Rendering Synthetic Objects into Legacy Photographs

Kevin Karsch Varsha Hedau David Forsyth Derek Hoiem

University of Illinois at Urbana-Champaign
{karsch1,vhedau2,daf,dhoiem}@uiuc.edu

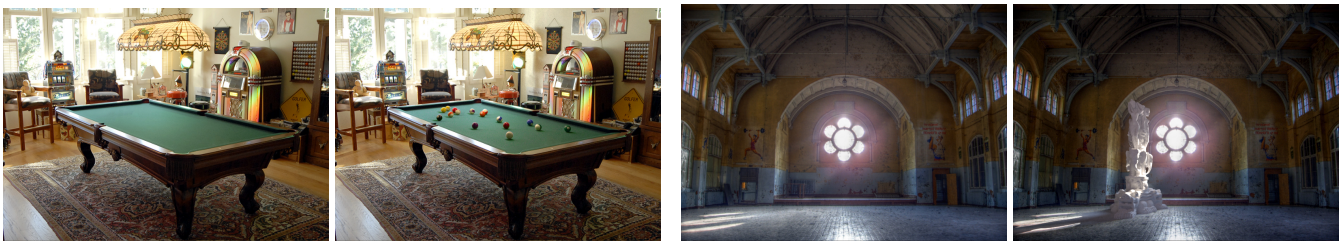


Figure 1: *With only a small amount of user interaction, our system allows objects to be inserted into legacy images so that perspective, occlusion, and lighting of inserted objects adhere to the physical properties of the scene. Our method works with only a single LDR photograph, and no access to the scene is required.*

Abstract

We propose a method to realistically insert synthetic objects into existing photographs without requiring access to the scene or any additional scene measurements. With a single image and a small amount of annotation, our method creates a physical model of the scene that is suitable for realistically rendering synthetic objects with diffuse, specular, and even glowing materials while accounting for lighting interactions between the objects and the scene. We demonstrate in a user study that synthetic images produced by our method are confusable with real scenes, even for people who believe they are good at telling the difference. Further, our study shows that our method is competitive with other insertion methods while requiring less scene information. We also collected new illumination and reflectance datasets; renderings produced by our system compare well to ground truth. Our system has applications in the movie and gaming industry, as well as home decorating and user content creation, among others.

CR Categories: I.2.10 [Computing Methodologies]: Artificial Intelligence—Vision and Scene Understanding; I.3.6 [Computing Methodologies]: Computer Graphics—Methodology and Techniques

Keywords: image-based rendering, computational photography, light estimation, photo editing

Links: DL PDF WEB

1 Introduction

Many applications require a user to insert 3D meshed characters, props, or other synthetic objects into images and videos. Currently, to insert objects into the scene, some scene geometry must be manually created, and lighting models may be produced by photographing mirrored light probes placed in the scene, taking multiple photographs of the scene, or even modeling the sources manually. Either way, the process is painstaking and requires expertise.

We propose a method to realistically insert synthetic objects into existing photographs without requiring access to the scene, special equipment, multiple photographs, time lapses, or any other aids. Our approach, outlined in Figure 2, is to take advantage of small amounts of annotation to recover a simplistic model of geometry and the position, shape, and intensity of light sources. First, we automatically estimate a rough geometric model of the scene, and ask the user to specify (through image space annotations) any additional geometry that synthetic objects should interact with. Next, the user annotates light sources and light shafts (strongly directed light) in the image. Our system automatically generates a physical model of the scene using these annotations. The models created by our method are suitable for realistically rendering synthetic objects with diffuse, specular, and even glowing materials while accounting for lighting interactions between the objects and the scene.

In addition to our overall system, our primary technical contribution is a semiautomatic algorithm for estimating a physical lighting model from a single image. Our method can generate a full lighting model that is demonstrated to be physically meaningful through a ground truth evaluation. We also introduce a novel image decomposition algorithm that uses geometry to improve lightness estimates, and we show in another evaluation to be state-of-the-art for single image reflectance estimation. We demonstrate with a user study that the results of our method are confusable with real scenes, even for people who believe they are good at telling the difference. Our study also shows that our method is competitive with other insertion methods while requiring less scene information. This method has become possible from advances in recent literature. In the past few years, we have learned a great deal about extracting high level information from indoor scenes [Hedau et al. 2009; Lee et al. 2009; Lee et al. 2010], and that detecting shadows in images is relatively straightforward [Guo et al. 2011]. Grosse *et al.* [2009] have also shown that simple lightness assumptions lead to powerful surface estimation algorithms; Retinex remains among the best methods.

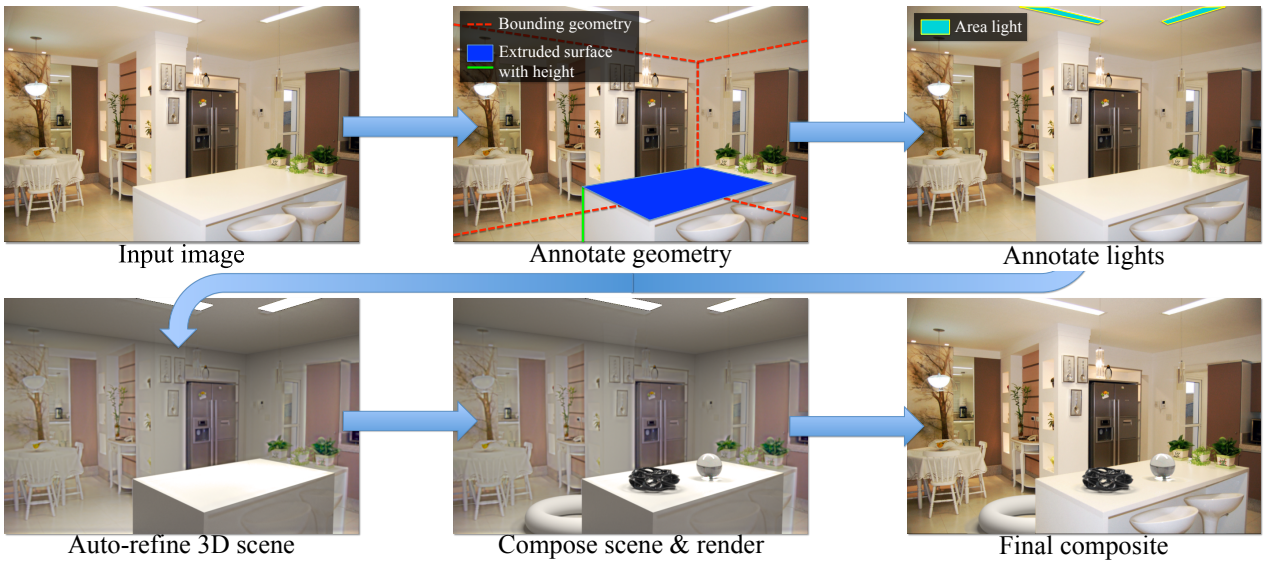


Figure 2: *Our method for inserting synthetic objects into legacy photographs. From an input image* (top left)*, initial geometry is estimated and a user annotates other necessary geometry* (top middle) *as well as light positions* (top right)*. From this input, our system automatically computes a 3D scene, including a physical light model, surface materials, and camera parameters* (bottom left)*. After a user places synthetic objects in the scene* (bottom middle)*, objects are rendered and composited into the original image* (bottom right)*. Objects appear naturally lit and adhere to the perspective and geometry of the physical scene. From our experience, the markup procedure takes only a minute or two, and the user can begin inserting objects and authoring scenes in a matter of minutes.*

2 Related work

Debevec's work [1998] is most closely related to ours. Debevec shows that a light probe, such as a spherical mirror, can be used to capture a physically accurate radiance map for the position where a synthetic object is to be inserted. This method requires a considerable amount of user input: HDR photographs of the probe, converting these photos into an environment map, and manual modeling of scene geometry and materials. More robust methods exist at the cost of more setup time (e.g. the plenopter [Mury et al. 2009]). Unlike these methods and others (e.g. [Fournier et al. 1993; Alnasser and Foroosh 2006; Cossairt et al. 2008; Lalonde et al. 2009]), we require no special equipment, measurements, or multiple photographs. Our method can be used with only a single LDR image, e.g. from Flickr, or even historical photos that cannot be recaptured.

