Intrinsic Image Decomposition Using Deep Convolutional Networks

Matt Vitelli
mvitelli@stanford.edu

***Abstract*—This project applies deep convolutional networks to the task of intrinsic image decomposition. By learning a set of non-linear filters, I am able to directly map RGB pixels to shading and reflectance estimates of the scene. Finally, I compare the convolutional network's decompositions to a popular decomposition algorithm known as the Retinex method. The convolutional network outperforms the Retinex method on a common benchmark dataset.**

I. Introduction

Intrinsic image decomposition still remains a relatively unsolved problem since its introduction nearly four decades ago. The primary reason for this is that intrinsic image decomposition is inherently ill-posed; an algorithm must somehow determine how to decompose a single image into two signals – namely, a reflectance field (also known as an albedo map) and a shading field representing the scene lighting. Some parameterizations also attempt to decompose the original image into additional signals such as an indirect radiance map or a specular BRDF map [1].

The goal of this project is to utilize convolutional networks as a means of computing intrinsic image decompositions in a plausible manner that closely approximates the ways in which such quantities are computed in computer graphics. I believe this will not only result in more accurate decomposition results, but also provide a bridge between creating virtual content and synthesizing it from real world images. An added benefit of having such decompositions is that they provide a helpful first step towards inverting the rendering equation.

II. Related Work

Due to recent improvements in 3D computer vision techniques, there has been a growing interest in intrinsic image decomposition. [3] created a joint lighting estimation and intrinsic image refinement model using coarse scene geometry annotated by users. [1] utilized depth and surface normal estimates with spatial regularization to separate direct lighting, indirect lighting, and albedo maps in a convex optimization framework. [4] decomposed intrinsic images from RGB-D videos by first computing a 3D mesh of the environment using KinectFusion followed by averaging the scene colors from each frame projected onto the mesh in order to provide increased robustness to specular highlights. They also used the scene geometry to simulate common shaded regions and incorporate occlusions into their decomposition method.

While nearly all of the methods above phrase intrinsic image decomposition as a convex problem, to my knowledge no publications exist that model intrinsic image decomposition as a non-linear filtering problem. In this sense, the method proposed in the following sections is entirely novel, as is the application of convolutional networks as a means of computing the intrinsic image decomposition.

III. Problem Statement

The intrinsic image decomposition problem is defined as follows: given a single RGB image I of a natural scene, compute a RGB value for each pixel at a location x,y representing the object's reflectance $A(x,y)$ along the camera ray and compute a RGB $S(x,y)$ value for the pixel representing the shading at the pixel's corresponding point in 3D space. Under Lambertian

assumptions, the image formation I can be modeled as

$$I(x,y) = A(x,y)S(x,y) \quad (1)$$

While the equation above doesn't hold true in general, many objects in the natural world exhibit some form of Lambertian reflectance, so the model tends to hold for many real world scenes. As is evident from equation (1), the problem of computing $A(x,y)$ and $S(x,y)$ is inherently ill-posed, as we must somehow estimate two values from a single estimate $I(x,y)$. Because of this, we must rely on some collection of priors to aid in estimating both quantities.

IV. Technical Approach

In contrast to existing methods that rely on carefully engineered priors that typcially operate in the logarithmic domain [1, 2, 4], I attempt to learn priors using a popular deep learning technique known as convolutional networks.

I modeled my network architecture with the notion that intrinsic image decomposition is primarily a filtering operation. Common deep learning approaches such as pooling tend to discard valuable information embedded within the image and are less applicable to full image regression tasks. Because of this, my network is implemented as a series of raw convolutions followed by non-linear constraints known as rectified units (RelUs). The objective function is to minimize L2 loss with respect to the ground truth light maps, as seen in the diagram below. The network is relatively shallow due to the memory constraints of the GPUs I had access to for training. I used a series of decreasing filter sizes with increasing depth layers to compensate for these constraints.

