

Ground truth dataset and baseline evaluations for intrinsic image algorithms

Roger Grosse

Micah K. Johnson

Edward H. Adelson

William T. Freeman

Massachusetts Institute of Technology
Cambridge, MA 02139

rgrosse@gmail.com

Abstract

The intrinsic image decomposition aims to retrieve “intrinsic” properties of an image, such as shading and reflectance. To make it possible to quantitatively compare different approaches to this problem in realistic settings, we present a ground-truth dataset of intrinsic image decompositions for a variety of real-world objects. For each object, we separate an image of it into three components: Lambertian shading, reflectance, and specularities. We use our dataset to quantitatively compare several existing algorithms; we hope that this dataset will serve as a means for evaluating future work on intrinsic images.

1. Introduction

The observed color at any point on an object is influenced by many factors, including the shape and material of the object, the positions and colors of the light sources, and the position of the viewer. Barrow and Tenenbaum [1] proposed representing distinct scene properties as separate “intrinsic” images. We focus on one particular case: the separation of an image into *illumination*, *reflectance*, and *specular* components. The illumination component accounts for shading effects, including shading due to geometry, shadows and interreflections. The reflectance component, or albedo, represents how the material of the object reflects light independent of viewpoint and illumination. Finally, the specular component accounts for highlights that are due to viewpoint, geometry and illumination. All together, this decomposition is expressed as

$$I(x) = S(x)R(x) + C(x), \quad (1)$$

where $I(x)$ is the observed intensity at pixel x , $S(x)$ is the illumination, $R(x)$ is the albedo, and $C(x)$ is the specular term.

Such a decomposition could be advantageous for certain computer vision algorithms. For instance, shape-from-shading algorithms could benefit from an image with only

shading effects, while image segmentation would be easier in a world without cast shadows. However, estimating this decomposition is a fundamentally ill-posed problem: for every observed value there are multiple unknowns. There has been progress on this decomposition both on a single image and from image sequences [16, 15, 12]. Researchers have shown promising results on their own sets of images; what has been missing from work thus far is detailed comparisons to other approaches and quantitative evaluation.

One reason for this shortcoming is the lack of a common dataset for intrinsic images. Other computer vision problems, such as object recognition and stereo, have common datasets for evaluation and comparison [4, 11], and the citation count for these works is anecdotal evidence that the existence of such datasets has contributed to progress on these problems.

In this work, we provide a ground-truth dataset for intrinsic images. This dataset will facilitate future evaluation and comparison of intrinsic image decomposition algorithms. Using this dataset, we evaluate and compare several existing methods.

2. Previous Work

Vision scientists have long been interested in understanding how humans separate illumination and reflectance. Working in a Mondrian (i.e., piecewise constant) world, Land and McCann proposed the Retinex theory [8], which showed that albedos could be separated from illumination if the illumination was assumed to vary slowly. Under Retinex, small gradients are assumed to correspond to illumination and large gradients are assumed to correspond to reflectance. In later work, Horn showed how to recover the albedo image using these assumptions [7].

While Retinex works well in a Mondrian world, the assumptions it makes do not hold for all real-world images. Later work has focused on methods for classifying edges as illumination or reflectance according to different heuristics. Sinha and Adelson considered a world of

painted polyhedra and searched for globally consistent labelings of edges or edge junctions as caused by shape or reflectance changes [13]. Bell and Freeman and also Tappen et al. used machine learning to interpret image gradients from local cues [2, 15]. Later, Tappen et al. demonstrated promising results using a non-linear regression based on local, low-dimensional estimators [14]. More recently, Shen et al. showed that Retinex-based algorithms can be improved by assuming similar textures correspond to similar reflectances [12].

In some scenarios, multiple images of the same scene under different illumination conditions are available. Weiss showed that, in this case, the reflectance image can be estimated from the set of images by assuming a random (and sparse) distribution of shading derivatives [16].

