

# DeRenderNet: Intrinsic Image Decomposition of Urban Scenes with Shape-(In)dependent Shading Rendering

Yongjie Zhu, Jiajun Tang, Si Li, and Boxin Shi, *Senior Member, IEEE*

**Abstract**—We propose DeRenderNet, a deep neural network to decompose the albedo and latent lighting, and render shape-(in)dependent shadings, given a single image of an outdoor urban scene, trained in a self-supervised manner. To achieve this goal, we propose to use the albedo maps extracted from scenes in videogames as direct supervision and pre-compute the normal and shadow prior maps based on the depth maps provided as indirect supervision. Compared with state-of-the-art intrinsic image decomposition methods, DeRenderNet produces shadow-free albedo maps with clean details and an accurate prediction of shadows in the shape-independent shading, which is shown to be effective in re-rendering and improving the accuracy of high-level vision tasks for urban scenes.

**Index Terms**—Intrinsic image decomposition, Inverse rendering, Albedo, Shading, Shadow

---

◆

## 1 INTRODUCTION

INTRINSIC image decomposition aims to decompose an input image into its intrinsic components that separate the material properties of observed objects from illumination effects [1]. It has been studied extensively that such decomposition results could be beneficial to many computer vision tasks, such as segmentation [2], recognition [3], 3D object compositing [4], relighting [5], *etc.* However, decomposing intrinsic components from a single image is a highly ill-posed problem, since the appearance of an object is jointly determined by various factors. An unconstrained decomposition without one or more terms of shape, reflectance, and lighting being fixed, will produce infinitely many solutions.

To address the shape-reflectance-lighting ambiguity, classic intrinsic decomposition methods adopt a simplified formulation by assuming an ideally diffuse reflectance for observed scenes. These methods decompose an observed image as the pixel-wise product of a reflectance map and a greyscale shading image, leaving the lighting color effects on the reflectance image [6], [7], [8], [9]. By learning the reflectance consistency of a time-lapse sequence, the illumination color can be correctly merged into the shading map [10]. However, it cannot be extracted from the shading without shape information, so the results of decomposition cannot be re-rendered into a new image. Re-rendering using the decomposed intrinsic components can be achieved by choosing objects with simple shapes [4], [5], [11]. In more recent research, attention has been put on decomposing shape, reflectance, and lighting on a complete scene [12], [13], by

still assuming Lambertian reflectance (represented by an albedo map). It becomes rather challenging to consider more realistic reflections such as specular highlights, shadows, and interreflection for single image intrinsic decomposition at the scene context, because the diverse objects and their complex interactions significantly increase the ambiguity of the solution space. Recently, a complex self-supervised outdoor scene decomposition framework [14] has been proposed to further separate shadows from the shading image with a multi-view training dataset, and relighting a city scene has been achieved by using a neural renderer with learned shape and lighting descriptors [15].

Lacking an appropriate dataset is the main obstacle that prevents a deep neural network from learning intrinsic decomposition at the scene level effectively. Existing datasets are either sparsely annotated (*e.g.*, IIW [6] and SAW [16]) or partially photorealistic with noisy appearance (*e.g.*, SUNCG [17], CGIntrinsics [8], PBRs [18]). We notice that the recently released FSVG dataset [19] created from videogames<sup>1</sup> complements well existing intrinsic image decomposition datasets in several aspects: 1) it contains dense annotations of albedo and depth maps (from which shape information can be acquired); 2) the renderings are of high quality with complex reflection effects and little noise; and 3) it covers unprecedentedly abundant outdoor urban scenes on a city scale. However, it is non-trivial to directly apply this data to the task of intrinsic image decomposition. The dataset is initially collected for high-level vision tasks such as segmentation and recognition,

- 
- Y. Zhu and S. Li are with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, China.  
E-mail: {zhuyongjie, lisi}@bupt.edu.cn
  - J. Tang and B. Shi (corresponding author) are with the National Engineering Laboratory for Video Technology, Department of Computer Science and Technology, Peking University, China. B. Shi is also affiliated with the Institute for Artificial Intelligence, Peking University, China.  
E-mail: {jiajun.tang, shiboxin}@pku.edu.cn

1. We want to comment that the computer graphics (CG) rendering and physics-based image formation (IF) are different and approximation is widely applied in graphics for efficiency consideration. Therefore, we mainly focus on the components which CG and IF render in the same way, such as diffuse shading and shadow. For components that CG does not follow IF (*e.g.*, specular component and interreflection), we will not discuss them in this work.

therefore only the albedo is directly usable for supervising decomposing intrinsic components.

In this paper, we propose **DeRenderNet** to learn intrinsic image decomposition from the outdoor urban scenes based on FSVG dataset [19]. As the name suggests, we “derender” the scene by decomposing the scene intrinsic components to obtain albedo, lighting, and shape-independent shading (which mainly contains cast shadows), and then **rendering** shape-dependent shading (dot product of normal and lighting) using a self-supervised neural network. Our major contributions include:

- We propose an image formation model (Section 3) and data preprocessing pipeline (Section 4) that takes full advantage of FSVG dataset [19] to conduct intrinsic image decomposition for outdoor urban scenes.
- We design a two-stage network (Section 5) to effectively extract four types of intrinsic components, including albedo, lighting in its latent space, shape-independent shading, and re-rendered shape-dependent shading with the guidance of the depth map by self-supervised learning.
- We demonstrate that our method avoids over-smoothing the albedo map by correctly decomposing shape-(in)dependent shadings (Section 6). The recovered shadow-free albedo with clean details could benefit high-level vision tasks for urban scenes (Section 7).

## 2 RELATED WORK

Recently, learning-based methods have shown advantages over conventional methods relying on hand-crafted priors. In this section, we discuss current learning-based methods for intrinsic image decomposition and, more broadly, those for inverse rendering.

### 2.1 Intrinsic Image Decomposition

The traditional intrinsic image decomposition refers to decompose an image into reflectance and shading. However, even for such a two-layer decomposition, ground truth data covering diverse scenes are difficult to collect. There are only a few datasets for this problem, such as single-object-based MIT intrinsic dataset [20], animation-movie-based MPI-Sintel dataset [21], sparse manually labeled IIW [6] and SAW datasets [16], OpenGL-rendered SUNCG dataset [17], and physically rendered CGIntrinsics dataset [8]. Among them, MIT, IIW, and SAW contain real captured images, MPI-Sintel, SUNCG, and CGIntrinsics are synthetic datasets. To date, the majority of intrinsic methods [7], [9], [22] used the IIW dataset [6] for training and validation. Narihira *et al.* [22] extracted deep features from two patches and trained a classifier to determine the pairwise lightness ordering. Zhou *et al.* [7] proposed a data-driven method to predict lightness ordering and integrated it into energy functions. Nestmeyer *et al.* [9] used signal processing techniques and a bilateral filter to get reasonable decomposition results. These methods rely on pairwise reflectance comparison to guide the network prediction due to the lack of densely labeled ground truth albedo. Li and Snavely [8] proposed a deep model to combine CGIntrinsics, IIW, and SAW datasets to

learn intrinsic decomposition with densely labeled synthetic data. Later, they [10] proposed an image sequence dataset with a fixed viewpoint to learn constant reflectance images over time and successfully put lighting color into shading images. Bonneel *et al.* [23] reviewed and evaluate some past intrinsic works for image editing. Fan *et al.* [24] revisited this problem and used a guiding network to finetune the albedo estimation results. Liu *et al.* [15] learned intrinsic decomposition for relighting city scenes on a time-lapse dataset collected from Google Street View.

In this paper, we also consider the colors in shading and further decompose shading into shape-dependent and shape-independent terms.

### 2.2 Inverse Rendering

The appearance of an observed image is jointly determined by shape, reflectance, and lighting. Previous works tried to take such components as input and train a neural network to estimate the result of forward rendering [25], or formulate it in a differentiable way [26]. Compared with forward rendering, inverse rendering is more widely used in computer vision applications, but it is also a difficult task. To alleviate the difficulty in joint intrinsic components estimation for general objects, single types of objects with relatively unified material property (such as human faces) have been studied. Sengupta *et al.* [5] assumed the human face is a Lambertian object and used supervised training on synthetic data, while later used real images for finetuning using self-supervised learning. Yamaguchi *et al.* [27] proposed a hybrid reflectance model that combined Lambertian and specular properties to get a more realistic face appearance. Meka *et al.* [4] used an encoder-decoder architecture to successfully estimate the material of single objects. Recently, research focus has turned to inverse rendering of a scene. For example, Yu *et al.* [12] used multi-view stereo (MVS) to get a rough geometry and then estimated a dense normal map by assuming a Lambertian model and albedo priors on outdoor scenes. Yu *et al.* [14] further estimated a global shadow map to get shadow-free albedo in order to relight outdoor scenes. Sengupta *et al.* [13] rendered indoor scenes with the Phong model and proposed a learning-based approach to jointly estimate albedo, normal, and lighting of an indoor image based on a synthetic dataset. However, reflectance on the scene level is much more complex than what a Lambertian or Phong model can describe. Li *et al.* [28] proposed a physically-based renderer with SVBRDF to render more realistic images to learn inverse rendering for complex indoor scenes.

In this paper, we extract intrinsic components to inverse-render a scene without explicitly estimating depth and specularity; instead we focus on extracting an independent shadow layer.