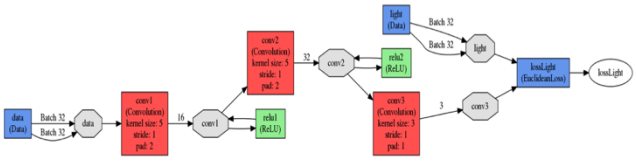

Figure 1. Convolutional Network Architecture

The convolutional network was trained using Caffe, a popular deep learning library provided by the Berkeley Vision and Learning Center. The implementation operates on RGB images of 320 by 240 pixels. Performing a forward pass through the network takes around 8 milliseconds, making it fast enough for real-time video processing.

V. Dataset Acquisition

Training deep networks requires vast amounts of training data. Because only a small handful of intrinsic image datasets exist comprised of a few dozen images, it is impractical to use such datasets for training purposes. To compensate for this, I wrote my own dataset generator using Unity 3D, a popular 3D game engine. The dataset generator outputted 3,000 images from a variety of camera poses, lighting conditions, and albedo configurations within the scene. All images make use of global illumination to accurately simulate light transport throughout the scene. The renderer outputted ground truth albedo maps, camera-space surface normals, depth maps, and light maps from every generated perspective. Using this data, the ground truth image can be recomputed by multiplying each albedo map with its corresponding light map. An example of the training data can be seen in figure 2 below. The outputs on the left correspond to the scene light maps, while the outputs on the right correspond to the scene albedo maps.

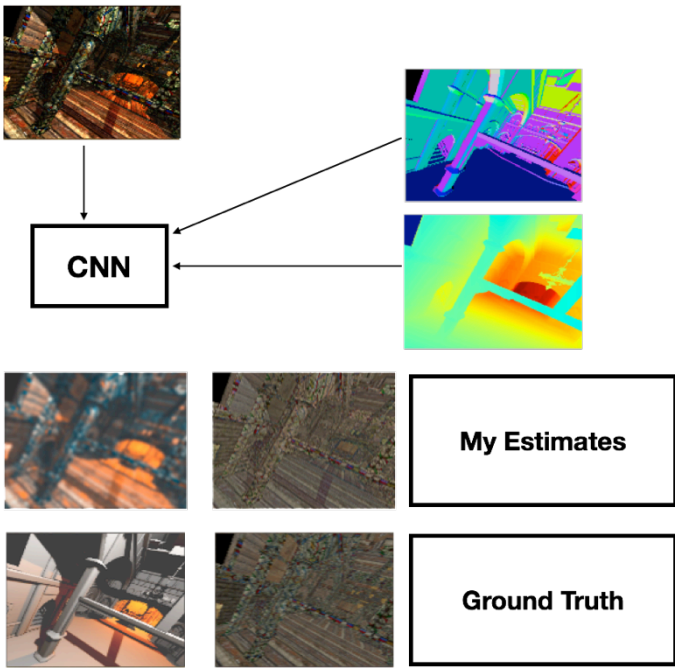

Figure 2. An example image from the dataset and the output of the convolutional network.

VI. Experiments

I trained several networks for this project. The first network tried to resolve albedos from the image and compute the light map from the resulting albedo. While this network exhibited a low training and test error, the results were unsatisfactory and neither the light maps nor the albedo maps looked physically plausible. The second network estimated the light maps and used the lighting estimates to solve for the albedo images. This second approach worked much better and resulted in smaller error rates for both the predicted light maps and predicted albedo maps. I also experimented with a third network that utilized depth and normal information in the first layer of the convolutional network to improve shading estimates, but I found in practice that depth and normal information did not significantly reduce the testing or training error.

VII. Results

I evaluated the trained convolutional network on the dataset provided by [2]. and compared it against both the lighting and albedo maps produced by the popular color Retinex method. For the color Retinex method, each channel was solved for individually and I used the tuned hyper-parameters produced by [2]'s solver. I computed ground truth light maps for each image in the dataset by dividing the lighting images by the ground truth reflectance mask. I computed the root-mean-squared-error (RMSE) between the Retinex method's albedos and the grouth truth albedos, the Retinex method's lighting estimates against the ground truth light maps, the convolutional network's albedos against the ground truth albedos, and the convolutional network's lighting estimates against the ground truth light maps. Only valid pixels in the dataset were used for error evaluation and the error was calculated across 200 images. The results are summarized in the table below.

