Error Analysis for Image-Based Rendering with Depth Information

Ha T. Nguyen and Minh N. Do

Abstract—**We quantitatively analyze the rendering quality of image-based rendering (IBR) algorithms using per-pixel depth. Assuming the ideal pinhole camera model, we show that IBR errors can be quantified using IBR configurations such as the depth and intensity estimate errors, the scene geometry and texture, the number of actual cameras, their positions and resolution. Experiments with synthetic and real scenes show that the theoretical bounds accurately characterize the rendering errors. In particular, the proposed analysis suggests that, in smooth regions, the decay rates of the virtual images' mean absolute errors are $\mathcal{O}(\lambda^{-1})$ and $\mathcal{O}(\lambda^{-2})$, where λ is the local density of actual samples, for 2D and 3D scenes, respectively. We discuss the implications of the proposed analysis on camera placement, budget allocation, and bit allocation.**

Index Terms—**Image-Based Rendering, Plenoptic Function, Nonuniform Interpolation, Jitter, Error Bound.**

I. Introduction

Image-based rendering (IBR) applications synthesize novel (or virtual) images, as taken by virtual cameras at arbitrary viewpoints, using a set of acquired images. With a range of applications, many algorithms have been proposed for IBR [28], [31]. However, little research has been addressed the fundamental issue of analyzing the effects of different factors on the rendering quality. These factors include the number of actual cameras and their characteristics (position and resolution), and the geometry and texture of the scene. The answer to this fundamental question is crucial for both theoretical and practical purposes; we cannot effectively control the rendering quality and the cost of IBR systems without accurate quantitative analysis of the rendering quality. In fact, many IBR systems have to rely on oversampling to counter undesirable aliasing effects.

In an early approach, McMillan and Bishop [16] formalized the IBR problem as a sampling and interpolation problem of the plenoptic function. The plenoptic function was defined by Adelson and Bergen [1] to characterize the radiant energy of the scene at all positions and directions. Chai et al. [2] analyzed the spectral support of the plenoptic function for layered depth scenes and found that, under some assumptions, it was bounded only by the minimum and maximum of the depths, not by the number of layers in the scene. Zhang and Chen [30] extended the spectral analysis of the plenoptic function to cover non-Lambertian and occluded scenes. Lin and Shum [14] pioneered the geometrical approach to analyze IBR algorithms.

Despite various advantages, including analysis of lens [18], most existing spectral-based techniques assume the plenoptic functions are bandlimited, which in general is *not* the case [6], [30]. In addition, spectral-based analysis often implies uniform sampling and sinc interpolation, although sinc interpolation is never used in practice. Furthermore, uniform sampling causes aliasing noise, resulting in objectionable visual artifacts on the rendered images [4]. The geometric approach in [14] focuses on the minimum sampling rate and "double image" artifacts of IBR algorithms, but does not analyze rendering errors.

In this paper, we propose a new approach to quantitatively analyze the rendering quality for IBR algorithms with depth information. We assume the ideal pinhole camera model, though somewhat strict, to derive concrete results. The analysis is proposed in the spatial domain; thus it can quantify the rendering quality in a *local* portion of interest of the scene. We first consider the case of 2D scenes without occlusions (Section III). The case is useful for translational camera configurations or when actual images are rectified before the rendering process. The proposed methodology then is developed to analyze 2D occluded scenes (Section IV) and 3D general scenes (Section V).

The main contribution of the paper is a methodology armed with a set of techniques to analyze the rendering quality of a class of IBR algorithms with per-pixel depth information. The rendering error is bounded based on the sample values, sample positions and their errors (i.e., the sample errors and jitters). We propose innovative techniques to derive properties of sample intervals and bounds for sample jitters in order to obtain bounds for the mean absolute error (MAE) of the rendered images. The derived error bounds highlight the effects on the rendering quality of various factors including depth and intensity estimate errors, the scene geometry and texture, the number of cameras, their positions and resolution. Experiments for synthetic and real scenes show that the theoretical bounds accurately characterize the rendering errors. Based on the proposed analysis, we discuss implications for IBR-related problems such as camera placement, budget allocation, and bit allocation.

This paper is organized as follows. The problem setup is presented in Section II. We present analysis of 2D unoccluded scenes in Section III. Generalization to 2D occluded scenes and 3D scenes are given in Sections IV and V, respectively. In Section VI, we show experiments to validate the theoretical

Ha T. Nguyen was with the Department of Electrical and Computer Engineering, University of Illinois, Urbana IL 61801. He is now with the Media Processing Lab, Sony Electronics, San Jose, CA 95112 (email: thaiha.nguyen@m4x.org).

Minh N. Do is with the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Beckman Institute, University of Illinois, Urbana IL 61801 (email: minhdo@uiuc.edu).

This work was supported by the National Science Foundation under Grant ITR-0312432 and the Vietnam Education Foundation (www.vef.gov).

bounds. Finally, we offer concluding remarks and discussions in Section VII. Some preliminary results of this paper were presented in our previous conference paper [21].

II. Problem setting

We describe the scene model and the camera model in Sections II-A and II-B, respectively. In each section, the models for the 2D case and the 3D case will be presented in parallel. In Section II-C, we briefly categorize IBR algorithms, in particular ones using per-pixel depth. Finally, we formally introduce the problem definition.

A. The scene model

Consider a 2D scene as in Fig. 1(a). Its surface is modeled as a 2D parameterized curve $\boldsymbol{S}(u) : [a,b] \to \mathbb{R}^2$, for some interval $[a,b] \subset \mathbb{R}$. Each $u \in [a,b]$ corresponds to a surface point $\boldsymbol{S}(u) = [X(u), Y(u)]^T$. We denote $T(u) : [a,b] \to \mathbb{R}$ as the texture map "painted" on the surface $\boldsymbol{S}(u)$. For 3D scenes, see Fig. 1(b), the scene surface is denoted as $\boldsymbol{S}(u,v) : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$, and the texture map is $T(u,v) : \Omega \to \mathbb{R}$.

Given a parametrization of the scene geometry $\boldsymbol{S}(u)$ or $\boldsymbol{S}(u,v)$, the texture map $T(u)$ or $T(u,v)$ is independent of the cameras and the scene geometry. We assume that the surface is Lambertian [7], that is, images of the same surface point at different cameras have the same intensity. Furthermore, we assume that the surface $\boldsymbol{S}$ and the texture T have derivative of second order at all points and all directions, except at the discontinuities.

B. The camera model

The scene-to-image mapping. A *2D calibrated pinhole camera* is characterized by the projection matrix $\mathbf{\Pi} \in \mathbb{R}^{2\times 3}$. For each surface point $\boldsymbol{S} = [X,Y]^T$ in the scene, let $\widetilde{\boldsymbol{S}} = [X,Y,1]^T$ be the homogeneous coordinate [7] of $\boldsymbol{S}$. The projection matrix $\mathbf{\Pi} = [\boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T]^T$ maps surface points $\boldsymbol{S} = [X,Y]^T$ into image point x using the projection equation

$$d \cdot [x, 1]^T = \mathbf{\Pi} \cdot [X, Y, 1]^T, \tag{1}$$

where $d = \boldsymbol{\pi}_2^T \widetilde{\boldsymbol{S}}$ is the depth of a surface point $\widetilde{\boldsymbol{S}}$ relative to the camera $\mathbf{\Pi}$. We use $\mathbf{\Pi}$ to refer to the camera.

Equation (1) implies a mapping from a surface point $\boldsymbol{S}(u)$, $u \in [a,b]$, to its image point x on the camera $\mathbf{\Pi}$ as

$$x \;=\; \frac{\boldsymbol{\pi}_1^T \widetilde{\boldsymbol{S}}(u)}{\boldsymbol{\pi}_2^T \widetilde{\boldsymbol{S}}(u)} \;\overset{\text{def}}{=}\; H_{\mathbf{\Pi}}(u). \tag{2}$$

We name $H_{\mathbf{\Pi}}(u)$ the *scene-to-image mapping*. For unoccluded scenes, the mapping $H_{\mathbf{\Pi}}(u)$ is monotonic in $[a,b]$.

A *3D calibrated pinhole camera* is characterized by the positional matrix $\mathbf{\Pi} = [\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3]^T \in \mathbb{R}^{3\times 4}$. The projection equation is

$$d \cdot [x, y, 1]^T = \mathbf{\Pi} \cdot [X, Y, Z, 1]^T, \tag{3}$$

where $d = \boldsymbol{\pi}_3^T \cdot \widetilde{\boldsymbol{S}}$ is the depth of $\boldsymbol{S}$ relative to $\mathbf{\Pi}$. We derive a *scene-to-image mapping* $\boldsymbol{H}_{\mathbf{\Pi}}(u,v)$ from surface points $\boldsymbol{S}(u,v)$, for $(u,v) \in \Omega$, to their image points (x,y) as

$$\begin{bmatrix} x \\ y \end{bmatrix} \overset{\text{def}}{=} \boldsymbol{H}_{\mathbf{\Pi}}(u,v) = \begin{bmatrix} H_x(u,v) \\ H_y(u,v) \end{bmatrix}, \tag{4}$$

where

$$H_x(u,v) = \frac{\boldsymbol{\pi}_1^T \cdot \widetilde{\boldsymbol{S}}(u,v)}{\boldsymbol{\pi}_3^T \cdot \widetilde{\boldsymbol{S}}(u,v)}, \quad H_y(u,v) = \frac{\boldsymbol{\pi}_2^T \cdot \widetilde{\boldsymbol{S}}(u,v)}{\boldsymbol{\pi}_3^T \cdot \widetilde{\boldsymbol{S}}(u,v)}. \tag{5}$$

The Jacobian matrix of $\boldsymbol{H}_{\mathbf{\Pi}}$ is

$$\frac{\partial \boldsymbol{H}_{\mathbf{\Pi}}(u,v)}{\partial(u,v)} = \begin{bmatrix} \partial H_x/\partial u & \partial H_x/\partial v \\ \partial H_y/\partial u & \partial H_y/\partial v \end{bmatrix}. \tag{6}$$

The image formation process. At an image point $x = H_{\mathbf{\Pi}}(u)$ of a 2D camera $\mathbf{\Pi}$, the light field $f_{\mathbf{\Pi}}(x)$ characterizes the brightness $T(u)$ of the surface point $\boldsymbol{S}(u)$. In other words, function $f_{\mathbf{\Pi}}(x)$ is a warped version of the texture map $T(u)$:

$$f_{\mathbf{\Pi}}(x) = T(H_{\mathbf{\Pi}}^{-1}(x)). \tag{7}$$

The corresponding equation for the 3D case is

$$f_{\mathbf{\Pi}}(x,y) = T\big(\boldsymbol{H}_{\mathbf{\Pi}}^{-1}(x,y)\big). \tag{8}$$

Let Δ_x be the pixel interval in the image plane, or the *resolution*, of a 2D camera $\mathbf{\Pi}$. For the 3D case, the resolution is $\Delta_x \Delta_y$, where Δ_x and Δ_y are the sample intervals in horizontal and vertical directions, respectively. In practice, the pixel intensities are obtained as integration of the light field and the sampling kernel. For simplicity, in this paper we assume the ideal pinhole camera model, that is the Dirac function as sampling kernel, to derive concrete results. Using this model, the intensity, $I_{\mathbf{\Pi}}[n]$ or $I_{\mathbf{\Pi}}[m,n]$, is

$$\begin{aligned} I_{\mathbf{\Pi}}[n] &= f_{\mathbf{\Pi}}(n\Delta_x), && \text{2D case,} \quad (9) \\ I_{\mathbf{\Pi}}[m,n] &= f_{\mathbf{\Pi}}(m\Delta_x, n\Delta_y), && \text{3D case.} \quad (10) \end{aligned}$$

Intensity and depth estimate error. In practice, the depth can be obtained using the range finders [11], [25] or using structure-from-motion techniques [10], [26]. For the 2D case, we assume that a set of surface points $[X,Y]^T$ are available at the actual cameras. Due to depth estimation errors, the surface points are registered as $[X_e, Y_e]^T$ instead of their actual value $[X,Y]^T$. The magnitude of the error $\boldsymbol{\varepsilon}_D = [X_e - X, Y_e - Y]^T$ is supposedly bounded by some bound $E_D > 0$.