## 3 IMAGE FORMATION MODEL

The goal of DeRenderNet is to decompose an image in its intrinsic components for real-world outdoor scenes, containing complex shape, reflectance, and lighting (shadows). To improve the classic formulation which uses the albedo-scaled shading, we define the *shape-dependent shading* that

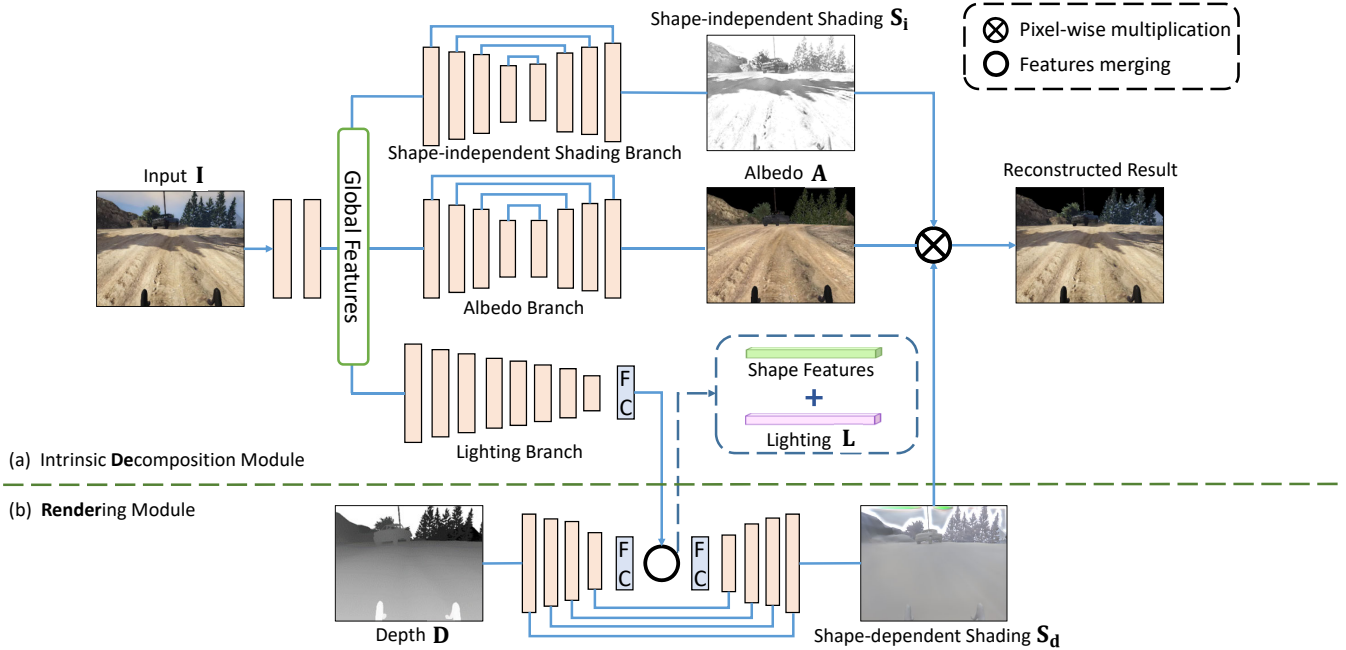


Fig. 1. The network architecture of DeRenderNet is composed of two modules: (a) Intrinsic Decomposition Module and (b) Rendering Module. Kernel sizes for all convolutional layers are  $5 \times 5$ . The intrinsic decomposition module takes an image  $I$  as input and estimates the intrinsic components: shape-independent shading  $S_i$ , albedo  $A$ , and latent lighting code  $L$ . The rendering module first takes the depth image  $D$  as input and extracts shape features in latent space, and then it is concatenated with  $L$  to decode the shape-dependent shading  $S_d$ . The estimated  $S_i$ ,  $A$ , and  $S_d$  can be used to generate a reconstructed image via pixel-wise multiplication.

is consistent with classic shading as the dot product between normal and lighting, and the *shape-independent shading*, which mainly contains cast shadows (which are not determined by the shape of an object itself but by the occlusion on the light path). We ignore the specular highlights and other global illumination effects (such as inter-reflection, transparency and translucency), since they are too complex in a scene with many objects. Fortunately, they are generally sparse, so we leave them as additive noise. Then the image formation can be expressed as

$$I = A \odot S_d(D, L) \odot S_i + \epsilon, \quad (1)$$

where albedo map  $A$  and shape-dependent shading  $S_d$  have the same physical meaning as other intrinsic decomposition works [8], [12]. To link our definition to diffuse shading, which is defined as  $\tilde{S}_d(\mathbf{n}, \Omega) = \sum_k \max(\mathbf{n}^T \omega_k, 0)$  ( $\omega_k \in \Omega$  is a 3-D vector encoding directional light intensity and direction sampled from the visible hemisphere of an environment map  $\Omega$ ), we design a rendering module  $\Theta$  that takes depth  $D$  and lighting  $L$  in a latent space as input to render  $S_d$  instead of writing down its explicit form (usually not differentiable), *i.e.*,  $\tilde{S}_d(\mathbf{n}, \Omega) = \tilde{S}_d(\Theta(D, L)) = S_d(D, L)$ .  $S_d$  does not take any global lighting effect such as shadows or interreflections into account, while shape-independent shading  $S_i$ , as the complement of  $S_d$ , is intended to encode the contribution of cast shadows and interreflections.

Given a scene image  $I$  as input, we decompose all intrinsic components  $A$ ,  $S_d$ ,  $S_i$ , with  $D$  and  $L$  as latent variables in Equation (1), as illustrated in Figure 1. We do not explicitly estimate the shape as one of the intrinsic components because inferring the scene depth from a single

image is another highly ill-posed problem [29], [30]. Due to the strong shape-light ambiguity in scenes, estimating the shape information constrained by reconstruction errors [12], [13] is fragile.

## 4 TRAINING DATA

To achieve shape-(in)dependent decomposition on shading, we need densely labeled depth maps in addition to albedo. We therefore propose a data preprocessing method to obtain additional supervision for our task.

### 4.1 FSVG Dataset

The FSVG dataset [19] is created by collecting and labeling an enormous amount of data (about 220k images) from videogames such as GTAV and The Witcher 3. It includes instance and semantic segmentation labels, densely labeled depth, optical flow, and pixel-wise albedo values. This dataset is useful in many high-level vision tasks thanks to both realistically rendered images and accurate ground truth labels. By collecting data from videogames, a large number of images in outdoor scenes with great diversity can be acquired, which is highly difficult and impractical using human labor. In this paper, we use densely labeled ground truth values of albedo and depth for directly and indirectly supervising intrinsic image decomposition task.

### 4.2 Data Preprocessing

To provide more useful supervision for our task, we pre-compute two types of additional labels: a shadow prior map and a normal map, as shown in Figure 2.

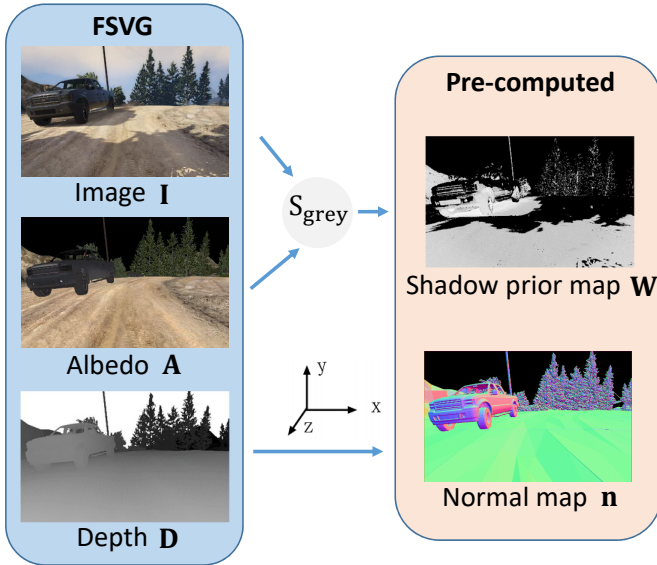


Fig. 2. Our training data contains two parts: the image, albedo, and depth directly available from FSVG dataset [19] (left column), and the shadow prior map and normal map (right column) pre-computed from the data on the left.

The shadow prior map provides an approximate probability distribution of shadow regions. To obtain the shadow prior map, we first calculate the pseudo-shading as  $S_{\text{pseudo}} = \mathbf{I}/\mathbf{A}$ . Then we set the regions where  $S_{\text{pseudo}} > 1$  to a large intensity value (10 in our experiment) to eliminate the effects of non-shadow parts and convert  $S_{\text{pseudo}}$  into a greyscale image  $S_{\text{grey}}$ . We define the shadow prior map as

$$\mathbf{W} = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{S_{\text{grey}}^2}{2}\right). \quad (2)$$

The shadow prior map alone is not sufficient to distinguish dark shadings and shadows, so we consider to involve depth for further decomposition. The depth map contains pixel-wise relative distance information, but it does not explicitly represent surface orientations, which encode important cues for intrinsic image decomposition. We therefore calculate the pixel-wise normal vector according to the depth map. We assume a perspective camera model, so the projection from 3D coordinates  $\mathbf{P} = (x, y, z)$  to 2D image coordinates  $\mathbf{p} = (u, v, 1)$  is given by

$$\mathbf{p} = \frac{1}{z} \mathbf{K} \mathbf{P}, \quad \mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

Accordingly the surface normal directions can be calculated as  $\bar{\mathbf{n}}^T = [-z_x(u, v), -z_y(u, v), 1]$ . We use unit-length normal vectors  $\mathbf{n} = \bar{\mathbf{n}}/\|\bar{\mathbf{n}}\|$ .

## 5 PROPOSED METHOD

In this section, we show how to solve the intrinsic image decomposition defined in Equation (1) by fully exploiting the available and pre-computed data, instead of relying on smoothness terms on albedo and shading [6], [8], [9], [24].

### 5.1 Network Structure Design

We have the following considerations: 1) we want to estimate as many as possible intrinsic components according to the data available; 2) we want to use depth to guide our intrinsic image decomposition but avoid using it as a necessary input; 3) we want to use learned shape features and lighting in latent space to overcome the limited representation power of parametric image formation models.

Based on the considerations above, we create a two-stage framework called **DeRenderNet**. The first stage of DeRenderNet is an intrinsic decomposition module that takes an observed image  $\mathbf{I}$  as input and estimates three intrinsic components: albedo  $\mathbf{A}$ , lighting  $\mathbf{L}$  in latent space, and shape-independent shading  $\mathbf{S}_i$ . The second stage of DeRenderNet is a rendering module, which generates shape-dependent shading  $\mathbf{S}_d$  based on lighting code  $\mathbf{L}$  extracted by the previous module and depth map  $\mathbf{D}$ .

