



Deposited via The University of York.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/184951/>

Version: Accepted Version

Article:

Yu, Ye and Smith, William Alfred Peter (2022) Outdoor inverse rendering from a single image using multiview self-supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. pp. 3659-3675. ISSN: 0162-8828

<https://doi.org/10.1109/TPAMI.2021.3058105>

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

Outdoor inverse rendering from a single image using multiview self-supervision

Ye Yu and William A. P. Smith

Abstract—In this paper we show how to perform scene-level inverse rendering to recover shape, reflectance and lighting from a single, uncontrolled image using a fully convolutional neural network. The network takes an RGB image as input, regresses albedo, shadow and normal maps from which we infer least squares optimal spherical harmonic lighting coefficients. Our network is trained using large uncontrolled multiview and timelapse image collections without ground truth. By incorporating a differentiable renderer, our network can learn from self-supervision. Since the problem is ill-posed we introduce additional supervision. Our key insight is to perform offline multiview stereo (MVS) on images containing rich illumination variation. From the MVS pose and depth maps, we can cross project between overlapping views such that Siamese training can be used to ensure consistent estimation of photometric invariants. MVS depth also provides direct coarse supervision for normal map estimation. We believe this is the first attempt to use MVS supervision for learning inverse rendering. In addition, we learn a statistical natural illumination prior. We evaluate performance on inverse rendering, normal map estimation and intrinsic image decomposition benchmarks.

Index Terms—inverse rendering, shape-from-shading, intrinsic image decomposition, illumination estimation

1 INTRODUCTION

INVERSE rendering is the problem of estimating one or more of illumination, reflectance properties and shape from observed appearance (i.e. one or more images). In this paper, we tackle the most challenging setting of this problem; we seek to estimate all three quantities from only a single, uncontrolled image of an arbitrary scene. Specifically, we estimate a normal map, diffuse albedo map, cast shadow map and spherical harmonic lighting coefficients. This subsumes, and is more challenging than, several classical computer vision problems: (uncalibrated) shape-from-shading, intrinsic image decomposition, shadow removal and lighting estimation.

Classical approaches [1], [2] cast these problems in terms of energy minimisation. Here, a data term measures the difference between the input image and the synthesised image that arises from the estimated quantities. We approach the problem as one of image to image translation and solve it using a deep, fully convolutional neural network. However, there are no existing methods or devices that allow the capture of accurate ground truth for inverse rendering of uncontrolled, outdoor scenes. Hence, the labels required for supervised learning are not readily available. Instead, we use the data term for self-supervision via a differentiable renderer (see Fig. 2).

Single image inverse rendering is an inherently ambiguous problem. For example, any image can be explained with zero data error by setting the albedo map equal to the image, the normal map to be planar and the illumination such that the shading is unity everywhere. Hence, the data term alone cannot be used to solve this problem. For this reason, classical methods augment the data term with generic [1] or object-class-specific [3] priors. Likewise, we also exploit a statistical prior on lighting. However, our key insight that

enables the CNN to learn good performance is to introduce additional supervision provided by an offline multiview reconstruction or from time-lapse video.

While photometric vision has largely been confined to restrictive lab settings, classical geometric methods are sufficiently robust to provide multiview 3D shape reconstructions from large, unstructured datasets containing very rich illumination variation [4], [5]. This is made possible by local image descriptors that are largely invariant to illumination. However, these methods recover only geometric information and any recovered texture map has illumination “baked in” and so is useless for relighting. We exploit the robustness of geometric methods to varying illumination to supervise our inverse rendering network. We apply a multiview stereo (MVS) pipeline to large sets of images of the same scene. We select pairs of overlapping images with different illumination, use the estimated relative pose and depth maps to cross project photometric invariants between views and use this for supervision via Siamese training. In other words, geometry provides correspondence that allows us to simulate varying illumination from a fixed viewpoint. Finally, the depth maps from MVS provide coarse normal map estimates that can be used for direct supervision of the normal map estimation. This is important for our training schedule but note that the subsequent self-supervised learning allows the network to perform significantly better normal map estimation than simply training a network in a supervised fashion with MVS normal maps. Time-lapse video provides another useful constraint since albedo and surface normal maps remain constant while lighting (and therefore shading and shadowing) vary.

1.1 Contribution

Deep learning has already shown good performance on components of the inverse rendering problem. This includes

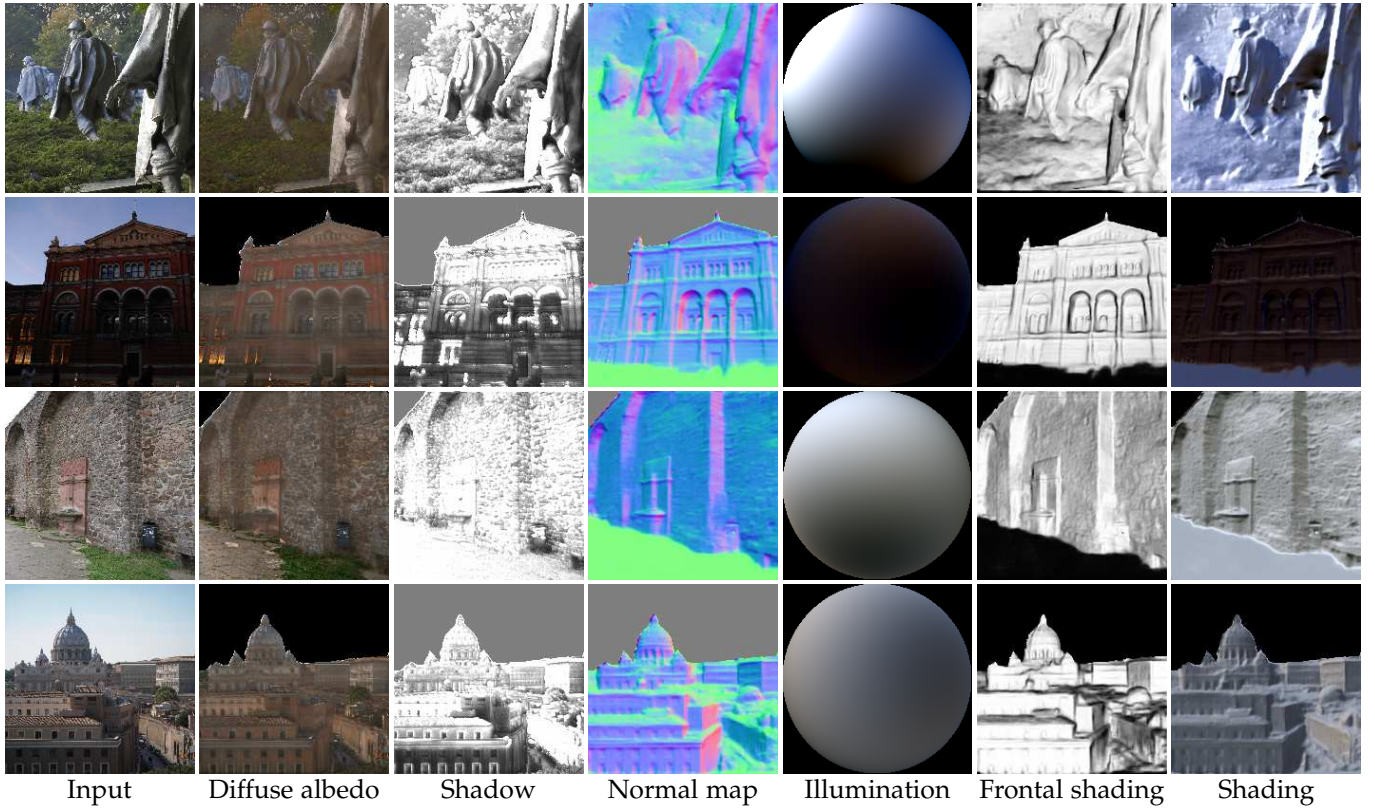


Fig. 1: Sample output from our inverse rendering network. From a single image (col. 1), we estimate albedo, shadow and normal maps and illumination (col. 2-5); re-rendering of our shape with (col. 6) frontal, white point source and with (col. 7) estimated spherical harmonic lighting.

monocular depth estimation [6], depth and normal estimation [7] and intrinsic image decomposition [8]. However, these works use supervised learning. For tasks where ground truth does not exist, such approaches must either train on synthetic data (in which case generalisation to the real world is not guaranteed) or generate pseudo ground truth using an existing method (in which case the network is just learning to replicate the performance of the existing method). In this context, we make the following contributions. To the best of our knowledge, we are the first to exploit MVS supervision for learning inverse rendering. Second, we are the first to tackle the most general version of the problem, considering arbitrary outdoor scenes and learning from real data, as opposed to restricting to a single object class [9] or using synthetic training data [10]. Third, we introduce a statistical model of spherical harmonic lighting in natural scenes that we use as a prior. Finally, the resulting network is the first to inverse render all of shape, reflectance and lighting in the wild, outdoors.

An earlier version of the work in this paper was originally presented in [11]. Here, we have extended the method in a number of ways. First, we additionally estimate a shadow map, avoiding cast shadows from being baked into the albedo map. Second, we improve the labels used for direct normal map supervision by explicitly detecting ground plane pixels (which have unreliable MVS depth estimates) and replacing them with an estimated ground plane normal direction. Third, we reformulate the least squares solution for lighting parameters to avoid numerical instability. Fourth, we introduce a new source of training

data by exploiting geometry and reflectance invariants in fixed position time-lapse video. Fifth, we also make changes to the architecture, use perceptual instead of mean squared error losses and use high resolution cross projections to avoid blurring the supervision signal. Taken together, these extensions improve performance and mean that we avoid the need to use generic priors such as smoothness as used in [11]. Finally, we provide a more thorough evaluation.

2 RELATED WORK

Classical approaches Classical methods estimate intrinsic properties by fitting photometric or geometric models using optimisation. Most methods require multiple images. From multiview images, a structure-from-motion/multiview stereo pipeline enables recovery of dense mesh models [5], [12] though illumination effects are baked into the texture. From images with fixed viewpoint but varying illumination, photometric stereo can be applied. More sophisticated variants consider statistical BRDF models [13], the use of outdoor time-lapse images [2] and spatially-varying BRDFs [14]. Attempts to combine geometric and photometric methods are limited. Haber et al. [15] assume known geometry (which can be provided by MVS) and inverse render reflectance and lighting from community photo collections. Kim et al. [16] represents the state-of-the-art and again use an MVS initialisation for joint optimisation of geometry, illumination and albedo. Some methods consider a single image setting. Jeson et al. [17] introduce

a local-adaptive reflectance smoothness constraint for intrinsic image decomposition on texture-free input images which are acquired with a texture separation algorithm. Barron et al. [1] present SIRFS, a classical optimisation-based approach that recovers all of shape, illumination and albedo using a sophisticated combination of generic priors. Romeiro et al. [18] simplify the 4D BRDF function to a bivariate representation by assuming an isotropic BRDF function is invariant to rotation about the half vector, and estimate the BRDF in a simple setting where illumination and geometry are known. Instead of tackling the problem in an optimisation-based manner, [19]–[21] formulate inverse rendering or sub-problems as a Bayesian probabilistic problem, paired with empirical and data-driven prior probabilities, and produce the solution by means of MAP estimation. Romeiro et al. [22] exploit the statistical distributions of natural environment maps at different scales as the prior probability distribution for the unknown illumination, and find the optimal BRDF function given the geometry, input image and the illumination prior distributions.

Deep depth prediction Direct estimation of shape alone using deep neural networks has attracted a lot of attention. Eigen et al. [6], [7] were the first to apply deep learning in this context. Subsequently, performance gains were obtained using improved architectures [23], post-processing with classical CRF-based methods [24]–[26] and using ordinal relationships for objects within the scenes [27]–[29]. Zheng et al. [10] use synthetic images for training but improve generalisation using a synthetic-to-real transformation GAN. However, all of this work requires supervision by ground truth depth. An alternative branch of methods explore using self-supervision from augmented data. For example, binocular stereo pairs can provide a supervisory signal through consistency of cross projected images [30]–[32]. Alternatively, video data can provide a similar source of supervision [33]–[36]. Tulsiani et al. [37] use multiview supervision in a ray tracing network. While all these methods take single image input, Ji et al. [38] tackle the MVS problem itself using deep learning.

