Detecting 3D Geometric Boundaries of Indoor Scenes Under Varying Lighting

Jie Ni
Univ. of Maryland
College Park, MD
jni@umiacs.umd.edu

Tim K. Marks Oncel Tuzel
Mitsubishi Electric Research Labs*
Cambridge, MA
{tmarks,oncel}@merl.com

Fatih Porikli
Australian National Univ./ NICTA
Canberra, Australia
fatih.porikli@anu.edu.au

Abstract

The goal of this research is to identify 3D geometric boundaries in a set of 2D photographs of a static indoor scene under unknown, changing lighting conditions. A 3D geometric boundary is a contour located at a 3D depth discontinuity or a discontinuity in the surface normal. These boundaries can be used effectively for reasoning about the 3D layout of a scene. To distinguish 3D geometric boundaries from 2D texture edges, we analyze the illumination subspace of local appearance at each image location. In indoor time-lapse photography and surveillance video, we frequently see images that are lit by unknown combinations of uncalibrated light sources. We introduce an algorithm for semi-binary nonnegative matrix factorization (SBNMF) to decompose such images into a set of lighting basis images, each of which shows the scene lit by a single light source. These basis images provide a natural, succinct representation of the scene, enabling tasks such as scene editing (e.g., relighting) and shadow edge identification.

1. Introduction

Edge detection is a fundamental problem in computer vision, providing important low-level features for myriad applications. Edges in images can result from various causes, including surface texture, depth discontinuities, differences in surface orientation, changes in material properties, and illumination variations. Many existing algorithms model edges as changes in low-level image properties, such as brightness, color, and texture, within an individual image [19, 5, 15, 14]. The problem of extracting 3D geometric boundaries, which are discrete changes in surface depth or orientation, has received less attention. As 3D geometric boundaries do not vary with changes in illumination or texture, they are robust characteristics of indoor scenes that can provide useful cues for many tasks including segmentation [20], scene categorization [17, 8], 3D reconstruction [7], and scene layout recovery [9].

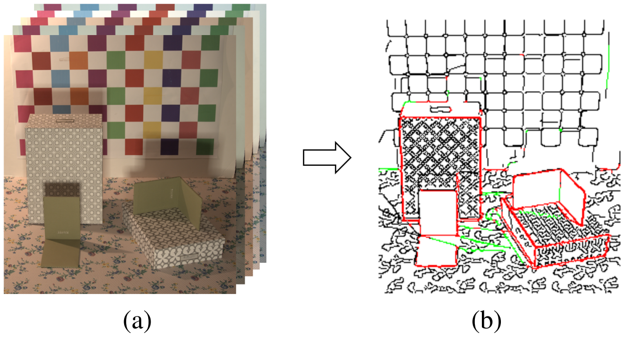


Figure 1. Problem overview: Given a set of images of a static indoor scene under varying lighting (a), we identify 3D geometric boundaries (shown in red) and shadow edges (shown in green), as opposed to texture edges (shown in black) (b).

We aim to detect 3D geometric boundaries from a set of images of an indoor static scene, captured from a fixed camera viewpoint, under uncontrolled lighting conditions due to unknown combinations of multiple unknown light sources. In time-lapse photography or video of indoor scenes, such as building surveillance or a living space imaged by a camera in a TV set, light sources in the environment are turned on/off independently, producing images with unknown combinations of light sources. In such situations, there is no control over which lighting combinations are observed, nor over the lights' locations or brightness. This breaks key assumptions of work on detecting depth edges using calibrated light sources. For example, structured light approaches such as [10] rely on strategically projected light patterns, and the multi-flash camera approach [18] requires a set of lights that encircle the camera and are individually activated in a controlled sequence. Furthermore, in our setup, the lights are not restricted to be point sources, and the distances from the lights (and camera) to the scene points are not infinite (these distances are comparable to the size of the scene). This contrasts with previous methods [21, 11, 22] that require distant lighting to recover 3D structure from 2D images under varying illumination.

We capitalize on the observation that on a non-specular

*The research described in this paper was all done at MERL.

(e.g., Lambertian) 3D surface, neighboring pixels have the same relative response to lighting even though they may have different albedos. The reason is that in a small neighborhood the 3D surface is locally planar (adjacent pixels come from surfaces with approximately the same normal), and the 3D distance between points in two neighboring pixels is much smaller than the distances from the surface to the light sources and camera. Based on this observation, we develop a method that can distinguish pixels on 3D geometric boundaries (pixels whose immediate neighborhoods include a discontinuity in surface normal or in depth) from pixels whose neighborhoods may contain sharp texture (intensity) boundaries but lie on a single surface.

We formulate 3D geometric boundary detection as a per-pixel classification problem by analyzing the rank of the *illumination subspace of local appearance* around each point. The illumination subspace of local appearance around a point is the subspace spanned by the appearance of the point's neighborhood across all lighting conditions.

Given a set of images, each of which is illuminated by multiple light sources, we learn *lighting basis images* using a type of nonnegative matrix factorization. A lighting basis image is the image that would be formed when the scene is illuminated by just one of the individual light sources that is present in the scene (note that this need not be a point light source). This is somewhat similar to the concept of *intrinsic images* [1, 23, 16, 21], which serve as mid-level scene descriptions. Unlike much of the previous work, our method assumes neither a single distant light source whose position varies smoothly over time, nor an orthographic camera.

