Bayesian Nonparametric Intrinsic Image Decomposition

Jason Chang, Randi Cabezas, and John W. Fisher III

CSAIL, MIT

Abstract. We present a generative, probabilistic model that decomposes an image into reflectance and shading components. The proposed approach uses a Dirichlet process Gaussian mixture model where the mean parameters evolve jointly according to a Gaussian process. In contrast to prior methods, we eliminate the Retinex term and adopt more general smoothness assumptions for the shading image. Markov chain Monte Carlo sampling techniques are used for inference, yielding state-of-the-art results on the MIT Intrinsic Image Dataset.

Keywords: Intrinsic images, Dirichlet process, Gaussian process, MCMC

1 Introduction

Intrinsic image analysis, first introduced in [2], is the problem of decomposing an image into various scene characteristics. Assuming a Lambertian surface model, where the perceived illumination is constant from all angles of incidence, the observed image decomposes into the product of the intrinsic shading and reflectance images. The reflectance image contains the albedo of the object surface, whereas the shading image captures the amount of reflected light from the surface. An example decomposition using the proposed approach is shown in Figure 1.

While interesting in its own right, intrinsic image analysis is also important for other fields of computer vision. For example, the shading image can be exploited in shape-from-shading algorithms to reveal the underlying 3D structure of an object or to infer elements of the scene illumination, such as the number, location, and color of the light sources. Use of the reflectance image improves many segmentation algorithms, where shading effects often introduce artifacts.

We consider the problem of intrinsic reflectance and shading decomposition from a single observation. The Retinex algorithm [3,11,12], one of the first proposed solutions, detects edges in the observed image and solves for a reflectance

Fig. 1: An example of the intrinsic image problem. Left-to-right: original image, inferred shading and reflectance images under the proposed method.

image that has matching gradients at the detected edges. Surprisingly, many methods still require these gradient-matching terms to achieve good results. We show that these terms are not required to achieve state-of-the-art results. While aspects are related to previous methods, the presented formulation differs by: (1) using a Dirichlet process Gaussian mixture model for the reflectance image instead of setting a fixed number of components; (2) using a Gaussian process to model the shading image for added expressiveness; (3) treating the image as an observation from a *generative*, stochastic process; and (4) developing inference techniques that are robust to initialization.

2 Related Work

Many algorithms have been developed to decompose images into their intrinsic components. Some use multiple images to disambiguate the decomposition (e.g., [21]), while others use data-driven, patch-based algorithms (e.g., [6]).

The original Retinex algorithm [12], which many algorithms build upon (e.g., [3,7,8,11,14,17,20]), still performs well decades after its original inception. Results on the MIT Intrinsic Image Dataset [9] show that the original formulation in 1971 outperforms all other algorithms prior to 2009. The different flavors of Retinex all include two underlying concepts: sharp edges should occur in the reflectance image, and the shading image should be smooth. Edges in the image are first detected, typically by thresholding intensity or chromaticity gradients. Gradients of the reflectance image are then favored to match gradients in the observed image at the detected edges. This type of interaction is often referred to as the "Retinex term". A smoothness assumption in the shading image is then used to propagate the bias of the Retinex term away from the edges.

Some recent extensions to the Retinex algorithm have improved results. Many authors have observed that a small set of distinct colors can often be used to model the reflectance image (e.g., [1,8,17,18,22]). In particular, Shen et al. [17,22] group reflectance values based on a local texture patch. They develop a "match weight" for each pairwise match that is used as a heuristic to weight reflectance differences in their energy functional. Gehler et al. [8] explicitly partitions the pixels based on their reflectance colors into K clusters. However, it is unclear how to set K *a priori*, since one would expect this value to be dependent on the particular image. In contrast, we model the reflectance image with a Dirichlet process mixture model that does not predefine a model order.

Smoothness in the shading image is most commonly enforced with a Markov random field (MRF) and an L_1 or L_2 penalty on the difference of neighboring shading pixels. We note that an L_2 penalty is equivalent to using an improper Gaussian MRF (GMRF) prior [13]. These types of model are used in [8], [17], and every method in the survey paper of [9]. In this work, we place a similar prior on the shading image. However, instead of restricting the smoothness to be a 4-connected GMRF as was done previously, we allow for a much broader class of smooth functions by placing a Gaussian process (GP) prior on the shading image. Stationary GMRFs are approximately finite realizations of GPs with

Table 1: Differences in Algorithms for Intrinsic Image Decomposition

	Gehler et al. [8]	**Proposed Model**
Shading Smoothness	4-connected GMRF	Gaussian process
Reflectance Prior	Uniform over fixed K clusters	Dirichlet process
Observations	Noiseless	Log-Normal noise
Probabilistic Model	Discriminative	Generative
Retinex Term	Yes	No
Inference	Iterative optimization	Marginalized MCMC

stationary covariance kernels. However, as we shall see, framing the model using a GP allows us to exploit two advantages: (1) inference is simplified with GPs; and (2) changing the smoothness is a matter of altering the covariance kernel without having to explicitly adapt to a different graphical MRF structure.

The two current state-of-the-art algorithms take quite different approaches. SIRFS [1], the current best-performing algorithm on [9], differs from most methods by inferring 3D geometry and treating the shading image as a by-product of the lighting conditions and 3D surface. One might draw the conclusion from these results that modeling the 3D structure is essential to good performance; however, as we will show, that is not necessarily the case. Furthermore, training and inference in SIRFS is challenging due to the large set of parameters.

Our model can be thought of as the Bayesian nonparametric extension to the second best-performing algorithm of Gehler et al. [8]. Table 1 summarizes explicit differences between the two approaches. While [8] has shown that the Retinex term improves results, it is difficult to incorporate such a term in a *generative* model. Moreover, our experiments show that by using a more expressive shading model and improved inference, the Retinex term is unnecessary to achieve state-of-the-art results. This work also departs from [8] by placing Bayesian priors that adapt to different noise characteristics and object complexities.