#### 5.1.1 Intrinsic Decomposition Module

The intrinsic decomposition module is a single-input-multi-output network that consists of three branches, as shown in Figure 1 (a). The first two layers of the network are used to extract global features from an input image  $\mathbf{I}$ , and then two residual block based architectures are used to predict  $\mathbf{A}$  and  $\mathbf{S}_i$ , respectively. The skip connections in residual blocks allow the high-frequency information to be merged with low-frequency features. Since  $\mathbf{L}$  is a latent code, we use a deep encoder followed by a fully connected layer to predict it. We find that such an architecture can help separating  $\mathbf{S}_i$  and  $\mathbf{S}_d$ , because  $\mathbf{S}_i$  has stronger gradients on edges. These features can pass through convolutional layers but are not preserved after traversing the fully connected layer. In this way, the intrinsic decomposition module can partly separate  $\mathbf{S}_i$  and  $\mathbf{S}_d$  features.

#### 5.1.2 Rendering Module

The rendering module is a multi-input-single-output network. It takes depth  $\mathbf{D}$  and lighting code  $\mathbf{L}$  as input, and outputs  $\mathbf{S}_d$ . We treat this network as a “rendering engine” that is able to generate realistic shading for complex scenes. We design it to handle shape-related complex reflection in a self-learning way. The main structure is an encoder-decoder, as shown in Figure 1 (b). The encoder part is used to encode a shape prior and to extract features of shape information. Shape features obtained by the last layer of the encoder are concatenated with lighting code  $\mathbf{L}$  before being sent into the decoder part to generate  $\mathbf{S}_d$ . We use a fully connected layer at the end of the encoder and the beginning of the decoder to extract high-frequency features and merge these two types of features together. Several skip connections are used to make full use of shape information.

### 5.2 Learning Shape-(in)dependent Shadings

DeRenderNet is trained with the labeled and pre-computed data in Figure 2. We use ground truth albedo labels and dense depth labels from the FSVG dataset [19] to supervise the albedo estimation and guide the self-supervised learning, respectively. Since the ground truth lighting, shading, and shadows are not available in current datasets, we propose a self-supervised learning method to estimate

lighting code  $\mathbf{L}$ , shape-dependent shading  $\mathbf{S}_d$ , and shape-independent shading  $\mathbf{S}_i$  by designing appropriate loss functions.

### 5.2.1 Supervised Loss

Since the images in the FSVG dataset [19] are equipped with ground truth albedo values, we use the albedo as direct supervision to train our intrinsic decomposition module:

$$\mathcal{L}_a = \frac{1}{M} \sum_{i=1}^M (\mathbf{A}_i - \mathbf{A}_i^*)^2, \quad (4)$$

where  $\mathbf{A}$  is the estimation of albedo image,  $\mathbf{A}^*$  is the ground truth value of albedo,  $M$  is the total number of valid pixels correspond to non-sky regions.

### 5.2.2 Self-supervised Losses

Although  $\mathbf{A}^*$  provides direct supervision for albedo estimation, there is no supervision guidance for the other two intrinsic attributes:  $\mathbf{L}$  and  $\mathbf{S}_i$ . Hence we propose a self-supervised method to make our intrinsic decomposition module learn how to effectively extract these two components and to make our rendering module learn how to render  $\mathbf{S}_d$  based on the estimated  $\mathbf{L}$  and the guiding depth map  $\mathbf{D}$  at the same time.

**Shape-independent Shading Loss:** We use the pre-computed shadow prior map, which is an approximate probability distribution indicating where shadows should appear, to guide the prediction of shape-independent shading  $\mathbf{S}_i$  using the following loss function:

$$\mathcal{L}_{si} = \frac{1}{M} \sum_{i=1}^M w_i \sum_{j \in \mathcal{N}(i)} (\mathbf{S}_{i_i} - \mathbf{S}_{i_j})^2. \quad (5)$$

Here  $w_i \in \mathbf{W}$  is the shadow prior at pixel  $i$ ,  $\mathcal{N}(i)$  denotes the neighborhood of the pixel at position  $i$ , and  $M$  is the same as in Equation (4). This loss function mainly encourages shape-independent shading  $\mathbf{S}_i$  to be considered as shadows, and we will further decouple it from the dependency on shape by the next loss function.

**Shape-dependent Shading Loss:** After putting high-frequency terms as an additive noise in  $\epsilon$ , it is natural to assume that the shading is smooth. More specifically, considering that the outdoor light source mainly comes from the distant sun, albedo-normalized scene radiance values on the same shadow-free plane should be roughly consistent. On the other hand, though using  $\mathbf{D}$  and  $\mathbf{L}$  significantly suppresses the edges of shadows, there still exists ambiguity in non-edge parts. To further remove this ambiguity, we define the shape-dependent shading loss as

$$\mathcal{L}_{sd} = \frac{1}{M} \sum_{i=1}^M \left[ (\mathbf{S}_{d_i}(\mathbf{D}, \mathbf{L}) - \sum_{m=1}^B \mathbf{n}_i^\top \mathbf{l}_m)^2 + (1 - w_i)(\mathbf{A}_i \mathbf{S}_{d_i}(\mathbf{D}, \mathbf{L}) - \mathbf{I}_i)^2 \right], \quad (6)$$

where  $\mathbf{n}_i$  is the normal vector at pixel  $i$  of image,  $\mathbf{l}_m$  is a directional light (a 3-D vector) extracted from the latent code  $\mathbf{L}$  (in total  $B$  vectors are extracted) to encourage similar surface normals to have similar “shading” under  $\mathbf{L}$ ,  $w_i$  is the same as in Equation (5), and  $M$  is the same as in Equation (4). In particular, we split  $\mathbf{L}$  as a total of  $B$  4-D

vectors and take the first three elements of each 4-D vector as 3-D vector  $\mathbf{l}_m$ . The first term in this loss aims to smooth the shape-dependent shading approximately in a planar surface, and the second term is used to suppress shadows in the rendering process of shape-dependent shadings.

**Reconstruction Loss:** Given an estimated albedo  $\mathbf{A}$ , a shape-dependent shading  $\mathbf{S}_d$  and a shape-independent shading  $\mathbf{S}_i$ , we compute a reconstructed image using Equation (1). Then we use a pixel-wise  $L^2$  loss between the reconstructed image and the input image as the reconstruction loss:

$$\mathcal{L}_{recon} = \frac{1}{M} \sum_{i=1}^M (\mathbf{A}_i \mathbf{S}_{d_i}(\mathbf{D}, \mathbf{L}) \mathbf{S}_{i_i} - \mathbf{I}_i)^2. \quad (7)$$

## 5.3 Implementation Details

We train our DeRenderNet to minimize

$$\mathcal{L} = \mathcal{L}_{recon} + \lambda_1 \mathcal{L}_a + \lambda_2 \mathcal{L}_{sd} + \lambda_3 \mathcal{L}_{si}, \quad (8)$$

where  $\lambda_1 = 0.8$  and  $\lambda_2, \lambda_3 = 0.5$ . We use all 220k training images from the FSVG dataset [19] to train our networks. Before training, the images are resized to  $320 \times 240$  due to the fully connected layers in DeRenderNet. For the middle layers, we use ReLU and Batch Normalization followed by convolutional layers. We use the same learning rate for all layers. According to previous lighting estimation studies [31], [32], the scene lighting information encoded in a latent space has a strong representation power, so we adopt the similar strategy. To figure out a proper number of dimensions of lighting code, experiments are conducted when the number of dimensions are searched from 0 to 2000. We empirically set the dimension of the latent lighting code to 1024 and  $B$  in Equation (6) as 256 based on the results from validation set.

Please note that we only use depth as input in the training stage to guide DeRenderNet to learn how to extract  $\mathbf{S}_d$  and  $\mathbf{S}_i$  from images. To analyze the influence of depth errors during training, we added some noise to the depth input. However, we found that this will not affect our results, probably because the useful features could be extracted by the network despite noisy input depth. Once the training is completed, we can use the intrinsic decomposition module to get  $\mathbf{S}_i$  and  $\mathbf{A}$  without depth input. Please refer to our supplementary material to see the shape-dependent shading results with the estimated depth from MegaDepth [29] trained on in-the-wild data.

## 6 EXPERIMENTAL RESULTS

For thoroughly testing the performance of DeRenderNet, we perform quantitative evaluations on both the FSVG dataset [19] and the IIW benchmark [6] where we use  $\mathbf{S}_d \odot \mathbf{S}_i$  as our “overall” shading in order to compare with other methods. Although we do not have access to the ground truth values of real-world scenes to finetune our model, we still evaluate the generalization capability of our model on the real-world KITTI dataset [33]. Finally, we do ablation studies to show the role of the shape-independent shading loss and the shape-dependent shading loss.

