

Plausible Shading Decomposition For Layered Photo Retouching

Carlo Innamorati

Tobias Ritschel

Tim Weyrich

Niloy Mitra

University College London



Figure 1: Our approach automatically splits input images into layers motivated by light transport, such as (a): occlusion, albedo, irradiance and specular, or (b): the six major spatial light directions, which can then be manipulated independently using off-the-shelf photo manipulation software and recomposed back to an improved image. For (a) shadows were made deeper; albedo hue changed, saturation of irradiance increased and the specular was blurred for a more glossy material. For (b) The front lighting was made weaker and light from the left had been tinted red.

Abstract

Photographers routinely compose multiple manipulated photos of the same scene (layers) into a single image, which is better than any individual photo could be alone. Similarly, 3D artists set up rendering systems to produce layered images to contain only individual aspects of the light transport, which are composed into the final result in post-production. Regrettably, both approaches either take considerable time to capture, or remain limited to synthetic scenes. In this paper, we suggest a system to allow decomposing a single image into a *plausible shading decomposition* (PSD) that approximates effects such as shadow, diffuse illumination, albedo, and specular shading. This decomposition can then be manipulated in any off-the-shelf image manipulation software and recomposed back. We do so by learning a convolutional neural network trained using synthetic data. We demonstrate the effectiveness of our decomposition on synthetic (i.e., rendered) and real data (i.e., photographs), and use them for common photo manipulation, which are nearly impossible to perform otherwise from single images.

1 Introduction

Professional photographers regularly compose multiple photos of the same scene into one image, giving themselves more flexibility and artistic freedom than achievable by capturing in a single photo. They do so, by ‘decomposing’ the scene into individual *layers*, e. g., by changing the light, manipulating the individual layers (e. g., typically using a software such as Adobe Photoshop), and then composing them into a single image. On other occasions this process is called stacking. Unfortunately, this process requires the effort of setting up and taking multiple images. An alternative that overcomes this limitation is rendering synthetic images. In this case, the image can be clearly decomposed into the individual aspects of light transports (e. g., specular highlights vs. diffuse shading). The light path notation [Heckbert 1990] provides a strict criterion for this decomposition. Regrettably, this requires the image to be “synthesizable”, i. e., material, geometry, and illumination should be known as well as a suitable simulation algorithm. This is often not the case for scenes obtained as single images.

In this work, we set out to devise a system that combines the strength of both approaches: the ability to work on real photos, combined with a separation into light transport layers. Starting from a single photograph, our system produces a decomposition into layers, which can then be individually manipulated and recombined into the desired image using off-the-shelf image manipulation software. Fig. 1 shows an example.

While many decompositions are possible, we suggest a specific layering model that is inspired by how many artists as well as practical contemporary rendering systems (e. g., in interactive applications such as computer games) work: a decomposition into shadow, diffuse illumination, albedo, and specular shading. This model is not completely physical, but simple, intuitive for artists and its inverse model is effectively learnable. We formulate shadow as a single scalar factor to brighten or darken the appearance, resulting from adding the diffuse and specular shading. The diffuse shading is further decomposed into illumination (color of the light) and reflectance (color of the object), while the specular shading is modeled directly. To invert this model, we employ a deep convolutional architecture (CNN) that is trained using synthetic data, for which the ground truth-decomposition of a photo into light transport layers is known.

In summary, we make the following contributions:

- splitting and re-combination of images based on light transport layers (shadow, diffuse light, albedo and specular shading);
- a CNN trained on synthetic data to perform such a split; and
- evaluating our approach on a range of real photographs and demonstrating utility for photo-manipulations.

2 Previous Work

Combining multiple photos (also referred to as a “stack”) of a scene where one aspect has changed in each layer is routinely used in computer graphics [Cohen et al. 2003]. For example, NVIDIA IRay actively supports rendered LPE layers (light path expressions [Heckbert 1990]) to be individually edited to simplify post-processing towards artistic effects without resorting to solving the inverse rendering problem. One aspect to change is illumination, such as flash-

no-flash [Eisemann and Durand 2004] or exposure levels [Mertens et al. 2009]. More advanced effect involve direction of light [Akers et al. 2003; Rusinkiewicz et al. 2006; Fattal et al. 2007], eventually resulting in a more sophisticated user interface [Boyadzhiev et al. 2013]. All these approaches require specialized capture to gather multiple images captured by making invasive changes to the scene, limiting their use in practice to change an image post-capture. In fact, several websites and dedicated YouTube channels have emerged to provide DIY instructions to setup such studio configurations.

