] trained stacked-CNN using a large dataset with noisy annotations. They minimized the sum of squared leave-one-out errors for image clusters to recover the annotations, and trained two CNNs to detect shadows.

Recently, Hosseinzadeh et al. [31] detected shadows using a patch-level CNN and a shadow prior map generated from hand-crafted features, while Nguyen et al. [24] developed scGAN with a sensitivity parameter to adjust weights in the loss functions. Though the shadow detection accuracy keeps improving on the benchmarks [19], [23], existing methods may still misrecognize black objects as shadows and miss unobvious shadows. The most recent work by Nguyen et al. [24] emphasized the importance of reasoning global semantics for detecting shadows. Compared to this work, we further consider the directional variance when analyzing the spatial context. Experimental results show that our method further outperforms [24] on the benchmarks for both the accuracy and BER value.

Shadow removal. Early works remove shadows by developing physical models deduced from the process of image formation [7], [11], [12], [13], [14], [15]. However, these approaches are not effective to describe the shadows in complex real scenes [16]. Afterwards, statistical learning methods are developed for shadow removal based on hand-crafted features (e.g., intensity [20], [21], [32], color [20], texture [20] and gradient [21]), which lack of the high-level semantic knowledge for discovering shadows.

Lately, features learned by convolutional neural networks (CNNs) are widely used for shadow removal. Khan et al. [16] applied multiple CNNs to learn to detect shadows, and formulated a Bayesian model to extract shadow matte and remove shadows in a single image. Very recently, Qu et al. [25] presented an architecture to remove shadows in end-to-end manner. The method applied three embedding networks (global localization network, semantic modeling network and appearance modeling network) to extract features in three levels. Wang et al. [26] designed two conditional generative adversarial networks in one framework to detect and remove shadows simultaneously. However, shadow removal is a challenge task; as pointed by Qu et al. [25] and Wang et al. [26], shadow removal needs a global view of the image to achieve global consistency in the prediction results. However, existing methods may still fail to reasonably restore the shadow regions, and they may also mistakenly change the colors in the non-shadow regions. In this work, we analyze the global image semantics in a direction-aware manner and formulate a color compensation mechanism to adjust pixel colors and luminosity by considering the non-shadow regions between the training pairs in the current benchmark datasets [25], [26]. Experimental results demonstrate that our method clearly outperforms the state-of-the-art methods, both qualitatively and quantitatively.

Difference from the conference paper. This work extends our earlier work [1] in three aspects. First, we adopt the shadow detection network with the DSC features to remove shadows by re-designing the outputs and formulating different loss functions to train the network. Second, we show that the pixel colors and luminosity in training pairs (shadow images and shadow-free images) of existing shadow removal datasets may not be consistent; to this end, we formulate a color compensation mechanism and use a transfer function to make consistent the pixel colors in ground truths before training our shadow removal network. Third, we perform more experiments to evaluate the design of our networks

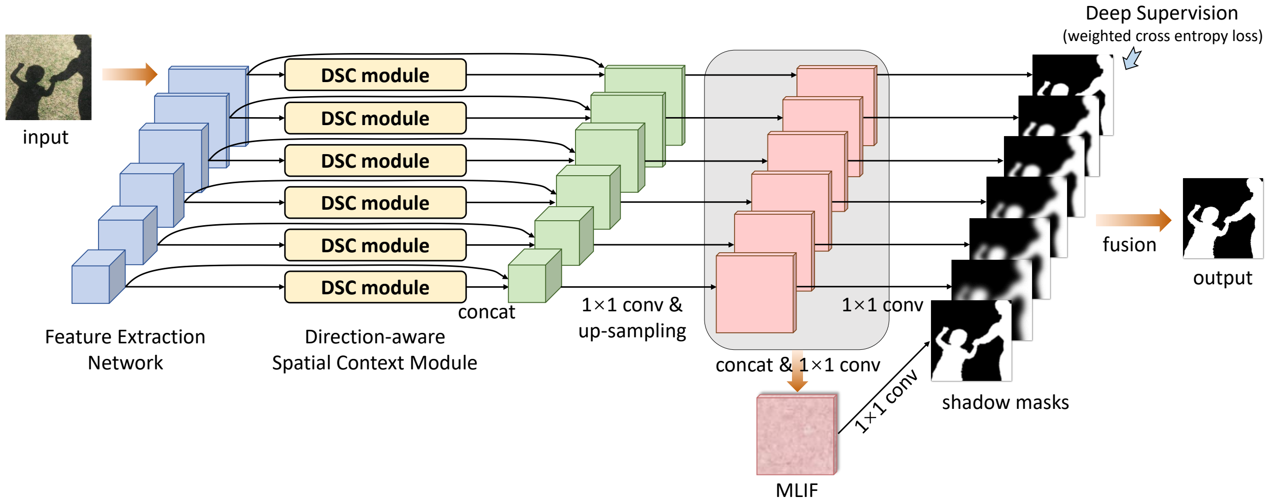


Fig. 2: The schematic illustration of the overall shadow detection network: (i) we extract features in different scales over the CNN layers from the input image; (ii) we embed a DSC module (see Figure 4) to generate direction-aware spatial context (DSC) features for each layer; (iii) we concatenate the DSC features with convolutional features at each layer and upsample the concatenated feature maps to the size of the input image; (iv) we combine the upsampled feature maps into the multi-level integrated features (MLIF), and predict a shadow mask based on the features for each layer by a deep supervision mechanism [29]; and (v) lastly, we fuse the resulting shadow masks to produce the final shadow detection result. See Section 3.3 for how we adopt this network for shadow removal.

for shadow detection and for shadow removal by considering more benchmark datasets and measuring the time performance, and show how our shadow removal network outperforms the best existing methods for shadow removal.

3 Methodology

Figure 2 presents the workflow of our overall shadow detection network, which employs multiple DSC modules (see Figure 4) to learn the direction-aware spatial context features in different scales. Our network takes the whole image as input and outputs the shadow mask in an end-to-end manner.

First, it begins by using a convolutional neural network (CNN) to extract a set of hierarchical feature maps, which encode the fine details and semantic information in different scales over the CNN layers. Second, for each layer, we employ a DSC module to harvest spatial contexts in a direction-aware manner and produce DSC features. Third, we concatenate the DSC features with the corresponding convolutional features, and upsample the concatenated feature maps to the size of the input image. Further, we combine the upsampled feature maps into multi-level integrated features (MLIF) with a convolution layer (via a 1×1 kernel), apply the deep supervision mechanism [29] to impose the supervision signals to each layer as well as to the MLIF, and predict a shadow mask for each of them. Lastly, we fuse all the predicted shadow masks into the final shadow detection output. To adopt the network for shadow removal, we replace shadow masks by shadow-free images as the ground truths, formulate a color compensation mechanism to adjust the shadow-free ground truth images for color and luminosity consistency, and use a Euclidean loss to optimize the network; see Section 3.3 for details.

In the following subsections, we first elaborate the DSC module that generates the DSC features (Section 3.1). After that, we present how we design the shadow detection network in Figure 2 using the DSC modules (Section 3.2), and then present how we adopt the network further for shadow removal (Section 3.3).

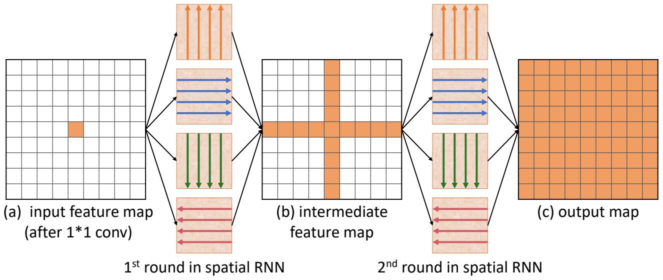


Fig. 3: The schematic illustration of how spatial context information propagates in a two-round spatial RNN.

3.1 Direction-aware Spatial Context

Figure 4 shows our DSC module architecture, which takes feature maps as input and outputs DSC features. In this subsection, we first describe the concept of spatial context features and the spatial RNN model (Section 3.1.1), and then elaborate how we formulate the direction-aware attention mechanism in a spatial RNN to learn the attention weights and generate DSC features (Section 3.1.2).