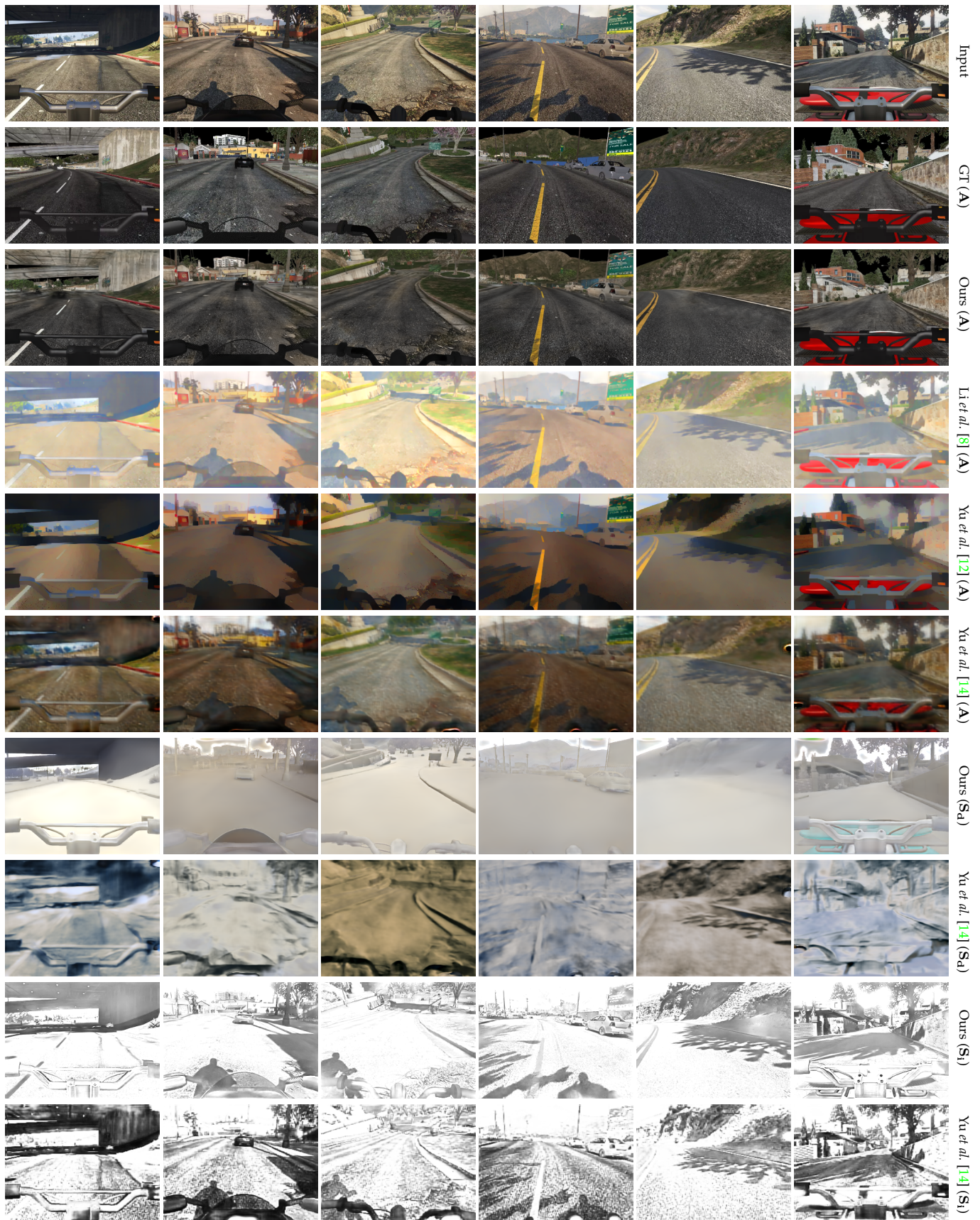


Fig. 3. Qualitative evaluation on the FSVG dataset [19]. We compare our albedo results against a synthetic dataset trained method [8] and multi-view dataset trained methods [12], [14]. The shape-(in)dependent shading estimates are further compared with [14].



Fig. 4. Qualitative results on the FSVG dataset [19]. We retrain the work by Yu *et al.* [12] on the FSVG dataset [19] using their official code as our baseline.

TABLE 1

Quantitative evaluation on the FSVG dataset [19]. We use  $S_d \odot S_i$  as our “overall” shading to compare with other methods.  $\downarrow$  means lower is better.

Methods	MSE $\downarrow$			LMSE $\downarrow$			DSSIM $\downarrow$		
	Albedo	Shading	Avg.	Albedo	Shading	Avg.	Albedo	Shading	Avg.
Li <i>et al.</i> [8]	0.0295	0.0416	0.0356	0.0124	0.0136	0.0130	0.2993	0.2123	0.2558
Fan <i>et al.</i> [24]	0.0283	0.0426	0.0354	0.0124	0.0144	0.0134	0.3077	0.2416	0.2746
Li <i>et al.</i> [10]	0.0276	0.0326	0.0301	0.0111	<b>0.0117</b>	0.0114	0.2908	<b>0.2007</b>	0.2458
Yu <i>et al.</i> [12]	0.0211	0.1670	0.0940	0.0098	0.0863	0.0481	0.2293	0.3362	0.2807
Yu <i>et al.</i> [14]	0.0382	0.1443	0.0912	0.0140	0.0923	0.0532	0.3333	0.3294	0.3314
IRN [12]@FSVG	0.0196	0.0305	0.0251	0.0073	0.0156	0.0114	0.2615	0.2056	0.2336
Ours	<b>0.0074</b>	<b>0.0271</b>	<b>0.0172</b>	<b>0.0046</b>	0.0140	<b>0.0093</b>	<b>0.1224</b>	0.2036	<b>0.1630</b>

TABLE 2

Quantitative evaluation on the IIW dataset [6]. We use  $S_d \odot S_i$  as our “overall” shading to compare with other methods.  $\downarrow$  means lower is better.

Methods	Training data	WHDR $\downarrow$
Bell <i>et al.</i> [6]	-	21.0
Zhou <i>et al.</i> [7]	IIW	19.9
Nestmeyer <i>et al.</i> [9]	IIW	19.5
Fan <i>et al.</i> [24]	IIW	14.5
Shi <i>et al.</i> [34]	ShapeNet	59.4
Narihira <i>et al.</i> [35]	Sinetl+MIT	37.3
Li <i>et al.</i> [10]	BigTime	20.3
Yu <i>et al.</i> [12]	MegaDepth	21.4
Yu <i>et al.</i> [14]	MegaDepth	24.9
Ours	FSVG+MegaDepth	21.0

## 6.1 Evaluation on FSVG

We quantitatively evaluate the intrinsic image decomposition using rendered outdoor urban scenes from the test split of the FSVG dataset [19] (50k images from 303 scenes). We compare with state-of-the-art intrinsic image decomposition methods [8], [10], [24] and inverse rendering methods [12], [14], with quantitative results summarized in Table 1. For fully evaluating albedo estimation, we use three metrics: MSE, LMSE, and DSSIM, which are commonly used [36].

Our method performs best in general, but the shading estimation performance is not the best because we leave some specular parts and local lights as additive noise. We further show examples of our estimated albedo  $A$  in comparison against [8], [12], and [14], and our estimated shape-dependent  $S_d$  and independent shading  $S_i$  in comparison with [14] in Figure 3. Our method not only recovers more details in  $A$  (e.g., textures of the road are not over-smoothed in our result), but also successfully removes shadows (e.g., cast by the motorcycle riders and trees) from  $A$  to  $S_i$ . From Figure 3, we can see that our method can generate a more reasonable light color than Yu *et al.*'s method [14]; by using the depth information to guide the network training, the cast shadows are naturally removed from  $S_d$  to  $S_i$ , as shown in our results (Row 7), while there are noticeable residues remaining in Yu *et al.*'s results [14] (Row 8). Comparing the cast shadow estimations of our method and Yu *et al.*'s [14] (Row 9 and 10), our method distinguishes more accurately dark albedo noise from shadows on the ground.

To further validate the contribution of our network architecture design, we retrain InverseRenderNet of Yu *et al.* [12] on the FSVG dataset [19] (denoted as IRN@FSVG) using their official code (some loss terms are not applicable). From Figure 4, we can see that the cast shadows are removed from our albedo, and our  $S_d$  shows the correct color of lighting. Quantitative results are shown in Table 1, where our results show a closer appearance to the ground truth than the baseline method.

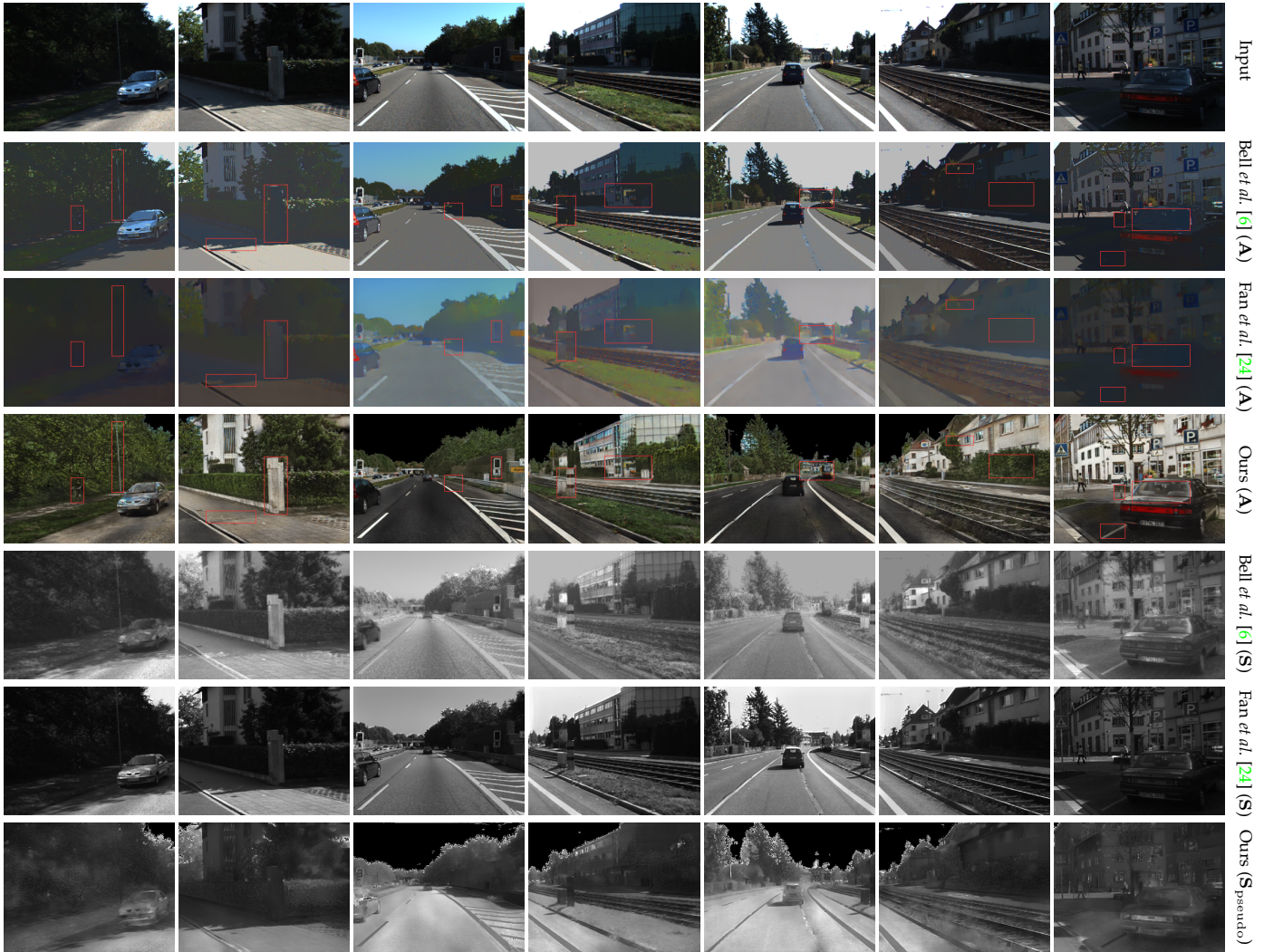


Fig. 5. Qualitative evaluation on the KITTI dataset [33]. We show  $S_{\text{pseudo}}$  in greyscale as our shading results for easier comparison with other methods. Please zoom-in on the electronic version to check the differences highlighted by red boxes.

