Shape from Second-bounce of Light Transport

Siying Liu[1], Tian-Tsong Ng[1], and Yasuyuki Matsushita[2]

[1] Institute for Infocomm Research Singapore
[2] Microsoft Research Asia

Abstract. This paper describes a method to recover scene geometry from the second-bounce of light transport. We show that form factors (up to a scaling ambiguity) can be derived from the second-bounce component of light transport in a Lambertian case. The form factors carry information of the geometric relationship between every pair of scene points, *i.e.*, distance between scene points and relative surface orientations. Modelling the scene as polygonal, we develop a method to recover the scene geometry up to a scaling ambiguity from the form factors by optimization. Unlike other shape-from-intensity methods, our method simultaneously estimates depth and surface normal; therefore, our method can handle discontinuous surfaces as it can avoid surface normal integration. Various simulation and real-world experiments demonstrate the correctness of the proposed theory of shape recovery from light transport.

1 Introduction

Interreflections, reciprocal reflections among reflecting surfaces, are observed in all real-world scenes. The way light transports varies with scene geometry and surface reflectance. Clearly, there is a mutual dependency between the light transport and scene environment. This fact is used for scene modeling, *e.g.*, by Nayar [1] for scene geometry and reflectance, and also by Yu *et al*. [2] and Machida *et al*. [3] for modeling bidirectional reflectance distribution functions (BRDFs), when prior knowledge of the scene is available (pseudo geometry and reflectance for [1], and accurate scene geometry for [2, 3]). Recent advances in computational photography enabled modeling of inverse light transport [4–7] from photographs without prior knowledge of the environment. These works open up a new open problem — can we infer scene geometry only from the light transport without any prior knowledge?

In this paper, we propose a new approach to inferring scene geometry from the measured light transport without using any prior knowledge about the scene. We focus our discussion on a Lambertian case and model the scene as composed of planar patches. Our approach can be viewed as an *inverse* radiosity method where the scene geometry is unknown a priori as illustrated in Fig. 1. We first show a form factor matrix, which represents how much light is transported from one scene point to another purely by geometric factor, up to a scaling ambiguity, can be obtained from the *second-bounce* of light transport. Using the form factor

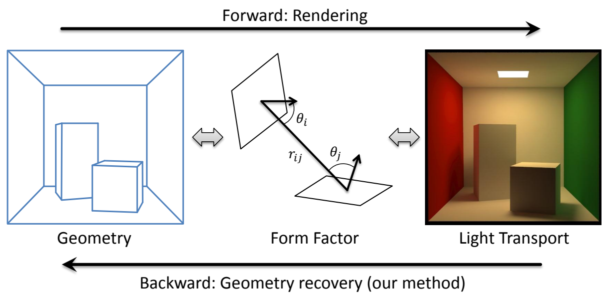

Fig. 1. The relationship between scene geometry and light transport. The forward case is a rendering process where light transport can be computed from known scene geometry, while the backward case is where the scene geometry is inferred from light transport.

matrix, we show that scene geometry can be recovered up to a scaling ambiguity. We develop a solution method for simultaneously estimating surface orientations and scene depths from the form factor matrix by optimization.

The primary contributions of this paper are twofold. First, it introduces the use of the second-bounce of light transport for recovering scene geometry. We describe the relationship between the scene geometry and the light transport and show what information is carried in the second-bounce component about the scene. To this end, we show that the scene geometry can be recovered up to a scaling ambiguity as well as diffuse albedo ratios. Second, the proposed method is effective even when the surface of interest has discontinuity. Unlike prior shape-from-intensity methods, our method simultaneously estimates surface orientation and depth (up to scaling ambiguity). This allows us to avoid integration of surface orientations; therefore, the assumption of continuous surface is no longer needed unlike other shape-from-intensity methods.

1.1 Prior work

Forward light transport is well studied in computer graphics such as in ray tracing [8] and radiosity [9, 8]. These use known scene geometry and BRDFs for producing photorealistic images. More recently, photographic modeling of forward light transport is drawing attention [10–13]. These methods take a number of images under different lightings for recording various complex lighting effects.

In graphics, inverse global illumination was introduced by Yu *et al*. [14] for estimating reflectance properties, rather than for geometry estimation. Inverse light transport is also used in computer vision. Seitz *et al*. [4] showed a method for estimating n-bounce component of light transport by probing a scene using a narrow beam light. Ng *et al*. [7] extended the method using a stratified matrix inversion for radiometric compensation of projector-camera systems. Nayar *et al*. [5] proposed a fast method for separating direct and global component of light transport using high frequency illumination. Gupta *et al*. [6] later discussed the relation between global illumination and defocused illumination.

The goal of this paper is shape recovery from the measured light transport. The closest to our work is Nayar *et al*. [1]. They proposed an iterative photometric stereo algorithm, which is the first work that uses interreflections as a useful cue for shape and reflectance estimation. Our method is different from their approach in that we infer scene geometry directly from the second-bounce component of light transport instead of relying on photometric stereo. In addition, our method is not limited to continuous surfaces because our method does not require integration of the surface orientations, but simultaneously estimates surface orientation and depth. On the other hand, compared with their method, our method requires more images as input for obtaining the second-bounce component of the light transport.

