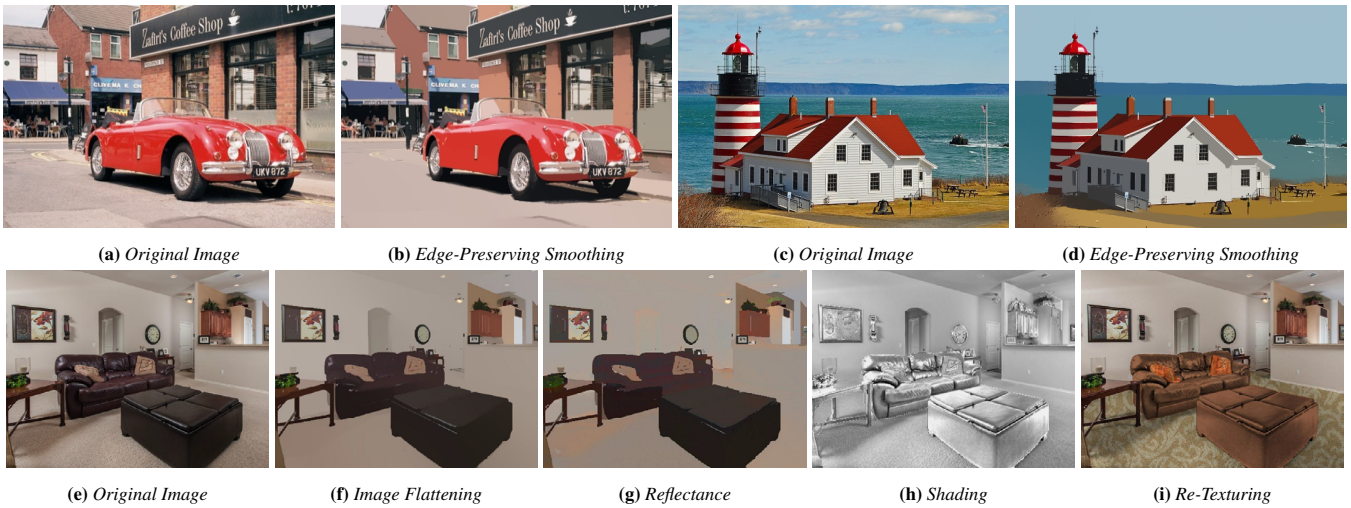
ACM Transactions on Graphics (special issue of SIGGRAPH 2015), 34(4).

An L_1 Image Transform for Edge-Preserving Smoothing and Scene-Level Intrinsic Decomposition

Sai Bi Xiaoguang Han Yizhou Yu

The University of Hong Kong



(a) *Original Image* **(b)** *Edge-Preserving Smoothing* **(c)** *Original Image* **(d)** *Edge-Preserving Smoothing*

(e) *Original Image* **(f)** *Image Flattening* **(g)** *Reflectance* **(h)** *Shading* **(i)** *Re-Texturing*

Figure 1: *Our algorithm for piecewise image flattening facilitates both edge-preserving smoothing and intrinsic decomposition. Two examples of edge-preserving smoothing are shown in (a)-(d), and one example of intrinsic decomposition is shown in (e)-(h). Intrinsic decomposition enables image editing, such as re-texturing (i). Original images courtesy Flickr users 47765927@N06 (a), 132341054@N03 (c) and 37213589@N08 (e).*

Abstract

Identifying sparse salient structures from dense pixels is a long-standing problem in visual computing. Solutions to this problem can benefit both image manipulation and understanding. In this paper, we introduce an image transform based on the L_1 norm for piecewise image flattening. This transform can effectively preserve and sharpen salient edges and contours while eliminating insignificant details, producing a nearly piecewise constant image with sparse structures. A variant of this image transform can perform edge-preserving smoothing more effectively than existing state-of-the-art algorithms. We further present a new method for complex scene-level intrinsic image decomposition. Our method relies on the above image transform to suppress surface shading variations, and perform probabilistic reflectance clustering on the flattened image instead of the original input image to achieve higher accuracy. Extensive testing on the *Intrinsic-Images-in-the-Wild* database indicates our method can perform significantly better than existing techniques both visually and numerically. The obtained intrinsic images have been successfully used in two applications, surface re-texturing and 3D object compositing in photographs.

CR Categories: I.4.3 [Image Processing and Computer Vision]: Enhancement—Smoothing; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Color, Shading;

Keywords: Salient Structures, Piecewise Image Flattening, Probabilistic Clustering, Intrinsic Images, Sparse Signal Recovery

1 Introduction

An image has hundreds of thousands of pixels. To effectively carry out an image manipulation or understanding task, such as image classification, segmentation, and stylization, it is vital to extract sparse salient structures, including perceptually important edges and contours, from such a large number of pixels. This is indeed the goal of contour detection and edge-preserving smoothing, which can be dated back to anisotropic smoothing [Perona and Malik 1990]. Ideally, the result of edge-preserving smoothing is a piecewise constant image with discontinuities occurring along salient edges. Spatial color variations that should be removed during such smoothing include both high-frequency and low-frequency signals. While high-frequency signals can be removed relatively easily with local filters, such as the bilateral filter [Tomasi 1998], low-frequency signals are more "stubborn" and removing them requires more global operations.

Interestingly, a similar challenge exists in intrinsic image decomposition, which attempts to separate reflectance and shading from their product, and has applications in surface re-texturing, object insertion and scene relighting. There is a recent trend on this topic to

shift from images of individual objects to images of complex scenes that may contain many surfaces and objects [Bell et al. 2014]. In comparison to images of objects, images of entire scenes are harder to analyze. An important reason is that shading could have low-frequency variations over a large surface with nearly uniform reflectance. A well received prior says nearby pixels with similar colors also have similar reflectance. However, small color differences between nearby pixels can accumulate and give rise to significant differences between distant pixels even when they have similar reflectance. For example, such significant differences could exist between distant points on a large surface with soft cast shadows that are smoothly varying. Such shading variations make reflectance image estimation very hard.

To tackle the aforementioned challenges, we introduce an image transform based on the L_1 norm in this paper. The goal of this transform is piecewise image flattening. If two neighboring pixels have similar colors, their transformed colors should be pulled even closer; if there exists a color discontinuity across two adjacent pixels, their transformed colors should be kept apart. This image transform is capable of reducing the color difference between two distant pixels if there are no salient edges, such as large reflectance changes, between them because it can reduce the color difference between any pair of adjacent pixels on a path between the distant pixels. In our formulation, the result of this image transform can be computed by iteratively solving linear systems.

A variant of this L_1-norm based image flattening can effectively preserve and sharpen salient edges and ridges while eliminating insignificant details, producing a nearly piecewise constant image with sparse structures. Such results approach the ideal outcome of edge-preserving smoothing.

We further develop a new solution pipeline for automatic scene-level intrinsic image decomposition. Our solution first performs probabilistic pixel clustering by sequentially applying the Dirichlet Process Gaussian Mixture Model (DPGMM) [Ferguson 1973] and probabilistic boosting trees [Tu 2005] to the pixel colors resulted from the above image transform. To further incorporate spatial coherence into clustering, we adopt a standard conditional random field with local pairwise connections to improve pixel labeling accuracy. We obtain final reflectance and shading images by solving an optimization formulated over superpixels. We have extensively tested our new solution on Cornell's large-scale scene-level database—*Intrinsic Images in the Wild* [Bell et al. 2014]. Experimental results indicate that our method can perform significantly better than existing techniques both visually and numerically. We have further successfully made use of the solved intrinsic images in two applications, surface re-texturing and 3D object compositing in photographs.

In summary, this paper has the following contributions.

• We introduce a new image transform based on the L_1 norm. This image transform is capable of flattening image regions with smooth color variations while preserving salient edges and contours. The transformed image is nearly piecewise constant. Such piecewise image flattening facilitates both edge-preserving smoothing and intrinsic image decomposition.

• We develop a new solution pipeline for automatic scene-level intrinsic image decomposition. This new pipeline is built upon the result from the above image transform. In this pipeline, we rely on probabilistic boosting trees to compute accurate pixelwise probabilities for CRF-based reflectance labeling. An optimization formulated over superpixels is also developed to better separate reflectance and shading images.