3.1.1 Spatial Context Features

Recurrent neural network (RNN) [33] is an effective model to process 1D sequential data via three arrays of nodes: an array of input nodes (to receive data), an array of hidden nodes (to update the internal states based on past and present data), and an array of output nodes (to output data). There are three kinds of data translations in an RNN: from input nodes to hidden nodes, from hidden nodes to output nodes, and between adjacent hidden nodes. By iteratively performing the data translations, the data received at the input nodes can be propagated across the hidden nodes, and eventually produce target results at the output nodes.

For processing image data with 2D spatial context, RNN has been extended to build the spatial RNN model [34]; see the schematic illustration in Figure 3. Taking a 2D feature map

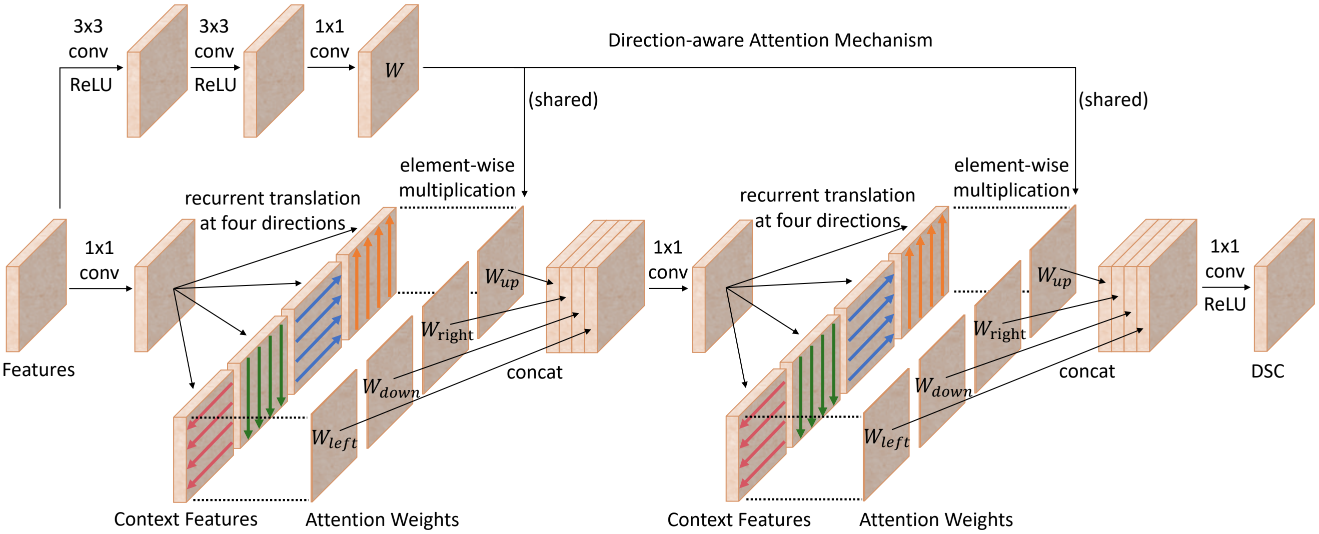


Fig. 4: The schematic illustration of the *direction-aware spatial context module* (*DSC module*). We compute direction-aware spatial context by adopting a spatial RNN to aggregate spatial contexts in four principal directions with two rounds of recurrent translations, and formulate the attention mechanism to generate maps of attention weights to combine context features for different directions. We use the same set of weights in both rounds of recurrent translations. Best viewed in color.

from a CNN as input, the spatial RNN model first uses a $1{\times}1$ convolution to perform an input-to-hidden data translation. Then, it applies four independent data translations to aggregate local spatial context along each principal direction (left, right, up, and down), and fuses the results into an intermediate feature map; see Figure 3(b). Lastly, the whole process is repeated to further propagate the aggregated spatial context in each principal direction and to generate the overall spatial context; see Figure 3(c).

Comparing with Figure 3(c), each pixel in Figure 3(a) knows only its local spatial context, while each pixel in Figure 3(b) further knows the spatial context in the four principal directions after the first round of data translations. Therefore, after two rounds of data translations, each pixel can obtain the necessary global spatial context for learning the features and solving the problem that the network is intended for.

To perform the data translations in a spatial RNN, we follow the IRNN model [35], since it is fast, easy to train, and has a good performance for long-range data dependencies [34]. Denoting $h_{i,j}$ as the feature at pixel (i,j), we perform one round of data translations to the right (similarly for the other directions) by repeating the following operation n times.

$$h_{i,j} = \max(\, \alpha_{\text{right}} \, h_{i,j-1} + h_{i,j} \; , \; 0 \,) \; , \tag{1}$$

where n is the width of the feature map and α_{right} is the weight parameter in the recurrent translation layer for the right direction. Note that α_{right}, as well as the weights for the other directions, are initialized to be an identity matrix and are learned by the training process automatically.

3.1.2 Direction-aware Spatial Context Features

To learn the spatial context in a direction-aware manner, we formulate the direction-aware attention mechanism in a spatial RNN to learn the attention weights and generate the direction-aware spatial context (DSC) features. This design forms the DSC module we presented above in Figure 4.

Direction-aware attention mechanism. The purpose of the direction-aware attention mechanism is to enable the spatial RNN to selectively leverage the spatial context aggregated along different directions by means of learning. See the top-left blocks in the DSC module shown in Figure 4. First, we employ two successive convolutional layers (with $3{\times}3$ kernels) followed by the ReLU [36] non-linear operation, and then the third convolutional layer (with $1{\times}1$ kernels) to generate $\mathbf{W}$. Then, we split $\mathbf{W}$ into four maps of attention weights denoted as $\mathbf{W}_{\text{left}}$, $\mathbf{W}_{\text{down}}$, $\mathbf{W}_{\text{right}}$, and $\mathbf{W}_{\text{up}}$. Mathematically, if we denote the above operators as f_{att} and the input feature maps as $\mathbf{X}$, we have

$$\mathbf{W} \;=\; f_{att}(\, \mathbf{X} \; ; \; \theta \,) \; , \tag{2}$$

where θ denotes the parameters to be learned by f_{att}, and f_{att} is also known as the attention estimator network.

See again the DSC module shown in Figure 4. The four maps of weights are multiplied with the spatial context features (from the recurrent data translations) along different directions in an element-wise manner. Hence, after we train the network, the network can learn θ for producing suitable attention weights to selectively leverage the spatial context in the spatial RNN.

Completing the DSC module. Next, we give additional details about the DSC module. As shown in Figure 4, after we multiply the spatial context features with the attention weights, we concatenate the results and use a $1{\times}1$ convolution to simulate a hidden-to-hidden data translation and reduce the feature dimensions to a quarter of the dimension size. Then, we perform the second round of recurrent translations and use the same set of attention weights to select the spatial context. We empirically find that the network delivers higher performance, if we share the attention weights rather than using two separate sets of weights. Note that these attention weights are learnt based on the deep features extracted from the input images, so they may vary from images to images. Lastly, we use a $1{\times}1$ convolution followed by the ReLU [36] non-linear operation on the concatenated feature maps to simulate the hidden-to-output translation and produce the output DSC features.

3.2 Our Shadow Detection Network

Our network is built upon the VGG network [37], where we apply a DSC module to each layer, except for the first layer, which involves a large memory footprint.

3.2.1 *Training*

Loss function. In natural images, shadows usually occupy smaller areas in the image space than the non-shadow regions. Hence, if the loss function simply aims for the overall accuracy, it will incline to match the non-shadow regions, which have far more pixels. Therefore, we use a weighted cross-entropy loss to optimize the shadow detection network in the training process.

In detail, assume that the ground truth value of a pixel is y (where y=1, if it is in shadow, and y=0, otherwise) and the prediction label of the pixel is p (where $p \in [0, 1]$). The weighted cross entropy loss L equals $L_1 + L_2$:

$$L_1 = -(\frac{N_n}{N_p + N_n}) y \log(p) - (\frac{N_p}{N_p + N_n})(1-y)\log(1-p) \ , \tag{3}$$

and

$$L_2 = -(1 - \frac{TP}{N_p}) y \log(p) - (1 - \frac{TN}{N_n})(1-y)\log(1-p) \ , \tag{4}$$

where TP and TN are the number of true positives and true negatives, and N_p and N_n are the number of shadow and non-shadow pixels, respectively, so N_p+N_n is the total number of pixels in the image space. In practice, L_1 helps balance the detection of shadows and non-shadows; if the area of shadows is less than that of the non-shadow region, we will penalize misclassified shadow pixels more than the misclassified non-shadow pixels. On the other hand, L_2 helps the network focus on learning the class (shadow or non-shadow) that is difficult to be classified [38]. This can be achieved, since the weight in the loss function for shadow (or non-shadow) class is large when the number of correctly-classified shadow (or non-shadow) pixels is small, and vice versa.