For single images, a more classic approach is to perform intrinsic decomposition into shading (irradiance) and diffuse reflectance (albedo) [Barrow and Tenenbaum 1978; Garces et al. 2012; Bell et al. 2014], possibly supported by a dedicated UI for images [Bousseau et al. 2009; Boyadzhiev et al. 2012], using annotated data [Bell et al. 2014], or videos [Ye et al. 2014; Bonneau et al. 2014]. Recently, CNNs have been successfully applied to this task producing state-of-the-art results [Narihira et al. 2015].

We also use a data-driven CNN-based approach to go beyond classic intrinsic image decomposition light transport layers with further separation into occlusion and specular components, that are routinely used when post-compositing layered renderings (see Sec. 4 and supplementary materials).

In other related efforts, researchers have looked into factorizing components, such as specular [Tan et al. 2004; Mallick et al. 2006] from single images, or ambient occlusion (AO) from single [Yang et al. 2015] or multiple captures [Hauagge et al. 2013]. We show that our approach can solve this problem at a comparable quality, but requires only a single photo and in combination yields further separation of diffuse shading and albedo without requiring a specialized method.

Despite the advances in recovering reflectance (e.g., with two captures and a stationarity assumption [Aittala et al. 2015], or with dedicated UIs [Dong et al. 2011]), illumination (e.g., Lalonde et al. [2009] estimate sky environment maps and Rematas et al. [2016] reflectance maps) and depth (e.g., Eigen et al. [2014] use a CNN to estimate depth) from photographs, no system doing a practical joint decomposition is known. Most relevant to our effort, is SIRFS [Barron and Malik 2015] that build data-driven priors for shape, reflectance, illumination, and use them in an optimization setup to recover the most likely shape, reflectance, and illumination under these priors (see Sec. 4 for explicit comparison to SIRFS).

In the context of image manipulations, specialized solutions exist: Oh et al. [2001] represent a scene as a layered collection of color and depth to enable distortion-free copying of parts of a photograph, and allow discounting effect of illumination on uniformly textured areas using bilateral filtering; Khan et al. [2006] enable automatically replacing one material with another (e.g., increase/decrease specularity, transparency, etc.) starting from a single high dynamic range image by exploiting our ‘blindness’ to certain physical inaccuracies; Carroll et al. [2011] achieve consistent manipulation of inter-reflections; or the system of Karsch et al. [2011] that combines many of the above into state-of-the art and compelling augmented image synthesis.

Splitting into light path layers is typical in rendering inspired by the classic light path notation [Heckbert 1990]. In this work, different from Heckbert’s physical $E(S/D)^*L$ formalism, we use a more edit-friendly factorization into shadow, diffuse light, diffuse material, and specular, instead of separating direct and indirect effects. While all the above works on photos, it was acknowledged that rendering beyond the laws of physics can be useful to achieve different artistic goals [Todo et al. 2007; Vergne et al. 2009; Ritschel et al. 2010; Richardt et al. 2014; Dong et al. 2015; Schmidt et al. 2015]. Our approach naturally supports this option, allowing users to freely change light transport layers, using any image-level software of their

choice, also beyond what is physically correct. For example, the StyLit system proposed by Fisser et al. [2016] correlates artistic style with light transport expressions. Specifically, it requires pixels in the image to be labeled with light path information, e.g., by rendering and aligning. Hence, it can take the output of our factorization to enable stylization single photographs without being restricted to rendered content.

3 Our Approach

Overview. Our system has three main components: (i) producing training data (Sec. 3.2); (ii) a convolutional neural network to decompose single images into light transport layers (Sec. 3.3); and an interactive system to manipulate the light transport layers before recomposing them into an image (Sec. 3.5).

The training data (Sec. 3.2) is produced by rendering a large number of 3D scenes into image tuples, where the first is the composed image, while the other images are the light transport layers. This step needs only to be performed once and the training data will be made available upon publication.

The layer decomposition (Sec. 3.3) is done using a CNN that consumes a photo and outputs all its light transport layers. This CNN is trained using the training data from the previous step. We selected a convolution-deconvolution architecture that is only to be trained once, can be executed efficiently on new input images, and its definition will be made publicly available upon publication (please refer to the supplementary for the architecture).

Optionally, we employ an upsampling step Sec. 3.4, that re-samples the fixed-resolution light transport layer CNN output, such that composing them in the arbitrarily high resolution of the original resolution produces precisely the original image without any bias, blur, or drift.

Finally, we suggest a system (Sec. 3.5) that executes the CNN on a photo at deployment time to produce light transport layers, which can then be individually and interactively manipulated, in any off-the-shelf image manipulation software, allowing operations on photos that previously were only possible on layered renderings, or using multiple captures.

Before detailing all the three steps, we will next introduce the specific image formation model underlying our framework in Sec. 3.1.