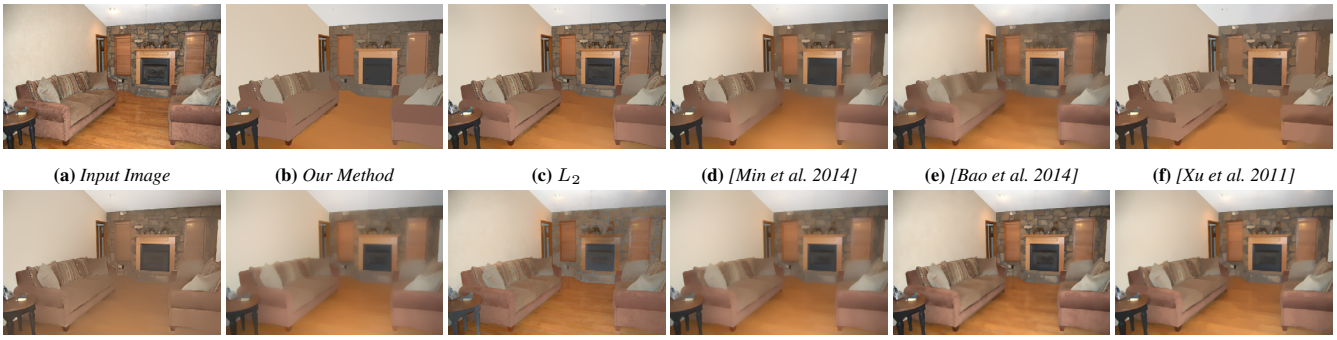
2 Related Work

Image Smoothing Edge-preserving image smoothing is a fundamental problem in image processing and computer graphics, and many algorithms [Rudin et al. 1992; Tomasi 1998; Farbman et al. 2008; Fattal 2009; Farbman et al. 2010; Paris et al. 2011; Xu et al. 2011; Bao et al. 2014; Min et al. 2014] have been developed to solve this problem in a local or global manner. Local approaches determine the final value of a pixel according to the pixels in a neighborhood while global approaches aim to remove insignificant details without destroying dominant structures by optimizing a global energy function. Among local methods, the bilateral filter [Tomasi 1998] replaces the original value of a pixel with a weighted average of nearby pixel values, and the weight is calculated according to similarity in both position and color. Farbman *et al.* [2008] introduce an edge-preserving operator based on a weighted least-squares optimization. Among global methods, Xu *et al.* [2011] locate a sparse set of important edges globally, and increase the steepness of transition at these edges while smoothing inessential details. Min *et al.* [2014] optimize a global objective function with a data term and a smoothness prior. Bao *et al.* [2014] extract a minimum spanning tree from the input image with pixels as nodes, and introduce a weighted tree filter where weights are determined by distance, color and pixel connectedness in the tree. Note that both our method and the method in [Xu et al. 2011] are based on edge sparsity. Since the method in [Xu et al. 2011] performs binary classifications of edge pixels according to global parameters, it tends to overlook and consequently blur locally important low-contrast edges while our method can still preserve such edges well.

Intrinsic Decomposition The notion of intrinsic images was first introduced in [Barrow and Tenenbaum 1978]. The Retinex theory [Land and McCann 1971] points out that any intensity change in an image is caused by either reflectance or shading, and in general large gradients correspond to reflectance changes while shading is smoother. In this spirit, Tappen *et al.* [2005] train a classifier to determine the cause of a pixel gradient. Extra constraints have also been introduced to improve the performance. Algorithms in [Shen et al. 2008; Zhao et al. 2012] assume that pixels with the same texture configuration have the same reflectance. Shen *et al.* [2011] propose that the reflectance at each pixel can be represented as a weighted average of its neighbors' reflectance values. More recently, Bonneel *et al.* [2014] separate gradients caused by reflectance and shading with a hybrid L_2 and L_p optimization.

Some recent developments adopt the global sparsity prior, which suggests any natural image is dominated by a relatively small set of material colors [Omer and Werman 2004]. Gehler *et al.* [2011] incorporate this prior by assuming that the reflectance value at each pixel is drawn from a mixture of Gaussian components referred to as basic colors, and builds a random field model based on this assumption. The algorithm in [Shen and Yeo 2011] enforces this constraint by minimizing the summation of pairwise absolute differences among a chosen set of pixels. Garces *et al.* [2012] cluster pixels into multiple groups with the K-means algorithm in the CIELab color space. To handle complex scene-level images, Bell *et al.* [2014] first estimate a set of distinct material colors using K-means and then perform pixel labeling with a fully connected conditional random field. Although these methods have reasonably good performance on images of real scenes, their clustering results are usually not sufficiently accurate due to shading variations. Our method in this paper tackles this challenge by performing piecewise image flattening to suppress shading variations before reflectance clustering.

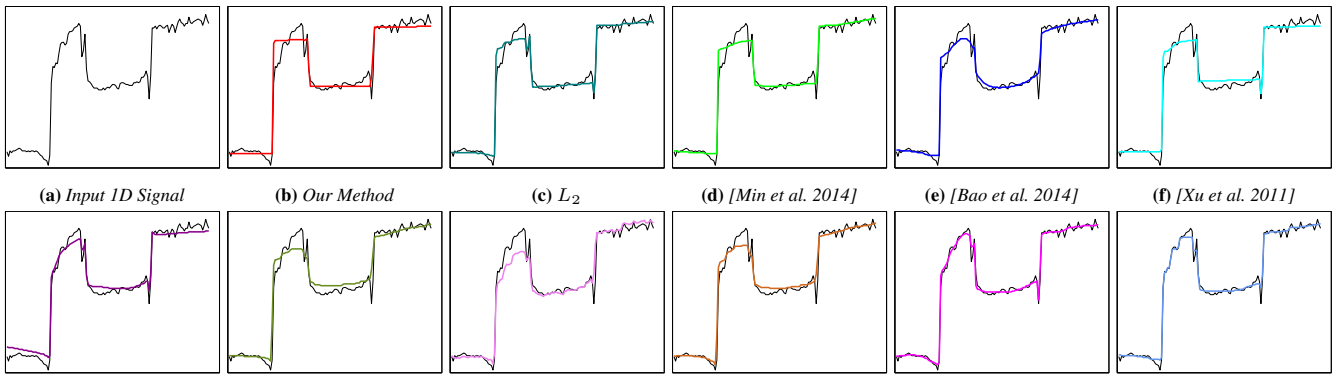
User interaction has been incorporated into intrinsic image decomposition. Bousseau *et al.* [2009] introduce constant-reflectance,



(a) *Input Image* **(b)** *Our Method* **(c)** L_2 **(d)** *[Min et al. 2014]* **(e)** *[Bao et al. 2014]* **(f)** *[Xu et al. 2011]*

(g) *[Paris et al. 2011]* **(h)** *[Farbman et al. 2010]* **(i)** *[Fattal 2009]* **(j)** *[Farbman et al. 2008]* **(k)** *[Tomasi 1998]* **(l)** *[Rudin et al. 1992]*

Figure 2: *Comparison between our piecewise image flattening and existing edge-preserving smoothing methods on an image of an indoor scene.*



(a) *Input 1D Signal* **(b)** *Our Method* **(c)** L_2 **(d)** *[Min et al. 2014]* **(e)** *[Bao et al. 2014]* **(f)** *[Xu et al. 2011]*

(g) *[Paris et al. 2011]* **(h)** *[Farbman et al. 2010]* **(i)** *[Fattal 2009]* **(j)** *[Farbman et al. 2008]* **(k)** *[Tomasi 1998]* **(l)** *[Rudin et al. 1992]*

Figure 3: *Comparison between our piecewise flattening and existing edge-preserving smoothing methods on a 1D signal.*

constant-illumination, and fixed-illumination brushes for interactively editing decomposition results. The same set of tools are also supported in [Shen et al. 2011; Bonneel et al. 2014]. Although automatic algorithms are better suited for processing large-scale datasets, their results inevitably have inaccuracies. Interactive techniques are complementary to automatic algorithms, and can be deployed to correct their mistakes.