As our generative imaging model satisfies the superposition property (the image resulting from a combination of lights is the sum of the images resulting from each of the lights independently), and we do not know which combinations of light sources are present in the input images, we introduce a type of Nonnegative Matrix Factorization (NMF) to solve for lighting basis images. In general NMF, a nonnegative data matrix is factored into a product of two nonnegative matrices [12, 13, 3]. To solve our problem, we introduce *semi-binary nonnegative matrix factorization* (SBNMF), in which a nonnegative data matrix is factored into a product of a nonnegative matrix and a binary matrix. We factor a matrix containing the input images into a nonnegative matrix of basis images and a binary weight matrix that indicates which light sources were on/off in each input image (see Figure 2). The recovered lighting basis images provide a compact scene representation under lighting variations. In addition to enabling scene editing tasks such as relighting, the basis images make it possible to distinguish true 3D geometry edges from shadow edges.

We apply our method for 3D geometric boundary detection to three datasets and compare with state-of-the-art contour detection methods. Precision-recall curves for each method on each dataset demonstrate the superior performance of our approach on this type of data.

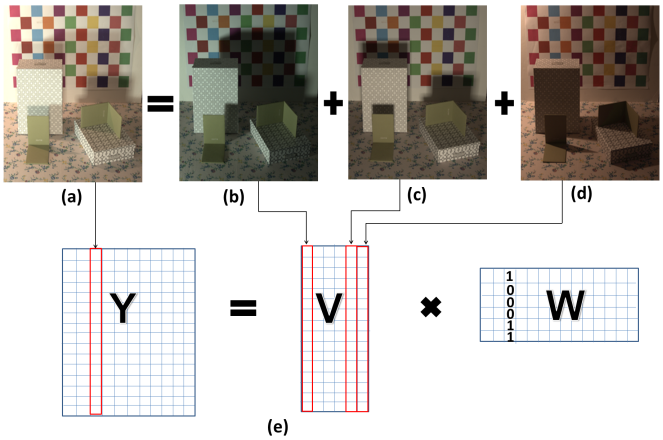


Figure 2. Overview of our factorization approach: An image (a) illuminated by a combination of several light sources can be represented as an additive combination of lighting basis images (b),(c),(d). Hence, (e) given a set of images $\mathbf{Y}$, we solve for basis images $\mathbf{V}$ via nonnegative matrix factorization, with binary constraints on the entries of the lighting weight matrix $\mathbf{W}$.

2. Indoor Scene Decomposition Under Variable Lighting

In this section, we describe our generative image model for indoor scenes and propose a new optimization algorithm for recovering the lighting bases.

2.1. Generative Image Model

Assume there are l light sources illuminating the indoor scene, with each light source controlled by an independent switch. We assign a binary variable w_i to indicate the status of each light source. Then we define a nonnegative lighting basis image $\mathbf{v}_i \in \mathbb{R}^{+n}$ as the image formed when only the ith light is on. Given an image $\mathbf{y}$ that is illuminated by any combination of the l light sources, it can be expressed as the superposition of individual basis images:

$$\mathbf{y} = \sum_{i=1}^{l} w_i \mathbf{v}_i, \quad w_i \in \{0, 1\}. \tag{1}$$

Note that throughout the paper, we write images as column vectors formed by stacking all the columns of the image.

We capture m images under various combinations of light sources and rearrange them into a data matrix $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_m] \in \mathbb{R}^{+n \times m}$. Following (1), this data matrix can be decomposed as:

$$\mathbf{Y} = \mathbf{V}\mathbf{W}, \tag{2}$$

where the columns of $\mathbf{V} \in \mathbb{R}^{+n \times l}$ correspond to the l basis images $\mathbf{v}_i$, and $\mathbf{W} \in \{0, 1\}^{l \times m}$ is an indicator matrix whose entries W_{ij} show the contribution of the ith light source to the jth input image (see Figure 2).

2.2. Recovering Basis Images via SBNMF

If the true lighting basis images are linearly independent, and we observe sufficient illumination variability (the rank of the true indicator matrix $\mathbf{W}$ is not less than the number of lights), then the number of lights in the scene, l, is given by the rank of the data matrix $\mathbf{Y}$. Note that if there are two or more lights that are always turned on and off together, they are considered a single light source.

We formulate recovery of the basis images and indicator matrix as a constrained optimization problem:

$$\min_{\mathbf{V},\mathbf{W}} \|\mathbf{Y} - \mathbf{V}\mathbf{W}\|_F^2, \;\; s.t. \;\; V_{ij} \geq 0, W_{jk} \in \{0,1\}, \forall i,j,k \tag{3}$$

which we call Semi-Binary Nonnegative Matrix Factorization (SBNMF). This is a challenging problem due to the non-convex objective function and the binary constraints on $\mathbf{W}$. Instead, we initially solve the continuous relaxation:

$$\min_{\mathbf{V},\mathbf{W}} \|\mathbf{Y} - \mathbf{V}\mathbf{W}\|_F^2, \;\; s.t. \;\; V_{ij} \geq 0, 0 \leq W_{jk} \leq 1, \forall i,j,k \tag{4}$$

where the binary constraints on W_{ij} are replaced by simple box constraints. This is a bi-convex problem which we solve using the Alternating Direction Method of Multipliers (ADMM) [4]. We rewrite (4) by introducing an auxiliary variable $\mathbf{X}$, and replacing positivity and box constraints by indicator functions:

$$\begin{aligned} \min_{\mathbf{X},\mathbf{V},\mathbf{W}} & \|\mathbf{Y} - \mathbf{X}\|_F^2 + \mathbf{I}_{[0,\infty)}(\mathbf{V}) + \mathbf{I}_{[0,1]}(\mathbf{W}) \\ s.t. & \quad \mathbf{X} - \mathbf{V}\mathbf{W} = \mathbf{0} \end{aligned} \tag{5}$$