	Retinex Method	**Convolutional Network**
Albedo RMSE	0.4140	0.2878
Lighting RMSE	0.3484	0.3420

Figure 3. RMSE evaluation between the Color Retinex method and the convolutional network

As is evident from figure 3, the convolutional network exhibits comparable error rates to the tuned Retinex method's lighting estimates. The computed albedos tend to be more accurate than the tuned Retinex method's estimates. An example of the outputs from both methods is shown in the figure below. In general, it seems as though the convolutional network consistently predicts dark patches to belong to the light map rather than the albedo map. This is likely due to biases in the training data. The fact that the convolutional network outperformed the Retinex method on an entirely new dataset seems very promising and is a testament to the convolutional network's generalization power.

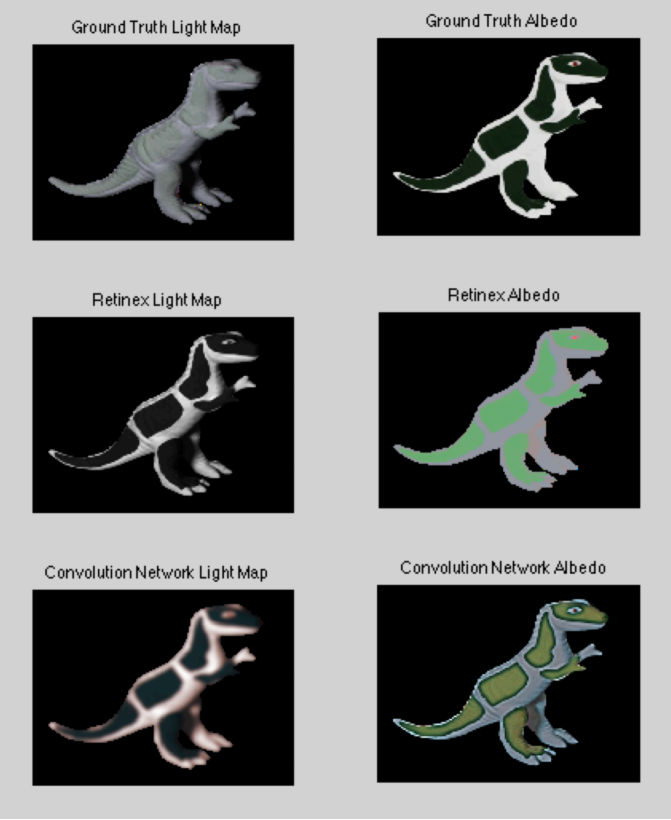

Figure 4. An example of the decomposition from both the Retinex method and the convolutional network.

VIII. FUTURE WORK

While this project has established that convolutional networks can be successfully applied to the task of intrinsic image decomposition, many questions remain unanswered. First, it would be useful to understand the relationships between filter sizes and network depths as a means of accurately resolving shading estimates. Similarly, the dataset used in this project provides only a simple model of the light transport and does not encompass the range of common illuminants observed in nature. Future work could seek to examine this relationship in more detail by testing deeper architectures and gathering a wider range of data based on common CIE illuminants.

IX. CONCLUSION

This project has demonstrated that intrinsic image decomposition can be greatly improved through the use of deep convolutional networks. The approach presented above seems to provide more accurate shading and albedo estimates than other common decomposition techniques such as the Retinex method.

X. REFERENCES

[1] Chen, Q. and Koltun, V. A Simple Model for Intrinsic Image Decompostion with Depth Cues. In *ICCV,* 2009.

[2] Grosse, R., Johnson, M. K., Adelson, E. H., and Freeman, W. T. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *Computer Vision, 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 2335–2342.

[3] Karsch, K., Hedau, V., Forsyth, D., and Hoiem, D. Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics (TOG)* 30, 6 (2011), 157.

[4] Lee, K. J., Zhao, Q., Tong X., Gong M., Izadi, S., Lee, S. U., Tan, P., and Lin, S. Estimation of intrinsic image sequences from image + depth video. In *ECCV,* 2012.