Using depth or point clouds in intrinsic decomposition has attracted much attention recently [Chen and Koltun 2013; Laffont et al. 2012]. However, these methods are not applicable to our scenario, where we only use a single image of a scene and the image does not have depth information. Barron and Malik [2012a] have developed a unified approach to shape, shading, and reflectance estimation from a single image, and have achieved outstanding performance on the MIT Intrinsic dataset [Grosse et al. 2009]. Nonetheless, their assumption about object depth continuity does not hold on scene-level images, where there exist large depth discontinuities in general.

3 Piecewise Image Flattening

In this section, we introduce an image transform based on the L_1 norm to achieve piecewise image flattening. As a result, it would be easier to group pixels using the transformed pixel colors.

This image transform is cast as an energy minimization, which flattens image regions with smooth color variations while still achieving a reasonable approximation of the original image. The energy function of this minimization is a weighted sum of three energy terms as follows,

$$E = E_l + \alpha E_g + \beta E_a, \tag{1}$$

where E_l, E_g, and E_a refer to local flattening, global sparsity and image approximation terms respectively, and α, β are weights of the last two terms. These energy terms will be elaborated below.

Local Flattening Let $\mathbf{I}_i$ be the 3D RGB color vector at pixel p_i of the input image, $[l_i,\ a_i,\ b_i]^T$ be the corresponding color vector in the CIELab color space with the range of each channel normalized to $[0, 1]$, and $\mathbf{x}_i$ be the RGB color vector at p_i of the transformed image. The energy term for local flattening is defined as follows,

$$E_l = \sum_i \sum_{p_j \in N_h(p_i)} w_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|_1, \tag{2}$$

where $N_h(p_i)$ is a local $h \times h$ neighborhood of p_i, and the affinity between pixels p_i and p_j is defined as

$$w_{ij} = \exp\left(-\frac{||\mathbf{f}_i - \mathbf{f}_j||_2^2}{2\sigma^2}\right), \tag{3}$$

where $\mathbf{f}_i = [\kappa * l_i,\ a_i,\ b_i]^T$, κ and σ are constants. When $\kappa < 1$, it makes the energy term less sensitive to luminance variations. This setting is useful for scenarios where luminance variations need to be suppressed, as in the reflectance clustering stage of intrinsic image decomposition.

This energy term accumulates weighted L_1 differences between pixel pairs that are within each other's neighborhoods. Because neighboring pixels with similar colors in the input image are assigned larger weights, the L_1 differences of the transformed colors at such pixels are forced to be very small, giving rise to the desired flattening effect.

Suppose there are n pixels in the image and m_l neighboring pixel pairs in (2). Let $\mathbf{z_r}$ be the vector concatenating the R channel at all pixels of the transformed image. $\mathbf{z}_g$ and $\mathbf{z}_b$ are defined similarly, and are the concatenations of the G and B channels at all pixels. Let $\mathbf{z}$ be the concatenation of $\mathbf{z}_r$, $\mathbf{z}_g$ and $\mathbf{z}_b$. That is, $\mathbf{z}(=[\mathbf{z}_r^T\ \mathbf{z}_g^T\ \mathbf{z}_b^T]^T)$ is a $3n$-dimensional vector. Let $\mathbf{M}=\{M_{ij}\}$ be a $m_l \times n$ matrix, where $M_{ki}=w_{ij}, M_{kj}=-w_{ij}$ if p_i and p_j are neighboring pixels that form the k-th pixel pair. The energy term in (2) can be rewritten in a matrix-vector form as follows.

$$E_l = ||\mathbf{Lz}||_1, \quad \text{where } \mathbf{L}_{3m_l\times 3n} = \begin{pmatrix} \mathbf{M} & & \\ & \mathbf{M} & \\ & & \mathbf{M} \end{pmatrix}. \quad (4)$$

Global Sparsity As there are typically a limited number of distinct reflectance values in an image, for computational efficiency, we rely on superpixels to enforce such global sparsity. We generate a fixed number (this number is denoted as n_s) of superpixels using the algorithm in [Felzenszwalb and Huttenlocher 2004]. Within each superpixel, we choose a representative pixel whose color is closest to the average color of the superpixel. All such representative pixels form a set S_r, which is a subset of the pixels in the original image. The energy term for global sparsity is defined as follows,

$$E_g = \sum_{p_i\in S_r}\sum_{p_j\in S_r} w_{ij}\|\mathbf{x}_i-\mathbf{x}_j\|_1, \quad (5)$$

where w_{ij} is defined in (3). This energy term considers interaction between all pairs of representative pixels. As in (2), pixel pairs with similar colors in the input image are assigned larger weights to make it more likely for such pixels to have identical transformed colors, which in turn improves the global sparsity of the transformed pixel colors. This energy term is primarily used in intrinsic decomposition, and will be left out in edge-preserving smoothing.

Let $m_g(=n_s(n_s-1))$ be the total number of pairs among representative pixels, and $\mathbf{H}=\{H_{ij}\}$ be a $m_g \times n$ matrix, where $H_{ki}=w_{ij}$ and $H_{kj}=-w_{ij}$ if p_i and p_j are the k-th pair of representative pixels. Again, the energy term in (5) can be rewritten in a matrix-vector form as follows.

$$E_g = ||\mathbf{Gz}||_1, \quad \text{where } \mathbf{G}_{3m_g\times 3n} = \begin{pmatrix} \mathbf{H} & & \\ & \mathbf{H} & \\ & & \mathbf{H} \end{pmatrix}. \quad (6)$$

Image Approximation With the above energy terms, it is obvious that there exists a trivial solution where all pixels are assigned the same value. To avoid such trivial solutions, we further require that the transformed image have minimal deviation from the input image. This energy term for image approximation is defined as follows.

$$E_a = ||\mathbf{z}-\mathbf{z}^{in}||_2^2, \quad (7)$$

where $\mathbf{z}^{in}$ represents the concatenation of all pixel colors from the input image. With this term, transformed pixel colors are limited within a range around the original pixel colors, and therefore, cannot collapse to the same value.

Algorithm 1 Split-Bregman method for piecewise image flattening

1: **procedure** IMAGEFLATTEN(ϵ,λ)
2: **Initial:** $\mathbf{z}^0=\mathbf{z}^{in}; d_1^0, b_1^0, d_2^0, b_2^0=0;$
3: **while** $||\mathbf{z}^k-\mathbf{z}^{k-1}||_2^2>\epsilon$ **do**
4: $\mathbf{A}=\beta\mathbf{I}_{3n\times 3n}+\lambda\mathbf{L}^T\mathbf{L}+\lambda\alpha^2\mathbf{G}^T\mathbf{G}$
5: $\mathbf{v}=\beta\mathbf{z}^{in}+\lambda\mathbf{L}^T(d_1^k-b_1^k)+\lambda\alpha\mathbf{G}^T(d_2^k-b_2^k)$
6: Update $\mathbf{z}^{k+1}$ by solving $\mathbf{Az}^{k+1}=\mathbf{v}$
7: $d_1^{k+1}=\textbf{Shrink}(\mathbf{Lz}^{k+1}+b_1^k, \frac{1}{\lambda})$
8: $d_2^{k+1}=\textbf{Shrink}(\alpha\mathbf{Gz}^{k+1}+b_2^k, \frac{1}{\lambda})$
9: $b_1^{k+1}=b_1^k+\mathbf{Lz}^{k+1}-d_1^{k+1}$
10: $b_2^{k+1}=b_2^k+\alpha\mathbf{Gz}^{k+1}-d_2^{k+1}$
11: $k=k+1$
return $\mathbf{z}^k$
12: **procedure** SHRINK($\mathbf{y}$, γ)
return $\frac{\mathbf{y}}{\|\mathbf{y}\|}*\max(\|\mathbf{y}\|-\gamma, 0)$

L_1 Solver Since the L_1 norm appears in two of the energy terms in (1), a numerical solver that is capable of handling the non-differentiable L_1 norm is required. To this end, we have applied the Split Bregman method in [Goldstein and Osher 2009] to minimize our energy function efficiently. By introducing intermediate variables, the Split Bregman method solves an L_1 regularized optimization problem iteratively by transforming the original optimization into a series of differentiable unconstrained convex optimization problems. The definition of any convex optimization in the series depends on the intermediate variables passed from the previous iteration, and convergence can be achieved within a relatively small number of iterations.