Apart from shape-from-intensity methods, other prior art on shape or depth recovery include shape from structured light [15] and structure-from-motion (SfM) [16]. Both approaches use triangulation for determining depths. In terms of calibration requirements, these methods require calibration of intrinsic parameters of the imaging devices, while our method does not require intrinsic calibration.

Our method uses form factors for shape estimation. The computation of form factors has a long history back to Lambert in 1760 [17]. Schröder and Hanrahan derived a closed-form solution for the case of general polygons [18]. Our method uses form factors in an inverse manner for estimating the scene geometry.

2 Interreflection and Scene Geometry

2.1 Forward case: the Rendering Equation

The rendering equation [19] is written as

$$L_{out}(\mathbf{p}, \omega_o) = L_e(\mathbf{p}, \omega_o) + \int_{M^2} \rho(\mathbf{p}, \omega_i, \omega_o) L_{out}(\mathbf{p}', -\omega_i) V(\mathbf{p}, \mathbf{p}') \frac{\cos\theta_i \cos\theta_o}{\|\mathbf{p} - \mathbf{p}'\|^2} dA_{\mathbf{p}'}, \tag{1}$$

where $L_{out}(\mathbf{p}, \omega_o)$ is the reflected or outgoing radiance in direction ω_o, L_e is the emission corresponding to light sources, ρ is the Bidirectional Reflectance Distribution Function (BRDF) of the scene, and V is the binary visibility function. The visibility function $V(\mathbf{p}, \mathbf{p}')$ is 1 if scene points $\mathbf{p}$ and $\mathbf{p}'$ are connected by a line of sight and 0 otherwise. The integral is over the area of M^2 of all scene surfaces, and weighted by a purely geometric factor known as the *form factor*.

The above rendering equation applies for a continuous surface. Discretization of the surface leads to a matrix representation. For a surface with n facets,[3]

[3] In this paper, we use the term “facet” to describe the smallest piece of a surface subdivision and the term “patch” for any larger pieces, up to and including the biggest polygons formed by combining facets.

radiance and albedo values are assumed to be constant over each facet, then the rendering equation can be written in operator notation as [20]:

$$\mathbf{l}_{out} = \mathbf{l}_e + \mathbf{KGl}_{out} = \mathbf{l}_e + \mathbf{Al}_{out}, \text{ where } \mathbf{A} = \mathbf{KG}. \tag{2}$$

$\mathbf{l}_{out}$ is a vector of $L_{out}(\mathbf{p}, \omega_o)$, $\mathbf{l}_e$ is a vector of $L_e(\mathbf{p}, \omega_o)$, $\mathbf{G}$ is a purely geometric operator that takes outgoing or reflected radiance and propagates it within the scene to obtain incident radiance, and $\mathbf{K}$ is a local linear reflection operator based on the BRDF of the surface:

$$\mathbf{K} = \begin{bmatrix} \rho_1 & 0 & \cdots & 0 \\ 0 & \rho_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \rho_n \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 0 & G_{12} & \cdots & G_{1n} \\ G_{21} & 0 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ G_{n1} & \cdots & \cdots & 0 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0 & \rho_1 G_{12} & \cdots & \rho_1 G_{1n} \\ \rho_2 G_{21} & 0 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \rho_n G_{n1} & \cdots & \cdots & 0 \end{bmatrix}. \tag{3}$$

The interreflections between two points or facets $\mathbf{p}_i$ and $\mathbf{p}_j$ can be described by the G_{ij} expression[4]:

$$G_{ij} = \frac{V(\mathbf{p}_i, \mathbf{p}_j) \cos\alpha \cos\beta}{\|\mathbf{r}_{ij}\|^2} = \frac{V(\mathbf{p}_i, \mathbf{p}_j)(-\widehat{\mathbf{r}}_{ij} \cdot \widehat{\mathbf{n}}_i)(\widehat{\mathbf{r}}_{ij} \cdot \widehat{\mathbf{n}}_j)}{\|\mathbf{r}_{ij}\|^2}, \tag{4}$$

where $\mathbf{r}_{ij} = \mathbf{p}_j - \mathbf{p}_i$, α and β are the angles between $\mathbf{r}_{ij}$ and their respective surface normals. G_{ii} is undefined for any i, and G_{ij} vanishes if $\mathbf{p}_i$ and $\mathbf{p}_j$ are mutually invisible.