For texture errors, we assume that in the formation of actual pixels, noisy estimates $T_e(u)$ of $T(u)$ are registered. Again, the error $\varepsilon_T = T_e(u) - T(u)$ is supposedly bounded by some bound $E_T > 0$. In summary, we assume that

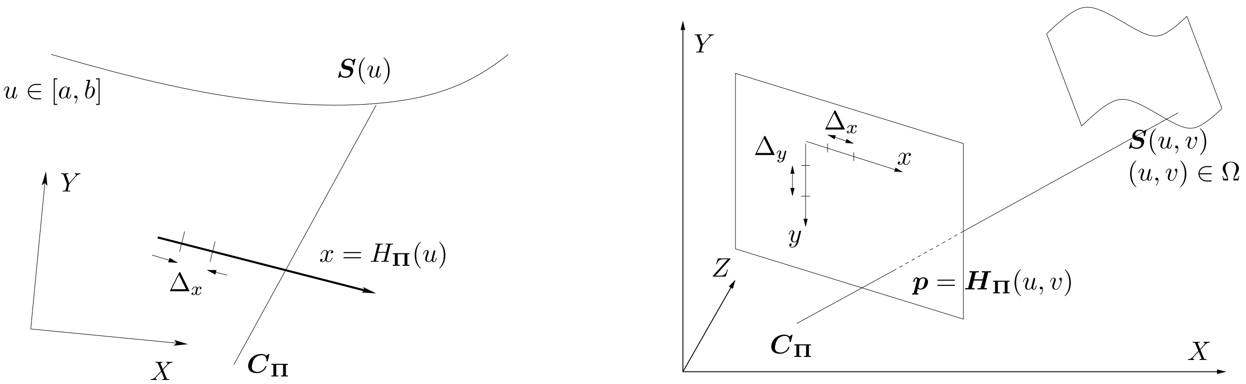
$$\|\boldsymbol{\varepsilon}_D\|_2 \le E_D, \qquad |\varepsilon_T| \le E_T. \tag{11}$$

Similar assumption as (11) can be made for the 3D case. Note that bounds E_T, E_D can be related to the signal to noise ratio of the actual images and depth maps.

C. Problem statement

IBR algorithms. Many IBR algorithms have been proposed [13], [28], [31]. This paper is concerned with IBR algorithms using per-pixel depth and image-space interpolation [15], [17], [20], [27]. We present our analysis for the Propagation Algorithm [20], although the proposed techniques are applicable to other algorithms. The Propagation Algorithm consists of three main steps as follows:

1. Information Propagation. All the intensity information available at the actual image planes is propagated to the virtual


$u \in [a,b]$
$\boldsymbol{S}(u)$
Y
$x = H_{\mathbf{\Pi}}(u)$
Δ_x
X
$\boldsymbol{C}_{\mathbf{\Pi}}$
Y
Δ_x
Δ_y
x
y
$\boldsymbol{S}(u,v)$
$(u,v) \in \Omega$
Z
$\boldsymbol{p} = \boldsymbol{H}_{\mathbf{\Pi}}(u,v)$
$\boldsymbol{C}_{\mathbf{\Pi}}$
X

(a) The 2D scene-camera model.

(b) The 3D scene-camera model.

Fig. 1. The calibrated scene-camera model. The scene surface is modeled as a 2D parameterized curve $\boldsymbol{S}(u)$ (Fig. 1(a)) or as a 3D parameterized surface $\boldsymbol{S}(u,v)$ (Fig. 1(b)). The texture map, denoted $T(u)$ for the 2D case or $T(u,v)$ for the 3D case, is "painted" on the object surface. We assume the pinhole camera model with calibrated positional matrix $\mathbf{\Pi} = [\boldsymbol{R}, \boldsymbol{T}] \in \mathbb{R}^{2\times 3}$ or $\mathbf{\Pi} \in \mathbb{R}^{3\times 4}$ for the 2D case or the 3D case, respectively. Finally, the camera resolution is characterized by the pixel interval Δ_x on the image line (2D case), or by the pixel intervals Δ_x, Δ_y in horizontal and vertical direction on the image plane (3D case).

image plane. This step is feasible since the depth information is available.

2. Occlusion Removal. All the points in whose neighborhood there is another point with sufficiently smaller depth are removed; these points are likely occluded at the virtual camera.

3. Intensity Interpolation. The virtual image is interpolated using the remaining samples. We assume that piecewise linear interpolation is used for the 2D case and Delaunay triangulation-based linear interpolation is used for the 3D case. Both methods are widely used in practice thanks to their simplicity and decent interpolation qualities. Furthermore, we hope to help IBR practitioners find the paper directly useful. We note that the proposed analysis also applies for interpolation techniques using higher order splines [5], [24], [29].

Problem statement. Suppose the virtual image at virtual camera $\mathbf{\Pi}_v$ is rendered using images and depth maps of N actual cameras $\{\mathbf{\Pi}_i\}_{i=1}^N$. We want to quantify the effects on the rendering quality of projection matrices $\{\mathbf{\Pi}_i\}_{i=1}^N$ and $\mathbf{\Pi}_v$, the resolution (Δ_x for the 2D case or $\Delta_x\Delta_y$ for the 3D case), the depth and intensity error bounds E_D, E_T, the texture map $T(u)$ or $T(u,v)$, and the surface geometry $\boldsymbol{S}(u)$ or $\boldsymbol{S}(u,v)$.

III. Analysis for 2D Unoccluded Scenes

We start by introducing the methodology in Section III-A. In Section III-B, properties of sample intervals at the virtual image plane are derived. We analyze the jitters caused by depth estimate errors in Section III-C. In Section III-D, we derive a bound for the rendering error.

A. Methodology

We present the results for linear interpolation. The linear interpolation $\widehat{f}(x)$ of $f(x)$ in the interval $[x_1, x_2]$, see Fig. 4, is defined as

$$\widehat{f}(x) \overset{\text{def}}{=} \frac{x_2 - x}{x_2 - x_1} \cdot [f(x_1 + \mu_1) + \varepsilon_1] + \frac{x - x_1}{x_2 - x_1} \cdot [f(x_2 + \mu_2) + \varepsilon_2], \tag{12}$$

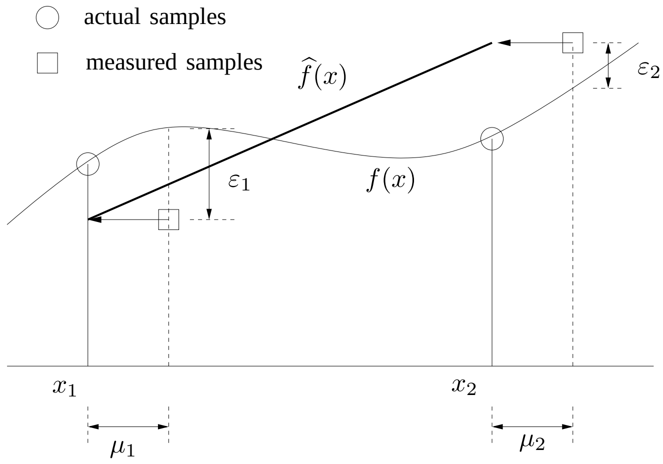


Fig. 2. Linear interpolation. The interpolation error can be bounded using sample errors $\varepsilon_1, \varepsilon_2$, sample positions x_1, x_2, and their jitters μ_1, μ_2.

where μ_1, μ_2 are sample jitters and $\varepsilon_1, \varepsilon_2$ are sample errors. The $\mathcal{L}_\infty$ norm of a function $g(x)$ is defined as

$$\|g\|_{L_\infty[x_1,x_2]} = \sup_{x\in[x_1,x_2]} |g(x)| \tag{13}$$

Proposition 1: Consider a function $f(x)$ that is twice continuously differentiable. The linear interpolation $\widehat{f}(x)$ given in (12), for $x \in [x_1, x_2]$, has the error bounded by

$$\begin{aligned} |\widehat{f}(x) - f(x)| \quad \leq \quad & \frac{1}{8}(x_2 - x_1)^2 \cdot \|f''\|_\infty + \max\{|\varepsilon_1|, |\varepsilon_2|\} \\ & + \max\{|\mu_1|, |\mu_2|\} \cdot \|f'\|_\infty. \end{aligned} \tag{14}$$

Proof: Using simple application of the Taylor expansion. For details, see [19]. ■

Remark 1: Proposition 1 can be considered as a local error analysis, providing a bound for the interpolation error in individual intervals. In the sequel, we will simply write $\|f\|_\infty$ with the understanding that the the L_∞ norm is taken in the local interval of interest.

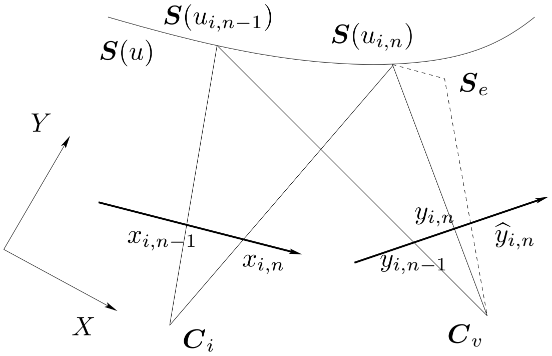


Fig. 3. Sample intervals and jitters at virtual camera $\mathbf{\Pi}_v$. Samples $\{y_{i,n}\}$ in virtual image plane are propagated from actual pixels $\{x_{i,n}\}$. The jitter $\mu = \widehat{y}_{i,n} - y_{i,n}$ is caused by a noisy estimate $\boldsymbol{S}_e$ of surface point $\boldsymbol{S}(u_n)$.

The error bound in (14) is a summation of three terms. Apart from intrinsic properties of the function $f(x)$, the first term depends on the sample intervals $(x_2 - x_1)$, the second term depends on the sample errors $\varepsilon_1, \varepsilon_2$, and the third term is related to the jitters μ_2, μ_2. Note that the bound is tight in the sense that equality does happen, for example, for linear functions $f(x)$.

Remark 2: Generalization of Proposition 1 is possible for interpolation using splines of higher orders [5]. In these cases, only the first term in (14) will change to the product of higher powers of the sample interval $|x_2 - x_1|$, the L_∞ norm of higher order derivative of $f(x)$, and a constant.

In the following, we briefly explain how Proposition 1 can be used to analyze the rendering quality. In the Information Propagation step, consider N actual cameras $\{\mathbf{\Pi}_i\}_{i=1}^N$ and a virtual camera $\mathbf{\Pi}_v$. We denote $\{x_{i,n}\}$ the set of actual pixels of $\mathbf{\Pi}_i$ and $\{u_{i,n}\}$ are such that $x_{i,n}$ is the image of 2D surface point $\boldsymbol{S}(u_{i,n})$ (see Fig. 3). Let $y_{i,n}$ be the image of $\boldsymbol{S}(u_{i,n})$ at the virtual camera $\mathbf{\Pi}_v$. These points $\{y_{i,n}\}$ will serve as samples in the interpolation process performed at virtual image plane.

The second step, the Occlusion Removal step, is irrelevant in this section because the scene is supposed to be free of occlusions. In the Intensity Interpolation step, for each actual camera $\mathbf{\Pi}_i$, for $i = 1, \ldots, N$, let $\mathcal{Y}_i = \{y_{i,n}\}$ be the set of points on the virtual image plane propagated from $\{x_{i,n}\}$. Let the union of $\{\mathcal{Y}_i\}_{i=1}^N$ be

$$\mathcal{Y} = \bigcup_{i=1}^{N} \mathcal{Y}_i = \{y_m\}_{m=1}^{N_\mathcal{Y}}, \tag{15}$$

ordered so that $y_m \le y_{m+1}$. Hence, $\mathcal{Y}$ contains all the *actual samples* propagated from the actual cameras.

