arXiv:1612.07899v1 [cs.CV] 23 Dec 2016

DARN: a Deep Adversial Residual Network for Intrinsic Image Decomposition

Louis Lettry, Kenneth Vanhoey, and Luc Van Gool

Computer Vision Lab, ETH Zurich, Switzerland

{lettryl,vanhoey,vangool}@vision.ee.ethz.ch

Abstract

We present a new deep supervised learning method for intrinsic decomposition of a single image into its albedo and shading components. Our contributions are based on a new fully convolutional neural network that estimates absolute albedo and shading jointly. As opposed to classical intrinsic image decomposition work, it is fully data-driven, hence does not require any physical priors like shading smoothness or albedo sparsity, nor does it rely on geometric information such as depth. Compared to recent deep learning techniques, we simplify the architecture, making it easier to build and train. It relies on a single end-to-end deep sequence of residual blocks and a perceptually-motivated metric formed by two discriminator networks. We train and demonstrate our architecture on the publicly available MPI Sintel dataset and its intrinsic image decomposition augmentation. We additionally discuss and augment the set of quantitative metrics so as to account for the more challenging recovery of non scale-invariant quantities. Results show that our work outperforms the state of the art algorithms both on the qualitative and quantitative aspect, while training convergence time is reduced.

1. Introduction

The image formation process is a complex phenomenon of light entering a scene, being transformed and reaching an observer. Barrow and Tenenbaum [2] define it as a mixture of the intrinsic scene characteristics like range, orientation, reflectance and illumination. Being able to reverse the process by decomposing an image into intrinsic components is a useful pre-process for many computer vision and graphics tasks. We use deep learning to tackle the task of intrinsic image decomposition, which aims at splitting an image composed of diffuse materials into the per-pixel product of diffuse albedo $\mathcal{A}$ (*i.e.*, base color) and shading $\mathcal{S}$ such that: $\mathcal{I} = \mathcal{A} \cdot \mathcal{S}$ (see Fig. 1). Albedo is a lighting-invariant quantity, while shading gives important cues on the light environment and object geometry and material. In computer graphics, intrinsic decomposition is at the base of shadow removal, and scene relighting or recoloring, which are all essential in augmented reality. Typical vision tasks like shape from shading, 3D reconstruction and depth or normal estimation benefit from a shading-only image, while segmentation conversely is mainly interested in the albedo image.

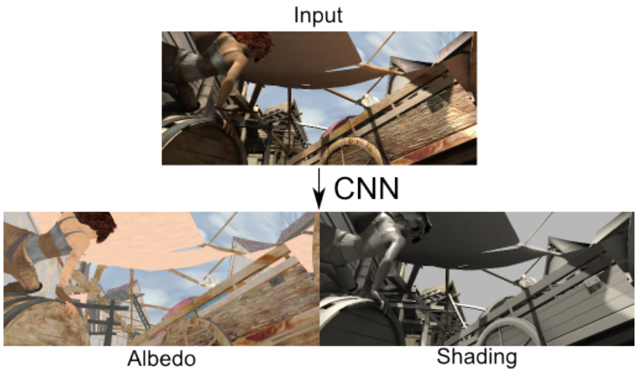

Figure 1: Intrinsic image decomposition separates $\mathcal{I}$ into the point-wise product of albedo $\mathcal{A}$ and shading $\mathcal{S}$. Albedo represents the scene's objects' base colors while shading includes all effects induced by lighting.

For many of these tasks, it is essential that the predicted decomposition is accurate. This is a hard task due to the inherent ambiguity on scale, influenced by unknown exposure at capture time. Therefore, the state of the art methods predict two quantities $\mathcal{A}$ and $\mathcal{S}$ regardless of scale, and both are separately evaluated on scale-invariant metrics [23]. The output thus requires manual tuning before being usable, and there is no guarantee on energy preservation: $\mathcal{I} \neq \mathcal{A} \cdot \mathcal{S}$. Finally, despite an elaborated multi-scale convolutional neural network (CNN) architecture that requires substantial training effort, qualitative results show improvement potential.

We argue that i) predictions should be energy-preserving, and ii) $\mathcal{A}$ being a color, it has an absolute value we should aim at. In this context, we propose a deep generative CNN that predicts albedo and shading jointly with increased accuracy and ensuring energy preservation by design, along with two new evaluation metrics that incorporate scale.

We design our network based on three key observations. First, we conjecture that predicting $\mathcal{A}$ and $\mathcal{S}$ separately requires additional training effort and is prone to inconsistencies that violate $\mathcal{I} = \mathcal{A}{\cdot}\mathcal{S}$, hence we learn them fully jointly and interdependently. Second, $\mathcal{I}$ is a good initial guess for both $\mathcal{A}$ and $\mathcal{S}$. Hence, inspired by residual networks [13], we learn a deviation from the input rather than from zero. Third, state of the art results show typical generative CNN artifacts like blur, color bleeding and contrast oscillations. We think they can be avoided by using a loss function in the form of a discriminator network [10]. We leverage these observations into the following contributions:

- a powerful and easier to train generative adversarial network (GAN) for fully joint learning of $\mathcal{A}$ and $\mathcal{S}$,
- a new state of the art in intrinsic image decomposition, outperforming previous work qualitatively and quantitatively on scale-invariant and scale-sensitive metrics,
- additional stricter quantitative metrics that incorporate scale, and that we discuss and motivate, and
- publicly available code provided upon publication.

