

# Unsupervised Deep Single-Image Intrinsic Decomposition using Illumination-Varying Image Sequences

Louis Lettry  
CVL, ETH Zürich

llettryl@vision.ee.ethz.ch

Kenneth Vanhoey  
CVL, ETH Zürich  
Unity Technologies

kenneth@research.kvanhoey.eu

Luc van Gool  
CVL, ETH Zürich  
PSI-ESAT, KU Leuven

vangool@vision.ee.ethz.ch

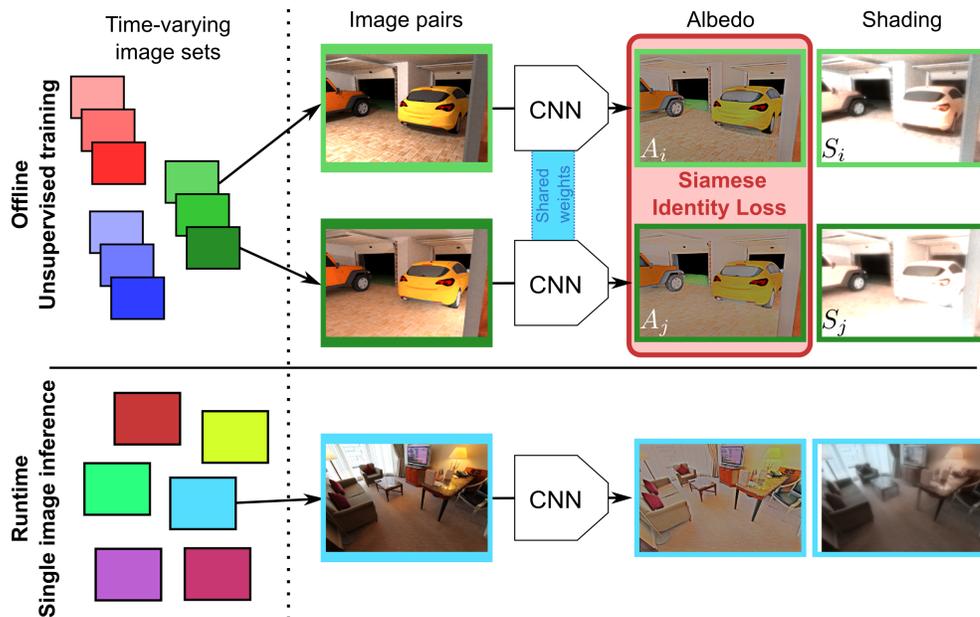


Figure 1: An end-to-end convolutional neural network (CNN) trained without ground truth supervision decomposes an input image into its albedo and shading images. In a pre-computation step (top), a unique CNN is trained by independently processing two images (taken from a time-varying sequence) so that a loss function can be expressed in a siamese manner on both their decompositions comparatively (*i.e.*, red background): it can learn from observing the changes in the input’s shading. At runtime (bottom), the trained CNN offers decomposition of previously unseen single images.

## Abstract

is not scalable to sufficient variety and realism. Machine learning based Single Image Intrinsic Decomposition (SIID) methods decompose a captured scene into its albedo and shading images by using the knowledge of a large set of known and realistic ground truth decompositions. Collecting and annotating such a dataset is an approach that cannot scale to sufficient variety and realism. We free ourselves from this limitation by training on unannotated images.

Our method leverages the observation that two images of the same scene but with different lighting provide useful in-

formation on their intrinsic properties: by definition, albedo is invariant to lighting conditions, and cross-combining the estimated albedo of a first image with the estimated shading of a second one should lead back to the second one’s input image. We transcribe this relationship into a siamese training scheme for a deep convolutional neural network that decomposes a single image into albedo and shading. The siamese setting allows us to introduce a new loss function including such cross-combinations, and to train solely on (time-lapse) images, discarding the need for any ground truth annotations.

As a result, our method has the good properties of *i*)

taking advantage of the time-varying information of image sequences in the (pre-computed) training step, ii) not requiring ground truth data to train on, and iii) being able to decompose single images of unseen scenes at runtime. To demonstrate and evaluate our work, we additionally propose a new rendered dataset containing illumination-varying scenes and a set of quantitative metrics to evaluate SIID algorithms. Despite its unsupervised nature, our results compete with state of the art methods, including supervised and non data-driven methods.

## 1. Introduction

Visual acquisition is the result of a complex process in which light travels through a scene and arrives on a sensor. This measured data is then post-processed (e.g., by a computer or our brain) to form an image representation. Intrinsic image decomposition is the inverse process in which one tries to recover intrinsic properties of a scene from a single 2D image representation. Many applications benefit from having access to this disentangled representation to allow for improved scene understanding, feature and shadow detection, stylization, relighting, object insertion and many more [4]. In this paper, we tackle the problem of separating a single image  $\mathcal{I}$  into an albedo image  $\mathcal{A}$  (i.e., a lighting and acquisition-independent representation of the scene) and its complementary image termed shading<sup>1</sup>  $\mathcal{S}$ .

Formally, Single-Image Intrinsic Decomposition (SIID) infers  $\mathcal{A}$  and  $\mathcal{S}$  from the input image  $\mathcal{I}$  such that the pixel-wise product  $\mathcal{I} = \mathcal{A} \cdot \mathcal{S}$  is respected [2] (see Fig. 1).  $\mathcal{A}$  is a color image whose pixels represent the base color or (diffuse) reflectance of what constitutes the acquired scene.  $\mathcal{S}$  is also an image whose pixels represent the alteration of the base color when the local illumination hits the surface and gets reflected towards the viewer.  $\mathcal{S}$  can be colored as well, due to colored illumination (e.g., due to sunset illumination) or inter-reflections on neighboring objects (see Fig. 3 and 5). As a result, SIID is severely ill-posed: there are twice more unknowns than knowns in the equation.

To guide an optimization procedure towards a desirable goal for this under-determined problem, one can define priors based on observation of a simplified Mondrian world (e.g.,  $\mathcal{A}$  is piecewise constant or  $\mathcal{S}$  is smooth [21]), reduce degrees of freedom (e.g.,  $\mathcal{S}$  is greyscale [13, 34, 3]) and/or provide additional input data (e.g., image depth [10]). On real-world use cases however, automatic SIID remains unsolved [4]. Machine learning offers a way to learn valid priors, rather than impose them. However, this requires annotated ground truth (GT) data (i.e., dense per-pixel triplets  $(\mathcal{I}, \mathcal{A}, \mathcal{S})$  in our case), which cannot be obtained through a

<sup>1</sup>The term shading refers to light-induced effects, yet we (abusively) also include acquisition-induced effects – like tone-mapping – in this term.

scalable process [13]. We avoid this constraint by training without GT supervision.

We note that observing time/illumination changes of a static scene provides useful information to guide the learning process. Indeed, albedo is invariant to changes in illumination. Hence, two images from the same scene taken at different times should be decomposed into the same albedo and varying shading. By setting up a siamese training scheme, in which two images get processed in parallel by the same convolutional neural network (CNN), we can express loss functions that encompass both their decompositions. This way, the network gets optimized based on the relationship of decompositions it produces for pairs of images (see Fig. 1, top). As a result, by observing pairs and adding a small regularization, our method learns how to process SIID in an unsupervised manner: it trains solely on (time-lapse) images, discarding the need for any GT annotation. To our knowledge, it is the first deep learning solution for SIID to be unsupervised thus to avoid training on datasets that are infeasible to annotate or to which one risks severe overfitting. Moreover, the CNN forms a feed-forward network that can process a single image of a previously unseen scene (see Fig. 1, bottom): it does not require multiple inputs at runtime.