We use the above loss function for each layer in the shadow detection network presented in Figure 2. Hence, the overall loss function L_{overall} is a summation of the individual loss on all the predicted shadow masks over the different scales:

$$L_{\text{overall}} \ = \ \sum_i w_i L_i \ + \ w_m L_m \ + \ w_f L_f \ , \tag{5}$$

where w_i and L_i denote the weight and loss of the i-th layer (level) in the overall network, respectively; w_m and L_m are the weight and loss of the MLIF layer; and w_f and L_f are the weight and loss of the fusion layer, which is the last layer in the overall network to produce the final detection result; see Figure 2. Note that all the weights w_i, w_m and w_f are empirically set to be one.

Training parameters. To accelerate the training process while reducing the overfitting, we initialize the parameters in the feature extraction layers (see the frontal part of the network in Figure 2) by the well-trained VGG network [37] and parameters in the other layers by random noise. Stochastic gradient descent is used to optimize the whole network with a momentum value of 0.9 and a weight decay of $5{\times}10^{-4}$. We set the learning rate as 10^{-8} and terminate the learning process after 12k iterations. Moreover, we horizontally flip images for data argumentation. Note that we build the model on Caffe [39] with a mini-batch size of one, and update the model parameters in every ten training iterations.

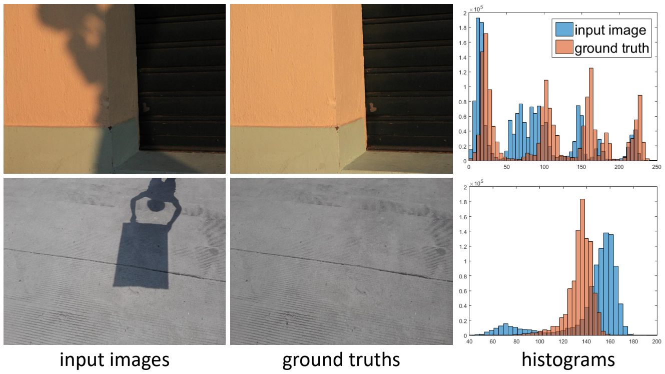


Fig. 5: Inconsistency between input images (shadow images) and ground truths (shadow-free images). Top row is “IMG_6456.jpg” from SRD [25] and bottom row is “109-5.png” from ISTD [26].

3.2.2 *Testing*

In the testing process, our network produces one shadow mask for each layer, including the MLIF layer and the fusion layer, with a supervision signal added to each layer. After that, we compute the mean of the shadow masks over the MLIF layer and the fusion layer to produce the final prediction result. Lastly, we apply the fully connected conditional random field (CRF) [40] to improve the detection result by considering the spatial coherence among the neighborhood pixels.

3.3 Our Shadow Removal Network

To adopt our shadow detection network shown in Figure 2 for shadow removal, we have the following three modifications:

- First, we formulate a color compensation mechanism to address the color inconsistency between the training pairs, i.e., shadow images (input images) and shadow-free images (ground truths), and then to adjust the shadow-free images (Section 3.3.1).
- Second, we replace the shadow masks by the adjusted shadow-free images as the supervision (i.e., ground truths) in the network for shadow removal; see Figure 2.
- Third, we replace the weighted cross-entropy loss by a Euclidean loss to train and optimize the network with the adjusted shadow-free images (Section 3.3.2).

3.3.1 *Color Compensation Mechanism*

Training data for shadow removal is typically prepared by first taking a picture of the scene with shadows, and then taking another picture without the shadows by removing the associated objects. Since the environmental luminosity and camera exposure may vary, a training pair may have inconsistent colors and luminosity; see Figure 5 for examples from two different benchmark datasets (SRD [25] & ISTD [26]), where the inconsistency is clearly revealed by the color histograms. Existing network-based methods learn to remove shadows by optimizing the network to produce an output that matches the target ground truth. Hence, given inconsistent training pairs, the network could produce biased results and make the predicted images brighter or darker.

To address the problem, we design a color compensation mechanism by finding a color transfer function for each pair of training images (input shadow images and ground truth shadow-free images). Let I_s and I_n be a shadow image (input) and a shadow-free image (ground truth) of a training pair, respectively, and Ω_s and Ω_n be the shadow region and non-shadow region, respectively, in the image space. In our formulation, we aim to find color transfer function T_f that minimizes the color compensation error E_c between the shadow image and shadow-free image over the non-shadow region (indicated by the shadow mask):

$$E_c \ = \ | \ I_s \ - \ T_f(I_n) \ |^2_{\Omega_n} \ . \tag{6}$$

We formulate T_f using the following linear transformation (which we empirically find sufficient for adjusting the colors in I_n to match with the colors in I_s):

$$T_f(x) \ = \ \mathbf{M}_\alpha \ \cdot \begin{pmatrix} r \\ g \\ b \\ 1 \end{pmatrix} \ , \tag{7}$$

where x is a pixel in I_n with color values (r, g, b) and $\mathbf{M}_\alpha$ is a $3{\times}4$ matrix, which stores the parameters in the color transfer function. Note that we solve Eq. (6) for T_f using the least-squares method by considering pixel pairs in the non-shadow regions of I_s and I_n. Then, we can apply T_f to adjust the whole image of I_n for each training pair, replace the shadow masks in Figure 2 by the adjusted shadow-free image (i.e., $T_f(I_n)$) as the new supervision, and train the shadow removal network in an end-to-end manner.

3.3.2 Training

Loss function. We adopt a Euclidean loss to optimize the shadow removal network. In detail, we denote the network prediction as $\tilde{I}_n$, use the "LAB" color space for both $T_f(I_n)$ and $\tilde{I}_n$ in the training, and calculate the loss L^r on the whole image domain:

$$L^r \ = \ | \ T_f(I_n) \ - \ \tilde{I}_n \ |^2_{\Omega_n \cup \Omega_s} \ . \tag{8}$$

We use the above loss function for each layer in the shadow removal network. The overall loss function L^r_{overall} is the summation of the loss L^r on all layers:

$$L^r_{\text{overall}} \ = \ \sum_i w^r_i L^r_i \ + \ w^r_m L^r_m \ + \ w^r_f L^r_f \ . \tag{9}$$

Similar to Eq. (5), we empirically set all the weights (w^r_i, w^r_m, and w^r_f) to be one.

Training parameters. Again, we initialize the parameters in the feature extraction layers (see the frontal part of the network shown in Figure 2) by the well-trained VGG network [37] to accelerate the training process and reduce over-fitting, and the parameters in the other layers are initialized by random noise. Adam [41] is used to optimize the shadow removal network with the first momentum value of 0.9, the second momentum value of 0.99, and a weight decay of $5{\times}10^{-4}$. This optimization approach adaptively adjusts the learning rates for each individual parameter in the network. It decreases the learning rate for the frequently-updated parameters and increases the learning rate for the rarely-updated parameters. We set the basic learning rate as 10^{-5} and reduce it by multiplying 0.316 at 90k and 130k iterations. The learning stops at 160k iterations. Moreover, the images are horizontally and vertically flipped, randomly cropped and rotated for data argumentation. The model is built on Caffe [39] with a mini-batch size of one.

3.3.3 Testing

In the testing process, our network directly produces shadow-free image for each layer, including the MLIF layer and the fusion layer, with a supervision signal added to each layer. After that, we compute the mean of the shadow-free images over the MLIF layer and the fusion layer to produce the final result.

4 Experiments on Shadow Detection

In this section, we present experiments to evaluate our shadow detection network: comparing it with the state-of-the-art methods, evaluating its network design and time performance, and showing shadow detection results. In the next section, we will evaluate the performance of shadow removal network.