2. Related Work

Intrinsic decomposition. The image formation process can be defined as the pixel-wise product of the albedo $\mathcal{A}$ defining a surface's base color, and the shading $\mathcal{S}$ defining the captured influence of light reflection and shadowing on each pixel: $\mathcal{I} = \mathcal{A} \cdot \mathcal{S}$ (see Fig. 1). Intrinsic decomposition is the process of estimating $\mathcal{A}$ and $\mathcal{S}$ from $\mathcal{I}$: see [1] for a comprehensive recent review. This is a hard ill-posed problem as there are twice as many unknowns than known observations. Hence, prior work focuses on adding constraints to produce a deterministic plausible solution.

In the single-image case, prior statistics on both $\mathcal{A}$ and $\mathcal{S}$ [14, 21, 25, 3, 1] and/or manual user intervention [5, 22] is used. Priors are typically the *retinex* assumptions [19]: slow variation (*i.e.*, smoothness) of $\mathcal{S}$, which varies with the normals in the scene, and sparsity of $\mathcal{A}$, which is generally unique per object or pattern in the scene. The mentioned methods work well in a Mondrian world [7], but fail on complex scenes, like outdoors.

Depth cues in the form of a 3D proxy or depth maps impose important local constraints on reflectance thus shading [20, 7]. However, depth either has to be measured alongside the image (*e.g.*, with a depth sensor) or deduced from multiple images with varying lighting [8] or viewpoints [18, 12]. Non-explicit depth can also be exploited by sophisticated reasoning on the variation in pixel color with varying lighting [17]. Conversely, we consider the image as the only available input.

The availability of larger datasets (see below) allowed deep learning techniques to be applied on the single-view case. [23] proposes a multi-scale CNN architecture composed of two branches so as to treat both global features and local details. These branches are merged into a single one, which is later split again to predict $\mathcal{A}$ and $\mathcal{S}$ separately. The resulting $\mathcal{A}$ and $\mathcal{S}$ are scale-invariant and not consistent: it violates $\mathcal{I} = \mathcal{A}{\cdot}\mathcal{S}$. This can cause problems to several applications, like normal estimation or shape from shading and may require prior manual correction. [26] use deep learning to infer data-driven priors from sparse human annotations. They then feed them into an existing energy minimization framework for intrinsic decomposition, leveraging a CRF to predict $\mathcal{A}$ and $\mathcal{S}$. They show good qualitative evaluation, but do not compare quantitatively.

We also propose a CNN, with some major changes that guarantee energy preserving consistent predictions that outperform the state of the art.

Deep learning. The training of CNN has greatly benefited from several architectural advancements we will take advantage of. Residual networks (ResNet) estimate the difference from the input instead of a full mapping from scratch [13]. Because $\mathcal{A}$ (resp. $\mathcal{S}$) and $\mathcal{I}$ have a lot in common, we argue that this is beneficial to our problem.

Finally, generative adversarial networks (GAN) append a discriminator network to a generator network [10]. The role of the former is to distinguish between a generated output and a ground truth (in our case albedo or shading). It is jointly trained with the generator, who tries to fool the discriminator by producing indistinguishable outputs. In that sense, a discriminator can be seen as a perceptual loss function. Exploiting this allows us to outperform prior work which suffer from typical CNN artifacts like blurriness, color bleeding and contrast oscillations.

Datasets. Data-driven approaches [17, 23, 26] need to learn priors during training, either explicitly or implicitly. Databases can be leveraged for that. However, acquiring ground truth albedo and shading images in the real world is difficult: it requires a precise controlled white and uniform light environment, hence is not scalable to outdoor scenes.

The MIT intrinsic images dataset focuses on single (segmented) objects [11]. Due to the small number of elements in the dataset (*i.e.*, 20 objects), training a CNN on it would overfit to it and not generalize well. [17] focuses on a single synthetic outdoor building under multiple viewpoints and lighting conditions. Conversely, *Intrinsic images in the*

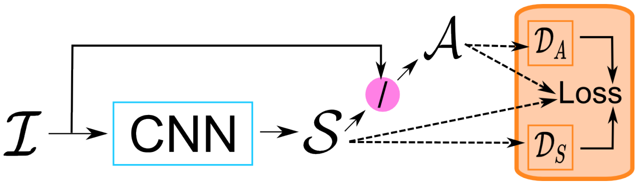

Figure 2: Global summary of the proposed GAN architecture. Element-wise division between shading and input (magenta) is used to define resulting albedo. Hence consistency is obtained by construction. The detailed architecture of the CNN (cyan box) is detailed in Fig 3.

wild [3] provide sparse user annotations on indoor scenes, from which albedo and shading can be deduced. Finally, the Sintel short animation movie has been made publicly available with its different rendering layers, including albedo and shading [6]. Despite its flaws (*e.g.*, mostly brownish colors, surrealistic effects like fluorescence), it is the best alternative at hand due to its size and precise ground truth. Hence, like related work, we train and evaluate on it (section 4), and show how the learned model applies on more realistic cases like the MIT intrinsic images in section 5.3.

3. Deep Intrinsic Decomposition

We decompose an image $\mathcal{I}(x) \in \mathbb{R}^3$ into albedo $\mathcal{A}(x) \in \mathbb{R}^3$ and shading $\mathcal{S}(x) \in \mathbb{R}^3$ following

$$\mathcal{I}(x) = \mathcal{A}(x) \cdot \mathcal{S}(x) \tag{1}$$

where $\cdot$ denotes the element-wise product, and x a pixel coordinate (that we will drop in all following notations). Note that we consider shading to have 3 channels to allow the modeling of colored illumination.