Image-based Content Creation. Like us, Lalonde *et al.* [2007] aim to allow a non-expert user populate an image with objects. Objects are segmented from a large database of images, which they automatically sort to present the user with source images that have similar lighting and geometry. Insertion is simplified by automatic blending and shadow transfer, and the object region is resized as the user moves the cursor across the ground. This method is only suitable if an appropriate exemplar image exists, and even in that case, the object cannot participate in the scene's illumination. Similar methods exist for translucent and refractive objects [Yeung et al. 2011], but in either case, inserted objects cannot reflect light onto other objects or cast caustics. Furthermore, these methods do not allow for mesh insertion, because scene illumination is not calculated. We avoid these problems by using synthetic objects (3D textured meshes, now plentiful and mostly free on sites like Google 3D Warehouse and turbosquid.com) and physical lighting models.

Single-view 3D Modeling. Several user-guided [Liebowitz et al. 1999; Criminisi et al. 2000; Zhang et al. 2001; Horry et al. 1997; Kang et al. 2001; Oh et al. 2001; Sinha et al. 2008] or automatic [Hoiem et al. 2005; Saxena et al. 2008] methods are able to perform 3D modeling from a single image. These works are generally interested in constructing 3D geometric models for novel view synthesis. Instead, we use the geometry to help infer illumination and to handle perspective and occlusion effects. Thus, we can use simple box-like models of the scene [Hedau et al. 2009] with planar billboard models [Kang et al. 2001] of occluding objects. The geometry of background objects can be safely ignored. Our ability to appropriately resize 3D objects and place them on supporting surfaces, such as table-tops, is based on the single-view metrology work of Criminisi [2000]; also described by Hartley and Zisserman [2003]. We recover focal length and automatically estimate three orthogonal vanishing points, using the method from Hedau *et al.* [2009], which is based on Rother's technique [2002].

Materials and Illumination. We use an automatic decomposition of the image into albedo, direct illumination and indirect illumination terms (*intrinsic images* [Barrow and Tenenbaum 1978]). Our geometric estimates are used to improve these terms and material estimates, similar to Boivin and Gagalowicz [2001] and Debevec [1998], but our method improves efficiency of our illumination inference algorithm and is sufficient for realistic insertion (as demonstrated in Sections 5 and 6). We must work with a single legacy image, and wish to capture a physical light source estimate so that our method can be used in conjunction with any physical rendering software. Such representations as an irradiance volume do not apply [Greger et al. 1998]. Yu *et al.* show that when a comprehensive model of geometry and luminaires is available, scenes can be relit convincingly [Yu et al. 1999]. We differ from them in that our estimate of geometry is coarse, and do not require multiple images. Illumination in a room is not strongly directed, and cannot be encoded with a small set of point light sources, so the methods of Wang and Samaras [Wang and Samaras 2003] and Lopez-Moreno *et al.* [Lopez-Moreno et al. 2010] do not apply. As we show in our user study, point light models fail to achieve the realism that physical models do. We also cannot rely on having a known object present [Sato et al. 2003]. In the past, we have seen that people are unable to detect perceptual errors in lighting [Lopez-Moreno et al.

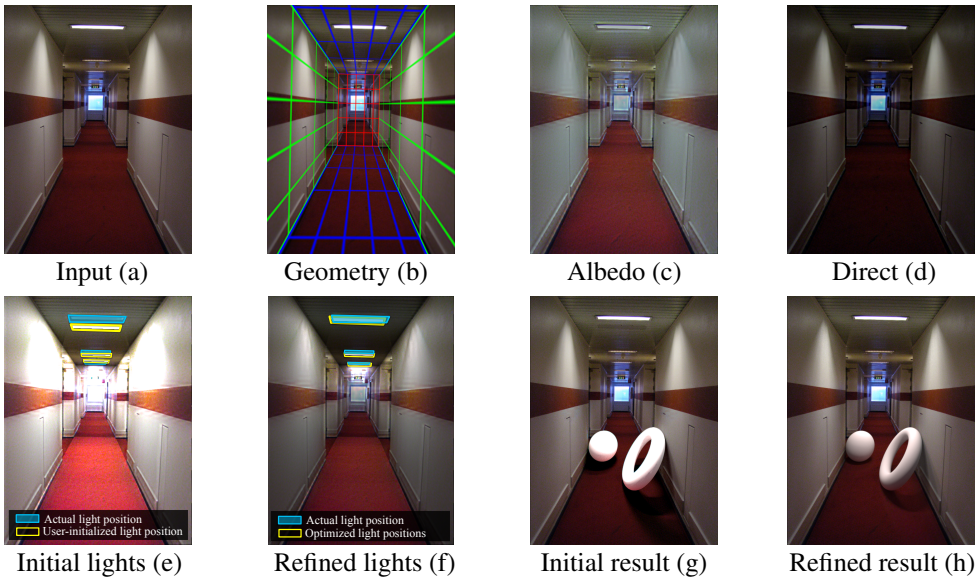


Figure 3: *Overview of our interior lighting algorithm. For an input image* (a)*, we use the modeled geometry (visualization of 3D scene boundaries as colored wireframe mesh,* (b)*) to decompose the image into albedo* (c) *and direct reflected light* (d)*. The user defines initial lighting primitives in the scene* (e)*, and the light parameters are re-estimated* (f)*. The effectiveness of our lighting algorithm is demonstrated by comparing a composited result* (g) *using the initial light parameters to another composited result* (h) *using the optimized light parameters. Our automatic lighting refinement enhances the realism of inserted objects. Lights are initialized away from the actual sources to demonstrate the effectiveness of our refinement.*

2010]. Such observations allow for high level image editing using rough estimates (e.g. materials [Khan et al. 2006] and lighting [Kee and Farid 2010]). Lalonde and Efros [2007] consider the color distribution of images to differentiate real and fake images; our user study provides human assessment on this problem as well.

There are standard computational cues for estimating intrinsic images. Albedo tends to display sharp, localized changes (which result in large image gradients), while shading tends to change slowly. These rules-of-thumb inform the Retinex method [Land and McCann 1971] and important variants [Horn 1974; Blake 1985; Brelstaff and Blake 1987]. Sharp changes of shading do occur at shadow boundaries or normal discontinuities, but cues such as chromaticity [Funt et al. 1992] or differently lit images [Weiss 2001] can control these difficulties, as can methods that classify edges into albedo or shading [Tappen et al. 2005; Farenzena and Fusiello 2007]. Tappen et al. [2006] assemble example patches of intrinsic image, guided by the real image, and exploiting the constraint that patches join up. Recent work by Grosse *et al.* demonstrates that the color variant of Retinex is state-of-the-art for single-image decomposition methods [Grosse et al. 2009].

3 Modeling

To render synthetic objects realistically into a scene, we need estimates of geometry and lighting. At present, there are no methods for obtaining such information accurately and automatically; we incorporate user guidance to synthesize sufficient models.

Our lighting estimation procedure is the primary technical contribution of our method. With a bit of user markup, we automatically decompose the image with a novel intrinsic image method, refine initial light sources based on this decomposition, and estimate light shafts using a shadow detection method. Our method can be broken into three phases. The first two phases interactively create models of geometry and lighting respectively, and the final phase renders and composites the synthetic objects into the image. An overview of our method is sketched in Algorithm 1.

3.1 Estimating geometry and materials

To realistically insert objects into a scene, we only need enough geometry to faithfully model lighting effects. We automatically obtain a coarse geometric representation of the scene using the technique of Hedau *et al.* [2009], and estimate vanishing points to recover camera pose automatically. Our interface allows a user to correct errors in these estimates, and also create simple geometry (tables and or near-flat surfaces) through image-space annotations. If necessary, other geometry can be added manually, such as complex objects near inserted synthetic objects. However, we have found that in most cases our simple models suffice in creating realistic results; all results in this paper require no additional complex geometry. Refer to Section 4.1 for implementation details.