3.1 Model

We propose two different image formation models: one that is invariant under light direction and one that captures the directional dependency.

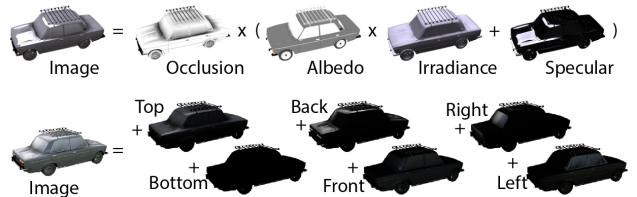


Figure 2: The components of our two imaging models.

Non-directional Model. We model the color C of a pixel as

$$C = O(\rho I + S), \quad (1)$$

where $O \in (0, 1) \in \mathbb{R}$ denotes the *occlusion*, which is the fraction of directions in the upper hemisphere that is blocked from the light; the variable $I \in (0, 1)^3 \in \mathbb{R}^3$ denotes the *diffuse illumination* (irradiance), i.e., color of the light, without any directional dependence; $\rho \in (0, 1)^3 \in \mathbb{R}^3$ describes the *albedo* (diffuse reflectance), i.e., the color of the surface itself; and finally, $S \in (0, 1)^3 \in \mathbb{R}^3$ is the *specular shading*, where we do not separate between the reflectance and the illumination, and do not capture any directional dependence.

Directional Model. The directional model is

$$C = \sum_{i=1}^N \mathbf{R} \left(\int_{\Omega} L_i(\omega) b_i(\omega) d\omega \right), \quad (2)$$

where L_i is the incoming light, and \mathbf{R} the reflection operator, mapping incoming to outgoing light. In other words, we express pixel color as a sum of reflections of n basis illuminations.

Here, (b_1, \dots, b_n) can be any set of spherical functions that sum to 1 at every direction, i.e., $\sum_i^n b_i(\omega) = 1$ (partition of unity). One such decomposition is the spherical harmonics basis of any order or the cube basis (that is one for a single cube face). In our approach, we suggest to use a novel *soft cube* decomposition that combines strengths of both: It is very selective in the directional domain, has finitely many components but also does not introduce a sharp cut in the directional domain. It is defined as the clamped dot product between the i -th cube side direction \mathbf{c}_i raised to a sharpening power $\sigma = 20$:

$$b_i(\omega) = \max(\langle \omega, \mathbf{c}_i \rangle, 0)^\sigma / \sum_{j=1}^6 \max(\langle \omega, \mathbf{c}_j \rangle, 0)^\sigma.$$

An additional benefit of the (soft) cube decomposition is, that it is, other than SH, strictly positive, facilitating loading of layers into applications that do not (well) support negative values, such as Photoshop. Other bases are possible in this framework, allowing to tailor it to specific domain, where a prior on light directions might exist (e.g., portrait photos).

There are many values of O , I , ρ and S to explain an observed color C , so the decomposition is not unique. Inverting this mapping from a single observation is likely to be impossible. At the same time, humans are clearly able to solve this task. One explanation can be that they rely on context, on the spatial statistics of multiple observations $c(\mathbf{x})$, such that a decomposition into light transport layers becomes possible. In other words, simply not all arrangements of decompositions are equally likely. As described next, we employ a CNN to learn this decomposition in a similar fashion.



Figure 3: Samples from our set of synthetic training data.

3.2 Training data

Training data comprises of synthetic images that show a random shape, with partially random reflectance shaded by random environment map illumination.

Shape. Shape geometry comprises of 300 random cars from from ShapeNet [Chang et al. 2015]. Note that the models were assumed to be upright. This class was chosen, as it presents both smooth surfaces as well as hard edges typical for mechanical objects. Note that our results show many classes very different from cars, such as fruits, statues, mechanical appliances, etc. Please note that we specifically restricted training to only cars to evaluate how the CNN generalizes to other object classes. Other problems like optical flow have been solved using CNNs on general scenes despite being trained on very limited geometry, such as training exclusively on chairs [Dosovitskiy et al. 2015].

Reflectance. Reflectance using the physically-corrected Phong model [Lafortune and Willem 1994], sampled as follows: The diffuse colors come directly from ShapeNet models. The specular component k_s is assumed to be a single color. A random decision is made if the material is assumed to be electric or dielectric. If it is electric, we choose the specular color to be the average color of the diffuse texture. Otherwise, we choose it to be a uniform random grey value. Glossiness is set as $n = 3.0^{10\xi}$, where ξ is a random value in $U[0, 1]$.