The decomposition problem is ill-posed: many possibilities for $\mathcal{A}$ and $\mathcal{S}$ respect the above equation, and the scale of $\mathcal{I}$ depends on unknown acquisition parameters, like exposure. Hence recent and deep learning methods predict "scale-invariant" components, *i.e.*, following $\mathcal{I} = \alpha\mathcal{A} \cdot \beta\mathcal{S}$. This requires a subsequent (often manual) optimization for both α and β *individually*, which is prone to introduce distortion and inconsistencies. They argue that some intrinsic decomposition applications do not require absolute nor relative consistent $\mathcal{A}$ and $\mathcal{S}$ [11].

Conversely, we argue that energy preservation is important to allow for a subsequent principled and joint normalization of $\mathcal{A}$ and $\mathcal{S}$, *i.e.*, tuning α following $\mathcal{I} = \alpha\mathcal{A} \cdot \mathcal{S}/\alpha$. Moreover, $\mathcal{A}$ can ideally be considered an absolute value that is invariant to illumination changes and the image formation process. Such a definition is beneficial for applications relying on an illumination invariant albedo like, *e.g.*, human face reconstruction. Thus we take this definition as training goal.

In section 3.1, we present the CNN structure that, by construction, allows us to predict consistent (respecting Eqn. (1)) $\hat{\mathcal{A}}$ and $\hat{\mathcal{S}}$, including scale. In section 3.2, we present the loss function we optimize for, including a perceptually motivated loss.

3.1. Network architecture

Formally, our learning task trains a generative fully convolutional network $\mathcal{N}_\phi$ having parameters ϕ at predicting a coherent pair $(\hat{\mathcal{A}}, \hat{\mathcal{S}})$ from an input image $\mathcal{I}$ such that

$$\mathcal{N}_\phi(\mathcal{I}) = (\hat{\mathcal{A}}, \hat{\mathcal{S}})\,. \tag{2}$$

Consistent predictions. We want to impose the strict respect of Eqn. (1) for our predictions: $\mathcal{I} = \hat{\mathcal{A}} \cdot \hat{\mathcal{S}}$. To do so, we predict either $\mathcal{A}$ or $\mathcal{S}$, deduce the other by inverting the element-wise product of Eqn. (1). This is naturally implemented in a CNN by incorporating an element-wise division of $\mathcal{I}$ by the predicted $\mathcal{S}$ at the end of the convolution layers, as illustrated in Fig. 2 by the magenta circle. This does not hinder CNN training because both predicted elements can be used in the loss function and even brings the advantage that gradients are naturally fused and can safely be back-propagated. Moreover, this guarantees the respect of Eqn. (1) by construction, which we believe to be important for further processing in several applications.

In the remainder of this paper, we present our network as first estimating $\mathcal{S}$ and deducing $\mathcal{A}$ by division as shown in Fig. 2. Note that the inverse can be done without any additional effort. In section 5.2, we compare both variants. Finally, we introduce the notations $\mathcal{C}$ (resp. $\hat{\mathcal{C}}$), and the vocabulary "component", which generically represent either $\mathcal{A}$ or $\mathcal{S}$ (resp. $\hat{\mathcal{A}}$ or $\hat{\mathcal{S}}$). We use it when the task is component-agnostic.

Residual learning. $\mathcal{I}$ shares a lot of common features with $\mathcal{A}$ and $\mathcal{S}$, hence the identity function is a good initial guess. However, CNN are known to struggle when the solution is close to the identity [13]. Residual networks have been proposed to tackle this drawback of deep networks. We incorporate this structure into our network to facilitate the estimation of $\mathcal{S}$.

Fig. 3 describes the CNN branch (cyan box in Fig. 2) that learns the mapping from $\mathcal{I}$ to $\mathcal{S}$. We build it as a sequence of residual blocks (RB) as proposed in [13]. A RB is composed of two convolution layers made of 64 convolutions of size 3×3. We use batch normalization [15] after every convolution as a regularization for the network. They are followed by a Rectified Linear unit (ReLu). An element-wise summation with the input is done before the ReLu of the second convolution. 10 such blocks are then connected in sequence. Finally, note that our generative network is

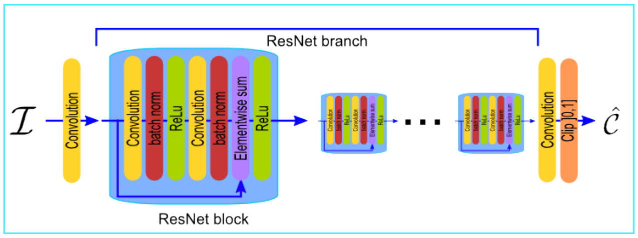

Figure 3: The generative CNN (cyan box in Fig. 2) is composed of 10 ResNet blocks in sequence. ResNet allows the CNN to focus on residual learning.

fully convolutional, hence it can process images of any size at test time.