As far as establishing ground-truth for intrinsic images, Tappen et al. created small sets of both computer-generated and real intrinsic images. The computer-generated images consisted of shaded ellipsoids with piecewise-constant reflectance [15]. The real images were created using green marker on crumpled paper [14]. Beyond this, we know of no other attempts to establish ground truth for intrinsic images.

3. Methods

Our main contribution is a dataset of images of real objects decomposed into Lambertian shading, Lambertian reflectance, and specularities according to Eqn. (1). The specular term accounts for light rays that reflect directly off the surface, creating visible highlights in the image. The diffuse term corresponds to Lambertian reflectance, representing light rays that enter the surface, reflect and refract, and leave at arbitrary angles. In addition, we assume that the diffuse term can be further decomposed into shading and reflectance terms, where the shading term is the image that would occur if a uniform-reflectance version of the object was in the same illumination conditions. These assumptions constrain the types of effects that can occur in the three component images of Eqn. (1) and suggest a workflow for obtaining these images.

Note that most intrinsic image work is only interested in recovering *relative* shading and reflectance of a given scene. In other words, the red, green, and blue channels of an algorithm’s estimated reflectance (shading) image are each allowed to be any scalar multiple of the true reflectance (shading). Therefore, we provide only relative shading and reflectance. All visualizations in this paper assume white light (i.e. a grayscale shading image), but this does not matter for the algorithms or results.

We begin by separating the diffuse and specular components. We use a cross-polarization approach where a polarizing filter is placed over both the light and camera [10]. With this approach, the light rays that reflect off the surface

of an object to form highlights remain polarized. Therefore, the highlights can be removed by placing a polarizing filter on the camera and rotating it until its axis is orthogonal to the axis of the reflected light. We position a small chrome sphere in the scene to gauge the appropriate position of the camera’s polarizing filter. With the filter in the position that removes highlights, we capture a diffuse image of the object.

3.1. Separating shading and reflectance

We have developed two different methods for separating the diffuse component into shading and reflectance. The first method begins with an object that has existing reflectance variation. In this case, we remove the albedo variation by painting the object with a thin coat of flat spray-paint and re-photographing it under the same illumination. The second method is appropriate for objects that begin with a uniform color, such as unglazed ceramics and paper. For these objects, we apply colored paint or marker and re-photograph them under the same illumination. In both methods, with the polarizing filter set to remove specularities, we obtain a diffuse image of the object with reflectance variation (I_{lamb}) and another without reflectance variation (I_{uni}). Additionally, with the filter set to maximize specularities, we capture another photograph of the object with reflectance variation (I_{orig}). Examples illustrating this process are shown in Figure 1.

Because we are interested in relative, rather than absolute, shading and reflectance, we may use I_{uni} as the shading image. In principle, the terms in Eqn. (1) can be computed as follows:

$$S = I_{uni}, \tag{2}$$

$$R = I_{lamb}/I_{uni}, \tag{3}$$

$$C = (I_{orig} - I_{lamb})/2. \tag{4}$$

In practice, when using a single light direction, the ratio in (3) can be unstable in shadowed regions. Therefore, we instead capture an additional pair of diffuse images I'_{lamb} and I'_{uni} with a different lighting direction and compute

$$R = (I_{lamb} + I'_{lamb})/(I_{uni} + I'_{uni}). \tag{5}$$

3.2. Alignment

In order to do pixelwise arithmetic on our images, the images must be accurately aligned. Alignment is particularly challenging for the objects that require spraypainting, as the objects must be removed from the scene to be painted and then replaced in their exact positions.¹

To re-position objects with high accuracy, we attach the objects to a solid platform and rest the platform on a mount

¹Spraypainting the objects in place could result in paint on the ground plane and would affect the ambient illumination near the object.