Deep intrinsic image decomposition Intrinsic image decomposition is a partial step towards inverse rendering. It decomposes an image into reflectance (albedo) and shading but does not separate shading into shape and illumination. Even so, the lack of ground truth training data makes this a hard problem to solve with deep learning. Recent work either uses synthetic training data and supervised learning [8], [39]–[42] or self-supervision/unsupervised learning. Very recently, Li et al. [43] used uncontrolled time-lapse images allowing them to combine an image reconstruction loss with reflectance consistency between frames. This work was further extended using photorealistic, synthetic training data [44]. Ma et al. [45] also trained on time-lapse sequences and introduced a new gradient constraint which encourage better explanations for sharp changes caused by shading or reflectance. Baslamisli et al. [46] applied a similar gradient constraint while they used supervised training. Shelhamer et al. [47] propose a hybrid approach where a CNN estimates a depth map which is used to constrain a classical optimisation-based intrinsic image estimation.

Deep inverse rendering To date, solving the full inverse rendering problem using deep learning has received relatively little attention. One line of work simplifies the problem by restricting to a single object class, e.g. faces [9], meaning that a statistical face model can constrain the geometry and reflectance estimates. This enables entirely self-supervised training. Shu et al. [48] extend this idea with an adversarial loss. Sengupta et al. [49] on the other hand, initialise with supervised training on synthetic data, and fine-tuned their network in an unsupervised fashion on real images. Nestmeyer et al. [50] target realistic face relighting, combining an inverse rendering network with a renderer that combines both physics-based and learnt elements. Going beyond faces, Kanamori and Endo [51] consider whole body inverse rendering for relighting. Here, occlusion of the illumination environment becomes important so they show how to infer a light transport map. Aittala et al. [52] restrict geometry to almost planar objects and lighting to a flash in the viewing direction under which assumptions they can obtain impressive results. Gao et al. [53] consider the same planar scenario but with multiple images, enabling recovery of spatially varying reflectance properties. More general settings have been considered including natural illumination [54]. Philip et al. [55] focus on relighting of outdoor scenes but require multiple images and a 3D geometric proxy. Kulkarni et al. [56] show how to learn latent variables that correspond to extrinsic parameters allowing image manipulation. The only prior work we are aware of that tackles the full inverse rendering problem requires direct supervision [57]–[59]. Hence, it is not applicable to scene-level inverse rendering, only objects, and relies on synthetic data for training, limiting the ability of the network to generalise to real images. Very recently, Sengupta et al. [60] combined an inverse rendering network with a trainable, residual rendering network. By pre-training both on synthetic data in a supervised fashion, they can subsequently finetune the inverse rendering network on real data using self-supervision (with the residual rendering network kept fixed). Li et al. [61] trained a cascade of inverse rendering networks in which reflectance is modelled as a spatially-varying BRDF and global lighting as a spatially-varying spherical Gaussian. The need for synthetic training data limited these approaches to indoor scenes.

3 PRELIMINARIES

In this section we define notation, introduce basic models and assumptions and describe our shape representation.

We assume that a perspective camera observes a scene, such that the projection from 3D world coordinates, (u, v, w) , to 2D image coordinates, (x, y) , is given by:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix}, \quad \mathbf{P} = \mathbf{K} [\mathbf{R} \quad \mathbf{t}], \quad \mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where λ is an arbitrary scale factor, $\mathbf{R} \in SO(3)$ a rotation matrix, $\mathbf{t} \in \mathbb{R}^3$ a translation vector, f the focal length and (c_x, c_y) the principal point.

The inverse rendered shape estimate could be represented in a number of ways. For example, many previous methods estimate a viewer-centred depth map. However,

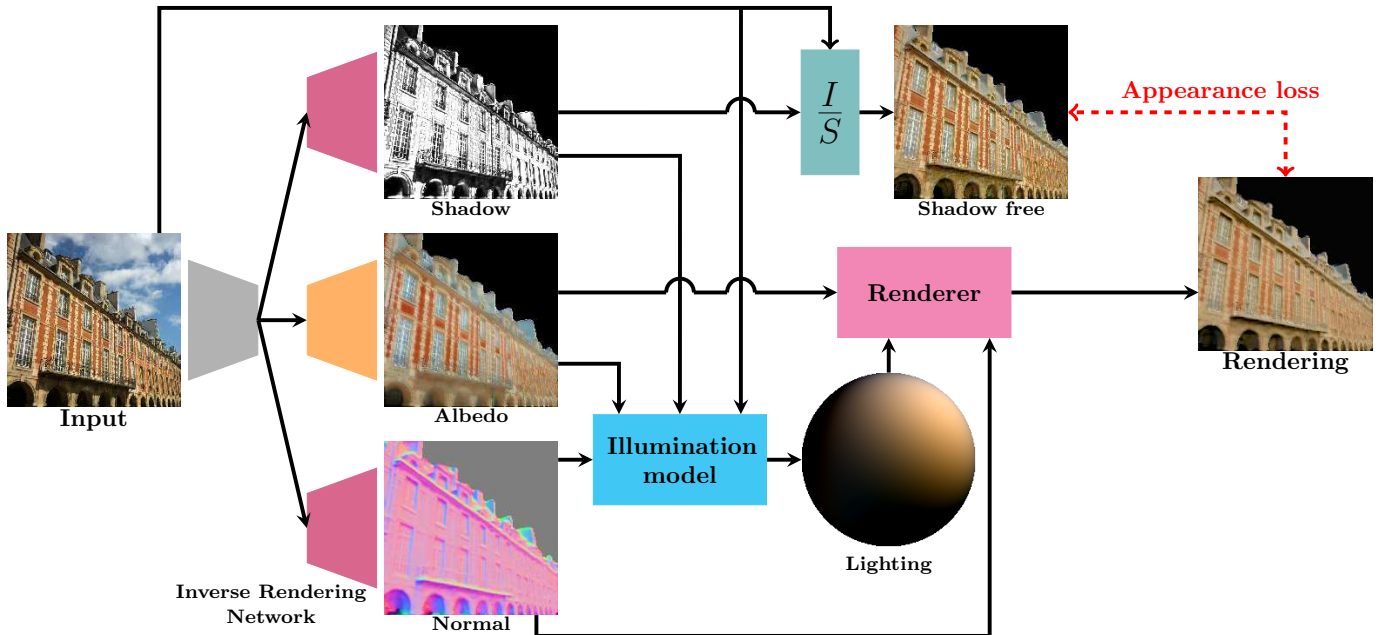


Fig. 2: At inference time, our network regresses shadow, diffuse albedo and normal maps from a single, uncontrolled image. These are used to infer shadow free and shading only images from which we compute least squares optimal spherical harmonic lighting coefficients. At training time, we introduce self-supervision via an appearance loss computed using a differentiable renderer and the estimated quantities.

local reflectance, and hence appearance, is determined by surface orientation, i.e. the local surface normal direction. So, to render a depth map for self-supervision, we would need to compute the surface normal. From a perspective depth map $w(x, y)$, the surface normal direction in camera coordinates is given by:

$$\bar{\mathbf{n}} = \begin{bmatrix} -fw_x(x, y) \\ -fw_y(x, y) \\ (x - c_x)w_x(x, y) + (y - c_y)w_y(x, y) + w(x, y) \end{bmatrix} \quad (2)$$

from which the unit length normal is given by: $\mathbf{n} = \bar{\mathbf{n}}/\|\bar{\mathbf{n}}\|$. The derivatives of the depth map in the image plane, $w_x(x, y)$ and $w_y(x, y)$, can be approximated by finite differences. However, (2) requires knowledge of the intrinsic camera parameters. This would severely restrict the applicability of our method. For this reason, we choose to estimate a surface normal map directly. An additional advantage of working in the surface normal domain is that the normal map can capture high frequency shape details without strictly satisfying integrability constraints. This leads to higher quality image renderings.

Although the surface normal can be represented by a 3D vector, since $\|\mathbf{n}\|_2 = 1$ it has only two degrees of freedom. So, our network estimates two quantities per-pixel: n_x/n_z and n_y/n_z . Since $n_z > 0$ for visible pixels we can compute the surface normal from the estimated quantities as:

$$\mathbf{n} = \frac{[n_x/n_z, n_y/n_z, 1]^T}{\|[n_x/n_z, n_y/n_z, 1]\|}. \quad (3)$$

We assume that appearance can be approximated by a local reflectance model under environment illumination, modulated by a scalar shadowing/ambient occlusion term. Specifically we use a Lambertian diffuse model with order 2 spherical harmonic lighting.

Alone, this cannot model phenomena such as cast shadows, spatially varying illumination and specularities. In [11], this results in these phenomena being baked into one or both of the albedo and normal maps. Of these phenomena, the most severe are cast shadows. We introduce an additional term that acts multiplicatively on the appearance predicted by the local spherical harmonics model. Without appropriate constraint, the introduction of this additional channel could lead to trivial solutions. Hence, we constrain it in two ways. First, we restrict it to the range $[0, 1]$ so that it can only downscale appearance. Second, it is a scalar quantity acting equally on all colour channels. Together, these restrictions encourage this channel to explain cast shadows and we refer to it as a shadow map. However, note that we do not expect it to be a physically valid shadow map nor that it contains only shadows.

Under this model, RGB intensity can be computed as

$$\mathbf{i}(x, y) = \begin{bmatrix} i_r(x, y) \\ i_g(x, y) \\ i_b(x, y) \end{bmatrix} = \boldsymbol{\alpha}(x, y) \odot s(x, y)\mathbf{B}(\mathbf{n}(x, y))\mathbf{l}, \quad (4)$$

where \odot is the Hadamard (element-wise) product, $\mathbf{l} \in \mathbb{R}^{27}$ contains the order 2, colour spherical harmonic colour illumination coefficients, $\boldsymbol{\alpha}(x, y) = [\alpha_r(x, y), \alpha_g(x, y), \alpha_b(x, y)]^T$ is the colour diffuse albedo, $s(x, y) \in [0, 1]$ is the shadowing weight and the order 2 basis $\mathbf{B}(\mathbf{n}) \in \mathbb{R}^{3 \times 27}$ is given by $\mathbf{B}(\mathbf{n}) = \mathbf{I}_3 \otimes \mathbf{b}(\mathbf{n})$ where \otimes is the Kronecker product and

$$\mathbf{b}(\mathbf{n}) = [1, n_x, n_y, n_z, 3n_z^2 - 1, n_x n_y, n_x n_z, n_y n_z, n_x^2 - n_y^2]. \quad (5)$$

This appearance model neglects high frequency illumination effects and interreflections. However, we found that in practice this model works well for typical outdoor scenes.

4 ARCHITECTURE

Our inverse rendering network (see Fig. 2) is an image-to-image network that regresses albedo, surface normal and shadow maps from a single image and uses these to estimate lighting. We now describe these components in more detail.

4.1 Trainable encoder-decoder

We implement a deep fully-convolutional neural network with skip connections based on the U-net architecture [62]. The specific choice of architecture is orthogonal to our contribution and we have experimented with using different variations without significant effect on the result. We use a single encoder and separate deconvolution decoders for albedo, shadow and normal map prediction. Albedo maps have 3 channel RGB output, normal maps have two channels that are converted to a normal map using (3) and shadow maps have a single channel. Both albedo and shadow maps are transformed to the $[0, 1]$ range by applying the hyperbolic tangent function followed by a scaling and offset. Both the convolutional subnetwork and deconvolutional subnetwork contain 15 layers and the activation functions are ReLUs.

4.2 Illumination separability

In order to estimate illumination parameters, one option would be to use a fully connected branch from the output of our encoder and train our network to predict it directly. Sengupta et al. [49] use such an approach by training a small CNN that takes image, albedo and normal features as input and uses 1×1 convolutions, average pooling and a fully connected layer to predict lighting. While such a network can potentially learn robust lighting estimation, it is a black box whose behaviour is not interpretable and there is no guarantee that the estimated illumination is optimal with respect to the estimated image, albedo and normals. We tried training such a network as part of our inverse rendering network but training did not converge - perhaps because of our more challenging unconstrained outdoor scenes as opposed to the face images used in [49]. We observe that lighting can be inferred from the input image and estimated albedo, shadow and normal maps, making its explicit prediction redundant. We avoid this redundancy by computing least squares optimal lighting in-network from the estimated maps. An additional advantage of this approach is that the architecture remains fully convolutional and so can process images of any size at inference time.

We detect and mask the sky region by applying PSPNet [63]. The sky region does not adhere to our reflectance model and albedo and normal maps are meaningless in these regions. Hence, we use only the non-sky foreground in illumination estimation.