3.2. Loss function

Let $\mathcal{I}_{gt}$, $\mathcal{A}_{gt}$ and $\mathcal{S}_{gt}$ denote the ground truth image, albedo and shading, respectively. Loosely speaking, the goal of the learning task is to predict $\hat{\mathcal{A}}$ and $\hat{\mathcal{S}}$ "as close as possible" to $\mathcal{A}_{gt}$ and $\mathcal{S}_{gt}$. Formally, we train the network parameters ϕ to minimize a loss L composed of three terms representing a data loss, a data gradient loss and an adversarial loss:

$$\begin{aligned} L(\hat{\mathcal{A}}, \hat{\mathcal{S}}) =& L_{data}(\hat{\mathcal{A}}, \hat{\mathcal{S}}) + L_{\nabla}(\hat{\mathcal{A}}, \hat{\mathcal{S}}) \\ &+ \lambda L_{adv}(\hat{\mathcal{A}}, \hat{\mathcal{S}}) . \end{aligned} \tag{3}$$

Data loss. First and foremost, data loss ensures that predictions fit the ground truth data. As opposed to prior work [11, 23], we want a measure sensitive to scale that imposes consistent global intensity. Therefore, we define it as a L_2 norm over both the albedo and the shading:

$$L_{data}(\hat{\mathcal{A}}, \hat{\mathcal{S}}) = ||\mathcal{A}_{gt} - \hat{\mathcal{A}}||^2 + ||\mathcal{S}_{gt} - \hat{\mathcal{S}}||^2 . \tag{4}$$

Gradient loss. Both albedo and shading present sharp discontinuities which are essential as well as smooth planar surfaces. To favor estimations that exhibit variations consistent with both components and avoid over-smoothing, we thus complement the loss with a L_2 norm over the gradient of both the albedo and the shading:

$$L_{\nabla}(\hat{\mathcal{A}}, \hat{\mathcal{S}}) = ||\nabla \mathcal{A}_{gt} - \nabla \hat{\mathcal{A}}||^2 + ||\nabla \mathcal{S}_{gt} - \nabla \hat{\mathcal{S}}||^2 . \tag{5}$$

As shown in section 5.2, this term improves numerical results, especially on the shading.

Adversarial loss. We observed that state of the art results show typical generative CNN visual artifacts. Therefore we introduce an *adversarial loss* for both $\mathcal{A}$ and $\mathcal{S}$. This takes the form of two binary classifier CNN, called "discriminators", one per component: $\mathcal{D}_{\mathcal{A}}(\mathcal{A})$ and $\mathcal{D}_{\mathcal{S}}(\mathcal{S})$, respectively.

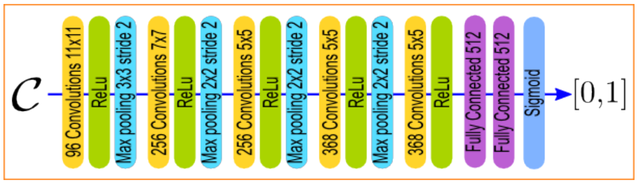

Figure 4: The adversarial loss is defined by a continuously trained discriminator network. It is a binary classifier that differentiates between ground truth data and generated data predicted by the generative network (Fig. 2).

A discriminator's output is the likelihood of a given component of belonging to the groundtruth data or generated data (see Fig. 4). The goal of the generator network then becomes to fool the discriminator so that it cannot find distinguishable features, contributing to get rid of the typical generative CNN artifacts.

Formally, we append the two discriminator networks $\mathcal{D}_{\mathcal{A}}$ and $\mathcal{D}_{\mathcal{S}}$ after both predictions $\hat{\mathcal{A}}$ and $\hat{\mathcal{S}}$, respectively (see Fig. 2). The whole forms a Generate Adversarial Network (GAN) [10]. $\mathcal{D}_{\mathcal{A}}$ and $\mathcal{D}_{\mathcal{S}}$ are then used to form a (deep) perceptually-motivated loss layer for the generative network. The loss it defined by

$$L_{adv}(\hat{\mathcal{A}}, \hat{\mathcal{S}}) = -log(\mathcal{D}_{\mathcal{A}}(\hat{\mathcal{A}}) \cdot \mathcal{D}_{\mathcal{S}}(\hat{\mathcal{S}})) , \tag{6}$$

which is low when the $\mathcal{D}_{\mathcal{C}}$ cannot classify $\hat{\mathcal{C}}$ to be generated ones. When the generator is trained, the discriminator is fixed and only serves as a perceptual measure. Backpropagation through the discriminator enriches the error information by transforming it in a more expressive representation, which guides the learning so as to eliminate distinctive features. The drawback is that its intensity might surpass other error information coming from less deep locations, which requires the addition of a weighting parameter λ in Eqn. (3).

The architecture of the discriminators follow standard classification architectures, by having convolutions with strided MaxPooling before a few fully-connected layers which lead to the final probability prediction (see Fig. 4).

During training, due to the poorness of the first estimations of the generator network, L_{adv} is unused (*i.e.*, $\lambda = 0$) for a few iterations (400 in our case), which improves training stability [10]. After that, the discriminators train the adversarial loss alongside the generator in an iterative expectation-maximization (EM)-like procedure: one being updated while the other is fixed, and vice versa.

The discriminator is trained using a binary classification loss:

$$L_{Discr}(\mathcal{C}_{gt}, \hat{\mathcal{C}}) = -log(\mathcal{D}_{\mathcal{C}}(\mathcal{C}_{gt})) - log(1 - \mathcal{D}_{\mathcal{C}}(\hat{\mathcal{C}})) . \tag{7}$$

The output of a discriminator being a likelihood of a sample to be taken from the ground truth, we learn network parameters that maximize the likelihood for ground truth samples and minimize it for generated ones [10].

4. Training & Evaluation

In this section, we discuss training data and details, as well as metrics used to quantitatively evaluate our results.