3 Generative Model

As is common in intrinsic image analysis, we assume a Lambertian surface model, where an image decomposes into the product of a shading image and a reflectance image. We now present a generative model, depicted in Figure 2, that contains this explicit decomposition. For the remainder of this paper, we will work in the log domain where the log of the observed image, x, is assumed to be generated from the sum of the log shading and the log reflectance image.

The log reflectance image is generated from a standard Dirichlet process Gaussian mixture model (DPGMM) as follows: (1) infinite-length mixture model weights, π, are drawn from a stick-breaking process [16]; (2) the mean RGB color for each cluster, μ_k, is drawn from a multivariate Gaussian prior; and (3) the cluster assignment for each pixel, i, denoted z_i, is drawn from a categorical dis-

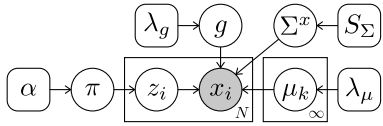

Fig. 2: The generative graphical model. See text for description. $\lambda_\mu = \{\theta, \Sigma^\mu\}$ and $\lambda_g = \{\kappa, \sigma_g^2, \nu, l\}$ denote sets of hyper-parameters.

tribution with parameters π. The following expressions summarize this process:

$$p(\pi) = \text{GEM}(\pi\ ;\ 1, \alpha), \tag{1}$$

$$p(\mu) = \prod\nolimits_k p(\mu_k) = \prod\nolimits_k \mathcal{N}(\mu_k\ ;\ \theta, \Sigma^\mu)\,, \tag{2}$$

$$p(z|\pi) = \prod\nolimits_i p(z_i|\pi) = \prod\nolimits_i \text{Cat}(z_i\ ;\ \pi). \tag{3}$$

The hyper-parameters, α, θ, and Σ^μ, are chosen to specify broad priors. The $3K \times 1$ vector of means is denoted by μ, where K is the number of realized clusters for 3 color channels. The log reflectance image, denoted μ_z, is then formed by setting each pixel, $[\mu_z]_i$, to the corresponding cluster mean: $[\mu_z]_i = \mu_{z_i}$. The reflectance image is then a $3N \times 1$ vector for an image with N pixels.

The log shading image, denoted g, is generated from a zero-mean Gaussian process (GP) with a stationary covariance kernel, κ. Shading images of interest (e.g., in the MIT Intrinsic Image Dataset [9]) are often generated from white-colored incident light. However, we find that allowing colored shading images generally results in better convergence. As such, we model g as a 3D Gaussian process with a covariance kernel that is a function of location and color. Furthermore, we are only interested in the values at the fixed grid locations. Since any subset of variables in a GP is jointly Gaussian, we can express the GP as

$$p(g) = \text{GP}(g\ ;\ \kappa) = \mathcal{N}(g\ ;\ 0, \Sigma^g)\,, \tag{4}$$

where Σ^g denotes the finite-dimensional covariance matrix obtained by evaluating the kernel, κ, at the grid points. The specific covariance kernel parameters govern the smoothness properties of g and are learned from training data.

Finally, we assume that the observed pixels in the log image are drawn independently from the following Gaussian distribution:

$$p(x|\mu, z, g, \Sigma^x) = \prod\nolimits_i p(x_i|\mu, z_i, g_i, \Sigma^x) = \prod\nolimits_i \mathcal{N}(x_i\ ;\ \mu_{z_i} + g_i, \Sigma^x)\,. \tag{5}$$

While one could assume a fixed observation covariance, Σ^x, we have found that it is difficult to set *a priori*, and instead treat Σ^x as a latent variable. One possibility is to use a cluster-specific covariance instead of a global covariance (e.g., via a Normal Inverse-Wishart prior). However, as described in Section 4, a global observation covariance that is also Toeplitz lends itself to efficient inference of g. As we are unaware of conjugate priors on positive definite *Toeplitz* matrices, the prior on Σ^x is uniform over a *discrete* set of covariances, S_Σ:

$$\Sigma^x = S_\Sigma(u), \quad u \sim \text{Uniform}(|S_\Sigma|). \tag{6}$$

The elements of S_Σ are chosen to be 3×3 matrices with color correlations logarithmically spaced in $[2^{-10}, 2^0]$ and marginal variances logarithmically spaced in $[2^{-7}, 2^0]$. This choice does not affect results significantly as long as the range is sufficiently broad. Visualizations can be found in [4].

Relation to DPGMMs. Typical DPGMMs draw each pixel from one of the infinite Gaussians with mean μ_k, regardless of the pixel location. The proposed model departs from the DPGMM by *jointly* changing the μ_k's in space according to g. One can view each pixel, i, as being drawn from a Gaussian with spatially-varying mean, $\mu_k(i) = \mu_k + g_i$. As such, we refer to this model as the spatially-varying DPGMM (SV-DPGMM). Additional details are included in [4].

4 Posterior Inference

One motivation for generative models is that computation of marginal event probabilities are generally more robust to noise as compared to point estimates such as the maximum *a posteriori* estimate. Consequently, we reason over the full distribution of the SV-DPGMM rather than use optimization approaches. MCMC methods, such as Gibbs sampling or the Metropolis-Hastings algorithm, are commonly used in complex probabilistic models such as the SV-DPGMM.

Before developing the inference techniques, we introduce some notation. Covariance matrices are denoted by Σ, possibly superscripted by an associated random variable. Corresponding precision matrices are denoted by $\Lambda \triangleq \Sigma^{-1}$. We use i and j for pixel indices in $[1, N]$, k and ℓ for cluster indices in $[1, K]$, and m and n for color channel indices in $[1, 3]$. As the posterior inference is complex, we build the algorithm over the next three sections.