where indicator function $\mathbf{I}_S(\mathbf{Z})$ equals 0 if every entry of matrix $\mathbf{Z}$ is in set S and equals ∞ otherwise. Next we form the augmented Lagrangian:

$$\begin{aligned} L(\mathbf{X},\mathbf{V},\mathbf{W},\mathbf{U}) = & \|\mathbf{Y} - \mathbf{X}\|_F^2 + \mathbf{I}_{[0,\infty)}(\mathbf{V}) + \mathbf{I}_{[0,1]}(\mathbf{W}) \\ & + (\mu/2)\|\mathbf{X} - \mathbf{V}\mathbf{W} + \mathbf{U}\|_F^2 - (\mu/2)\|\mathbf{U}\|_F^2 \end{aligned} \tag{6}$$

where $\mathbf{U}$ is the scaled dual variable and μ is the augmented Lagrangian parameter[1]. ADMM solves the augmented Lagrangian dual function by a sequence of convex subproblems where the biconvex function is decoupled:

$$\begin{aligned} (\mathbf{X}^{k+1}, \mathbf{V}^{k+1}) = & \arg\min_{\mathbf{X},\mathbf{V} \geq 0} \Big(\|\mathbf{X} - \mathbf{Y}\|_F^2 + \\ & (\mu/2)\|\mathbf{X} - \mathbf{V}\mathbf{W}^k + \mathbf{U}^k\|_F^2 \Big) \end{aligned} \tag{7}$$

$$\mathbf{W}^{k+1} = \arg\min_{0 \leq \mathbf{W} \leq 1} \|\mathbf{X}^{k+1} - \mathbf{V}^{k+1}\mathbf{W} + \mathbf{U}\|_F^2 \tag{8}$$

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \mathbf{X}^{k+1} - \mathbf{V}^{k+1}\mathbf{W}^{k+1} \tag{9}$$

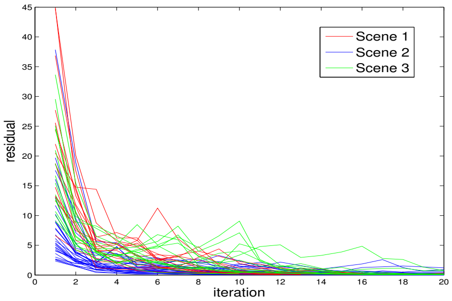


Figure 3. Convergence curve of Algorithm 1 showing residual error (4) vs. iteration number.

Algorithm 1 Semi-Binary Nonnegative Matrix Factorization

```
Input: A set of images $\mathbf{Y} \in \mathbb{R}^{+n \times m}$
Compute the number of lights $l = \mathrm{rank}(\mathbf{Y})$
Initialize $\mathbf{X}^0 = \mathrm{zeros}(n,m)$, $\mathbf{V}^0 = \mathrm{zeros}(n,l)$,
    $\mathbf{W}^0 = \mathrm{rand}(l,m)$, $\mathbf{U}^0 = \mathrm{zeros}(n,m)$, $\mu = 10^{-3}$, $k = 0$
while not converged do
    Update primal variables $\mathbf{X}^{k+1}$, $\mathbf{V}^{k+1}$, $\mathbf{W}^{k+1}$ according to (7) and (8)
    Update dual variable $\mathbf{U}^{k+1}$ according to (9)
end while
Round each entry of the indicator matrix $\mathbf{W}$ to binary
Solve for final lighting basis images $\mathbf{V}$ using (10)
Output: Basis images $\mathbf{V}$ and binary light indicator matrix $\mathbf{W}$
```

These subproblems are iteratively solved until convergence of primal and dual residuals [4]. Following that, we round each entry of $\mathbf{W}$ to $\{0,1\}$, and compute the final basis images $\mathbf{V}$ based on this binary indicator matrix using nonnegative least squares:

$$\min_{\mathbf{V}} \|\mathbf{Y} - \mathbf{V}\mathbf{W}\|_F^2 \quad s.t. \quad \mathbf{V}_{ij} \geq 0, \; \forall i,j. \tag{10}$$

Note that since $\mathbf{W}$ is constant in this optimization (10), the problem is convex. Our decomposition algorithm is summarized in Algorithm 1.

In Figure 3, we plot convergence curves of the proposed decomposition algorithm for 70 different light configurations (data matrices) from 3 different scenes (indicated with color code) starting from random initializations. In general, the algorithm converged in fewer than 20 iterations. Although the problem is non-convex, in 87% of the trials our algorithm recovered the true solution, while in the remaining ones it converged to a local optimum. In our implementation, we used the CVX optimization toolbox [6] to solve each convex subproblem.

3. Detecting 3D Geometric Boundaries

The set of images of a convex Lambertian surface under arbitrary variations of point lights at infinity forms a convex cone in $\mathbb{R}^n$, where the dimension of the cone depends on

[1] Here we use the scaled form of the augmented Lagrangian function in which the scaled Lagrangian multiplier is redefined as $\mathbf{U} = \mathbf{Z}/\mu$, where $\mathbf{Z}$ is the original Lagrange multiplier.