2.2 Backward case: From Light Transport T to Form Factor G

Following [19] and Eq. (2), we can obtain

$$\mathbf{l}_{out} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{l}_e. \tag{5}$$

Assuming the camera does not see the light source directly, and we do not have emissive surfaces, we can replace $\mathbf{l}_e$ with the effective emission that corresponds to the direct reflection from the light source, $\mathbf{l}_e = \mathbf{Fl}_{in}$, where $\mathbf{l}_{in}$ is the incident light from a light source such as a projector, and $\mathbf{F}$ is the light transport matrix that corresponds to the first-bounce reflection. Assuming a focused light source, the first-bounce matrix $\mathbf{F}$ is diagonal. Hence, we have

$$\mathbf{l}_{out} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{Fl}_{in} = \mathbf{Tl}_{in}, \ \mathbf{T} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{F}, \tag{6}$$

where $\mathbf{T}$ is the light transport matrix. Hence, we can write $\mathbf{A}$ in terms of $\mathbf{T}$ as

$$\mathbf{A} = \mathbf{I} - \mathbf{FT}^{-1}. \tag{7}$$

[4] In [1], the geometric kernel takes into account the effective area of the illuminator facet. Here, we assume facets i and j are interchangeably the illuminator and reflector and have sufficiently small area *s.t.* G_{ij} and G_{ji} are approximately equal.

With Neumann series expansion, we can expand a light transport matrix into a matrix series where the second term corresponds to the second-bounce:

$$\mathbf{T} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{F} = (\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \cdots)\mathbf{F}. \tag{8}$$

We can see that the matrix $\mathbf{A}$ is related to the second-bounce of light transport.

According to [4], for a Lambertian scene, the diagonal elements of $\mathbf{F}$ are given by the reciprocals of the diagonal elements of $\mathbf{T}$:

$$\mathbf{F}[i,i] = \frac{1}{\mathbf{T}^{-1}[i,i]}. \tag{9}$$

Therefore, if we limit our discussion to the Lambertian case, $\mathbf{A}$ can be computed using Eq. (9) and Eq. (7). For geometry estimation, we can extract $\mathbf{G}$ from $\mathbf{A}$. Given $\mathbf{A}$, we can compute the relative albedo ρ_{ij} for all the scene points [5]:

$$\rho_{ij} \doteq \frac{A_{ij}}{A_{ji}} = \frac{\rho_i G_{ij}}{\rho_j G_{ji}} = \frac{\rho_i}{\rho_j}. \tag{10}$$

Given ρ_{ij}, we can recover $\mathbf{G}$ up to a scale:

$$\mathbf{A} = \tilde{\mathbf{K}}\tilde{\mathbf{G}} \text{ and } \tilde{\mathbf{G}} = \tilde{\mathbf{K}}^{-1}\mathbf{A}, \tag{11}$$

where

$$\tilde{\mathbf{K}} = \frac{1}{\rho_j}\mathbf{K} = \begin{bmatrix} \rho_{1j} & 0 & \cdots & 0 \\ 0 & \rho_{2j} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \rho_{nj} \end{bmatrix}, \quad \text{and} \quad \tilde{\mathbf{G}} = \rho_j \mathbf{G}. \tag{12}$$

3 Geometry extraction from the geometric form factors

For two mutually visible points [6] $\mathbf{p}_1$ and $\mathbf{p}_3$ as shown in Fig. 2 (a), the geometric form factor is given by $G_{13}(\mathbf{r}_{13}, \widehat{\mathbf{n}}_1, \widehat{\mathbf{n}}_3) = \frac{(-\widehat{\mathbf{r}}_{13}\cdot\widehat{\mathbf{n}}_1)(\widehat{\mathbf{r}}_{13}\cdot\widehat{\mathbf{n}}_3)}{\|\mathbf{r}_{13}\|^2}$. For mutually visibility, we need to have $(\widehat{\mathbf{r}}_{13} \cdot \widehat{\mathbf{n}}_1) > 0$ and $(\widehat{\mathbf{r}}_{13} \cdot \widehat{\mathbf{n}}_3) < 0$. Resolving $\widehat{\mathbf{n}}_i$ and $\mathbf{r}_{ij}$ for all scene points recovers both depth and surface normal. However, this method is unable to recover geometry for scene points which are not visible by others, *e.g.*, geometry extraction is impossible for a globally convex surface. In this section, we will examine the settings under which the scene geometry can be extracted.

[5] As an observation, given the relative albedo ρ_{ij}, we can recover the absolute albedo value for all scene points as long as the absolute albedo value of one of the scene points is known.

[6] A discrete surface is composed of small facets that are often assumed to have uniform property. Hence, a discrete facet is conceptually similar to a discrete point. For ease of discussion, we may use the term "facet" and "point" interchangeably.

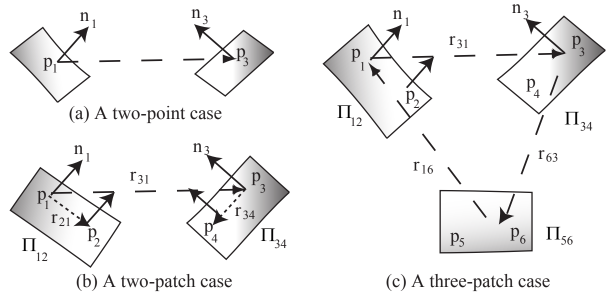

Fig. 2. Three different setup for geometry extraction from the geometric form factor terms.