4.1. Dataset

Sintel is an open-source short computer animation movie that contains complex indoor and outdoor scenes. For research purposes, it has been published in various formats, among which its intrinsic shading and albedo layers, as the "MPI Sintel dataset" [6].

It is available for 18 sequences, 17 of which are composed of 50 images and one of 40, for a total of 890. The albedo images have been rendered without illumination. The shading images include illumination effects, and have been produced by rendering with a constant gray albedo. Because the "clean pass" version does not respect Eqn. (1), we follow [23] and recompose the original image as $\mathcal{I}_{gt} = \mathcal{A}_{gt} \cdot \mathcal{S}_{gt}$. This forms a consistent dataset called the "ResynthSintel" dataset used in our training and testing.

For fair comparison, we use the two variants of training/testing splits [23]. The *image split* randomly assigns half of the images to each set. The *scene split* assigns half of the movie's scenes to each set. The latter is motivated by the fact that consecutive frames within a scene are similar. Hence the training and testing sets of the *image split* are close, and an over-fitting network will perform well. Performing on the *scene split* is more challenging: it requires more generalization capacity.

During training, we used standard data augmentation techniques by randomly cropping patches of size 250 × 250 within the images after scaling by a random factor in $[0.8, 1.2]$, rotating by a random angle of maximum $15°$, and using random horizontal mirroring with a likelihood of 0.5.

4.2. Evaluation

For quantitative evaluation, error metrics have to be chosen. To be able to compare to related work [11, 7, 23], we consider the metrics they used: two data-related metrics (si-MSE and si-LMSE) and a perceptually motivated one (DSSIM). We will show that we outperform or compete with the state of the art results. However, because these metrics are defined to be insensitive to scale, we will additionally consider scale-sensitive measures, on which we outperform previous state-of-the-art. We now present all measures in detail.

4.2.1 Scale-Invariant Metrics

si-MSE. Denoted MSE in previous work, we rename it *scale invariant mean squared error* to avoid confusion:

$$\text{si-MSE}(\hat{\mathcal{C}}) = ||\mathcal{C}_{gt} - \alpha\hat{\mathcal{C}}||^2/N\,, \tag{8}$$

where $\mathcal{C} \in \{\mathcal{A}, \mathcal{S}\}$, $\alpha = \text{argmin}_\alpha\, ||\mathcal{C}_{gt} - \alpha\hat{\mathcal{C}}||^2$ and N is the number of pixels in $\mathcal{C}_{gt}$. Note that α is *separately* optimized for $\hat{\mathcal{A}}$ and $\hat{\mathcal{S}}$. Thus, any scale shift on either of them has no influence on the error, and errors in scale correlation between both predictions are not penalized either. Despite the classical flaw of heavy outlier weighting [4], si-MSE is a decent data-term, widely used in previous work and other applications like single-view depth estimation [9].

si-LMSE. Denoted LMSE in previous work, we rename it *scale invariant local mean squared error* for clarity.

$$\text{si-LMSE}(\hat{\mathcal{C}}) = \frac{1}{P}\sum_{\hat{\mathcal{P}}\in\hat{\mathcal{C}}} \text{si-MSE}(\hat{\mathcal{P}})\,, \tag{9}$$

where $\hat{\mathcal{P}}$ is square patch taken from $\hat{\mathcal{C}}$ of size 10% of its largest dimension and P is the number of patches. They are regularly extracted on a grid so as to have a 50% overlap between neighboring patches. Note that in this case, the scale parameter α (see Eqn. (8)) is optimized for each patch *individually*. This measure evaluates local structure similarity and is thus finer grained than the MSE.

DSSIM. The *structural similarity image index* [24] is a perceptually-motivated measure that accounts for multiple independent structural and luminance differences. It is here transformed into a dissimilarity measure:

$$DSSIM(\hat{\mathcal{C}}) = (1 - SSIM(\hat{\mathcal{C}}))/2\,. \tag{10}$$

4.3. Scale-Aware Metrics

We argue that there is no proper data-term measuring consistent scale-dependent reconstruction. Therefore, we propose to add two measures: MSE and rs-MSE.

MSE. First, the traditional scale-sensitive MSE measures our overall goal: numerically get as close as possible to the ground truth,

$$\text{MSE}(\hat{\mathcal{C}}) = ||\mathcal{C}_{gt} - \hat{\mathcal{C}}||^2/N\,. \tag{11}$$

rs-MSE. Second, we introduce the relative scale mean squared error. As noted by [11], Eqn. (11) is "too strict for most algorithms on our data". While we acknowledge that the ill-posed nature of intrinsic decomposition leads to

Sintel	si-MSE			si-LMSE			DSSIM		
Image Split	A	S	Avg	A	S	Avg	A	S	Avg
Baseline: Shading Constant	5.31	4.88	5.10	3.26	2.84	3.05	21.40	20.60	21.00
Baseline: Albedo Constant	3.69	3.78	3.74	2.40	3.03	2.72	22.80	18.70	20.75
Retinex [11]	6.06	7.27	6.67	3.66	4.19	3.93	22.70	24.00	23.35
Lee et al. [20]	4.63	5.07	4.85	2.24	1.92	2.08	19.90	17.70	18.80
Barron et al. [1]	4.20	4.36	4.28	2.98	2.64	2.81	21.00	20.60	20.80
Chen and Koltun [7]	3.07	2.77	2.92	1.85	1.90	1.88	19.60	16.50	18.05
Direct Intrinsics [23]	**1.00**	**0.92**	**0.96**	0.83	0.85	0.84	20.14	15.05	17.60
Our work	1.24	1.28	1.26	**0.69**	**0.70**	**0.70**	**12.63**	**12.13**	**12.38**