In the following sections, we build, demonstrate and evaluate our novel method. More specifically:

- In section 3, we introduce a siamese training procedure that trains a CNN on pairs of images taken from a time-varying scene. This allows us to phrase the loss functions that rule the relationship between the estimated decompositions of both images. Despite the lack of GT supervision, we obtain results that compete with the state of the art methods.
- In section 5 we propose to complete the set of evaluation metrics used to benchmark SIID algorithms. We show that existing numerical benchmarks presented and used separately or jointly in different papers are insufficient to capture all desirable properties of SIID algorithms. Therefore, we build a set of metrics – two of which are new – and interpret what each one measures before comparing our results to related work using them.
- Additionally, in section 4, we propose a synthetic dataset that is generated using physically-based rendering. It is an extension of the recent SUNCG dataset [29], which we augment for unsupervised SIID by rendering static scenes under varying lighting conditions and tone mappings.

## 2. Related Work

Our interests lie with previous Intrinsic Image Decomposition (IID) methods (section 2.1), datasets used for training and evaluation of SIID algorithms (section 2.2) and evaluation metrics (which is discussed in section 5).

### 2.1. Intrinsic decomposition methods

Single-image IID is the process of decomposing an image  $\mathcal{I}$  into a dense albedo map  $\mathcal{A}$  and shading map  $\mathcal{S}$ . A recent survey by *Bonneel et al.* [4] reviews and compares related work on its usability for typical CG applications. The problem is severely ambiguous: twice as many unknowns as knowns have to be estimated. Providing more constraints is thus essential to guide an optimization scheme and reduce degrees of freedom.

#### 2.1.1 Reducing degrees of freedom

**Human-devised priors** based on observation (*e.g.*,  $\mathcal{A}$  is globally sparse and piecewise constant and  $\mathcal{S}$  is smooth [21]) have been used to regularize optimizations. Many derivatives exist [28, 12, 11, 1, 33]. Most generate decent decompositions on Mondrian-like images, but none generalize to the true complexity of photographed everyday scenes. We believe no human-devised priors can fully capture the complex reality, hence we prefer learning from data as much as possible.

**Time-varying input data.** Another source for disambiguation of the optimization problem is additional information: *e.g.*, user input [6, 5] or depth information [10]. Methods processing streams of images of varying viewpoints have been developed as [25] which uses temporal feedback to optimize on newly acquired frames in real-time. Weiss’ seminal paper [31] and derivatives [24, 19, 32] propose a method that takes multiple images depicting a static scene with temporal variation as input, and outputs a single (constant) albedo image along with a list of (varying) shading images, one per input image. While these works bear some similarity with ours, there is a fundamental difference in applicability and generalizability.

Their work is based on an iterative algorithm that optimizes a loss function with respect to a full sequence of input images. Therefore, the full sequence needs to be available at runtime, and the result is valid for this sequence only. We similarly use time-varying tuples (many pairs of varying scenes in our case) and optimize with respect to the decompositions of all inputs, but only in a precomputation step, *i.e.*, at train time. At inference time however, our learned model is applicable to *single images* of *unseen scenes*, hence forms a universal end-to-end SIID method.

**Resort to sub-problems.** Finally, other priors restrict the range of values for  $\mathcal{S}$  (*e.g.*, grayscale and/or  $\mathcal{S} \leq 1$ , which disallows specularities). The overwhelming majority of IID works make such choices [4]. Notably, [16] defines  $\mathcal{I} = \mathcal{A}\mathcal{D} + \mathcal{S}_p$ , where a diffuse  $\mathcal{D} < 1$  and the specular  $\mathcal{S}_p \in \mathbb{R}^+$  components are colored. Similarly, we use the most general and harder variant ( $\mathcal{I} = \mathcal{A} \cdot \mathcal{S}$ ,  $\mathcal{S} \in \mathbb{R}^{3+}$ ). We motivate this choice in section 3.1.

#### 2.1.2 Learning-based solutions

A nowadays popular and promising trend is to learn the correct priors from data, either explicitly before feeding them to classical optimizations (*e.g.*, leveraging a CRF) [3, 34] or implicitly in an end-to-end framework [30, 16, 22]. The largest problem in the task of SIID concerns the source of data to train on. Since albedo cannot be observed without light, observing GT albedo and shading separately is hardly possible in the real world, and all existing datasets have strong limitations (see Sec. 2.2). Supervised end-to-end learning solutions rely on synthetic datasets [30, 22, 16]. As a result they suffer from overfitting on too small or artificial, biased datasets [30], hindering generalizability to real-world photographs. New efforts have been made to alleviate this generalizability problem, as [15] who proposed a self-supervised approach in two steps: first a rendering network is trained in a supervised manner to estimate a shading given a normal map and a point light, then a decomposing network is trained through the rendering network and a self reconstruction loss to estimate the albedo, normal map and illumination of the input image. Others rely on sparse human annotations, followed by applying the classical hand-devised priors [3, 34]: the learned model is by definition human-centric thus may be limited in generalizability. Conversely, we propose the first unsupervised end-to-end deep learning solution that does not rely on any annotation.

## 2.2. Datasets for Intrinsic Image Decomposition

Datasets for IID allow for two things: training supervised learning-based methods on, and numerical evaluation. Such a dataset should preferably come with dense GT annotations, *i.e.*,  $(\mathcal{I}, \mathcal{A}, \mathcal{S})$  triplets, but this is expensive to create at best, resulting in a realism versus size/pixel density tradeoff. We discuss the capabilities and limitations of five datasets illustrated in Fig. 2(a-d).

#### 2.2.1 Realistic and Scarce

**MIT IID** [13] is the only GT dataset on real-world data. It contains 20 single-object scenes lit by 10 different illumination conditions. Dense (*i.e.*, pixelwise) decompositions were defined after a tedious acquisition process involving controlled light and paint-sprayed objects. Its small size

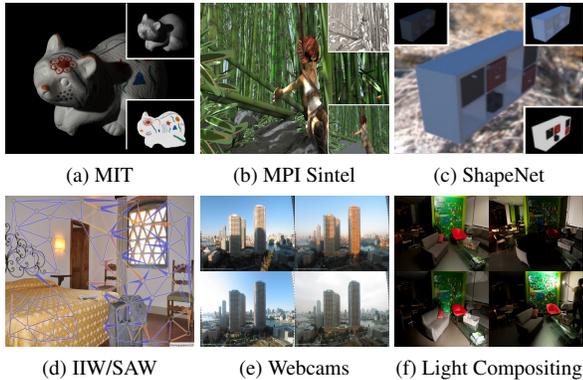


Figure 2: One sample per existing dataset. (a-c) is available with dense GT annotations (insets), (d) with sparse relative ones (arrows), and (e-f) possess no groundtruth.

and lack of variety makes it unusable for training convolutional neural networks (CNN) but provides a benchmark for evaluating object decompositions.

**Intrinsic Images in the Wild [3]** (IIW) introduces a dataset of 5,230 real-life indoor images with sparse reflectance annotations by humans, who were asked to compare (similar, greater or smaller than) albedo intensity (*i.e.*, grayscale level) of random point pairs in the images. This is taken as a sparse GT reference to measure the Weighted Human Disagreement Rate (WHDR) of SIID algorithms applied on the IIW images. Training a dense regression CNN is feasible but the sparse annotations provide insufficient cues (see Section. 5).