3.1 Problem Setup and Assumptions

In this work, we consider a light transport acquisition setup with a projector-camera system. Assuming no serious scattering due to the transmission media or subsurface scattering, the directional nature of the projector light allows correspondence between the projector pixels and scene points to be established. The directional light from the projector is perspective in nature. However, if we assume that the scene depth is small, the projection is approximately orthographic. With this assumption, the problem of geometry extraction is greatly simplified, as we can assume that the correspondence points in the scene approximately preserve the grid structure of the projector pixels. In a coordinate frame where the z-axis is aligned with the optical axis of the projector, we can assume that the x-y coordinate of the scene points form a rectangular grid, which is known up to a scale, while the z-coordinate is the only unknown.

3.2 A case with two scene points

In the case with just two scene points as in Fig. 2 (a), knowing the value of the form factor is not sufficient to recover the surface normal nor the depth uniquely. To see the set of all possible solutions, we can rewrite Eq. (4) as

$$\widehat{\mathbf{r}}_{13} \cdot \widehat{\mathbf{n}}_1 = -\frac{G_{13}\|\mathbf{r}_{13}\|^2}{\widehat{\mathbf{r}}_{13} \cdot \widehat{\mathbf{n}}_3}. \tag{13}$$

In our setting, for the scene point position in $\mathbb{R}^3$, only the z-coordinate is unknown. As Eq. (13) can only be unique up to a relative depth in z-direction, there is no loss of generality to fix the z-coordinate for one of the scene point, say $\mathbf{p}_3$. Then, the distance vector is governed by z_1, *i.e.*, the z-coordinate of $\mathbf{p}_1$, alone. For every $\widehat{\mathbf{n}}_1$ in Eq. (13), it is possible to find a $\mathbf{r}_{13}$ for all $\widehat{\mathbf{n}}_3$. As $\widehat{\mathbf{n}}_1$ and $\widehat{\mathbf{n}}_3$ are unit normal vectors, they live in a spherical space $\mathbb{S}^2$. Hence, the space of all solutions $(\widehat{\mathbf{n}}_1, \widehat{\mathbf{n}}_3) \in \mathbb{S}^2 \times \mathbb{S}^2$, which is highly ambiguous. This also indicates that having more independent pairs of points does not help, as the normals are totally unconstrained.

3.3 A case with two patches

To resolve the ambiguity in the form factor expression, we need to introduce more constraints. One way to do so is to group a set of adjacent points to form a patch where the points share a common surface normal. Fig. 2 (b) shows an example of two patches, each having two points. The newly introduced constraints are

$$\widehat{\mathbf{n}}_1 = \widehat{\mathbf{n}}_2, \quad \widehat{\mathbf{n}}_3 = \widehat{\mathbf{n}}_4, \quad \widehat{\mathbf{r}}_{12} \cdot \widehat{\mathbf{n}}_1 = 0, \text{ and } \widehat{\mathbf{r}}_{34} \cdot \widehat{\mathbf{n}}_3 = 0. \tag{14}$$

With the four scene points, we have four distinctive and non-zero form factor terms, *i.e.*, G_{13}, G_{14}, G_{23} and G_{24}. Altogether, there are 5 unknowns, *i.e.*, $\widehat{\mathbf{n}}_1$, $\widehat{\mathbf{n}}_3$, $\widehat{\mathbf{r}}_{12}$, $\widehat{\mathbf{r}}_{13}$ and $\widehat{\mathbf{r}}_{34}$, with 7 degrees of freedom, where $\widehat{\mathbf{r}}_{ij}$ has only 1 degree of freedom as we fix the (x, y) coordinate. Given the 6 equations, the solution space is 1 dimensional. The system is sufficiently constrained if we add another point to either patch, as it introduces 1 additional unknown but 3 more equations. Therefore, in our algorithm, we group 3 points in a patch.

3.4 A case with three patches

As shown in Eq. (12), we can only obtain the form factor up to an unknown albedo value, therefore the actual expression for the form factor term for two mutually visible points $\mathbf{p}_1$ and $\mathbf{p}_3$ is

$$G_{13}(\mathbf{r}_{13}, \widehat{\mathbf{n}}_1, \widehat{\mathbf{n}}_3) = C\frac{(-\widehat{\mathbf{r}}_{13} \cdot \widehat{\mathbf{n}}_1)(\widehat{\mathbf{r}}_{13} \cdot \widehat{\mathbf{n}}_3)}{\|\mathbf{r}_{13}\|^2}, \tag{15}$$

where C is an unknown constant. In the case of uncalibrated projector, the constant C is also needed to account for the unknown scale inherent in the (x, y) coordinate for scene points.