3.2 Estimating illumination

Estimating physical light sources automatically from a single image is an extremely difficult task. Instead, we describe a method to obtain a physical lighting model that, when rendered, closely resembles the original image. We wish to reproduce two different types of lighting: *interior lighting*, emitters present within the scene, and *exterior lighting*, shafts of strongly directed light which lie outside of the immediate scene (e.g. sunlight).

Interior lighting. Our geometry is generally rough and not canonical, and our lighting model should account for this; lights should be modeled such that renderings of the scene look similar to the original image. This step should be transparent to the user. We ask the user to mark intuitively where light sources should be placed, and then refine the sources so that the rendered image best matches the original image. Also, intensity estimation and color cast can be difficult to estimate, and we correct these automatically (see Fig 3).

Initializing light sources. To begin, the user clicks polygons in the image corresponding to each source. These polygons are projected onto the geometry to define an area light source. Out-of-view sources are specified with 3D modeling tools.

Improving light parameters. Our technique is to choose light parameters to minimize the squared pixel-wise differences between the rendered image (with estimated lighting and geometry) and the target image (e.g. the original image). Denoting $R(\mathbf{L})$ as the rendered image parameterized by the current lighting parameter vector $\mathbf{L}$, R^* as the target image, and $\mathbf{L}_0$ as the initial lighting parameters,

LEGACYINSERTION(img, USER)
Model geometry (Sec 4.1), auto-estimate materials (Sec 4.2)
$geometry \leftarrow$ DETECTBOUNDARIES(img)
$geometry \leftarrow$ USER('Correct boundaries')
$geometry \leftarrow$ USER('Annotate/add additional geometry')
$geometry_{mat} \leftarrow$ ESTMATERIALS($img, geometry$) *[Eq 3]*
Refine initial lights and estimate shafts (Sec 3.2)
$lights \leftarrow$ USER('Annotate lights/shaft bounding boxes')
$lights \leftarrow$ REFINELIGHTS($img, geometry$) *[Eq 1]*
$lights \leftarrow$ DETECTSHAFTS(img)
Insert objects, render and composite (Sec 3.3)
$scene \leftarrow$ CREATESCENE($geometry, lights$)
$scene \leftarrow$ USER('Add synthetic objects')
return COMPOSITE(img, RENDER($scene$)) *[Eq 4]*

Algorithm 1: *Our method for rendering objects into legacy images*

we seek to minimize the objective

$$\underset{\mathbf{L}}{\operatorname{argmin}} \sum_{i \in \text{pixels}} \alpha_i (R_i(\mathbf{L}) - R_i^*)^2 + \sum_{j \in \text{params}} w_j (\mathbf{L}_j - \mathbf{L}_{0_j})^2 \quad \text{subject to: } 0 \leq \mathbf{L}_j \leq 1 \; \forall j \tag{1}$$

where w is a weight vector that constrains lighting parameters near their initial values, and α is a per-pixel weighting that places less emphasis on pixels near the ground. Our geometry estimates will generally be worse near the bottom of the scene since we may not have geometry for objects near the floor. In practice, we set $\alpha = 1$ for all pixels above the spatial midpoint of the scene (height-wise), and α decreases quadratically from 1 to 0 at floor pixels. Also, in our implementation, $\mathbf{L}$ contains 6 scalars per light source: RGB intensity, and 3D position. More parameters could also be optimized. For all results, we normalize each light parameter to the range $[0, 1]$, and set the corresponding values of w to 10 for spatial parameters and 1 for intensity parameters. A user can also modify these weights depending on the confidence of their manual source estimates. To render the synthetic scene and determine R, we must first estimate materials for all geometry in the scene. We use our own intrinsic image decomposition algorithm to estimate surface reflectance (albedo), and the albedo is then projected onto the scene geometry as a diffuse texture map, as described in Section 4.2.

Intrinsic decomposition. Our decomposition method exploits our geometry estimates. First, indirect irradiance is computed by *gathering* radiance values at each 3D patch of geometry that a pixel projects onto. The gathered radiance values are obtained by sampling observed pixel values from the original image, which are projected onto geometry along the camera's viewpoint. We denote this indirect irradiance image as Γ; this term is equivalent to the integral in the radiosity equation. Given the typical Lambertian assumptions, we assume that the original image B can be expressed as the product of albedo ρ and shading S as well as the sum of reflected direct light D and reflected indirect light I. Furthermore, reflected gathered irradiance is equivalent to reflected indirect lighting under these assumptions. This leads to the equations

$$B = \rho S, \quad B = D + I, \quad I = \rho\Gamma, \quad B = D + \rho\Gamma. \tag{2}$$

We use the last equation as constraints in our optimization below.

We have developed an objective function to decompose an image B into albedo ρ and direct light D by solving

$$\underset{\rho, D}{\operatorname{argmin}} \sum_{i \in \text{pixels}} |\Delta\rho|_i + \gamma_1 m_i (\nabla\rho)_i^2 + \gamma_2 (D_i - D_{0_i})^2 + \gamma_3 (\nabla D)_i^2 \quad \text{subject to} \quad B = D + \rho\Gamma, \quad 0 \leq \rho \leq 1, \quad 0 \leq D, \tag{3}$$

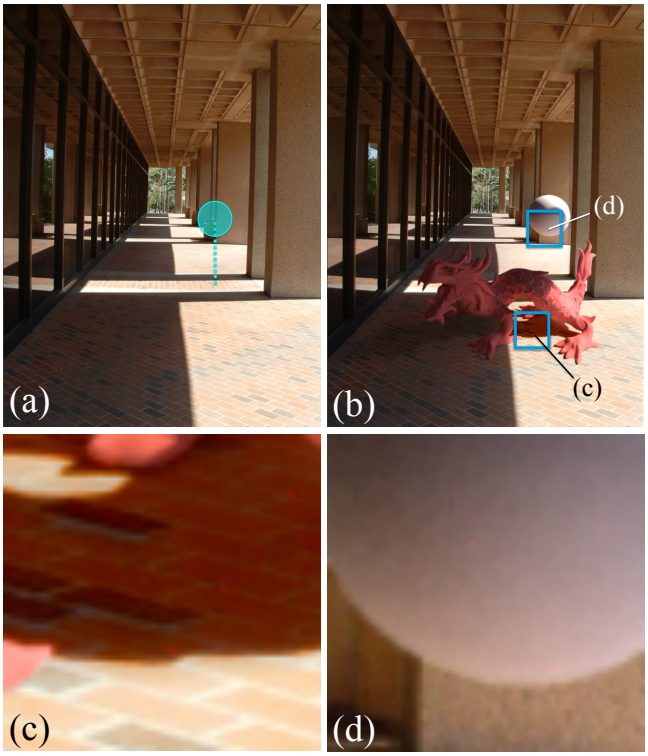


Figure 4: *Inserted objects fully participate with the scene lighting as if they were naturally a part of the image. Here, an input image* (a) *is augmented with inserted objects and illuminated with a bright shaft of light* (b)*. Interreflected red light from the dragon onto the brick floor is evident in* (c)*, and the underside of the inserted sphere has a slight red tint from light reflecting off of the brick* (d)*. A registration probe in* (a) *displays the scale and location of the sphere in* (b)*. Best viewed on a high resolution, high contrast display.*

where $\gamma_1, \gamma_2, \gamma_3$ are weights, m is a scalar mask taking large values where B has small gradients, and small values otherwise, and D_0 is the initial direct lighting estimate. We define m as a sigmoid applied to the gradient magnitude of B: $m_i = 1 - 1/(1 + e^{-s(||\nabla B||_i^2 - c)})$, setting $s = 10.0, c = 0.15$ in our implementation.