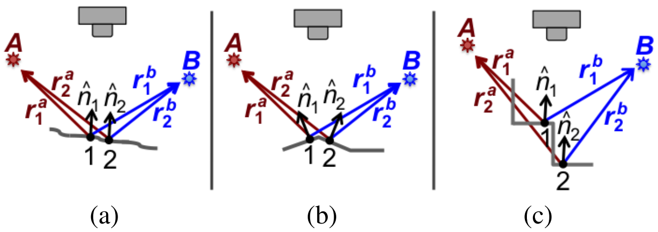


Figure 4. An image patch (containing points 1 and 2) may contain (a) no 3D geometric boundary, (b) a discontinuity in surface normal, or (c) a discontinuity in depth (from camera's point of view).

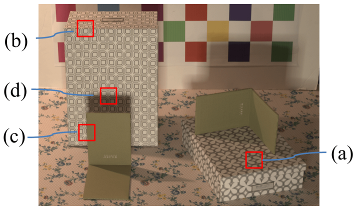


Figure 5. Categories of edges. Examples of patches containing 3D geometric boundaries: (b) discontinuity in normal and (c) discontinuity in depth. Examples of edges that are not 3D geometric boundaries: (a) texture edges and (d) shadow edge.

the number of distinct surface normals [2]. For typical indoor scenes, however, the distant lighting assumption is not valid. To allow for nearby lighting, we consider one small image patch at a time and analyze how the appearance of that patch varies over multiple lighting conditions. We show that if all pixels in a patch come from a single smooth surface in the scene, then the patch appearance across lightings forms a one-dimensional subspace. If the patch contains a 3D geometric boundary, however, then its appearance subspace will generally have dimension greater than one.

3.1. Illumination Subspace of Local Appearance

For simplicity, we justify our method here for Lambertian surfaces with only a direct lighting component, but an analogous argument will work for a broader class of reflectance functions and indirect lighting (e.g., multiple reflections). To simplify the explanation, we discuss only point light sources, because an extended isotropic light source can be arbitrary well approximated as a superposition of multiple point light sources.

Figure 4 shows three Lambertian scenes, each illuminated by two point light sources, A and B. (We explain the notation for source A; source B is analogous.) The surface normal at point i is $\hat{\mathbf{n}}_i$, and the vector from point i to light A is $\mathbf{r}_i^a$ (the corresponding unit vector is $\hat{\mathbf{r}}_i^a$). The intensity of the point on the image plane that corresponds to surface point i is I_i^a (for light source A) or I_i^b (for light B):

$$I_i^a = \gamma_i^a \frac{\hat{\mathbf{n}}_i^{\mathrm{T}}\hat{\mathbf{r}}_i^a}{\|\mathbf{r}_i^a\|^2} E^a \rho_i, \qquad I_i^b = \gamma_i^b \frac{\hat{\mathbf{n}}_i^{\mathrm{T}}\hat{\mathbf{r}}_i^b}{\|\mathbf{r}_i^b\|^2} E^b \rho_i. \tag{11}$$

Here $\hat{\mathbf{n}}_i^{\mathrm{T}}\hat{\mathbf{r}}_i^a$ is the cosine of the angle between $\hat{\mathbf{n}}_i$ and $\mathbf{r}_i^a$, E^a is the radiance intensity of light source A, and ρ^i is the surface albedo at point i. Binary value $\gamma_i^a = 1$ if point i is illuminated by source A, whereas $\gamma_i^a = 0$ if point i is not lit by source A due to an attached or cast shadow.

In each of the three scenes in Figure 4, points 1 and 2 are quite close from the perspective of the camera, so they will both be included in the same small image patch. In scene (a), the patch contains no sudden changes in normal and no depth discontinuities. Thus the 3D distance between points 1 and 2 is small compared to the distances from each point to each light, and hence for scene (a) we have the following approximate equalities:

$$\hat{\mathbf{n}}_1 \approx \hat{\mathbf{n}}_2, \qquad \mathbf{r}_1^a \approx \mathbf{r}_2^a, \qquad \mathbf{r}_1^b \approx \mathbf{r}_2^b. \tag{12}$$

Since in scene (a) all points in the patch share approximately the same normal and the same vector to each light source, we can eliminate the subscripts i in (11) and use $\hat{\mathbf{n}}$, $\mathbf{r}^a$, and $\mathbf{r}^b$ for all points in the patch. For now, we will also assume that every point i in the patch shares a single value for γ_i^a (which we will call γ^a) and shares a single value γ^b of γ_i^b, which means that for each light source, the entire patch is either illuminated by or shadowed from that light (the patch contains no shadow edges). We will consider shadow edges in Section 3.2.

Let $\mathbf{P}^a$ and $\mathbf{P}^b$ represent the vector of pixel intensities of the patch imaged under light A alone and light B alone, respectively. For the case in Figure 4(a), we have the approximate equality $\mathbf{P}^a = k^a \boldsymbol{\rho}$:

$$\underset{\mathbf{P}^a}{\begin{bmatrix} I_1^a \\ I_2^a \\ \vdots \end{bmatrix}} \approx \underbrace{\frac{\gamma^a E^a \hat{\mathbf{n}}^{\mathrm{T}}\hat{\mathbf{r}}^a}{\|\mathbf{r}^a\|^2}}_{k^a} \underset{\boldsymbol{\rho}}{\begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \end{bmatrix}}, \tag{13}$$

where the scalar k^a is constant across all pixels in the patch, and $\boldsymbol{\rho}$ is the vector of surface albedos for all of the pixels in the patch. For the same patch under light source B, we have the analogous equation: $\mathbf{P}^b = k^b \boldsymbol{\rho}$.