To disambiguate C, we check for the geometric consistency with three patches as shown in Fig. 2 (c). As the solution obtained by evaluating the form factor terms for patch pairs (Π_{12}, Π_{34}) and (Π_{12}, Π_{56}) should agree with that for (Π_{12}, Π_{56}), we can validate the solution of the former with that of the latter. The solutions should best tally when we choose a correct constant C. In the setting of Fig. 2 (c), there are 9 unknowns with 12 degrees of freedom while having 15 equations gives a sufficiently constrained solution space. Without any assumption, the constant C in Eq. (15) is fundamentally unresolvable, as there is a physically feasible surface geometry with a different albedo corresponding to a C. However, with the orthographic assumption mentioned in Sec. 3.1, the constant C is no longer linearly related to depth. With an incorrect C, geometry can be inconsistent in the triangular patch configuration of Fig. 2 (c). Hence, the orthographic assumption breaks the scale ambiguity.

4 Algorithm

As shown in Sec. 3, we need to group at least three scene points into patches in order to obtain a sufficiently constrained system. As another issue, the geometry derived from disjoint point sets will be in different coordinates. In this section, we will look into the criterion for point grouping and the way to bring the geometry at disjoint coordinates into the global coordinate. We will also look at efficient ways for geometry extraction through hierarchical computation or incorporating the prior knowledge of planar surface in the scene.

4.1 Point grouping

The assumption and guiding principle for point grouping is essentially based on the co-planar property of a point set. An arbitrary set of points is not guaranteed to be co-planar. Hence, we select points that satisfy the following criterion:

- *Adjacency*: The points are adjacent to each other.
- *Mutual invisibility*: Two points with $V_{ij} = 0$ are not mutually visible. $V_{ij} = 0$ implies $G_{ij} = 0$.

Two points satisfying the above criterion are likely to be co-planar. In our implementation, we consider points in a 2×2 neighborhood as being adjacent.

4.2 Pairwise patch selection

To make the form factor expression in Eq. (4) more succinct, for three co-planar points $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ sharing a common unit normal vector $\widehat{\mathbf{n}}_1$, we can express $\widehat{\mathbf{n}}_1$ as

$$\widehat{\mathbf{n}}_1 = \frac{\mathbf{r}_{12} \times \mathbf{r}_{23}}{\|\mathbf{r}_{12} \times \mathbf{r}_{23}\|}, \text{ where } \mathbf{r}_{ij} = \mathbf{p}_j - \mathbf{p}_i. \tag{16}$$

If there is an additional point $\mathbf{p}_4$ on the same patch, we need to introduce a constraint to ensure co-planarity:

$$(\mathbf{r}_{12} \times \mathbf{r}_{23}) \cdot \mathbf{r}_{24} = 0. \tag{17}$$

As two points on a patch are having $G_{ij} = 0$ to begin with, we assume that the constraints such as Eq. (17) are automatically satisfied and will not form part of the equations that we are solving. Hence, for two mutually visible patches with N points each, there are $N \times N$ equations for G_{ij} with $2N$ unknowns which correspond to the z-coordinate of the $2N$ points, thus forming a sufficiently constrained system.

In practice, as G_{ij}'s are obtained from measurement, the G_{ij}'s with a low intensity tend to have a low signal to noise ratio and should not be used for computation. As a result, we can have fewer equations while the number of unknowns remains unchanged. To ensure that a patch pair forms a constrained system, we use the following criteria to select a pair of patches with N_a and N_b points respectively:

$$\sum_{i=1}^{N_a} \sum_{j=1}^{N_b} \mathbf{1}(G_{ij} > \epsilon) \geq N_a + N_b \text{ where } \mathbf{1}(\text{true}) = 1 \text{ and } \mathbf{1}(\text{false}) = 0 \tag{18}$$

Algorithm 1 "Closed-loop" check for reconstructed patches

Require: A list $\mathbf{L}$ of valid patch pairs based on Eq. (18)

1: Sort $\mathbf{L}$ in descending order of the sum of entries in $\mathbf{G}$ for all patch pairs.
2: Start with any patch from the first pair in $\mathbf{L}$ and treat it as a parent patch.
3: Record visited patches into a list $\mathbf{L}_v$. List is blank for the initial patch.
4: **repeat**
5: For a parent patch, identify all its possible branches.
6: Find the best branch based on the sorted list $\mathbf{L}$. If the best branch is in $\mathbf{L}_v$, take the next one.
7: Reconstruct the patch pair and update the depth map and normal map.
8: Push the traversed patch into $\mathbf{L}_v$.
9: Use the traversed patch as parent and Goto 1.
10: **until** The branch patch is the same as the starting patch.
11: Compute the depth error between the initial and final patches.

In our algorithm, we reconstruct the geometry for a pair of patches at a time and then bring the resultant disjoint geometry into the same coordinate frame through the common points connecting the different pairs of patches. However, if there is no direct or indirect visibility link between two points, bringing them into a common coordinate is impossible. The condition for the existence of a global coordinate for all points is that the form factor matrix $\mathbf{G}$ forms a fully connected graph. For every point in a common coordinate frame, we verify its geometry by examining the depth consistency in the closed paths associated to the point. Such closed paths could be many, therefore we only consider the one with the highest intensity and involving at least two other points on different patches. In this process, we are able to identify the reliability of the geometry reconstruction for a point. This consistency check through a "closed-loop" algorithm is presented in Algorithm 1. It is also intended to disambiguate the unknown constant as described in Sec. 3.4.