The sources of rendering errors come from the intensity errors and jitters at $\{y_m\}_m$, in addition to the interpolation technique in use. We address these issues in the following sections.

B. Properties of sample intervals

In this section, we want to investigate the properties of the sample intervals $\{(y_{m+1} - y_m)\}_m$. As we see later, we are most interested in the summation $\sum_m (y_{m+1} - y_m)^k, k \in \mathbb{N}$. We analyze the sample intervals $(y_{m+1} - y_m)$ by considering each individual set $\mathcal{Y}_i$ as a point process. The set $\mathcal{Y}$ can be regarded as the union of point processes $\{\mathcal{Y}_i\}_{i=1}^N$.

Lemma 1: In each infinitesimal $[x, x + dx]$, the point process $\mathcal{Y}$, defined as in (15), can be approximated as a Poisson process with density

$$\lambda_x(x) = \frac{1}{\Delta_x H_v'(u)} \cdot \sum_{i=1}^{N} H_i'(u), \tag{16}$$

where $u = H_v^{-1}(x)$ is such that point x is the image of surface point $\boldsymbol{S}(u)$ at the virtual camera $\mathbf{\Pi}_v$.

Proof: Consider an infinitesimal interval $[x, x + dx]$ on the virtual image plane. In this interval, we can assume that $H_i'(u)$ is constant; thus, sample intervals $\{(y_{i,n+1} - y_{i,n})\}_n$ can be considered identically distributed. It is known that if the component processes $\{\mathcal{Y}_i\}_{i=1}^N$ have *identically distributed* intervals, their superposition can be approximated as a Poisson process in the distribution sense [3], [12]. Hence, locally the set of sample points $\mathcal{Y}$ can be approximated as a Poisson process.

Let $[u, u + du]$ be the portion of the scene whose image is the interval $[x, x + dx]$ at the virtual image plane. Hence $u = H_v^{-1}(x)$ and

$$dx = H_v(u + du) - H_v(u) \approx H_v'(u) du.$$

For each actual camera $\mathbf{\Pi}_i$, for $i = 1, \ldots, N$, the average number of pixels that are images of $\boldsymbol{S}(\tau)$, for $\tau \in [u, u + du]$, is $H_i'(u) du / \Delta_x$. The average number of points in $\mathcal{Y}$ hence can be computed as

$$E[N_p] = \frac{du}{\Delta_x} \sum_{i=1}^{N} H_i'(u).$$

The density $\lambda_v(x)$ hence can be obtained as

$$\lambda_v(x) = \frac{E[N_p]}{dx} = \frac{1}{\Delta_x H_v'(u)} \cdot \sum_{i=1}^{N} H_i'(u).$$

■

If the density $\lambda_x(x)$ is a constant over the whole interval $[a, b]$, the set of points $\mathcal{Y} = \{y_m\}_{m=1}^{N_\mathcal{Y}}$ can be approximated as a Poisson process. However, since $\lambda_x(x)$ changes over $[a, b]$, $\mathcal{Y}$ is approximated as a *generalized*, or *inhomogeneous*, Poisson process [23]. We use this key result to derive properties of sample intervals.

Proposition 2: The point process $\mathcal{Y}$ can be approximated as a generalized Poisson process with density function $\lambda_x(x)$ satisfying (16) for $x \in [H_v(a), H_v(b)]$. The sum of powers of the sample intervals can be computed as

$$\sum_{m=1}^{N_\mathcal{Y}-1} (y_{m+1} - y_m)^k \approx k! Y_k \Delta_x^{k-1}, \tag{17}$$

where

$$Y_k = \int_a^b \left(\sum_{i=1}^{N} H_i'(u) \right)^{1-k} (H_v'(u))^k \, du. \tag{18}$$

Proof: Using the result of Lemma 1, the point process $\mathcal{Y}$ can be approximated as a Poisson process of density $\lambda_x(x)$ in each infinitesimal interval $[x, x+dx]$. As a consequence, the average number of points $y_m \in \mathcal{Y}$ falling into $[x, x+dx]$ is $\lambda_x(x)dx$, and the expectation of intervals $E[(y_{m+1}-y_m)^k]$ inside $[x, x+dx]$ is equal to $k!/\lambda_x(x)^k$. Hence:

$$\sum_{n=1}^{N_{\mathcal{Y}}-1} (y_{m+1}-y_m)^k \approx \int_{H_v(a)}^{H_v(b)} \frac{k!}{\lambda_x(x)^k}\lambda_x(x)dx$$

By changing the variable under the integral from x to u, using $dx = H_v'(u)du$, we indeed obtain the desired result. ■

Note that Y_k, called multiple-view term of order k, for $k \in \mathbb{N}$, depends only on the relative positions of the (actual and virtual) cameras and the scene. In unoccluded scenes, the derivative $H_i'(u)$ has positive infimum and finite supremum. As a consequence:

$$0 < \lim_{N\to\infty} Y_k N^{k-1} < \infty. \tag{19}$$

Hence, $Y_k = \mathcal{O}(N^{1-k})$. In particular, $Y_1 = H_v(b) - H_v(a)$ is a constant equal to the length of the image of $\boldsymbol{S}(u)$, for $u \in [a,b]$, in the virtual image plane.

C. Bound for sample jitters

Another source of IBR error is the jitters caused by noisy depth estimates. Let $\boldsymbol{S} = [X, Y]^T$ be a surface point, and y be the image of $\boldsymbol{S}$ at the virtual camera Π_v. We denote $\boldsymbol{S}_e = [X_e, Y_e]^T$ a noisy estimate of $\boldsymbol{S}$ with reconstruction error $\boldsymbol{\varepsilon}_D = \boldsymbol{S}_e - \boldsymbol{S}$, and $\widehat{y}$ to be the image of $\boldsymbol{S}_e$ at $\boldsymbol{\Pi}_v$ (see Fig. 3). In this section, we derive a bound for the sample jitters $\mu = \widehat{y} - y$.

Proposition 3: The jitter $\mu = \widehat{y} - y$ at virtual camera $\boldsymbol{\Pi}_v$ caused by the depth estimate error $\boldsymbol{\varepsilon}_D$ is bounded by

$$|\mu| \le E_D B_v. \tag{20}$$

In the above inequality, B_v is determined as follows:

$$B_v = \sup_{u\in[a,b], i=1,\dots,N} \left\{ \frac{\|\boldsymbol{C}_v - \boldsymbol{C}_i\|_2}{d(u)^2} \right\}. \tag{21}$$

Proof: Suppose the jitter μ is propagated from actual camera $\boldsymbol{\Pi}_i$, for some $i = 1, \dots, N$. Let $\boldsymbol{\varepsilon} = [\varepsilon_X, \varepsilon_Y, 0]^T$ and $\boldsymbol{p} = [p_X, p_Y, 0]^T = \widetilde{\boldsymbol{S}}(u) - \widetilde{\boldsymbol{C}}_v$. We can easily verify

$$\begin{aligned} \widetilde{\boldsymbol{S}}_e &= \widetilde{\boldsymbol{C}}_v + \boldsymbol{p} + \boldsymbol{\varepsilon}, \\ \widetilde{\boldsymbol{S}} &= \widetilde{\boldsymbol{C}}_v + \boldsymbol{p}, \\ \boldsymbol{\pi}_l^T \widetilde{\boldsymbol{C}}_v &= 0, \quad l = 1, 2. \end{aligned}$$

Denote $\boldsymbol{\Pi}_v = [\boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T]^T$. Using the above equalities and simple manipulations we get

$$\mu = \frac{\boldsymbol{\pi}_1^T \widetilde{\boldsymbol{S}}_e}{\boldsymbol{\pi}_2^T \widetilde{\boldsymbol{S}}_e} - \frac{\boldsymbol{\pi}_1^T \widetilde{\boldsymbol{S}}}{\boldsymbol{\pi}_2^T \widetilde{\boldsymbol{S}}} = \frac{\boldsymbol{p}^T(\boldsymbol{\pi}_2\boldsymbol{\pi}_1^T - \boldsymbol{\pi}_1\boldsymbol{\pi}_2^T)\boldsymbol{\varepsilon}}{(\pi_2^T \widetilde{\boldsymbol{S}}_e)\cdot(\pi_2^T \widetilde{\boldsymbol{S}})}. \tag{22}$$

Note that $\boldsymbol{S}_e$ always lies in the ray connecting $\boldsymbol{C}_i$ and $\boldsymbol{S}$, or $\widetilde{\boldsymbol{S}} - \widetilde{\boldsymbol{C}}_i = \alpha \cdot \boldsymbol{\varepsilon}$ for some real number α. Hence, $\boldsymbol{p} = \alpha \cdot \boldsymbol{\varepsilon} + \widetilde{\boldsymbol{C}}_i - \widetilde{\boldsymbol{C}}_v$. It can be shown, for all $\boldsymbol{\varepsilon}$, that

$$\boldsymbol{\varepsilon}^T(\boldsymbol{\pi}_2\boldsymbol{\pi}_1^T - \boldsymbol{\pi}_1\boldsymbol{\pi}_2^T)\boldsymbol{\varepsilon} = 0. \tag{23}$$

Since both $\boldsymbol{\varepsilon}, \boldsymbol{p}$ have the third coordinate equal to 0, only the upper-left 2×2 block of matrix $(\boldsymbol{\pi}_2\boldsymbol{\pi}_1^T - \boldsymbol{\pi}_1\boldsymbol{\pi}_2^T)$ needs to be investigated. If we let $\boldsymbol{R}_v$ be the rotation matrix of $\boldsymbol{\Pi}_v$, the maximum singular value of the upper-left 2×2 block of matrix $(\boldsymbol{\pi}_2\boldsymbol{\pi}_1^T - \boldsymbol{\pi}_1\boldsymbol{\pi}_2^T)$ is in fact $\det(\boldsymbol{R}_v) = 1$. Hence, using (22) and (23) we obtain

$$\begin{aligned} \mu &= \frac{(\widetilde{\boldsymbol{C}}_i^T - \widetilde{\boldsymbol{C}}_v^T)(\boldsymbol{\pi}_2\boldsymbol{\pi}_1^T - \boldsymbol{\pi}_1\boldsymbol{\pi}_2^T)\boldsymbol{\varepsilon}}{(\pi_2^T \widetilde{\boldsymbol{S}}_e)\cdot(\pi_2^T \widetilde{\boldsymbol{S}})} \\ &\le \frac{\|\widetilde{\boldsymbol{C}}_i - \widetilde{\boldsymbol{C}}_v\|_2 \cdot \|\boldsymbol{\varepsilon}\|_2}{d(u)^2}. \end{aligned}$$

The last inequality implies (20). ■

The bound $E_D B_v$ depends on the depth estimate error E_D and the relative positions between the virtual camera and the actual cameras and the scene defined by B_v.

D. Error analysis

Combining the results of Propositions 2 and 3, we derive in this section an error bound for the *mean absolute error* (MAE) of the virtual image. Note that, in practice, smaller MAE does not necessarily imply better visual quality. In addition, the derivation for other metrics such as mean square error (MSE) is also possible, although more involved.

Let $e(x) = \widehat{f}_v(x) - f_v(x)$ be the interpolation error and N_{pixel} be the number of virtual pixels being images of surface points $\boldsymbol{S}(u)$ for $u \in [a,b]$. The mean absolute error MAE is defined as

$$\text{MAE} = \frac{1}{N_{\text{pixel}}} \sum_{n=1}^{N_{\text{pixel}}} |e(n\Delta_x)|. \tag{24}$$

Theorem 1: The mean absolute error MAE of the virtual image is bounded by

$$\text{MAE} \le \frac{3Y_3}{4Y_1}\Delta_x^2 \|f_v''\|_\infty + E_T + E_D B_v \|f_v'\|_\infty, \tag{25}$$

where Y_1, Y_3 are defined as in (18), B_v is as in (21), and E_D, E_T are as in (11).