Illumination. Illumination is sampled from a set of 122 HDR environment maps in resolution 512×256 that have an uncalibrated absolute range of values but are representative for typical lighting settings: indoor, outdoor, as well as studio lights.

Rendering. After fixing shape, material, and illumination, we synthesize a single image from a random view (rotation is only about vertical). To compute C , we compute all components individually, and compose them according to Eq. 1. The occlusion term O is computed using screen-space occlusion [Ritschel et al. 2009]. The diffuse shading I is computed using pre-computed irradiance environment maps [Ramamoorthi and Hanrahan 2001]. Similarly, specular shading is the product of the specular color k_s selected according to the above protocol, and a pre-convolved illumination map for gloss level n . Diffuse albedo k_s is directly available in ShapeNet. While we could also try to infer the glossiness, it would not be clear how to use its non-linear effect with classic layering. No indirect illumination or local interactions are rendered. When learning the directional-dependent variant, we render six images, where the illumination was pre-convolved with the i -th decomposed illumination.

While this image synthesis is far from being physically accurate, it can be produced easily, systematically and for a very large number of images, making it suitable for learning the layer statistics. Overall we produce 100 k images in a resolution of 256×256 (ca. 10 GB) in 5 hours on a current PC with a decent GPU.

Units. Care has to be taken in what color space learned and training data is to be processed. As the illumination is HDR, the resulting image is an HDR rendering. However, as our input images will be LDR at deployment time, we need to match their range. To this end, automatic exposure control is used to map those values into the LDR range, by selecting the 0.95 luminance percentile of a random subset of the pixels and scale all values such that this value maps to 1. The rendered result C is stored after gamma-correction with $\gamma = 2.0$. All other components are stored in physically linear units ($\gamma = 1.0$) and are processed in physical linear units by the CNN and the end-application using the layers. Doing the final gamma-correction will consequently be up to the application using the layers later on (as shown in our edit examples).



Figure 4: Decomposition of input images (left) into the six directional layers (left) for different objects.

3.3 Decomposition

We perform decomposition using a CNN [Krizhevsky et al. 2012] trained using the data produced as described above. Input to the network is a single image such as a photograph. Output for the non-directional variant are the four images (occlusion, diffuse illumination, albedo, specular shading), the light transport layers, where occlusion is scalar and the others are three-vector-valued. Output for the directional variant, the output are 6 directional diffuse and specular layers, to be composed with one AO, and one albedo layer.

This design follows the convolution-deconvolution with crosslinks idea, resulting in an hourglass scheme [Ronneberger et al. 2015]. The network is fully-convolutional. We start at a resolution of 256×256 that is reduced down to 2×2 through stride-two convolutions. We then perform two stride-one convolutions to increase the number of feature layers in accordance to the required number of output layers (i.e. quadruple for the light transport layers, sextuple for the directional light layers). The deconvolution part of the network comprises of blocks performing a resize-convolution (upsampling followed by a stride-one convolution), crosslinking and a stride-one convolution. Every convolution in the network is followed by a ReLU [Nair and Hinton 2010] non-linearity except for the last layer, for which a Sigmoid non-linearity is used instead. This is done to normalize the output to the range $[0, 1]$. Images larger or smaller than the required input size of 256×256 will be appropriately scaled and/or padded to be square with white pixels. All receptive fields are 3×3 pixels in size except for the first and last two layers that are 5×5 .

As the loss function, we combine a per-light transport layer L2 loss with a novel three-fold *recombination* loss, that encourages the network to produce combinations that result in the input image and fulfills the following requirements: (i) the layers have to produce the input, so $C = AO(I_p + S)$; (ii) the components should explain the image without AO, i.e., $C/AO = I_p + S$; and (iii) diffuse reflected light should explain the image without AO and specular, so $C/AO - S = I_p$. If the network was able to always perform a perfect decomposition, the L2 loss alone would be sufficient. As it makes errors in practice, the second loss biases those errors to at least happen in such a way that the combined result does not deviate from the input. All losses are in the same RGB-difference range and were weighted equally for simplicity.

Overall, the network is a rather straight-forward modern design, but trained to solve a novel task (light transport layers) on novel kind of training data (synthesized, directionally-dependant information). We used TensorFlow [Chang et al. 2015] for our implementation platform and each model requires only several hours to train (both

have been trained for 12 hours). A more detailed description of the network's architecture can be found in the supplemental materials.

3.4 Upsampling

Upsampling (Fig. 5) is an optional step applied to input images that have a resolution arbitrarily higher than the one the CNN is trained on (256×256). Input to this process are all the layers in the low resolution. Output are the layers in that arbitrarily high resolution, such that applying the composition equation results in the high-resolution image.