For an input image with K foreground pixels, we stack the K RGB values to form the vector $\mathbf{i}_{\text{obs}} \in \mathbb{R}^{3K}$. We assume that nonlinear gamma has been applied with a fixed $\gamma = 2.2$. We invert the nonlinear gamma and equate with our model from (4) extended to the whole image:

$$\begin{aligned} \mathbf{i}_{\text{obs}}^\gamma &= [i_r^1, \dots, i_r^K, i_g^1, \dots, i_g^K, i_b^1, \dots, i_b^K]^T \\ &= \boldsymbol{\alpha} \odot (\mathbf{1}_{3 \times 1} \otimes \mathbf{s}) \odot \mathbf{B} \mathbf{l} \end{aligned} \quad (6)$$

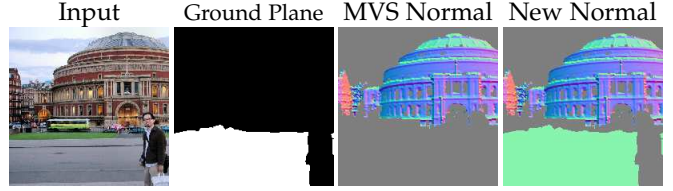


Fig. 3: The mask for ground plane is obtained by PSPNet [63]. We then define the normal for masked ground plane by the normal of fitted camera plane.

where each row of $\mathbf{B} \in \mathbb{R}^{3K \times 27}$ contains the spherical harmonic basis for the estimated surface normal at one pixel and the foreground pixels from the estimated albedo and shadow maps are stacked to form $\boldsymbol{\alpha} \in \mathbb{R}^{3K}$ and $\mathbf{s} \in \mathbb{R}^K$ respectively. We can now rearrange (6) into linear least squares form with respect to the lighting vector:

$$\mathbf{i}_{\text{obs}}^\gamma = [\mathbf{1} \otimes (\boldsymbol{\alpha} \odot (\mathbf{1}_{3 \times 1} \otimes \mathbf{s})) \odot \mathbf{B}] \mathbf{l} = \mathbf{A} \mathbf{l}. \quad (7)$$

We can now solve for the spherical harmonic illumination coefficients in a least squares sense over all foreground pixels. This can be done using any method, so long as the computation is differentiable such that losses dependent on the estimated illumination can have their gradients back-propagated into the inverse rendering network. We solve using the pseudoinverse of \mathbf{A} , i.e. $\mathbf{l}^* = \mathbf{A}^+ \mathbf{i}_{\text{obs}}^\gamma$. Note that the pseudoinverse \mathbf{A}^+ has a closed form derivative [64]. Fig. 2 shows the inferred normal, albedo and shadow maps, and a visualisation of the least squares estimated lighting.

5 DATA AND PREPROCESSING

In order to supervise the training of our network, we begin by applying a multiview reconstruction pipeline to large, uncontrolled, outdoor image collections. The application of structure-from-motion followed by multiview stereo (which we refer to simply as MVS) enables both camera poses and dense 3D scene models to be estimated from large, uncontrolled image sets. Of particular importance to us, these pipelines are relatively insensitive to illumination variation between images in the dataset since they rely on matching local image features that are themselves illumination insensitive. We use a modification of the recent COLMAP [65] pipeline. We emphasise that MVS is run offline prior to training and that at inference time our network uses only single images of novel scenes. We use the MVS output for three sources of supervision.

From the MVS geometry, we can compute per-view depth maps and subsequently estimate surface normal maps using (2). These normal maps are used as coarse direct supervision during training. However, the MVS models almost always do not reconstruct the ground plane. We find that this significantly disrupts training such that our network does not learn to predict good shape or albedo estimates in the ground plane region. For this reason, we preprocess the MVS normal maps to replace ground plane normals with an assumed ground plane normal direction. To this end, we detect ground plane pixels using PSPNet [63] and inpaint normal direction to detected pixels in the normal map. Although the true ground plane direction is

unknown, a reasonable estimate can be made from the MVS reconstruction. We assume that the cameras positions are located approximately a fixed height above the ground and fit a plane to their positions using principal components analysis (PCA). The normal to this plane defines the ground plane normal which we rotate into camera coordinates and use for inpainting. Note, the estimate of the ground plane orientation is done using the whole MVS reconstruction for a scene and is the same for all images in that scene. Hence, if a particular image has large parts of the ground plane occluded, the estimated ground plane for that image is not disrupted. The only per-image part of the process is estimating the ground segmentation that defines which pixels have their normal directions replaced by the estimated ground plane orientation. A sample result is shown in Fig. 3.

Our primary dataset is from MegaDepth [28]. This contains dense depth maps and camera calibration parameters estimated from crawled Flickr images. The pre-processed images have arbitrary shapes and orientations. For ease of training, we resize images such that the smaller side is of size 200 pixels, then crop multiple 200×200 square images. We choose our crops to maximise the number of pixels with defined depth values. Where possible, we crop multiple images from each image, achieving augmentation as well as standardisation. We adjust the camera parameters to account for the scale and crop. We create mini-batches in which all pairs of images within the mini-batch have overlapping views (defined as having similar camera locations and similar centres of mass of their backprojected depth maps). We also enforce sufficient illumination variation by cross-projecting (see Section 6.3) each image into the others and discarding pairs where the correlation coefficient of intensity histograms is close to 1). Finally, before inputting an image to our network, we detect and mask the sky region using PSPNet [63]. Apart from this, we input the images as stored, without undoing the assumed nonlinear gamma. In total, we use 117,030 images from MegaDepth. For parts of our evaluation, we finetune using additional datasets (described in Section 7). These are preprocessed in the same way.

6 LOSSES AND TRAINING STRATEGY

As shown in Fig. 2, we use a data term (the error between predicted and observed appearance) for self-supervision. However, inverse rendering using only a data term is ill-posed (an infinite set of solutions can yield zero data error) and so we use additional sources of supervision, all of which are essential for good performance. We describe all sources of supervision in this section.

6.1 Self-supervision via differentiable rendering

Given estimated normal, shadow and albedo maps and spherical harmonic illumination coefficients, we compute a predicted image using (4). This local illumination model is straightforward to differentiate. Self-supervision is provided by the error between the predicted and observed RGB colour values. We found that performance was significantly improved by computing this error in a shadow-free space. To do so, we divide out the estimated shadow map from

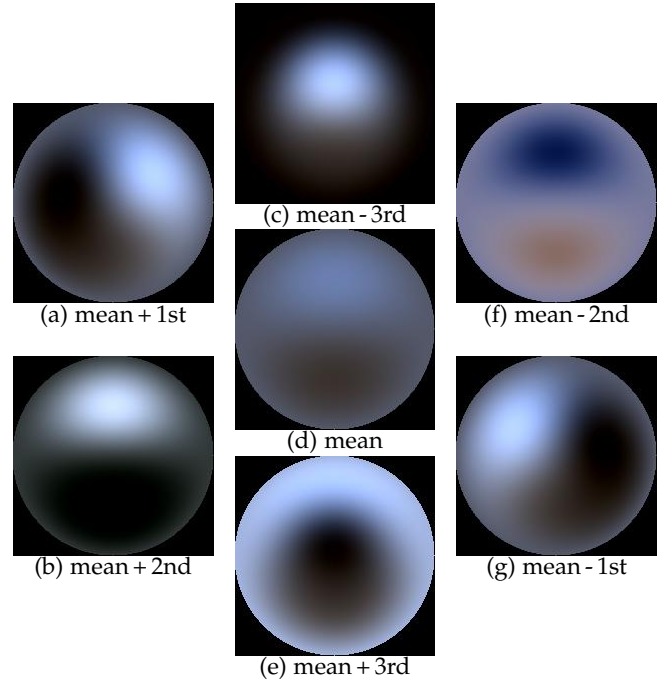


Fig. 4: Statistical illumination model. The central image shows the mean illumination. The two diagonals and the vertical show the first 3 principal components.

the linearised observed image and clamp to one (avoiding numerical instabilities caused by very small shadow values):

$$\mathbf{i}_{\text{SF}} = \min [1, \mathbf{i}_{\text{obs}}^\gamma \oslash (\mathbf{1}_{3 \times 1} \otimes \mathbf{s})], \quad (8)$$

where \oslash is Hadamard (element-wise) division. We compare this shadow free image to the image predicted using only the local spherical harmonic model and our estimated illumination and albedo and normal maps. Inspired by [66], computing this appearance loss directly in RGB space does not yield the best results since it is not the best measure of the perceptual quality of the reconstruction. For this reason, we use two transformations to perceptually meaningful feature spaces and compute an L2 loss in those spaces:

$$\ell_{\text{appearance}} = \varepsilon(\boldsymbol{\alpha} \odot \mathbf{B}\mathbf{l}, \mathbf{i}_{\text{SF}}), \quad (9)$$

where

$$\varepsilon(\mathbf{x}, \mathbf{y}) = w_{\text{VGG}} \|\text{VGG}(\mathbf{x}) - \text{VGG}(\mathbf{y})\|_2 + w_{\text{LAB}} \|\text{LAB}(\mathbf{x}) - \text{LAB}(\mathbf{y})\|_2, \quad (10)$$

where VGG computes features from the first two convolution blocks of a pre-trained VGG network [67] and LAB transforms to the LAB colour space. We mask these losses pixel-wise using the sky mask, using appropriate downsampling of the mask within the VGG layers. We set the weights as $w_{\text{VGG}} = 2.5$ and $w_{\text{LAB}} = 0.5$ in all experiments.

6.2 Natural illumination model and prior

The spherical harmonic lighting model in (4) enables efficient representation of complex lighting. However, even within this low dimensional space, not all possible illumination environments are natural. The space of natural illumination possesses statistical regularities [68]. We can use this

knowledge to constrain the space of possible illumination and enforce a prior on the illumination parameters. To do this, we build a statistical illumination model (see Fig. 4) using a dataset of 79 HDR spherical panoramic images taken outdoors [69], [70]. For each environment, we compute the spherical harmonic coefficients, $\mathbf{l}_i \in \mathbb{R}^{27}$. Since the overall intensity scale is arbitrary, we also normalise each lighting matrix to unit norm, $\|\mathbf{l}_i\|_2 = 1$, to avoid ambiguity with the albedo scale. Our illumination model in (6) uses surface normals in a viewer-centred coordinate system. So, the dataset must be augmented to account for possible rotations of the environment relative to the viewer. Since the rotation around the vertical (v) axis is arbitrary, we rotate the lighting coefficients by angles between 0 and 2π in increments of $\pi/18$. In addition, to account for camera pitch or roll, we additionally augment with rotations about the u and w axes in the range $(-\pi/6, \pi/6)$. This gives us a dataset of 139,356 environment maps. We then build a statistical model, such that any illumination can be approximated as:

$$\mathbf{l} = \mathbf{Q} \text{diag}(\sigma_1, \dots, \sigma_D) \boldsymbol{\alpha} + \bar{\mathbf{l}}. \quad (11)$$

where $\mathbf{Q} \in \mathbb{R}^{27 \times D}$ contains the principal components, $\sigma_1^2, \dots, \sigma_D^2$ are the corresponding eigenvalues, $\bar{\mathbf{l}} \in \mathbb{R}^{27}$ is the mean lighting coefficients and $\boldsymbol{\alpha} \in \mathbb{R}^D$ is the parametric representation of \mathbf{l} . We use $D = 18$ dimensions. Under the assumption that the original data is Gaussian distributed then the parameters are normally distributed: $\boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. When we compute lighting, we do so within the subspace of the statistical model. i.e. we substitute (11) into (7) and solve linearly for $\boldsymbol{\alpha}$. In addition, we introduce a prior loss on the estimated lighting vector: $\ell_{\text{lighting}} = \|\boldsymbol{\alpha}\|_2^2$.

6.3 Multiview stereo supervision

Cross-projection We use the MVS poses and depth maps to establish correspondence between views, allowing us to cross-project quantities between overlapping images. Given an estimated depth map, $w(x, y)$, in view i and camera matrices for views i and j , a pixel (x, y) can be cross-projected to location (x', y') in view j via:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{K}_j \begin{bmatrix} \mathbf{R}_j & \mathbf{t}_j \end{bmatrix} \begin{bmatrix} \mathbf{R}_i^T & -\mathbf{R}_i^T \mathbf{t}_i \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} w(x, y) \mathbf{K}_i^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ 1 \end{bmatrix} \quad (12)$$

In practice, we perform the cross-projection in the reverse direction, computing non-integer pixel locations in the source view for each pixel in the target view. We can then use bilinear interpolation of the source image to compute quantities for each pixel in the target image. Since the MVS depth maps contain holes, any pixels that cross project to a missing pixel are not assigned a value. Similarly, any target pixels that project outside the image bounds of the source are not assigned a value. We denote the cross-projection of any quantity \mathbf{x}_i defined on the pixels of image i onto the pixels of image j as $\text{proj}_{i \rightarrow j}(\mathbf{x}_i)$.