Our objective function is grounded in widespread intrinsic image assumptions [Land and McCann 1971; Blake 1985; Brelstaff and Blake 1987], namely that shading is spatially slow and albedo consists of piecewise constant patches with potentially sharp boundaries. The first two terms in the objective coerce ρ to be piecewise constant. The first term enforces an L1 sparsity penalty on edges in ρ, and the second term smoothes albedo only where B's gradients are small. The final two terms smooth D while ensuring it stays near the initial estimate D_0. We set the objective weights to $\gamma_1 = 0.2$, $\gamma_2 = 0.9$, and $\gamma_3 = 0.1$. We initialize ρ using the color variant of Retinex as described by Grosse *et al.* [2009], and initialize D as $D_0 = B - \rho\Gamma$ (by Eq. 2). This optimization problem can be solved in a variety of ways; we use an interior point method (implemented with MATLAB's optimization toolbox). In our implementation, to improve performance of our lighting optimization (Eq. 1), we set the target image as our estimate of the direct term, and render our scene only with direct lighting (which greatly reduces the time in recalculating the rendered image). We choose our method as it utilizes the estimated scene geometry to obtain better albedo estimates, and reduces the computation cost of solving Eq. 1, but any decomposition method could be used (e.g. Retinex).

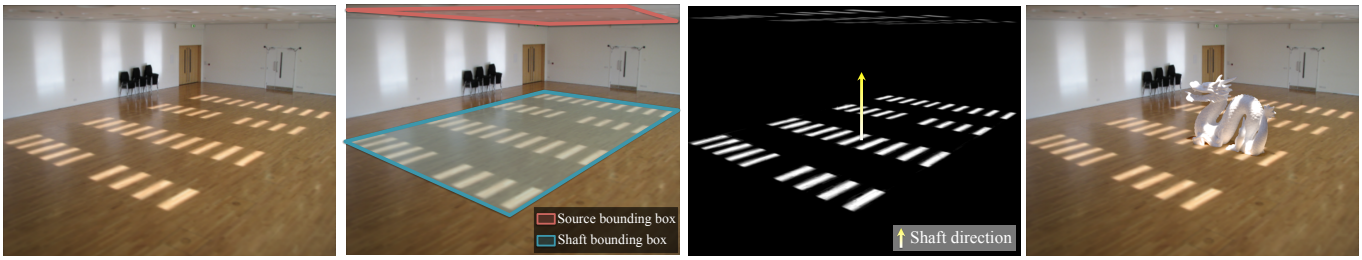


Figure 5: *Our algorithm for estimating exterior lighting (light shafts). Given an input image* (a)*, the user specifies bounding boxes around the shafts and their sources* (b)*. The shafts are detected automatically, and the shaft direction is estimated using the centroid of the the bounding boxes in 3D* (c)*. A physical lighting model (e.g. a masked, infinitely far spotlight) is created from this information, and objects can be rendered inserted realistically into the scene* (d)*.*

Figure 6: *A difficult image for detecting light shafts. Many pixels near the window are saturated, and some shaft patterns on the floor are occluded, as in the image on the left. However, an average of the matte produced for the floor and wall provides an acceptable estimate (used to relight the statue on the right).*

Figure 7: *Our system is intuitive and quick. This result was modeled by a user unfamiliar with our interface (after a short demonstration). From start to finish, this result was created in under 10 minutes (render time not included). User's markup shown on left.*

Exterior lighting (light shafts). Light shafts are usually produced by the sun, or some other extremely far away source. Thus, the type of light we wish to model can be thought of as purely directional, and each shaft in a scene will have the same direction.

We define a light shaft with a 2D polygonal projection of the shaft and a direction vector. In Figure 5, the left image shows a scene with many light shafts penetrating the ceiling and projecting onto the floor. Our idea is to detect either the source or the projections of shafts in an image and recover the shaft direction. The user first draws a bounding box encompassing shafts visible in the scene, as well as a bounding box containing shaft sources (windows, etc.). We then use the shadow detection algorithm of Guo *et al.* [2011] to determine a scalar mask that estimates the confidence that a pixel is *not* illuminated by a shaft. This method models region based appearance features along with pairwise relations between regions that have similar surface material and illumination. A graph cut inference is then performed to identify the regions that have same material and different illumination conditions, resulting in the confidence mask. The detected shadow mask is then used to recover a soft shadow matte using the spectral matting method of Levin *et al.* [2008]. We then use our estimate of scene geometry to recover the direction of the shafts (the direction defined by the two midpoints of the two bounding boxes). However, it may be the case that either the shaft source or the shaft projection is not visible in an image. In this case, we ask the user to provide an estimate of the direction, and automatically project the source/shaft accordingly. Figure 5 shows an example of our shaft procedure where the direction vector is calculated automatically from the marked bounding boxes. Shafts are represented as masked spotlights for rendering.

In some cases, it is difficult to recover accurate shadow mattes for a window on a wall or a shaft on the floor individually. For instance, it is difficult to detect the window in Figure 6 using only the cues from the wall. In such cases, we project the recovered mask on the floor along the shaft direction to get the mapping on the wall and average matting results for the wall and floor to improve the results. Similarly, an accurate matte of a window can be used to improve the matte of a shaft on the floor (as in the right image of Figure 1).

3.3 Inserting synthetic objects

With the lighting and geometry modeled, a user is now free to insert synthetic 3D geometry into the scene. Once objects have been inserted, the scene can be rendered with any suitable rendering software.[1] Rendering is trivial, as all of the information required by the renderer has been estimated (lights, geometry, materials, etc).

To complete the insertion process, we composite the rendered objects back into the original photograph using the additive differential rendering method [Debevec 1998]. This method renders two images: one containing synthetic objects $\mathcal{I}_{obj}$, and one without synthetic objects $\mathcal{I}_{noobj}$, as well as an object mask M (scalar image that is 0 everywhere where no object is present, and $(0, 1]$ otherwise). The final composite image $\mathcal{I}_{final}$ is obtained by

$$\mathcal{I}_{final} = M \odot \mathcal{I}_{obj} + (1 - M) \odot (\mathcal{I}_b + \mathcal{I}_{obj} - \mathcal{I}_{noobj}) \quad (4)$$

where $\mathcal{I}_b$ is the input image, and $\odot$ is the Hadamard product.

[1] For our results, we use LuxRender (http://www.luxrender.net)

Figure 8: *Our method allows for light source insertion and easy material reassignment. Here, a glowing ball is inserted above a synthetic glass sphere, casting a caustic on the table. The mirror has been marked as reflective, allowing synthetic objects to realistically interact with the scene.*

Figure 9: *The instrument used for collecting ground truth illumination data. The left image shows the apparatus (a white, diffuse ball resting on a plastic, height-adjustable pole). Using knowledge of the physical scene, we can align a rendered sphere over the probe for error measurements* (right)*.*

Figure 10: *The chart used in our ground truth reflectance experiments (Sec. 5.2).*

4 Implementation details

4.1 Modeling geometry

Rough scene boundaries (*bounding geometry*) are estimated first along with the camera pose, and we provide tools for correcting and supplementing these estimates. Our method also assigns materials to this geometry automatically based on our intrinsic decomposition algorithm (Sec. 3.2).

Bounding geometry. We model the bounding geometry as a 3D cuboid; essentially the scene is modeled as a box that circumscribes the camera so that up to five faces are visible. Using the technique of Hedau *et al.* [2009], we automatically generate an estimate of this box layout for an input image, including camera pose. This method estimates three vanishing points for the scene (which parameterize the box's rotation), as well as a 3D translation to align the box faces with planar faces of the scene (walls, ceiling floor). However, the geometric estimate may be inaccurate, and in that case, we ask the user to manually correct the layout using a simple interface we have developed. The user drags the incorrect vertices of the box to corresponding scene corners, and manipulates vanishing points using a pair of line segments (as in the Google Sketchup[2] interface) to fully specify the 3D box geometry.

Additional geometry. We allow the user to easily model *extruded geometry*, i.e. geometry defined by a closed 2D curve that is extruded along some 3D vector, such as tables, stairs, and other axis-aligned surfaces. In our interface, a user sketches a 2D curve defining the surface boundary, then clicks a point in the footprint of the object which specifies the 3D height of the object [Criminisi et al. 2000]. Previously specified vanishing points and bounding geometry allow for these annotations to be automatically converted to a 3D model.