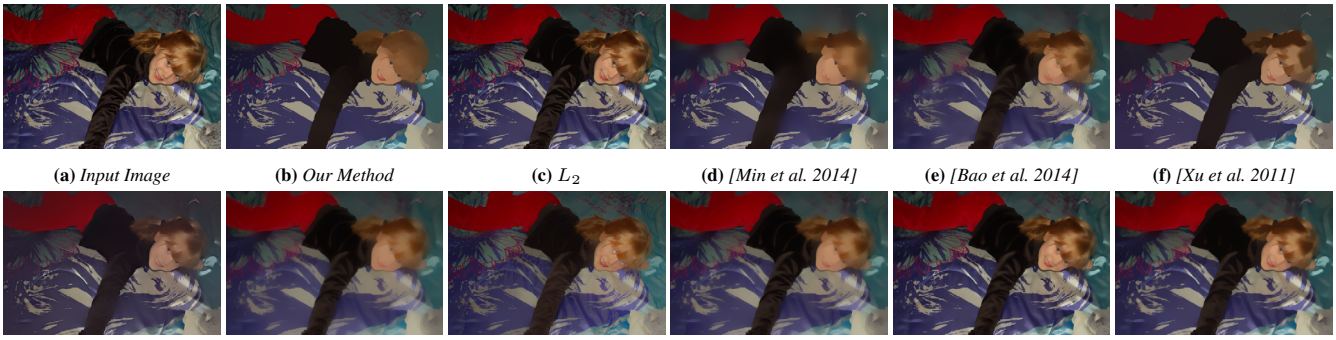
For the specific L_1 regularized minimization in our context, the convex optimization problems after transformation are actually in the form of linear least-squares problems, which can be expressed as follows.

$$\mathbf{z}^{k+1} = \underset{\mathbf{z}}{\text{argmin}} \Big(\beta\|\mathbf{z}-\mathbf{z}^{in}\|_2^2+\lambda\|d_1^k-\mathbf{Lz}-b_1^k\|_2^2 + \lambda\|d_2^k-\alpha\mathbf{Gz}-b_2^k\|_2^2\Big), \quad (8)$$

where b_1^k, b_2^k, d_1^k and d_2^k are the intermediate variables introduced by the Split Bregman method [Goldstein and Osher 2009]. The solution to (8) can be easily obtained by solving its corresponding normal equation, which is a sparse linear system. Algorithm 1 summarizes all the steps involved in minimizing the energy function in (1). The sparse linear system is defined in Lines 4-6. In this algorithm, there are two additional parameters, ϵ and λ, which

(a) *Input Image* **(b)** *Piecewise Flattening* **(c)** *Edge-Preserving Smoothing*

Figure 4: *Comparison between our original image flattening algorithm and the revised algorithm for edge-preserving smoothing.*



(a) *Input Image* **(b)** *Our Method* **(c)** L_2 **(d)** *[Min et al. 2014]* **(e)** *[Bao et al. 2014]* **(f)** *[Xu et al. 2011]*

(g) *[Paris et al. 2011]* **(h)** *[Farbman et al. 2010]* **(i)** *[Fattal 2009]* **(j)** *[Farbman et al. 2008]* **(k)** *[Tomasi 1998]* **(l)** *[Rudin et al. 1992]*

Figure 5: *Comparison of edge-preserving smoothing results from our method and other state-of-the-art methods. Input image courtesy Flickr user 11777792@N06.*

respectively control the termination of the iterations and the relative weight of the L_1 energy terms in the transformed least-squares problems.

3.1 Edge-Preserving Smoothing

Piecewise image flattening has different priorities in edge-preserving smoothing and intrinsic decomposition. For instance, while it is much desired to remove edges caused by abrupt shading changes during intrinsic decomposition, these edges should be preserved during edge-preserving smoothing if they are perceptually important. Such salient edges often have relatively large gradients. In addition, long thin ridges are visually salient and should be preserved during edge-preserving smoothing although they do not significantly affect the numerical accuracy of an intrinsic decomposition algorithm. Due to these considerations, we revise the pairwise affinity in (3) as follows to capture thin features with large gradients,

$$w_{ij} = \exp\left(-\frac{\max\left(||\mathbf{f}_i - \mathbf{f}_j||_2^2, \eta\hat{g}_{ij}^2\right)}{2\sigma^2}\right), \tag{9}$$

where $\hat{g}_{ij}$ represents the maximum gradient magnitude along the line segment between pixels p_i and p_j, and η is a weighting coefficient. The gradient of a color image is defined as the average gradient of individual RGB channels. In our experiments, we use the Sobel operator for gradient estimation.

As mentioned earlier, the global sparsity term in (5) is designed to limit the number of distinct reflectance values in an image. This energy term becomes irrelevant, and therefore, is removed from the energy function in (1) when edge-preserving smoothing is performed. As a result, it is unnecessary to generate superpixels for this global sparsity term either.

To demonstrate the necessity of the modification described in (9), Figure 4 shows a comparison between our original piecewise image flattening algorithm and the revised algorithm for edge-preserving smoothing. Figure 6 shows additional edge-preserving smoothing results from our revised algorithm. It can be seen that our method successfully removed inessential details while preserving dominant structures and salient edges in the input images. The following parameter setting is used for all edge-preserving smoothing results presented in this paper: $\beta = 2.5$, $\kappa = 0.3$, $\eta = 0.4$, $\sigma = 1.0$, $h = 11$, $\lambda = 5.0$, $\epsilon = 0.001$.

3.2 Comparisons

We have compared our image flattening and smoothing algorithms with many state-of-the-art algorithms for edge-preserving smoothing [Rudin et al. 1992; Tomasi 1998; Farbman et al. 2008; Fattal 2009; Farbman et al. 2010; Paris et al. 2011; Xu et al. 2011; Bao et al. 2014; Min et al. 2014]. Among them, the L_0 gradient minimization algorithm in [Xu et al. 2011] was developed with similar motivations as ours. In addition to these published algorithms, we have also compared against an L_2 version of our own algorithm by replacing the L_1 norm in (2) and (5) with the squared L_2 norm.

As shown in Figures 2 and 5, our algorithms have successfully and thoroughly removed minor color and shading variations in the images without blurring important features and contours, achieving a nearly piecewise constant image representation. On the other hand,

Figure 6: *Additional edge-preserving image smoothing results from our method. Left: input images; Right: smoothed images. Input image on the first row courtesy Flickr user 34517490@N00.*

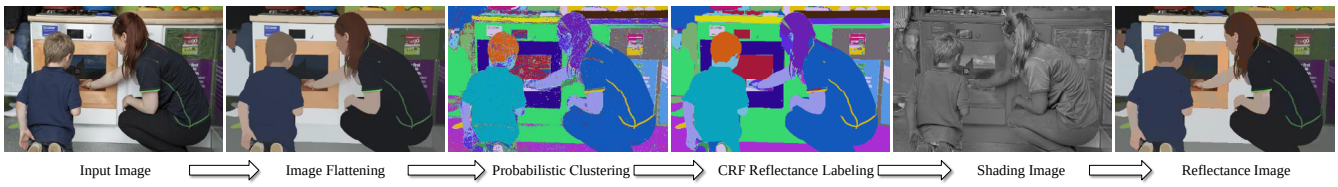


Figure 7: *Our pipeline for intrinsic image decomposition. Input image courtesy Flickr user 39796699@N02.*

other algorithms either mistakenly blurred originally sharp contours or left too much detailed variation intact. In particular, the L_2 version of our algorithm kept more color and shading variations than the L_1 version. Although the L_0 norm is optimal for edge preservation, energy functions formulated using the L_0 norm are extremely hard to optimize. Therefore, the method in [Xu et al. 2011] relies on a truncated L_2 norm to approximate the L_0 effect, giving rise to less than ideal performance.

We have also conducted experiments on 1D signals to make the results more intuitive. As shown in Figure 3, our original flattening algorithm has clearly transformed the input signal into four almost constant segments. In comparison, although other methods have also achieved edge-preserving smoothing effects, they do not have the same flattening performance as ours. Note that the result from the algorithm in [Min et al. 2014] also has four straight segments only without any spikes, but two of them have nonzero slopes, giving rise to the type of smooth variations visible in Figure 2(d).

