

## Anisotropic density estimation for photon mapping

Fu-Jun Luan<sup>1</sup>, Li-Fan Wu<sup>1</sup>, and Kun Xu<sup>1</sup> (✉)

© The Author(s) 2015. This article is published with open access at Springerlink.com

**Abstract** Photon mapping is a widely used technique for global illumination rendering. In the density estimation step of photon mapping, the indirect radiance at a shading point is estimated through a filtering process using nearby stored photons; an isotropic filtering kernel is usually used. However, using an isotropic kernel is not always the optimal choice, especially for cases when eye paths intersect with surfaces with anisotropic BRDFs. In this paper, we propose an anisotropic filtering kernel for density estimation to handle such anisotropic eye paths. The anisotropic filtering kernel is derived from the recently introduced anisotropic spherical Gaussian representation of BRDFs. Compared to conventional photon mapping, our method is able to reduce rendering errors with negligible additional cost when rendering scenes containing anisotropic BRDFs.

**Keywords** photon mapping; density estimation; anisotropic; anisotropic spherical Gaussian

### 1 Introduction

Global illumination is a long and important research direction in computer graphics. Photon mapping [1, 2] has always been a widely used technique for global illumination due to its high rendering quality and good efficiency. It is a two-pass algorithm. In the first pass, a large number of photons are emitted from light sources, traced through the scene, and stored in a photon map. In the second pass, each eye ray is traced from the viewpoint into the scene until it hits a diffuse surface; the indirect radiance

at this intersection point can be approximated by averaging contributions from nearby photons within an encompassing disk of a fixed radius. This step is usually referred to as density estimation.

Density estimation allows the same photons to be reused in different eye paths, and hence makes photon mapping to be more efficient than Monte Carlo ray tracing [3, 4]. The radius of the disk is an important parameter in density estimation, which largely affects the bias and variance of the results. Larger radius gives lower variance but higher bias, while smaller radius gives higher variance but lower bias. Instead of equally weighting all photons, smooth filtering kernels can be applied in order to further reduce bias, such as the cone filter [1] and the Epanechnikov kernel [5, 6].

We focus on using photon mapping to render scenes with anisotropic BRDFs. Since density estimation can only be applied on diffuse surfaces, eye rays towards surfaces with anisotropic BRDFs need further tracing in the scene until hitting a diffuse surface. We refer to these eye paths, from viewpoint through reflections at anisotropic surfaces to density estimation points at diffuse surfaces, as anisotropic eye paths. Intuitively thinking, density estimation through anisotropic eye paths can be beneficial from anisotropic filtering kernels. However, existing kernels are mainly isotropic. One exception is photon differentials [7, 8]. They use an anisotropic filtering kernel for density estimation derived from ray differentials stored in photons, and it is effective in reducing bias when rendering caustics. However, they only take into consideration of the information in light paths but not eye paths in constructing filtering kernel, hence they will not be beneficial to render anisotropic eye paths.

To improve efficiency in rendering anisotropic eye paths, we propose a novel anisotropic filtering

<sup>1</sup> TNList, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: xukun@tsinghua.edu.cn (✉).

Manuscript received: 2014-11-21; accepted: 2015-03-04

kernel for density estimation, which considers the anisotropic BRDFs on the eye path. Our method works as follows: first, we represent anisotropic BRDFs using the recently proposed anisotropic spherical Gaussians (ASGs) [9] of BRDFs and obtain anisotropic eye paths by importance sampling anisotropic BRDFs; next, we construct the anisotropic filtering kernel in the direction space according to the gradient of the ASG; after that, the final anisotropic filtering kernel is obtained by projecting it from the direction space to the tangent plane of the density estimation point. Compared to existing works, our experiments demonstrate that our anisotropic kernel yields a better accuracy in rendering anisotropic scenes without incurring additional rendering costs.

## 2 Related works

**Photon mapping.** Photon mapping is a popular technique of global illumination rendering, which is based on photon density estimation [10]. Arvo [11] first combined density estimation with light transport simulation by introducing particle tracing algorithm. Jensen [1, 2] introduced photon mapping which approximates the radiance by local photon density estimation. Although photon mapping is a biased algorithm, it is good at rendering various lighting effects like caustics.