We have just demonstrated that if a patch contains no sudden changes in normal nor in depth (and no shadow edges), its pixel intensities under any light source will equal a scalar multiple of $\boldsymbol{\rho}$. In other words, the subspace spanned by the appearance of that local patch under all light sources (which we call the *illumination subspace of local appearance*) will have dimension 1. Note that this is true regardless of the surface texture (albedo). Even if the surface albedo of the patch contains high-contrast texture edges, its illumination subspace of local appearance will still be one-dimensional. This result is at the heart of our method for finding geometric edges, because the same will not generally be true if a patch contains a 3D geometric edge.

For example, if a patch contains a discontinuity in normal, as in Figure 4(b), then the first approximation in (12) does not hold, and the intensity of each point in the patch will depend on the cosine of the angle between its surface normal and its direction to the light source. If a patch contains a different type of 3D geometric edge, a depth discontinuity as in Figure 4(c), then the last two approximations in (12) do not hold (because the lights are not at infinity), and the intensity of each point in the patch will again depend on the cosine of the angle between its surface normal and its direction to the light source. In general, if a patch contains a 3D geometric edge, its illumination subspace of local appearance will have dimension greater than 1.

Confidence Map of 3D Geometric Boundaries We can now detect geometric boundaries by identifying patches whose illumination subspaces of local appearance have dimension greater than one. For each pixel location, we extract a τ-pixel patch centered at that location from m input images (m light combinations), and arrange them as column vectors in a $\tau \times m$ matrix, $\mathbf{Z}$:

$$\mathbf{Z} = [\mathbf{P}^{(1)}, \mathbf{P}^{(2)}, \ldots, \mathbf{P}^{(m)}], \tag{14}$$

where vector $\mathbf{P}^{(j)}$ contains all τ pixel values of the patch extracted from image j at that pixel location. To determine the rank of the illumination subspace of local appearance for that patch location, we apply Singular Value Decomposition (SVD) to $\mathbf{Z}$ and obtain the singular values $\{\sigma_i^{\mathbf{P}}\}$ (ordered largest to smallest). In the absence of noise, a one-dimensional illumination subspace would yield just one nonzero singular value $\sigma_1^{\mathbf{P}}$, with $\sigma_2^{\mathbf{P}} = 0$. For each pixel location, we compute a confidence value of the presence of a 3D geometric boundary as the ratio of the second to the first singular value for the patch centered at that location:

$$c(\mathbf{P}) = \sigma_2^{\mathbf{P}} / \sigma_1^{\mathbf{P}}. \tag{15}$$

3.2. Removing Shadow Edges

Parts (b) and (c) in Figures 4 and 5 illustrate both types of 3D geometric boundaries: (b) discontinuity in the normal and (c) discontinuity in depth. Our method of Section 3.1 successfully detects both types (b) and (c), and it is not fooled by texture edges (Figure 5(a)). However, shadow edges (Figure 5(d)) are often detected as false positives. A patch contains a shadow edge if for one of the light sources, some pixels of the patch are illuminated and others are in shadow. We have observed that in most cases, each shadow edge is caused by only a single light source. Based on this observation, we can use our ability to decompose a set of images of a scene into its single-light-source lighting basis images (Section 2.2) to eliminate most of the false positives caused by shadow edges.

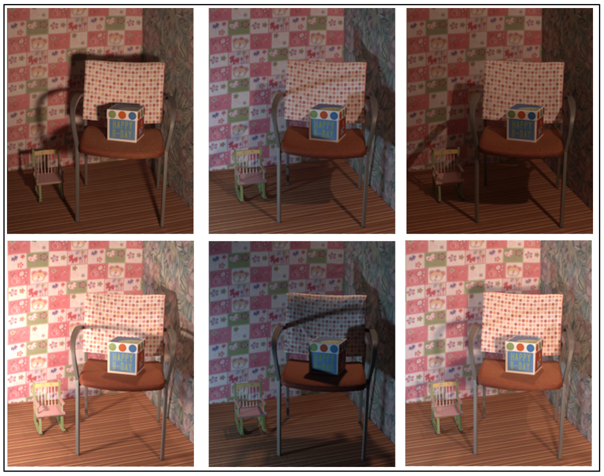


Figure 6. Sample input images (under various light combinations) of Scene 2.

We can eliminate the shadows produced by light source i by subtracting basis image $\mathbf{v}_i$ from the set of images $\mathbf{Y}$:

$$\mathbf{Y}^{(i)} = \mathbf{Y} - \mathbf{v}_i \mathbf{w}^i, \tag{16}$$

where $\mathbf{w}^i$ is the ith row of lighting indicator matrix $\mathbf{W}$, and $\mathbf{Y}^{(i)}$ denotes the scene images re-rendered with light i turned off.[2] Applying our boundary detection technique to $\mathbf{Y}^{(i)}$ results in a boundary confidence map $C^{(i)}$ in which the shadow edges resulting from the ith light source are eliminated. The final response map C is obtained by taking the minimum at each pixel location across all confidence maps $\{C^{(i)}\}_{i=1}^{l}$, so that if a shadow edge disappears when any one of the light sources is removed, that edge will not be present in the final response map.

We summarize our boundary detection procedure in Algorithm 2.