Direct normal map supervision The per-view depth maps provided by MVS can be used to estimate normal maps, albeit that they are typically coarse and incomplete (see e.g. Fig. 3, col. 4). We compute guide normal maps from

the depth maps and intrinsic camera parameters estimated by MVS using (2). The guide normal maps, $\mathbf{n}_{\text{guide}}$, are used for direct supervision by computing an angular error loss $\ell_{\text{NM}} = \arccos(\mathbf{n}_{\text{guide}} \cdot \mathbf{n}_{\text{est}})$ where \mathbf{n}_{est} are the estimated surface normals. This loss is summed over foreground pixels in the supervision depth map.

Albedo consistency loss Diffuse albedo is an intrinsic quantity. Hence, we expect that albedo estimates of the same scene point from two overlapping images should be the same, even if the illumination varies between views. We train our network in a Siamese fashion on overlapping pairs within each mini-batch and use cross projection to compute an albedo consistency loss, again computing the loss as the sum of two perceptual losses:

$$\ell_{\text{albedo}} = \varepsilon(\boldsymbol{\alpha}_i, \text{proj}_{j \rightarrow i}(\boldsymbol{\alpha}_j)), \quad (13)$$

where $\boldsymbol{\alpha}_i$ and $\boldsymbol{\alpha}_j$ are the estimated albedo maps in the i th and j th images respectively. This loss is masked both by the sky mask in image i and by the cross-projected pixels for which image j has a defined MVS depth value. The weights w_{VGG} and w_{LAB} are set the same as in (9). The albedo consistency loss is visualised by the blue arrows in Fig. 6.

Cross-rendering loss We also use a hybrid of cross-projection and appearance loss. This remedies a deficiency in using albedo consistency alone. Cross-projection at low (network input) resolution yields blurred output. The result is that the albedo consistency loss encourages blurred (but consistent) albedo maps. We formulate another loss that encourages albedo consistency but which does not require cross projection of low resolution albedo maps. This cross-rendering loss also helps disentangle scenes into view-dependent illumination and physically-constant diffuse albedo. Here, we recombine predicted albedo with illumination estimated from another view, whose appearance should be reproduced by such a recombination.

We cross-project the images in the dataset at their original high resolution, using the high resolution depth maps. We define the cross projection of image j into image i using the original high resolution image, \mathbf{i}_j^{HR} , in view j as:

$$\mathbf{i}_{j \rightarrow i} = \text{downsample}(\text{proj}_{j \rightarrow i}(\mathbf{i}_j^{\text{HR}})), \quad (14)$$

where the downsample function downsamples to network input resolution. Note that this cross projection need only be done once as preprocessing step on the training dataset.

We then mix the albedo and normal predictions from image i with the lighting and cross projected shadow map from image j and compare it against the high resolution cross projection from image j . Again, we do so in a shadow free space, so define the cross-projected shadow free image:

$$\mathbf{i}_{j \rightarrow i}^{\text{SF}} = \min \left[\mathbf{1}, \mathbf{i}_{j \rightarrow i} \odot \left(\mathbf{1}_{3 \times 1} \otimes \text{proj}_{j \rightarrow i}(\mathbf{s}_j) \right) \right], \quad (15)$$

and then compute the cross-rendering loss as:

$$\ell_{\text{cross-rend}} = \varepsilon(\mathbf{i}_{j \rightarrow i}^{\text{SF}}, \boldsymbol{\alpha}_i \odot \mathbf{B}_i \mathbf{R}_{j \rightarrow i} \mathbf{l}_j), \quad (16)$$

where \mathbf{B}_i is the spherical harmonic basis computed from the normal map estimated from image i , \mathbf{l}_j is the spherical harmonic lighting coefficients estimated in image j and $\mathbf{R}_{j \rightarrow i}$ rotates the coefficients from the camera coordinate system in image j to image i . We visualise the process of computing this loss in Fig. 5.

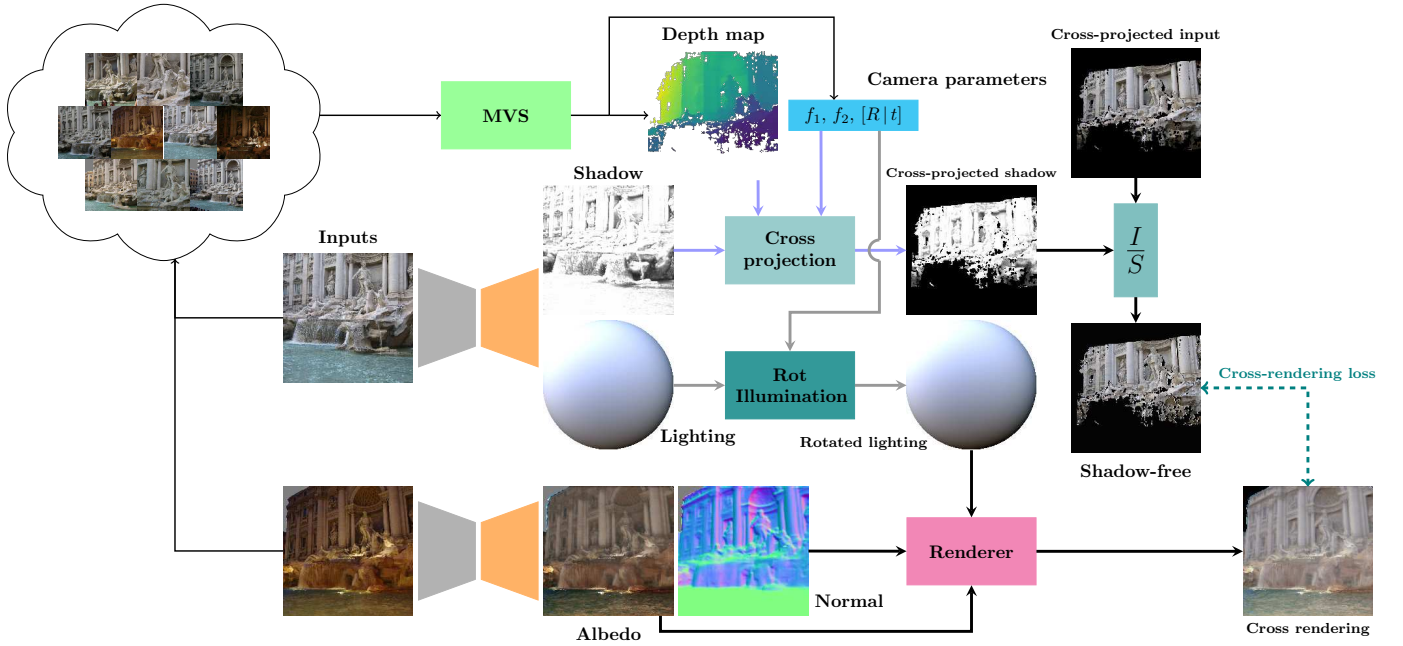


Fig. 5: Cross-rendering loss. A cross-rendered image is generated by relighting the albedo and normal predictions from one image and the lighting prediction from the other. Before rendering, we rotate the lighting to align it with the new view given relative camera poses. The rendering is shadow free, so the shadow is removed from ground truth relighting image. Both shadow and ground truth relighting image are cross-projected from the view from which new lighting is taken.

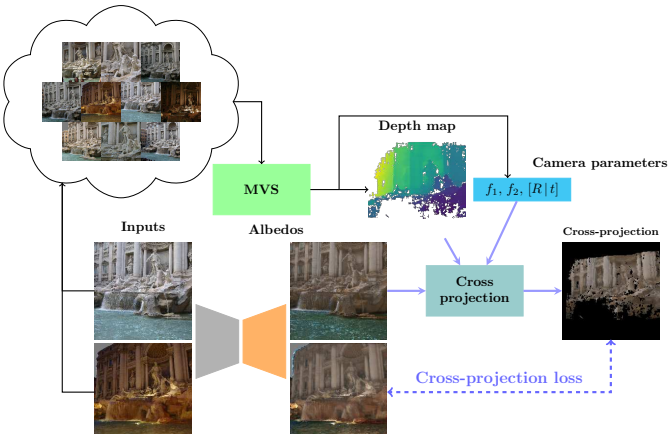


Fig. 6: Albedo consistency loss. Given the depth map and camera parameters from performing MVS over image collections, the albedo prediction from one view can be cross-projected to another view. For each image, we compute the albedo consistency loss by measuring the difference between albedo prediction and cross-projected albedo prediction from other views within each batch.

6.4 Training

We train our network to minimise:

$$\ell = w_1 \ell_{\text{appearance}} + w_2 \ell_{\text{NM}} + w_3 \ell_{\text{albedo}} + w_4 \ell_{\text{cross-rend}} + w_5 \ell_{\text{lighting}} \quad (17)$$

where the weights are set as $w_1 = 0.1$, $w_2 = 1.0$, $w_3 = 0.1$, $w_4 = 0.1$ and $w_5 = 0.005$.

We use the Adam optimiser [71]. We found that for convergence to a good solution it is important to include

Method	Reflectances		Normals	
	MSE	LMSE	Mean	Median
MegaDepth [28]	-	-	43.6	43.0
Godard et al. [32]	-	-	82.4	82.1
Nestmeyer et al. [72]	0.0149	0.0169	-	-
BigTime [43]	0.0116	0.0135	-	-
SIRFS [1]	0.0070	0.0275	50.6	48.5
InverseRenderNet [11]	0.0112	0.0128	40.0	38.0
ours	0.0093	0.0111	31.2	30.0

TABLE 1: Quantitative inverse rendering results on MegaDepth dataset [28]. Reflectance (albedo) errors are measured against multiview inverse rendering result [16] with optimal per-channel scaling applied and normals against MVS results.

a pre-training phase. During this phase, the surface normals used for illumination estimation and for the appearance-based losses are the MVS normal maps. This means that the surface normal prediction decoder is only learning from the direct supervision loss, i.e. it is learning to replicate the MVS normals. After this initial phase, we switch to full self-supervision where the predicted appearance is computed entirely from estimated quantities.

7 EVALUATION

There are no existing benchmarks for inverse rendering of outdoor scenes, partly because there is no existing method for satisfactorily estimating scene-level geometry, reflectance and lighting for in-the-wild, outdoor scenes. For this reason, we developed our own outdoor benchmark based on the MegaDepth dataset [28]. We augment this with four further benchmarks that evaluate performance of a subset of the network outputs. First, we use the BigTime time-

Method	Reconstruction		Consistency	
	MD [28]	BT [43]	MD [28]	BT [43]
Nestmeyer et al. [72]	0.1003	0.0333	0.0140	0.0067
BigTime [43]	0.0372	-	0.0089	-
InverseRenderNet [11]	0.0158	0.0124	0.0168	0.0118
ours	0.0082	0.0082	0.0082	0.0072

TABLE 2: Reconstruction error and reflectance consistency for MegaDepth [28] and BigTime dataset [43]. MSE is used for computing reconstruction errors, and MSE with optimal scaling is used for computing albedo consistency.