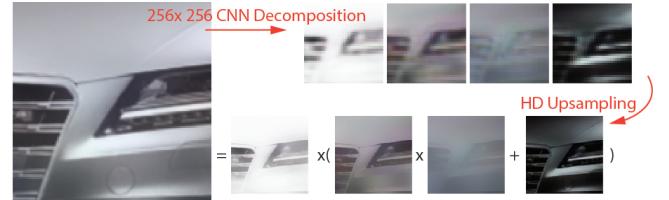


Figure 5: Our CNN computes a decomposition in a fixed resolution of 256×256 with the results (top insets). Given the HD original, we perform upsampling (bottom insets) that assures they combine to the HD input when blended.

This is achieved as follows, independently for every high-resolution pixel and in 100 iterations per pixel: Initially all values are set to the CNN output. At each iteration, we hold all layer components fixed, and in turn solve for the missing one, given the color. We then blend this result gradually (weight 0.001) with the previous result. This is repeated for all layers. Additionally, the light layer is forced to not change the chroma. When values leave the unit RGB cube, they are back-projected. The result is a layering in an arbitrary resolution that follows the CNN decomposition, yet produces the high resolution image precisely (energy-conserving). An immediate practical consequence of this is, that any image loaded into our system after the decomposition into layers looks precisely like the input without any initial bias (blur or color shift) introduced by the CNN processing. Please note that we explicitly mention upsampling for the results where we use this mode (only for edits).

3.5 Composition

For composition any arbitrary software that can handle layering, such as Adobe Photoshop and Adobe After Effects, can be used. We do not limit the manipulation to produce a composition that



Figure 6: $O \cdot (\rho \cdot I + S)$ editing (See text “Edit” Sec. 4).

is physically valid, because this is typically limiting the artistic freedom at this part of the pipeline [Todo et al. 2007; Ritschel et al. 2009; Schmidt et al. 2015]. Our decomposition is so simple that it can be implemented using a Photoshop macro that merely sets the appropriate additive and multiplicative blend modes, followed by a final gamma mapping. The content is then ready to be manipulated with existing tools with WYSIWYG feedback.

4 Results

We report results in form of typical decompositions on images, edits enabled by this decomposition, and numeric evaluation. The full supplemental material with many more decompositions is found at geometry.cs.ucl.ac.uk/projects/2017/PSD/results.php.

Decompositions. How well a network performs is best seen when applying it to real images. Regrettably, we do not know the reference light transport layer-decomposition or directional decomposition, so the quality can only be judged qualitatively. Therefore, results of decomposing images into light transport layers is seen in Fig. 8 while decomposition into directions is shown in Fig. 4.

Edits. Typical edits are shown in Fig. 6 and the directional variant in Fig. 7. Note that we support both global manipulations, such as

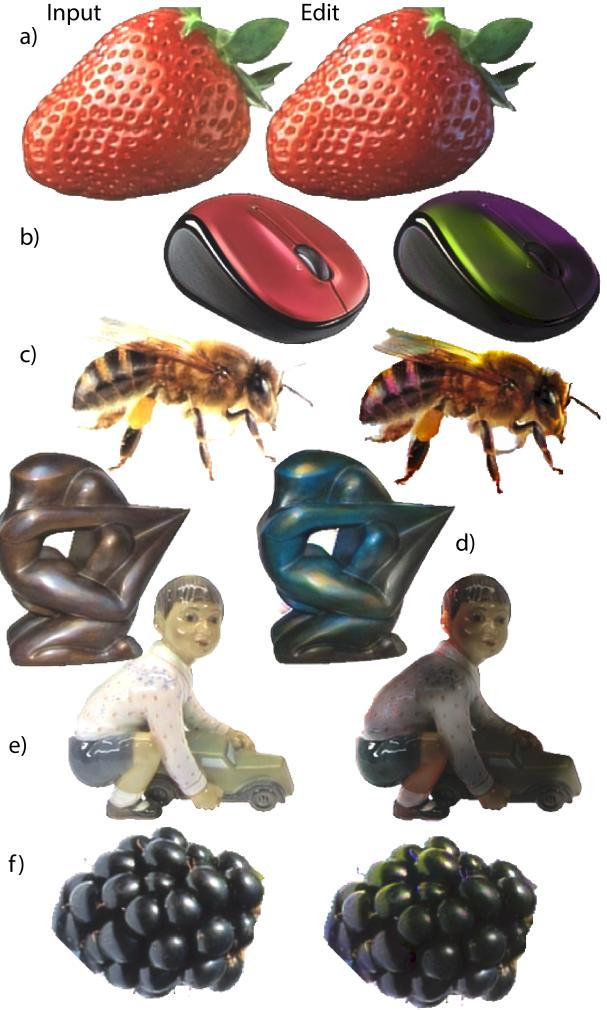


Figure 7: Directional editing (See text “Edit” Sec. 4).

changing the weight of all values in a layer, and local manipulations, such as blurring the highlights or albedo individually.