Algorithm 2 Detect geometric boundary from images under variable lighting

```
Input: A set of $m$ images $\mathbf{Y}$, lighting basis images $\{\mathbf{v}_i\}_{i=1}^{l}$
    and binary light indicator matrix $\mathbf{W}$.
for i=1,2,...,$l$ do
    Get image set $\mathbf{Y}^{(i)}$ using equation (16)
    At each pixel location, extract patches $\{\mathbf{P}^{(j)}\}_{j=1}^{m}$ from
        $\mathbf{Y}^{(i)}$, form matrix $\mathbf{Z}$ as in (14). Compute edge response
        map $C^{(i)}$ using (15) to get per-pixel confidence values.
end for
Output the final response map $C$ by taking the minimum at
    each pixel among $\{C^{(i)}\}_{i=1}^{l}$
```

4. Experiments

To evaluate our SBNMF method, we collected new datasets, which we use to compare our method with existing contour detection techniques.

[2] An alternative approach is to let $\mathbf{Y}^{(i)}$ equal the set of all of the lighting basis images excluding basis image i. This could be more stable if the set of input images is unbalanced (if some light sources are turned on more frequently than others).

4.1. Dataset Collection

We collected three datasets, each of which contains photographs of a different indoor scene under all possible lighting variations caused by turning individual light sources on or off. (See example images of the scenes in the figures throughout this paper.) For each scene, we set up five independent near-field (not at infinity) light sources, which ranged in extent from nearly point sources (exposed soft white lightbulbs) to area sources (portable fixtures containing multiple parallel fluorescent tubes). Each light was placed in a different location for each of the three scenes. By turning the 5 lights on/off in all combinations, we captured 32 different images of each scene with a Canon Rebel T2i DSLR camera. Although each dataset includes the single-light-source images, we did not use these as input images in any of our experiments, to ensure that our algorithm can handle difficult real-world situations in which we would have no control over the lighting. Figure 6 shows sample images from one of the datasets.

4.2. Scene Decomposition Results

For each run of our experiments on a dataset, we randomly choose a subset of the images from the dataset as our input image set, being sure not to select any of the five single-light-source images (which correspond to the lighting basis images we want to recover). We then apply Algorithm 1 to decompose this input image set to recover the lighting basis images $\mathbf{V}$ (examples from one scene are shown in Figure 7) and the indicator weights $\mathbf{W}$ that tell which lights were on/off in each input image. After recovering the lighting basis images, we can re-render the scene under new, unseen lighting conditions by displaying new linear combinations of the basis images (varying the coefficients continuously between 0 and 1). A video demonstrating these relighting results is in the Supplementary Material.

4.3. 3D Geometric Boundary Detection Results

We now apply our Algorithm 2 for detecting 3D geometric boundaries to input image sets from each of the 3 datasets, using square patches of size 3×3 pixels. We compare with the Canny detector [5] and the gPb detector [14], which is a state-of-the-art method for detecting boundaries in 2D images. Figure 8 displays detection results for two of the scenes. Since Canny and gPb are both designed for use on single images, we apply them to each input image individually. For Canny, the final set of edges is the union of the edges from all of the individual images, which we convert into a continuous-valued response map using the values of the Canny thresholds at which each edge is first detected. For gPb, we average the value of the probability map of [14] across all input images to obtain the final response map. The Canny detector [5], shown in (b) in Figure 8, cannot distinguish texture edges (nor shadow edges) from 3D geometric boundaries. The gPb [14] method (c) extracts the geometric contours of some foreground objects without being confused by small-scale texture, but it still detects large texture patterns in the background. Furthermore, gPb does not detect the contours of objects when they are at a small scale, nor when the background is similar in appearance to the objects (e.g., the arms and right front leg of the chair in Scene 2). In contrast, our approach is effective at distinguishing 3D geometric boundaries from both texture edges and shadow edges, giving true 3D geometric contours of objects the highest response map values. Our method can detect both curved geometric boundaries (e.g., the arms of the chair in Scene 2) and straight edges. At the same time, our method avoids false edge detections on smoothly curved surfaces that do not contain geometric boundaries (e.g., the concave top surface of the seat cushion in Scene 2).

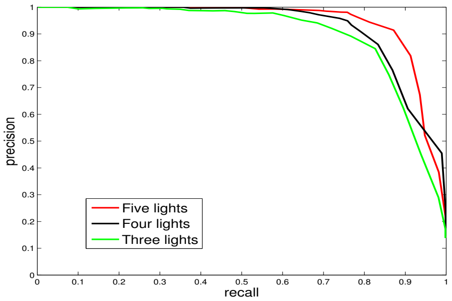


Figure 11. Precision-recall curves of our method on Scene 1 under different numbers of light sources. When the set of input images is generated using fewer light sources, performance degrades.

Quantitative evaluations on all three scenes verify the advantages of our method for detecting 3D geometric boundaries. For each scene, we manually annotated the ground truth 3D geometric boundaries. Figure 9 shows precision-recall (PR) curves for all methods tested. To evaluate our shadow edge removal method, we test our approach both before shadow edge removal (Section 3.1) and after shadow edge removal (Section 3.2 and Algorithm 2). Points on the PR curves were obtained by gradually varying the thresholds for each method's response map. In Figure 10, we compare binary edge detection results across methods on one of the scenes by setting the threshold for each method that corresponds to a recall value of 0.8. The PR curves demonstrate that the Canny detector [5] has the worst performance of the methods tested due to its false positives from texture edges. The gPb method [14] performs significantly better than Canny on Scenes 2 and 3 by not detecting the small-scale texture edges. Overall, both of our methods outperform the two existing methods by a large margin. Our approach after shadow edge removal gives particularly strong performance.