## 6.2 Evaluation on KITTI

We use the KITTI dataset [33] which also contains outdoor urban scenes to test the generalization capability of DeRenderNet on real-world data, as shown in Figure 5. Compared with the traditional optimization-based method [6] and a learning-based method [24] that used cross-dataset supervised loss, our method shows more reasonable albedo estimates, by retaining richer texture details while discarding undesired shadows.

## 6.3 Evaluation on IIW

We also compare different methods on the popular IIW benchmark [6] for indoor intrinsic image decomposition. Among them, Yu *et al.* [12], [14] used the MegaDepth dataset [29], which has ground truth albedo computed by a multi-view inverse rendering algorithm [37] and depth to train their model. To narrow the gap between our training data and the IIW dataset [6], we use MegaDepth dataset [29] to finetune our model with weakly-supervised learning. We use InverseRenderNet [12] to get the dense normal map and use a pre-trained DeRenderNet on the FSVG dataset [19] to

get shape-independent shading. Then we finetune DeRenderNet with the same loss functions. We report the mean weighted human disagreement rate (WHDR) [6] in Table 2. Despite the domain gap, our method achieves comparable performance to other intrinsic decomposition methods.

## 6.4 Ablation study

To analyze the effects of our proposed shape-(in)dependent shading losses, we compare the performance of DeRenderNet by applying different shading losses in our training strategy (qualitative examples shown in Figure 6). The upper example in Figure 6 shows that our method can recover a more complete shadow appearance with  $\mathcal{L}_{\text{si}}$ . The lower example shows that our method correctly distinguishes shape-dependent (with shadow removed) and shape-independent shadings (with shadow reconstructed) with  $\mathcal{L}_{\text{sd}}$ .

## 7 APPLICATION

In this section, we use the task of cross rendering to validate the fidelity of our intrinsic decomposition results and show that our method could be beneficial to downstream high-level vision tasks taking object detection as an example.



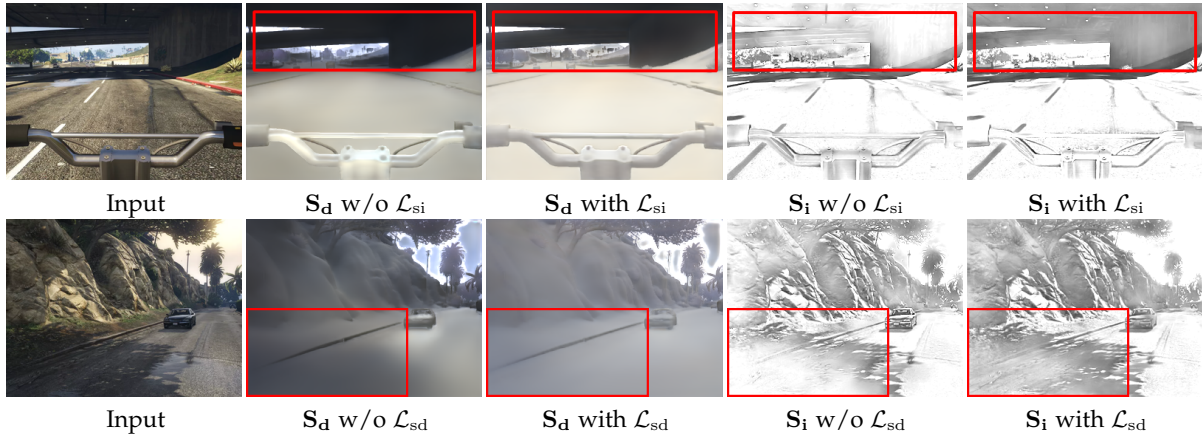


Fig. 6. Ablation study of shape-independent shading loss  $\mathcal{L}_{si}$  and shape-dependent shading loss  $\mathcal{L}_{sd}$ . Red boxes highlight noticeable differences.

### 7.1 Cross Rendering

We show our cross-rendering results with an example in Figure 7. The first column are the input images, and the second column are the decomposed shape-dependent shading from input images. The third column are generated shading by our rendering module with the guidance of depth map and exchanged lighting codes (the lighting is exchanged between rows) in the latent space. According to the shadows of inputs, we can see that the light sources come from the left direction in the top case and the right direction in the bottom case. We can infer from the relighted objects (such as shown in green boxes in Figure 7) that our rendering module can re-render reasonable shape-dependent shadings without shadows according to different lighting conditions.

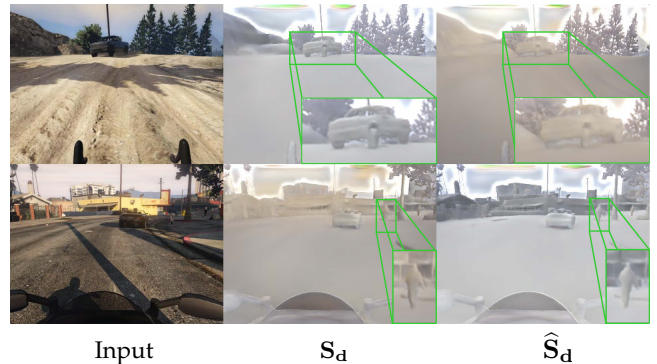


Fig. 7. Cross rendering shading results.  $S_d$  is the directly estimated shape-dependent shading, and  $\hat{S}_d$  is the cross rendered shading results where the lighting is exchanged between rows. The lighting effects of  $\hat{S}_d$  are correctly rendered. Green boxes show zoom-in details.

### 7.2 Application in high-level vision

Decomposed intrinsic components could potentially benefit high-level vision tasks such as object detection and segmentation by feeding the albedo map, which is less influenced by lighting and shadow, as their input. We show such an application example in Figure 8. We use a Mask R-CNN [38] pre-trained on the COCO dataset [39] as our test model<sup>2</sup>, and show its detection and segmentation results on the original image with albedo estimated by Li *et al.* [8], and with albedo estimated with our method. The bicycle, person, and car in the upper example and the truck and car in the lower example are detected with high confidence based on our albedo, but are incorrectly labeled in other cases.

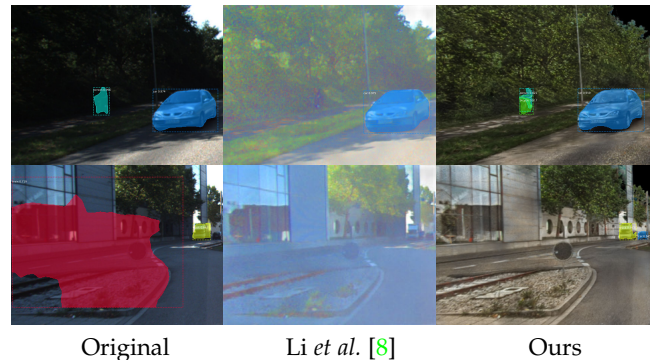


Fig. 8. An example where our intrinsic image decomposition results benefit object detection and segmentation. Please zoom-in for details.

## 8 CONCLUSION

We have proposed DeRenderNet to successfully decompose the albedo, shape-independent shading, and render shape-dependent shadings for outdoor urban scenes, with estimated latent lighting code and depth guidance. We demonstrate that with sufficient data, deep neural networks can learn how to decompose complex scenarios into their intrinsic components in a self-supervised manner. We hope our intrinsic image decomposition results can be helpful to other computer vision tasks, especially for urban scenes. For

example, by removing the cast shadows along the roadside and avoiding over-smoothing the albedo map, autonomous vehicles can detect the lanes on the road more reliably. Quantitative evaluation of the benefits brought by extracted intrinsic components to high-level vision problems for urban scenes could be an interesting future topic.

There are ways to extend this work to broader applications. For example, our rendering module could be combined with depth estimation methods to generate shape-dependent shading, and our intrinsic decomposition module can generate a pseudo map of albedo and shadow as

<sup>2</sup>. The code is available at <https://github.com/multimodallearning/pytorch-mask-rcnn>.

guidance for other tasks. There are still limitations to our methods. Our framework is trained only with albedo and depth supervision from the FSVG dataset [19], which is collected by exploiting CG rendering technology. Though useful for diffuse shape-(in)dependent shading decomposition, it is hard for CG to accurately model complex reflectance (e.g., specular component and interreflection) since approximations are widely applied for efficiency consideration.

In the future, we will investigate a more physically-reliable dataset with more comprehensive labels and better-designed network architecture to further handle intrinsic image decomposition in more challenging scenes.

## ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China under Grant No. 61872012, 62088102, National Key R&D Program of China (2019YFF0302902), and Beijing Academy of Artificial Intelligence (BAAI).