Note that our energy formulation for image flattening is somewhat similar to the energy function used in image edit propagation [An and Pellacini 2008]. The most obvious difference is that we use the L_1 norm in the local flattening and global sparsity terms while existing work on edit propagation typically uses the squared L_2 norm throughout its energy function. In fact, such a small difference has the following important implications. *First*, according to the theory of sparse signal recovery [Donoho and Stark 1989; Donoho and Logan 1992], the L_1 norm promotes sparse solutions while the L_2 norm does not. A sparse solution in our context implies that there only exist a sparse set of locations where the difference between two adjacent pixels is sufficiently large. This is consistent with the sparsity priors on edges, contours and reflectance. *Second*, for a pair of neighboring pixels with similar colors, their color difference in the solution obtained using the L_1 norm is smaller than that using the L_2 norm squared; however, for neighboring pixels with a color discontinuity, an opposite conclusion can be reached. Thus, the L_1 norm makes similar colors even closer while keeping dissimilar colors apart, which eventually gives rise to the piecewise flattening effect. The reason behind this phenomenon is that the L_1 norm of the difference between two similar colors occupies a larger proportion of the total energy than the squared L_2 norm of the same color difference and the opposite is true for two dissimilar colors.

4 Intrinsic Image Decomposition

Our algorithm for intrinsic image decomposition works as follows. First, the algorithm for piecewise image flattening in Section3 is performed on the input image to obtain a transformed image. Second, probabilistic pixel clustering is performed on the transformed image such that most pixels in the same cluster share similar reflectance. The number of clusters is automatically determined through an existing self-adaptive clustering algorithm. A probabilistic discriminative classifier is further trained to better predict the probability of each pixel belonging to a certain cluster. Given the clusters as well as pixelwise probabilities, we enhance spatial coherence by relabeling the pixels using a conditional random field. Finally, we solve reflectance by enforcing shading smoothness, and finalize the reflectance and shading intrinsic images.

4.1 Clustering

We apply a self-adaptive probabilistic clustering method called Dirichlet Process Gaussian Mixture Model (DPGMM) [Ferguson 1973; Blei and Jordan 2006] to automatically determine a proper number of pixel clusters, as in [Chang et al. 2014]. DPGMM is an infinite mixture model with the Dirichlet Process as a prior distribution over the number of clusters. To minimize the interference of shading variation on the clustering result, we perform clustering on the piecewise flattened image obtained using the algorithm in Section 3. Prior to clustering, RGB colors from the flattened image are first converted to the CIELab color space. Clustering is then performed on the following pixelwise vectors, $\mathbf{f}_c = [\kappa * l',\ a',\ b']^T$, where $[l',\ a',\ b']$ are CIELab color channels computed from the piecewise flattened image. Clustering with GMM needs to resolve a number of parameters including the mean vector and the covariance matrix. To reduce the possibility of being stuck in low-quality local minima, we only take the number of clusters from DPGMM while discarding its clustering result. Then we use the result of K-means clustering as an initialization for a subsequent GMM clustering step, which generates the final clustering result.

Since GMM is a parametric model that represents the overall distribution of multiple clusters together, it cannot accurately depict the boundaries of these clusters and may inadvertently merge tiny clusters with nearby large clusters. To alleviate such problems and generate more accurate pixelwise probabilities for the subsequent pixel labeling step, we use the GMM clustering result to train a multiclass discriminative classifier called a probabilistic boosting tree (PBT) [Tu 2005], where each node integrates a collection of weak classifiers to form a strong classifier. The number of classes in the PBT is equal to the number of clusters returned by DPGMM. We sample m_t pixels (in our implementation, m_t is always set to 30,000) from the image with half of them lying on cluster boundaries and the rest randomly chosen from image regions inside the clusters. The number of internal samples from each cluster is proportional to the cluster size. Among all Gaussians in the GMM, the class label of each sample is determined by the Gaussian with the largest probability density at the sample. The following quadratic basis formed using CIELab color channels computed from the original input image, $\mathbf{f}_p = [l^2,\ a^2,\ b^2,\ l*a,\ l*b,\ a*b,\ l,\ a,\ b]^T$, is adopted for representing the samples during PBT training and testing. A trained PBT predicts the posterior probability, $P_{i,k} = P(C_k|\mathbf{f}_{p_i})$, of a sample, $\mathbf{f}_{p_i}$, belonging to a cluster C_k.

4.2 Reflectance Labeling

We use a standard conditional random field (CRF) to perform spatially coherent reflectance labeling, which assigns a label to every pixel such that the same label is assigned to the subset of pixels sharing the same reflectance. The number of distinct reflectance

values is set equal to the number of pixel clusters automatically determined by DPGMM in the previous subsection.

Let $\phi(i)$ be the reflectance label of pixel p_i. The energy function of the CRF is defined as follows.

$$E(\phi) = E_{unary} + \gamma E_{pair}, \tag{10}$$

$$\text{where } E_{unary} = -\sum_i \log P_{i,\phi(i)}, \tag{11}$$

$$E_{pair} = \sum_i \sum_{p_j \in N_h(p_i)} w'_{ij} \tau(\phi(i), \phi(j)), \tag{12}$$

$$\text{where } w'_{ij} = \exp\left(-\frac{\|\mathbf{f}_i - \mathbf{f}_j\|_2^2}{2\sigma^2}\right), \tag{13}$$

$$\tau(\phi(i), \phi(j)) = \begin{cases} 1, & \phi(i) \neq \phi(j); \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

where $N_h(p_i)$ is a $h \times h$ neighborhood centered at pixel p_i, and $\mathbf{f}_i = [\kappa * l_i,\ a_i,\ b_i]^T$, where $[l_i,\ a_i,\ b_i]$ are CIELab color channels computed from the input image.

This energy function consists of two terms. The first term is the unary term that measures the consistency between the current labeling scheme and the posterior cluster membership probabilities generated by the PBT trained in the previous subsection. The second pairwise term measures the spatial coherence of the current labeling scheme by checking label consistency between neighboring pixels. While there exist multiple options to minimize this energy function, we adopt graphcuts [Zabih et al. 1999] to obtain the final reflectance labeling efficiently. Note that there exist more complex CRF models, such as the fully connected conditional random field [Krähenbühl and Koltun 2013] used in [Bell et al. 2014]. Nonetheless, due to the high-quality pixelwise posterior probabilities generated by our piecewise flattening and probabilistic clustering steps, a standard locally connected CRF can already produce decent reflectance labeling results.

4.3 Reflectance and Shading Estimation

The reflectance labeling result from the previous step indicates which subset of pixels should have similar reflectance. The actual reflectance still needs to be resolved. Since the reflectance of a surface made from the same material may still have minor variations, we do not force pixels with the same label having exactly the same reflectance. Instead, we divide an image into n'_s superpixels, each of which only consists of pixels with the same label, and allow reflectance to have minor differences across superpixels with the same label. We assume all pixels within the same superpixel share the same scalar reflectance. Let $\bar{I}_k$ be the average intensity of the pixels within superpixel q_k, $\bar{R}_k$ be the scalar reflectance of q_k, and $\bar{S}_k = \bar{I}_k / \bar{R}_k$ be the "average" scalar shading over q_k, In our algorithm, the scalar reflectance of a superpixel is an unknown, and these unknowns are solved by imposing the shading smoothness prior across neighboring superpixels and minimizing the following cost function.

$$\sum_{k \sim l} (\hat{S}_k - \hat{S}_l)^2 + \xi \sum_{k \sim l, \phi(k) = \phi(l)} (\hat{R}_k - \hat{R}_l)^2, \tag{15}$$

$$\hat{R}_k = \log \bar{R}_k,\ \hat{S}_k = \log \bar{I}_k - \hat{R}_k,$$

where $k \sim l$ means superpixels q_k and q_l are immediate neighbors, and ξ is a constant. The first term of the cost function enforces the smoothness of average scalar shading at adjacent superpixels, while the second term penalizes differences between scalar reflectance values at adjacent superpixels with the same label. Here we use the L_2 norm instead of the L_1 norm because the result from minimizing the least-squares cost tends to be smoother. This cost function can be efficiently minimized by solving the normal equation, which is a linear system.