In our interface, users can also specify *occluding surfaces*, complex surfaces which will occlude inserted synthetic objects (if the inserted object is behind the occluding surface). We allow the user to create occlusion boundaries for objects using the interactive spectral matting segmentation approach [Levin et al. 2008]. The user defines the interior and exterior of an object by scribbling, and a segmentation matte for the object is computed. These segmentations act as cardboard cutouts in the scene; if an inserted object intersects the segmentation and it is farther from the camera, then it will be occluded by the cutout. We obtain the depth of an object by assuming the lowermost point on its boundary to be its contact point with the floor. Figures 2 and 7 show examples of both extruded and occluding geometry.

[2] http://sketchup.google.com

4.2 Modeling materials

We assign a material to all estimated geometry based on the albedo estimated during intrinsic image decomposition (Sec 3.2). We project the estimated albedo along the camera's view vector onto the estimated geometry, and render the objects with a diffuse texture corresponding to projected albedo. This projection applies also to out-of-view geometry (such as the wall behind the camera, or any other hidden geometry). Although unrealistic, this scheme has proven effective for rendering non-diffuse objects (it is generally difficult to tell that out-of-view materials are incorrect; see Fig 20).

5 Ground truth evaluations

Here, we evaluate the physical accuracy of lighting estimates produced by our method as well as our intrinsic decomposition algorithm. We do not strive for physical accuracy (rather, human believability), but we feel that these studies may shed light on how physical accuracy corresponds to people's perception of a real (or synthetic) image. Our studies show that our lighting models are quite accurate, but as we show later in our user study, people are not very good at detecting physical inaccuracies in lighting. Our reflectance estimates are also shown to be more accurate than the color variant of Retinex, which is currently one of the best single-image diffuse reflectance estimators.

5.1 Lighting evaluation

We have collected a ground truth dataset in which the surface BRDF is known for an object (a white, diffuse ball) in each image. Using our algorithm, we estimate the lighting for each scene and insert a synthetic sphere. Because we know the rough geometry of the scene, we can place the synthetic sphere at the same spatial location as the sphere in the ground truth image.

Dataset. Our dataset contains 200 images from 20 indoor scenes illuminated under varying lighting conditions. We use an inflatable ball painted with flat white paint as the object with known BRDF, which was matched and verified using a Macbeth Color Checker. The ball is suspended by a pole that protrudes from the ground and can be positioned at varying heights (see Fig 9). The images were taken with a Casio EXILIM EX-FH100 using a linear camera response function ($\gamma = 1$).

Results. For a pair of corresponding ground truth and rendered images, we measure the error by computing the pixel-wise difference of all pixels that have known BRDF. We measure this error

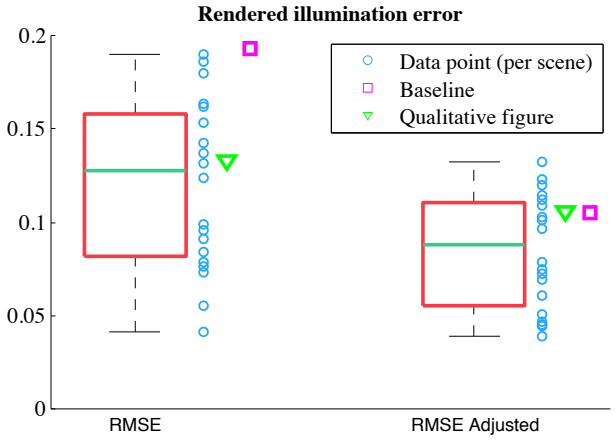


Figure 11: *We report results for both the root mean squared error (RMSE) and the RMSE after subtracting the mean intensity per sphere (RMSE adjusted). The RMSE metric illustrates how our method compares to the ground truth in an absolute metric, and the RMSE adjusted metric gives a sense of how accurate the lighting pattern is on each of the spheres (indicating whether light size/direction is correct). For each metric, we show a box plot where the green horizontal line is the median, and the red box extends to the 25th and 75th percentiles. The averaged RMSE per scene (10 spheres are in each scene) is shown as a blue circle. A baseline (purple square) was computed by rendering all spheres with uniform intensity, and set to be the mean intensity of all images in the dataset. The green triangle indicates the error for the qualitative illustration in Fig 12. No outliers exist for either metric, and image intensities range from [0,1].*

for each image in the dataset, and report the root mean squared error (RMSE). Overall, we found the RMSE to be 0.12 ± 0.049 for images with an intensity range of $[0, 1]$. For comparing lighting patterns on the spheres, we also computed the error after subtracting the mean intensity (per sphere) from each sphere. We found that this error to be 0.085 ± 0.03. Figure 11 shows the RMSE for the entire dataset, as well as the RMSE after subtracting the mean intensity (RMSE adjusted), and a baseline for each metric (comparing against a set of uniformly lit spheres with intensity set as the mean of all dataset images). Our method beats the baseline for every example in the RMSE metric, suggesting decent absolute intensity estimates, and about 70% of our renders beat the adjusted RMSE baseline. A qualitative visualization for five spheres in one scene from the dataset is also displayed in Figure 12. In general, baseline renders are not visually pleasing but still do not have tremendous error, suggesting qualitative comparisons may be more useful when evaluating lightness estimation schemes.

5.2 Intrinsic decomposition evaluation

We also collected a ground truth reflectance dataset to compare to the reflectance estimates obtained from our intrinsic decomposition algorithm. We place a chart with known diffuse reflectances (ranging from dark to bright) in each scene, and measure the error in reflectance obtained by our method as well as Retinex. We show that our method achieves more accurate absolute reflectance than Retinex in nearly every scene in the dataset.

Dataset. Our reflectance dataset contains 80 images from different indoor scenes containing our ground truth reflectance chart (shown in Fig 10). We created the chart using 15 Color-aid papers; 10 of which are monochrome patches varying between 3% reflectance

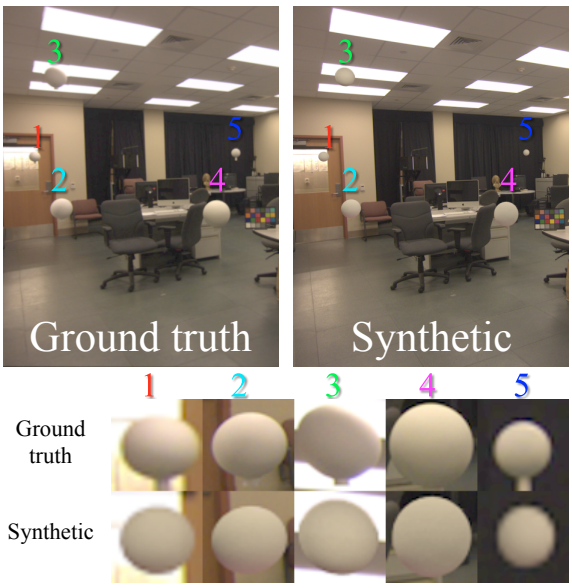


Figure 12: *Qualitative comparison of our lighting algorithm to ground truth lighting. The top left image shows a scene containing five real spheres with authentic lighting (poles edited out for visual compactness). We estimate illumination using our algorithm and render the spheres into the scene at the same spatial locations* (top right). *The bottom image matrix shows close-up views of the ground truth and rendered spheres. See Fig 11 for quantitative results.*

(very dark) and 89% reflectance (very bright). Reflectances were provided by the manufacturer. Each image in the dataset was captured by with the same camera and response as in Sec. 5.1.

Results. Using our decomposition method described in Sec. 3.2, we estimate the per-pixel reflectance of each scene in our dataset. We then compute the mean absolute error (MAE) and root mean squared error (RMSE) for each image over all pixels with known reflectance (i.e. only for the pixels inside monochromatic patches). For further comparison, we compute the same error measures using the color variant of Retinex (as described in Grosse *et al.* [2009]) as another method for estimating reflectance. Figure 13 summarizes these results. Our decomposition method outperforms Retinex for almost a large majority of the scenes in the dataset, and when averaged over the entire dataset, our method produced an MAE and RMSE of .141 and .207 respectively, compared to Retinex's MAE of .205 and RMSE of .272. These results indicate that much improvement can be made to absolute reflectance estimates when the user supplies a small amount of rough geometry, and that our method may improve other user-aided decomposition techniques, such as the method of Carroll *et al.* [2011].