**Shading Annotations in the Wild [18]** (SAW) extends and complements IIW [3] with partly dense shading annotations by humans, who were asked to classify pixels as belonging to either smooth shadow areas or non-smooth shadow boundaries. It is taken as a semi-dense GT reference to measure the SAW quality of SIID algorithms applied on the SAW dataset.

We include all these measures in our panel of metrics measuring various aspects of SIID quality in section 5.

### 2.2.2 Synthetic and Dense

CG rendering algorithms allow for approaching photorealistic image quality while accessing and exporting intrinsic layers, like albedo: they offer dense GT by definition. However, creating a dataset that is completely realistic and covers the visual variety of the real world is impossible, since realistic rendering requires substantial expert human effort and computation time.

**MPI Sintel [8]** contains frames from 48 scenes (along with GT albedo and shading) of the Sintel CG short movie.

However, it is biased: non-realistic effects (*e.g.*, fluorescent fluids) and harmful modeling tricks (*e.g.*, shadow baking in the albedo) have been used, so training on it hardly generalizes to real-world images [30, 22].

**Non-Lambertian ShapeNet [16]** is closer to photorealism, but contains only single-objects, just like MIT IID. 25K ShapeNet [9] objects’ intrinsic layers were lit by 98 different HDR environment maps and rendered using Mitsuba [14], for a total of 2.4M training images.

## 3. Unsupervised siamese training

We wish to avoid the usage of human-devised priors as much as possible to guide a learned solution, which is why we believe data-driven approaches are adequate. Deep learning brings a renewed interest for solving the task of SIID, but as we have seen in section 2.2, supervised learning requires large-scale annotated datasets. We propose a first solution using unsupervised deep learning leveraging pairs of images in which only the shading has changed. In section 4 we detail our training data, while we here focus on the model and training goals.

### 3.1. Model & architecture

Our goal is to build a universal model that decomposes an image  $\mathcal{I}$  into an albedo  $\mathcal{A}$  and shading  $\mathcal{S}$  using an image-to-image regression CNN  $\mathcal{F}_c$  with parameters  $c$  such that  $(\mathcal{A}, \mathcal{S}) = \mathcal{F}_c(\mathcal{I})$ . For generality, we choose the albedo/shading decomposition following

$$\mathcal{I} = \mathcal{A} \cdot \mathcal{S}, \quad (1)$$

where  $\cdot$  denotes a per-pixel product,  $\mathcal{I}^{m \times n \times 3}$ ,  $\mathcal{A}^{m \times n \times 3} \in [0, 1]$  and  $\mathcal{S}^{m \times n \times 3} \in [0, \infty[$ .  $\mathcal{A}$  represents intrinsic colors, which we represent using the usual 3-channel RGB values. Unlike most related work, we allow  $\mathcal{S}$  to be colored and to grow beyond unit value. This allows to represent natural light phenomena, like colored lighting and bright highlights (see the cityscape or the red reflection on the yellow pepper in Fig. 3). Note that solving this problem is harder than many variants in which shading is grayscale (*i.e.*, single-channel), or disallows specularities (*i.e.*,  $\mathcal{S} \in [0, 1]$ ). Yet we wish our network to be universal for the SIID problem. In section 3.2, we slightly constrain this increased liberty to guide the training towards a viable solution.

We choose an architecture in which  $\mathcal{S}$  is regressed and  $\mathcal{A}$  is deduced by element-wise division<sup>2</sup>, which guarantees consistent results. It builds upon the latest trends in training image-to-image CNN’s and is summarized in Fig. 4. Our network is composed of an autoencoder with skip connections at every level. The data is downsampled (by maxpooling with a stride of size 2), respectively upsampled (by a

<sup>2</sup>we empirically observed the same behavior as [22], regressing  $\mathcal{S}$  and deducing  $\mathcal{A}$  produced better results.



Figure 3: Real-life photographs. Left: two views of the same scene at different times: the color variation resides in the lighting and acquisition process. Right: red light reflected onto the yellow pepper, and a specularity. Capturing these effects in  $\mathcal{S}$  requires it to be color-valued and without upper bound.

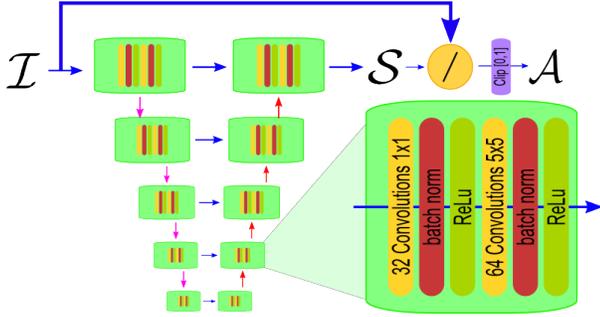


Figure 4: Our architecture is an autoencoder with skip connections (blue arrows), strided maxpooling (magenta arrows) and upscaling (red arrows).

bilinear interpolating upscaling operation of size 2 as well). Every level of the encoder and decoder is composed of a succession of 2 convolutions-batchnorm-ReLU layers: the first convolution applies a projection of the feature space with  $1 \times 1$  filters into a 32-dimensional space, while the second one has 64 filters of size  $5 \times 5$ . Finally, we use the element-wise division presented in [22] to enforce consistency of the decomposition. Note that we add a clipping layer so as to force  $\mathcal{A} \in [0, 1]$ . Since division is derivable, both  $\mathcal{A}$  and  $\mathcal{S}$  can be used in loss functions, allowing for backpropagation of errors on both components simultaneously. Training has been done with 2 siamese images (randomly taken from the same image sequence) in mini-batches of size six. We used the Adam [17] optimizer with a learning rate exponentially decreasing from  $10^{-3}$  to  $10^{-5}$  over 30k iterations, taking 22h on an NVidia GTX Titan X.

### 3.2. Siamese training losses

Following the assumption of static, time-varying scenes with illumination changes, a natural constraint arises: between images of the same view, only the shading is changing, and albedo is fixed. We implement this in a siamese training procedure in which we train a network on pairs of

images  $(\mathcal{I}_i, \mathcal{I}_j)$  taken from a time-varying image sequence  $\mathcal{T} = \{\forall i, \mathcal{I}_i\}$ . For each image  $\mathcal{I}_i$  a forward pass generates a decomposition pair  $(\mathcal{A}_i, \mathcal{S}_i)$  (cf. Fig. 1) and a joint loss is backpropagated. We next present the different components of this loss. Note that while training is siamese (*i.e.*, requires pairs), inference is done on single images (cf. Fig. 1, bottom).

**Albedo Similarity.** Our main training target states *the estimated albedo's of any two images of  $\mathcal{T}$  should be as close as possible*, in the  $L_2$  sense:

$$L_a = \|\mathcal{A}_i - \mathcal{A}_j\|_2^2, \text{ for } i, j \in \mathcal{T}. \quad (2)$$

Note that this is equivalent (up to a scaling factor) to the formulation stating that *the cross-product of estimated albedo  $\mathcal{A}_i$  and shading  $\mathcal{S}_j$  should be as close as possible to input image  $\mathcal{I}_j$* :

$$L_a = \|\mathcal{I}_j - \mathcal{A}_j \mathcal{S}_i\|_2^2, \text{ for } i, j \in \mathcal{T}, \quad (3)$$

which can be obtained by multiplying equation (2) by  $\mathcal{S}_i^2$ . Both formulations gave equivalent results in our experiments.

Nevertheless, it still leaves the problem under-determined. Without regularization, training will inevitably lead to local pitfalls: *e.g.*, all solutions of the form  $\mathcal{A} = \varepsilon$ ,  $\mathcal{S} = \mathcal{I}/\varepsilon$ ,  $\forall \varepsilon \in ]0, 1]$ . Hence we add a few regularizing terms.