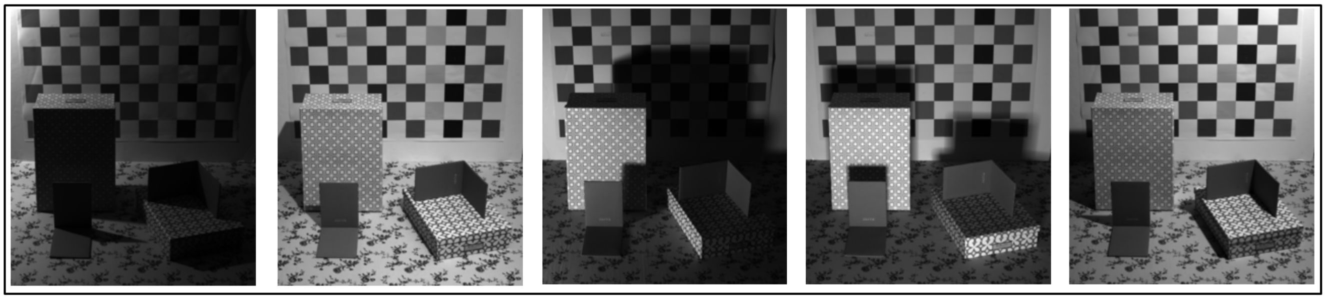


Figure 7. Scene decomposition results (recovered lighting basis images) for Scene 1.

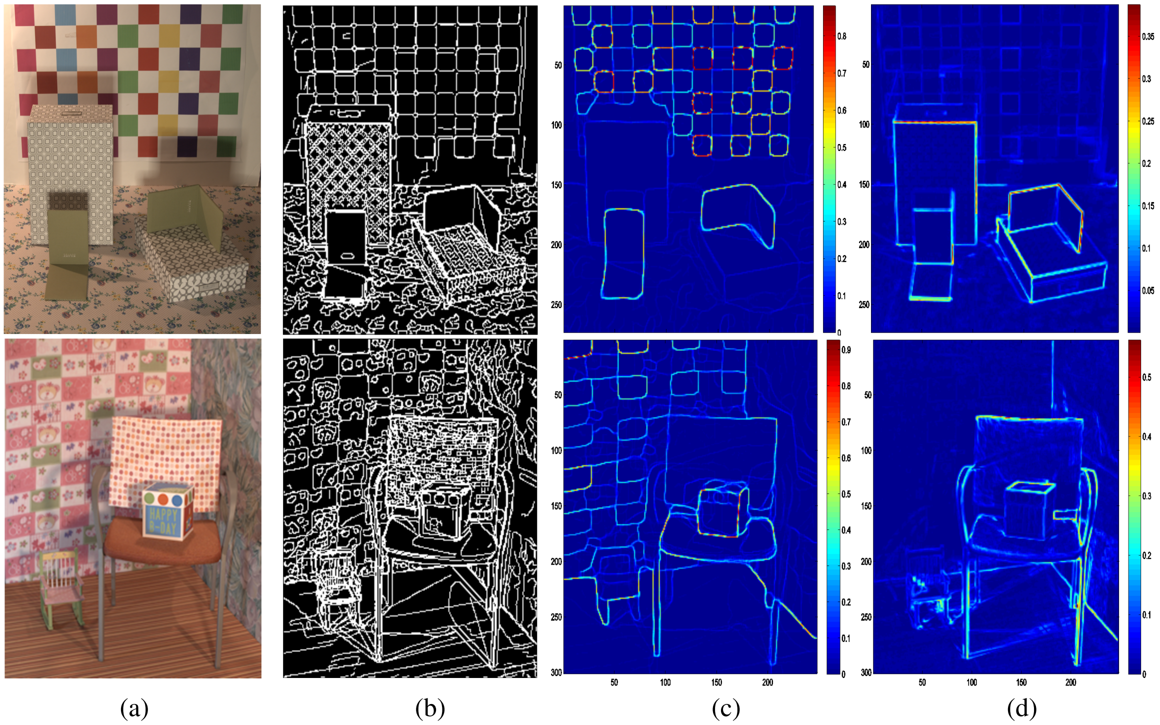
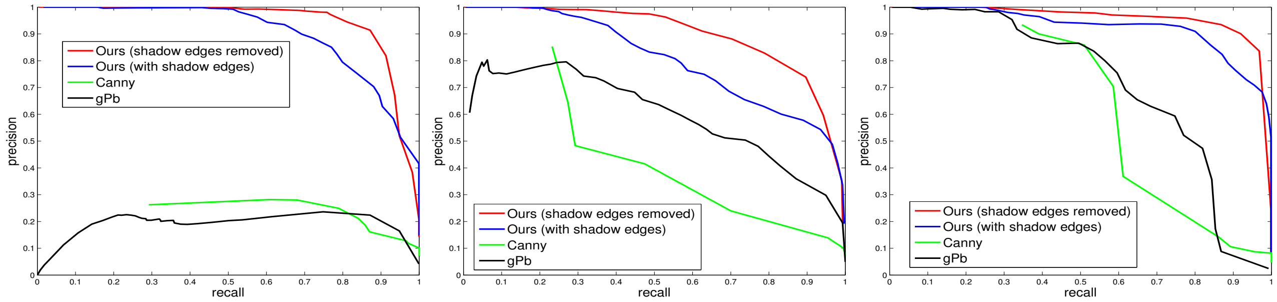


Figure 8. Edge detection results on Scene 1 (first row) and Scene 2 (second row): (a) Original image (b) Edges from Canny detector [5] (c) Probability of edges from gPb [14] (d) 3D geometric boundary response map of our approach.