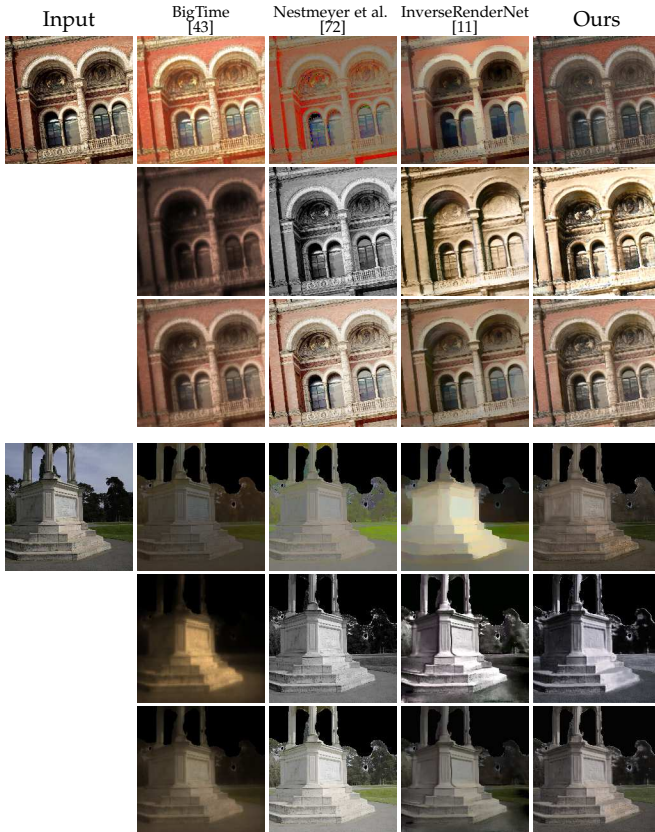


Fig. 7: Qualitative results for reconstruction. The first and fourth rows are albedo predictions, and second and fifth rows are shading predictions from labelled methods. The reconstruction results for each method is composed by albedo and shading and is shown on third and last rows.

lapse dataset [43] to evaluate albedo consistency under varying illumination. Second, we evaluate on the related task of intrinsic image decomposition using an indoor benchmark [73] which demonstrates the generalisation ability of our network. Third, we quantify our normal prediction using the DIODE benchmark [74], which contains high quality outdoor depth and normal maps along with corresponding images. Finally, we evaluate illumination estimation on the benchmark dataset in [75] consisting of multi-illumination and multi-view images along with ground truth HDR illumination environment maps.

7.1 Evaluation on MegaDepth

We split the MegaDepth dataset [28] into training and testing data, and evaluate our performance on testing data. We

evaluate the inverse rendering results through four aspects. Firstly, the normal estimation performance can be directly compared against the MVS geometry. We quantify performance using angular error in degrees. Second, we evaluate albedo estimation against the output a state-of-the-art multi-view inverse rendering algorithm [16]. We are not aware of any existing method or device that can measure accurate scene-level albedo under in-the-wild conditions. Hence, we could not obtain ground truth for our inverse rendering benchmark. The method of [16] provides pseudo ground truth that, because it exploits multi-view/multi-illuminant images and known camera calibration, we expect to be considerably more accurate than the single image methods to which we do compare. In particular, given the output from their pipeline, we perform rasterisation to generate albedo ground truth for every input image. Note that both sources of ground truth here are themselves only estimations, e.g. the albedo ground truth contains ambient occlusion baked in. The colour balance of the estimated albedo is arbitrary, so we compute per-channel optimal scalings prior to computing errors. We use two metrics - MSE (mean squared error) and LMSE (local mean squared error), which are commonly used for evaluating albedo predictions. The quantitative evaluations are shown in Tab. 1. For depth prediction methods, we first compute the optimal scaling onto the ground truth geometry, then compute numerical derivatives to find surface normals. These methods can only be evaluated on normal prediction. Intrinsic image methods can only be evaluated on albedo prediction. We can see that our network performs best in both albedo and normal predictions. Qualitative results can be found in Fig. 9. Note that, relative to the inverse rendering methods, our result is able to explain cast shadows in the shadow map, avoiding baking them into the albedo map. As supplementary material we also show results on a video, performing inverse rendering on each frame independently.

Except the direct evaluations against the ground truth inverse rendering results, we propose two additional metrics which are reconstruction accuracy and albedo consistency. The introduced two metrics reflects how well our inverse rendering results explain the input image, and whether the physical invariants are consistent across the scene. They indirectly evaluate the correctness of our inverse rendering result. Quantitative results are shown in Tab. 2 Here we measure albedo consistency by cross-projecting albedo predictions to overlapping images, followed by calculating the difference between the now-aligned albedo predictions. We show qualitative results in Fig. 7 and 8. Relative to [11], our albedo maps preserve high frequency detail since we do not use a smoothness prior. Note also that we are able to extract albedo maps with consistent colour from images with very different illumination colour (Fig. 8).

7.2 Evaluation on BigTime

Time-lapse data allows us to use the same reconstruction and consistency metrics without the need for cross-projection. We do so on the BigTime time-lapse dataset [43]. The quantitative results are shown in Tab. 2. Note that the BigTime [43] intrinsic image method is excluded from the comparison because the method itself is entirely trained

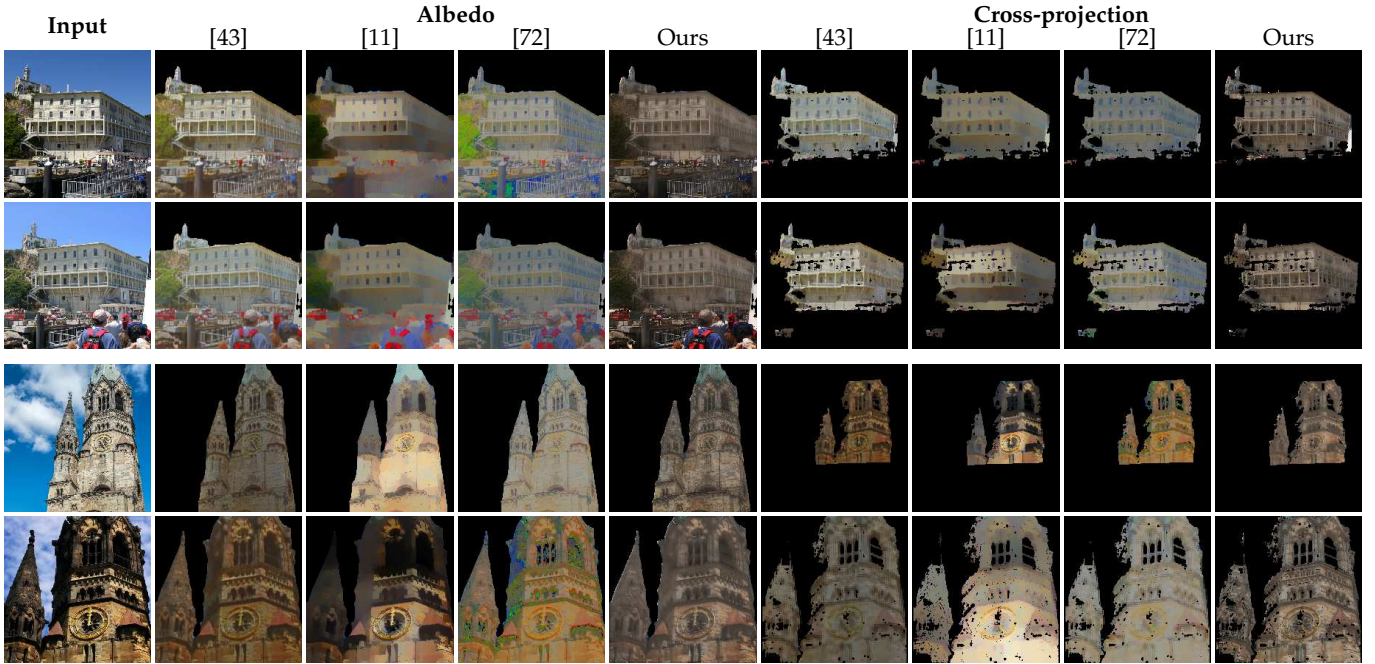


Fig. 8: Qualitative results for albedo consistency. For each consecutive pair of rows we show results for two overlapping images of the same scene. Albedo predictions are shown in col. 2-5 which are then cross-projected to the viewpoint of the other image in the pair in col. 6-9. Results shown for col. 2 and 6: BigTime [43], col. 3 and 7: InverseRenderNet [11], col. 4 and 8: Nestmeyer et al. [72], col. 5 and 9: ours.

based on the dataset. All other methods including our proposed work are trained without fine-tuning on this dataset. To fairly perform the evaluations, we select 15 scenes out of nearly 200 scenes which contains large portion of indoor scenes and some outdoor scenes from the dataset. Some example results are shown in Fig. 10. In the last example in particular, note our ability to avoid baking cast shadows into the albedo map.

7.3 Evaluation on IIW

The standard benchmark for intrinsic image decomposition is Intrinsic Images in the Wild [73] (IIW) which is almost exclusively indoor scenes. Since our training regime requires large multiview image datasets, we are restricted to using scene-tagged images crawled from the web, which are usually outdoors. In addition, our illumination model is learnt on outdoor, natural environments. For these reasons, we cannot perform training or fine-tuning on this indoor benchmark. However, we perform fine-tuning using two other datasets to improve performance on the intrinsic image task on indoor scenes. Specifically, we first add the BigTime dataset [43] for which we can compute self rendering, albedo consistency and cross-rendering losses without the need for any cross projection. Second, we sidestep the lack of surface normal labels in BigTime [43] by introducing the indoor normal supervisions from the NYU dataset [76]. To train our network with the mixture of all three sources of data, each training iteration is separated into two sub-steps. Firstly, the network is trained by MegaDepth [28] and NYU [76] to learn indoor self-reconstruction and normal prediction. Secondly, by freezing the normal prediction decoder, only shadow and albedo prediction decoders are trained

Methods	Training data	WHDR
Nestmeyer [72]	IIW	19.5
Li et al. [44]	IIW	17.5
Sengupta et al. [60]	IIW	16.7
Narihira et al. [39]	Sintel+MIT	37.3
Shi et al. [77]	ShapeNet	59.4
BigTime [43]	BigTime	20.3
InverseRenderNet [11]	MegaDepth	21.4
Ours	MD+BT+NYU	21.1

TABLE 3: Quantitative results on IIW benchmark using WHDR percentage (lower is better). The second column shows which dataset on which the networks were trained. The upper block are trained on IIW training set.

by BigTime [43]. We show qualitative results in Fig. 11 and quantitative results in Tab. 3. Note that our performance is very close to the state-of-the-art for methods not fine-tuned on the IIW training set, despite the fact that our inverse rendering solution is more constrained than intrinsic image methods (we must explain shading in terms of geometry, lighting and shadows whereas intrinsic image methods are free to compute arbitrary shading maps).

7.4 Evaluation on DIODE

DIODE [74] is a very recent geometry estimation benchmark comprising images and registered depth/normal maps acquired with a Lidar scanner. The benchmark is divided into indoor and outdoor scenes and we evaluate only on the outdoor scene test set. To date, the only published result for normal map estimation is the method of Eigen et al. [7] against which we quantitatively compare in Tab. 4. We show qualitative example results on this dataset in Fig. 12.

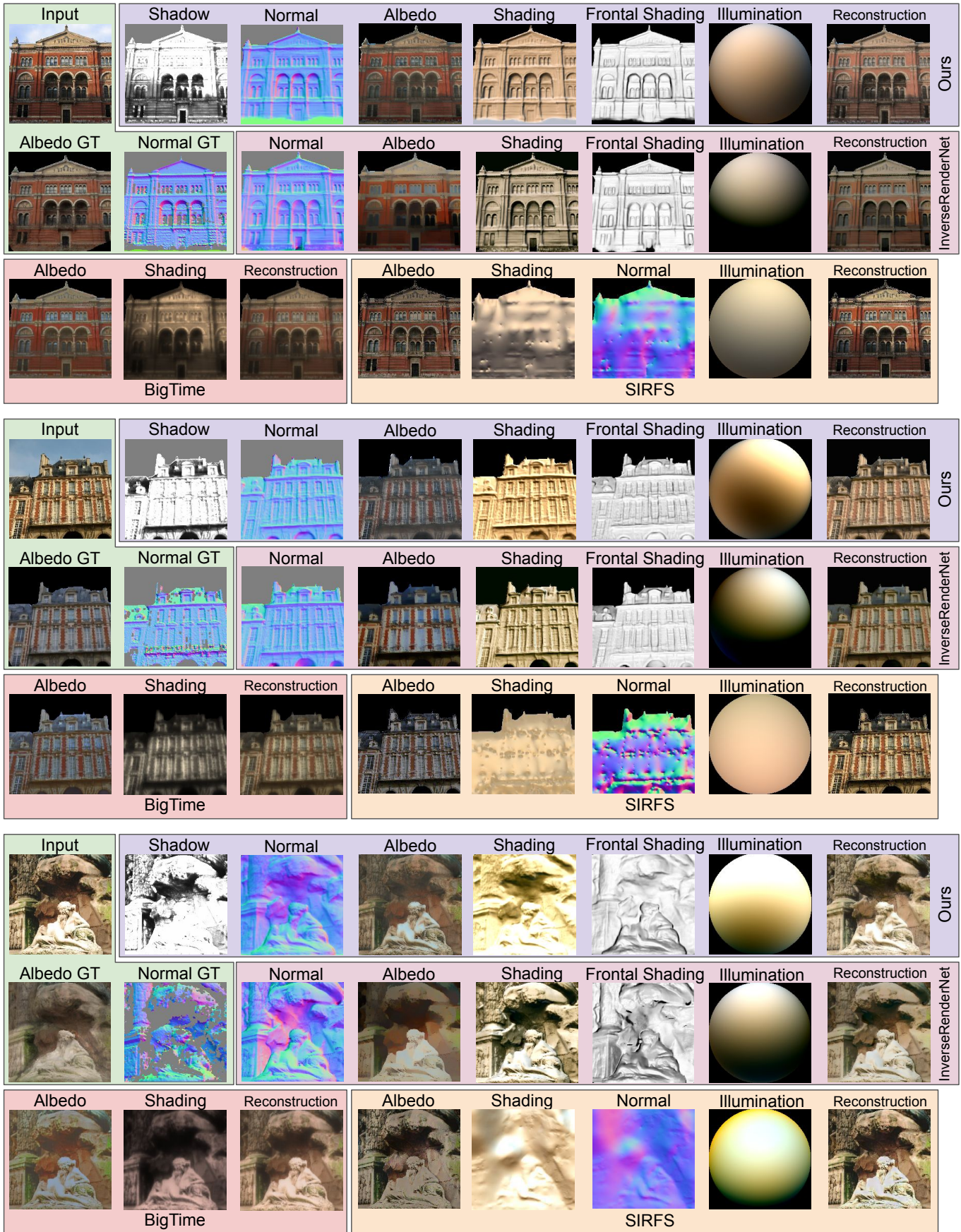


Fig. 9: Qualitative results for our inverse rendering benchmark. We show comparison against InverseRenderNet [11], BigTime [43] and SIRFS [1].