Table 1: Quantitative scale-invariant results ($\times 100$) after double cross-validation on the Sintel *image split*.

a scale ambiguity, we still think that Eqn. (1) should be preserved. Hence, we allow *relative* tuning between $\hat{\mathcal{A}}$ and $\hat{\mathcal{S}}$, using a unique scale parameter α:

$$\text{rs-MSE}(\hat{\mathcal{A}}, \hat{\mathcal{S}}) = (||\mathcal{A}_{gt} - \alpha\hat{\mathcal{A}}||^2 + ||\mathcal{S}_{gt} - \hat{\mathcal{S}}/\alpha||^2)/2N\,, \tag{12}$$

where $\alpha = \text{argmin}_\alpha\, \text{rs-MSE}(\hat{\mathcal{A}}, \hat{\mathcal{S}})$. This measures allows only relative scale optimization so that the following relationship is preserved:

$$\mathcal{I} = \alpha\mathcal{A} \cdot \mathcal{S}/\alpha\,, \tag{13}$$

hence is consistent with Eqn. (1). In other words: relative consistent intensity variations are tolerated with this measure, while global intensity must be preserved as well as structural information.

4.4. Technical details

Our generative network is trained efficiently during 8K iterations with batch size 5. The discriminators are learned alongside the generator (except for the first 400 iterations, see section 3.2)) in an EM-like procedure. To remain competitive, they are trained 3 times more: we iterate between 3 discriminator updates with fixed generator, followed by 1 generator update with fixed discriminator, for a total of $24K$ discriminator updates. Complete training took around 20 hours. We empirically find that $\lambda = 10^{-4}$ is the most efficient for our task. We use the Adam [16] optimization method with a learning rate starting at 10^{-4} and decreasing to 10^{-6}. Finally, we followed the double cross validation procedure to produce are results, which are the average of two evaluation using networks trained on reciprocal testing and training sets [23].

5. Results

In this section, we present qualitative and quantitative comparisons to related work (section 5.1), study the impact of each component of our architecture (section 5.2) and discuss generalization capacity and limitations with examples on the MIT intrinsic images dataset (section 5.3).

5.1. Comparison

We first compare on the scale-invariant metrics and *image split* test images (table 1 and Fig. 5). “Direct Intrinsics” (DI) [23] and our work are the only deep learning approaches, and the only ones predicting only from images, excluding depth. Numbers show that the deep learning approaches substantially outperform classical methods, even with depth as an additional input. A striking improvement of our method w.r.t. DI is the sharpness both in the spatial and color domain. This can be seen on the albedo, where contours and especially the background is sharp. Similarly, improved shading can be observed: the hair of the main character is well approximated in Fig. 5 (top left scene), thanks to the consistency enforcement of our approach. This is confirmed by the strong improvement on the DSSIM measure, which penalized local structural errors: our method corrects those by avoiding the typical generative CNN artifacts of blur, contrast oscillations and color bleeding. We think this is also the cause of our improvement on the si-LMSE measure, because our results are locally more consistent. Finally, we show slightly worse numerical results on the si-MSE metric on the *image split* data (table 1). Recall that this does not consider scale nor albedo-shading consistency in its measure and that we optimize for the more challenging scale-sensitive recovery.

On the more challenging *scene split* however, our algorithm improves on all numerical results, including the si-MSE metric (table 2). This suggests that our learned CNN has better generalization capabilities. We also compare on our newly proposed scale-sensitive metrics (table 2). Our method also heavily outperforms DI on both the MSE and rs-MSE metrics.

Finally, we refer the interested reader to our supplementary HTML page and videos for more results on the Sintel dataset.