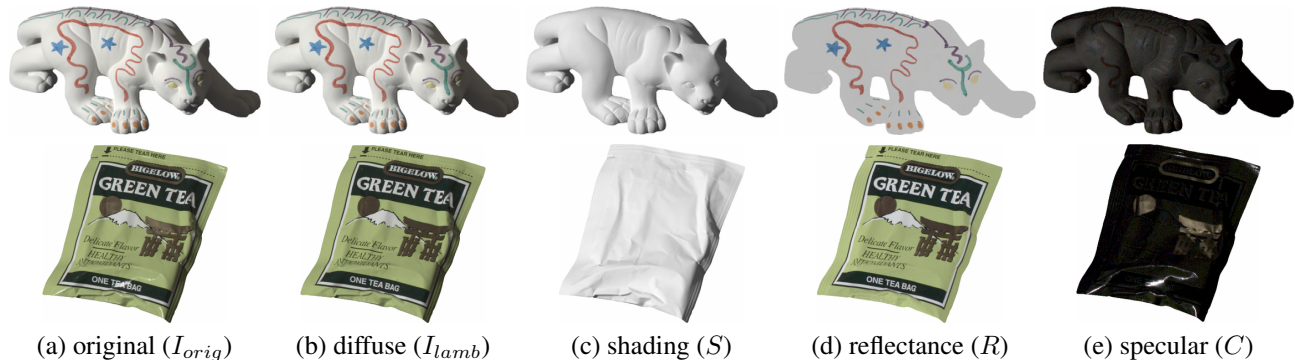


Figure 1. Capturing ground-truth intrinsic images. (a) We capture a complete image I_{orig} using a polarizing filter set to maximize specularities and (b) a diffuse image I_{lamb} with the filter set to remove specularities. (c) We paint the object to obtain the shading image S . From these images, we can estimate (d) the reflectance image R and (e) specular image C . The images shown here were captured using a linear camera and have been contrast-adjusted to improve visibility.

created by a set of four spheres, as in Fig. 2. To create the mount, we affix three spheres to the ground plane so that they all touch. A fourth sphere can rest in the recess between the three spheres. We duplicate this mount in several locations across the ground plane and affix the platform to the fourth sphere in each mount. Using this system, we find that we can remove and replace objects in the scene with no noticeable shift between images.

Although there is no object motion, there is the possibility of camera motion. We have found that a traditional polarizing filter, attached to the end of the lens, can cause several pixels of shift between images when rotated. Our polarizing filter is therefore attached to a harness placed in front of the lens (on a separate tripod) so that it can be rotated without touching the camera.

For accidental camera motion, we redo the experiment if the motion is larger than 1 pixel, or manually align the images when the motion is smaller than 1 pixel. To translate an image without loss of resolution, we exploit the shift property of the Fourier transform: a translation in space can be applied by multiplying the Fourier transform of the image with a complex sinusoid with linearly varying phase. There is no loss of frequency resolution since the magnitudes of the Fourier transform are not modified. We avoid wrap-around artifacts by cropping the images.

3.3. Interreflections

By defining the reflectance image as in (3), we assume the illumination at each point is constant between these two images, and the only difference is the albedo. However, the illumination itself changes as a result of light interreflected between different objects and/or different parts of the same object. We have minimized indirect illumination by covering the ground plane and all nearby walls with black paper. We have also removed objects from the dataset that have large artifacts due to interreflections.

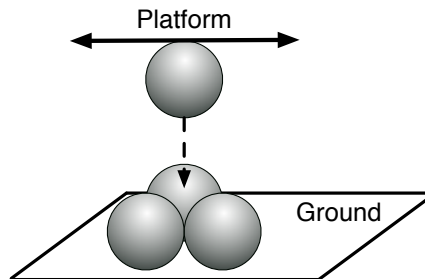


Figure 2. System for accurately re-positioning objects. Three spheres are affixed to the ground plane so that they all touch. A fourth sphere is attached to a platform that supports the object being photographed. The position of the platform is constrained by placing the fourth sphere in the recess between the three bottom spheres. The platform is supported by multiple sets of spheres in this arrangement.

3.4. Multiple lighting conditions

Some algorithms for estimating intrinsic images, such as [16], require a sequence of photographs of the same scene with many illumination conditions. In addition to the two fixed light positions, we captured diffuse images with ten more light positions using a handheld lamp with a polarizing filter. For each position of the light, the polarizing filter on the camera was adjusted to minimize the specularity on a chrome sphere placed in the scene.