Varying the Number of Light Sources To explore how the number of light sources available in a scene affects the detection of 3D geometric boundaries, we examined how performance was affected by selecting more and more restricted sets of input images (choosing subsets of the input set that had fewer light sources present). The precision-recall curves for three, four, and five light sources are compared in Figure 11. As expected, the additional information provided by increasing the number of light sources leads to improved boundary detection.

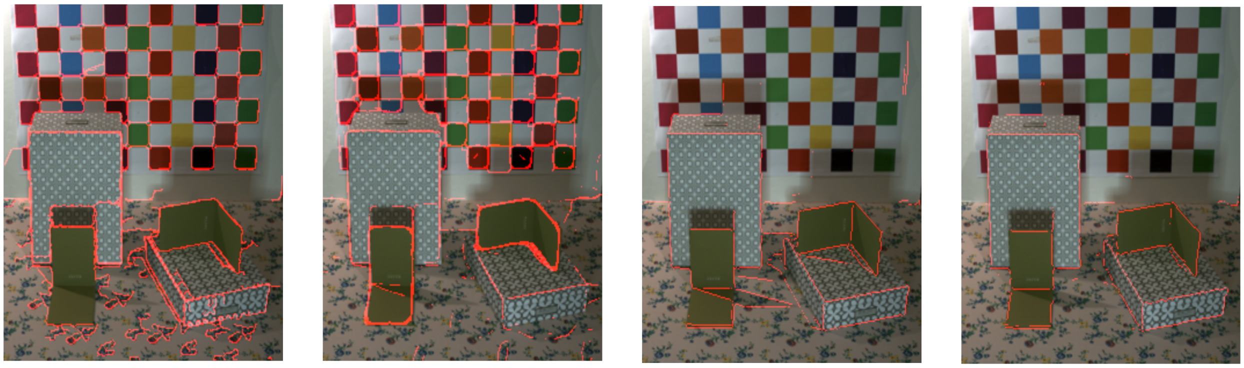
5. Conclusions and Future Work

We propose an image-based technique to identify geometric boundaries of an indoor scene with no prior knowledge of the scene geometry or light source positions. We provide a SBNMF method to factor a set of images under varying lighting into a set of lighting basis images. These basis images provide a natural scene representation and enable follow-up scene editing tasks such as relighting, as well as elimination of shadow edges. Our algorithms successfully factorize the scene, yielding boundary detection results that outperform state-of-the-art contour detection methods. In the future, we plan to extend our factorization method by considering continuous sources of illumination such as sunlight and skylight in addition to binary sources. In addition, boundaries inferred by our algorithm will benefit subsequent analysis such as image segmentation, object recognition, and inference of scene layout.



(a) (b) (c)

Figure 9. Comparison of precision-recall curves of different edge detectors on (a) Scene 1, (b) Scene 2, and (c) Scene 3.



(a) (b) (c) (d)

Figure 10. Thresholded binary edge detection results on Scene 1, (thresholded for recall 0.80): (a) Canny edges [5] (b) Edges from gPb [14] (c) Edges from our approach before shadow edge removal (Sec. 3.1) (d) Edges from our approach after shadow edge removal (Sec. 3.2).

References

[1] H. Barrow and J. Tenenbaum. Recovering intrinsic scene characteristics from images. *Comp. Vis. Systems*, 1978. 2

[2] P. N. Belhumeur and D. J. Kriegman. What is the set of images of an object under all possible lighting conditions? In *CVPR*, 1996. 4

[3] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52:155–173, Sep 2007. 2

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011. 3

[5] J. Canny. A computational approach to edge detection. *IEEE TPAMI*, 8(6):679–698, 1986. 1, 6, 7, 8

[6] CVX Research. CVX: Matlab software for disciplined convex programming, v. 2.0 beta. http://cvxr.com/cvx, 2012. 3

[7] E. Delage, H. Lee, and A. Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In *CVPR*, 2006. 1

[8] P. Espinace, T. Kollar, A. Soto, and N. Roy. Indoor scene recognition through object detection. In *ICRA*, 2010. 1

[9] V. Hedau, D. Hoiem, and D. A. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. 1

[10] C. Kim, J. Park, J. Yi, and M. Turk. Structured light based depth edge detection for object shape recovery. In *CVPR Workshop*, 2005. 1

[11] S. Koppal and S. Narasimhan. Appearance derivatives for isonormal clustering of scenes. *IEEE TPAMI*, 31(8):1375–1385, 2009. 1

[12] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2000. 2

[13] C. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Comp.*, 19(10):2756–2779, 2007. 2

[14] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *CVPR*, 2008. 1, 6, 7, 8

[15] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE TPAMI*, 26:530–549, 2004. 1

[16] F. Porikli. Multiplicative background-foreground estimation under uncontrolled illumination using intrinsic images. In *IEEE Workshop on Motion and Video Computing*, 2005. 2

[17] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009. 1

[18] R. Raskar, K.-H. Tan, R. Feris, J. Yu, and M. Turk. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. In *SIGGRAPH*, 2004. 1

[19] L. G. Roberts. *Machine Perception of Three-Dimensional Solids*. Garland Publishing, New York, 1963. 1

[20] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012. 1

[21] K. Sunkavalli, W. Matusik, H. Pfister, and S. Rusinkiewicz. Factored time-lapse video. In *ACM SIGGRAPH*, 2007. 1, 2

[22] K. Sunkavalli, T. Zickler, and H. Pfister. Visibility subspaces: uncalibrated photometric stereo with shadows. In *ECCV*, 2010. 1

[23] Y. Weiss. Deriving intrinsic images from image sequences. In *ICCV*, 2001. 2