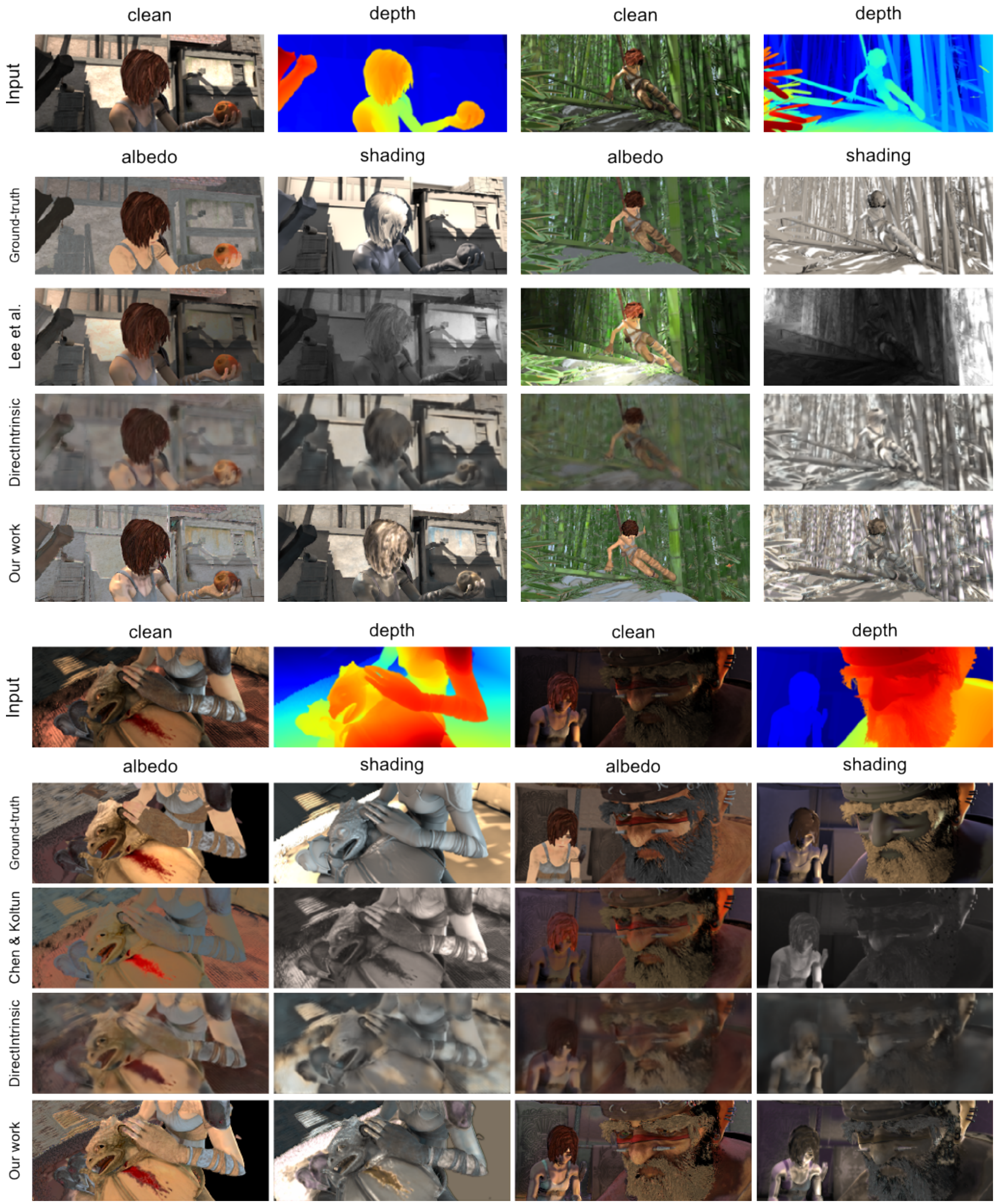

Figure 5: Qualitative results on the Sintel *image split*. Comparison to [20, 23] (top) and [7, 23] (bottom). Depth is not used by our method nor direct intrinsics.

Sintel *Scene Split*	si-MSE A	 S	 avg	si-LMSE A	 S	 avg	DSSIM A	 S	 avg	MSE A	 S	 Avg	rs-MSE
[23]	2.01	2.24	2.13	1.31	1.48	1.39	20.73	15.94	18.33	2.50	4.83	3.66	3.09
Ours	**1.77**	**1.84**	**1.81**	**0.98**	**0.95**	**0.97**	**14.21**	**14.05**	**14.13**	**2.10**	**2.19**	**2.15**	**1.85**

Table 2: Quantitative results (×100) on the Sintel *scene split*. MSE and rs-MSE measures for [23] have been calculated using results produced by their provided code and pretrained network.

Sintel *Image Split*	si-MSE A	 S	 Avg	si-LMSE A	 S	 Avg	DSSIM A	 S	 Avg	MSE A	 S	 Avg	rs-MSE
Full DARN	1.41	1.55	1.48	0.69	0.78	0.73	12.43	12.80	12.62	1.63	1.83	1.73	1.52
w/o L_{∇}	1.42	1.61	1.52	0.70	0.79	0.74	12.58	12.92	12.75	1.61	1.90	1.75	1.55
w/o L_{adv}	1.47	1.58	1.53	0.72	0.79	0.75	12.54	12.89	12.71	1.71	1.87	1.79	1.56
w/o res. blocks. (RB)	1.56	1.87	1.72	0.74	0.90	0.82	13.12	13.40	13.26	1.91	2.36	2.14	1.76
w/o RB,L_{∇},L_{adv}	1.59	1.90	1.74	0.76	0.91	0.83	13.45	13.67	13.56	1.96	2.45	2.20	1.79
Full DARN (swapped A/S)	1.30	2.34	1.82	0.61	1.76	1.19	14.43	13.94	14.19	1.46	2.51	1.98	1.85

Table 3: Evaluation of each individual part of our algorithm: comparison between the full DARN model and variants with different parts stripped off. All variants were similarly trained by fixing the seed for the pseudo-random number generator.

5.2. Impact of individual contributions

Here, we analyze the effect of each part of our network by comparing results of our Deep Adversarial Residual Network (DARN) with results of learning with the following parts stripped off: gradient loss L_{∇}, adversarial loss L_{adv}, residual blocks (*i.e.*, remove the purple layers – the element-wise sums – in the architecture of Fig. 3), or those three removed altogether. We also analyze the variant where $\hat{\mathcal{A}}$ and $\hat{\mathcal{S}}$ are inverted in Fig. 2: albedo is predicted and shading deduced. All variants were trained identically with 4000 iterations, lasting around 10 hours.

Table 3 shows quantitative results: components are ranked by impact on the error, from least to highest. All components prove useful, but the residual blocks have the most impact, especially on the shading (see MSE and rs-MSE metrics). Finally, swapping $\hat{\mathcal{A}}$ and $\hat{\mathcal{S}}$ generates significantly worse results, hinting that learning shading, which is smoother and more regular across the color channel, is easier than albedo, containing higher texture frequencies, especially in the Sintel dataset.