Proof: Note that the n-th virtual pixel has position $x_n = n\Delta_x$ in the virtual image plane, hence MAE can be approximated as

$$\text{MAE} \approx \frac{1}{H_v(b) - H_v(a)} \int_{H_v(a)}^{H_v(b)} |e(x)|dx. \tag{26}$$

We break down the integral above into intervals $[y_m, y_{m+1})$ and apply Proposition 1 to each interval to get:

$$\begin{aligned} \int_{H_v(a)}^{H_v(b)} |e(x)|dx &= \sum_{n=1}^{N_{\mathcal{Y}}-1} \int_{y_m}^{y_{m+1}} |e(x)|dx \\ &\le \sum_{n=1}^{N_{\mathcal{Y}}-1} \Big[\frac{1}{8}(y_{m+1}-y_m)^3 \|f_v''\|_\infty \\ &\qquad + (y_{m+1}-y_m)\big(E_T + E_D B_v \|f_v'\|_\infty\big) \Big] \\ &= \frac{3}{4} Y_3 \Delta_x^2 \|f_v''\|_\infty + Y_1 \left(E_T + E_D B_v \|f_v'\|_\infty\right). \end{aligned}$$

In the last inequality, $\sum (y_{m+1}-y_m)^k$ are replaced by $k! Y_k \Delta_x^{k-1}$ using (18). Substituting the above bound of the integral $\int |e(x)|dx$ into (26), we indeed obtain (25). ■

E. Discussion

We first emphasize the power of the geometric and local analysis resulted in Theorem 1. The bound can be applied locally and give meaningful bounds in most parts of the scene where f is smooth and thus $\|f_v'\|$ and $\|f_v''\|$ are is small in that regions. Regions with sharp edges and occlusions can be isolated.

The error bound in (25) consists of three terms. In the first term, $\|f_v''\|_\infty$ and Y_1 are fixed given the position of the virtual camera, whereas Y_3 depends on the camera configuration and the scene. We can consider Y_3 as the spatial information contributed by the actual cameras. Overall, the first term characterizes the gain of using multiple actual cameras.

To decrease the first term in an IBR setting, we can either use actual cameras of finer resolution Δ_x or increase the number of actual cameras N. Theorem 1 indicates that both methods yield comparable effects on the rendering quality. Moreover, the error bound decays as $\mathcal{O}(\lambda^{-2})$, where

$$\lambda = N/\Delta_x \tag{27}$$

can be interpreted as the local density of actual samples.

The second term, E_T, characterizes the noise level at actual cameras. This can be considered as the limit of the rendering quality imposed by the quality of actual images.

The third term contains the precision E_D of range finders and two factors $\|f_v'\|_\infty, B_v$ that depend on the relative position between the virtual camera and the actual cameras and the scene. Among these factors, E_D can be reduced by using better range finders, and $\|f_v'\|_\infty$ is fixed once the virtual camera is specified. The last factor, B_v, depends in part on the relative positions between the virtual camera and the actual cameras. In practice, when the depth error E_D is large, we can reduce B_v by limiting the set of used actual cameras to only nearby cameras. This is intuitive and conforms with [2]. However, by limiting the number of actual cameras, the density of actual samples, i.e. λ, will also be reduced, which in turn affects the first term in (25). In this case, Theorem 1 could guide us on what is the best way by trading the third term with the first term in the error bound.

IV. Analysis for 2D occluded scenes

In this section, we consider 2D occluded scenes by introducing two adjustments to the analysis in Section III. First, the presence of occlusions requires modification of the sample density $\lambda_x(x)$. Second, intervals containing discontinuities of the virtual image, either caused by the intensity or depth discontinuities, need to be analyzed using the new methodology of Proposition 4. For simplicity, we assume that all the occluded samples are successfully removed, and the set of remaining samples are dense enough so that there exists at most one discontinuity in each sample interval. General analysis is possible, though less elegant, and produces similar conclusions.

A. Methodology

We extend the Proposition 1 to consider discontinuous functions. In the following, we bound the linear interpolation error for functions with only one discontinuity. First, for simplicity we introduced notations:

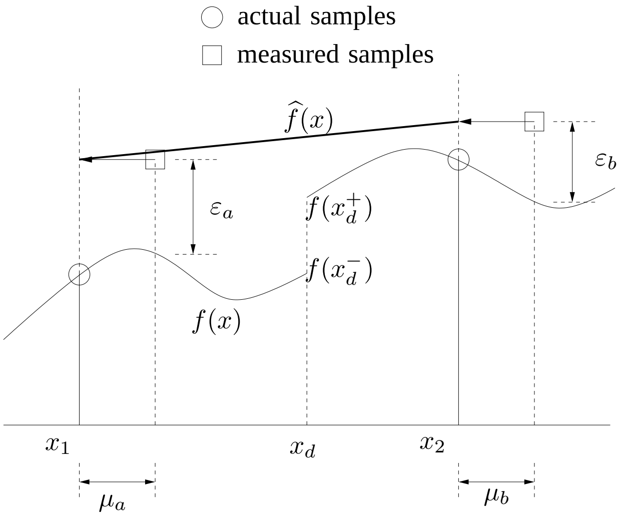


Fig. 4. Linear interpolation of a discontinuous function. Compared to its corresponding bound of the interpolation error in Section IV-A, the error bound for discontinuous function increases by an amount depending on the position of the discontinuity and the jump of the function and its derivative at the discontinuity.

$$\Delta_1 = x_d - x_1, \quad \Delta_2 = x_2 - x_d, \quad \Delta = x_2 - x_1. \tag{28}$$

Proposition 4: Consider a function $f(x)$ that is twice continuously differentiable except at *the* discontinuity x_d. The aggregated error over $[x_1, x_2]$ of the linear interpolation can be bounded by

$$\begin{aligned}\int_{x_1}^{x_2} |\widehat{f}(x) - f(x)| \quad \leq \quad & \frac{1}{8}\Delta^3 \cdot \|f''\|_\infty + \frac{1}{2}\Delta_1\Delta_2 \cdot |J_1| + \frac{3}{2}\Delta \cdot |J_0| \\ & + \Delta\Big(\max_{i=1,2}\{|\varepsilon_i|\} + \max_{i=1,2}\{|\mu_i|\} \cdot \|f'\|_\infty \Big) \end{aligned} \tag{29}$$

where J_0, J_1 are the jumps of $f(x)$ and its derivative at the discontinuity x_d:

$$J_0 = f(x_d^+) - f(x_d^-), \quad J_1 = f'(x_d^+) - f'(x_d^-). \tag{30}$$

Proof: See Appendix A. ■

Remark 3: The bound in Proposition 4 is proposed for the aggregated error over the interval $[x_1, x_2]$. This is different from the pointwise bound given in Proposition 1. Proposition 4 can be considered as a local analysis, providing a bound for the interpolation error in individual intervals. Because of the discontinuity at x_d, the aggregated error increases by an amount of

$$\frac{1}{2}\Delta_1\Delta_2 \cdot |J_1| + \frac{3}{2}\Delta \cdot |J_0|. \tag{31}$$

If $J_0 = J_1 = 0$, the bound of Proposition 4 simplifies to the case where $f(x)$ is twice continuously differentiable in $[x_1, x_2]$. The bound is characterized by sample intervals, sample errors, and jitters, in addition to intrinsic properties of $f(x)$. Similar remarks can be drawn for interpolation using splines of higher orders [5].

The bound in (29) suggests that we need to investigate the sample intervals, especially observed sample intervals around the discontinuities, and sample jitters in the context of IBR.

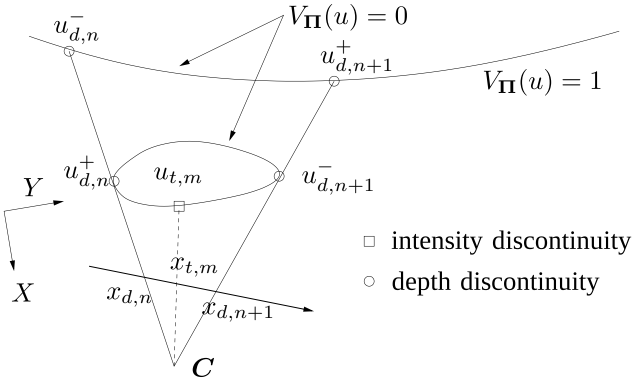


Fig. 5. A 2D occluded scene. We differentiate two kinds of discontinuities: those due to occlusions (such as $x_{d,n}$ with parameters $u^+_{d,n}$ and $u^-_{d,n}$) and those due to the texture $T(u)$ (such as $x_{t,n}$ with parameter $u_{t,m}$).

Modification of the sample density. Consider a 2D occluded scene (see Fig. 5). For a camera $\mathbf{\Pi}$, we define the visibility function

$$V_{\mathbf{\Pi}}(u) = \begin{cases} 1, & \text{if } \boldsymbol{S}(u) \textit{ is} \text{ visible at } \mathbf{\Pi} \\ 0, & \text{if } \boldsymbol{S}(u) \textit{ is not} \text{ visible at } \mathbf{\Pi}. \end{cases} \quad (32)$$

Proposition 2 is modified as

$$\lambda_x(x) = \frac{1}{\Delta_x H'_v(u)} \cdot \sum_{i=1}^{N} V_i(u) H'_i(u), \quad (33)$$

where u is the parameter of the surface point $\boldsymbol{S}(u)$ having image at x:

$$u = \arg\min_u \{d(u) : H_{\mathbf{\Pi}}(u) = x\}. \quad (34)$$

For this modification, Y_k will also be changed to

$$Y_k = \int_a^b \left(\sum_{i=1}^{N} V_i(u) H'_i(u) \right)^{1-k} \left(V_v(u) H'_v(u) \right)^k du. \quad (35)$$

Intuitively, the modification in (35) signifies that, if a surface point $\boldsymbol{S}(u)$ is occluded at an actual camera $\mathbf{\Pi}_i$, or equivalently $V_i(u) = 0$, this camera $\mathbf{\Pi}_i$ contributes no information to the rendering of virtual pixel $x = H_v(u)$. Similarly, if $\boldsymbol{S}(u)$ is occluded at the virtual camera $\mathbf{\Pi}_v$, or equivalently $V_v(u) = 0$, no information from actual cameras is necessary.

Incorporation of jumps. We differentiate two categories of discontinuities at the virtual image $f_v(x)$, namely, the depth discontinuities and the texture discontinuities (see Fig. 5). The *depth* discontinuities are at image object boundaries (backgrounds and foregrounds). Let $\mathcal{X}_d$ be the set of depth discontinuities. For each point $x_{d,n} \in \mathcal{X}_d$, denote

$$u^+_{d,n} = \lim_{x \to x^+_{d,n}} H_v^{-1}(x), \quad (36)$$

$$u^-_{d,n} = \lim_{x \to x^-_{d,n}} H_v^{-1}(x). \quad (37)$$

The above equations are well defined since $H_v^{-1}(x)$ is a one-to-one mapping everywhere except at discontinuities of $f_v(x)$. Intuitively, $u^+_{d,n}$ is the parameter on the background and $u^-_{d,n}$ is the parameter on the foreground, or vice-versa.

The *texture* discontinuities are discontinuities of the texture $T(u)$. We denote the set of texture discontinuities $\mathcal{X}_t$. For consistency, we also use notation $u^+_{t,n}$ and $u^-_{t,n}$, as in (36) and (37), for $x_{t,n} \in \mathcal{X}_t$, though they are in fact equal.

For each discontinuity

$$x_n \in \mathcal{X} = \mathcal{X}_t \bigcup \mathcal{X}_d, \quad (38)$$

the interval $[y_{m_n}, y_{m_n+1}]$ containing x_n is called an *observed interval* (or sampled interval). The following lemma is a classical result of Poisson processes.