**Shading Chromaticity Smoothness.** Unlike many SIID algorithms, our shading model is general: it allows for colored shading ( $\mathcal{S}$  is tristimulus) including specularities (*i.e.*,  $\mathcal{S}$  has no upper bound). This brings a lot of freedom, and our experience showed that without supervision, this can lead to over-colored shading and dull albedo's.

Shading strongly correlates with geometry (it can contain high frequencies if the geometry does): a shading smoothness loss [21] may thus be undesirable (see Fig. 5, top). The chromaticity however varies smoothly (Fig. 5, bottom) since light sources tend to be limited in number and colors, and are often distant. Our formulation ( $\mathcal{S} \in [0, +\infty[^3$ ) allows us to emphasize the regularization on the chromaticity. To do so, we convert the estimated shading in the CIE-Lab color space and limit the shading gradient of the ab dimension as follows:

$$L_c = \kappa \|\nabla \mathcal{S}_{ab}\|_2^2. \quad (4)$$

We empirically found that  $\kappa$ 's ideal range lies in  $[10, 100]$ , less the loss had basically no impact on the training and more tended to force the shading into a grayscale shading. We set it to be equal to 75, letting the network to favor mainly monochromatic shadings while allowing for colored light changes.

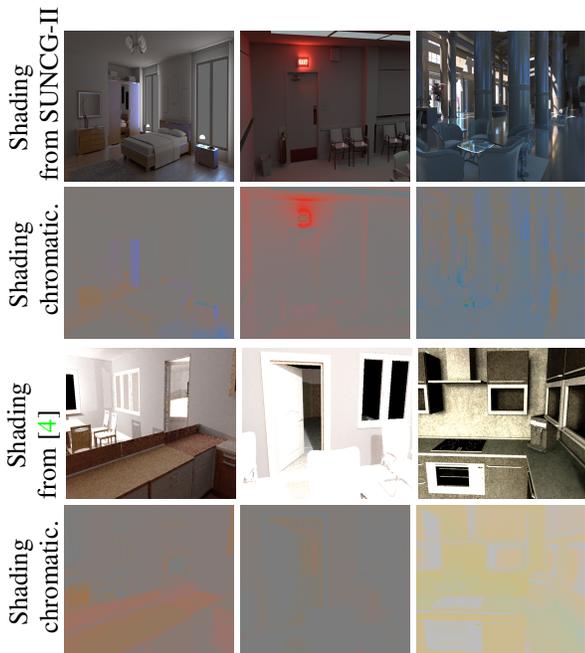


Figure 5: Ground-truth shadings obtained with physically-based rendering (odd rows) and its chroma-only image obtained by setting  $L = 50$  in the CIE-Lab color space (even rows). One can notice that the shadings’ chromaticities vary smoothly.

We still keep a small weight on overall shading smoothness in reaction to albedo luminance bleeding into the shading:

$$L_c = \lambda \|\nabla S_i\|_2^2. \quad (5)$$

where  $\lambda = 0.5$  is small so as to let the aforementioned losses take the lead in the optimization.

**Initialization.** To constrain the remaining degrees of freedom, we make the assumption that most of the albedo’s color and texture is well approximated by the actual (temporally-varying) input images. So we add a loss that encourages albedo to be close input images in the early training stages:

$$L_i = \mu \|\mathcal{I}_j - \mathcal{A}_i\|_2^2, \text{ for } i \neq j, i, j \in \mathcal{T}. \quad (6)$$

where  $\mu$  decreases linearly from 1 to 0.01 during the first 50% of the training, then remains fixed. This strongly initializes the model, while loosening it during training in favor of Eqn. (2).

Note that  $i \neq j$ : we favor proximity between image  $\mathcal{I}_j$  at time  $j$  and albedo  $\mathcal{A}_i$  at time  $i$ , hence the name “Temporal Regularization”. We also experimented with the simpler variant  $i = j$ , but this explicitly motivates the network to

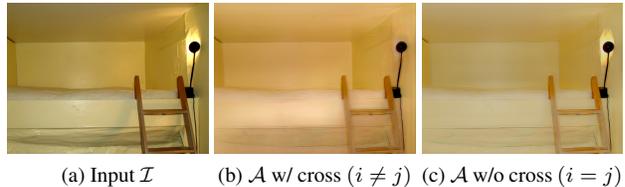


Figure 6: Albedo decompositions using the  $L_a$  term with and without temporal variation.

keep some shading coming from  $\mathcal{I}_i$  in its decomposition  $\mathcal{A}_i$ . The difference in shading between  $\mathcal{I}_j$  and  $\mathcal{I}_i$  prevents this undesired effect. Fig. 6 illustrates comparative results: much less shading spills into  $\mathcal{A}$  when using  $i \neq j$ .

**Reconstruction Consistency.** So as to obtain consistent decompositions (*i.e.*, multiplying  $\mathcal{S}$  and  $\mathcal{A}$  should produce  $\mathcal{I}$  again), following the argument of [22]. However, because of the clipping layer at the end of the network (see Fig. 4), and because we do not use any GT supervision as in [22], consistency could be lost during the optimization. Hence, we add a loss term to counter this:

$$L_r = \nu \|\mathcal{I}_i - \mathcal{A}_i \mathcal{S}_i\|_2^2, \forall i \in \mathcal{T}. \quad (7)$$

where  $\nu = 100$  to strongly discourage any deviation from equation (1) In practice, this loss reaches and stays close to 0 after 15% of our train time.

## 4. Timelapse Datasets

Training CNNs, especially deep ones, requires large-scale GT annotations. However, both synthetic CG rendering and crowdsourcing human annotations on real images are hardly scalable processes: they are not sufficiently realistic or dense and both require extensive and expert human intervention. As an alternative, we propose to work on an abundant data source: timelapses. We define timelapses as a collection of images acquired from a fixed viewpoint of a scene, with time-varying environmental parameters like weather. Under an assumption of staticity, they observe a constant albedo with different illuminations and acquisition processes (*e.g.*, tone-mapping).

Typical web cameras (see Fig. 3, left) form the target training data of our approach, for its ease of acquisition and realism. However, they often violate the staticity assumption. For example, the webcamclipart dataset [20] (Fig. 2(e)) contains 54 webcams that acquired several images per day over a year, and showed that elements may move (including the camera itself), and that weather (*e.g.*, fog, snow) changes the intrinsic albedo of the scene. We leave sanitization of this data for future work and create a



Figure 7: SUNCG-II: three variants (varying lighting and tone-mapping) of seven example scenes of our dataset.

new synthetically rendered dataset in which we have full control over these aspects.

#### 4.1. SUNCG-II

We present SUNCG-Intrinsic Images (SUNCG-II), a synthetic dataset that guarantees staticity to train on (cf. Fig. 7). It is an extension of the SUNCG dataset, which is a recent database of modeled apartments and houses introduced in [29]. Geometry, aspect of each surface, light parameters and preset interior viewpoints with full camera calibrations are included. This data serves as a base to render views (*i.e.*, a fixed scene acquired from a fixed viewpoint and intrinsic camera parameters) with physically-based path tracing using the Mitsuba renderer [14]. The primary objective of SUNCG was to obtain realistic GT interiors for different CV applications such as depth or normal estimation, semantic labeling, or scene completion.