4.1 Shadow Detection Datasets & Evaluation Metrics

Benchmark datasets. We employ two benchmark datasets. The first one is the SBU Shadow Dataset [23], which is the largest publicly available annotated shadow dataset with 4089 training images and 638 testing images, which cover a wide variety of scenes. The second dataset we employed is the UCF Shadow Dataset [19]. It includes 145 training images and 76 testing images, and covers outdoor scenes with various backgrounds. We train our shadow detection network using the SBU training set.

Evaluation metrics. We employ two commonly-used metrics to quantitatively evaluate the shadow detection performance. The first one is the accuracy metric:

$$accuracy \ = \ \frac{TP + TN}{N_p + N_n} \ , \tag{10}$$

where TP, TN, N_p and N_n are true positives, true negatives, number of shadow pixels, and number of non-shadow pixels, respectively, as defined in Section 3.2. Since N_p is usually much smaller than N_n in natural images, we employ the second metric called the balance error rate (BER) to obtain a more balanced evaluation by equally treating the shadow and non-shadow regions:

$$BER \ = \ (1 - \frac{1}{2}(\frac{TP}{N_p} + \frac{TN}{N_n})) \times 100 \ . \tag{11}$$

Note that unlike the accuracy metric, for BER, the lower its value, the better the detection result is.

4.2 Comparison with the State-of-the-arts

Comparison with shadow detection methods. We compare our method with four recent shadow detection methods: scGAN [24], stacked-CNN [23], patched-CNN [31] and Unary-Pairwise [27]. The first three are network-based methods, while the last one is based on hand-crafted features. For a fair comparison, we obtain their shadow detection results either directly from the results provided by the authors or by generating them using implementations provided by the authors with recommended parameter setting.

Table 1 reports the comparison results, from which we can see that our method outperforms all the others on both accuracy and BER for both benchmark datasets. Note that our shadow detection network is trained using the SBU training set [23], but it still outperforms others on the UCF dataset, thus showing its generalization capability. Further, we show visual comparison results in Figures 6 and 7, which show various challenging cases, e.g., a light shadow next to a dark shadow, shadows around complex backgrounds, and black objects around shadows. Without understanding the global image semantics, it is hard to locate these shadows, and the non-shadow regions could be easily misrecognized as shadows. From the results, we can see that our method can effectively locate shadows and avoid false positives compared

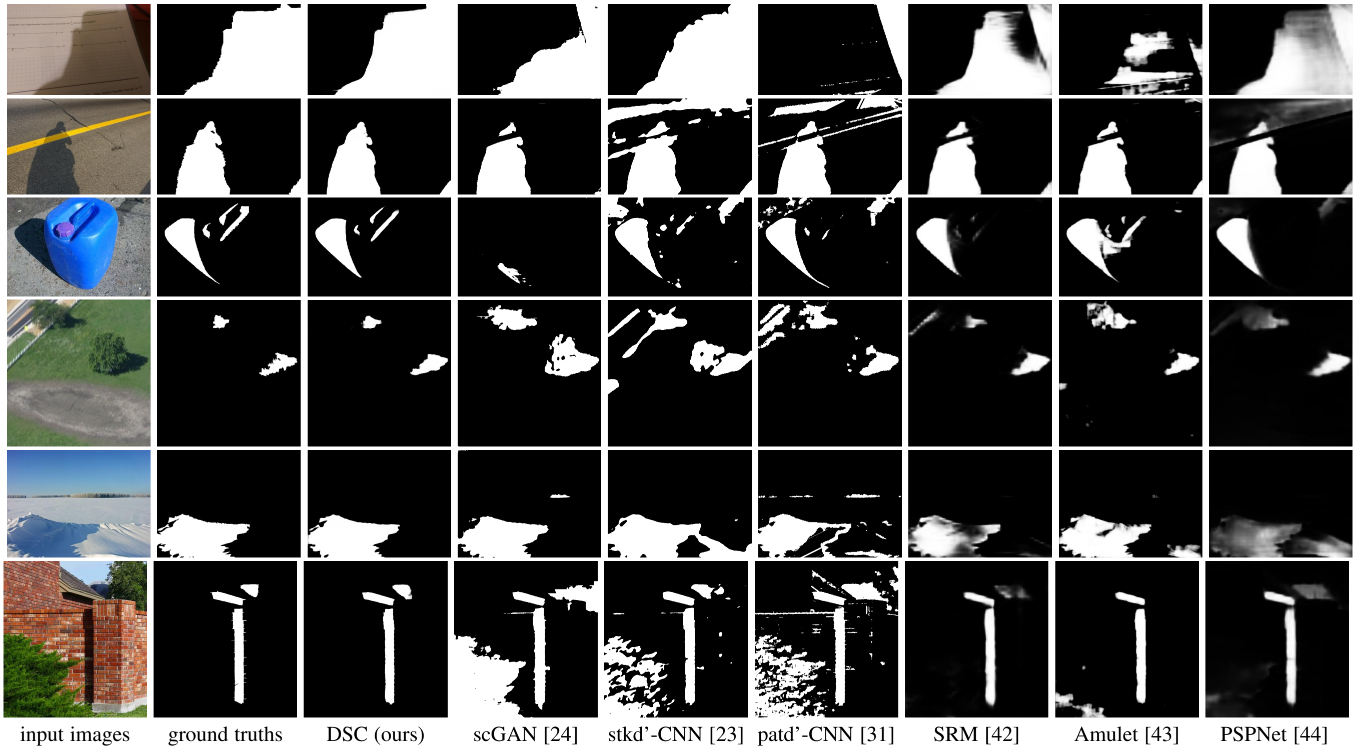


Fig. 6: Visual comparison of shadow masks produced by our method and other methods (4th-9th columns) against ground truths shown in 2nd column. Note that stkd'-CNN and patd'-CNN stand for stacked-CNN and patched-CNN, respectively.

to other methods, e.g., for black objects misrecognized as shadows by others, our method could still recognize them as non-shadows.

Comparison with saliency detection and semantic segmentation methods. Deep networks for saliency detection and semantic image segmentation may also be used for shadow detection by training the networks using datasets of annotated shadows. Thus, we perform another experiment using two recent deep models for saliency detection (SRM [42] and Amulet [43]) and a recent deep model for semantic image segmentation (PSPNet [44]).

For a fair comparison, we re-train their models on the SBU training set using implementations provided by the authors, and adjust the training parameters to obtain the best shadow detection results. The last three rows in Table 1 report the comparison results on accuracy and BER metrics. Although these methods achieve good results for both metrics, our method still outperforms them for both benchmark datasets. Please also refer to the last three columns in Figures 6 and 7 for visual comparison results.

4.3 Evaluation on the Network Design

Component analysis. We perform an experiment to evaluate the effectiveness of the DSC module design. Here, we use the SBU dataset and consider two baseline networks. The first baseline (denoted as "basic") is a network constructed by removing all the DSC modules from the overall network shown in Figure 2. The second baseline (denoted as "basic+context") considers spatial context but ignores the direction-aware attention weights. Compared with the first baseline, this network has all the DSC modules, but it removes the direction-aware attention mechanism inside the DSC modules, i.e., removing the computation of $\mathbf{W}$ and directly concatenating the context features without multiplying them with the attention weights; see Figure 4. This is equivalent to setting all the attention weights $\mathbf{W}$ to be one.

TABLE 1: Comparing our method (DSC) with the state-of-the-art methods for shadow detection (scGAN [24], stacked-CNN [23], patched-CNN [31] and Unary-Pairwise [27]), for saliency detection (SRM [42] and Amulet [43]), and for semantic image segmentation (PSPNet [44]).

	SBU [23]		UCF [19]	
method	accuracy	BER	accuracy	BER
DSC (ours)	**0.97**	**5.59**	**0.95**	**8.10**
scGAN [24]	0.90	9.10	0.87	11.50
stacked-CNN [23]	0.88	11.00	0.85	13.00
patched-CNN [31]	0.88	11.56	-	-
Unary-Pairwise [27]	0.86	25.03	-	-
SRM [42]	0.96	7.25	0.94	9.81
Amulet [43]	0.93	15.13	0.92	15.17
PSPNet [44]	0.95	8.57	0.93	11.75

TABLE 2: Component analysis. We train three networks using the SBU training set and test them using the SBU testing set [23]: "basic" denotes the architecture shown in Figure 4 but without all DSC modules; "basic+context" denotes the "basic" network with spatial context but not direction-aware spatial context; and "DSC" is the overall network in Figure 4.

network	BER	improvement
basic	6.55	-
basic+context	6.23	4.89%
DSC	**5.59**	10.27%

Table 2 reports the comparison results, showing that our basic network with multi-scale features and the weighed cross entropy loss function can produce good results. Moreover, considering spatial context and DSC features can lead to further obvious improvement; see also Figure 8 for visual comparison results.