4.1 Iterative Posteriors Inference without Marginalization

Conditioned on the GP, g, the SV-DPGMM simplifies to a traditional DPGMM. We sample this via the DP Sub-Cluster method [5], which restricts each Gibbs iteration to the current non-empty clusters and proposes split and merge moves. The relevant posterior distributions can expressed as:

$$p(\pi|z) = \mathrm{Dir}(\pi \,;\, N_1, \ldots, N_K, \alpha), \tag{7}$$

$$p(\mu|\Sigma^x, z, g, x) = \prod_{k=1}^{K} \mathcal{N}\Big(\mu_k \,;\, \overline{\theta}(x_{\mathcal{I}_k} - g_{\mathcal{I}_k}, \Sigma^x), \overline{\Sigma}^{\mu}(x_{\mathcal{I}_k} - g_{\mathcal{I}_k}, \Sigma^x)\Big), \tag{8}$$

$$p(\Sigma^x|\mu, z, g, x) \propto \sum_{u=1}^{|S_\Sigma|} p(x|\mu, z, g, \Sigma^x = S_\Sigma(u)), \tag{9}$$

$$p(z|\pi, \mu, \Sigma^x, g, x) = \prod_{i=1}^{N} \sum_{k=1}^{K} \mathbb{1}[z_i = k]\, \pi_k\, \mathcal{N}(x_i \,;\, \mu_k + g_i, \Sigma^x), \tag{10}$$

where $\mathcal{I}_k \triangleq \{i; z_i = k\}$ is the set of pixel indices assigned to cluster k, $N_k \triangleq |\mathcal{I}_k|$ counts the number of pixels assigned to cluster k, and $\overline{\theta}$ and $\overline{\Sigma}^{\mu}$ denote posterior hyper-parameters that are functions of the data through the conjugate prior.

Algorithm 1 SV-DPGMM Iterative Inference via MCMC

1. Initialize z and g to be all 0.
2. Sample $(z, \mu, \Sigma^x | g, x)$ using the DP Sub-Clusters algorithm [5].
3. Sample $(g | \mu, \Sigma^x, z, x)$ from Equation (11) using equivalent kernel [19] techniques.
4. Repeat from Step 2 until convergence.

We note that the posterior on Σ^x is just the prior weighted by the likelihood because of the uniform prior over a discrete set (see Equation (6)).

Conditioned on the cluster assignments, z, and cluster parameters, μ, the posterior on g is known to be Gaussian with the following distribution (cf. [15]):

$$p(g|\mu, z, \Sigma^x, x) = \mathcal{N}\big(g \; ; \; \Sigma^g \Lambda^{g+x}(x - \mu_z), \Sigma^g - \Sigma^g \Lambda^{g+x} \Sigma^g\big) \,, \tag{11}$$

where $\Lambda^{g+x} \triangleq (\Sigma^{g+x})^{-1} \triangleq (\Sigma^g + \Sigma^x \otimes \mathrm{I}_N)^{-1}$, $\otimes$ denotes the Kronecker product, and I_N denotes an $N \times N$ identity matrix. We note that $\Sigma^x \otimes \mathrm{I}_N$ is a $3N \times 3N$ block diagonal matrix where each 3×3 block represents the observation covariance for a 3-channel, colored pixel. If the GP uses a stationary covariance kernel, sampling from Equation (11) is well approximated using equivalent kernel methods [19]. Details of the approximation are shown in [4].

Equations (7)–(11) express the conditional distributions of all latent variables. Posterior inference can then alternate between sampling these expressions, as described in Algorithm 1. This procedure is very closely related to the procedure of [8], except that we solve Equation (11) analytically while [8] utilizes conjugate gradient iterations. Algorithm 1 empirically converges to local extrema and is sensitive to initialization. The method of [8] attempts to circumvent this issue by choosing the best solution from multiple initializations.

4.2 Marginalized Posterior Inference

Both the reflectance, μ, and shading, g, contribute additively in the log domain. Consequently, errors in one can be incorrectly explained by the other. Such problems are addressed in Bayesian inference by treating one variable as a nuisance parameter and marginalizing it out. While this is often intractable, marginalization of the shading image in the SV-DPGMM results in a closed-form expression. Since each distribution conditioned on z and Σ^x is Gaussian, the joint distribution, $p(x, \mu, g | z, \Sigma^x)$, must be jointly Gaussian, and any marginal or conditional distribution must also be Gaussian. We show in [4] that marginalizing over g results in $p(\mu | z, \Sigma_x, x) = \mathcal{N}(\mu \; ; \; \theta^*, \Sigma^*)$, where each element of the mean, θ^*, and precision, $\Lambda^* = (\Sigma^*)^{-1}$, is defined as

$$\Lambda^*_{km,\ell n} = \Lambda^\mu_{m,n} \quad + \sum\nolimits_{i \in \mathcal{I}_k} \sum\nolimits_{j \in \mathcal{I}_\ell} \Lambda^{g+x}_{im,jn}, \qquad \forall k = \ell, \tag{12}$$

$$\Lambda^*_{km,\ell n} = \qquad\quad \sum\nolimits_{i \in \mathcal{I}_k} \sum\nolimits_{j \in \mathcal{I}_\ell} \Lambda^{g+x}_{im,jn}, \qquad \forall k \neq \ell, \tag{13}$$

$$[\Lambda^* \theta^*]_{km} = [\Lambda^\mu \theta]_m + \sum\nolimits_{i \in \mathcal{I}_k} \sum\nolimits_j \sum\nolimits_n x_{jn} \Lambda^{g+x}_{im,jn}. \tag{14}$$

Equations (12)–(14) define a system of $3K$ linear equations for the reflectance colors that differs from Equation (8) by *marginalizing* over the shading image. This modification avoids dependence on possibly erroneous estimates of g. The current form requires the inversion of Σ^{g+x}, a large $3N \times 3N$ matrix, which is computationally burdensome. The covariance matrix, Σ^{g+x}, is evaluated on a square grid and will be Toeplitz for stationary covariance kernels. In the limit as the domain of observations extends to infinity, the precision will also be Toeplitz. If we approximate Λ^{g+x} as Toeplitz, Equations (12)–(14) become convolutions and are efficiently computed in the Fourier domain. In practice, we find that this approximation does not work well and consider the following alternative.

We note that the system of equations in Equations (12)–(14) only contains $4.5(K^2 + K)$ variables estimated from approximately N^2 variables. We remind the reader that K is the number of reflectance clusters (typically < 10) and N is the number of pixels (typically $> 50{,}000$). As such, there are many more observations than are necessary to reliably categorize θ^* and Λ^*. We therefore approximate the posterior on μ from a random *subset* of the data, where each cluster has at least 10 pixels and there are a total of at least 1,000 pixels.