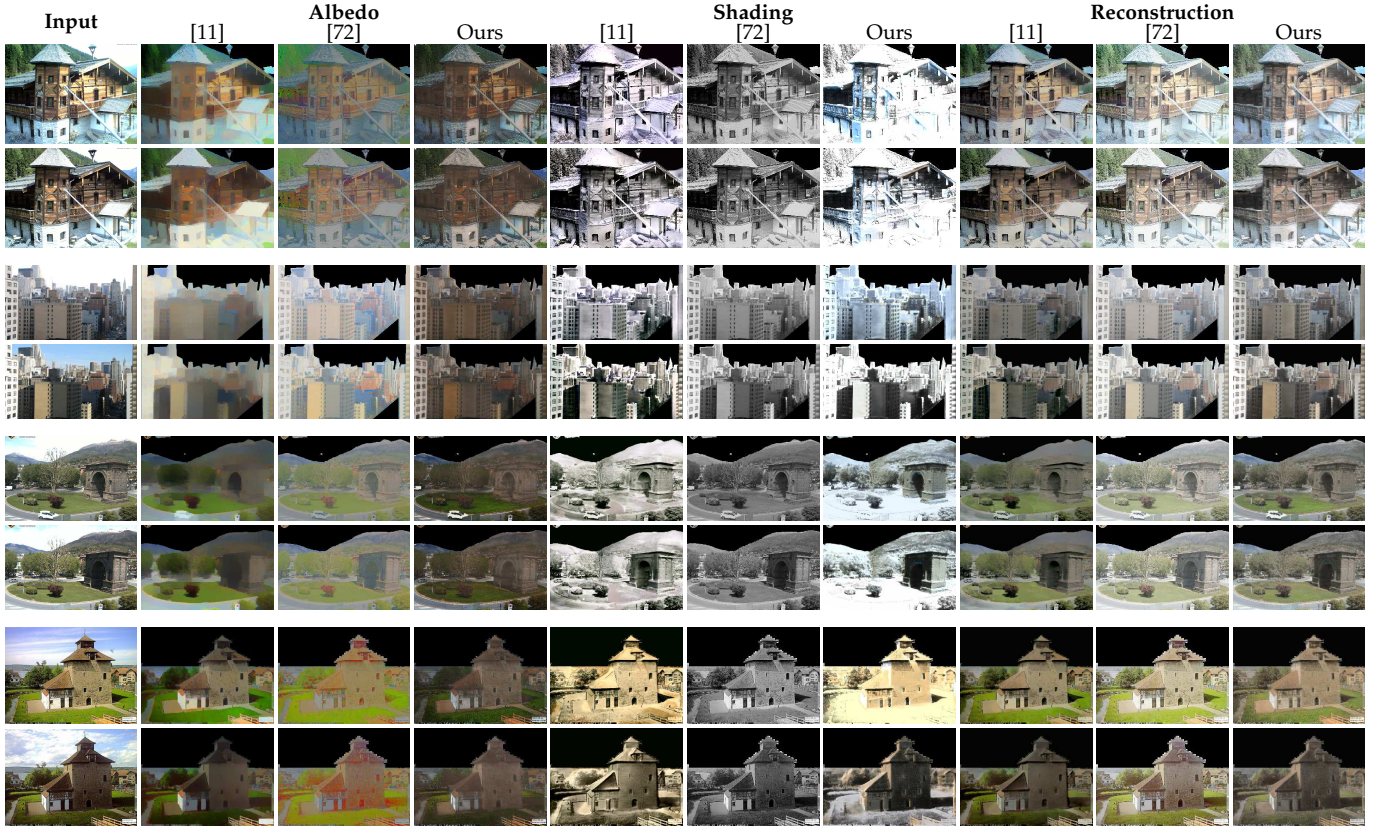


Fig. 10: Qualitative results on BigTime data [43]. Each consecutive pair of rows shows results for two different frames from a time-lapse sequence. Col. 1: input, col. 2-4: albedo, col. 5-7: shading, col. 8-10: reconstruction. For image we show comparison between InverseRenderNet [11], Nestmeyer et al. [72] and ours.

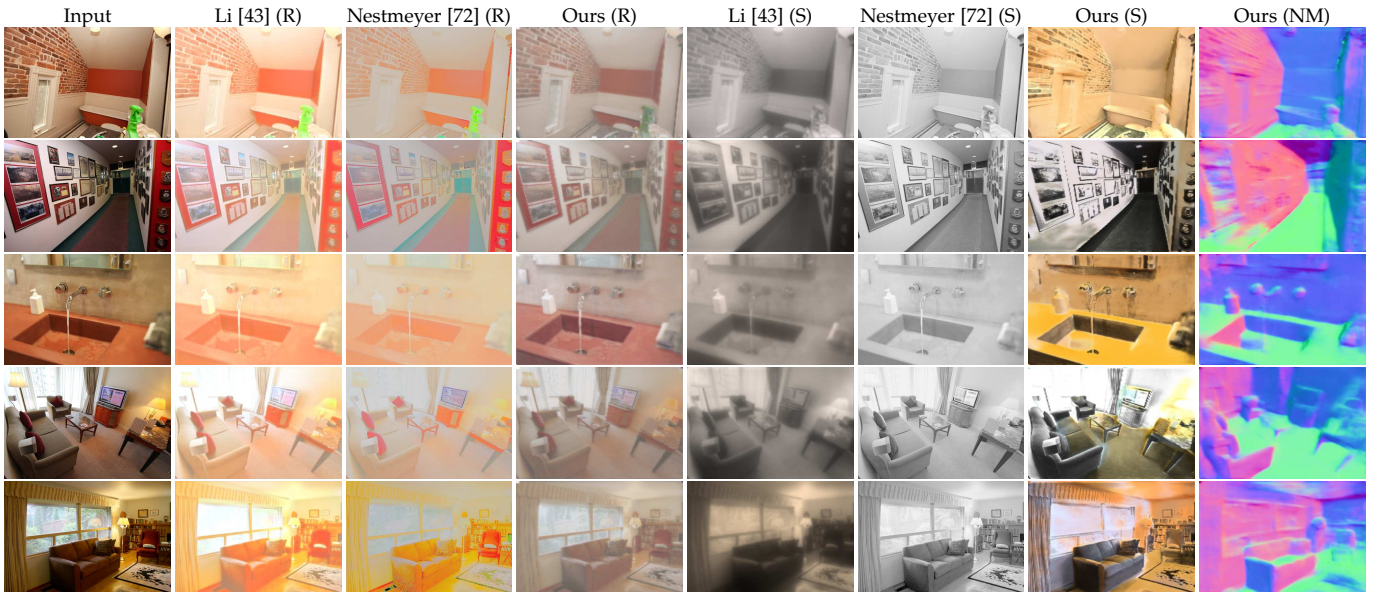


Fig. 11: Qualitative intrinsic image results for IIW benchmark. Col. 2-4: reflectance predictions from [43], [72] and ours. Col. 5-7: corresponding shading predictions. Col. 8: surface normal prediction for our method.

7.5 Illumination estimation

The benchmark dataset in [75] consists of images capturing a monument under six different illumination conditions from multiple viewpoints. Along with this multi-view and multi-illumination image collection, the six HDR illumina-

tion environment maps and necessary alignments between image and environment map pairs are also provided. In order to evaluate our performance on lighting estimation, we feed each image into our network and compare our lighting prediction with ground truth. Since our network can only infer the lighting represented by spherical har-

Methods	Mean	Median
Eigen et al. [7]	31.9	24.7
ours	23.9	15.5

TABLE 4: Quantitative surface normal prediction errors on the DIODE dataset [74]. We show mean and median angular errors in degrees for the outdoor test set.



Fig. 12: Qualitative results for surface normal prediction on DIODE dataset [74]. Left to right: input, ground truth normal map, estimated normal map.

Methods	Global scale	Per-colour scale
SIRFS [1]	0.100	0.089
InverseRenderNet [11]	0.050	0.041
Ours	0.038	0.033

TABLE 5: Quantitative results for illumination estimation. We show global-scale and per-colour-scale MSE errors.

monics, we project the ground truth environment map onto order 2 spherical harmonics. To ensure a fair evaluation the comparison regarding backside lighting is omitted, so we compare only the front side by measuring the distance between hemispheres lit by our lighting prediction and by ground truth spherical harmonic lighting. We show quantitative evaluations in Tab. 5 and qualitative evaluations in Fig. 13. Since there is an overall scale ambiguity between albedo and lighting, in quantitative evaluations, two scaling methods are applied. The global scaling metric is computed by applying an optimal global intensity scale prior to measuring errors, and per-colour scaling method is conducted by applying an optimal scaling to each colour channel. Accordingly, the qualitative results for these two scaling methods are shown in Fig. 13.

7.6 Ablation study

In this section, we seek insight on some key design choices by comparing the performance of our proposed network

Methods	MD reflectance (MSE)	IIW	Illumination (global scale)
Full	0.0093	21.1	0.038
w/o VGG	0.0095	21.6	0.053
w/o cross-projection loss	0.0153	21.6	0.052
w/o cross-rendering loss	0.0101	21.4	0.052

TABLE 6: Quantitative comparison for ablation study. Here, we select some representative metrics used in Sec. 7.1, 7.3 and 7.5, which are MSE-based reflectance errors on MegaDepth test data (col. 1), IIW benchmark score (col. 2), and global-scaled Illumination estimation errors (col. 3).

and ablated models on evaluation datasets used above. We report the comparison between our full model and ablated model trained without VGG loss, without cross-projection loss and without cross-rendering loss. The quantitative results are summarised in Tab. 6. Our proposed model outperforms the ablated methods on all metrics across all datasets. Fig. 14 visualises the qualitative comparison on different training setups. Here, without cross-projection loss the network cannot remove shadows from albedo prediction. Without cross-rendering loss the network can only predict over-smoothed albedo. Without VGG loss, albedo prediction looks ‘bleached’, since shading and shadow predictions capture too much darkness - shading predictions of this ablated variant shows less brightness contrast and darker intensity than that of our full model. While training without VGG loss results in larger errors, we found that finetuning the weights w_1 , w_3 and w_4 of losses for specific tasks leads to better performance than our full model. However, it cannot providing consistently the best performance across all metrics with a uniform weights setting.

8 LIMITATIONS AND FAILURE CASES

We present illustrative failure cases in Fig. 15. In the first row, the dark plaque in the centre is mainly explained by shadowing rather than dark albedo. Meanwhile the actual cast shadow, while present in the shadow map, still corrupts the albedo. The second row illustrates poor performance in regions that lack MVS supervision. In particular, the estimation of shadows cast on the ground plane are usually worse compared with shadows on buildings. Similarly, MVS reconstructions fail on dynamic regions containing people and so our result tends to treat pedestrians as planar surfaces (see the third row). In addition, Fig. 13 highlights the remaining colour ambiguity between illumination and albedo. As shown in the globally scaled results, our result always yields a biased prediction in terms of the colour, although the intensity distribution is plausible.