5.3 Physical accuracy of intermediate results

From these studies, we conclude that our method achieves comparatively accurate illumination and reflection estimates. However, it is important to note that these estimates are heavily influenced by the rough estimates of scene geometry, and optimized to produce a perceptually plausible rendered image (with our method) rather than to achieve physical accuracy. Our method adjusts light positions so that the rendered scenes look most like the original image, and our reflectance estimates are guided by rough scene geometry. Thus, the physical accuracy of the light positions and reflectance bear little correlation on the fidelity of the final result.

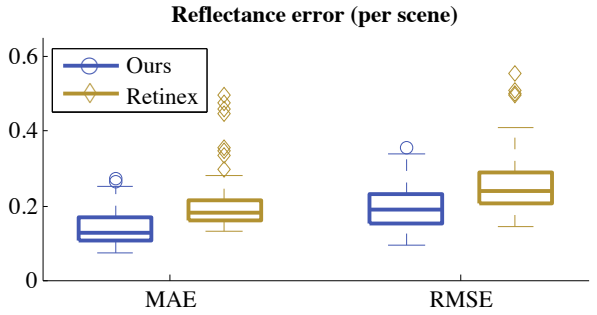


Figure 13: *Summary of the reflectance evaluation. Errors are measured per scene using a ground truth reflectance chart and reported in MAE and RMSE. For each method and metric, a box plot is shown where the center horizontal line indicates the median, and the box extends to the 25th and 75th percentiles. Results from our decomposition method are displayed in blue (outliers as circles); Retinex results are displayed in gold (outliers as diamonds).*

To verify this point, for each of the scenes in Sec 5.1, we plotted the physical accuracy of our illumination estimates versus the physical accuracy of both our light position and reflectance estimates (Fig 14). Light positions were marked by hand and a Macbeth ColorChecker was used for ground truth reflectance. We found that the overall Pearson correlation of illumination error and lighting position error was 0.034, and the correlation between illumination error and reflectance error was 0.074. These values and plots indicate a weak relation for both comparisons. Thus, our method is particularly good at achieving the final result, but this comes at the expense of physical inaccuracies along the way.

Figure 14: *The physical accuracy of our light position estimates as well as reflectance have little influence on the accuracy of illumination. This is likely because the light positions are optimized so that the rendered scene looks most like the original image, and the reflectance estimates are biased by our rough geometry estimates.*

6 User study

We also devised a user study to measure how well users can differentiate between real images and synthetic images of the same scene under varying conditions. For the study, we show each participant a sequence of images of the same background scene containing various objects. Some of these images are photographs containing no synthetic objects, and other images contain synthetic objects inserted with one of three methods: our method, a variant of Debevec's light probe method[3] [1998], or a baseline method (our method but with a simplified lighting model). Images are always shown in pairs and each of the paired images contain the exact same objects (although some of these objects may be synthetically inserted). The task presented to each user is a two-alternative forced choice test: for each pair of images, the user must choose (by mouse click) the image which they believe appears *most realistic*.

Methods. Our study tests three different methods for inserting synthetic objects into an image. For the first, we use our method as described in Section 3, which we will call **ours**. We also compare to a method that uses Debevec's light probe method for estimating the illumination, combined with our coarse geometry and reflectance estimates, referred to as **light probe**. To reproduce this method, we capture HDR photographs of a mirrored sphere in the scene from two angles (for removing artifacts/distortion), use these photographs to create a radiance map, model local geometry, and composite the rendered results [Debevec 1998]. Much more time was spent creating scenes with the light probe method than our own. The third method, denoted as **baseline**, also uses our geometry and reflectances but places a single point light source near the center of the ceiling rather than using our method for estimating light sources. *Note that each of these methods use identical reflectance and geometry estimates; the only change is in illumination.*

Variants. We also test four different variations when presenting users with the images to determine whether certain image cues are more or less helpful in completing this task. These variants are **monochrome** (an image pair is converted from RGB to luminance), **cropped** (shadows and regions of surface contact are cropped out of the image), **clutter** (real background objects are added to the scene), and **spotlight** (a strongly directed out of scene light is used rather than diffuse ceiling light). Note that the spotlight variant requires a new lighting estimate using our method, and a new radiance map to be constructed using the light probe method; also, this variant is not applicable to the baseline method. If no variant is applied, we label its variant as **none**.

Study details. There are 10 total scenes that are used in the study. Each scene contains the same background objects (walls, table, chairs, etc) and has the same camera pose, but the geometry within the scene changes. We use five real objects with varying geometric and material complexity (shown in Fig 15), and have recreated synthetic versions of these objects with 3D modeling software. The 10 different scenes correspond to unique combinations and placements of these objects. Each method was rendered using the same software (LuxRender), and the inserted synthetic geometry/materials remained constant for each scene and method. The rendered images were tone mapped with a linear kernel, but the exposure and gamma values differed per method. Tone mapping was performed so that the set of all scenes across a particular method looked most realistic (i.e. our preparation of images was biased towards realistic appearance for a skilled viewer, rather than physical estimates).

We recruited 30 subjects for this task. All subjects had a minimal graphics background, but a majority of the participants were computer scientists and/or graduate students. Each subject sees 24 pairs of images of identical scenes. 14 of these pairs contain one real and one synthetic image. Of these 14 synthetic images, five are created using our method, five are created using the light probe method, and the remaining four are created using the baseline method. Variants are applied to these pairs of images so that each user will see exactly one combination of each method and the applicable variants. The other 10 pairs of images shown to the subject are all synthetic; one image is created using our method, and the other using the light probe method. No variants are applied to these images.

Users are told that their times are recorded, but no time limit is

[3] We use Debevec's method for estimating illumination through the use of a light probe, coupled with our estimates of geometry and reflectance.

Figure 15: *Examples of methods and variants for Scene 10 in our user study. In the top row, from left to right, we show the real image, and synthetic images produced by our method, the light probe method, and the baseline method. In the bottom row, the four variants are shown.*

enforced. We ensure that all scenes, methods, and variants are presented in a randomly permuted order, and that the image placement (left or righthand side) is randomized. In addition to the primary task of choosing the most realistic image in the image pair, users are asked to rate their ability in performing this task both before and after the study using a scale of 1 (poor) to 5 (excellent).

Results. We analyze the results of the different methods versus the real pictures separately from the results of our method compared to the light probe method. When describing our results, we denote N as the sample size. When asked to choose which image appeared more realistic between our method and the light probe method, participants chose our image 67% of the time (202 of 300). Using a one-sample, one-tailed t-test, we found that users significantly preferred our method (p-value $\ll 0.001$), and on average users preferred our method more than the light probe method for all 10 scenes (see Fig 16).

In the synthetic versus real comparison, we found overall that people incorrectly believe the synthetic photograph produced with our method is real 34% of the time (51 of 150), 27% of the time with the light probe method (41 of 150), and 17% for the baseline (20 of 120). Using a two-sample, one-tailed t-test, we found that there was not a significant difference in subjects that chose our method over the light probe method ($p = 0.106$); however, there was a significant difference in subjects choosing our method over the baseline ($p = 0.001$), and in subjects choosing the light probe method over the baseline ($p = 0.012$). For real versus synthetic comparisons, we also tested the variants as described above. All variants (aside from "none") made subjects perform worse overall in choosing the real photo, but these changes were not statistically significant. Figure 17 summarizes these results.

We also surveyed four non-naïve users (graphics graduate students), whose results were not included in the above comparisons. Contrary to our assumption, their results *were* consistent with the other 30 naïve subjects. These four subjects selected 2, 3, 5, and 8 synthetic photographs (out of 14 real-synthetic pairs), an average of 35%, which is actually higher than the general population average of 27% (averaged over all methods/variants), indicating more trouble in selecting the real photo. In the comparison of our method to the light probe method, these users chose our method 5, 7, 7, and 8 times (out of 10 pairs) for an average of 68%, consistent with the naïve subject average of 67%.

Discussion. From our study, we conclude that both our method and the light probe method are highly realistic, but that users can tell a real image apart from a synthetic image with probability higher than chance. However, even though users had no time restrictions, they still could not differentiate real images from both our method and the light probe method reliably. As expected, both of these synthetic methods outperform the baseline, but the baseline still did surprisingly well. Applying different variants to the pairs of images hindered subjects' ability to determine the real photograph, but this difference was not statistically significant.