Denoting the subset of pixel indices as $\mathcal{S}$, we then define a new realization of the GP on the subset of indices as $g_{\mathcal{S}}$ which is distributed according to $p(g_{\mathcal{S}}) = \mathcal{N}(g_{\mathcal{S}} \; ; \; 0, \Sigma^{g_{\mathcal{S}}})$. Following the same formulation as above, we can then approximate the posterior on the mean colors as

$$p(\mu|z, \Sigma^x, x) \approx p(\mu|z_{\mathcal{S}}, \Sigma^x, x_{\mathcal{S}}) = \mathcal{N}(\mu \; ; \; \hat{\theta}^*, \hat{\Sigma}^*), \tag{15}$$

where the approximate mean and precision are defined as

$$\hat{\Lambda}^*_{km,\ell n} = \Lambda^{\mu}_{m,n} \quad + \sum\nolimits_{i \in \mathcal{I}_k \cap \mathcal{S}} \sum\nolimits_{j \in \mathcal{I}_\ell \cap \mathcal{S}} \Lambda^{g_{\mathcal{S}}+x}_{im,jn}, \qquad \forall k = \ell, \tag{16}$$

$$\hat{\Lambda}^*_{km,\ell n} = \qquad\quad \sum\nolimits_{i \in \mathcal{I}_k \cap \mathcal{S}} \sum\nolimits_{j \in \mathcal{I}_\ell \cap \mathcal{S}} \Lambda^{g_{\mathcal{S}}+x}_{im,jn}, \qquad \forall k \neq \ell, \tag{17}$$

$$[\hat{\Lambda}^* \hat{\theta}^*]_{km} = [\Lambda^{\mu} \theta]_m + \sum\nolimits_{i \in \mathcal{I}_k \cap \mathcal{S}} \sum\nolimits_{j \in \mathcal{S}} \sum\nolimits_{n} x_{jn} \Lambda^{g_{\mathcal{S}}+x}_{im,jn}. \tag{18}$$

Due to the subsampling process, $\Lambda^{g_{\mathcal{S}}+x} = (\Sigma^{g_{\mathcal{S}}} + \Sigma^x \otimes \mathrm{I}_{|\mathcal{S}|})^{-1}$ can now be computed efficiently. We note that this approximation performs well in practice. The resulting inference procedure is summarized in Algorithm 2.

4.3 Marginalized Split/Merge Posterior Inference

In this section, we describe an improved procedure that changes z while marginalizing out both μ and g. As mentioned previously, we exploit the recent DP Sub-

Algorithm 2 SV-DPGMM Marginalized Inference via MCMC

1. Initialize z and g to be all 0.
2. Sample $(z, \mu, \Sigma^x|g, x)$ using the DP Sub-Clusters algorithm [5].
3. Sample $(\mu|\Sigma^x, z, x)$, marginalizing out g, from Equation (15).
4. Sample $(g|\mu, \Sigma^x, z, x)$ from Equation (11) using equivalent kernel [19] techniques.
5. Repeat from Step 2 until convergence.

Cluster sampling algorithm [5] to sample from the posterior of z. The core idea underlying the DP Sub-Cluster algorithm is to form two "sub-clusters" for each regular-cluster, and to use the sub-clusters to propose split moves. The prior distributions for the sub-clusters are chosen such that the posteriors are of the same form as Equations (7)–(10). Conditioned on the sub-clusters, a proposed split or merge is then used in a Metropolis-Hastings MCMC [10] framework that accepts the proposal with what is known as the Hastings ratio (cf. [5] for details).

Similar to the marginalization of the shading image g, we show in [4] that a related derivation can be used to express $p(x|z, \Sigma^x)$ as

$$p(x|z,\Sigma^x) = \frac{\left|\Lambda^{g+x}\right|^{1/2}|\Lambda^{\mu}|^{K/2}}{(2\pi)^{3N/2}|\Lambda^*|^{1/2}} \exp\left[\frac{1}{2}\left(\theta^{*\top}\Lambda^*\theta^* - K\theta^{\top}\Lambda^{\mu}\theta - x^{\top}\Lambda^{g+x}x\right)\right] \quad (19)$$

where the dependence on z and Σ^x are implied through Equations (12)–(14) for θ^* and Λ^*. A split of cluster k into clusters $\hat{k}$ and $\hat{\ell}$ using the DP Sub-Clusters algorithm, marginalizing over μ and g, is accepted with Hastings ratio

$$H_{\text{split}} = \frac{\alpha\Gamma(N_{\hat{k}})\Gamma(N_{\hat{\ell}})}{\Gamma(N_{\hat{k}}+N_{\hat{\ell}})} \cdot \frac{p(x|\hat{z},\Sigma^x)}{p(x|z,\Sigma^x)} \prod_{i\in\mathcal{I}_k} \frac{\overline{\pi}_{\hat{k}}\,\mathcal{N}(x_i\,;\overline{\mu}_{\hat{k}},\Sigma^x)+\overline{\pi}_{\hat{\ell}}\,\mathcal{N}(x_i\,;\overline{\mu}_{\hat{\ell}},\Sigma^x)}{\overline{\pi}_{\hat{z}_i}\,\mathcal{N}(x_i\,;\overline{\mu}_{\hat{z}_i},\Sigma^x)}, \quad (20)$$

where $\hat{z}$ is the newly split cluster labels, and $\overline{\pi}$ and $\overline{\mu}$ are sub-cluster parameters defined in [5]. Note that $p(x|z, g, \Sigma^x)$ integrates out the mean parameter. A similar marginalization applies to merge moves, resulting in the following Hastings ratio for a proposed merge of clusters k and ℓ into cluster $\hat{k}$:

$$H_{\text{merge}} = \frac{\Gamma(N_k+N_\ell)}{\alpha\Gamma(N_k)\Gamma(N_\ell)} \cdot \frac{p(x|\hat{z},\Sigma^x)}{p(x|z,\Sigma^x)} \prod_{i\in\mathcal{I}_{\hat{k}}} \frac{\pi_{z_i}\,\mathcal{N}(x_i\,;\mu_{z_i},\Sigma^x)}{\pi_k\,\mathcal{N}(x_i\,;\mu_k,\Sigma^x)+\pi_\ell\,\mathcal{N}(x_i\,;\mu_\ell,\Sigma^x)}. \quad (21)$$

This marginalized split/merge sampling method is summarized in Algorithm 3.