In Fig. 6, the first car (a) changes the albedo hue without affecting the highlight color. The second car (b) removed the diffuse part resulting in a very specular car. The banana (c) image shows increased highlights and deepened shadows. The first shoe image (d) was made more specular and the second (e) less, while also changing albedo hue and making shadows darker. Finally, the statue material (f) was changed to plaster by removing specular and setting albedo to identity, to bronze by removing diffuse shading and to yellow plastic by adjusting all components.

In Fig. 7 the first edit (a) changes the hue of the right color to blue. The back light on the mouse (b) was turned violet and the front light green. The bee (c) is lit more from the side with colored light. The bronze statue (d) is lit blueish from the left. The statue (e) is edited to be lit from the back. The strawberry illumination (f) was made colored from the top.

Numerical evaluation. Intrinsic images assume S to be zero (no specular) and combine our terms o and D , the occlusion and the diffuse illumination, into a single “shading” term that is separated



Figure 8: Decomposition of input images into light transport layers. Please see “Decomposition” in Sec. 4.

from the reflectance ρ .

$$C = O(\rho \cdot I + S) \approx O(\rho \cdot I + 0) = \underbrace{O \cdot I}_{\text{Shading } I'} \cdot \rho \quad (3)$$

A comparison on of our decomposition and typical approaches to generate intrinsic images is shown in Fig. 9. In table 1 we compare

against the same techniques but on our test dataset.

Limitations. Like in many CNN based learning approaches, the shortcoming of our two networks are hard to pin down. Not surprisingly they perform well on training data and generalizes reasonably across other object classes, still they fail when they see completely new type of data. While one obvious way to try to improve per-

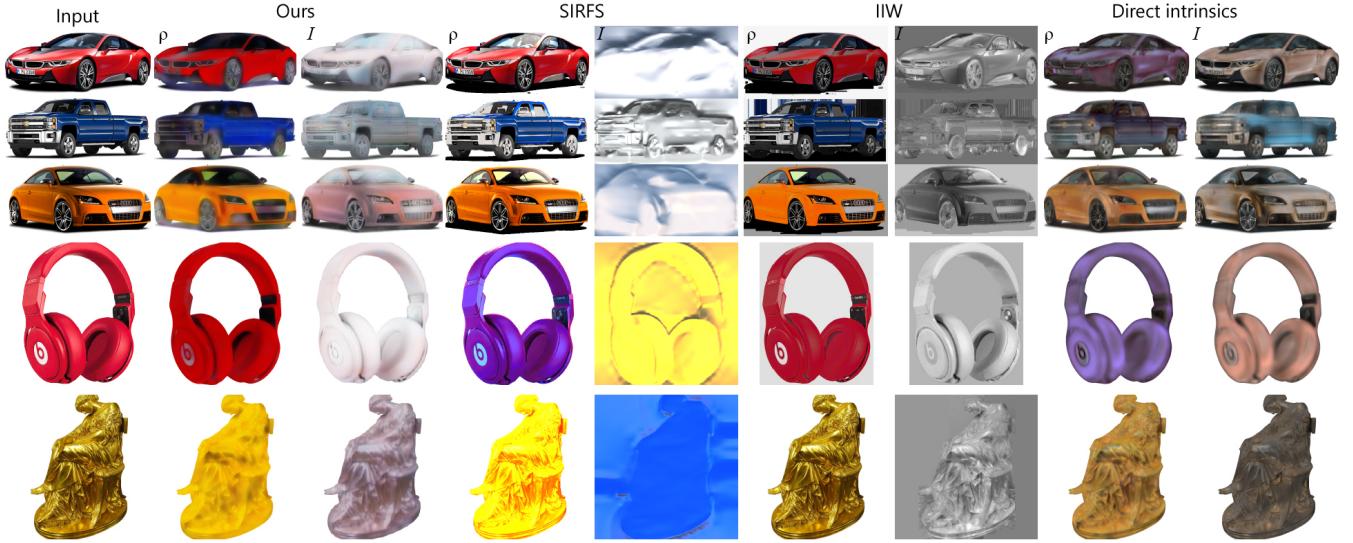


Figure 9: Comparison of our approach to three different reflectance and shading estimation techniques [Barron and Malik 2015], [Bell et al. 2014], [Narihira et al. 2015]. We run their method on real images and compare their results to ours.