Lemma 2: [8], [23] Let $(y_{m_n+1} - y_{m_n})$ be the observed interval around each discontinuity x_n. The length of intervals $\Delta_{2,n} = y_{m_n+1} - x_n$ and $\Delta_{1,n} = x_n - y_{m_n}$ are independent and follow exponential distributions of parameter $\lambda(H_v(u_n^+))$ and $\lambda(H_v(u_n^-))$, respectively.

Corollary 1: The following equations hold:

$$E[y_{m_n+1} - y_{m_n}] = \frac{1}{\lambda(H_v(u_n^+))} + \frac{1}{\lambda(H_v(u_n^-))} \quad (39)$$

$$E[\Delta_{1,n}\Delta_{2,n}] = \frac{1}{\lambda(H_v(u_n^+))} \cdot \frac{1}{\lambda(H_v(u_n^-))}. \quad (40)$$

We define operators $\mathcal{J}_0(f)$ and $\mathcal{J}_1(f)$ as

$$\mathcal{J}_0(f) = \sup_x \left\{ \left| \lim_{y \to x^+} f(y) - \lim_{y \to x^-} f(y) \right| \right\} \quad (41)$$

$$\mathcal{J}_1(f) = \sup_x \left\{ \left| \lim_{y \to x^+} f'(y) - \lim_{y \to x^-} f'(y) \right| \right\}. \quad (42)$$

Next, we propose a similar result of the Theorem 1 for 2D occluded scenes.

Theorem 2: The mean absolute error MAE of the virtual image using the Propagation Algorithm is bounded by

$$\begin{aligned} \text{MAE} \leq\ & \frac{3Y_3}{4Y_1} \Delta_x^2 \|f''_v\|_\infty + E_T + E_D B_v \|f'_v\|_\infty \\ & + \frac{3}{2} D_0 \Delta_x \mathcal{J}_0(f_v) + \frac{1}{2} D_1 \Delta_x^2 \mathcal{J}_1(f_v), \end{aligned} \quad (43)$$

where $f_v(x)$ is the virtual image, B_v is as in (21), Y_k is defined in (35), $\mathcal{J}_0(f)$ and $\mathcal{J}_1(f)$ are defined in (41) and (42), and D_0, D_1 are

$$D_0 = \sum_{x_d \in \mathcal{X}} \frac{1}{\lambda_x(x_d^+)} + \frac{1}{\lambda_x(x_d^-)} \quad (44)$$

$$D_1 = \sum_{x_d \in \mathcal{X}} \frac{1}{\lambda_x(x_d^+)} \cdot \frac{1}{\lambda_x(x_d^-)}. \quad (45)$$

Proof: The proof is similar to the proof of Theorem 1; we need to consider in addition the aggregated error in observed intervals $[y_{m_n}, y_{m_n+1}]$ around jumps $\{x_n \in \mathcal{X}\}$. Hence, the error bound needs to increase by an amount

$$\frac{3}{2} |y_{m_n+1} - y_{m_n}| \cdot \mathcal{J}_0(f) + \frac{1}{2} \Delta_{1,n}\Delta_{2,n} \cdot \mathcal{J}_1(f). \quad (46)$$

The summation these terms, for all x_n, in fact results in the additional fourth and fifth terms. ■

Remark 4: Compared to Theorem 1, the bound in (43) has additional fourth and fifth terms to incorporate the discontinuities of the virtual image $f_v(x)$. Overall, the fourth term decays as $\mathcal{O}(\lambda^{-1})$ and the fifth term decays as $\mathcal{O}(\lambda^{-2})$, where λ is the local density of actual samples. Note further that, around discontinuities, the Poisson approximation might be less accurate since fewer number of actual cameras contribute information to the rendering process.

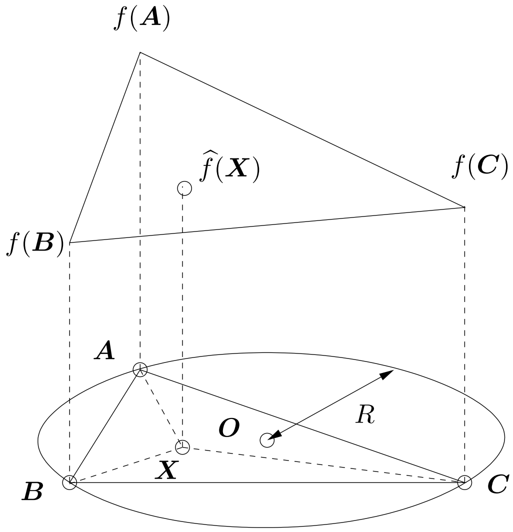


Fig. 6. Triangulation-based linear interpolation is often used with the Delaunay triangulation. In each triangle, the interpolation error can be bounded, as in Proposition 5, using the triangle's circumcircle radius R, the sample errors ε, and the sample jitters μ.

V. ANALYSIS FOR 3D SCENES

In this section, we extend the analysis into the 3D case. A natural generalization of piecewise linear interpolation into 2D is the Delaunay triangulation-based linear interpolation. We present the 3D methodology for individual triangles in Section V-A. Then, we show properties of Poisson Delaunay triangles in Section V-B and a bound for sample jitters in Section V-C. Finally, an error analysis for 3D scenes without occlusions is given in Section V-D.

A. Methodology

In this section, we investigate the interpolation error for an individual triangle. We define the $\mathcal{L}_\infty$ norm of the gradient $\nabla f(x,y)$ and the Hessian matrix $\nabla^2 f(x,y)$ as follows:

$$\|\nabla f(x,y)\|_\infty = \sup_{(x,y)} \left\{\|\nabla f(x,y)\|_2\right\} \qquad (47)$$

$$\|\nabla^2 f(x,y)\|_\infty = \sup_{(x,y)} \left\{\sigma_{\max}\left[\nabla^2 f(x,y)\right]\right\}, \qquad (48)$$

where $\sigma_{\max}[\boldsymbol{M}]$ denotes the maximum singular value [9] of a matrix $\boldsymbol{M}$. The linearly interpolated value at a 2D point $\boldsymbol{X}$ inside a triangle $\boldsymbol{ABC}$ is defined as

$$\widehat{f}(\boldsymbol{X}) = \frac{S_{\boldsymbol{A}}}{S} f(\boldsymbol{A}) + \frac{S_{\boldsymbol{B}}}{S} f(\boldsymbol{B}) + \frac{S_{\boldsymbol{C}}}{S} f(\boldsymbol{C}), \qquad (49)$$

where $S_{\boldsymbol{A}}, S_{\boldsymbol{B}}, S_{\boldsymbol{C}}$, and S denote the area of triangles $\boldsymbol{XBC}, \boldsymbol{AXC}, \boldsymbol{ABX}$, and $\boldsymbol{ABC}$, respectively. In other words, $\widehat{f}(\boldsymbol{X})$ is a bivariate linear function that is equal to $f(\boldsymbol{X})$ at locations $\boldsymbol{A}, \boldsymbol{B}$, and $\boldsymbol{C}$ (see Fig. 6). In the presence of sample errors and jitters, sample values $f(\boldsymbol{A}), f(\boldsymbol{B})$, and $f(\boldsymbol{C})$ in (49) are replaced by $f(\boldsymbol{A}+\boldsymbol{\mu_A})+\varepsilon_{\boldsymbol{A}}, f(\boldsymbol{B}+\boldsymbol{\mu_B})+\varepsilon_{\boldsymbol{B}}$, and $f(\boldsymbol{C}+\boldsymbol{\mu_C})+\varepsilon_{\boldsymbol{C}}$, respectively.

Proposition 5: We consider a function $f(x,y)$ that is twice continuously differentiable. The linear interpolation on a triangle given in (49) has the error bounded by

$$|\widehat{f}(x,y) - f(x,y)| \le \frac{1}{2}R^2 \cdot \|\nabla^2 f\|_\infty + \max\{|\varepsilon|\} + \max\{\|\boldsymbol{\mu}\|_2\} \cdot \|\nabla f\|_\infty, \qquad (50)$$

where R is the circumcircle radius of the triangle $\boldsymbol{ABC}$.

Proof: We show the proof for $\varepsilon = 0$ and $\boldsymbol{\mu} = \boldsymbol{0}$. In this case, the error bound in the right-hand side of (50) reduces into the first term. The techniques to incorporate the sample error (second term) and jitter (third term) are similar to the proof of Proposition 1.

Let $\boldsymbol{O}$ be the center of the circumcircle of triangle $\boldsymbol{ABC}$. Using vector manipulations, it can be shown that

$$R^2 - \|\boldsymbol{X}-\boldsymbol{O}\|_2^2 = \frac{S_{\boldsymbol{A}}}{S}\|\Delta_{\boldsymbol{A}}\|_2^2 + \frac{S_{\boldsymbol{B}}}{S}\|\Delta_{\boldsymbol{B}}\|_2^2 + \frac{S_{\boldsymbol{C}}}{S}\|\Delta_{\boldsymbol{C}}\|_2^2, \qquad (51)$$

where $\Delta_{\boldsymbol{A}} = \boldsymbol{A}-\boldsymbol{X}, \Delta_{\boldsymbol{B}} = \boldsymbol{B}-\boldsymbol{X}$, and $\Delta_{\boldsymbol{C}} = \boldsymbol{C}-\boldsymbol{X}$. Using the 2D Taylor expansion we can obtain

$$f(\boldsymbol{A}) = f(\boldsymbol{X}) + \nabla f(\boldsymbol{X})^T \cdot \Delta_{\boldsymbol{A}} + \frac{1}{2}\Delta_{\boldsymbol{A}}^T \cdot \nabla^2 f(\boldsymbol{X}_a) \cdot \Delta_{\boldsymbol{A}}$$

for some point $\boldsymbol{X}_a$. Similar equations can be obtained for $\boldsymbol{B}$ and $\boldsymbol{C}$ as well. Hence,

$$\begin{aligned}
|\widehat{f}(\boldsymbol{X}) - f(\boldsymbol{X})| &= \frac{1}{2} \cdot \Big| \frac{S_{\boldsymbol{A}}}{S}\Delta_{\boldsymbol{A}}^T \cdot \nabla^2 f(\boldsymbol{X}_a) \cdot \Delta_{\boldsymbol{A}} + \\
&\quad \frac{S_{\boldsymbol{B}}}{S}\Delta_{\boldsymbol{B}}^T \cdot \nabla^2 f(\boldsymbol{X}_b) \cdot \Delta_{\boldsymbol{B}} + \frac{S_{\boldsymbol{C}}}{S}\Delta_{\boldsymbol{C}}^T \cdot \nabla^2 f(\boldsymbol{X}_c) \cdot \Delta_{\boldsymbol{C}} \Big| \\
&\le \frac{1}{2}\|\nabla^2 f\|_\infty \Big(\frac{S_{\boldsymbol{A}}}{S}\|\Delta_{\boldsymbol{A}}\|_2^2 + \frac{S_{\boldsymbol{B}}}{S}\|\Delta_{\boldsymbol{B}}\|_2^2 + \frac{S_{\boldsymbol{C}}}{S}\|\Delta_{\boldsymbol{C}}\|_2^2 \Big) \\
&\le \frac{1}{2}\|\nabla^2 f\|_\infty R^2.
\end{aligned}$$

■

The bound in (50) suggests that we need to investigate the properties of Delaunay triangles and the sample jitters. The next two sections will present these properties.

B. Properties of Poisson Delaunay triangles

We assume in this section that the scene is unoccluded. We start by proposing an equivalence of Lemma 1 for the 2D case. A 2D process p is called *identically distributed scattering* [22] if the density of points of p over an arbitrary region ω follows a fixed probability mass distribution independent of ω. Intuitively, there is a profound similarity between the 1D and 2D cases, since they are both related to the probability mass function (pmf) of the number of points falling inside an arbitrary region. Hence, in the following, we assume that the Hypothesis 1 below holds.