## REFERENCES

- [1] H. Barrow, J. Tenenbaum, A. Hanson, and E. Riseman, "Recovering intrinsic scene characteristics from images," *Computer Vision Systems*, pp. 3–26, 1978. **1**
- [2] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro, "Improving semantic segmentation via video propagation and label relaxation," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2019. **1**
- [3] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2019. **1**
- [4] A. Meka, M. Maximov, M. Zollhoefer, A. Chatterjee, H.-P. Seidel, C. Richardt, and C. Theobalt, "LIME: Live intrinsic material estimation," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2018. **1, 2**
- [5] S. Sengupta, A. Kanazawa, C. D. Castillo, and D. W. Jacobs, "SfSNet: Learning shape, reflectance and illuminance of faces in the wild," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2018. **1, 2**
- [6] S. Bell, K. Bala, and N. Snavely, "Intrinsic images in the wild," *Proc. of ACM SIGGRAPH*, vol. 33, no. 4, 2014. **1, 2, 4, 5, 7, 8**
- [7] T. Zhou, P. Krähenbühl, and A. A. Efros, "Learning data-driven reflectance priors for intrinsic image decomposition," in *Proc. of International Conference on Computer Vision (ICCV)*, 2015. **1, 2, 7**
- [8] Z. Li and N. Snavely, "CGIntrinsics: Better intrinsic image decomposition through physically-based rendering," in *Proc. of European Conference on Computer Vision (ECCV)*, 2018. **1, 2, 3, 4, 6, 7, 9**
- [9] T. Nestmeyer and P. V. Gehler, "Reflectance adaptive filtering improves intrinsic image estimation," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017. **1, 2, 4, 7**
- [10] Z. Li and N. Snavely, "Learning intrinsic image decomposition from watching the world," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2018. **1, 2, 7**
- [11] J. T. Barron and J. Malik, "Shape, illumination, and reflectance from shading," *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015. **1**
- [12] Y. Yu and W. A. Smith, "InverseRenderNet: Learning single image inverse rendering," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2019. **1, 2, 3, 6, 7, 8**
- [13] S. Sengupta, J. Gu, K. Kim, G. Liu, D. W. Jacobs, and J. Kautz, "Neural inverse rendering of an indoor scene from a single image," in *Proc. of International Conference on Computer Vision (ICCV)*, 2019. **1, 2, 3**
- [14] Y. Yu, A. Meka, M. Elgharib, H.-P. Seidel, C. Theobalt, and W. Smith, "Self-supervised outdoor scene relighting," in *Proc. of European Conference on Computer Vision (ECCV)*, 2020. **1, 2, 6, 7, 8, 11, 13**
- [15] A. Liu, S. Ginosar, T. Zhou, A. A. Efros, and N. Snavely, "Learning to factorize and relight a city," in *Proc. of European Conference on Computer Vision (ECCV)*, 2020. **1, 2**
- [16] B. Kovacs, S. Bell, N. Snavely, and K. Bala, "Shading annotations in the wild," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017. **1, 2**
- [17] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017. **1, 2**
- [18] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser, "Physically-based rendering for indoor scene understanding using convolutional neural networks," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017. **1**
- [19] P. Krhenbühl, "Free supervision from video games," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2018. **1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 14, 15**
- [20] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman, "Ground truth dataset and baseline evaluations for intrinsic image algorithms," in *Proc. of International Conference on Computer Vision (ICCV)*, 2009. **2**
- [21] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. of European Conference on Computer Vision (ECCV)*, 2012. **2**
- [22] T. Narihira, M. Maire, and S. X. Yu, "Learning lightness from human judgement on relative reflectance," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015. **2**
- [23] N. Bonneel, B. Kovacs, S. Paris, and K. Bala, "Intrinsic decompositions for image editing," in *Computer Graphics Forum*, vol. 36, no. 2, 2017, pp. 593–609. **2**
- [24] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf, "Revisiting deep intrinsic image decompositions," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2018. **2, 4, 7, 8**
- [25] O. Nalbach, E. Arabadzhiyska, D. Mehta, H.-P. Seidel, and T. Ritschel, "Deep shading: convolutional neural networks for screen space shading," in *Computer Graphics Forum*, vol. 36, no. 4, 2017, pp. 65–78. **2**
- [26] S. Zhao, W. Jakob, and T.-M. Li, "Physics-based differentiable rendering: from theory to implementation," in *ACM SIGGRAPH 2020 Courses*, 2020. **2**
- [27] S. Yamaguchi, S. Saito, K. Nagano, Y. Zhao, W. Chen, K. Olszewski, S. Morishima, and H. Li, "High-fidelity facial reflectance and geometry inference from an unconstrained image," *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*, vol. 37, no. 4, p. 162, 2018. **2**
- [28] Z. Li, M. Shafiei, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker, "Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and SVBRDF from a single image," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2020. **2**
- [29] Z. Li and N. Snavely, "MegaDepth: Learning single-view depth prediction from internet photos," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2018. **3, 5, 8, 11**
- [30] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, "GeoNet: Geometric neural network for joint depth and surface normal estimation," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2018. **3**
- [31] H. Weber, D. Prévost, and J.-F. Lalonde, "Learning to estimate indoor lighting from 3D objects," in *Proc. of International Conference on 3D Vision (3DV)*, 2018. **5**
- [32] Y. Hold-Geoffroy, A. Athawale, and J.-F. Lalonde, "Deep sky modeling for single image outdoor lighting estimation," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2019. **5**
- [33] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research*, 2013. **5, 8, 11, 13**
- [34] J. Shi, Y. Dong, H. Su, and S. X. Yu, "Learning non-lambertian object intrinsics across shapenet categories," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017. **7**
- [35] T. Narihira, M. Maire, and S. X. Yu, "Direct Intrinsics: Learning albedo-shading decomposition by convolutional regression," in *Proc. of International Conference on Computer Vision (ICCV)*, 2015. **7**
- [36] Q. Chen and V. Koltun, "A simple model for intrinsic image decomposition with depth cues," in *Proc. of International Conference on Computer Vision (ICCV)*, 2013. **7**
- [37] K. Kim, A. Torii, and M. Okutomi, "Multi-view inverse rendering under arbitrary illumination and albedo," in *Proc. of European Conference on Computer Vision (ECCV)*, 2016. **8**
- [38] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017. **9**
- [39] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in

context,” in *Proc. of European Conference on Computer Vision (ECCV)*, 2014. 9

[40] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. of International Conference on Learning Representations (ICLR)*, 2015. 11

[41] D. Anguelov, C. Dulong, D. Filip, C. Früh, S. Lafon, R. Lyon, A. S. Ogale, L. Vincent, and J. Weaver, “Google Street View: Capturing the world at street level,” *Computer*, vol. 43, no. 6, pp. 32–38, 2010. 11



**Yongjie Zhu** received the B.E. degree from Beijing University of Posts and Telecommunications in 2019. He is currently pursuing his Master’s degree at Beijing University of Posts and Telecommunications. His research interest is physical-based rendering in computer vision.



**Jiajun Tang** received the B.S. degree from Peking University in 2020. He is currently pursuing his Ph.D. degree at Peking University. His research interest is in 3D computer vision.



**Si Li** received the Ph.D. degree from Beijing University of Posts and Telecommunications in 2012. Now she is an Associate Professor at the School of Artificial Intelligence, Beijing University of Posts and Telecommunications. Her current research interests include computer vision and machine learning.



**Boxin Shi** received the B.E. degree from the Beijing University of Posts and Telecommunications, the M.E. degree from Peking University, and the Ph.D. degree from the University of Tokyo, in 2007, 2010, and 2013. He is currently a Boya Young Fellow Assistant Professor and Research Professor at Peking University, where he leads the Camera Intelligence Group. Before joining PKU, he did postdoctoral research with MIT Media Lab, Singapore University of Technology and Design, Nanyang Technological University

from 2013 to 2016, and worked as a researcher in the National Institute of Advanced Industrial Science and Technology from 2016 to 2017. He won the Best Paper Runner-up Award at International Conference on Computational Photography 2015. He has served as an editorial board member of *IJCV* and an area chair of *CVPR/ICCV*.

## I DATA PREPARATION

Due to the lack of high-quality intrinsic component labels for real scene data, we use FSVG dataset [19] to get training images with corresponding albedo and depth labels. The raw shape data are given as disparity images (3 channels), we first convert the disparity images into depth maps by:

$$\mathbf{D} = \frac{1}{\text{disp}[2] + \text{disp}[1] \times 256 + \text{disp}[0] \times 256^2}, \quad (9)$$

where  $\text{disp}[i]$  denotes the 8-bit integer in the  $i$ -th channel. Then we re-scale the depth by multiplying 100 and normalize it using the  $\log$  function for network input.

## II NETWORK ARCHITECTURE

In this section, we introduce the detailed network architectures of the DeRenderNet. Taking a single image as input, the intrinsic decomposition module consists of two convolutional layers for extracting global features, two residual blocks for  $\mathbf{A}$  and  $\mathbf{S}_i$  prediction (see Figure I(a)), and one encoder branch for  $\mathbf{L}$  prediction (see Figure I(b)). The rendering module has an encoder-decoder architecture (see Figure I(c)). Specifically, the encoder consists of eight convolutional layers followed by a fully connected layer to extract shape features, and the decoder consists of a fully connected layer to fuse shape features and latent light code and eight convolutional layers to predict  $\mathbf{S}_d$ . The detailed structures of the convolutional layer and deconvolutional layer we used are shown in Figure II.

## III IMPLEMENTATION DETAILS

Our framework is implemented in PyTorch 1.4.0, and Adam optimizer [40] is used with default parameters. We train DeRenderNet in an end-to-end manner using a batch size of 8 for 20 epochs until convergence on a GTX 1080Ti GPU. The learning rate is initially set to  $5 \times 10^{-4}$  and halved every 5 epochs. The training process takes roughly 36 hours to reach convergence.

## IV MORE QUALITATIVE RESULTS

In Figure III, we show additional results of our method and Yu *et al.* [14] (trained on a real-world dataset) on KITTI dataset [33] where Yu *et al.* [14] estimates albedo, direct shading, and shadow.

Here in Figure IV and Figure V, we show more qualitative results of DeRenderNet on FSVG dataset [19]. To further show the generalization capacity of DeRenderNet, we also show in Figure VI-VIII our decomposition results on images in the wild, which are collected from Google Street View [41]. We use the MegaDepth [29] pretrained model to estimate the depth maps from these street view images, and the estimated depth maps are re-scaled and transformed into  $\log$  space as in Section I before generating shape-dependent shading  $\mathbf{S}_d$ .