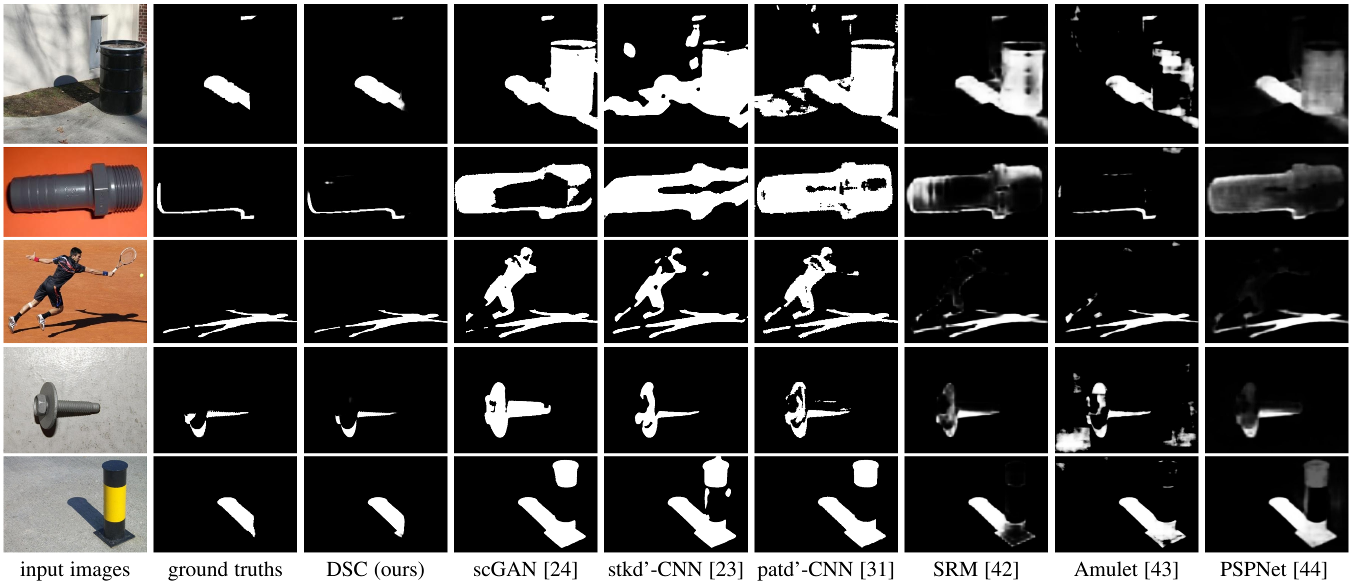


Fig. 7: More visual comparison results on shadow detection (continue from Figure 6).

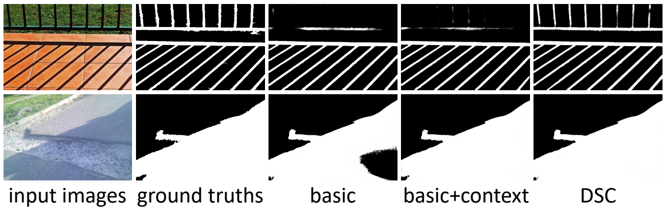


Fig. 8: Visual comparison results of component analysis.

DSC architecture analysis. We encounter two questions when designing the network structure with DSC modules: (i) how many rounds of recurrent translations in the spatial RNN; and (ii) whether to share the attention weights or to use separate attention weights in different rounds of recurrent translations.

We modify our network for these two parameters and produce the comparison results shown in Table 3. From the results, we can see that having two rounds of recurrent translations and sharing the attention weights in both rounds produce the best result. We believe that when there is only one round of recurrent translations, the global context information cannot be well propagated over the spatial domain, so there is insufficient information exchange for learning the shadows, while having three rounds of recurrent translations with separate copies of attention weights will introduce excessive parameters that make the network hard to be trained.

Feature extraction network analysis. We perform an experiment to evaluate the feature extraction network as shown in Figure 2. We use the deeper network, ResNet-101 [45] with 101 layers, to replace the VGG network, which only has 16 layers. Taking the ResNet-101 into account, we use res2c, res3b3, res4b22, and res5c to produce the DSC features at different scales and keep the other network parts and parameter settings unchanged.

The BER values we obtained are 5.59 and 5.73 for VGG network and ResNet-101, respectively, showing that they have similar performance. The deeper network provides stronger semantic features, but it loses detail information due to the small-sized feature maps when accounting for the limited GPU memory.

TABLE 3: DSC architecture analysis. By varying the parameters in the DSC architecture (see the second and third columns below), we can have produce a slightly different overall network and explore their performance (see the last column).

number of rounds	shared $\mathbf{W}$?	BER
1	-	5.85
2	Yes	**5.59**
3	Yes	5.85
2	No	6.02

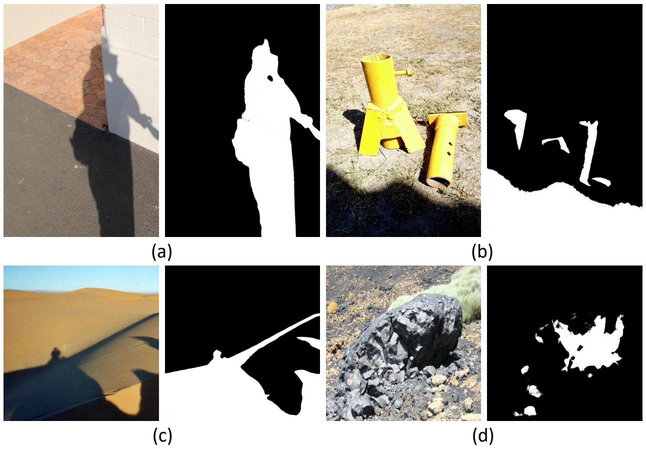


Fig. 9: More shadow detection results produced from our method.

4.4 Additional Results

More shadow detection results. Figure 9 shows more results: (a) light and dark shadows next to each other; (b) small and unconnected shadows; (c) no clear boundary between shadow and non-shadow regions; and (d) shadows of irregular shapes. Our method can still detect these shadows fairly well, but it fails in some extremely complex scenes: (a) a scene with many small shadows (see the 1^{st} row in Figure 11), where the features in the deep layers lose the detail information and features in the shallow layers lack the semantics for the shadow context; (b) a scene with a large black region (see the 2^{nd} row in Figure 11), where there are