Method	DSSIM	NRMSE
Ours	.0661 ± .0146	.3323 ± .1310
DI	.0862 ± .0165	.7698 ± .4818
IIW	.0775 ± .0158	.7698 ± .4594
SIRFS	.0846 ± .0187	1.315 ± 1.074

Table 1: Evaluation of our test dataset and other intrinsic image algorithms. We report the mean and standard deviation results of two well-known error metrics: DSSIM and NRMSE. We run the experiment on a batch of 100 examples from our test dataset comparing the ground truth albedo to our results and our competitors’.

formance would be to add more training data (e.g., different types of shape families, different illumination and materials, etc.) we would like to understand better what datasets to add to maximize improvement. This remains an elusive goal in CNN-based systems as of now.

5 Conclusion

We have suggested the first decomposition of general images into light transport layers, that were previously only possible either on synthetic images, or when capturing multiple images and manipulating the scene. We have shown that overcoming these limitations allows producing high-quality images, but it also saves capture time and removes the limitation to renderings. Future work could investigate other decompositions such as global and direct illumination, sub-surface-scattering or directional illumination or other inputs, such as videos.

References

- AITTALA, M., WEYRICH, T., AND LEHTINEN, J. 2015. Two-shot svBRDF capture for stationary materials. *ACM Trans. Graph. (Proc. SIGGRAPH)* 34, 4.
- AKERS, D., LOSASSO, F., KLINGNER, J., AGRAWALA, M., RICK, J., AND HANRAHAN, P. 2003. Conveying shape and features
- BARRON, J. T., AND MALIK, J. 2015. Shape, illumination, and reflectance from shading. *IEEE PAMI*.
- BARROW, H., AND TENENBAUM, J. 1978. Recovering intrinsic scene characteristics. *Comput. Vis. Syst.*.
- BELL, S., BALA, K., AND SNAVELY, N. 2014. Intrinsic images in the wild. *ACM Trans. Graph. (Proc. SIGGRAPH)* 33, 4, 159.
- BONNEEL, N., SUNKAVALLI, K., TOMPKIN, J., SUN, D., PARIS, S., AND PFISTER, H. 2014. Interactive intrinsic video editing. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 33, 6.
- BOUSSEAU, A., PARIS, S., AND DURAND, F. 2009. User-assisted intrinsic images.
- BOYADZHIEV, I., BALA, K., PARIS, S., AND DURAND, F. 2012. User-guided white balance for mixed lighting conditions. *ACM Trans. Graph.* 31, 6.
- BOYADZHIEV, I., PARIS, S., AND BALA, K. 2013. User-assisted image compositing for photographic lighting. *ACM Trans. Graph.* 32, 4.
- CARROLL, R., RAMAMOORTHI, R., AND AGRAWALA, M. 2011. Illumination decomposition for material recoloring with consistent interreflections. *ACM Trans. Graph.* 30, 4.
- CHANG, A. X., FUNKHOUSER, T. A., GUIBAS, L. J., HANRAHAN, P., HUANG, Q., LI, Z., SAVARESE, S., SAVVA, M., SONG, S., SU, H., XIAO, J., YI, L., AND YU, F. 2015. Shapenet: An information-rich 3d model repository. *CoRR abs/1512.03012*.
- COHEN, M. F., COLBURN, A., AND DRUCKER, S. 2003. Image stacks.
- DONG, Y., TONG, X., PELLACINI, F., AND GUO, B. 2011. AppGen: interactive material modeling from a single image.
- DONG, B., DONG, Y., TONG, X., AND PEERS, P. 2015. Measurement-based editing of diffuse albedo with consistent interreflections. *ACM Trans. Graph.* 34, 4.