To increase the reliability of geometry estimation, we perform the above-mentioned steps in a hierarchical manner, from a finer resolution to a coarser one. At one level, we estimate the geometry and group points with similar normals into patches. The patch size grows with increasing level, hence the system of equations for pairwise reconstruction gets more and more constrained and produces more reliable estimation.

Fast method for piece-wise planar scenes: If the scene is known to be piece-wise planar a priori, it is more efficient to adopt a top-down approach for the reconstruction. Except for convex surfaces that do not interact with each other, planar surfaces correspond to "blocks" of zeros. It is worth-noting that the form factor matrix resembles the weight matrix $\mathbf{W}$ in the *Normalized Cut* [21] problem. With this observation, we can segment the scene into planar surface.

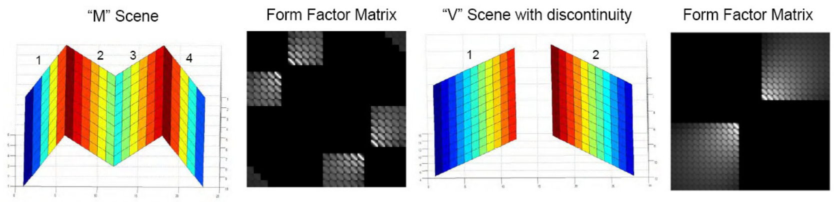

Fig. 3. From left to right: A simulated "M" scene with 11×23 facets and its $\mathbf{G}$ matrix; a simulated inverted "V" scene with discontinuity made of 12×28 facets and its $\mathbf{G}$ matrix. The inner wedge of the "M" scene is made up of 2 convex planes which do not illuminate each other. Note the discontinuity between the 2 planes in the inverted "V" scene. The form factor matrices are log-scaled for display purposes.

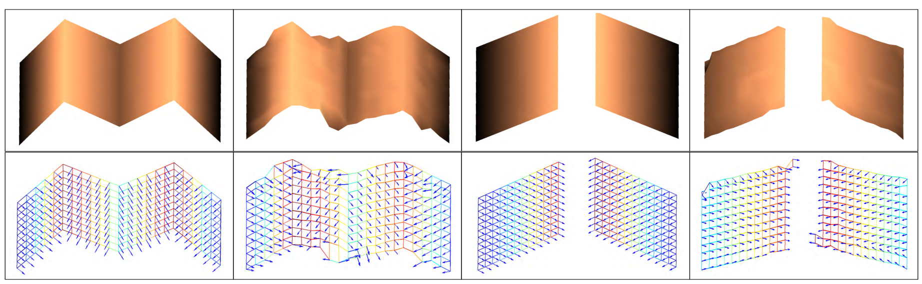

Fig. 4. Top row: Reconstruction results for both clean and noisy simulated "M" and inverted "V" scenes. Bottom row: The recovered surface normal corresponding to the scenes in the top row (normal plotted in opposite directions for display purpose).

5 Experimental Results

To verify our theoretical results, we performed experiments on both synthetic and real data. For synthetic scenes, the reconstruction is based on simulated form factor matrices; while for the real data, the light transport $\mathbf{T}$ of the scene is measured and the form factor matrix $\mathbf{G}$ is derived from $\mathbf{T}$.

5.1 Synthetic scene

For this experiment, we focus on recovering the shape of simulated 3-D models. To demonstrate the robustness of the proposed method, we perturbed the form factor matrix by additive Gaussian noise. Fig. 3 shows the simulated models and their corresponding form factor matrices.

Fig. 4 shows the reconstruction results for both simulated scenes, using both clean and noisy data. In the noiseless case, perfect recovery of both surface normal and depth can be achieved. Observe that the scale of the reconstruction is the same as the data as we begin with a form factor matrix in the scene's

Table 1. Shape recovery result. Normal RMSE and angular error between a pair of surfaces are shown. (all errors are measured in degree)

Scene	Normal Angular RMSE	$\angle$ planes 1&2	$\angle$ planes 2&3	$\angle$ planes 3&4
"M" (clean)	0.0018	0.0	0.0	0.0
"M" (noisy)	14.13	5.60	4.30	10.20
inv. "V" (clean)	0.02	0.0	-	-
inv. "V" (noisy)	13.11	6.57	-	-

coordinate frame. The recovered structures are subject to a translation in the z-direction as the global depth reference point was arbitrarily set. For the noisy case, both form factor matrices are corrupted by zero-mean Gaussian noise of standard deviation 0.5. In the presence of noise, the shape is better recovered at places such as the joint of 2 planes where the interreflections are stronger. As compared to the recovered depth values, the surface normals are better recovered because they are common among all facet pairs in a particular system of equations. For performance evaluation, we computed the angular error between all estimated surface normals and their ground truth. We also compared the recovered angles between planes with their ground truth in the simulated models. The results are presented in Table. 1.