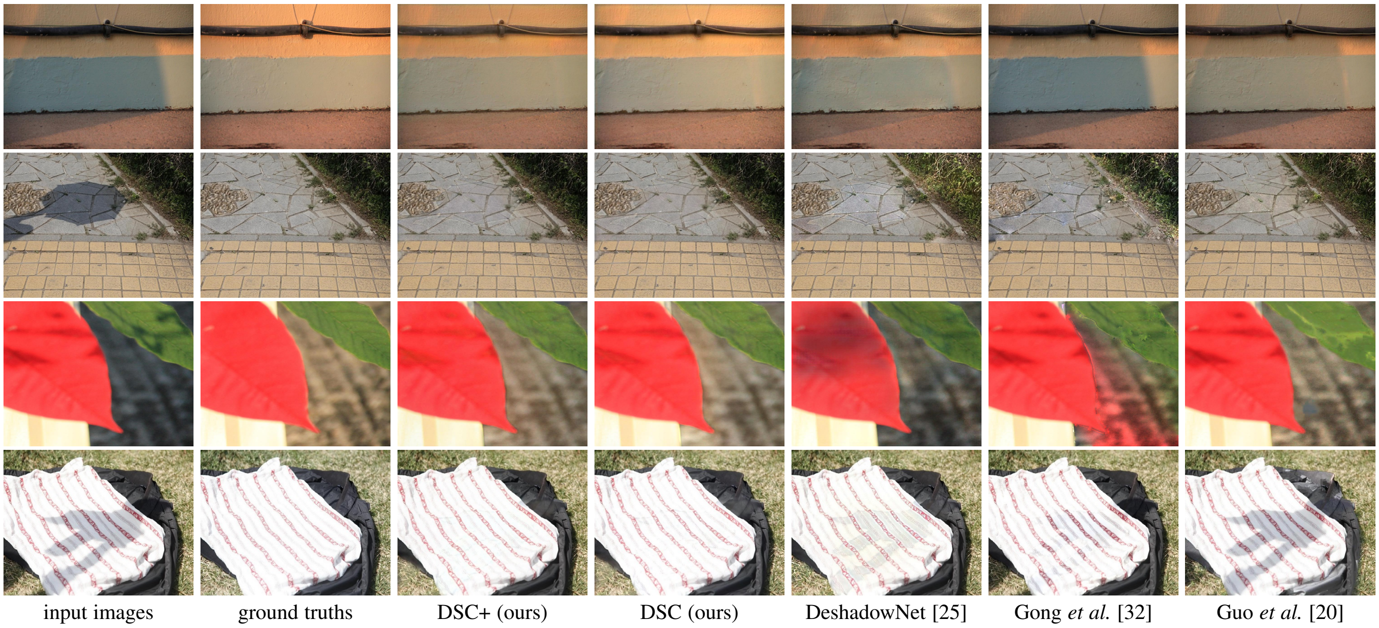


Fig. 10: Visual comparison of shadow removal results on the SRD dataset [25].

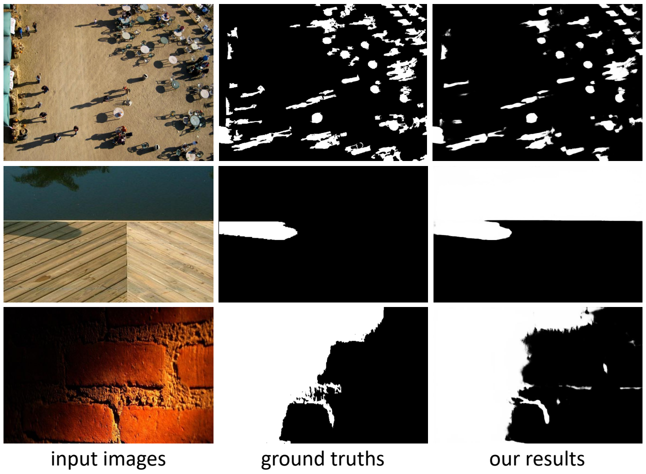


Fig. 11: Failure cases on shadow detection.

insufficient surrounding context to indicate whether it is a shadow or simply a black object; and (c) a scene with soft shadows (see the 3^{rd} row in Figure 11), where the difference between the soft shadow regions and the non-shadow regions is small.

Time performance. Our network is fast enough, due to its fully convolutional architecture and the simple implementation of RNN model [35]. We trained and tested our network for shadow detection on a single GPU (NVIDIA GeForce TITAN Xp). It takes around 16.5 hours to train the whole network on the SBU training set and around 0.16 seconds on average to process one image (400×400) in testing. For the post-processing step with CRF [40], it takes another 0.5 seconds for testing each image.

5 Experiments on Shadow Removal

5.1 Shadow Removal Datasets & Evaluation Metrics

Benchmark datasets. We employ two shadow removal benchmark datasets. The first one is SRD [25], which is the first large-scale dataset with shadow image and shadow-free image pairs, containing 2680 training pairs and 408 testing pairs. It includes images under different illuminations and a variety of scenes, and the shadows are casted on different reflectance phenomena with various shapes and silhouettes. The second one is ISTD [26], which contains the triplets of shadow image, shadow mask, and shadow-free image, including 1330 training triplets and 540 testing triplets. This dataset covers various shadow shapes under 135 different cases of ground materials.

TABLE 4: Comparing our method (DSC) with the state-of-the-art methods for shadow removal in terms of RMSE. Note that the code of ST-CGAN [26] and DeshadowNet [25] are not publicly available, so we can only directly compare with their RMSE results (i.e., 6.64 and 7.47) on their respective datasets.

	SRD [25]	ISTD [26]
DSC (ours)	**6.21**	**6.67**
ST-CGAN [26]	-	7.47
DeshadowNet [25]	6.64	-
Gong *et al.* [32]	8.73	8.53
Guo *et al.* [20]	12.60	9.30
Yang *et al.* [46]	22.57	15.63

Evaluation metrics. We quantitatively evaluate the shadow removal performance by calculating the root-mean-square error (RMSE) in "LAB" color space between the ground truth and the predicted shadow-free image, following [20], [25], [26]. Hence, a low RMSE value indicates good performance.

5.2 Comparison with the State-of-the-arts

The state-of-the-art shadow removal methods compute the RMSE directly between the predicted shadow-free image and the ground truth shadow-free image (without any color adjustment). Hence, for a fair quantitative comparison between our method and the state-of-the-art methods, we apply our network trained on the original shadow-free images that are without the color adjustment. We denote this network as "DSC", and our network trained on the shadow-free images with the color adjustment as "DSC+".

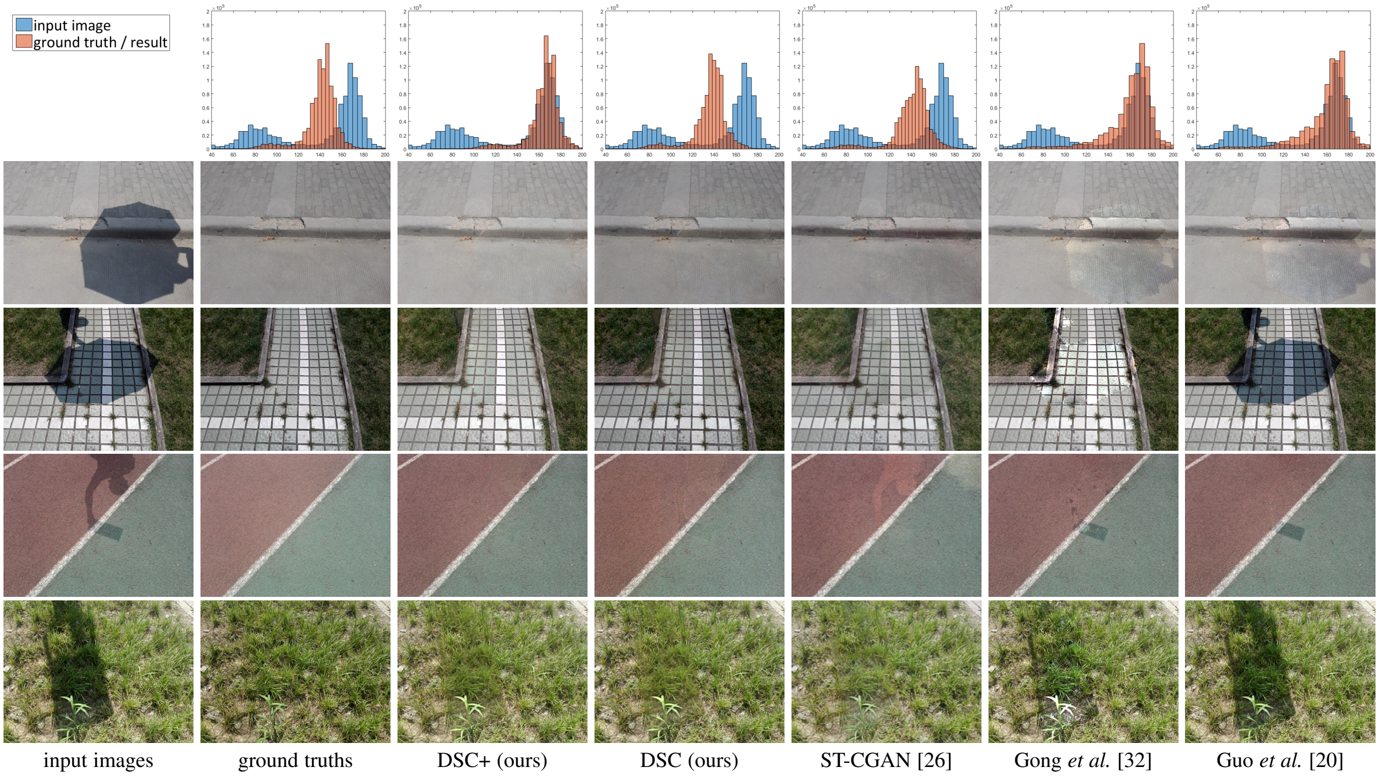


Fig. 12: Visual comparison of shadow removal results on the ISTD dataset [26]. The histograms in the top (first) row reveal the intensity distribution of the images in the second row, where the blue histograms show the intensity distribution of the leftmost input image and the red histograms show the intensity distribution of the ground truth or result images in the second row below the histograms.