4. Experiments

In this section, we quantitatively compare several existing approaches to estimating intrinsic images. We first compare algorithms which use only single grayscale images, and then look at how much can be gained by using additional sources of information, such as color or additional images of the same scene with different lighting conditions.



Figure 3. The full set of objects in our dataset, contrast-adjusted to improve visibility.

In this section, we will use I , S and R to denote the image and Lambertian shading and reflectance images, respectively. $R(x, y, c)$ denotes the reflectance value at pixel (x, y) and color channel c , and $R(x, y)$ denotes the grayscale reflectance value $1/3 \sum_c R(x, y, c)$. All images will be grayscale unless otherwise noted. Lowercase r , s , and i indicate log images, e.g., $r = \log R$. The subscripts x and y , e.g., i_x or i_y , indicate horizontal and vertical image gradients.

4.1. Inputs and Outputs

Most intrinsic image algorithms assume a Lambertian shading model. In order that the ground truth be well-defined, we provide the algorithms with the diffuse image as input. We make this simplification only for evaluation purposes; in practice the algorithms still behave reasonably when given images containing specularities.

Because our images contain only a single direct light source, the shading at each pixel can be represented as a scalar $S(x, y)$. Therefore, the grayscale image decomposes as a product of shading and reflectance:

$$I(x, y) = S(x, y)R(x, y). \quad (6)$$

The algorithms are all evaluated on their ability to recover the shading/reflectance decomposition of the grayscale image, although some algorithms will use additional information to do so.

4.2. Error metric

Quantitatively comparing different algorithms requires choosing a meaningful error metric. Some authors have used mean squared error (MSE), but we found that this metric is too strict for most algorithms on our data. Incorrectly classifying a single edge can often ruin the relative shading and reflectance for different parts of the image, and this often dominates the error scores. To address this problem, we define a more lenient error criterion called local mean squared error (LMSE).

Since the ground truth is only defined up to a scale factor, we define the scale-invariant MSE for a true vector x and the

estimate \hat{x} :

$$\text{MSE}(x, \hat{x}) = \|x - \hat{x}\|^2, \quad (7)$$

with $\hat{\alpha} = \arg \min_{\alpha} \|x - \alpha \hat{x}\|^2$. We would like each local region of our estimated intrinsic images to look like the ground truth. Therefore, given the true and estimated shading images S and \hat{S} , we define local MSE (LMSE) as the MSE summed over all local windows w of size $k \times k$ and spaced in steps of $k/2$:

$$\text{LMSE}_k(S, \hat{S}) = \sum_{w \in W} \text{MSE}(S_w, \hat{S}_w). \quad (8)$$

Our score for the image is the average of the LMSE scores of the shading and reflectance images, normalized so that an estimate of all zeros has the maximum possible score of 1:

$$\frac{1}{2} \frac{\text{LMSE}_k(S, \hat{S})}{\text{LMSE}_k(S, 0)} + \frac{1}{2} \frac{\text{LMSE}_k(R, \hat{R})}{\text{LMSE}_k(R, 0)}. \quad (9)$$

Larger values of the window size k emphasize coarse-grained information while smaller values emphasize fine-grained information. All the results in this paper use $k = 20$, but we have found that the results are about the same even for much larger or smaller windows. We have found that this error criterion closely corresponds to our own judgments of the quality of the decomposition.

4.3. Algorithms

The evaluated algorithms assign interpretations to image gradients. To compare the algorithms, we used a uniform procedure for estimating the intrinsic images based on the algorithms' decisions:

1. Compute the horizontal and vertical gradients, i_x and i_y , of the log (grayscale) input image.
2. Interpret these gradients by estimating the gradients \hat{r}_x and \hat{r}_y of the log reflectance image.
3. Compute the log reflectance image \hat{r} which matches these gradients as closely as possible:

$$\hat{r} = \arg \min_r \sum_{x,y} |r_x(x, y) - \hat{r}_x(x, y)|^p + |r_y(x, y) - \hat{r}_y(x, y)|^p. \quad (10)$$

Except where otherwise stated, we use a least-squares penalty $p = 2$ for reconstructing the image. This requires solving a Poisson equation on a 2-D grid. Since we ignore the background of the image, \hat{r}_x and \hat{r}_y are only estimated for the pixels on the object itself. We use the PyAMG package to compute the optimum log reflectance image \hat{r} . We have also experimented with the robust loss function $p = 1$.