Hypothesis 1: The superposition of 2D point processes with identically distributed scattering property can be approximated as a 2D Poisson process.

On the image plane of the virtual camera $\boldsymbol{\Pi}_v$, let $\mathcal{Y}$ be the set of points *propagated* from actual pixels [20].

Proposition 6: The point process $\mathcal{Y}$ can be approximated as a 2D generalized Poisson process with density function

$$\lambda(x,y) = \frac{1}{\Delta_x \Delta_y} \sum_{i=1}^{N} \frac{\det\Big(\partial \boldsymbol{H}_i / \partial(u,v)\Big)}{\det\Big(\partial \boldsymbol{H}_v / \partial(u,v)\Big)}, \qquad (52)$$

where $(u,v) = \boldsymbol{H}_v^{-1}(x,y)$.

Proof: Since we assume that Hypothesis 1 holds, in each infinitesimal region, the point process $\mathcal{Y}$ can be considered as a 2D Poisson process. Hence, overall, $\mathcal{Y}$ can be considered as a generalized Poisson process. The density $\lambda(x,y)$ can be computed, similarly to Lemma 1, as the average number of points falling on an unit area. This indeed results in (52). ■

Once we approximate the set of propagated points $\mathcal{Y}$ as a 2D Poisson process, the next step is to investigate properties of Poisson Delaunay triangles. In the following, we exploit results from stochastic geometry.

Lemma 3: [22, Chapter 5] The circumradius R and the area S of Delaunay triangles of a 2D Poisson process of density λ are *independent*. The circumradius R has the probability density function (pdf)

$$2(\pi\lambda)^2 r^3 e^{-\pi\lambda r^2}, \quad r > 0. \tag{53}$$

The moments $E[S^k]$ can be computed using explicit formula. In particular

$$E[R^2] = \frac{2}{\pi\lambda}, \quad E[S] = \frac{1}{2\lambda}, \quad E[S^2] = \frac{35}{8\pi^2\lambda^2}. \tag{54}$$

C. Bound for sample jitters

Let $\boldsymbol{S} = [X,Y,Z]^T$ be a surface point, and $\boldsymbol{p}$ be the image of $\boldsymbol{S}$ at the virtual camera $\boldsymbol{\Pi}_v$. We denote $\boldsymbol{S}_e = [X_e, Y_e, Z_e]^T$ a noisy estimate of $\boldsymbol{S}$ with reconstruction error $\boldsymbol{\varepsilon}_D = \boldsymbol{S}_e - \boldsymbol{S}$, and $\widehat{\boldsymbol{p}}$ the image of $\boldsymbol{S}_e$ at $\boldsymbol{\Pi}_v$.

Proposition 7: The jitter $\boldsymbol{\mu} = \widehat{\boldsymbol{p}} - \boldsymbol{p}$, at virtual camera $\boldsymbol{\Pi}_v$ with camera center $\boldsymbol{C}_v$, caused by depth estimate errors can be bounded by

$$\|\boldsymbol{\mu}\|_2 \leq \sqrt{2} E_D B_v. \tag{55}$$

In the above inequality, B_v is computed as

$$B_v = \sup_{(u,v)\in\Omega, i=1,\dots,N} \left\{ \frac{\|\boldsymbol{C}_v - \boldsymbol{C}_i\|_2}{d(u,v)^2} \right\}. \tag{56}$$

Proof: The jitter $\boldsymbol{\mu}$ is a two-coordinate vector $\boldsymbol{\mu} = [\mu_x, \mu_y]^T$. Using Proposition 3, the norms of both μ_x and μ_y can be bounded by $E_D B_v$. Hence,

$$\|\boldsymbol{\mu}\|_2 = \sqrt{\mu_x^2 + \mu_y^2} \leq \sqrt{2} E_D B_v.$$

■

The bound $E_D B_v$ depends on the depth estimate error E_D and the relative position between the virtual camera and the actual cameras and the scene defined by B_v.

D. Analysis for 3D unoccluded scenes

Consider the intensity function $f_v(x,y) = T\big(\boldsymbol{H}_v^{-1}(x,y)\big)$ at virtual camera $\boldsymbol{\Pi}_v$. Let $e(x,y) = \widehat{f}_v(x,y) - f_v(x,y)$ be the interpolation error and $\mathcal{N}_\Omega$ be the set of virtual pixels (m,n) being images of surface points $\boldsymbol{S}(u,v)$ for $(u,v) \in \Omega$. Denote $\#\mathcal{N}_\Omega$ the number of pixels in $\mathcal{N}_\Omega$. The *mean absolute error* MAE is defined as

$$\mathrm{MAE} = \frac{1}{\#\mathcal{N}_\Omega} \sum_{(m,n)\in\mathcal{N}_\Omega} |e(m\Delta_x, n\Delta_y)|. \tag{57}$$

Theorem 3: The *mean absolute error* MAE of the virtual image using the Propagation Algorithm is bounded by

$$\mathrm{MAE} \leq \frac{X_2}{\pi X_1}\Delta_x\Delta_y \|\nabla^2 f_v\|_\infty + E_T + \sqrt{2}E_D B_v \|\nabla f_v\|_\infty, \tag{58}$$

where f_v is the the virtual image, B_v is as in (56), E_D, E_T are as in (11), and

$$X_k = \int_\Omega \left(\sum_{i=1}^N \det\left(\frac{\partial \boldsymbol{H}_i(u,v)}{\partial(u,v)} \right) \right)^{1-k} \left(\det\left(\frac{\partial \boldsymbol{H}_v(u,v)}{\partial(u,v)} \right) \right)^k dudv. \tag{59}$$

Proof: Let $\mathcal{D}$ be the set of Delaunay triangles, and $\Omega_{xy} = \boldsymbol{H}_v(\Omega)$ be the image region of the surface $\boldsymbol{S}(u,v)$ at the virtual camera. The MAE can be approximated as

$$\begin{aligned} \mathrm{MAE} &\approx \frac{1}{S_{\Omega_{xy}}} \int_{\Omega_{xy}} |e(x,y)| dxdy & (60) \\ &= \frac{1}{S_{\Omega_{xy}}} \sum_{\Delta_i \in \mathcal{D}} \int_{\Delta_i} |e(x,y)| dxdy \\ &\leq \frac{1}{S_{\Omega_{xy}}} \sum_{\Delta_i \in \mathcal{D}} S_{\Delta_i} \Big(\frac{1}{2} R_i^2 \|\nabla^2 f_v\|_\infty + \max\{|\varepsilon|\} \\ &\qquad + \max\{\|\boldsymbol{\mu}\|_2\} \cdot \|\nabla f_v\|_\infty \Big). & (61) \end{aligned}$$

In each infinitesimal patch $d\omega$ around $(x,y) \in \Omega_{xy}$, we can approximate $R^2 \approx 2/(\pi\lambda(x,y))$ (see Lemma 3). Hence

$$\sum_{\Delta_i \in \mathcal{D}} S_{\Delta_i} R_i^2 \approx \int_{\Omega_{xy}} \frac{2}{\pi\lambda(x,y)} d\omega = \frac{2X_2}{\pi} \cdot \Delta_x \Delta_y. \tag{62}$$

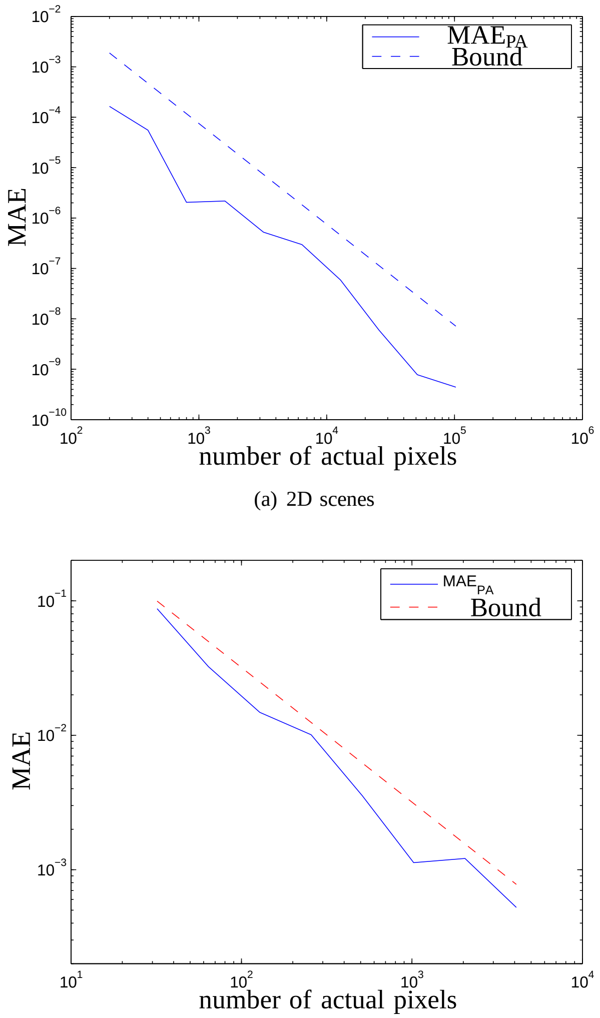
By changing the variables from (x,y) to (u,v), and substituting (62) into inequality (61), we indeed get (58). ■

Remark 5: The first term of (58), $X_1 = S_{\Omega_{xy}}$, is the area of the scene's image on the virtual image plane and does not depend on the actual camera configuration. The value of X_2, called 3D multiple-view term of second order, encodes the geometrical information of the actual cameras and the scene. We note that X_2 decays as $\mathcal{O}(N^{-1})$. The first term also depends on the resolution $\Delta_x\Delta_y$. Overall, in smooth regions, the first term has decay order $\mathcal{O}(\lambda^{-1})$, where λ is the local density of actual samples. The second term is the intensity estimate error bound E_T. The third term relates to the depth estimate error bound E_D (linearly) and the geometrical position between the scene and the virtual camera (via B_v).

Remark 6: A notable difference between the 3D case and the 2D case resides in the decay order of the first term. In (25), the first term has decay order $\mathcal{O}(\lambda^{-2})$, while in (58) the decay order is $\mathcal{O}(\lambda^{-1})$. To see this difference, note that the first term in inequality (50) contains R^2 having the same dimension with the sample density λ, whereas in (29), the first term contains $(x_2 - x_1)^2$ of the same dimension with λ^2.

VI. Experiments

We present numerical experiments to show that the bounds Theorems 1 and 3 are reasonably tight and accurately predict the trends of rendering error with respect to IBR parameters such as the number of actual pixels, depth errors. The experiments use a synthetic scene in Section VI-A and a real scene in Section VI-B. Section VI-B also serves as an example on estimating the bound in practice.



(a) 2D scenes

(b) 3D scenes

Fig. 7. The mean absolute error MAE (solid) and the theoretical bound (dashed) plotted against the number of actual pixels in the loglog axis. We note that the errors indeed have decay $\mathcal{O}(\lambda^{-2})$ for 2D scenes and $\mathcal{O}(\lambda^{-1})$ for 3D scenes, where λ is the local sample density, consistent with the results of Theorems 1 and 3.

A. *Synthetic scene*

We adopt a simple translational camera configuration in our experiments. All the actual and virtual cameras are located in the X-axis, looking to the direction of the Y-axis. The 2D scene consists of a flat surface with distance $d = 10$ to the cameras and the texture $T(u) = \sin(u)$ painted on the surface. We use camera resolution $\Delta_x = 0.01$. To validate the first term in (25), we set $E_T = E_D = 0$, and vary the number of actual cameras N. In Fig. 7(a), we show the mean absolute error MAE (solid) and the theoretical bound (dashed) plotted against the number of actual pixels in the loglog axis. We observe that both the MAE and the theoretical bound decay with slope $s = -2$, consistent with the result of Theorem 1.