5 Parameter Learning

We now present two methods for learning model parameters. The first is supervised and uses training data to find the set of parameters that works best across all training examples. The second is unsupervised and places Bayesian hyper-priors on the parameters. The only parameters to learn are those of the covariance kernel in the GP, g. We use the Matérn class of kernels. Additionally,

Algorithm 3 SV-DPGMM Marginalized Split/Merge Inference via MCMC

1. Initialize z and g to be all 0.
2. Run DP Sub-Clusters to find likely splits conditioned on g (Σ^x is concurrently sampled within DP Sub-Clusters).
3. Sample $(z|\Sigma^x, x)$ via Metropolis-Hastings MCMC by proposing all splits or merges and accept with the Hastings ratios in Equations (20) and (21).
4. Sample $(\mu|\Sigma^x, z, x)$ marginalizing over g from Equation (15).
5. Sample $(g|\mu, \Sigma^x, z, x)$ from Equation (11) using equivalent kernel [19] techniques.
6. Repeat from Step 2 until convergence.

as mentioned previously, allowing for small amounts of color in the shading images improves convergence. As such, we alter the Matérn kernel to the following:

$$\kappa(c, r\ ;\ \sigma_c, \sigma_g^2, \nu, l) = \sigma_c^{\mathbb{1}[c\neq 0]}\ \sigma_g^2\ \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{r\sqrt{2\nu}}{l}\right)^{\nu} K_\nu \left(\frac{r\sqrt{2\nu}}{l}\right), \tag{22}$$

where c is the change in the color channel, r is the change in 2D location, $K_\nu(\cdot)$ is a modified Bessel function of the second kind, and $\lambda_g \triangleq \{\sigma_c, \sigma_g^2, \nu, l\}$ is the set of hyper-parameters to learn.

Supervised Learning. In the following sections, we test on the MIT Intrinsic Image Dataset [9]. Unfortunately, because the 20 images from [9] were released in two batches, some published methods are only trained or tested on a subset of the images. For example, [8] uses 16 of the 20 images, while [1] uses all 20 images. Furthermore, each method uses different training and test sets; [8] performs leave-one-out-cross-validation (LOOCV), while [1] separates the set into 10 training images and 10 test images. For an accurate comparison, we learned separate parameters using LOOCV and the separate training/test sets used in [1]. For each image, we ran the inference algorithm under a discrete set of parameter choices. The set of parameters that minimized the arithmetic mean of *RS*-MSE was chosen (similar to [8]). This error metric will be described shortly.

Unsupervised Learning. An alternative, Bayesian approach for *unsupervised* learning is to place a hyper-prior on the parameters, λ_g. For simplicity, we place a uniform prior on λ_g over a discrete set of plausible values. Inference then proceeds in the same sequence as before, with the added step of sampling λ_g from the posterior distribution, $\lambda_g \sim p(\lambda_g|g) \propto p(g|\lambda_g)$. This requires computing the likelihood of a GP realization with parameters λ_g, and can be efficiently approximated with methods described in [4].

6 Post-Processing for Color Constancy

One ambiguity in the shading and reflectance decomposition has not been explicitly addressed; namely, any color channel of the log-shading image can be shifted by an arbitrary amount if the log-reflectance image is shifted by the negative of the same amount. For example, this could correspond to changing the color of the light in the shading from white to blue and adding a yellow tint to reflectance image. The SV-DPGMM approach implicitly restricts these ambiguities. Because the GP is assumed to be zero-mean with correlated color channels, the shading image largely favors white lights and grayscale shading images. This is undesirable in some situations, one of which is shown in Figure 3.

Barron and Malik [1] address this color constancy issue by placing a prior on log reflectance values and assuming spherical harmonic lighting models. We take a slightly different approach since neither is easily applicable. We learn the distribution of the log shading and log reflectance values from the ground truth

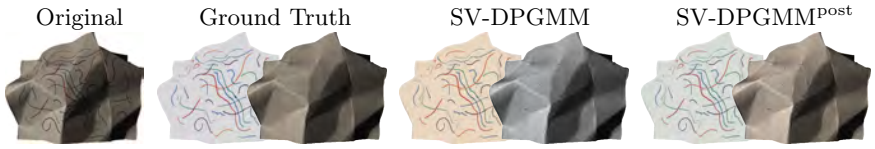

Fig. 3: An example of correcting color constancy as a post processing step.

training data via a kernel density estimate. It would be ideal if these distributions could be incorporated into the generative model, but the non-parametric nature of the distributions eliminate the exploited conjugacy in the inference. As such, we perform a post-processing step that finds the optimal global color-shift in the coupled shading/reflectance space. Additional details can be found in [4]. We note that this procedure can be used with any intrinsic image algorithm.

7 Experimental Results

For each image in the MIT Intrinsic Image dataset [9], we run Algorithm 3 for 50 iterations to ensure convergence, which typically occurs with 5–10 iterations. We then take the mean of 25 samples from the stationary distribution. This takes approximately 1–20 minutes, depending on the image. Since the simulated Markov chains tend to explore a local mode, we run 10 chains independently and show the resulting pixel-wise median shading and reflectance images. Each chain essentially explores a local mode of shading and reflectance, and the median of the 10 independent chains finds the mode that is in the middle. As we soon show, while this procedure slightly improves results, running a single chain still achieves state-of-the-art results. Publicly available source code can be downloaded from http://people.csail.mit.edu/jchang7/.