9 CONCLUSIONS

We have shown for the first time that the task of inverse rendering can be learnt from real world images in uncontrolled conditions. Our results show that “shape-from-shading” in the wild is possible and our results are far superior to classical methods. We are also competitive against state-of-the-art methods trained specifically for related subtasks. It is interesting to ponder how this feat is achieved. We believe the reason this is possible is because of the large range of

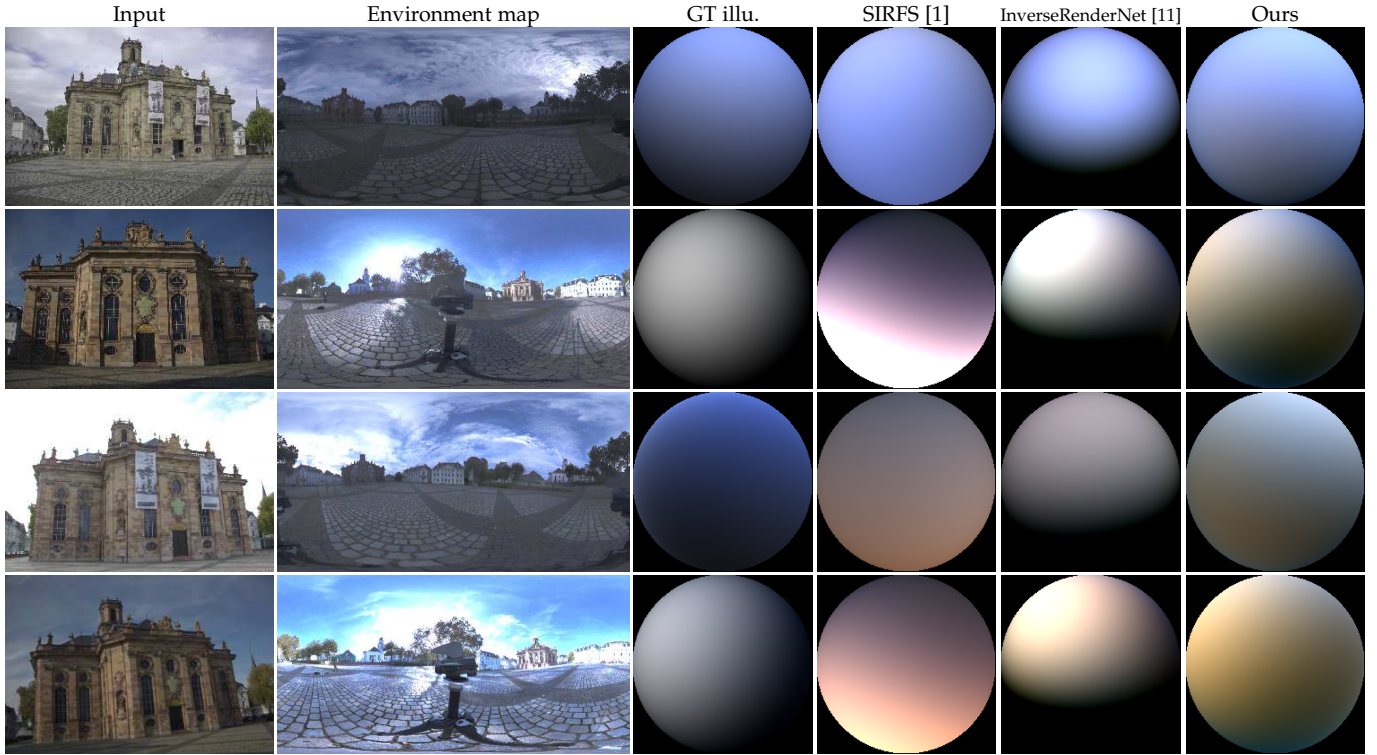


Fig. 13: Qualitative evaluation for illumination estimation. Captured environment maps are shown in col. 2, which are projected to spherical harmonics and used for relighting hemisphere to generate ground truth illumination estimations as demonstrated in col. 3. Illumination estimates from SIRFS [1], InverseRenderNet [11] and our proposed method are shown in cols 4-6. The first two rows are per-colour scaled, and last two are global scaled.

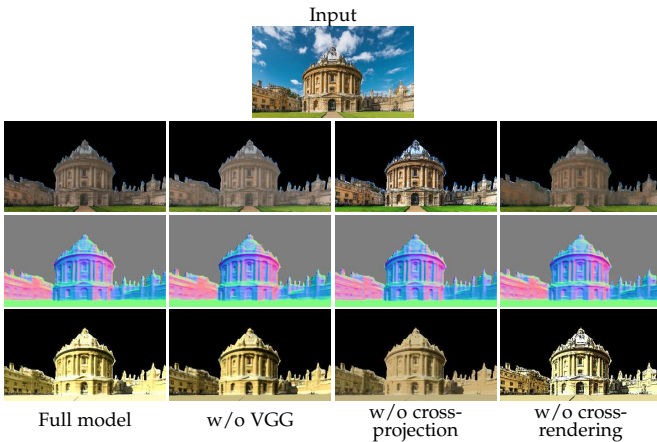


Fig. 14: Qualitative comparison for ablation study. Row 2: albedo, row 3: surface normals, row 4: shading. Ablation conditions are shown column-wise.

cues that the deep network can exploit, for example shading, texture, ambient occlusion, perhaps even high level semantic concepts learnt from the diverse data. For example, once a region is recognised as a “window”, the possible shape and configuration is much restricted. Recognising a scene as a man-made building suggests the presence of many parallel and orthogonal planes. These sort of cues would be extremely difficult to exploit in hand-crafted solutions.

An interesting question is whether our network learns general principles of shape-from-shading that generalise be-

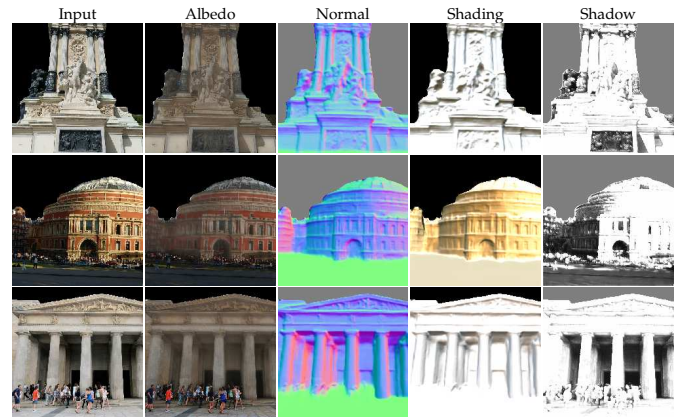


Fig. 15: Failure cases. Left to right: input, albedo estimate, normal estimate, shading and shadow estimate.

yond the types of outdoor, primarily urban scenes on which the network was trained. Another question is whether the network’s perceptions and resolutions of ambiguities are similar to that of a human. We show some qualitative examples to suggest that both of these are indeed the case. We begin in Fig. 16 by showing shape estimation results on the plates shading illusion. We note initially that our network is able to estimate meaningful shape from images that are completely unlike the training data. The plates are all the right side up, i.e. they are convex. The illusion was originally presented in the orientation shown in (c), where lighting

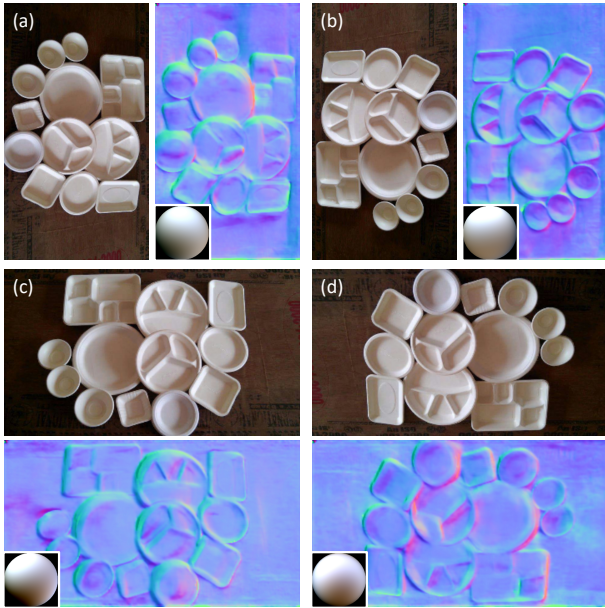


Fig. 16: Shape estimation results on the plates shading illusion. For four orientations of the image (a-d) we show the input image and the estimated normal map with estimated lighting inset.

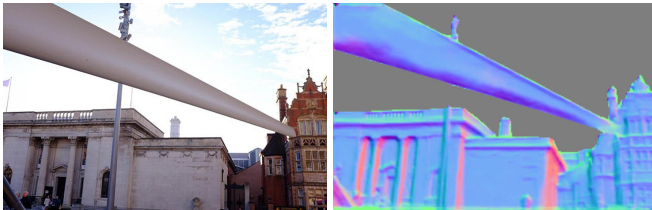


Fig. 17: Shape estimation on the turbine blade illusion [79].

is from the side. With no prior for a particular direction for side lighting, human viewers flip between interpreting the plates as convex or concave. In (a) and (b), the human preference for lighting from above [78] is shared by our network which interprets the plates as all convex (a) and all concave (b), estimating top lighting in both cases. For (c) and (d), although our lighting model can capture side lighting (Fig. 4, first component), because the scene is locally ambiguous it is resolved differently for different parts of the image (mix of convex and concave interpretations) and the least squares lighting estimate is an average of the two interpretations, resulting in an approximately top down lighting estimate.

In Fig. 17 we show shape estimation results on the turbine blade illusion [79]. Here, the convexity prior preferred by humans interprets the S-shaped cross section as a cylinder leading to an inconsistent lighting condition relative to the rest of the scene. This makes the blade look fake, as if it has been added afterwards. In this case, our network does not seem to have learnt such a strong convexity prior and correctly reconstructs the S-shaped profile. This can partly be explained by the fact that our network is forced to reconstruct a single, global illumination condition which differs from the human interpretation of this scene.

In Fig. 18 we show two example results on Trompe-

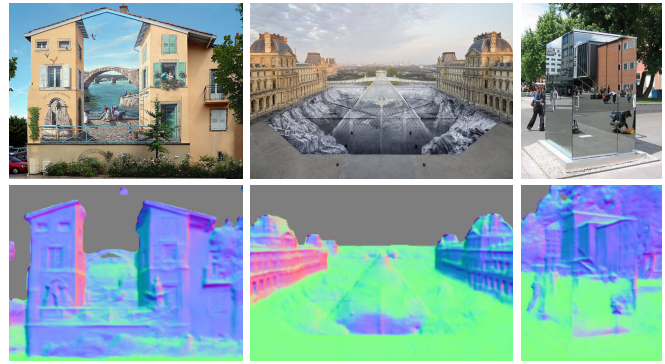


Fig. 18: Shape estimation in the presence of Trompe-l'œil and mirror illusions.

l'œil illusions and one for a mirror illusion. Trompe-l'œil use albedo variations (i.e. painted surface texture) to depict 3D scenes, giving the illusion of additional geometric variation. A mirror illusion is similar except that the albedo variation is a reflection of another part of the real scene. In all examples, the shape reconstruction is fooled by the illusion. In the example in the first column, both the sky segmentation network and our inverse rendering network are fooled. The shape of the painted windows, doors and fountain as well as the false building corners are recovered. For the reflection illusion in the third column, the shape of the reflected scene is reconstructed. This is not surprising since we do not model reflections and the MVS supervision would not correctly deal with reflective surfaces.

There are many promising ways in which this work can be extended. First, our modelling assumptions could be relaxed, for example using more general reflectance models and estimating global illumination effects. Second, our network could be combined with a depth prediction network. Either the two networks could be applied independently and then the depth and normal maps merged [80], or a unified network could be trained in which the normals computed from the depth map are used to compute the losses we use in this paper. Networks that explicitly reason about scene layout [81] could be used to directly supervise ground plane predictions. Third, our network could benefit from losses used in training intrinsic image decomposition networks [43]. Fourth, our lighting prior could be extended to better handle indoor scenes. Finally, our fixed reflectance and illumination model could be made partially learnable in order to be able to better explain real world appearance [60]. Our code, trained model and inverse rendering benchmark data is available at <https://github.com/YeeU/InverseRenderNet>.