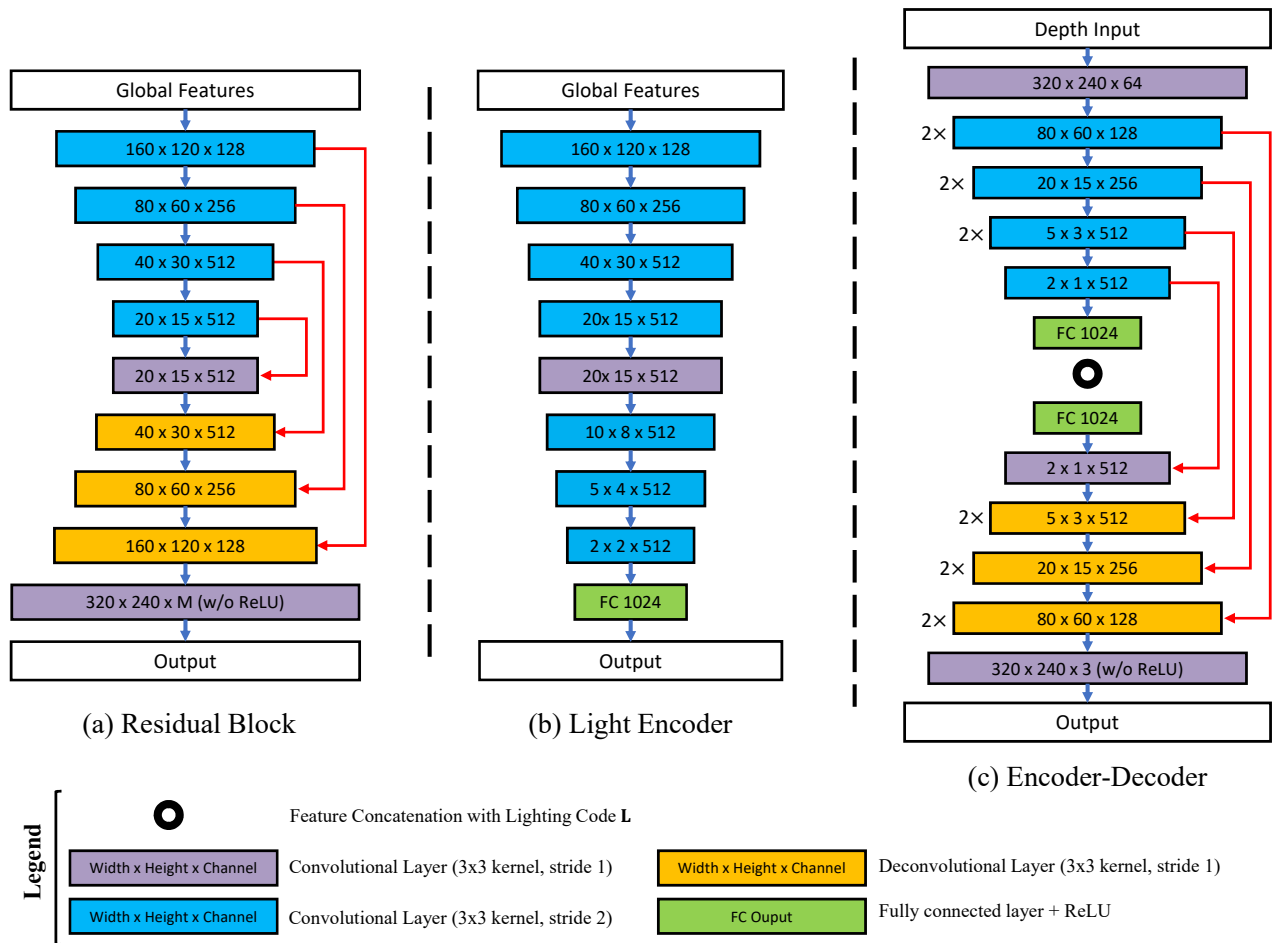


Fig. I. Network architecture.

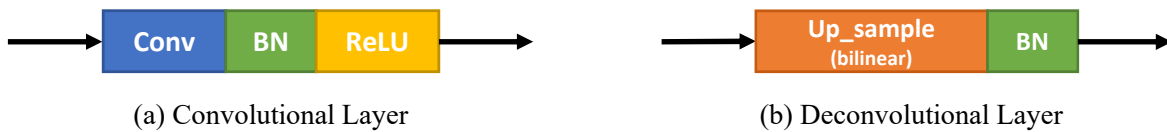


Fig. II. Detailed structures of the convolutional layer and the deconvolutional layer we used.

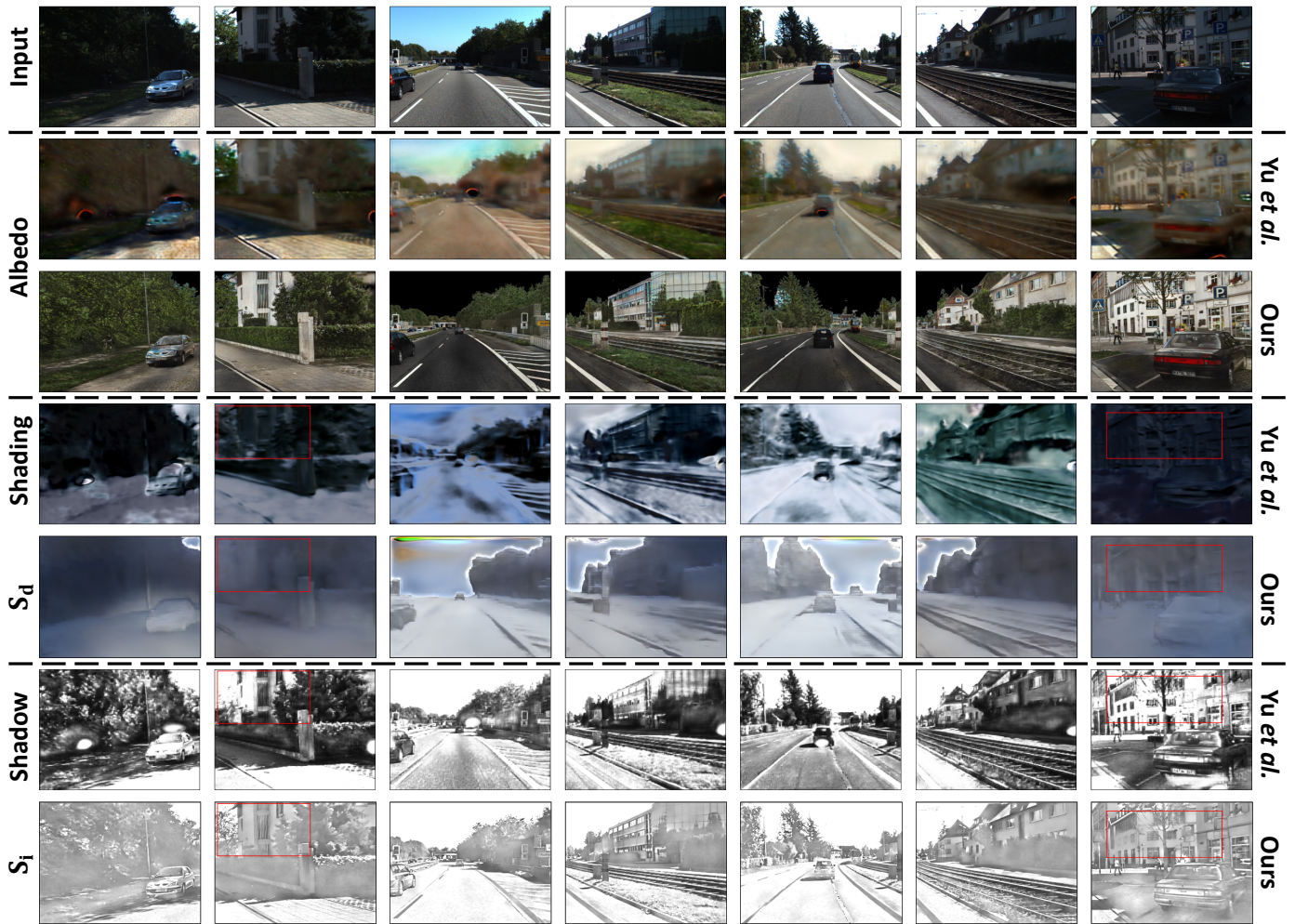
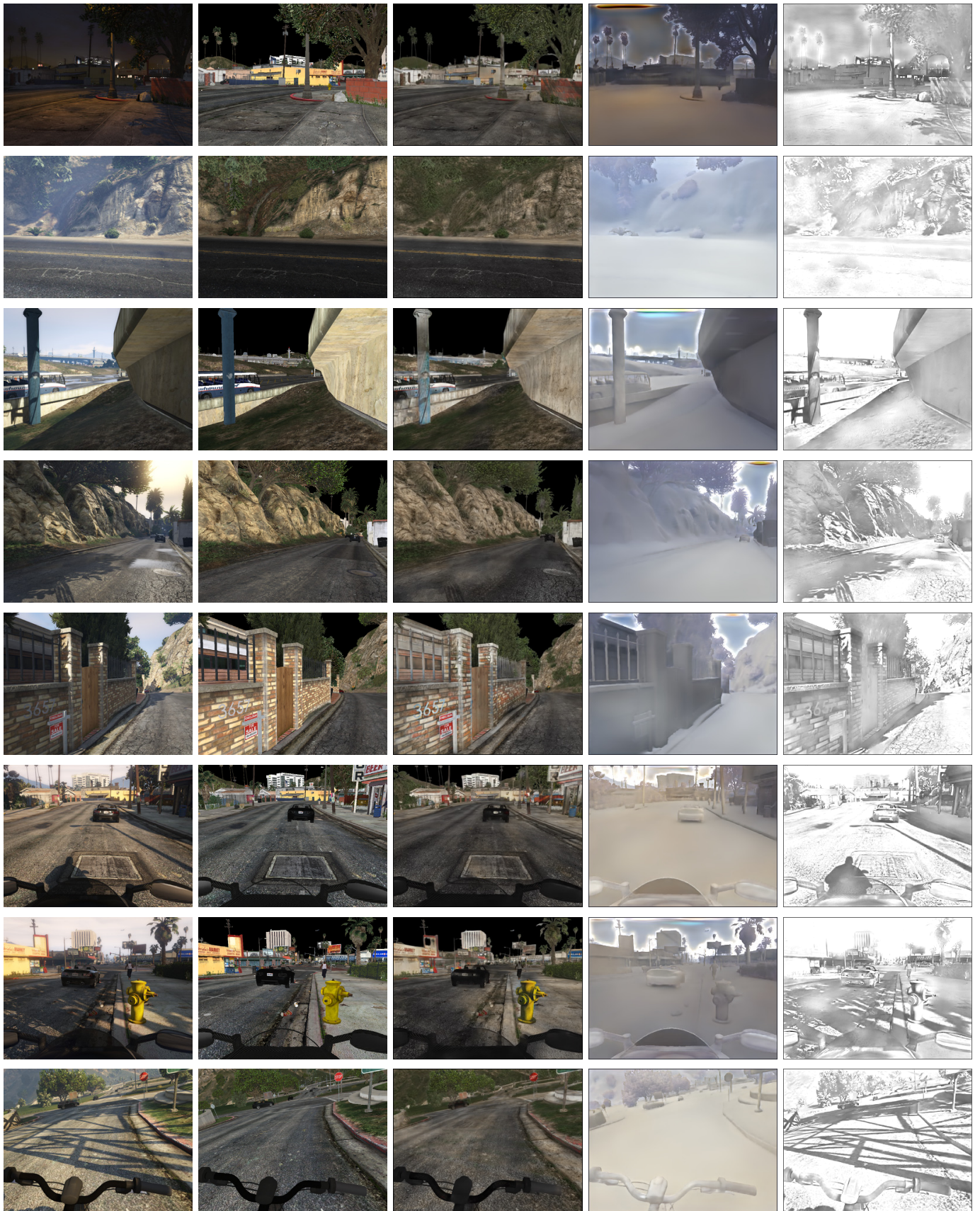