Handling surface discontinuity: To highlight the proposed method's strength in handling depth discontinuity, we simulated an inverted "V" scene with a gap in the center. Fig. 4 shows the successful reconstruction of this scene. Unlike most shape-from-intensity methods which require the surface to be continuous for the integration of surface orientation, our method is not restricted by surface continuity. Facets lying on occluding boundaries do not have interactions with the rest and therefore do not form any valid equations with them. As a result, these facets will be left unreconstructed since there is insufficient information to determine their relative positions from the others. The same applies to facets lying on the joint between 2 planes. As it is co-planar with both planes, its form factor with facets on both planes equates to 0.

Handling constant factor in G: The constant factor C in Eq. (15) can be determined empirically through closed-loop checks. As we have fixed the x-, y-components of the distance vector and evaluate only the z-component, such a scaling would cause a non-linear change in z. If this factor is not compensated for, the error will show up in the closed-loop check as it propagates through all pair-wise depth estimation before looping back to the starting patch. The error here is defined as the minimum depth error among all close-loop paths. Hence, we can conduct a coarse-to-fine 1-D search to determine the correct factor to cancel C off. To see how C affects the reconstruction, we simulated the recovery of a "V" scene by fitting in different values. The results are presented in Fig. 5. Note that C is being multiplied into the form factors in this experiment but in

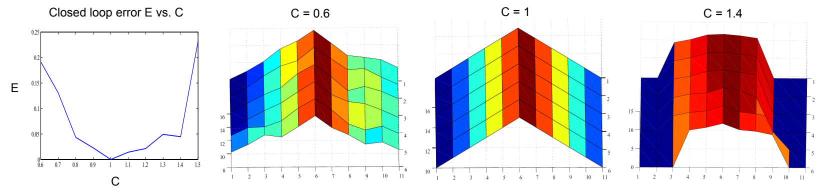

Fig. 5. Top left: A plot of closed-loop error versus C. Second from top left onwards: Reconstruction results by setting C to various values. The recovered shape gets distorted as C deviates from 1.

reality we seek to find a reciprocal to compensate for C. As C deviates from 1, the distortion causes planar surfaces to bend as z changes non-linearly with C.

5.2 Real-world scene

In this experiment we seek to recover the shape of a real-world scene from its measured light transport matrix. For experimental setup, we used a Canon 5D camera and a Dell 2400MP projector. In our experiment, we consider grayscale light transport for simplicity by assuming that the projector-camera color mixing matrix is diagonal. To ensure interreflections is faithfully measured, we used High Dynamic Range capturing with 12 stops of exposures to acquire $\mathbf{T}$ by a brute-force method. The acquired $\mathbf{T}$ is verified by bounce separation. In general, the light transport matrix $\mathbf{T}$ obtained by a projector-camera system has a dimension of $N_c \times N_p$, where N_c and N_p are respectively the number of camera pixels and projector pixels. In this work, we establish a pixel mapping between the camera and the projector by corresponding a camera pixel to the projector pixel that induces a maximum response on it. In our setup, there are more than one camera pixels being mapped to a projector pixel and we group these camera pixels to form a super-pixel. The intensity of a super-pixel is given by the mean of the group of camera pixels. With super-pixels, the resulting $\mathbf{T}$ matrix takes a square dimension of $N_p \times N_p$. If a super-pixel corresponds to a facet in the scene, with the pixel grouping procedure, we are inherently making uniform-intensity facet assumption.

Fig. 6(a) shows the result of a real "M" scene. The angle between planes 1 and 2 is 80^o and that between planes 3 and 4 is 55^o. (b) shows the derived form factor matrix $\mathbf{G}$. For the real data, we first determine the unknown scale factor C $(= 0.5)$ and multiply $\mathbf{G}$ by $\frac{1}{C}$. Fig. 6(c) and (d) show the recovered normal map and shape; (e) shows the final result after plane fitting. The reconstructed shape is quite close to the original scene. The estimated angle between planes 1 and 2 is 70.43^o and that between planes 3 and 4 is 50.55^o, giving rise to angular errors of 9.57^o and 4.45^o respectively.

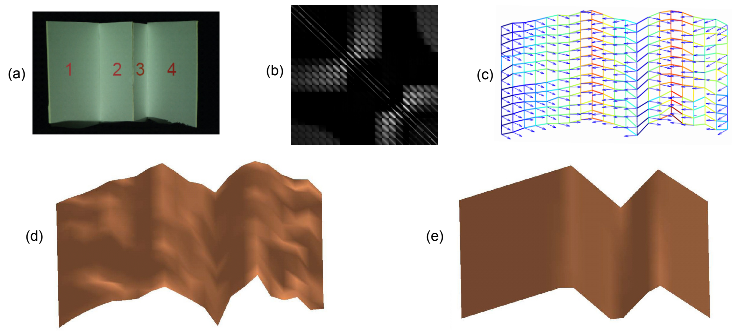

Fig. 6. (a) An image of an "M" scene. (b) The derived form factor matrix **G**. (c) and (d) The recovered surface normal (plotted in opposite direction for display purpose) and shape. Closed loop error is minimized when $C = 0.5$. (e) The recovered shape after plane fitting.