ACKNOWLEDGMENT

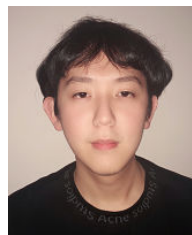
W. Smith was supported by the Royal Academy of Engineering under the Leverhulme Trust Senior Fellowship scheme. The Titan Xp used for this research was donated by the NVIDIA Corporation.

REFERENCES

- [1] J. T. Barron and J. Malik, "Shape, illumination, and reflectance from shading," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015.

- [2] F. Langguth, "Photometric stereo for outdoor webcams," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 262–269.
- [3] O. Aldrian and W. Smith, "Inverse rendering of faces with a 3d morphable model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1080–1093, 2013.
- [4] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," *Proc. International Conference on Computer Vision (ICCV)*, pp. 1–8, 2007.
- [5] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, 2010.
- [6] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Advances in neural information processing systems (NIPS)*, 2014, pp. 2366–2374.
- [7] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. International Conference on Computer Vision (ICCV)*, 2015, pp. 2650–2658.
- [8] L. Lettry, K. Vanhoey, and L. Van Gool, "DARN: a deep adversarial residual network for intrinsic image decomposition," in *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 1359–1367.
- [9] A. Tewari, M. Zollhöfer, H. Kim, P. Garrido, F. Bernard, P. Pérez, and C. Theobalt, "Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction," in *Proc. International Conference on Computer Vision (ICCV)*, vol. 2, no. 3, 2017, p. 5.
- [10] C. Zheng, T.-J. Cham, and J. Cai, "T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks," in *Proc. European Conference on Computer Vision (ECCV)*, 2018, pp. 767–783.
- [11] Y. Yu and W. A. Smith, "InverseRenderNet: Learning single image inverse rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [12] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 29:1–29:13, 2013.
- [13] N. Alldrin, T. Zickler, and D. Kriegman, "Photometric stereo with non-parametric and spatially-varying reflectance," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
- [14] D. B. Goldman, B. Curless, A. Hertzmann, and S. M. Seitz, "Shape and spatially-varying brdfs from photometric stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 1060–1071, 2010.
- [15] T. Haber, C. Fuchs, P. Bekaer, H. P. Seidel, M. Goesele, and H. P. A. Lensch, "Relighting objects from image collections," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 627–634.
- [16] K. Kim, A. Torii, and M. Okutomi, "Multi-view inverse rendering under arbitrary illumination and albedo," in *Proc. European Conference on Computer Vision (ECCV)*, 2016, pp. 750–767.
- [17] J. Jeon, S. Cho, X. Tong, and S. Lee, "Intrinsic image decomposition using structure-texture separation and surface normals," in *Proc. European Conference on Computer Vision (ECCV)*, 2014, pp. 218–233.
- [18] F. Romeiro, Y. Vasilyev, and T. Zickler, "Passive reflectometry," in *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2008, pp. 859–872.
- [19] S. Lombardi and K. Nishino, "Reflectance and illumination recovery in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 129–141, 2015.
- [20] G. Oxholm and K. Nishino, "Shape and reflectance estimation in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 376–389, 2015.
- [21] S. Lombardi and K. Nishino, "Radiometric scene decomposition: Scene reflectance, illumination, and geometry from rgb-d images," in *Proc. International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 305–313.
- [22] F. Romeiro and T. Zickler, "Blind reflectometry," in *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2010, pp. 45–58.
- [23] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. International Conference on 3D Vision (3DV)*, 2016, pp. 239–248.
- [24] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille, "Towards unified depth and semantic prediction from a single image," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2800–2809.
- [25] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5162–5170.
- [26] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, "Multi-scale continuous CRFs as sequential deep networks for monocular depth estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2017.
- [27] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2002–2011.
- [28] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [29] W. Chen, Z. Fu, D. Yang, and J. Deng, "Single-image depth perception in the wild," in *Proc. Advances in neural information processing systems (NIPS)*, 2016, pp. 730–738.
- [30] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *Proc. International Conference on Computer Vision (ICCV)*, 2017, pp. 66–75.
- [31] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *Proc. European Conference on Computer Vision (ECCV)*, 2016, pp. 740–756.
- [32] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [33] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, no. 6, 2017, p. 7.
- [34] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "SfM-net: Learning of structure and motion from video," *arXiv preprint arXiv:1704.07804*, 2017.
- [35] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2022–2030.
- [36] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth prediction," in *Proc. International Conference on Computer Vision (ICCV)*, October 2019.
- [37] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik, "Multi-view supervision for single-view reconstruction via differentiable ray consistency," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, no. 2, 2017, p. 3.
- [38] M. Ji, J. Gall, H. Zheng, Y. Liu, and L. Fang, "SurfaceNet: an end-to-end 3d neural network for multiview stereopsis," in *Proc. International Conference on Computer Vision (ICCV)*, 2017, pp. 2307–2315.
- [39] T. Narihira, M. Maire, and S. X. Yu, "Direct intrinsics: Learning albedo-shading decomposition by convolutional regression," in *Proc. International Conference on Computer Vision (ICCV)*, 2015, pp. 2992–2992.
- [40] G. Han, X. Xie, J. Lai, and W.-S. Zheng, "Learning an intrinsic image decomposer using synthesized rgb-d dataset," *IEEE Signal Processing Letters*, vol. 25, no. 6, pp. 753–757, 2018.
- [41] S. Bi, N. K. Kalantari, and R. Ramamoorthi, "Deep Hybrid Real and Synthetic Training for Intrinsic Decomposition," in *Eurographics Symposium on Rendering - Experimental Ideas & Implementations*, W. Jakob and T. Hachisuka, Eds. The Eurographics Association, 2018.
- [42] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf, "Revisiting deep intrinsic image decompositions," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8944–8952.
- [43] Z. Li and N. Snavely, "Learning intrinsic image decomposition from watching the world," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [44] —, "CGIntrinsics: Better intrinsic image decomposition through physically-based rendering," in *Proc. European Conference on Computer Vision (ECCV)*, 2018.
- [45] W.-C. Ma, H. Chu, B. Zhou, R. Urtasun, and A. Torralba, "Single image intrinsic decomposition without a single intrinsic image," in *Proc. European Conference on Computer Vision (ECCV)*, 2018, pp. 201–217.

- [46] A. S. Baslamisli, H.-A. Le, and T. Gevers, "CNN based learning using reflection and retinex models for intrinsic image decomposition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [47] E. Shelhamer, J. T. Barron, and T. Darrell, "Scene intrinsics and depth from a single image," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2015, pp. 37–44.
- [48] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samarasinghe, "Neural face editing with intrinsic image disentanglement," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5444–5453.
- [49] S. Sengupta, A. Kanazawa, C. D. Castillo, and D. W. Jacobs, "SfSNet: Learning shape, reflectance and illuminance of faces 'in the wild'," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [50] T. Nestmeyer, I. Matthews, J.-F. Lalonde, and A. M. Lehrmann, "Structural decompositions for end-to-end relighting," *arXiv preprint arXiv:1906.03355*, 2019.
- [51] Y. Kanamori and Y. Endo, "Relighting humans: occlusion-aware inverse rendering for full-body human images," *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia)*, vol. 37, no. 6, 2018.
- [52] M. Aittala, T. Aila, and J. Lehtinen, "Reflectance modeling by neural texture synthesis," *ACM Trans. Graph.*, vol. 35, no. 4, p. 65, 2016.
- [53] D. Gao, X. Li, Y. Dong, P. Peers, K. Xu, and X. Tong, "Deep inverse rendering for high resolution SVBRDF estimation from an arbitrary number of images," *ACM Trans. Graph.*, vol. 37, no. 4, July 2019.
- [54] X. Li, Y. Dong, P. Peers, and X. Tong, "Modeling surface appearance from a single photograph using self-augmented convolutional neural networks," *ACM Trans. Graph.*, vol. 36, no. 4, p. 45, 2017.
- [55] J. Philip, M. Gharbi, T. Zhou, A. Efros, and G. Drettakis, "Multi-view relighting using a geometry-aware network," *ACM Trans. Graph. (Proceedings of SIGGRAPH)*, vol. 38, no. 4, July 2019. [Online]. Available: <http://www.sop.inria.fr/reves/Basilic/2019/PgzED19>
- [56] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, "Deep convolutional inverse graphics network," in *Proc. Advances in neural information processing systems (NIPS)*, 2015, pp. 2539–2547.
- [57] M. Janner, J. Wu, T. D. Kulkarni, I. Yildirim, and J. Tenenbaum, "Self-supervised intrinsic image decomposition," in *Proc. Advances in neural information processing systems (NIPS)*, 2017, pp. 5936–5946.
- [58] G. Liu, D. Ceylan, E. Yumer, J. Yang, and J.-M. Lien, "Material editing using a physically based rendering network," in *Proc. International Conference on Computer Vision (ICCV)*, 2017, pp. 2261–2269.
- [59] Z. Li, Z. Xu, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker, "Learning to reconstruct shape and spatially-varying reflectance from a single image," in *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia)*, 2018, p. 269.
- [60] 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. International Conference on Computer Vision (ICCV)*, 2019.
- [61] 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. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2475–2484.
- [62] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [63] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890.
- [64] G. H. Golub and V. Pereyra, "The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate," *SIAM Journal on numerical analysis*, vol. 10, no. 2, pp. 413–432, 1973.
- [65] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.
- [66] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 649–666.
- [67] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. European Conference on Computer Vision (ECCV)*, 2016, pp. 694–711.
- [68] R. O. Dror, T. K. Leung, E. H. Adelson, and A. S. Willsky, "Statistics of real-world illumination," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.
- [69] H. Labs, *sIBL Archive*, 2007–2012. [Online]. Available: <http://www.hdrilabs.com/sibl/archive.html>
- [70] *HDRI-Skies*, 2020. [Online]. Available: <https://hdri-skies.com/>
- [71] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [72] T. Nestmeyer and P. V. Gehler, "Reflectance adaptive filtering improves intrinsic image estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, no. 3, 2017, p. 4.
- [73] S. Bell, K. Bala, and N. Snavely, "Intrinsic images in the wild," *ACM Trans. Graph. (Proceedings of SIGGRAPH)*, vol. 33, no. 4, 2014.
- [74] I. Vasiljevic, N. Kolkin, S. Zhang, R. Luo, H. Wang, F. Z. Dai, A. F. Daniele, M. Mostajabi, S. Basart, M. R. Walter, and G. Shakhnarovich, "DIODE: A Dense Indoor and Outdoor DDepth Dataset," *CoRR*, vol. abs/1908.00463, 2019. [Online]. Available: <http://arxiv.org/abs/1908.00463>
- [75] Y. Yu, A. Meka, M. Elgharib, H.-P. Seidel, C. Theobalt, and W. A. P. Smith, "Self-supervised outdoor scene relighting," in *Proc. European Conference on Computer Vision (ECCV)*, 2020.
- [76] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgb-d images," in *ECCV*, 2012.
- [77] J. Shi, Y. Dong, H. Su, and X. Y. Stella, "Learning non-lambertian object intrinsics across shapenet categories," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5844–5853.
- [78] D. A. Kleffner and V. S. Ramachandran, "On the perception of shape from shading," *Perception & Psychophysics*, vol. 52, no. 1, pp. 18–36, 1992.
- [79] G. Mather and R. Lee, "Turbine blade illusion," *i-Perception*, vol. 8, no. 3, p. 2041669517710031, 2017.
- [80] Y. Yu and W. A. P. Smith, "Depth estimation meets inverse rendering for single image novel view synthesis," in *Proc. of the ACM SIGGRAPH European Conference on Visual Media Production (CVMP)*, 2019.
- [81] C. Zou, A. Colburn, Q. Shan, and D. Hoiem, "Layoutnet: Reconstructing the 3d room layout from a single rgb image," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2051–2059.



Ye Yu received the B.Sc. degree in Computer Science from University of Science and Technology Beijing, China in 2010 and the M.Sc. degree in Visual Information Processing from the Imperial College London, UK in 2011. He is currently pursuing the Ph.D. degree at the University of York where he is a member of the Computer Vision and Pattern Recognition research group. His research interests include inverse rendering and image rendering.



William Smith received the BSc degree in computer science, and the PhD degree in computer vision from the University of York, York, United Kingdom. He is currently a Reader with the Department of Computer Science, University of York, York, United Kingdom. He holds a Royal Academy of Engineering/The Leverhulme Trust Senior Research Fellowship. His research interests are in shape and appearance modelling, model-based supervision and physics-based vision. He has published more than 100 papers in

international conferences and journals.