In the following section, we compare SV-DPGMM with Retinex and the two state-of-the-art methods from [1] and [8]. For an accurate comparison, we train the model parameters using the same training and test sets described in each of the previous methods. We compute three metrics from [1]: *S*-MSE, *R*-MSE, and *RS*-MSE. *S*-MSE and *R*-MSE compute the global scale-invariant shading and reflectance mean squared error, respectively. *RS*-MSE is the metric from [9], which computes the average of local scale-invariant MSEs. We evaluate both the arithmetic and geometric mean (denoted with a 'g') across the images. We note that [8] and [9] use the arithmetic mean while [1] uses the geometric mean.

7.1 SV-DPGMM Ablation Testing

We first compare different inference methods for SV-DPGMMs using LOOCV on the 16 images of the original dataset presented in [9]. We consider the following inference methods: iterative inference via Algorithm 1 (SV-DPGMM^{it1}); iterative inference via Algorithm 1 while sampling shading first (SV-DPGMM^{it2}); marginalized inference via Algorithm 2 (SV-DPGMM^{marg1}); marginalized inference via Algorithm 2 while sampling shading first (SV-DPGMM^{marg2}); and

marginalized split/merge inference via Algorithm 3 (SV-DPGMM). Additionally, we consider a procedure which replaces all sampling steps of Algorithm 3 with optimization (SV-DPGMM$^{\text{opt}}$). Table 2 summarizes the different inference schemes. We see that the methods based on Algorithms 1–2 are quite sensitive since their results vary dramatically based on whether the shading or reflectance is first estimated. In contrast, Algorithm 3 computes these jointly and does not suffer from this sensitivity. Since the training is based on *RS*-MSE, it is reasonable that SV-DPGMM does not perform the best across all metrics.

Next, we consider the following variants of the SV-DPGMM model: unsupervised training (SV-DPGMM$^{\text{unsup}}$); supervised training on a single Markov chain (SV-DPGMM$^{\text{single}}$); supervised training and computing the median across 10 Markov chains (SV-DPGMM); and SV-DPGMM with the color constancy post-processing (SV-DPGMM$^{\text{post}}$). Additionally, we compare to a model using a 10-component Dirichlet *distribution* mixture model instead of the Dirichlet *process* (SV-DPGMM$^{K=10}$). The results for SV-DPGMM$^{\text{single}}$ were obtained by averaging the *errors* for 10 Markov chains, instead of combining the 10 Markov chains with a median image. Table 3 summarizes results from the different variants. The unsupervised method generally performs worse than the supervised training. In principle, unsupervised learning has an advantage, in that it yields a set of parameters for each observed image. However, the sample space that includes the GP covariance kernel may be too complex to sufficiently explore. Combining multiple chains, using a Dirichlet *process*, and post-processing to enforce color constancy all improve results. We note that the *RS*-MSE does not change with post-processing since it is invariant to global shifts in any color channel.

7.2 Algorithm Comparison

Table 4 compares SV-DPGMM$^{\text{post}}$ with the Retinex algorithm and the method of [8] with ([8]+Ret.) and without ([8]−Ret.) Retinex. *S*-MSE is the only metric on which the SV-DPGMM yields worse performance. Upon examination of the individual results, we have found that this abnormally high error is due to making a large error in one of the shading estimate. We remind the reader that the only differences between SV-DPGMM and [8]−Ret. are the DP prior, a more expressive GP shading smoothness, and more robust inference. Moreover, many of the simplified inference algorithms of Tables 2–3 also outperform current methods. We believe that our optimization procedure for a more expressive model is only comparable to [8] due to the particular realization converging to a local extrema. Multiple initializations can circumvent this issue, as was done in [8].

Table 5 compares results with SIRFS [1] when training on half the images and testing on the other half. We note that published results from [1] and those obtained with their public source code are slightly different. SV-DPGMM performs better in three of the six metrics without needing to model the 3D scene geometry. We visualize results of each algorithm from the LOOCV training in Figure 4. In general, the reflectance image is more piecewise constant in color and there is less bleeding of the reflectance into the shading. Figure 5 shows additional images. SV-DPGMM occasionally makes large errors (e.g., first row

Table 2: Comparing SV-DPGMM Inference Methods

	S-MSE	R-MSE	RS-MSE	gS-MSE	gR-MSE	gRS-MSE
SV-DPGMM$^{\text{it1}}$	0.0548	0.0309	0.0362	0.0202	0.0196	0.0205
SV-DPGMM$^{\text{it2}}$	0.0532	0.0238	0.0302	0.0193	0.0146	0.0181
SV-DPGMM$^{\text{marg1}}$	0.0300	0.0146	0.0248	0.0097	0.0085	0.0121
SV-DPGMM$^{\text{marg2}}$	0.0321	0.0175	0.0271	0.0106	0.0109	0.0154
SV-DPGMM	0.0321	0.0144	0.0239	0.0093	0.0078	0.0111
SV-DPGMM$^{\text{opt}}$	0.0352	0.0172	0.0286	0.0120	0.0104	0.0157

Table 3: Comparing SV-DPGMM Model Variations

	S-MSE	R-MSE	RS-MSE	gS-MSE	gR-MSE	gRS-MSE
SV-DPGMM$^{\text{unsup}}$	0.0298	0.0166	0.0260	0.0096	0.0098	0.0136
SV-DPGMM$^{\text{single}}$	0.0328	0.0151	0.0249	0.0100	0.0087	0.0124
SV-DPGMM$^{K=10}$	0.0321	0.0147	0.0241	0.0095	0.0083	0.0120
SV-DPGMM	0.0321	0.0144	0.0239	0.0093	0.0078	0.0111
SV-DPGMM$^{\text{post}}$	0.0317	0.0135	0.0239	0.0072	0.0060	0.0111

Table 4: Leave-One-Out-Cross-Validation on 16 images from [9]

	S-MSE	R-MSE	RS-MSE	gS-MSE	gR-MSE	gRS-MSE
Retinex	0.0400	0.0292	0.0297	0.0219	0.0225	0.0185
[8]−Ret.	0.0311	0.0172	0.0304	0.0107	0.0134	0.0156
[8]+Ret.	0.0287	0.0205	0.0277	0.0119	0.0150	0.0166
SV-DPGMM$^{\text{post}}$	0.0317	0.0135	0.0239	0.0072	0.0060	0.0111

Table 5: Separate Train/Test Validation on 20 images from [9]

	S-MSE	R-MSE	RS-MSE	gS-MSE	gR-MSE	gRS-MSE
SIRFS Reported	-	-	-	0.0064	0.0098	0.0125
SIRFS Locally Run	0.0201	0.0158	0.0247	0.0068	0.0115	0.0125
SV-DPGMM	0.0306	0.0148	0.0229	0.0113	0.0092	0.0136
SV-DPGMM$^{\text{post}}$	0.0303	0.0141	0.0229	0.0092	0.0074	0.0136

in Figure 5), which are likely due to allowing color in the shading images. The prior could be changed on a per-image basis to correct these errors.