There exist subtle but important differences between our formulation in (15) and those in existing methods such as [Bell et al. 2014; Garces et al. 2012]. First, shading smoothness constraints are imposed on neighboring superpixels in our formulation while they are imposed on neighboring pixels in existing methods. The reason for us to use superpixels is that superpixels are larger than pixels, and they can better tolerate potential small misalignment between true reflectance discontinuities and image region boundaries computed by pixel clustering or labeling. Second, the second energy term in (15) is an extra term that typically does not exist in existing methods. It is a soft penalty term that allows reflectance to have minor variations over superpixels with the same label. Since this setup is a better approximation of the reflectance of real-world surfaces, it can improve the quality of the estimated intrinsic images.

Let $\mathbf{I}_i = [I_i^1\ I_i^2\ I_i^3]$ be the color vector at pixel p_i in the original image. Once we have the scalar reflectance at every superpixel, the final reflectance and shading at p_i can be written as

$$\mathbf{R}_i = \frac{3R_i}{\sum_c I_i^c} \mathbf{I}_i, \tag{16}$$

$$\mathbf{S}_i = \frac{\sum_c I_i^c}{3R_i} [1\,1\,1]^T, \tag{17}$$

where the scalar reflectance R_i at p_i is set to $\bar{R}_k$ if p_i belongs to superpixel q_k.

5 Results on Intrinsic Decomposition

We adopt the large-scale public database, *Intrinsic Images in the Wild*, built in [Bell et al. 2014] as the benchmark for evaluation purposes. This database has 5230 manually annotated images of complex real indoor scenes. Most images in the database have 500×300 pixels each. A set of Poisson-disk-sampled points are chosen in each image, and these points are connected by edges in the Delaunay triangulation. With this process, each image contains 44 ± 16 points and 106 ± 45 edges. Between every pair of connected points, users were invited to evaluate which point has a darker surface color or they have the same level of brightness. Figure 8a shows the sample points and judgements on an example image, where an arrow-shaped edge indicates that the point at the arrowhead has a darker surface reflectance, and an edge with uniform thickness indicates that the surface reflectances at the two endpoints have approximately the same magnitude. In addition, the color of an edge reflects the level of confidence on the edge label with the blue color for higher confidence and orange for lower confidence. A new error metric called "weighted human disagreement rate"(WHDR) has been introduced in [Bell et al. 2014] to evaluate intrinsic decomposition results. This metric measures the level of agreement between the judgements made by algorithms being evaluated and those of humans.

5.1 Parameter Setting

We have fully implemented our intrinsic decomposition algorithm. In general, it takes 5-10 minutes for our MATLAB code to process an image with 500×300 pixels on an Intel 3.4GHz processor. Our code could be significantly accelerated through multicore parallel computing. We ran our algorithm with different parameter

(a) *Original Image* **(b)** *Our Method:* 5.2% **(c)** *[Bell et al. 2014]:* 10.6% **(d)** *[Shen et al. 2011]:* 22.4% **(e)** *[Zhao et al. 2012]:* 25.8%

(f) *Baseline Reflectance:* 26.1% **(g)** *[Garces et al. 2012]:* 26.9% **(h)** *Color Retinex:* 29.3% **(i)** *[Bonneel et al. 2014]:* 31.3% **(j)** *Grayscale Retinex:* 35.5%

Figure 8: *Comparison of intrinsic image decomposition results from our method and other state-of-the-art methods.*

settings on a validation dataset formed by 300 randomly chosen images from the above database, and found out that the following parameter setting works the best on the validation dataset: $\alpha = 0.01$, $\beta = 2.5$, $\sigma = 1.0$, $\kappa = 0.3$, $h = 11$, $n_s = 500$, $\lambda = 5.0$, $\epsilon = 0.001$, $\gamma = 2.7$, $n_s' = 2500$, $\xi = 30$. We simply fixed our parameters in this setting in all subsequent experiments.

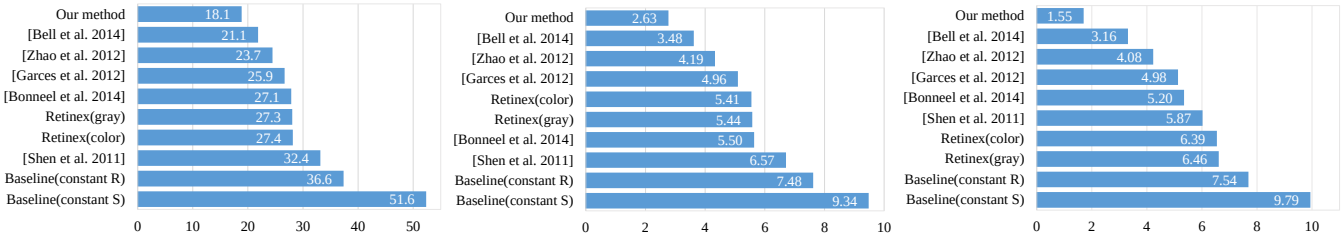
5.2 Comparison with State-of-the-Art Methods

As shown in Figure 9, we have compared our intrinsic decomposition algorithm with several state-of-the-art methods on the aforementioned image database of real scenes. We use the same criteria as in [Bell et al. 2014] when evaluating the performance of these algorithms. These criteria are Mean $\text{WHDR}_{10\%}$ over all edges, Mean rank of $\text{WHDR}_{10\%}$ over all edges, as well as Mean rank of $\text{WHDR}_{10\%}$ over all edges in images with densely sampled points. As in [Bell et al. 2014], to measure a testing error for each image, the optimal parameter setting for each existing algorithm is obtained through leave-one-out cross-validation. Our algorithm achieves the best performance under all three criteria among all the methods we evaluated. This is consistent with our visual inspection of the results. In particular, the advantage of our method becomes more significant when it comes to the densely sampled images, and in this case our mean rank rises to 1.55. The reason for such an impressive performance is that our method preserves important features and boundaries very well in the decomposition results due to the edge-preserving nature of our image flattening algorithm.

Figure 8 shows a visual comparison among all methods we evaluated. The input image is challenging in that it has large shading variations such as highlights and shadows, and the scene is also quite complex. From the results we can clearly see that our method is better at dealing with highlights and soft shadows. For instance, soft shading variations on the curtains (caused by the folds) as well as highlights on the sofa have been mostly preserved in the shading image while results from other methods, such as Figures 8c, 8d and 8e, still keep them in the reflectance image. The reason behind this is that our image flattening algorithm can successfully suppress color variations caused by shadows and highlights, giving rise to more accurate clustering and labelling results. Moreover, our method has also achieved better shading continuity as well as reflectance sparseness.

5.3 Comparisons among Design Choices

Since our algorithm has multiple steps and each step has multiple design choices, we have also performed an ablation study. Individual results in this study were obtained by replacing or removing one component only while keeping all other components of our pipeline untouched.



(a) *Mean $WHDR_{10\%}$ (in percentage) over all edges* **(b)** *Mean rank of $WHDR_{10\%}$ over all edges* **(c)** *Mean rank of $WHDR_{10\%}$ over dense edges only*

Figure 9: *Comparison with state-of-the-art intrinsic decomposition methods on the Intrinsic-Image-in-the-Wild dataset.*

(a) *Mean $WHDR_{10\%}$ over all edges* **(b)** *Original Image* **(c)** *Our Full Pipeline* **(d)** *Shading without Superpixels* **(e)** *GMM Probability*

Figure 10: *Comparisons among different design choices. Original image in (b) courtesy Flickr user 45716136@N00.*

L_0, L_1 and L_2 image flattening As discussed earlier, the L_0 gradient minimization algorithm in [Xu et al. 2011] and the L_2 version of our algorithm share similar motivations as our L_1 flattening algorithm. Therefore, we have compared these image flattening methods using the final WHDR scores they can help achieve. The WHDR score achieved without any image flattening serve as the baseline. As shown in Figure 10, the flattening algorithm based on the L_1 norm obtained the lowest mean error. Also, the performance drop would be 1.8 if we completely removed the flattening step from our pipeline.