Next, we validate the error bound (58) of Theorem 3 for a 3D synthetic scene consisting of a flat surface with constant depth $z = 10$ and the texture map $T(u, v) = \sin(u) + \sin(v)$. The Propagation Algorithm [20] is used for a planar camera configuration. All the actual and virtual cameras are placed in the xy-plane and focus to the direction of the z-axis. Specifically, $N = 10$ actual cameras are randomly placed in the square of dimensions 2×2 centered around the virtual camera position at $[5, 5, 0]^T$.

To validate the first term, we set $E_D = E_T = 0$ and plot in Fig. 7(b) the mean absolute errors MAE (solid) and the error bound (dashed) against the total number of actual pixels (equivalent to the local density of actual samples λ). The variation of λ is obtained by changing the resolution $\Delta_x \Delta_y$. Observe that the MAE indeed decays as $\mathcal{O}(\lambda^{-1})$, conforming to Theorem 3.

To validate the second term, we fix $\Delta_x = \Delta_y = 0.02$ and $E_D = 0$, and vary E_T. For each value of E_T, the intensity estimate errors are chosen randomly in the interval $[-E_T, E_T]$ following the uniform distribution. In Fig. 8, we show the mean absolute error MAE (solid) and the theoretical bound (dashed) plotted against the intensity estimate error bound E_T. Observe that the actual MAE fluctuates around one half of the error bound. The reason is that the error bound of Theorem 3 is derived for the worst case, whereas the actual MAE tends to follow the average errors.

Finally, we validate the last term of (58) by fixing $\Delta_x = \Delta_y = 0.02, E_T = 0$, and varying E_D. For each value of E_D, the depth estimate errors are chosen randomly in the interval $[-E_D, E_D]$ following the uniform distribution. In Fig. 9, we show the mean absolute error MAE (solid) and the theoretical bound (dashed) plotted against the depth estimate error bound E_D. Observe that the MAE indeed appears below the error bound and approximately linear to E_D.

B. *Real scene*

A data set of a real scene is used to validate the error bound of Theorem 1.[1] This section can also be considered as an example of the computation of various factors in the error bound in practice.

All the actual and virtual cameras are located in the X-axis looking to the direction of the Y-axis. The virtual image at $C_v = [4, 0, 0]^T$ is rendered, scanline by scanline, using the images and depth maps from actual cameras $C_1 = [2, 0, 0]^T$ and $C_2 = [6, 0, 0]^T$. The mean absolute error is computed using the available ground truth (see Fig. 10). The error bound of Theorem 1 is approximately computed, based on the data set, as follows.

Intensity estimate error bound E_T. Since the intensities are quantized into integers in the interval $[0, 255]$, we consider the intensity errors belong to the interval $(-0.5, 0.5]$. Hence, we choose

$$E_T = 1/2. \tag{63}$$

Bound of jitters $E_D B_v$. The data set provide the disparities, instead of depth, between two actual cameras $C_1 = [2, 0, 0]^T$ and $C_2 = [6, 0, 0]^T$. Hence, the bound of jitters $E_D B_v$ is directly computed instead of the depth estimate error bound E_D. Since the disparities between C_1 and C_2 are rounded to quarters of pixels, the disparities between C_1, C_2 and $C_v =$

[1] The data set is available at http://cat.middlebury.edu/stereo/newdata.html.

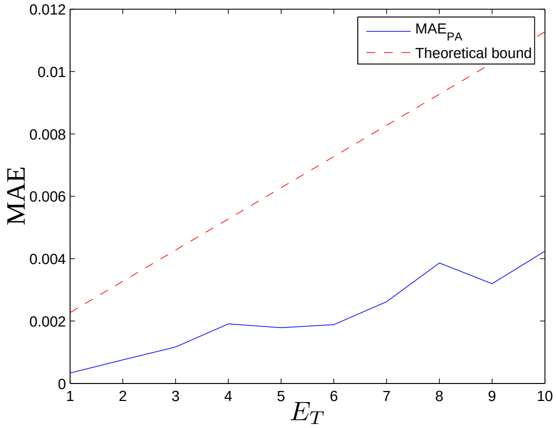


Fig. 8. The mean absolute error (MAE) (solid) and the theoretical bound (dashed) plotted against the intensity estimate error bound E_T.

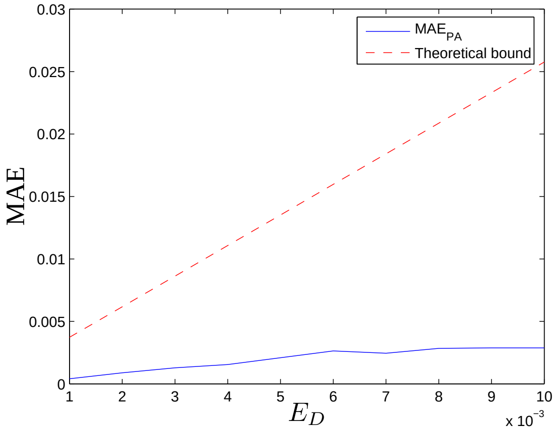


Fig. 9. The mean absolute error (MAE) (solid) and the theoretical bound (dashed) plotted against the depth estimate error bound E_D.

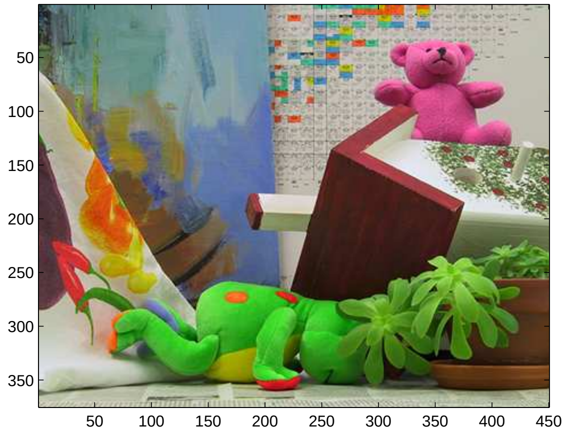


Fig. 10. The ground truth image of the scene at the virtual camera $C_v = [4,0,0]^T$.

$[4,0,0]^T$ are rounded to one eighth of a pixel. Hence, we adopt

$$E_D B_v = 1/16. \tag{64}$$

The resolution Δ_x. We assume the images in have 450 columns spread over the image line of length 2. Hence

$$\Delta_x = 1/225. \tag{65}$$

In practice, it turns out that the choice of the image line's length, and hence the resolution Δ_x, is not crucial. Its effect will be neutralized by the computation of $\|f'_v\|_\infty$ and $\|f''_v\|_\infty$.

Multiple-view terms Y_k. For a camera located at $X = X_0$ looking to the direction of the Y-axis, its projection matrix is

$$\mathbf{\Pi} = \begin{bmatrix} 1 & 0 & -X_0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{66}$$

Suppose that the scene surface is a parameterized curve $[X(u), Y(u)]^T$, for $u \in [a,b]$. The corresponding scene-to-image mapping is

$$H_{\mathbf{\Pi}}(u) = \frac{X(u) - X_0}{Y(u)}. \tag{67}$$

In this camera setting with two actual cameras at $C_1 = 2$ and $C_2 = 6$ and the virtual camera at $C_v = 4$, it can be verified that

$$H'_1(u) + H'_2(u) = 2H'_v(u), \quad u \in [a,b]. \tag{68}$$

As a consequence

$$Y_3/Y_1 = 1/4. \tag{69}$$

Note that the length of the image line does not affect the ratio Y_3/Y_1, although it does change Y_k individually.

L_∞ norms $\|f'_v\|_\infty$ and $\|f''_v\|_\infty$. Since there exist noises and discontinuities in the virtual image f_v, a preprocessing step is necessary to estimate $\|f_v^{(k)}\|_\infty$. To limit the effect of noise, the virtual image is first convolved with the derivative of order k of a Gaussian kernel

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{x^2}{2\sigma^2}\right). \tag{70}$$

In our experiment, we use the filter of length 10 pixels and $\sigma = 1$. Next, since the virtual image is discontinuous, we use the 95%-point value (instead of the 100%-point value, i.e. the maximum) of the convolution as the L_∞ norm. To have an idea, these values of $\|f'_v\|_\infty$ for different scanlines have average 53.19 and standard deviation 16.70, while the corresponding values for $\|f''_v\|_\infty$ are 36.86 and 11.13, respectively.

For each scanline, the error bounds of (25) are computed using the procedure described above. In Fig. 11, we show the mean absolute error (solid) of the virtual image rendered using the Propagation Algorithm [20] compared to the estimated bounds (dashed). Observe that the bound is tighter for scanlines with smoother intensity function $f_v(x)$.

VII. Conclusions and Discussions

We proposed a new framework to quantitatively analyze the rendering quality of IBR algorithms using per-pixel depth. We showed that IBR errors can be bounded based on sample intervals, sample errors, and jitters. We derived theoretical bounds for the mean absolute errors (MAEs) that successfully captured the effects of various factors such as depth and intensity estimate errors, the scene geometry and texture, the

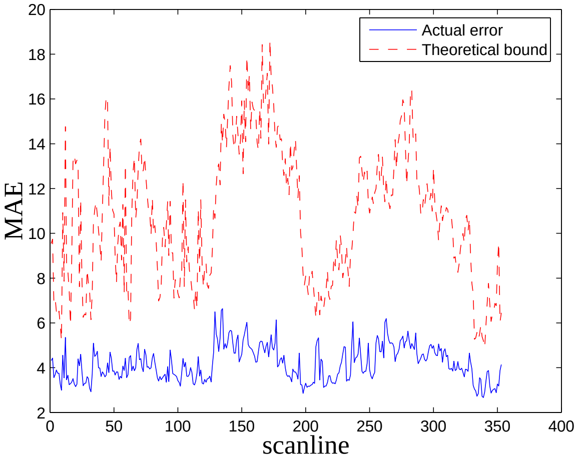


Fig. 11. The mean absolute error (solid) of the virtual image rendered using the Propagation Algorithm [20] compared to the estimated error bound of Theorem 1 (dashed) for each scanline of the scene shown in Fig. 10.

number of cameras and their characteristics. In particular, the proposed analysis suggests that the decay order of the MAE is $\mathcal{O}(\lambda^{-1})$ for 3D scenes and $\mathcal{O}(\lambda^{-2})$ for 2D scenes. The error bounds are also validated by experiments using synthetic and real scenes.

In the following, we discuss some implications of the proposed analysis. For simplicity, we use the result of Theorem 1.

Where to put the actual cameras? Theorem 1 suggests a potential application for camera placement. Suppose that we render a virtual image at camera $\mathbf{\Pi}_v$ given a number of N actual cameras with given depth and texture estimate errors E_D, E_T. We want to find the optimal camera positions, that is, optimal matrices $\{\mathbf{\Pi}\}_{i=1}^N$. Given that E_D, E_T, and N are fixed, the last two terms of the error bound in Theorem 1 are also fixed. To decrease the error bound, the only way is to minimize Y_3 (see (18) for $k = 3$):

$$Y_3 = \int_a^b \left(\sum_{i=1}^N H_i'(u) \right)^{-2} \left(H_v'(u)\right)^3 du. \tag{71}$$

In case Y_3 cannot be analytically minimized, numerical methods can be used to approximate the optimal configuration. If we want to optimize the actual camera positions for a set of virtual cameras, equation (71) should be the integral of Y_3 for all the virtual cameras. It will be interesting to investigate this approach and one proposed by Zhang and Chen [32].

Budget allocation. Suppose that a monetary budget c is available to buy range finders and cameras of cost c_D, c_T, respectively. The question is to how to allocate the budget c into range finders and cameras to best render the virtual image at $\mathbf{\Pi}_v$.