We propose to adapt and augment this dataset to model several shading and image acquisition variants for each static scene and viewpoint. For each viewpoint, we randomly sample several variants in light sources, and post-process the images with several variants of tone-mapping. As a result, we created 7,892 views from 817 scenes, multiplied by 5 varying lighting conditions and 5 different tone-mappings, producing a total of 106,609 images (after removing around half of the images because they have too little light, *i.e.*, with mean intensity less than 20). It is to be noted that SUNCG comes with 45,622 scenes, thus we currently exploited only 1.8% of the available scenes.

For each image, we have also rendered the corresponding GT albedo and shading maps. We only use this GT for evaluation, and to compare how far our unsupervised training is off w.r.t. a supervised variant. While the dataset we created and will publish is composed of time-varying data with annotated GT, we emphasize that our unsupervised method

does not use the GT data.

While we think synthetic data should be avoided due its lack of realism, having access to a realistic timelapse rendering framework provides control to every parameters involved in the image generation process (*i.e.* scene, lighting, camera, etc). This is particularly useful to investigate aspects of the intrinsic decomposition that are often overlooked eventhough they play an important role as these alter the perception of intrinsic properties in a (a priori) uncontrollable manner. Especially, physical acquisition processes and post processing such as white balancing, tone-mapping, and many others, noticeably modify the acquired image from the original physical scene for perceptual and aesthetic reasons.

#### Technical details

To determine lighting and tone-mapping variants of each view, we use the following procedure. First, we remove every transparent object in the scene (*e.g.*, windows, vases) as they incur many rendering artifacts. Second, we remove any light source, including the environment maps. Third, we randomly add 1 to 3 point lights in a half cuboid of radius  $3 \times 1.5 \times 3$  scene units (in camera reference) in front of the camera. Finally, we render the scene and apply the post-processing tonemap operation [26] with parameters  $key \sim U(0.1, 0.6)$  and  $burn \sim U(0.0, 0.2)$ .

The photorealistic path-traced images ( $\mathcal{I}$ ) were rendered with 128 samples per pixel. Albedo maps ( $\mathcal{A}$ ) were rendered by fetching the material’s diffuse color/texture information only, while shading maps ( $\mathcal{S}$ ) were calculated by element-wise division:  $\mathcal{S} = \mathcal{I}/\mathcal{A}$ . Validity masks were also produced, discarding infinite depth points and black pixels (*i.e.*,  $\mathcal{A} = 0$ ): these pixels are ignored when training.

LMSE	MIT	Bonneel
[11]	8.28	5.31
[34]	6.12	<b>1.04</b>
[30]	5.92	1.38
[3]	5.59	1.43
<b>Our</b>	<b>3.01</b>	1.31
<b>Our superv.</b>	3.27	1.79

Table 1: LMSE ( $\times 10^{-2}$ ) of the state-of-the-art methods w.r.t. the GT-annotated datasets MIT [13] and Bonneel [4].

## 4.2. Experimental setting

For comparison and evaluation purposes, we trained several variants in training goals. We present them here along with the notations we will use in the results section. “Our” denotes the standard unsupervised version of this work, trained on SUNCG-II data. “Our supervised” is the supervised variant, trained summing two  $L_2$  norms (w.r.t. the GT from SUNCG-II) on  $\mathcal{A}$  and  $\mathcal{S}$ . “Our IIW” uses a different dataset (*i.e.*, IIW [3]) and its sparse annotations with augmentations [34]: we train for optimizing the WHDR score by supervising the training with the annotated sparse pixel relationships.

Finally, we will use the “light compositing” (L.C.) dataset [7] consisting of 6 scenes observed from a single viewpoint but different single-flashlight illuminations (cf. Fig. 2(f)). It is too small to be used for training, but forms an interesting dataset for comparing the consistency of albedo decompositions.

## 5. Results

There are many applications to IID both in computer graphics (*e.g.*, shading-preserving texture editing, shading-less histogram matching, stylization, relighting, object insertion) and computer vision (*e.g.*, scene understanding, robust feature detection for structure from motion, optical flow or segmentation, and shadow detection) [4]. Depending on the target application, one may have different qualitative expectations from a decomposition algorithm: *e.g.*, texture should be preserved in  $\mathcal{A}$ , or  $\mathcal{A}$  and  $\mathcal{S}$  should be strictly consistent, *i.e.*, enforce equation (1).

While several metrics [13, 3, 18] have been suggested to evaluate IID, it has been observed that none give the full picture [18]. Therefore, we now assemble and extend a set of metrics that covers many requirements of IID algorithms, *i.e.*:

- proximity to dense GT
- agreement with human judgments, and
- consistency of decomposition.

Our argument is that they are all necessary to give the full (or at least a wider) picture: no metric taken alone is sufficient to validate an IID algorithm.

Alongside quantitative measures, we present qualitative results so as to link numbers with visual quality on the recent realistic rendered scenes by Bonneel *et al.* [4] and on real images from Bell *et al.* [7]. We evaluate and compare using our full set of metrics and show that despite not being supervised, our method competes with state-of-the-art methods on reference-based measures, and surpasses them on consistency of decomposition. More detailed results are provided in an additional document.

## 5.1. Proximity to Dense Ground Truth

Ideally, decompositions should lean closely to true decompositions represented by dense GT. The most widely-used full-reference metric in IID is the Local Mean Squared Error (LMSE) [13, 30, 4].

First, we evaluate the learning capacity of our unsupervised siamese scheme (denoted “Our”) by comparing with a fully supervised training equivalent (denoted “Our supervised”) against SUNCG-II ground truth. Both use the same random 80/20 scene split on SUNCG-II so as to minimize view similarity between train and test data. We obtain LMSE errors of 1.16 and 1.14, respectively. Drawing conclusions from this is complex as trainings converge at a different pace and towards different goals. Nevertheless, it hints that learning from time-varying shading without ground truth is nearly as informative as training with ground truth data.

For comparing related work, we measure LMSE w.r.t. two small datasets having GT annotations in Tab. 1: the real-world MIT dataset [13] and the (close to) realistic CG dataset of Bonneel *et al.* [4]. MIT contains simple objects, with a handful albedo’s only, while Bonneel *et al.*’s dataset contains higher-frequency albedo details, closer to casual images. Fig. 8 shows qualitative results alongside GT decompositions. One can notice that the GT shadings are colored, as are ours.

Despite avoiding GT supervision, our method beats classical methods (*e.g.*, [11]) and leans close to those that use data-driven supervision [3, 34, 30] in Tab. 1. Note that our supervised variant is not much better on average. This may hint at a bias towards the training data domain, hindering generalization beyond it. Our method being unsupervised, it seems to suffer less from this flaw.