When choosing between our method and the light probe method, subjects chose our method with equal or greater probability than the light probe method for each scene in the study. This trend was probably the result of our light probe method implementation, which used rough geometry and reflectance estimates produced by our algorithm, and was not performed by a visual effects or image-based lighting expert. *Had such an expert generated the renderings for the light probe method, the results for this method might have improved, and so led to a change in user preference for comparisons involving the light probe method. The important conclusion is that we can now achieve realistic insertions without access to the scene.*

Surprisingly, subjects tended to do a worse job identifying the real picture as the study progressed. We think that this may have been caused by people using a particular cue to guide their selection initially, but during the study decide that this cue is unreliable or incorrect, when in fact their initial intuition was accurate. If this is the case, it further demonstrates how realistic the synthetic scenes look as well as the inability of humans to pinpoint realistic cues.

Many subjects commented that the task was more difficult than they thought it would be. Self assessment scores reflected these comments as self evaluations decreased for 25 of 30 subjects (i.e. a subject rated him/herself higher in the entry assessment than in the exit assessment), and in the other five subjects, the assessment remained the same. The average entry assessment was 3.9, compared to the average exit assessment of 2.8. No subject rated him/herself higher in the exit assessment than in the entry assessment.

The fact that naïve subjects scored comaparably to non-naïve subjects indicates that this test is difficult even for those familiar with computer graphics and synthetic renderings. All of these results indicate that people are not good at differentiating real from synthetic photographs, and that our method is state of the art.

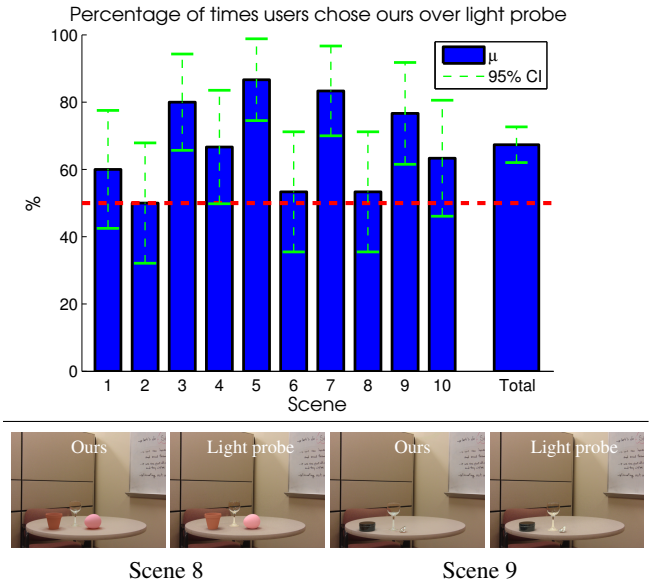
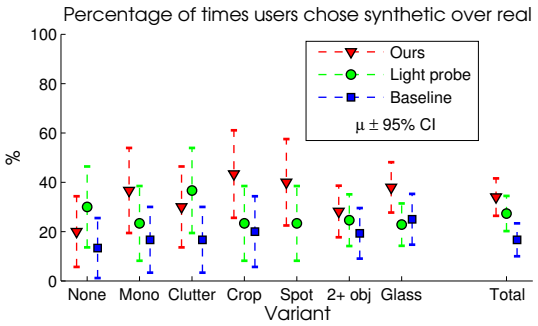


Figure 16: *When asked to pick which method appeared more realistic, subjects chose our method over the light probe method at least 50% of the time for each scene (67% on average), indicating a statistically significant number of users preferred our method. The blue bars represent the mean response (30 responses per bar, 300 total), and the green lines represent the 95% confidence interval. The horizontal red line indicates the 50% line. The images below the graph show two scenes from the study that in total contain all objects. Scene 8 was one of the lowest scoring scenes (53%), while scene 9 was one of the highest scoring (77%).*



Percentage of times users chose synthetic over real

$N = 30$	ours	light probe	baseline	total
none	20	30	13.3	21.1
monochrome	36.7	23.3	16.7	26.6
clutter	30	36.7	16.7	27.8
cropped	43.3	23.3	20	28.9
spotlight	40	23.3	N/A	31.7
total	34	27.3	16.7	26.7

	ours	light probe	baseline	total
2+ objects	28.2	24.6	19.3	24.4
glass	37.9	22.8	25	28.7

Figure 17: *Results for the three methods compared to a real image. In the graph, the mean response for each method is indicated by a triangle (ours), circle (light probe), and square (baseline). The vertical bars represent the 95% binomial confidence interval. The tables indicate the average population response for each category. We also considered the effects of inserting multiple synthetic objects and synthetic objects made of glass, and these results were consistent with other variants. Both our method and the light probe method performed similarly, indicated especially by the overlapping confidence intervals, and both methods clearly outperform the baseline. Variants do appear to have a slight affect on human perception (making it harder to differentiate real from synthetic).*

Figure 18: *Home redecorating is a natural application for our method. A user could take a picture of a room, and visualize new furniture or decorations without leaving home.*

7 Results and discussion

We show additional results produced with our system in Figs 18-21. Lighting effects are generally compelling (even for inserted emitters, Fig 8), and light interplay occurs automatically (Fig 4), although result quality is dependent on inserted models/materials. We conclude from our study that when shown to people, results produced by our method are confused with real images quite often, and compare favorably with other state-of-the-art methods.

Our interface is intuitive and easy to learn. Users unfamiliar with our system or other photo editing programs can begin inserting objects within minutes. Figure 7 shows a result created by a novice user in under 10 minutes.

We have found that many scenes can be parameterized by our geometric representation. Even images without an apparent box structure (e.g. outdoor scenes) work well (see Figs 19 and 20).

Quantitative measures of error are reassuring; our method beats natural baselines (Fig 12). Our intrinsic decomposition method incorporates a small amount of interaction and achieves significant improvement over Retinex in a physical comparison (Fig 13), and the datasets we collected (Sec 5) should aid future research in lightness and material estimation. However, it is still unclear which metrics should be used to evaluate these results, and qualitative evaluation is the most important for applications such as ours.

7.1 Limitations and future work

For extreme camera viewpoints (closeups, etc), our system may fail due to a lack of scene information. In these cases, luminaires may not exist in the image, and may be difficult to estimate (manually or automatically). Also, camera pose and geometry estimation might be difficult, as there may not be enough information available to determine vanishing points and scene borders.

Intrinsic image extraction may fail, either because the problem is still very difficult for diffuse scenes or because surfaces are not diffuse. For example, specular surfaces modeled as purely diffuse may cause missed reflections. Other single material estimation schemes could be used [Boivin and Gagalowicz 2001; Debevec 1998], but for specular surfaces and complex BRDFs, these methods will also likely require manual edits. It would be interesting to more accurately estimate complex surface materials automatically. Robust interactive techniques might also be a suitable alternative (i.e. [Carroll et al. 2011]).

Figure 19: *Our algorithm can handle complex shadows* (top)*, as well as out-of-view light sources* (bottom)*.*

Figure 20: *Specular materials naturally reflect the scene* (top)*, and translucent objects reflect the background realistically* (bottom)*.*

Insertion of synthetic objects into legacy videos is an attractive extension to our work, and could be aided, for example, by using multiple frames to automatically infer geometry [Furukawa and Ponce 2010], surface properties [Yu et al. 1999], or even light positions. Tone mapping rendered images can involve significant user interaction, and methods to help automate this process the would prove useful for applications such as ours. Incorporating our technique within redecorating aids (e.g. [Merrell et al. 2011; Yu et al. 2011]) could also provide a more realistic sense of interaction and visualization (as demonstrated by Fig 18).

8 Conclusion

We have demonstrated a system that allows a user to insert objects into legacy images. Our method only needs a few quick annotations, allowing novice users to create professional quality results, and does not require access to the scene or any other tools used previously to achieve this task. The results achieved by our method appear realistic, and people tend to favor our synthetic renderings over other insertion methods.