To improve efficiency and robustness of photon mapping, Hachisuka et al. [12] proposed progressive photon mapping (PPM), which breaks the memory requirement bottleneck for storing a large number of photons and eliminates bias gradually by reducing the radius of the radiance estimate kernel progressively. Stochastic progressive photon mapping [13] is more robust to complex scenes and is capable of producing more distributed ray tracing effects, e.g., depth of field and motion blur. Hachisuka et al. [14] introduced an error estimation framework for progressive photon mapping, while Knaus and Zwicker [15] presented an asymptotic analysis of converging rates of variance and bias. Knaus and Zwicker [15] introduced a probabilistic framework of progressive photon mapping, which does not need to maintain local photon statistics and can be implemented easier. Kaplanyan and Dachsbacher [16] proposed adaptive progressive

photon mapping by selecting local kernel bandwidth adaptively and achieved a higher convergence rate.

Many works have been done in order to further improve the robustness of photon mapping. Schjøth et al. [7, 8] proposed photon differentials, which shape an anisotropic filtering kernel for density estimation derived from ray differentials created in the photon tracing pass and can produce better caustic quality with less photons. Spencer and Jones [17] introduced photon relaxation which redistributes the photons into a blue noise distribution. By manipulating the underlying points with feature detection and preservation, photon relaxation achieves noise reduction without increasing bias.

Recently, multiple importance sampling (MIS) [18] has been used in photon density estimation. Vorba [19] extended photon mapping into a bidirectional way and used MIS to combine light paths and eye paths of different lengths. A unified algorithm [20, 21] combining bidirectional path tracing (BPT) and progressive photon mapping (PPM) is more robust to handle complex illuminations and specular-diffuse-specular paths. These MIS based algorithms are more efficient than both BPT and PPM.

**Anisotropic appearance.** Many real world materials are anisotropic, such as metal and wood. Anisotropic appearance exhibits changes with respect not only to the azimuthal difference between incoming and viewing directions, but also to the azimuthal angle of incoming direction.

Kajiya [22] introduced the first anisotropic BRDF model. After that, a number of parametric anisotropic BRDF models have been proposed, such as Ward's model [23] and Ashikhmin's model [24]. Various models have also been proposed to handle specific anisotropic materials, such as hair [25], wood [26], and cloth [27, 28].

Recently, Xu et al. [9] introduced a representation called anisotropic spherical gaussian (ASG), which is efficient and effective in representing anisotropic spherical functions, such as anisotropic BRDFs and visibilities. Furthermore, ASGs have close-form solutions for integral, multiplication, and convolution operators. Due to its effectiveness at approximating anisotropic BRDFs, in our work we use it to represent anisotropic BRDFs.

### 3 Background

Photon mapping [1, 2] is a two-pass global illumination algorithm. It is good at producing illumination effects like caustics and is efficient in rendering low-variance images. In the first pass, a large number of photons are emitted from light sources, traced through the scene, and stored in a photon map. In the second pass, each eye ray is traced from the viewpoint into the scene until it hits a diffuse surface. Then the indirect radiance at this intersection point can be evaluated through density estimation. Specifically, the indirect radiance is evaluated as the average of fluxes of its  $N$  nearest photons:

$$L_o(\mathbf{o}) \approx \frac{1}{\pi r^2} \sum_{k=1}^N \rho(\mathbf{i}_k, \mathbf{o}) \Phi_k \quad (1)$$

where  $k$  iterates over the  $N$  nearest photons,  $\Phi_k$  and  $\mathbf{i}_k$  are the flux and incident direction of the  $k$ -th photon respectively,  $\mathbf{o}$  is the outgoing direction,  $\rho$  denotes the BRDF, and  $r$  is the radius of the disk that encompasses the  $N$  nearest photons. The above equation equally weights all the nearest neighboring photons. To reduce bias, we can weight each photon according to the distance from the photon to the density estimation point. Then, Eq. (1) can be rewritten as

$$L_o(\mathbf{o}) \approx \frac{1}{\pi r^2} \sum_{k=1}^N w(d_k/r) \rho(\mathbf{i}_k, \mathbf{o}) \Phi_k \quad (2)$$

where  $d_k$  is the distance from the  $k$ -th photon to the density estimation point, and the kernel function  $w(\cdot)$  is a monotonic decreasing function. Different kernels have been proposed, such as the cone filter [1] and the Epanechnikov kernel [5, 6], both of which have been demonstrated to be useful in reducing bias.