## 5.2. Agreement with Human Annotations

Large-scale crowdsourced human annotations on real images have been collected in the IIW [3] and SAW [18] papers, respectively. The corresponding metrics (*i.e.*, WHDR and SAW, see Sec. 2.2) measure the alignment of IID results with these annotations. The SAW metric evaluates the

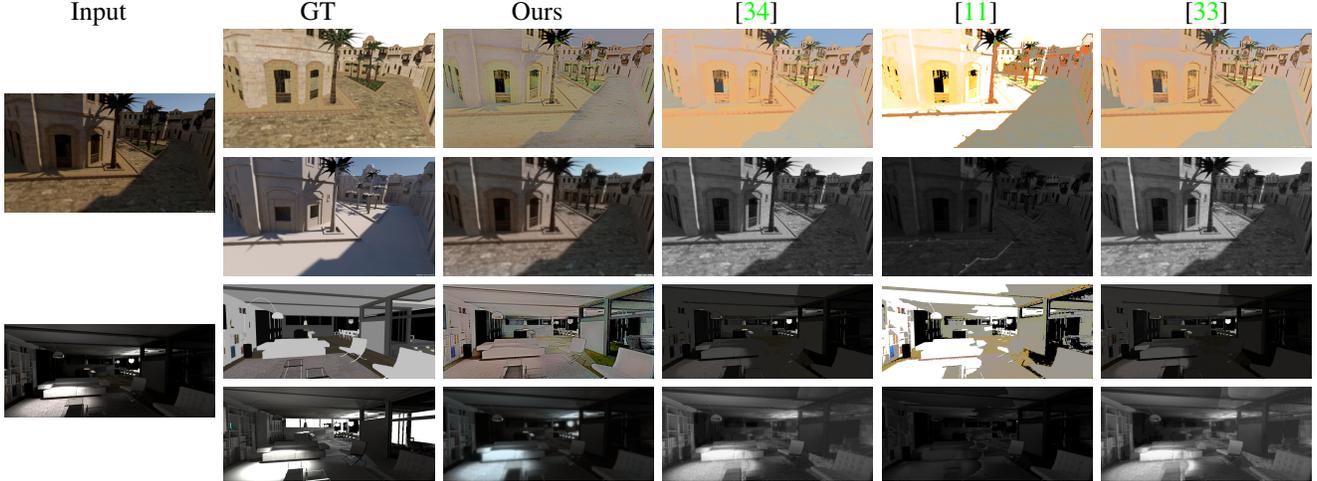


Figure 8: Decompositions on the GT dataset of Bonneel et al. [4].

Method	WHDR	SAW precision @		
		50%	70%	80%
Constant Albedo	36.5	82.7	82.2	78.7
[27]	24.0	90.0	78.4	70.1
[21]	23.5	93.4	87.5	77.3
[11]	22.6	95.8	84.7	75.4
[33]	23.2	<b>98.3</b>	90.2	80.4
[3]	<b>19.2</b>	97.8	88.9	79.1
[34]	20.1	97.8	92.9	80.3
[30]	40.7	89.2	80.9	73.9
[18]	N/A	93.8	84.5	DNC
<b>Our</b>	35.6	97.8	<b>95.3</b>	<b>88.3</b>
<b>Our superv.</b>	36.4	91.5	75.7	64.0

Table 2: WHDR [3] and SAW (precision at given recall values) [18] evaluation.

smoothness of the decomposed shading, while the WHDR is a sparse metric comparing the relative relationship between pairs of albedo intensity points (around 65% equality and 35% inequality). Despite being sparse, both metrics are the best there exist up to date, so we quantitatively measure albedo and shading quality using them. We recalculated all WHDR and SAW values following the fair protocol of [3] taking the best of two runs with and without sRGB to RGB conversion and using the train/test split of [34]. Hence our results differ from those presented in [34, 18].

WHDR and SAW results can be seen in Tab. 2. Our method lags behind on the WHDR but has an excellent SAW score, especially on high recall values. With the help of qualitative results in Figures 8, 9 and 10, this result can be interpreted as follows. Our decompositions preserve more

texture in the albedo (cf. the floor in Fig. 8 and the wooden texture behind the basket in Fig. 10). As a result, shadings show less texture, which benefits concordance with the SAW measure.

While this is a good property, it is also harmful for the WHDR: as noted by [3, 34], the WHDR measure does not account for the albedo’s high frequencies. Indeed, it is based on sparse human annotations: only low frequencies are accounted for. Moreover, chromaticity is also absent since annotations care about *albedo intensity*. As a result, a method that defines albedo as piece-wise constant (*e.g.*, by pushing texture-induced high frequencies to the shading) or greyscale can have an excellent WHDR. They tend to misrepresent the shading by including all textures: *e.g.*, [34] in Fig. 10.

We directly observed this in two ways too. First, when removing shading regularization (equations (4) and (5)) (which allows colored high frequencies into the shading) our WHDR score improves to 30.0 while the SAW precision at 80% recall dropped to 82.1. Second, we also tried overfitting to IIW annotations (see “Our IIW” in Section 4.2 and Tab. 2) using our architecture: the WHDR improves to 21.1, but at the cost of dramatic visual results and SAW measure (*e.g.*, 70.6 at 70% recall).

Nevertheless, despite WHDR being debatable as a metric, we acknowledge there is still too much residue of shading present in our estimated albedo, especially regarding hard shadows. This forms a limitation of our method and an incentive for future work.

### 5.3. Decomposition Consistency

Finally, we introduce two new metrics to measure IID requirements that are unattended to in prior work, but are nonetheless important for several applications.

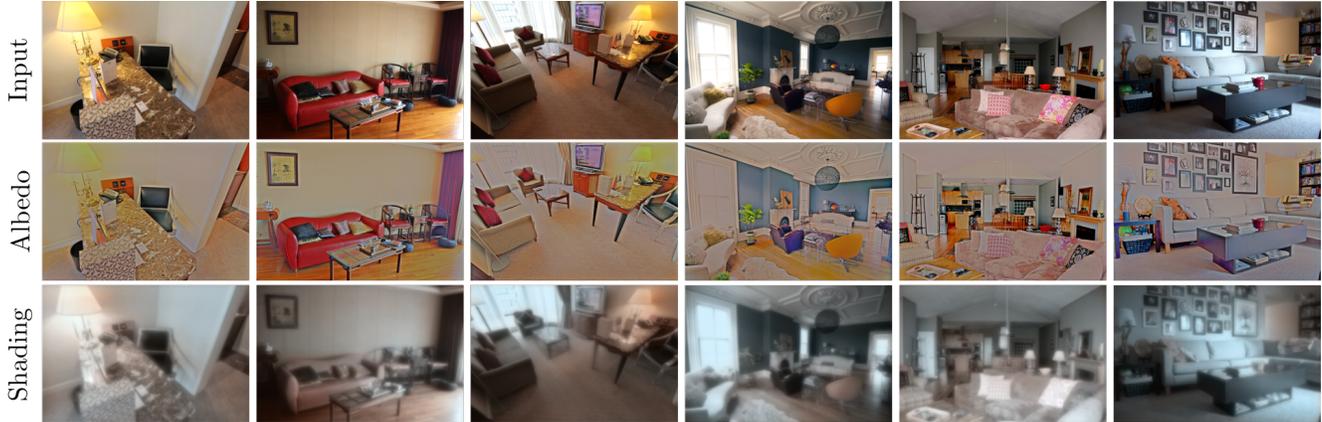


Figure 9: Decompositions using our CNN applied on the I1W dataset.

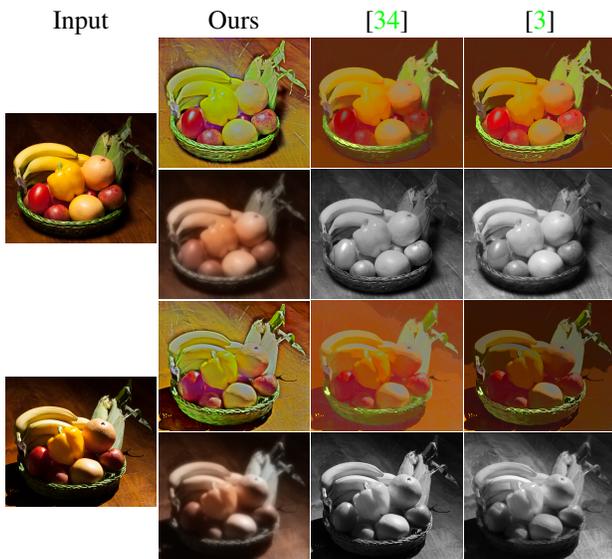


Figure 10: IID on two different lightings per scene on the Light Compositing dataset [7].