5.3. Generalization & Limitations

We trained on the Sintel dataset and analyze how the learned model behaves on other data with the MIT dataset [11] in Fig. 6. The top example is decently handled: both $\hat{\mathcal{A}}$ and $\hat{\mathcal{S}}$ are decent approximations of their respective ground truths. The bottom one on the other hand shows inconsistencies: while the MIT dataset contains only gray shading, the shading contains blue-greenish colors.

These results suggest that a Sintel-trained model behaves well on any example that fits the Sintel assumptions (Sintel is known to be biased towards brownish albedo's and blueish shading). Training on the MIT dataset would surely improve these results but we think that, because of the small size of the dataset, this is doomed to severe overfitting. Hence, we call for more realistic large-scale datasets that will hopefully become widespread with the development of scalable and realistic computer graphics renderers.

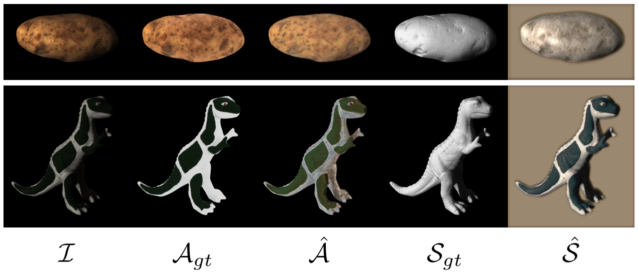

Figure 6: Qualitative results of our CNN trained on MPI Sintel and applied on the MIT intrinsic images dataset. In some cases (top), Sintel training generalizes well. In others (bottom), one can see there is a dataset bias towards blue/green shading.

6. Conclusion

We presented a new architecture that learns single-image intrinsic decomposition. By composing a powerful CNN architecture, our contribution outperforms state-of-the-art methods while being easier to train and implement. Besides numerical improvements, the main advantage is the guarantees that are given: Eqn. (1) is respected in any case, leading to consistent results. We believe its simplicity and consistency make it easily usable for many applications, hence we will provide code upon publication.

As one of the key motivations is to produce a consistent albedo regardless of environmental or acquisition factors, an interesting future work would be to exploit our contribution for temporally consistent predictions, or to further decompose $\hat{S}$ into sub-components.

References

[1] J. T. Barron and J. Malik. Shape, illumination, and reflectance from shading. *TPAMI*, 2015. 2, 6

[2] H. Barrow and J. Tenenbaum. Recovering intrinsic scene characteristics. *Comput. Vis. Syst., A Hanson & E. Riseman (Eds.)*, pages 3–26, 1978. 1

[3] S. Bell, K. Bala, and N. Snavely. Intrinsic images in the wild. *ACM Trans. on Graphics (SIGGRAPH)*, 33(4), 2014. 2, 3

[4] S. Bermejo and J. Cabestany. Oriented principal component analysis for large margin classifiers. *Neural Networks*, 14(10):1447 – 1461, 2001. 5

[5] A. Bousseau, S. Paris, and F. Durand. User assisted intrinsic images. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2009)*, 28(5), 2009. 2

[6] 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. 3, 5

[7] 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, 5, 6, 7

[8] S. Duchêne, C. Riant, G. Chaurasia, J. Lopez-Moreno, P.-Y. Laffont, S. Popov, A. Bousseau, and G. Drettakis. Multiview intrinsic images of outdoors scenes with an application to relighting. *ACM Transactions on Graphics*, 2015. 2

[9] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2366–2374. Curran Associates, Inc., 2014. 5

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. 2, 4, 5

[11] 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, 4, 5, 6, 8

[12] M. Hachama, B. Ghanem, and P. Wonka. Intrinsic scene decomposition from rgb-d images. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 810–818, Washington, DC, USA, 2015. IEEE Computer Society. 2

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 2, 3

[14] B. K. Horn. *Robot Vision*. McGraw-Hill Higher Education, 1st edition, 1986. 2

[15] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 3

[16] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference for Learning Representations*, 2014. 6

[17] P.-Y. Laffont and J.-C. Bazin. Intrinsic decomposition of image sequences from local temporal variations. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015. 2

[18] P.-Y. Laffont, A. Bousseau, S. Paris, F. Durand, and G. Drettakis. Coherent intrinsic images from photo collections. *ACM Transactions on Graphics (SIGGRAPH Asia Conference Proceedings)*, 31, 2012. 2

[19] E. H. Land and J. J. McCann. Lightness and retinex theory. *JOSA*, 61(1):1–11, 1971. 2

[20] K. J. Lee, Q. Zhao, X. Tong, M. Gong, S. Izadi, S. U. Lee, P. Tan, and S. Lin. Estimation of intrinsic image sequences from image & depth video. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI*, ECCV'12, pages 327–340, Berlin, Heidelberg, 2012. Springer-Verlag. 2, 6, 7

[21] L. Shen, T. Ping, and S. Lin. Intrinsic image decomposition with non-local texture cues. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008*, pages 1–7. IEEE, 2008. 2

[22] 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. 2

[23] 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. 1, 2, 4, 5, 6, 7, 8

[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. 5

[25] Q. Zhao, P. Tan, Q. Dai, L. Shen, E. Wu, and S. Lin. A closed-form solution to retinex with nonlocal texture constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1437–1444, July 2012. 2

[26] T. Zhou, P. Krähenbühl, and A. A. Efros. Learning data-driven reflectance priors for intrinsic image decomposition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3469–3477, 2015. 2