### 4 Our method

As shown in Fig. 1, considering a typical anisotropic eye path, starting from viewpoint to a point  $P_0$  on an anisotropic surface, then reflected to the density estimation point  $P_1$  on a diffuse surface, the anisotropic filtering kernel is computed and applied through 4 steps. First, based on the anisotropic spherical Gaussian (ASG) representation [9] of BRDFs, we obtain the 2D BRDF slice of reflected direction  $\mathbf{r}$  at point  $P_0$ , and approximate it as an ASG through ASG warping [9] (Section 4.1); next,

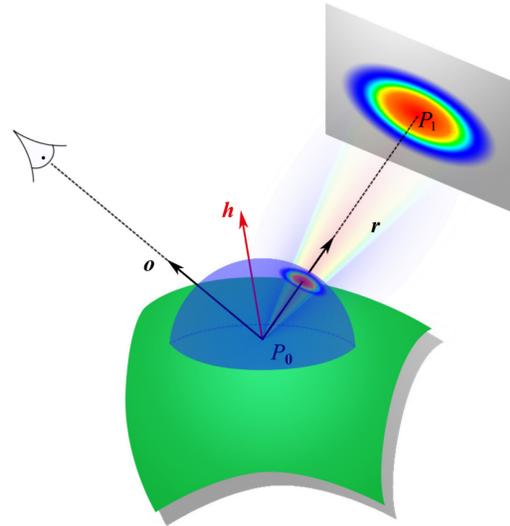


Fig. 1 Illustration of anisotropic eye paths.

we construct the anisotropic filtering kernel in the direction space according to the gradient of the warped ASG (Section 4.2); after that, we project the kernel in the direction space to the local coordinate system of the density estimation point  $P_1$  to obtain the final anisotropic filtering kernel (Section 4.3); finally, we will show how this anisotropic kernel can be used in density estimation (Section 4.4).

#### 4.1 Spherical warping

We use ASGs [9] to represent the anisotropic BRDF at  $P_0$ . We choose ASGs as our BRDF representation since ASGs are simple to compute and have good scalability in approximating anisotropic signals. As shown in their work, commonly used parametric anisotropic BRDFs, such as Ward’s model [23] and Ashikhmin’s model [24], can be well approximated by one ASG. ASG based BRDF representation is based on the microfacet model [29, 30]. Specifically, the normal distribution function (NDF) is approximated using one ASG, and then the BRDF at a specific view slice is obtained through an ASG spherical warping operator. As shown in Fig. 1, denote the view direction as  $\mathbf{o}$ , the reflected direction from  $P_0$  to  $P_1$  as  $\mathbf{r}$ , the anisotropic BRDF  $\rho(\mathbf{r}, \mathbf{o})$  is approximated by a warped ASG:

$$\rho(\mathbf{r}, \mathbf{o}) \approx M(\mathbf{r}, \mathbf{o}) G(\mathbf{r}; [\mathbf{x}, \mathbf{y}, \mathbf{z}], [\lambda, \mu]) \quad (3)$$

where  $M$  is a smooth function that combines the shadowing term and Fresnel term;  $G$  is an ASG;  $\mathbf{x}, \mathbf{y}, \mathbf{z}$  are the tangent, bi-tangent, lobe axes, respectively;  $\lambda$  and  $\mu$  are the bandwidths for  $\mathbf{x}$ - and  $\mathbf{y}$ -axes, respectively. Those parameters of the ASG

$G$  are computed from the NDF function (which is also an ASG) and the view direction  $\mathbf{o}$  through the ASG warping operator (please refer to their paper [9] for more details of ASGs and the ASG warping operator), hence it is also referred to as the warped ASG. For simplicity, we denote the warped ASG  $G(\mathbf{r}; [\mathbf{x}, \mathbf{y}, \mathbf{z}], [\lambda, \mu])$  as  $G(\mathbf{r})$  for short.