**Reciprocity Error** measures the loss of information occurring during the decomposition by comparing the original image ( $\mathcal{I}$ ) and the reconstructed one ( $\mathcal{A} \cdot \mathcal{S}$ ). This measure is of predominant importance for applications like image editing and object insertion. We define the *Mean Reconstruction Error* (MRE) as

$$\text{MRE}(\mathcal{I}) = \operatorname{argmin}_{\alpha} \sum \|\mathcal{I} - \alpha \mathcal{A} \cdot \mathcal{S}\|. \quad (8)$$

For fairness of comparison with other methods who export 8-bit quantized and rescaled results in image file formats, we proceed similarly. Moreover, we optimize for the rescaling parameter  $\alpha$  to virtually undo potential scalings needed for the 8-bit range quantization or for visualization. Tiny

errors remain due to quantization: when training done and converged, our method should have zero MRE.

**Temporal Inconsistency** measures how much the albedo decompositions of a set of images capturing the same static scene under different (lighting) conditions differ. This is important for robust shadow detection, relighting and any illumination varying (*i.e.*, video) application. Let  $\mathcal{T}^{\mathcal{A}} = \{\forall i, \mathcal{A}_i\}$  be a set of albedo decompositions of such an image sequence  $\mathcal{T}$ . We define the *Mean Albedo Consistency Error* as

$$\text{MACE}(\mathcal{T}^{\mathcal{A}}) = \frac{1}{3\mathcal{P}|\mathcal{T}^{\mathcal{A}}|^2} \sum_{i,j \in \mathcal{T}^{\mathcal{A}}} \sum_c |\mathcal{A}_i^c - \mathcal{A}_j^c|, \quad (9)$$

where  $\mathcal{A}_i$  is the albedo decomposition of image  $\mathcal{T}_i$ , and the sum is normalized by the number of pixels per image  $\mathcal{P}$ , the number of ordered pairs  $(i, j)$ , which equals  $|\mathcal{T}^{\mathcal{A}}|^2$ .  $c$  runs over colors channels.

Additionally, we define  $\text{MACE}_t$ : a relaxation of MACE that does not consider dark pixels that have intensity values below a threshold  $t$ , and normalizes accordingly. That is because dark pixels in an input image give little information on what the valid decomposition should be, and algorithms typically have to guess such pixel colors by extrapolation. Hence,  $\text{MACE}_0$  evaluates extrapolation capabilities, but we also use another value ( $t = 10$ ) to assess on the more feasible pixels only. For  $t > 0$ , we exclude from the calculation any pair  $(\mathcal{A}_i, \mathcal{A}_j)$  whose non-dark pixels have small overlap area, *i.e.*, less than 20% of the full image.

Note that the MPRE [34] is similar, but defined on products of estimated  $\mathcal{A}$  and  $\mathcal{S}$  across temporal variation in a sequence. While elegant, we believe it is not adequate because albedo errors are weighed by shading intensity. This attenuates errors in underexposed areas (which we think is legitimate), but emphasizes those made in saturated areas, which

Method	MRE			MACE <sub>10</sub>		
	MIT	L.C.	SAW	MIT	L.C.	Webc.
[11]	9.07	15.90	24.86	9.62	DNC	26.37
[3]	2.28	2.84	1.53	35.85	40.87	27.76
[34]	1.87	2.32	1.57	29.51	40.23	24.21
[30]	9.32	11.71	18.42	30.61	30.31	25.26
<b>Our</b>	<b>0.12</b>	<b>0.28</b>	<b>0.34</b>	<b>8.59</b>	<b>17.54</b>	<b>11.51</b>
<b>Our supervised</b>	0.36	2.97	0.68	16.47	35.33	35.62

Table 3: MRE and MACE metrics (*i.e.*, mean pixel deviation in the range  $[0, 255]$ ). [11] did not converge (DNC) on most images of the L.C. sequences.

is arguable, especially with models where  $\mathcal{S} \in [0, +\infty[$ .

**Result Analysis.** Tab. 3 shows consistency results over two time-varying datasets, and the SAW images. Our method is lossless ( $MRE \approx 0$ , only tiny quantization errors remain) and best preserves temporal consistency (see Fig. 10) across the datasets observed. Surprisingly, it is more temporally consistent even on the narrow set of albedos of the MIT dataset. We believe this is because most methods produce an albedo whose average intensity is close to the input image, and this temporally varies a lot on the MIT and Light Compositing datasets. Conversely, our method generates consistent intensities, more independent from the lightness. Zhou *et al.* [34] performs similarly well with a small MRE, but lacks temporal consistency. While Garces *et al.* [11] performs well when its assumptions (*e.g.*, piecewise constant albedo) are satisfied (*i.e.*, on the MIT dataset), it does not generalize well to real-life situations: it is the lossiest method and does not produce temporally consistent results on the real-life cluttered scenes.

#### 5.4. Summary

In an effort to gather and concisely present the various properties of the different approaches, we propose to plot them on a spider chart as in Fig. 11. It visually pinpoints strengths and weaknesses for each method and, from an application point of view, it makes selecting which algorithm suits one’s needs easier. We made it by converting all the metrics to a percentage between 0 and 1, where 1 means no mistake or error (*i.e.*, perfect score). The LMSE and MRE have been raised to a power of 4 to improve the visualization by emphasizing the differences.

Our proposed metrics fill an important gap by measuring aspects of decomposition unmeasured before, and explain how lossy methods such as [11] can perform well on the WHDR or SAW metrics: neither of them penalizes the loss of high frequencies. Yet, consistency metrics should always be evaluated in the context of reference-based ones or vi-

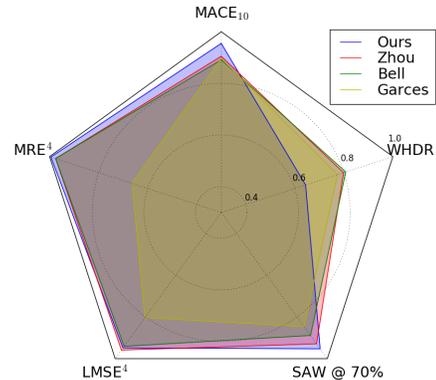


Figure 11: Performance on our five metrics. The MRE and LMSE are displayed with a power of 4 in order to help visualization.

sual results, since trivial solutions (*e.g.*,  $\mathcal{A} = \epsilon, \mathcal{S} = \frac{\mathcal{I}}{\epsilon}$ ) minimize the consistency metrics.

## 6. Conclusion & Limitations

We presented an unsupervised deep learning solution to the single-image intrinsic decomposition problem, which is a first to the best of our knowledge. It avoids training on datasets that are infeasible to annotate or to which one risks severe overfitting. In the pre-computation training step, our methodology takes advantage of the relationship between pairs of images of the same scene lit differently. A CNN gets trained by comparing the result of decomposing two images in parallel, and back-propagating the error to optimize its weights. Our new loss functions encompass cross-combinations of albedo and shading estimated from image pairs, so it can learn from seeing visual variation with lighting changes.

This allowed us to train the CNN without any kind of GT annotation on our new SUNCG-II dataset. One of our goals was to eliminate human-designed priors from the optimization goals, as we believe they cannot be general enough to solve the full problem. Analyzing image by pairs helped for this task, but there are still many degrees of freedom left, as the general problem we try to solve is severely underdetermined. Hence we resorted to a human-based prior to regularize the optimization (*i.e.*, equation (4), and equation (6) to a lesser extent). We see it as a long-term goal to get completely rid of it.