CIELab vs RGB In Section 3, when computing pixel affinity in the local flattening term, we use the CIELab color space and a lower weight for the lightness channel. We have compared this scheme against an alternative using the RGB color space and equal weights among the three channels, and the result in Figure 10 indicates the proposed scheme performs better than the alternative. The reason is that by assigning a lower weight to lightness, we can reduce the effect of shading variation and obtain a better flattening result.

DPGMM vs Fixed number of clusters We use DPGMM to determine the number of pixel clusters adaptively. The result in Figure 10 indicates this scheme achieves a slightly lower error rate than GMM with a fixed number of clusters (20 in our experiment) for every image.

DPGMM vs PBT clustering In Section 4.1, instead of directly using the probability values from DPGMM clustering in the unary term of CRF-based labeling, we train a PBT classifier and use the probabilities from PBT in CRF-based labelling. Figure 10 compares the results from these two possible choices, and demonstrates the effectiveness of PBT. Misled by inaccurate probabilities generated by the GMM model, the CRF produce inaccurate labeling results, which further give rise to obvious artifacts (marked with red rectangles) in the final reflectance and shading images in Figure 10e. On the other hand, as a non-parametric discriminative classifier, PBT is capable of assigning input pixels more accurate labels and their corresponding probabilities, resulting in better labeling results and more accurate reflectance and shading images. The performance drop would be 2.7 if we took probabilities from the GMM and PBT were not used at all.

Locally and Fully connected CRFs In our CRF-based reflectance labeling, each pixel is only connected to other pixels in a small neighbourhood. We have evaluated a fully connected CRF [Krähenbühl and Koltun 2013] with the same unary and pairwise energy as in our locally connected CRF except that the feature vector $\mathbf{f}$ is augmented with pixel position. The result (Figure 10) shows that the fully connected CRF performs slightly worse than our locally connected version. Suboptimal parameters could be one possible reason because CRFs are sensitive to the choice of parameters. Since our locally connected CRF is a special case of the fully connected CRF, with a more exhaustive parameter search, the fully connected CRF might outperform our locally connected version. We leave this as future work.

Pixels vs Superpixels in shading estimation Different from previous methods [Bell et al. 2014; Garces et al. 2012], which impose the shading smoothness prior on neighboring pixels, we have chosen to generate superpixels containing pixels with the same reflectance label. A comparison has also been conducted on these two choices, and the result in Figure 10 indicates that the use of superpixels can effectively lower the mean error, and generate more consistent shading. Moreover, Figure 10d shows the difference between these two choices visually. We can see that shading estimation with superpixels can achieve better shading continuity, especially at locations where there are large reflectance changes. In contrast, enforcing shading smoothness at the pixel level sometimes gives rise to large shading discontinuities between adjacent surfaces.

5.4 Limitations

Given the variety of scenes as well as illumination conditions in images, our method still has limitations and could be improved. First, although our image flattening algorithm can help preserve spread highlights in the shading image, highly concentrated highlights still cause trouble (Figure 11 top row). Second, high-frequency surface textures are sometimes factored into the shading image albeit they should belong to the reflectance image (Figure 11 middle row). This is a tradeoff made by our image flattening algorithm. In addition, many images in the database we use have colored illuminants, which is inconsistent with the well received assumption that shading is grayscale. A potential solution could deal with intrinsic image decomposition together with color constancy as in [Barron and Malik 2012a]. Moreover, the shading smoothness prior does not always hold, especially on images of real scenes where there are large depth variations and surface normal discontinuities (Figure 11 bottom row).

6 Applications

Surface Re-texturing Once an image has been decomposed into reflectance and shading images, it becomes straightforward to edit the material properties of objects in the image without affecting their shading. This is achieved by editing the reflectance layer first and then multiplying the shading layer back. This workflow can avoid flat and unrealistic results obtained with naive copy and paste (see the example at the top of Fig 12). Several surface re-texturing results are shown in Fig 12, where we also show the editing results obtained using the shading images generated by [Bell et al. 2014] for comparison. Our results look more realistic and have fewer artifacts especially in highlight areas (dashed red box in Fig 12) and soft shadow regions (dashed red box in Fig 12).

3D Object Compositing A second application we present here is the insertion of 3D objects into photographs. A crucial step in our pipeline for completing this task is illumination estimation from the shading image. We estimate illumination from shading using a simplified version of the method in [Barron and Malik 2012b]. It is

(a) *Original Image* **(b)** *Shading Image* **(c)** *Reflectance Image*

Figure 11: *Examples illustrating the limitations of our intrinsic decomposition method. Original images courtesy Flickr users 21723187@N04 (first row), 25186605@N04 (second row), and 22799676@N03 (third row).*

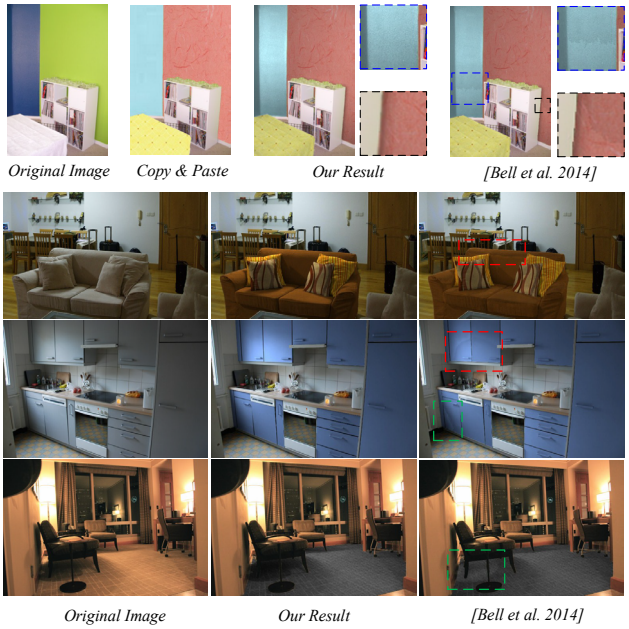


Figure 12: *Surface re-texturing. Original images courtesy FLickr users 31403417@N00 (second row), 21458229@N00 (third row), and 14146962@N07 (fourth row).*

formulated as the following optimization

$$\max_{Z,L} P(Z)P(L)\ s.t.\ S = \text{Shading}(Z, L), \tag{18}$$

where Z is the depth map and L is the illumination model represented with spherical harmonics. $P(Z)$ and $P(L)$ represent the same priors on depth and illumination as in [Barron and Malik 2012b]. We fix the shading S using the shading image obtained with our intrinsic decomposition algorithm. Shading(Z, L) is the rendering function. Although the depth map estimated from this optimization is usually of low quality, the estimated illumination is sufficiently accurate for 3D object compositing.

Consider inserting a 3D object into a region Ω (dashed yellow line in Fig 13) in an input photograph. The first step would be the selection of an area (dashed red region in Fig 13) close to Ω for illumination estimation. Because illumination in a real indoor scene is spatially varying, it is inaccurate to assume there exists a single illumination model for the entire shading image. In addition, the real surface in the selected area should own spatially varying normals to ensure the success of our illumination reconstruction. Once the illumination has been reconstructed, the 3D object is placed into the target position and rendered to generate the final composite. A comparison with [Bell et al. 2014] is also shown in Fig 13, where it can be observed that illumination estimation is sensitive to the accuracy of shading extraction, and the appearance of the inserted object rendered using our illumination is more consistent with the rest of the photograph. Fig 14 shows additional results of this application.

Note that our scheme for 3D object compositing is not entirely new. It serves as a place where we can demonstrate the quality of our intrinsic images. It also illustrates that object compositing is one of the tasks that could benefit from such results. There exist other more sophisticated methods [Karsch et al. 2011; Karsch et al. 2014] based on comprehensive scene geometry and illumination reconstruction. A significant difference from these methods is

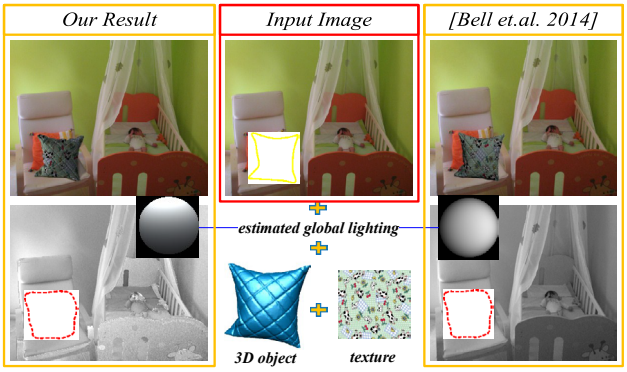


Figure 13: *3D object compositing. Input image courtesy Flickr user 15273615@N00.*

Figure 14: *Additional object compositing results. Image at the top of the 3rd column courtesy Flickr user 11540081@N05.*

that our scheme does not require geometry reconstruction for the whole scene.