TABLE 5: Evaluate our methods (DSC & DSC+) on the original ground truth (I_n) and the adjusted ground truth ($T_f(I_n)$). The performance is evaluated by using the RMSE metric.

	SRD [25]		ISTD [26]	
	I_n	$T_f(I_n)$	I_n	$T_f(I_n)$
DSC	6.21	6.66	6.67	8.54
DSC+	-	**6.12**	-	**4.90**

We consider the following five recent shadow removal methods in our comparison: ST-CGAN [26], DeshadowNet [25], Gong *et al.* [32], Guo *et al.* [20], and Yang *et al.* [46]. We obtain their shadow removal results directly from the authors or by generating them using the public code with the recommended parameter setting. Table 4 presents the comparison results; note that we do not have the result of ST-CGAN [26] on the SRD dataset [25] (and similarly, the result of DeshadowNet [25] on the ISTD dataset [26]), since we do not have the code for these two methods. DeshadowNet [25] and ST-CGAN [26] are the two most recent shadow removal methods, which exploit the global image semantics in a convolutional neural network by a multi-context architecture and adversarial learning. By further considering the global context information in a direction-aware manner, we can see from Table 4 that our method outperforms them on respective dataset, demonstrating the effectiveness of our network.

We provide visual comparison results on these two datasets in Figures 10 and 12, which show several challenging cases, e.g., dark non-shadow regions (the second row in Figure 10) and shadows across multiple types of backgrounds. From the results, we can see that our methods (DSC & DSC+) can effectively remove the shadows as well as maintain the input image contents in the non-shadow regions. By introducing the color compensation mechanism, our DSC+ model can further produce shadow-free images that are more consistent with the input images. In the comparison results, other methods may change the colors on the non-shadow regions or fail to remove parts of the shadows.

5.3 Evaluation on the Network Design

Color compensation mechanism analysis. The first two pairs of images (input images and ground truths) in the 2nd and 3rd rows of Figure 12 show the inconsistent color and luminosity (also revealed by the first histogram on the top row). Methods based on neural networks (e.g., DSC and ST-CGAN [26]) could produce inconsistent results due to inconsistency in the training pairs; see the 4th and 5th columns in Figure 12 from the left.

By first adjusting the ground truth shadow-free images, our DSC+ can learn to generate shadow-free images whose colors are more consistent and faithful to the input images; see the 3rd column in Figure 12 and the histogram above. Furthermore, we tried to use the adjusted shadow-free images ($T_f(I_n)$) instead of the original shadow-free images (I_n) as the ground truths to compute the RMSE for our methods (DSC and DSC+). Table 5 shows the comparison results: DSC has a large RMSE when compared with the adjusted ground truths (6.21 vs. 6.66 and 6.67 vs. 8.54), while DSC+ shows a clear improvement (6.66 vs. 6.12 and 8.54 vs. 4.90), especially on the ISTD dataset [26].

Color space analysis. We performed another experiment to evaluate the choice of color space in the data processing. In this experiment, we consider the "LAB" and "RGB" color spaces, and

TABLE 6: Train and test our method (DSC) on different color spaces. The performance is evaluated by using the RMSE metric.

color space	SRD [25]	ISTD [26]
LAB	6.21	**6.67**
RGB	**6.05**	6.92

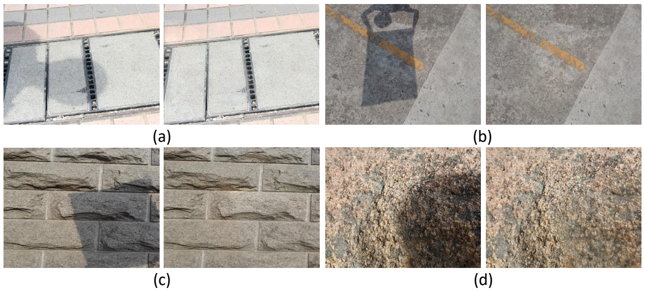


Fig. 13: More shadow removal results produced from our DSC+.

train a shadow removal network for each of them. As shown in the results presented in Table 6, the performance of the two networks are similar. Since the overall (summed) performance with the LAB color space is slightly better, we thus choose to use LAB in our method. However, in any case, both results clearly outperform the state-of-the-art methods on shadow removal shown in Table 4.

5.4 Additional Results

More shadow removal results. Figure 13 presents more results: (a) and (b) show shadows across backgrounds of different colors, (c) shows small, unconnected shadows of irregular shapes on the stones, and (d) shows shadows on a complex background. Our method can still reasonably remove these shadows. However, for the cases shown in Figure 14: (a) it overly removes the fragmented black tiles on the floor (see the red dashed boxes in the figure), where the surrounding context provides incorrect information, and (b) it fails to recover the original (bright) color of the handbag, due to the lack of information. We believe that more training data is needed for the network to learn and overcome these problems.

Time performance. Same as our shadow detection network, we trained and tested our shadow removal network on the same GPU (NVIDIA GeForce TITAN Xp). It takes around 22 hours to train the whole network on the SRD training set and another 22 hours to train it on the ISTD training set. In testing, it only needs around 0.16 seconds on average to process a 400×400 image.

6 Conclusion

We present a novel network for single-image shadow detection and removal by harvesting direction-aware spatial context. Our key idea is to analyze multi-level spatial context in a direction-aware manner by formulating a direction-aware attention mechanism in a spatial RNN. By training the network to automatically learn the attention weights for leveraging and composing the spatial context in different directions in a spatial RNN, we can produce direction-aware spatial context (DSC) features and formulate the DSC module. Then, we adopt multiple DSC modules in a multi-layer convolutional neural network to detect shadows by predicting the shadow masks in different scales, and design a weighted cross entropy loss function to make effective the training process. Further, we adopt the network for shadow removal by replacing the shadow masks with shadow-free images, applying a Euclidean loss to optimize the network, and introducing a color compensation mechanism to address the color and luminosity inconsistency problem. In the end, we test our network on two benchmark datasets for shadow detection and another two benchmark datasets for shadow removal, compare our network with various state-of-the-art methods, and show its superiority over the state-of-the-art methods for both shadow detection and shadow removal.

In future, we plan to explore our network for other applications such as saliency detection and semantic segmentation, further enhance the shadow removal results by exploring strategies in image completion, and studying time-varying shadows in videos.

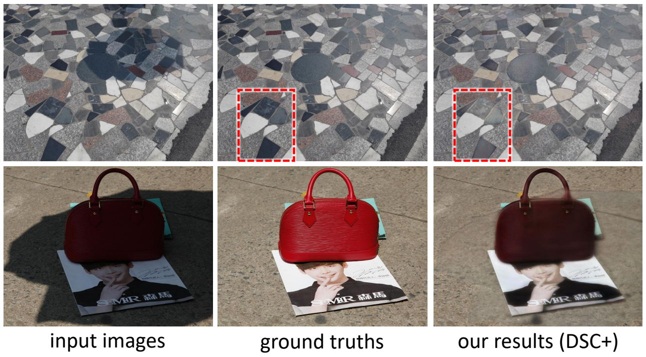


Fig. 14: Failure cases on shadow removal.

References

[1] X. Hu, L. Zhu, C.-W. Fu, J. Qin, and P.-A. Heng, “Direction-aware spatial context features for shadow detection,” in *CVPR*, 2018, oral presentation, to appear.

[2] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, “Estimating natural illumination from a single outdoor image,” in *ICCV*, 2009, pp. 183–190.