At runtime, the resulting CNN gets deployed on standalone images unseen before, and provides an albedo-shading decomposition. We evaluate on several SIID metrics, including newly proposed ones, so as to give a large evaluation panel users can choose from depending on their

target application. In general, results compete qualitatively and quantitatively with the state of the art methods, including those that require supervision. Unlike many methods, our results give high consistency guarantees, which makes our solution a good choice for applications that require consistency on large datasets, *e.g.*, robust feature detection for 3D reconstruction. Like many methods (see Fig. 8), hard shadows remain difficult to deal with.

We believe the metrics set we assembled brings a broader view on the strengths and weaknesses of each SIID method by evaluating several different or even orthogonal aspects of SIID quality (see Fig. 11). While defining an overall best method is tricky, this allows the end user to make a more informed choice on what method fits a target application best. However, linking our 5 quantitative axes with specific applications remains to be thoroughly studied. We leave this as an interesting application-oriented future work.

Finally, please note that a concurrent work [23] proposes a work similar to ours. It also uses timelapses to train CNNs for the task of intrinsic decomposition in an unsupervised manner. They focused on designing a spatio-temporal smoothness that can be efficiently applied to an arbitrary number of input images of the same scene and proposed a sanitized set of timelapses to use for training. We believe both our contributions open the path for many exciting works on SIID using unsupervised deep learning.

**Acknowledgments** The authors acknowledge financial support of the SNF grant “Wildtrack“ (CRSII2 147693/1) and hardware donation from NVIDIA.

## References

- [1] J. T. Barron and J. Malik. Shape, illumination, and reflectance from shading. *TPAMI*, 2015. 3
- [2] H. Barrow and J. Tenenbaum. Recovering intrinsic scene characteristics. *Comput. Vis. Syst., A Hanson & E. Riseman (Eds.)*, pages 3–26, 1978. 2
- [3] S. Bell, K. Bala, and N. Snavely. Intrinsic images in the wild. *ACM Trans. on Graphics (SIGGRAPH)*, 33(4), 2014. 2, 3, 4, 8, 9, 10, 11
- [4] N. Bonneel, B. Kovacs, S. Paris, and K. Bala. Intrinsic Decompositions for Image Editing. *Computer Graphics Forum (Eurographics State of the Art Reports 2017)*, 36(2), 2017. 2, 3, 6, 8, 9
- [5] N. Bonneel, K. Sunkavalli, J. Tompkin, D. Sun, S. Paris, and H. Pfister. Interactive intrinsic video editing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2014)*, 33(6), 2014. 3
- [6] A. Bousseau, S. Paris, and F. Durand. User assisted intrinsic images. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2009)*, 28(5), 2009. 3
- [7] I. Boyadzhiev, S. Paris, and K. Bala. User-assisted image compositing for photographic lighting. *ACM Trans. Graph.*, 32(4), July 2013. 8, 10
- [8] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012. 4
- [9] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 4
- [10] Q. Chen and V. Koltun. A simple model for intrinsic image decomposition with depth cues. In *2013 IEEE International Conference on Computer Vision*, pages 241–248, Dec 2013. 2, 3
- [11] E. Garces, A. Munoz, J. Lopez-Moreno, and D. Gutierrez. Intrinsic images by clustering. *Computer Graphics Forum (Proc. EGSR 2012)*, 31(4), 2012. 3, 8, 9, 11
- [12] P. V. Gehler, C. Rother, M. Kiefel, L. Zhang, and B. Schölkopf. Recovering intrinsic images with a global sparsity prior on reflectance. In *Proceedings of the 24th International Conference on Neural Information Processing Systems, NIPS’11*, pages 765–773, USA, 2011. Curran Associates Inc. 3
- [13] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman. Ground-truth dataset and baseline evaluations for intrinsic image algorithms. In *International Conference on Computer Vision*, pages 2335–2342, 2009. 2, 3, 8
- [14] W. Jakob. Mitsuba renderer, 2010. <http://www.mitsuba-renderer.org>. 4, 7
- [15] M. Janner, J. Wu, T. Kulkarni, I. Yildirim, and J. B. Tenenbaum. Self-Supervised Intrinsic Image Decomposition. In *Advances In Neural Information Processing Systems*, 2017. 3
- [16] H. S. Jian Shi, Yue Dong and S. X. Yu. Learning non-lambertian object intrinsics across shapenet categories. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 4
- [17] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference for Learning Representations*, 2014. 5
- [18] B. Kovacs, S. Bell, N. Snavely, and K. Bala. Shading annotations in the wild. *Computer Vision and Pattern Recognition (CVPR)*, 2017. 4, 8, 9
- [19] P. Y. Laffont and J. C. Bazin. Intrinsic decomposition of image sequences from local temporal variations. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 433–441, Dec 2015. 3
- [20] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Webcam Clip Art: Appearance and illuminant transfer from time-lapse sequences. *ACM Transactions on Graphics (SIGGRAPH Asia 2009)*, 28(5), December 2009. 6
- [21] E. H. Land and J. J. McCann. Lightness and retinex theory. *JOSA*, 61(1):1–11, 1971. 2, 3, 5, 9
- [22] L. Lettry, K. Vanhoey, and L. Van Gool. DARN: A deep adversarial residual network for intrinsic image decomposition. In *2018 IEEE Winter Conference on Applications of*

- Computer Vision (WACV)*, pages 1359–1367, March 2018. [3](#), [4](#), [5](#), [6](#)
- [23] Z. Li and N. Snavely. Learning intrinsic image decomposition from watching the world. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. [12](#)
- [24] Y. Matsushita, S. Lin, S. B. Kang, and H. Shum. Estimating intrinsic images from image sequences with biased illumination. pages 274–286, April 2004. [3](#)
- [25] A. Meka, M. Zollhöfer, C. Richardt, and C. Theobalt. Live intrinsic video. *ACM Transactions on Graphics (Proceedings SIGGRAPH)*, 35(4), 2016. [3](#)
- [26] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda. Photographic tone reproduction for digital images. *ACM Trans. Graph.*, 21(3):267–276, July 2002. [7](#)
- [27] J. Shen, X. Yang, Y. Jia, and X. Li. Intrinsic images using optimization. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '11*, pages 3481–3487, Washington, DC, USA, 2011. IEEE Computer Society. [9](#)
- [28] L. Shen and C. Yeo. Intrinsic images decomposition using a local and global sparse representation of reflectance. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 697–704, June 2011. [3](#)
- [29] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. [2](#), [7](#)
- [30] M. M. Takuya Narihira and S. X. Yu. Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In *International Conference on Computer Vision (ICCV)*, 2015. [3](#), [4](#), [8](#), [9](#), [11](#)
- [31] Y. Weiss. Deriving intrinsic images from image sequences. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 68–75 vol.2, 2001. [3](#)
- [32] J. Yu. Rank-constrained pca for intrinsic images decomposition. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3578–3582, Sept 2016. [3](#)
- [33] Q. Zhao, P. Tan, Q. Dai, L. Shen, E. Wu, and S. Lin. A closed-form solution to retinex with nonlocal texture constraints. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(7):1437–1444, July 2012. [3](#), [9](#)
- [34] T. Zhou, P. Krähenbühl, and A. A. Efros. Learning data-driven reflectance priors for intrinsic image decomposition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3469–3477, 2015. [2](#), [3](#), [8](#), [9](#), [10](#), [11](#)