Figure 21: *Complex occluding geometry can be specified quickly via segmentation* (top, couch)*, and glossy surfaces in the image reflect inserted objects* (bottom, reflections under objects).

References

ALNASSER, M., AND FOROOSH, H. 2006. Image-based rendering of synthetic diffuse objects in natural scenes. In *ICPR*, 787–790.

BARROW, H., AND TENENBAUM, J. 1978. Recovering intrinsic scene characteristics from images. In *Comp. Vision Sys.*, 3–26.

BLAKE, A. 1985. Boundary conditions for lightness computation in mondrian world. *Computer Vision, Graphics and Image Processing 32*, 314–327.

BOIVIN, S., AND GAGALOWICZ, A. 2001. Image-based rendering of diffuse, specular and glossy surfaces from a single image. In *Proc. ACM SIGGRAPH*, 107–116.

BRELSTAFF, G., AND BLAKE, A. 1987. Computing lightness. *Pattern Recognition Letters 5*, 2, 129–138.

CARROLL, R., RAMAMOORTHI, R., AND AGRAWALA, M. 2011. Illumination decomposition for material recoloring with consistent interreflections. *ACM Trans. Graph. 30* (August), 43:1–43:10.

COSSAIRT, O., NAYAR, S., AND RAMAMOORTHI, R. 2008. Light field transfer: global illumination between real and synthetic objects. *ACM Trans. Graph. 27* (August), 57:1–57:6.

CRIMINISI, A., REID, I., AND ZISSERMAN, A. 2000. Single view metrology. *Int. J. Comput. Vision 40* (November), 123–148.

DEBEVEC, P. 1998. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, 189–198.

FARENZENA, M., AND FUSIELLO, A. 2007. Recovering intrinsic images using an illumination invariant image. In *ICIP*, 485–488.

FOURNIER, A., GUNAWAN, A. S., AND ROMANZIN, C. 1993. Common illumination between real and computer generated scenes. In *Proceedings of Graphics Interface '93*, 254–262.

FUNT, B. V., DREW, M. S., AND BROCKINGTON, M. 1992. Recovering shading from color images. In *ECCV*, 124–132.

FURUKAWA, Y., AND PONCE, J. 2010. Accurate, dense, and robust multiview stereopsis. *IEEE PAMI 32* (August), 1362–1376.

GREGER, G., SHIRLEY, P., HUBBARD, P. M., AND GREENBERG, D. P. 1998. The irradiance volume. *IEEE Computer Graphics and Applications 18*, 32–43.

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

GUO, R., DAI, Q., AND HOIEM, D. 2011. Single-image shadow detection and removal using paired regions. In *CVPR*, 2033–2040.

HARTLEY, R., AND ZISSERMAN, A. 2003. *Multiple View Geometry in Computer Vision*, 2 ed. Cambridge University Press, New York, NY, USA.

HEDAU, V., HOIEM, D., AND FORSYTH, D. 2009. Recovering the spatial layout of cluttered rooms. In *ICCV*, 1849–1856.

HOIEM, D., EFROS, A. A., AND HEBERT, M. 2005. Automatic photo pop-up. *ACM Trans. Graph. 24* (July), 577–584.

HORN, B. K. P. 1974. Determining lightness from an image. *Computer Vision, Graphics and Image Processing 3*, 277–299.

HORRY, Y., ANJYO, K.-I., AND ARAI, K. 1997. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, 225–232.

KANG, H. W., PYO, S. H., ANJYO, K., AND SHIN, S. Y. 2001. Tour into the picture using a vanishing line and its extension to panoramic images. *Computer Graphics Forum 20*, 3, 132–141.

KEE, E., AND FARID, H. 2010. Exposing digital forgeries from 3-d lighting environments. In *WIFS*, 1–6.

KHAN, E. A., REINHARD, E., FLEMING, R. W., AND BÜLTHOFF, H. H. 2006. Image-based material editing. *ACM Trans. Graph. 25* (July), 654–663.

LALONDE, J.-F., AND EFROS, A. A. 2007. Using color compatibility for assessing image realism. In *ICCV*, 1–8.

LALONDE, J.-F., HOIEM, D., EFROS, A. A., ROTHER, C., WINN, J., AND CRIMINISI, A. 2007. Photo clip art. *ACM Trans. Graph. 26* (July).

LALONDE, J.-F., EFROS, A. A., AND NARASIMHAN, S. G. 2009. Webcam clip art: appearance and illuminant transfer from time-lapse sequences. *ACM Trans. Graph. 28* (December), 131:1–131:10.

LAND, E., AND MCCANN, J. 1971. Lightness and retinex theory. *J. Opt. Soc. Am. 61*, 1, 1–11.

LEE, D. C., HEBERT, M., AND KANADE, T. 2009. Geometric reasoning for single image structure recovery. In *CVPR*, 2136–2143.

LEE, D. C., GUPTA, A., HEBERT, M., AND KANADE, T. 2010. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. *Advances in Neural Information Processing Systems (NIPS) 24* (November), 1288–1296.

LEVIN, A., RAV-ACHA, A., AND LISCHINSKI, D. 2008. Spectral matting. *IEEE PAMI 30* (October), 1699–1712.

LIEBOWITZ, D., CRIMINISI, A., AND ZISSERMAN, A. 1999. Creating architectural models from images. In *Eurographics*, vol. 18, 39–50.

LOPEZ-MORENO, J., HADAP, S., REINHARD, E., AND GUTIERREZ, D. 2010. Compositing images through light source detection. *Computers & Graphics 34*, 6, 698–707.

MERRELL, P., SCHKUFZA, E., LI, Z., AGRAWALA, M., AND KOLTUN, V. 2011. Interactive furniture layout using interior design guidelines. *ACM Trans. Graph. 30* (August), 87:1–87:10.

MURY, A. A., PONT, S. C., AND KOENDERINK, J. J. 2009. Representing the light field in finite three-dimensional spaces from sparse discrete samples. *Applied Optics 48*, 3 (Jan), 450–457.

OH, B. M., CHEN, M., DORSEY, J., AND DURAND, F. 2001. Image-based modeling and photo editing. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, 433–442.

ROTHER, C. 2002. A new approach to vanishing point detection in architectural environments. *IVC 20*, 9-10 (August), 647–655.

SATO, I., SATO, Y., AND IKEUCHI, K. 2003. Illumination from shadows. *IEEE PAMI 25*, 3, 290–300.

SAXENA, A., SUN, M., AND NG, A. Y. 2008. Make3d: depth perception from a single still image. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 3*, AAAI Press, 1571–1576.

SINHA, S. N., STEEDLY, D., SZELISKI, R., AGRAWALA, M., AND POLLEFEYS, M. 2008. Interactive 3d architectural modeling from unordered photo collections. *ACM Trans. Graph. 27* (December), 159:1–159:10.

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

TAPPEN, M. F., ADELSON, E. H., AND FREEMAN, W. T. 2006. Estimating intrinsic component images using non-linear regression. In *CVPR*, vol. 2, 1992–1999.

WANG, Y., AND SAMARAS, D. 2003. Estimation of multiple directional light sources for synthesis of augmented reality images. *Graphical Models 65*, 4, 185–205.

WEISS, Y. 2001. Deriving intrinsic images from image sequences. In *ICCV*, II: 68–75.

YEUNG, S.-K., TANG, C.-K., BROWN, M. S., AND KANG, S. B. 2011. Matting and compositing of transparent and refractive objects. *ACM Trans. Graph. 30* (February), 2:1–2:13.

YU, Y., DEBEVEC, P., MALIK, J., AND HAWKINS, T. 1999. Inverse global illumination: recovering reflectance models of real scenes from photographs. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, 215–224.

YU, L.-F., YEUNG, S.-K., TANG, C.-K., TERZOPOULOS, D., CHAN, T. F., AND OSHER, S. J. 2011. Make it home: automatic optimization of furniture arrangement. *ACM Trans. Graph. 30* (August), 86:1–86:12.

ZHANG, L., DUGAS-PHOCION, G., SAMSON, J., AND SEITZ, S. 2001. Single view modeling of free-form scenes. In *CVPR*, 990–997.