[3] I. N. Junejo and H. Foroosh, “Estimating geo-temporal location of stationary cameras using shadow trajectories,” in *ECCV*, 2008, pp. 318–331.

[4] T. Okabe, I. Sato, and Y. Sato, “Attached shadow coding: Estimating surface normals from shadows under unknown reflectance and lighting conditions,” in *ICCV*, 2009, pp. 1693–1700.

[5] K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem, “Rendering synthetic objects into legacy photographs,” *ACM Transactions on Graphics (SIGGRAPH Asia)*, vol. 30, no. 6, pp. 157:1–157:12, 2011.

[6] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, “Detecting moving objects, ghosts, and shadows in video streams,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1337–1342, 2003.

[7] S. Nadimi and B. Bhanu, “Physical models for moving shadow and object detection in video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1079–1087, 2004.

[8] E. Salvador, A. Cavallaro, and T. Ebrahimi, “Cast shadow segmentation using invariant color features,” *Computer Vision and Image Understanding*, vol. 95, no. 2, pp. 238–259, 2004.

[9] A. Panagopoulos, C. Wang, D. Samaras, and N. Paragios, “Illumination estimation and cast shadow detection through a higher-order graphical model,” in *CVPR*, 2011, pp. 673–680.

[10] J. Tian, X. Qi, L. Qu, and Y. Tang, “New spectrum ratio properties and features for shadow detection,” *Pattern Recognition*, vol. 51, pp. 85–96, 2016.

[11] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew, “On the removal of shadows from images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 59–68, 2006.

[12] G. D. Finlayson, S. D. Hordley, and M. S. Drew, “Removing shadows from images,” in *ECCV*, 2002, pp. 823–836.

[13] F. Liu and M. Gleicher, “Texture-consistent shadow removal,” in *ECCV*, 2008, pp. 437–450.

[14] G. D. Finlayson, M. S. Drew, and C. Lu, “Entropy minimization for shadow removal,” *International Journal of Computer Vision*, vol. 85, no. 1, pp. 35–57, 2009.

[15] T.-P. Wu, C.-K. Tang, M. S. Brown, and H.-Y. Shum, "Natural shadow matting," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 2, p. 8, 2007.

[16] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri, "Automatic shadow detection and removal from a single image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 431–446, 2016.

[17] X. Huang, G. Hua, J. Tumblin, and L. Williams, "What characterizes a shadow boundary under the sun and sky?" in *ICCV*, 2011, pp. 898–905.

[18] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, "Detecting ground shadows in outdoor consumer photographs," in *ECCV*, 2010, pp. 322–335.

[19] J. Zhu, K. G. Samuel, S. Z. Masood, and M. F. Tappen, "Learning to recognize shadows in monochromatic natural images," in *CVPR*, 2010, pp. 223–230.

[20] R. Guo, Q. Dai, and D. Hoiem, "Paired regions for shadow detection and removal," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2956–2967, 2013.

[21] M. Gryka, M. Terry, and G. J. Brostow, "Learning to remove soft shadows," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 5, p. 153, 2015.

[22] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri, "Automatic feature learning for robust shadow detection," in *CVPR*, 2014, pp. 1939–1946.

[23] T. F. Y. Vicente, L. Hou, C.-P. Yu, M. Hoai, and D. Samaras, "Large-scale training of shadow detectors with noisily-annotated shadow examples," in *ECCV*, 2016, pp. 816–832.

[24] V. Nguyen, T. F. Y. Vicente, M. Zhao, M. Hoai, and D. Samaras, "Shadow detection with conditional generative adversarial networks," in *ICCV*, 2017, pp. 4510–4518.

[25] L. Qu, J. Tian, S. He, Y. Tang, and R. W. Lau, "DeshadowNet: A multi-context embedding deep network for shadow removal," in *CVPR*, 2017, pp. 4067–4075.

[26] J. Wang, X. Li, and J. Yang, "Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal," in *CVPR*, 2018, to appear.

[27] R. Guo, Q. Dai, and D. Hoiem, "Single-image shadow detection and removal using paired regions," in *CVPR*, 2011, pp. 2033–2040.

[28] Y. Vicente, F. Tomas, M. Hoai, and D. Samaras, "Leave-one-out kernel optimization for shadow detection," in *ICCV*, 2015, pp. 3388–3396.

[29] S. Xie and Z. Tu, "Holistically-nested edge detection," in *ICCV*, 2015, pp. 1395–1403.

[30] L. Shen, T. Wee Chua, and K. Leman, "Shadow optimization from structured deep edge detection," in *CVPR*, 2015, pp. 2067–2074.

[31] S. Hosseinzadeh, M. Shakeri, and H. Zhang, "Fast shadow detection from a single image using a patched convolutional neural network," *arXiv preprint arXiv:1709.09283*, 2017.

[32] H. Gong and D. P. Cosker, "Interactive shadow removal and ground truth for variable scene categories," in *BMVC*, 2014, pp. 1–11.

[33] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[34] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *CVPR*, 2016, pp. 2874–2883.

[35] Q. V. Le, N. Jaitly, and G. E. Hinton, "A simple way to initialize recurrent networks of rectified linear units," *arXiv preprint arXiv:1504.00941*, 2015.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[38] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *CVPR*, 2016, pp. 761–769.

[39] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[40] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *NIPS*, 2011, pp. 109–117.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[42] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu, "A stagewise refinement model for detecting salient objects in images," in *ICCV*, 2017, pp. 4019–4028.

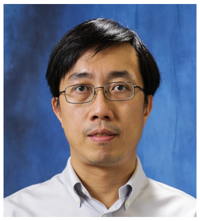
[43] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *ICCV*, 2017, pp. 202–211.

[44] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017, pp. 2881–2890.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[46] Q. Yang, K.-H. Tan, and N. Ahuja, "Shadow removal using bilateral filtering," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4361–4368, 2012.

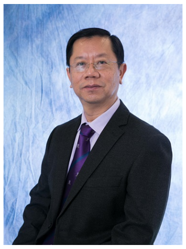
Xiaowei Hu received the B.Eng. degree in the Computer Science and Technology from South China University of Technology, China, in 2016. He is currently working toward the Ph.D. degree with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. His research interests include computer vision and deep learning.



Chi-Wing Fu joined the Chinese University of Hong Kong as an associate professor from 2016. He obtained his PhD in Computer Science from Indiana University Bloomington, USA. He served as the program co-chair of SIGGRAPH ASIA 2016 technical brief and poster, associate editor of Computer Graphics Forum, and program committee members in various conferences including IEEE Visualization. His research interests include computer graphics, visualization, and user interaction.

Lei Zhu received his Ph.D. degree in the Department of Computer Science and Engineering from the Chinese University of Hong Kong in 2017. He is working as a postdoctoral fellow in the Hong Kong Polytechnic University and a honorary postdoctoral fellow in the Chinese University of Hong Kong. His research interests include computer graphics, computer vision, and medical image processing.

Jing Qin received his Ph.D. degree in Computer Science and Engineering from the Chinese University of Hong Kong in 2009. He is currently an assistant professor in School of Nursing, The Hong Kong Polytechnic University. He is also a key member in the Centre for Smart Health, SN, PolyU, HK. His research interests include innovations for healthcare and medicine applications, medical image processing, deep learning, visualization and human-computer interaction and health informatics.



Pheng-Ann Heng received his B.Sc. from the National University of Singapore in 1985. He received his MSc (Comp. Science), M. Art (Applied Math) and Ph. D (Comp. Science) all from the Indiana University of USA in 1987, 1988, 1992 respectively. He is a professor at the Department of Computer Science and Engineering at The Chinese University of Hong Kong (CUHK). He has served as the Director of Virtual Reality, Visualization and Imaging Research Center at CUHK since 1999 and as the Director of Center for Human-Computer Interaction at Shenzhen Institute of Advanced Integration Technology, Chinese Academy of Science/CUHK since 2006. He has been appointed as a visiting professor at the Institute of Computing Technology, Chinese Academy of Sciences as well as a Cheung Kong Scholar Chair Professor by Ministry of Education and University of Electronic Science and Technology of China since 2007. His research interests include AI and VR for medical applications, surgical simulation, visualization, graphics and human-computer interaction.