### 4.2 Kernel construction

As shown in Fig. 1, imagining an elliptical filtering kernel is applied at a specific direction to the warped ASG. By observation, it is easy to notice that the optimal choice to preserve the structure in filtered results is to set the minor axis of the ellipse as the gradient direction and set the major axis as the tangent direction (i.e., the direction perpendicular to the gradient direction). Based on this observation, we construct our elliptical filtering kernel in the direction space according to the gradient of the warped ASG.

We first compute the gradient  $\mathbf{g}$  of the warped ASG  $G(\mathbf{r})$  using the formula below:

$$\mathbf{g} = \nabla G - (\nabla G \cdot \mathbf{r})\mathbf{r} \tag{4}$$

where  $\nabla G$  is computed by directly applying gradient operation to the warped ASG function in the 3D vector space. Since  $\mathbf{r}$  is a direction (i.e., constrained on the unit sphere) and it is not an arbitrary 3D vector, we need to minus  $(\nabla G \cdot \mathbf{r})\mathbf{r}$  to obtain the true gradient to make it perpendicular to direction  $\mathbf{r}$ . After that, we simply set the direction of the minor axis  $\mathbf{u}_d$  as the gradient direction  $\mathbf{u}_d = \mathbf{g}/\|\mathbf{g}\|$ , and set the major axis direction as the tangent direction  $\mathbf{v}_d = \mathbf{r} \times \mathbf{u}_d$ .

Now, we need to determine the lengths of the minor/major axes. We obtain the axis lengths by constraining relative value changes inside the ellipse within a predefined threshold  $\epsilon$ , and in our implementation we set  $\epsilon = 0.02$ . Along the minor axis, we approximate the value change using first order Taylor expansion at direction  $\mathbf{r}$ , and the minor axis length  $l_u$  is obtained by

$$(\partial G/\partial \mathbf{u}_d) \cdot l_u = \epsilon \cdot G \Rightarrow l_u = \frac{\epsilon \cdot G}{\partial G/\partial \mathbf{u}_d} \tag{5}$$

where  $\partial G/\partial \mathbf{u}_d$  is the directional derivative of the warped ASG function  $G$  along the minor axis direction  $\mathbf{u}_d$ . Since  $\mathbf{u}_d$  is the gradient direction,  $\partial G/\partial \mathbf{u}_d$  equals to the length of gradient  $\|\mathbf{g}\|$ . Along the major axis, since the directional derivative along it is zero, we use second order Taylor expansion to

obtain the major axis length  $l_v$ :

$$\frac{1}{2} \cdot \frac{\partial^2 G}{\partial \mathbf{v}_d^2} \cdot l_v = \epsilon \cdot G \Rightarrow l_v = 2 \frac{\epsilon \cdot G}{\partial^2 G/\partial \mathbf{v}_d^2} \tag{6}$$

where  $\partial^2 G/\partial \mathbf{v}_d^2$  is the second order derivative along the major axis direction  $\mathbf{v}_d$ . We now obtain the minor/major axes (i.e.,  $\mathbf{u}$  and  $\mathbf{v}$ ) by combining its directions and lengths:

$$\mathbf{u} = l_u \mathbf{u}_d, \quad \mathbf{v} = l_v \mathbf{v}_d \tag{7}$$

### 4.3 Planar projection

We now have obtained the ellipse for filtering in the direction space. We need to further project it to the tangent plane of the density estimation point  $P_1$ , since the density estimation is finally performed on this plane.

We denote the minor/major axes of the projected ellipse as  $\mathbf{s}$  and  $\mathbf{t}$ . Note that these two projected axes are not necessarily perpendicular to each other. As shown in Fig. 2, take the minor axis as an example, the projected minor axis  $\mathbf{s}$  should satisfy two conditions:

$$\mathbf{s} = \mathbf{u} + k \cdot \mathbf{r}, \quad \mathbf{s} \cdot \mathbf{n} = 0 \tag{8}$$

where  $k$  is a scalar coefficient and  $\mathbf{n}$  is the normal direction of the density estimation center. Combining the above two equations leads to

$$\mathbf{s} = \mathbf{u} - \left( \frac{\mathbf{u} \cdot \mathbf{n}}{\mathbf{r} \cdot \mathbf{n}} \right) \cdot \mathbf{r} \tag{9}$$