- DOSOVITSKIY, A., FISCHERY, P., ILG, E., HAZIRBAS, C., GOLKOV, V., VAN DER SMAGT, P., CREMERS, D., AND BROX, T. 2015. Flownet: Learning optical flow with convolutional networks. In *Proc. ICCV*, 2758–66.
- EIGEN, D., PUHR SCH, C., AND FERGUS, R. 2014. Depth map prediction from a single image using a multi-scale deep network. In *Proc. NIPS*.
- EISEMANN, E., AND DURAND, F. 2004. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph. (Proc. SIGGRAPH)* 23, 3.
- FATTAL, R., AGRAWALA, M., AND RUSINKIEWICZ, S. 2007. Multiscale shape and detail enhancement from multi-light image collections. *ACM Trans. Graph.* 26, 3.
- FIŠER, J., JAMRIŠKA, O., LUKÁČ, M., SHECHTMAN, E., ASENTI, P., LU, J., AND SÝKORA, D. 2016. StyLit: Illumination-guided example-based stylization of 3D renderings. *ACM Trans. Graph. (Proc. SIGGRAPH)* 35, 4.
- GARCES, E., MUÑOZ, A., LOPEZ-MORENO, J., AND GUTIERREZ, D. 2012. Intrinsic images by clustering.
- HAUGAGE, D., WEHRWEIN, S., BALA, K., AND SNAVELY, N. 2013. Photometric ambient occlusion. In *IEEE CVPR*.
- HECKBERT, P. S. 1990. Adaptive radiosity textures for bidirectional ray tracing. *ACM SIGGRAPH Computer Graphics* 24, 4.
- KARSCH, K., HEDAU, V., FORSYTH, D., AND HOIEM, D. 2011. Rendering synthetic objects into legacy photographs. In *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, vol. 30.
- KHAN, E. A., REINHARD, E., FLEMING, R. W., AND BÜLTHOFF, H. H. 2006. Image-based material editing. *ACM Trans. Graph. (Proc. SIGGRAPH)* 25, 3.
- KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.
- LAFORTUNE, E. P., AND WILLEMS, Y. D. 1994. Using the modified phong reflectance model for physically based rendering.
- LALONDE, J.-F., EFROS, A. A., AND NARASIMHAN, S. G. 2009. Estimating natural illumination from a single outdoor image. In *Proc. ICCV*.
- MALLICK, S. P., ZICKLER, T., BELHUMEUR, P. N., AND KRIEGMAN, D. J. 2006. Specularity removal in images and videos: A PDE approach. In *ECCV*.
- MERTENS, T., KAUTZ, J., AND VAN REETH, F. 2009. Exposure fusion: A simple and practical alternative to high dynamic range photography. *Comp. Graph. Forum (Proc. PG)* 28, 1.
- NAIR, V., AND HINTON, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *Proc. ICML*, 807–14.
- NARIHIRA, T., MAIRE, M., AND YU, S. X. 2015. Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In *Proc. ICCV*.
- OH, B. M., CHEN, M., DORSEY, J., AND DURAND, F. 2001. Image-based modeling and photo editing. In *Proc. SIGGRAPH*.
- RAMAMOORTHI, R., AND HANRAHAN, P. 2001. An efficient representation for irradiance environment maps. In *Proc. SIGGRAPH*.
- REMATAS, K., RITSCHEL, T., FRITZ, M., GAVVES, E., AND TUYTELAARS, T. 2016. Deep reflectance maps. In *CVPR*.
- RICHARDT, C., LOPEZ-MORENO, J., BOUSSEAU, A., AGRAWALA, M., AND DRETTAKIS, G. 2014. Vectorising bitmaps into semi-transparent gradient layers. *Comp. Graph. Forum (Proc. EGSR)* 33, 4, 11–19.
- RITSCHEL, T., GROSCH, T., AND SEIDEL, H.-P. 2009. Approximating dynamic global illumination in image space. In *Proc. ISD*.
- RITSCHEL, T., THORMÄHLEN, T., DACHSBACHER, C., KAUTZ, J., AND SEIDEL, H.-P. 2010. Interactive on-surface signal deformation. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 29.
- RONNEBERGER, O., FISCHER, P., AND BROX, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Proc. Med. Image Comp. and Comp.-Assisted Int.*
- RUSINKIEWICZ, S., BURNS, M., AND DECARLO, D. 2006. Exaggerated shading for depicting shape and detail. *ACM Trans. Graph. (Proc. SIGGRAPH)* 25, 3.
- SCHMIDT, T.-W., PELLACINI, F., NOWROUZEZHARAI, D., JAROSZ, W., AND DACHSBACHER, C. 2015. State of the art in artistic editing of appearance, lighting and material. In *Comp. Graph. Forum*.
- TAN, R. T., NISHINO, K., AND IKEUCHI, K. 2004. Separating reflection components based on chromaticity and noise analysis. *IEEE PAMI* 26, 10.
- TODO, H., ANJYO, K.-I., BAXTER, W., AND IGARASHI, T. 2007. Locally controllable stylized shading. *ACM Trans. Graph. (Proc. SIGGRAPH)* 26, 3, 17.
- VERGNE, R., PACANOWSKI, R., BARLA, P., GRANIER, X., AND SCHLICK, C. 2009. Light warping for enhanced surface depiction. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 28, ACM.
- YANG, W., JI, Y., LIN, H., YANG, Y., BING KANG, S., AND YU, J. 2015. Ambient occlusion via compressive visibility estimation. In *Proc. CVPR*.
- YE, G., GARCES, E., LIU, Y., DAI, Q., AND GUTIERREZ, D. 2014. Intrinsic video and applications. *ACM Trans. Graph. (Proc. SIGGRAPH)* 33, 4, 80.