Fig. III. Qualitative results on KITTI dataset [33]. Input images are the same as in Fig. 5. Compared to Yu et al. [14], our method can recover shadow-free albedo with clearer details. Red boxes highlight regions where our method more correctly separates self-occlusion from shading.



Input

GT (A)

Ours (A)

Ours (S<sub>d</sub>)

Ours (S<sub>i</sub>)

Fig. IV. Qualitative results on FSVG dataset [19].



Input

GT (A)

Ours (A)

Ours (S<sub>d</sub>)

Ours (S<sub>i</sub>)

Fig. V. Qualitative results on FSVG dataset [19].



Input

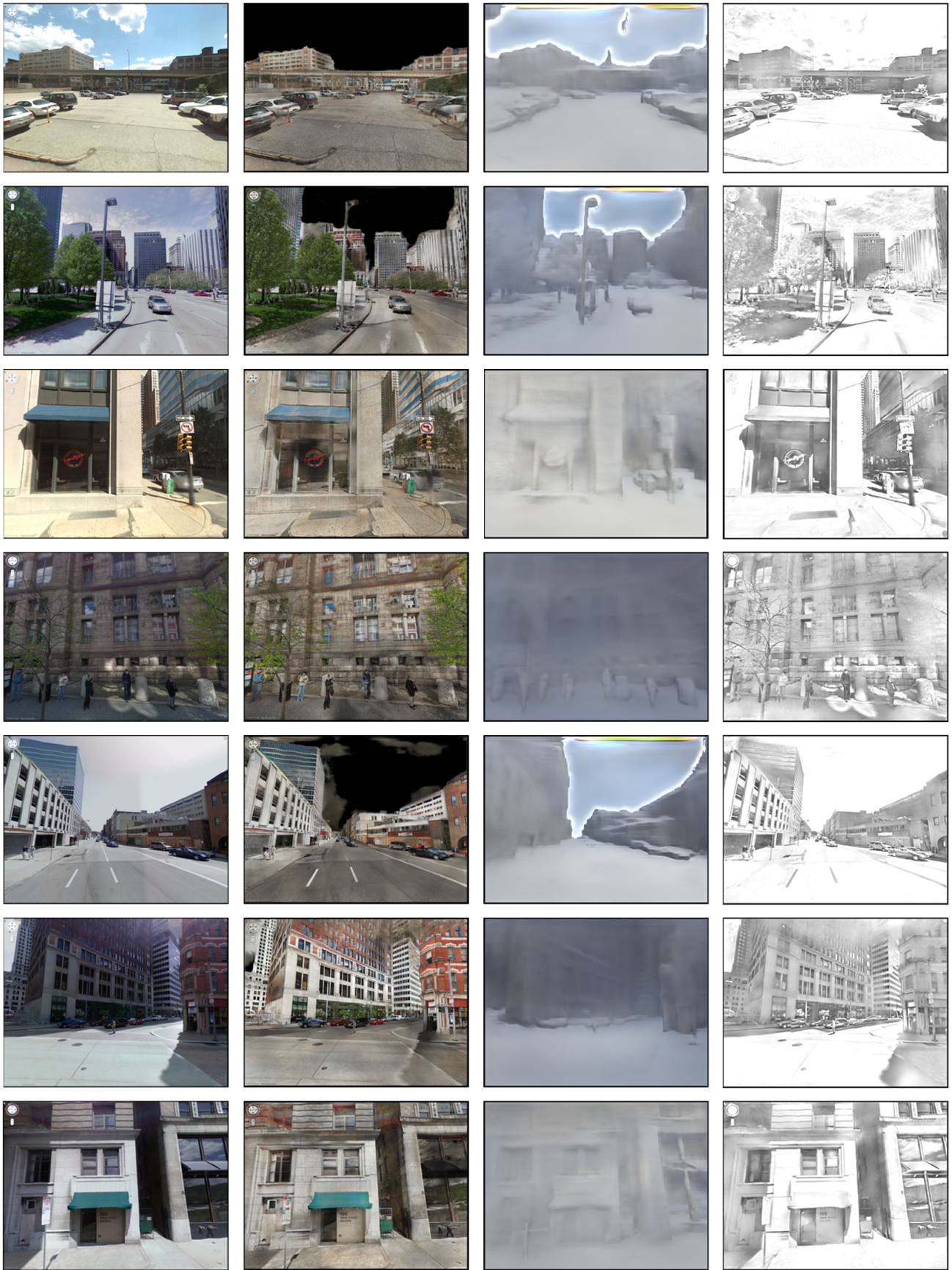
Ours (A)

Ours (S<sub>d</sub>)

Ours (S<sub>i</sub>)

Fig. VI. Qualitative results on images in the wild.





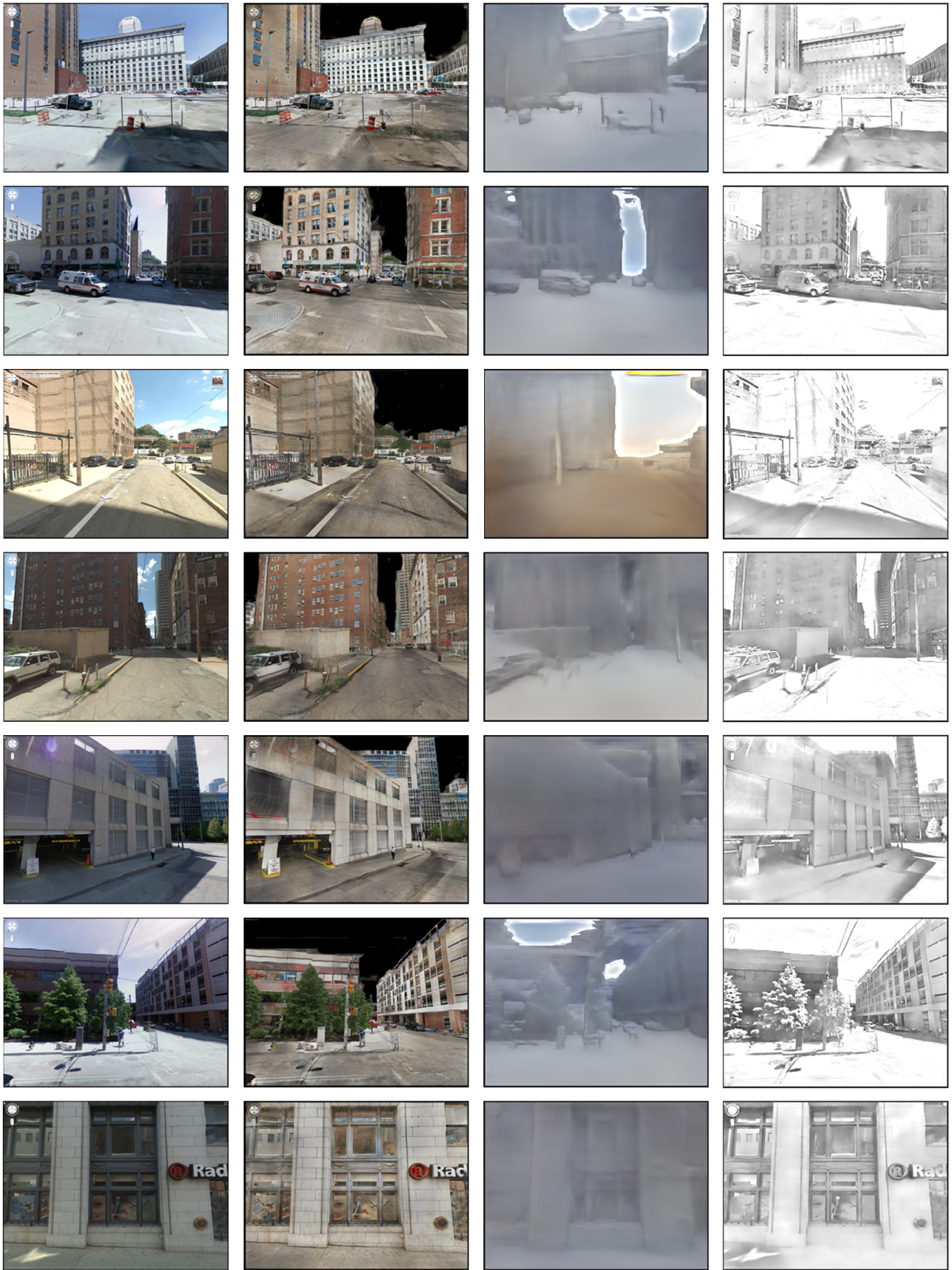
Input

Ours (A)

Ours ( $S_d$ )

Ours ( $S_i$ )

Fig. VII. Qualitative results on images in the wild.



Input

Ours (A)

Ours (S<sub>d</sub>)

Ours (S<sub>i</sub>)

Fig. VIII. Qualitative results on images in the wild.