7.3 Sensitivity to Noise

Lastly, we consider the case of noisy observations. Images from [9] do not have any camera noise, so we inject artificial additive Gaussian noise in the observed image. We note that this synthetic noise does not contain the same noise characteristics assumed in SV-DPGMM, which models Gaussian noise in the *log* domain. Results for varying levels of noise variance are shown in Figure 6. This plot illustrates that SV-DPGMM, which explicitly characterizes noise, outperforms other methods in the noisy regime even with the model mismatch.

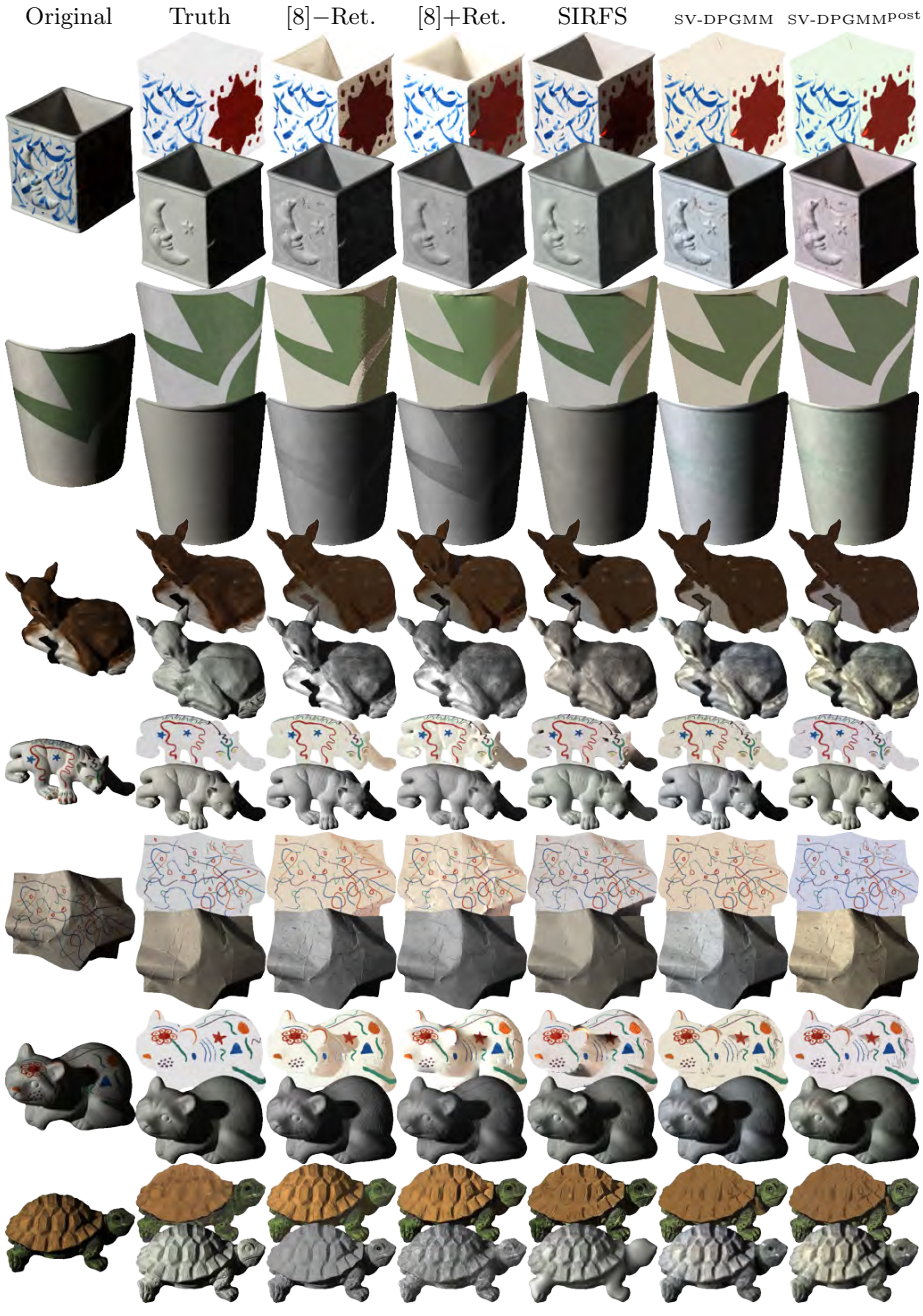

Fig. 4: Visual comparison of results. The rows show the estimated reflectance and shading images, respectively. SIRFS is trained via separate train/test sets. All other algorithms are trained using LOOCV.