We assume that, due to the registration process, the depth estimate error E_D is a function of the number of range finders N_D. The texture estimate error E_T and resolution Δ_x are similar for all cameras. Hence, the error bound in (25) depends only on Y_3 and E_D. The optimal budget allocation is to use N_D^* range finders and N_T^* cameras, where N_D^*, N_T^* are the solution of the following optimization:

$$\min_{c_D N_D + c_T N_T \leq c} \left\{ \frac{3\Delta_x^2 \|f_v''\|_\infty}{4Y_1} Y_3(N_T) + B_v \|f_v'\|_\infty E_D(N_D) \right\}. \tag{72}$$

Bit allocation. Suppose that depth maps and images are recorded at the encoder and need to be transmitted over some communications channel to IBR decoders. The virtual image is rendered at the decoder upon receiving the depth maps and images. The question is how to distribute the channel capacity R into R_D for depth maps and R_T for images to optimize the rendering quality of the virtual images.

Let $E_T = E_T(R_T)$ and $E_D = E_D(R_D)$ be distortions of intensity and depth images corresponding to the transmission rate R_T, R_D. Since the first term of Theorem 1 does not depend on E_D, E_T, our optimal distribution of channel capacity $\{R_D^*, R_T^*\}$ is the solution of the following optimization:

$$\min_{R_D + R_T \leq R} \Big\{ E_T(R_T) + B_v \|f_v''\|_\infty E_D(R_D) \Big\}. \tag{73}$$

A key to solve these optimization problems is the understanding of the multiple-view term Y_3. In some special case, such as translational configurations, Y_3 can be simplified using a relationship similar to (68).

Acknowledgements. We thank Professors Narendra Ahuja, David Forsyth, Bruce Hajek, and Yi Ma (University of Illinois at Urbana-Champaign, USA) for valuable hints and criticisms.

Appendix
Proof of Proposition 4

Denote the following functions as linear interpolations of corresponding samples:

$$\begin{aligned}
\widehat{f}_{12}(x) &= \frac{x - x_1}{x_2 - x_1} f(x_2) + \frac{x_2 - x}{x_2 - x_1} f(x_1) \\
\widehat{f}_{1d}(x) &= \frac{x - x_1}{x_d - x_1} f(x_d^-) + \frac{x_d - x}{x_d - x_1} f(x_1) \\
\widehat{f}_{d2}(x) &= \frac{x - x_d}{x_2 - x_d} f(x_2) + \frac{x_2 - x}{x_2 - x_d} f(x_d^+).
\end{aligned}$$

Let $e_{12}(x), e_{1d}(x)$, and $e_{d2}(x)$ be the corresponding interpolation errors. The aggregated interpolation error is defined as

$$E_{12} = \int_{x_1}^{x_2} |e_{12}(x)| dx \tag{74}$$

for $\widehat{f}_{12}(x)$ over the interval $[x_1, x_2]$. The aggregated errors E_{1d} and E_{d2} are defined similarly.

Lemma 4: The equality

$$\widehat{f}_{12}(x_d) = \frac{\Delta_1}{\Delta} f(x_d^+) + \frac{\Delta_2}{\Delta} f(x_d^-) + \frac{\Delta_1 \Delta_2}{\Delta} J_1 + B \tag{75}$$

holds for some B such that

$$|B| \leq \frac{1}{2} \Delta_1 \Delta_2 \cdot \|f''\|_\infty. \tag{76}$$

Proof: Using the Taylor expansion we write

$$f(x_1) = f(x_d^-) - \Delta_1 f'(x_d^-) + \frac{1}{2} \Delta_1^2 f''(\xi_1) \tag{77}$$

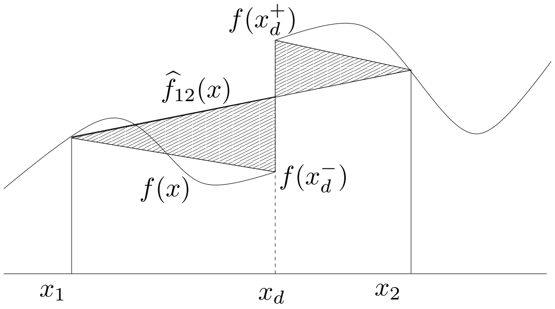


Fig. 12. Linear interpolation error for functions with discontinuities.

for some $\xi_1 \in [x_1, x_d]$. A similar equation can be also derived for x_2. Hence (75) holds for

$$B = \frac{\Delta_1^2 \Delta_2}{2\Delta} f''(\xi_1) + \frac{\Delta_1 \Delta_2^2}{2\Delta} f''(\xi_2). \qquad (78)$$

For B defined above, it is easy verify (75). ■

Next, we propose a bound, for the case $\mu_i = \varepsilon_i = 0$ for $i = 1, 2$, that is in fact tighter than the one proposed in Proposition 4.

Lemma 5: The aggregated error E_{12}, when there are no sample errors and jitters, can be bounded by

$$E_{12} \leq \frac{1}{12}\Delta^3 \cdot \|f''\|_\infty + \frac{\Delta_1^2 + \Delta_2^2}{2\Delta} \cdot |J_0| + \frac{1}{2}\Delta_1 \Delta_2 \cdot |J_1|. \quad (79)$$

Proof: We can bound E_{12} by the summation of E_{1d}, E_{d2}, and the area of the quadrangle formed by $[x_1, f(x_1)]^T$, $[x_2, f(x_2)]^T$, $[x_d, f(x_d^+)]^T$, and $[x_d, f(x_d^-)]^T$ (the shaded region in Fig. 12). Hence

$$E_{12} \leq E_{1d} + E_{d2} + \frac{\Delta_1}{2} |\widehat{f}_{12}(x_d) - f(x_d^-)| + \frac{\Delta_2}{2} |\widehat{f}_{12}(x_d) - f(x_d^+)|. \qquad (80)$$

Next, inequalities similar to Proposition 1 can be derived for E_{1d}, E_{d2}. Integrating both sides of these inequalities we obtain

$$E_{1d} \leq \frac{1}{12}\Delta_1^3 \cdot \|f''\|_\infty, \qquad E_{d2} \leq \frac{1}{12}\Delta_2^3 \cdot \|f''\|_\infty. \quad (81)$$

Substituting $\widehat{f}_{12}(x_d)$ as in (75) into (80), together with inequalities (81), we will indeed prove (79). ■

Finally, to extend Lemma 5 in the presence of sample errors and jitters, it is sufficient to prove the following lemma.

Lemma 6: The following inequality holds for $i = 1, 2$:

$$|f(x_i + \mu_i) + \varepsilon_i - f(x_i)| \leq |\varepsilon_i| + |\mu_i| \cdot \|f'\|_\infty + |J_0|. \quad (82)$$

Proof: For arbitrary $x, y \in [x_1, x_2]$, with $x \leq x_d \leq y$:

$$\begin{aligned} |f(y) - f(x)| &\leq |f(y) - f(x_d^+)| + |J_0| + |f(x_d^-) - f(x)| \\ &\leq |y - x_d| \cdot |f'(\theta_1)| + |x_d - x| \cdot |f'(\theta_2)| + |J_0| \\ &\leq |y - x| \cdot \|f'\|_\infty + |J_0|. \end{aligned}$$

The last inequality easily implies (82). ■

References

[1] E. H. Adelson and J. R. Bergen, “The plenoptic function and the elements of early vision,” in *Computational Models of Visual Processing*, M. Landy and J. A. Movshon, Eds. Cambridge, MA: MIT Press, 1991, pp. 3–20.

[2] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum, “Plenoptic sampling,” in *Proc. of SIGGRAPH*, 2000, pp. 307–318.

[3] E. Cinlar, “Superposition of point processes,” *Stochastic Point Processes: Statistical Analysis, Theory, and Applications*, pp. 546–606, 1972.

[4] R. L. Cook, “Stochastic sampling in computer graphics,” *ACM Trans. Graph.*, vol. 5, no. 1, pp. 51–72, 1986.

[5] C. de Boor, *A Practical Guide to Splines*. New York, NY: Springer-Verlag, 1978.

[6] M. N. Do, D. Marchand-Maillet, and M. Vetterli, “On the bandlimitedness of the plenoptic function,” in *Proc. IEEE Int. Conf. on Image Proc.*, vol. 3, September 2005, pp. 17–20.

[7] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. New York, NY: Prentice-Hall, 2002.

[8] R. G. Gallager, *Discrete Stochastic Processes*. Norwell, MA: Kluwer Academic Publishers, 1996.

[9] G. H. Golub and C. F. V. Loan, *Matrix Computations (3rd Ed.)*. Baltimore, MD: Johns Hopkins University Press, 1996.

[10] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY: Cambridge University Press, 2004.

[11] M. Hebert, “Active and passive range sensing for robotics,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 102–110.

[12] M. T. Hoopen and H. A. Reuver, “The superposition of random sequences of events,” *Biometrika*, vol. 53, no. 3–4, pp. 383–389, 1966.

[13] S. B. Kang, Y. Li, X. Tong, and H.-Y. Shum, *Foundations and Trends in Computer Graphics and Vision: Image-based Rendering*, B. Curless, L. V. Gool, and R. Szeliski, Eds. Hanover, MA: Now Publishers, 2006, vol. 2, no. 3.

[14] Z. Lin and H.-Y. Shum, “A geometric analysis of light field rendering,” *Int. J. Comput. Vision*, vol. 58, no. 2, pp. 121–138, July 2004.

[15] W. Mark, L. McMillan, and G. Bishop, “Post-rendering 3D warping,” in *Proc. I3D Graphics Symp.*, 1997, pp. 7–16.

[16] L. McMillan and G. Bishop, “Plenoptic modeling: An image-based rendering system,” in *Proc. of SIGGRAPH*, 1995, pp. 39–46.

[17] L. McMillan, “An image-based approach to three-dimensional computer graphics,” Ph.D. dissertation, University of North Carolina at Chapel Hill, 1997.

[18] R. Ng, “Fourier slice photography,” in *Proc. of SIGGRAPH*, 2005, pp. 735–744.

[19] H. T. Nguyen, “Multisensor signal processing: Theory and algorithms for image-based rendering and multichannel sampling,” Ph.D. dissertation, University of Illinois at Urbana-Champaign, Urbana, Illinois, October 2007.

[20] H. T. Nguyen and M. N. Do, “Image-based rendering with depth information using the propagation algorithm,” in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, vol. 2, Philadelphia, March 2005, pp. 589–592.

[21] ——, “Error analysis for image-based rendering with depth information,” in *Proc. IEEE Int. Conf. on Image Proc.*, October 2006, pp. 381–384.

[22] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd ed. New York, NY: Wiley, 2000.

[23] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 2nd ed. New York, NY: McGraw-Hill, 1984.

[24] P. M. Prenter, *Splines and Variational Methods*. New York, NY: Wiley, 1975.

[25] D. Scharstein and R. Szeliski, “High-accuracy stereo depth maps using structured light,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 195–202, June 2003.

[26] D. Scharstein, R. Szeliski, and R. Zabih, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *IEEE Workshop on Stereo and Multi-Baseline Vision*, pp. 131–140, December 2001.

[27] J. Shade, S. Gortler, L. He, and R. Szeliski, “Layered depth images,” in *Proc. of SIGGRAPH*, 1998, pp. 231–242.

[28] H.-Y. Shum, S.-C. Chan, and S. B. Kang, *Image-Based Rendering*. New York, NY: Springer, 2007.

[29] M. Unser, “Sampling - 50 years after Shannon,” *Proc. IEEE*, vol. 88, no. 4, pp. 569–587, April 2000.

[30] C. Zhang and T. Chen, “Spectral analysis for sampling image-based rendering data,” *IEEE Trans. Circ. and Syst. for Video Tech.*, vol. 13, no. 11, pp. 1038–1050, November 2003.

[31] ——, “A survey on image-based rendering - representation, sampling and compression,” *EURASIP Signal Processing: Image Communication*, pp. 1–28, January 2004.

[32] ——, “Active rearranged capturing of image-based rendering scenes theory and practice,” *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 520–531, April 2007.