6 Conclusion and Discussions

In this paper, we present a method to estimate the scene geometry, *i.e.*, both the depth and the surface normal simultaneously, from a light transport matrix obtained with a projector-camera system. This method can handle a scene with discontinuity. We focused on extracting the geometry information from the second-bounce component that encodes scene interreflections. This method works on convex surface with strong interreflections, which often makes the conventional shape-from-intensity methods fail. Ideally, a complete algorithm for geometry estimation from a light transport matrix should also make use of the first-bounce component, which will help on convex portion of a scene and complement our method. We leave the complete algorithm to future work. Light transport is often applied for relighting applications that assume static light transport. The capability to estimate geometry will open up opportunities in fast acquisition of dynamic-scene light transport and make light transport editing possible for graphics applications. In future, we will look into more robust signal processing techniques to improve the shape reconstruction.

Limitations. One limitation of the proposed reconstruction algorithm lies in the concavity of the scene. Standalone convex surface cannot be reconstructed. However, if there exist other surfaces in the scene forming concave pairs with it, the geometry of this locally convex surface can still be recovered, *e.g.*, the inner wedge of the "M" scene can be reconstructed despite its convex nature, as the 2 inner planes interact with the outer planes to form concave pairs.

References

1. Nayar, S., Ikeuchi, K., Kanade, T.: Shape from interreflections. Proc. of Int'l Conf. on Computer Vision (1990) 2 – 11

2. Yu, Y., Debevec, P., Malik, J., Hawkins, T.: Inverse global illumination: recovering reflectance models of real scenes from photographs. In: Proc. of ACM SIGGRAPH. (1999) 215–224
3. Machida, T., Yokoya, N., Takemura, H.: Surface reflectance modeling of real objects with interreflections. In: Proc. of Int'l Conf. on Computer Vision. (2003) 170–177
4. Seitz, S.M., Matsushita, Y., Kutulakos, K.N.: A theory of inverse light transport. Proc. of Int'l Conf. on Computer Vision (2005) 1440–1447
5. S. K. Nayar, G. Krishnan, M.D.G., Raskar, R.: Fast separation of direct and global components of a scene using high frequency illumination. Proc. of ACM SIGGRAPH (2006) 935–944
6. Gupta, M., Tian, Y., Narasimhan, S.G., Zhang, L.: (de) focusing on global light transport for active scene recovery. Proc. of Computer Vision and Pattern Recognition (2009)
7. Ng, T.T., Pahwa, R.S., Tan, K.H., Bai, J., Quek, T.Q.S.: Radiometric compensation using stratified inverses. Proc. of Int'l Conf. on Computer Vision (2009)
8. J. D. Foley, A. van Dam, S.K.F., Hughes, J.F. In: Computer Graphics: Principles and Practice. (1990)
9. Goral, C. M., T.K.E.G.D.P., Battaile, B.: Modelling the interaction of light between diffuse surfaces. Computre Graphics **18** (1984) 212–222
10. Debevec, P., Hawkins, T., Tchou, C., Duiker, H.P., Sarokin, W., Sagar, M.: Acquiring the reflectance field of a human face. In: Proc. of ACM SIGGRAPH. (2000) 145–156
11. Goesele, M., Lensch, H.P., Lang, J., Fuchs, C., Seidel, H.P.: Disco: acquisition of translucent objects. In: Proc. of ACM SIGGRAPH. (2004) 835–844
12. Sen, P., Chen, B., Garg, G., Marschner, S., Horowitz, M., Levoy, M., Lensch, H.: Dual photography. Proc. of ACM SIGGRAPH (2005) 745–755
13. Peers, P., Mahajan, D.K., Lamond, B., Ghosh, A., Matusik, W., Ramamoorthi, R., Debevec, P.: Compressive light transport sensing. ACM Trans. Graph. **28** (2009) 1–18
14. Yu, Y., Debevec, P., Malik, J., Hawkins, T.: Inverse global illumination: Recovering reflectance models of real scenes from photographs. In: Proc. of ACM SIGGRAPH. (1999) 215–224
15. Boyer, K.L., Kak, A.C.: Color-coded structured light for rapid active ranging. IEEE Trans. on Pattern Analysis and Machine Intelligence **9** (1987) 14–28
16. Harley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. (2004)
17. Lambert, J.H.: Photometry, or, on the measure and gradations of light, colors, and shade: Translation from the latin of photometria, sive, de mensura et gradibus luminis, colorum et umbrae, 1760. Illuminating Eng. Soc. of North America (2001)
18. Schröder, P., Hanrahan, P.: On the form factor between two polygons. Proc. of ACM SIGGRAPH (1993) 163–164
19. Kajiya, J.T.: The rendering equation. Proc. of ACM SIGGRAPH (1986) 143–150
20. Arvo, J., Torrance, K., Smiths, B.: A framework for the analysis of error in global illumination algorithms. Proc. of ACM SIGGRAPH (1994) 75–84
21. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence **22** (2000) 888–905