All implementations of the algorithms we test share steps 1 and 3; they differ in how they estimate the log reflectance derivatives in step 2. This way, we know that any differences are due to their ability to interpret the gradients, and not to their reconstruction algorithm. We now discuss particular algorithms.

4.3.1 Retinex

Printed materials tend to have sharp reflectance edges, while shadows and smooth surface curvature tend to produce soft shading edges. The Retinex algorithm (GR-RET), originally proposed by [8] and extended to two dimensions by [7, 3], takes advantage of this property by thresholding log-image gradients. In a grayscale image, large gradients are classified as reflectance, while small ones are classified as shading. For the horizontal gradient, this heuristic is defined as:

$$\hat{r}_x = \begin{cases} i_x & \text{if } |i_x| > T, \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

The same is done for the vertical gradient.

Extensions of Retinex which use color have met with much success in removing shadows from outdoor scenes. In particular, [5, 6] take advantage of the fact that most outdoor illumination changes lie in a two-dimensional subspace of log-RGB space. Their algorithms for identifying shadow edges are based on thresholding the projections of the log-image gradients into this subspace and the projections into the space perpendicular to it.

In our own dataset, because images contain only a single direct light source, all of the illumination changes lie in the span of the vector $(1, 1, 1)^T$, which is called the brightness subspace. The chromaticity space is the null space of that vector. Let i^{br} and i^{chr} be the projections of the log color image into these two subspaces, respectively. The color Retinex algorithm² uses two separate thresholds, one for brightness changes and one for color changes. More formally, The horizontal gradient of the log reflectance is estimated as

$$\hat{r}_x = \begin{cases} i_x^{br} & \text{if } \|i_x^{br}\| > T^{br} \text{ or } \|i_x^{chr}\| > T^{chr}, \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

²What we present is only loosely based on the original Retinex algorithm [8], which solved for each color channel independently.

where T^{br} and T^{chr} are independent thresholds that are chosen using cross-validation. The estimate for the vertical gradient has a similar form.

4.3.2 Learning approaches

Tappen et al. proposed a learning-based approach to estimating intrinsic images [15]. Using AdaBoost, they train a classifier to predict whether a given gradient is caused by shading or reflectance. The classifier outputs are pooled spatially using a Markov Random Field model which encourages gradients to have the same classification if they are likely to lie on the same contour. Inference is performed using loopy belief propagation and the likelihood scores are thresholded to produce the final output. We will abbreviate this algorithm (TAP-05).

More recently, the local regression approach of [14] supposes the shading gradients can be predicted as a locally linear function of local image patches. Their algorithm, called ExpertBoost, efficiently summarizes the training data using a small number of prototype patches, making it possible to efficiently approximate local regression. (Their full system also includes a method for weighting the local estimates by confidence, but we used the unweighted version in our experiments.) We abbreviate this algorithm (TAP-06). We trained both of these algorithms on our dataset using two-fold cross-validation with a random split.

4.3.3 Weiss’s multi-image algorithm

With multiple photographs of the same scene with different lighting conditions, it becomes easier to factor out the shading. In particular, cast shadows tend to move across the scene, so any particular location is unlikely to contain shadow edges in more than one or two images. Weiss [16] took advantage of this using a simple decision rule: assume the log-reflectance derivative is the median of the log-intensity derivatives over all of the images.