7 Conclusions

We have presented an image transform based on the L_1 norm for piecewise image flattening. This transform can effectively preserve and sharpen salient edges and contours while eliminating insignificant details, producing a nearly piecewise constant image with sparse structures. A variant of this image transform performs edge-preserving smoothing more effectively than existing state-of-the-art methods. We have further presented a new algorithm for complex scene-level intrinsic image decomposition. Experiments on the database from [Bell et al. 2014] indicate our method performs significantly better than existing intrinsic decomposition techniques both visually and numerically. The obtained intrinsic images have also been successfully used in two applications, surface re-texturing and 3D object compositing in photographs.

Acknowledgements

We wish to thank Nicolas Bonneel for sharing the binary code of [Bonneel et al. 2014] and the anonymous reviewers for their valuable comments. This work was partially supported by Hong Kong Research Grants Council under General Research Funds (HKU719313).

References

An, X., and Pellacini, F. 2008. Appprop: all-pairs appearance-space edit propagation. *ACM Trans. Graph. 27*, 3, 40.

Bao, L., Song, Y., Yang, Q., Yuan, H., and Wang, G. 2014. Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree. *IEEE Transactions on Image Processing 23*, 2, 555–569.

Barron, J. T., and Malik, J. 2012. Color constancy, intrinsic images, and shape estimation. In *12th European Conference on Computer Vision*, 57–70.

Barron, J. T., and Malik, J. 2012. Shape, albedo, and illumination from a single image of an unknown object. *CVPR*.

Barrow, H. G., and Tenenbaum, J. M. 1978. Recovering intrinsic scene characteristics from images. In *CVS78*, 3–26.

Bell, S., Bala, K., and Snavely, N. 2014. Intrinsic images in the wild. *ACM Trans. Graph 33*, 4, 159.

Blei, D. M., and Jordan, M. I. 2006. Variational inference for dirichlet process mixtures. *Bayesian Analysis 1*, 1, 121–143.

Bonneel, N., Sunkavalli, K., Tompkin, J., Sun, D., Paris, S., and Pfister, H. 2014. Interactive intrinsic video editing. *ACM Trans. Graph 33*, 6, 197.

Bousseau, A., Paris, S., and Durand, F. 2009. User-assisted intrinsic images. *ACM Trans. Graph 28*, 5.

Chang, J., Cabezas, R., and III, J. F. 2014. Bayesian nonparametric intrinsic image decomposition. In *European Conference on Computer Vision*.

Chen, Q., and Koltun, V. 2013. A simple model for intrinsic image decomposition with depth cues. In *ICCV*, IEEE, 241–248.

Donoho, D., and Logan, B. 1992. Signal recovery and the large sieve. *SIAM J. Appl. Math. 52*, 2, 577–591.

Donoho, D., and Stark, P. 1989. Uncertainty principles and signal recovery. *SIAM J. Appl. Math. 49*, 3, 906–931.

Farbman, Z., Fattal, R., Lischinski, D., and Szeliski, R. 2008. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics 27*, 3 (Aug.), 67:1–67:??

Farbman, Z., Fattal, R., and Lischinski, D. 2010. Diffusion maps for edge-aware image editing. *ACM Trans. Graph 29*, 6, 145.

Fattal, R. 2009. Edge-avoiding wavelets and their applications. *ACM Trans. Graph 28*, 3.

Felzenszwalb, P. F., and Huttenlocher, D. P. 2004. Efficient graph-based image segmentation. *International Journal of Computer Vision 59*, 2 (Sept.), 167–181.

Ferguson, T. S. 1973. A bayesian analysis of some nonparametric problems. *The Annals of Statistics 1*, 2, 209–230.

Garces, E., Muñoz, A., Lopez-Moreno, J., and Gutierrez, D. 2012. Intrinsic images by clustering. *Comput. Graph. Forum 31*, 4, 1415–1424.

Gehler, P. V., Rother, C., Kiefel, M., Zhang, L., and Schölkopf, B. 2011. Recovering intrinsic images with a global sparsity prior on reflectance. In *25th Annual Conference on Neural Information Processing Systems.*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, Eds., 765–773.

Goldstein, T., and Osher, S. J. 2009. The split bregman method for L1-regularized problems. *Journal on Imaging Sciences 2*, 2, 323–343.

GROSSE, R. B., JOHNSON, M. K., ADELSON, E. H., AND FREEMAN, W. T. 2009. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*, IEEE, 2335–2342.

KARSCH, K., HEDAU, V., FORSYTH, D., AND HOIEM, D. 2011. Rendering synthetic objects into legacy photographs. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, ACM, New York, NY, USA, SA '11, 157:1–157:12.

KARSCH, K., SUNKAVALLI, K., HADAP, S., CARR, N., JIN, H., FONTE, R., SITTIG, M., AND FORSYTH, D. A. 2014. Automatic scene inference for 3d object compositing. *ACM Trans. Graph.*, 32.

KRÄHENBÜHL, P., AND KOLTUN, V. 2013. Parameter learning and convergent inference for dense random fields. In *International Conference on Machine Learning*.

LAFFONT, P.-Y., BOUSSEAU, A., PARIS, S., DURAND, F., AND DRETTAKIS, G. 2012. Coherent intrinsic images from photo collections. *ACM Trans. Graph 31*, 6, 202.

LAND, E. H., AND MCCANN, J. J. 1971. Lightness and retinex theory. *Journal of the Optical Society of America 61*, 1, 1–11.

MIN, D., CHOI, S., LU, J., HAM, B., SOHN, K., AND DO, M. N. 2014. Fast global image smoothing based on weighted least squares. *IEEE Transactions on Image Processing 23*, 12, 5638–5653.

OMER, I., AND WERMAN, M. 2004. Color lines: Image specific color representation. In *CVPR (2)*, 946–953.

PARIS, S., HASINOFF, S., AND KAUTZ, J. 2011. Local laplacian filters: Edge-aware image processing with a laplacian pyramid. *ACM Transactions on Graphics* (Aug.).

PERONA, P., AND MALIK, J. 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell. 12*, 7, 629639.

RUDIN, L. I., OSHER, S. J., AND FATEMI, E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena 60*, 259–268.

SHEN, L., AND YEO, C. 2011. Intrinsic images decomposition using a local and global sparse representation of reflectance. In *CVPR*, IEEE Computer Society, 697–704.

SHEN, L., TAN, P., AND LIN, S. 2008. Intrinsic image decomposition with non-local texture cues. In *CVPR*, 1–7.

SHEN, J., YANG, X., JIANG, Y., AND LI, X. 2011. *Intrinsic images using optimization.* Institute of Electrical and Electronics Engineers.

TAPPEN, M., FREEMAN, W., AND ADELSON, E. 2005. Recovering intrinsic images from a single image. *IEEE Trans. Pattern Anal. Mach. Intell 27*, 1459–1472.

TOMASI. 1998. Bilateral filtering for gray and color images. In *ICCV*, 839–846.

TU, Z. W. 2005. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *ICCV*, II: 1589–1596.

XU, L., LU, C., XU, Y., AND JIA, J. 2011. Image smoothing via L_0 gradient minimization. *ACM Transactions on Graphics 30*, 6 (Dec.), 174:1–174:??

ZABIH, R., VEKSLER, O., AND BOYKOV, Y. Y. 1999. Fast approximate energy minimization via graph cuts. In *ICCV*, 377–384.

ZHAO, Q., TAN, P., DAI, Q., SHEN, L., WU, E., AND LIN, S. 2012. A closed-form solution to retinex with nonlocal texture constraints. *IEEE Trans. Pattern Anal. Mach. Intell 34*, 7, 1437–1444.