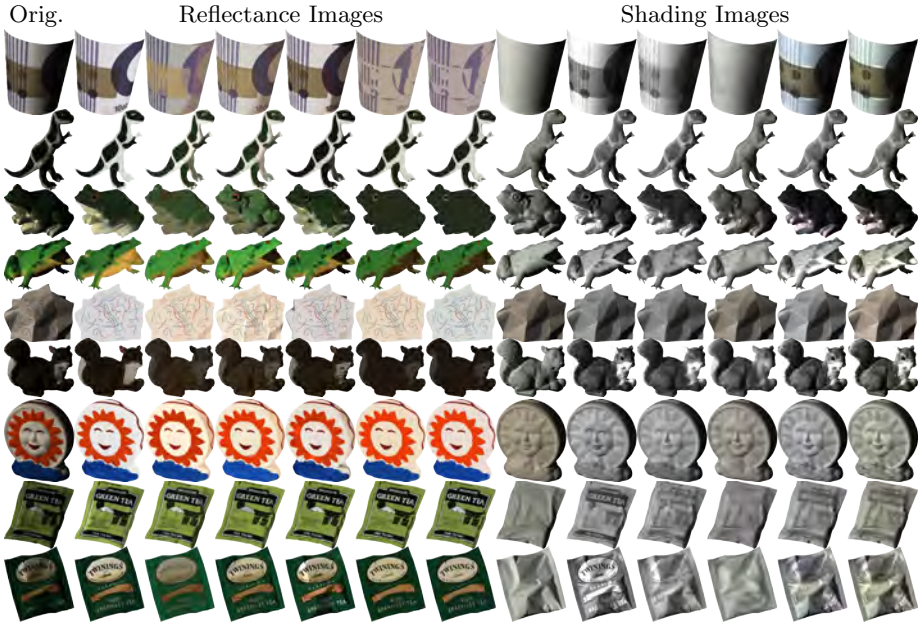

Fig. 5: Visual comparison of results. Left to right: original, reflectance images, and shading images. The reflectance and shading images from left to right: ground truth, [8]−Ret., [8]+Ret., SIRFS, SV-DPGMM, and SV-DPGMM$^{\text{post}}$.

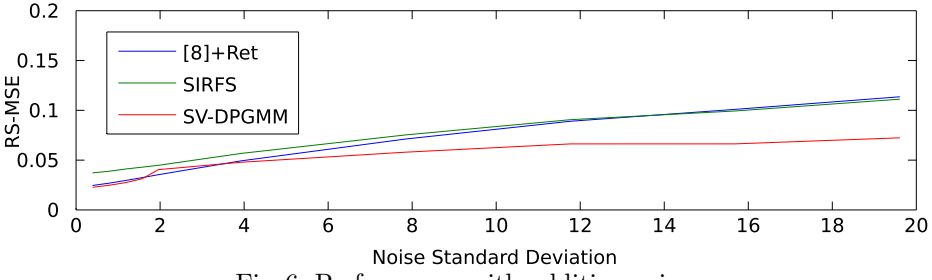

Fig. 6: Performance with additive noise.

8 Conclusion

We have presented the spatially-varying Dirichlet process Gaussian mixture model, a generative, Bayesian nonparametric model for intrinsic image decomposition. A Dirichlet process reflectance image is coupled with a Gaussian process shading image. Efficient marginalized MCMC inference results in state-of-the-art performance without modeling 3D geometry or using the Retinex term.

This research was partially supported by the Office of Naval Research Multidisciplinary Research Initiative (MURI) program, award N000141110688, and the Defense Advanced Research Projects Agency, award FA8650-11-1-7154.

References

1. Barron, J., Malik, J.: Shape, illumination, and reflectance from shading. Tech. rep., Univeristy of California, Berkeley (2013)
2. Barrow, H., Tenenbaum, J.: Recovering intrinsic scene characteristics from images. Computer Vision Systems (1978)
3. Blake, A.: Boundary conditions for lightness computation in Mondrian world. Computer Vision, Graphics, and Image Processing (1985)
4. Chang, J.: Sampling in Computer Vision and Bayesian Nonparametric Mixtures. Ph.D. thesis, Massachusetts Institute of Technology (2014)
5. Chang, J., Fisher, III, J.W.: Parallel sampling of DP mixture models using subclusters splits. In: Neural Information and Processing Systems (Dec 2013)
6. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision (2000)
7. Funt, B.V., Drew, M.S., Brockington, M.: Recovering shading from color images. In: European Conference on Computer Vision (1992)
8. Gehler, P.V., Carsten, R., Kiefel, M., Zhang, L., Schölkopf, B.: Recovering intrinsic images with a global sparsity prior on reflectance. In: Advances in Neural Information Processing Systems (2011)
9. Grosse, R., Johnson, M.K., Adelson, E., Freeman, W.T.: A ground-truth dataset and baseline evaluations for intrinsic image algorithms. In: International Conference on Computer Vision (2009)
10. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57(1), 97–109 (1970)
11. Horn, B.: Robot Vision. MIT Press, Cambridge, MA (1986)
12. Land, E., McCann, J.: Lightness and retinex theory. Journal of the Optical Society of America (1971)
13. Malioutov, D., Johnson, J., Choi, M., Willsky, A.: Low-rank variance approximation in gmrf models: Single and multiscale approaches. IEEE Transacations on Signal Processing 56(10), 4621–4634 (2008)
14. Matsushita, Y., Nishino, K., K., I., M., S.: Illumination normalization with time-dependent intrinsic images for video surveillance. IEEE Transacations on Pattern Analysis and Machine Intelligence 26(10), 1336–1347 (2004)
15. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge, MA (2006)
16. Sethuraman, J.: A constructive definition of Dirichlet priors. Statstica Sinica pp. 639–650 (1994)
17. Shen, L., Tan, P., Lin, S.: Intrinsic image decomposition with non-local texture cues. In: Computer Vision and Pattern Recognition (2008)
18. Shen, L., Yeo, C.: Intrinsic images decomposition using a local and global sparse representation of reflectance. In: Computer Vision and Pattern Recognition (2011)
19. Sollich, P., Williams, C.K.I.: Using the equivalent kernel to understand Gaussian process regression. In: Advances in Neural Information Processing Systems (2005)
20. Tappen, M.F., Freeman, W.T., Adelson, E.H.: Recovering intrinsic images from a single image. IEEE Transacations on Pattern Analysis and Machine Intelligence 27(9), 1459–1472 (2005)
21. Weiss, Y.: Deriving intrinsic images from image sequences. In: International Conference on Computer Vision (2001)
22. Zhao, Q., Tan, P., Dai, Q., Shen, L., Wu, E., Lin, S.: A closed-form solution to retinex with nonlocal texture constraints. IEEE Transactions on Pattern Analysis and Machine Intelligence (2012)