While Weiss’s algorithm is good for eliminating shadows, it leaves some shading residue in the reflectance image if there is a top-down bias in the light directions. However, note that shading due to surface normals tends to be smooth. This suggests the following heuristic: first run Weiss’s algorithm, and then run Retinex on the resulting reflectance image. This heuristic has been used by [9] as part of a video surveillance system. We abbreviate Weiss’s algorithm (W) and the combined algorithm (W+RET).

4.4. Results

4.4.1 Quantitative comparisons

To quantitatively compare the different algorithms, we computed two statistics. First, we averaged each algorithm’s LMSE scores, with all images weighted equally. Second,

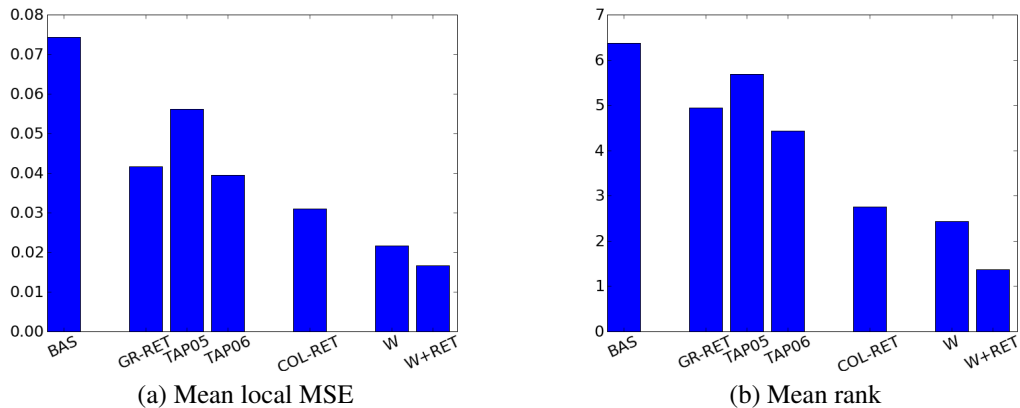


Figure 4. Quantitative comparison of all of the algorithms. (a) mean local MSE, as described in Section 4.2 (b) mean rank

we ranked all of the algorithms on each image and averaged the ranks. Both of these measures give approximately the same results. The results of all algorithms are shown in Figure 4.

The baseline (BAS) assumes each image is entirely shading or entirely reflectance, and this binary choice is made using cross-validation.

Surprisingly, Retinex performed better than TAP-05 and almost as well as TAP-06. This is surprising, because both algorithms include Retinex as a special case. We believe this is because, in a dataset with as much variety as ours, it is very difficult to separate shading and reflectance using only local grayscale information, so there is little advantage to using a better algorithm. Also, TAP-05 tends to overfit the training data. Finally, we note that, because Retinex has only a single threshold parameter, it can be more directly tuned to our evaluation metric using cross-validation.

As we would expect, color retinex (COL-RET) and Weiss’s algorithm (W), which have access to additional information, greatly outperform all of the grayscale algorithms. Also, combining Retinex with Weiss’s algorithm further improves performance by eliminating much of the shading residue from the reflectance image.

4.4.2 Examples

To understand the relative performance of the different algorithms, it is instructive to consider some particular examples. Our dataset can be roughly divided into three categories: artificially painted surfaces, printed objects, and toy animals. Figure 5 shows the algorithms’ outputs for an instance of each category.

The left-hand column shows a ceramic raccoon which we drew on with marker. This is a relatively easy image, because the retinex assumption is roughly satisfied, i.e. most sharp gradients are due to reflectance changes. Grayscale

retinex correctly identifies most of the markings as reflectance changes. However, it leaves some “ghost” markings in the shading image because pixel interpolation causes sharp edges to contain a mixture of large and small image gradients. TAP-05 eliminates many of these ghosts, because it uses information over a larger scale. Color retinex dramatically improves on the grayscale algorithms (as we would expect, because we used colored markers). However, all of these algorithms leave at least some residue of the cast shadows in the reflectance image. Weiss’s algorithm, which has access to multiple lighting conditions, completely eliminates the cast shadows. However, it leaves some residual shading in the reflectance image, because of the top-down bias in the light direction. Combining Weiss’s algorithm and Retinex eliminates most of this shading residue, giving a near-perfect decomposition.

The toy turtle is one of the most difficult objects in our dataset, because shading and reflectance changes are intermixed. All of the single-image algorithms do very poorly on this image because it is hard to estimate the decomposition using local information. Only Weiss’s algorithm, which can use multiple lighting conditions to disambiguate shading and reflectance, achieves a good reconstruction. This result is typical of the toy animal images, which are all very difficult for single-image algorithms.

4.4.3 Quadratic vs. L^1 reconstruction

As described in Section 4.3, all of the algorithms we consider share a common step, where they attempt to recover a reflectance image consistent with constraints on the gradients. If no image is exactly consistent, the decomposition will depend on the cost function applied to violated constraints. All of the results described above used the least squares penalty, but we also tested all of the algorithms using the robust L^1 cost function.

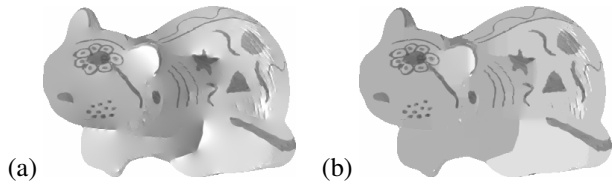


Figure 6. An example of the reflectance estimates produced by the color retinex algorithm using different cost functions for reconstruction. (a) L^2 (LMSE = 0.012) (b) L^1 (LMSE = 0.006). Note that using L^1 eliminates the halo effect, improving results both qualitatively and quantitatively.

The L^1 penalty slightly improves results on most images. One example where L^1 provided significant benefit is shown in Figure 6. The least-squares solution contains many “halos” around some of the edges, and these are eliminated using L^1 . In general, we have found that L^1 gives significant improvement over L^2 when the estimated constraints are far from satisfiable, when the image consists of scattered markings on an otherwise constant-reflectance surface, and when the least-squares solution is already close to the correct decomposition. Otherwise, L^1 gives only a subtle improvement. For instance, Weiss’s algorithm typically returns constraints which can be satisfied almost exactly, so least squares and L^1 return the same solution.

5. Conclusion

We have created a ground-truth dataset of intrinsic image decompositions of 16 real objects. These decompositions were obtained using polarization techniques and various paints, as well as a sphere-based system for precisely repositioning objects. We have made available our complete set of ground-truth data, along with the experimental code, at <http://people.csail.mit.edu/rgrosse/intrinsic> to encourage progress on the intrinsic images problem.

Using this dataset, we have evaluated several existing intrinsic image algorithms, assessing the relative merits of different single-image grayscale algorithms, as well as quantifying how much can be gained from additional sources of information. We found that, in a dataset as varied as ours, it is very hard to interpret image gradients using purely local grayscale cues. On the other hand, color is a very useful cue, as is a set of images of the same scene with different lighting conditions. We hope that the ability to train and evaluate on our dataset will spur further progress on intrinsic image algorithms.

Acknowledgements

We would like to thank Kevin Kleinguetl and Amity Johnson for helping with the data gathering and Marshall

Tappen for helpful discussions and for assisting with the algorithmic comparisons. This work was supported in part by the National Geospatial-Intelligence Agency, Shell, the National Science Foundation grant 0739255, NGA NEGI-1582-04-0004, MURI Grant N00014-06-1-0734, and gifts from Microsoft Research, Google, and Adobe Systems.

References

- [1] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. In A. Hanson and E. Riseman, editors, *Computer Vision Systems*, pages 3–26. Academic Press, 1978.
- [2] M. Bell and W. T. Freeman. Learning local evidence for shading and reflectance. In *Proc. of the Int. Conference on Computer Vision*, volume 1, pages 670–677, 2001.
- [3] A. Blake. Boundary conditions for lightness computation in Mondrian world. *Computer Vision, Graphics, and Image Processing*, pages 314–327, 32.
- [4] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative Model Based Vision*, 2004.
- [5] G. D. Finlayson, S. D. Hordley, and M. S. Drew. Removing shadows from images using retinex. In *Color Imaging Conference: Color Science and Engineering Systems, Technologies, and Applications*, pages 73–79, 2002.
- [6] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew. On the removal of shadows from images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(1):59–68, 2006.
- [7] B. K. P. Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.
- [8] E. H. Land and J. J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61(1):1–11, 1978.
- [9] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakauchi. Illumination normalization with time-dependent intrinsic images for video surveillance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(10):1336–1347, 2004.
- [10] S. K. Nayar, X.-S. Fand, and T. Boult. Separation of reflection components using color and polarization. *Int. Journal of Computer Vision*, 21(3):163–186, 1997.
- [11] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. of Computer Vision*, 47(1–3):7–42, 2002.
- [12] L. Shen, P. Tan, and S. Lin. Intrinsic image decomposition with non-local texture cues. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, 2008.
- [13] P. Sinha and E. Adelson. Recovering reflectance and illumination in a world of painted polyhedra. In *Proc. of the Fourth Int. Conf. on Computer Vision*, pages 156–163, 1993.
- [14] M. F. Tappen, E. H. Adelson, and W. T. Freeman. Estimating intrinsic component images using non-linear regression. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 1992–1999, 2006.
- [15] M. F. Tappen, W. T. Freeman, and E. H. Adelson. Recovering intrinsic images from a single image. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(9):1459–1472, 2005.







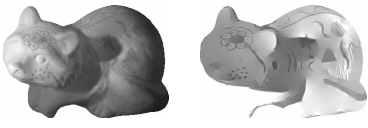

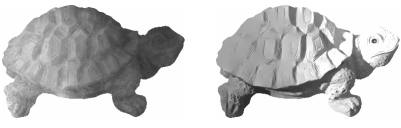
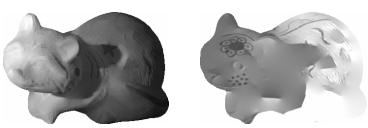
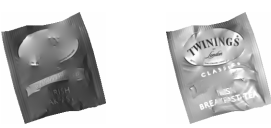
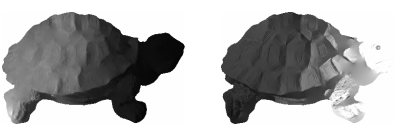


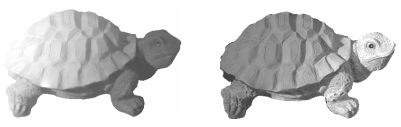


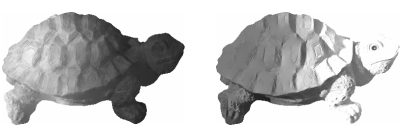
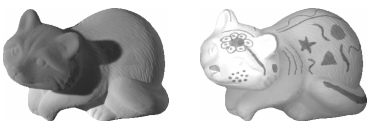

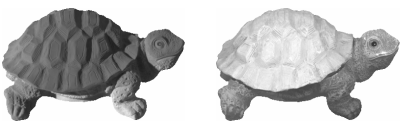



Image			
GT			
GR-RET	 LMSE = 0.037	 LMSE = 0.020	 LMSE = 0.088
TAP-05	 LMSE = 0.032	 LMSE = 0.109	 LMSE = 0.081
TAP-06	 LMSE = 0.036	 LMSE = 0.045	 LMSE = 0.059
COL-RET	 LMSE = 0.017	 LMSE = 0.024	 LMSE = 0.072
W	 LMSE = 0.017	 LMSE = 0.006	 LMSE = 0.026
W+RET	 LMSE = 0.005	 LMSE = 0.007	 LMSE = 0.023

Figure 5. Decompositions into shading (left) and reflectance (right) produced by all of the intrinsic image algorithms on three images from our dataset. See Section 4.4.2 for discussion.

- [16] Y. Weiss. Deriving intrinsic images from image sequences. In *Proc. of the Int. Conf. on Computer Vision*, volume 2, pages 68–75, 2001.