We can similarly obtain the projected major axis  $\mathbf{t}$  as

$$\mathbf{t} = \mathbf{v} - \left( \frac{\mathbf{v} \cdot \mathbf{n}}{\mathbf{r} \cdot \mathbf{n}} \right) \cdot \mathbf{r} \tag{10}$$

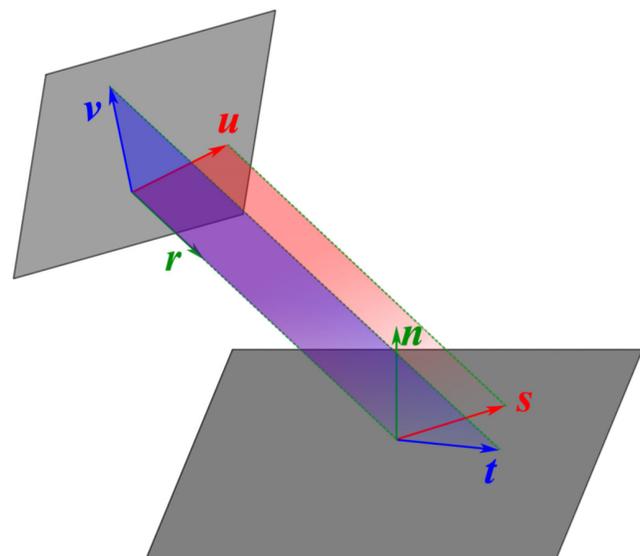


Fig. 2 Illustration of planar projection.

### 4.4 Density estimation

After obtaining the ellipse on the tangent plane of the density estimation point  $P_1$ , we now explain how to use it as a filtering kernel in density estimation. Specifically, we explain how to compute the weight for each photon in Eq. (2). First, we rescale the ellipse to make it fit the encompassing disk of the  $N$  nearest neighboring photons, i.e., rescale the ellipse to make the length of the major axis equal to the radius of the disk. The scaled minor/major axes become:

$$\mathbf{s}' = \frac{r}{\|\mathbf{t}\|} \cdot \mathbf{s}, \quad \mathbf{t}' = \frac{r}{\|\mathbf{t}\|} \cdot \mathbf{t} \tag{11}$$

where  $r$  is the radius of the encompassing disk. After that, for each photon, we denote the offset from the density estimation center to the photon position along the tangent plane as  $\mathbf{d}$ . As shown in Fig. 3, we can express the offset  $\mathbf{d}$  as a linear combination of the projected minor/major axes:  $\mathbf{d} = a\mathbf{s}' + b\mathbf{t}' = \frac{m}{\|\mathbf{s}'\|}\mathbf{s}' + \frac{n}{\|\mathbf{t}'\|}\mathbf{t}'$ . The coefficients  $a$  and  $b$  can be easily obtained by equations below:

$$\left. \begin{aligned} a &= \frac{m}{\|\mathbf{s}'\|} = \frac{(\mathbf{s}' \cdot \mathbf{d}) \cdot \|\mathbf{t}'\|^2 - (\mathbf{s}' \cdot \mathbf{t}') \cdot (\mathbf{t}' \cdot \mathbf{d})}{\|\mathbf{s}'\|^2 \|\mathbf{t}'\|^2 - (\mathbf{s}' \cdot \mathbf{t}')^2} \\ b &= \frac{n}{\|\mathbf{t}'\|} = \frac{(\mathbf{t}' \cdot \mathbf{d}) \cdot \|\mathbf{s}'\|^2 - (\mathbf{s}' \cdot \mathbf{t}') \cdot (\mathbf{s}' \cdot \mathbf{d})}{\|\mathbf{s}'\|^2 \|\mathbf{t}'\|^2 - (\mathbf{s}' \cdot \mathbf{t}')^2} \end{aligned} \right\} \tag{12}$$

The weight used in density estimation is computed as a Gaussian:

$$w(\mathbf{d}) = \exp(-a^2 - b^2) \tag{13}$$

Note that we need to normalize the sum of the weights of all photons into one when performing density estimation in Eq. (2).

## 5 Results

**Implementation.** We implement our method on a consumer level PC with an Intel Core i7-

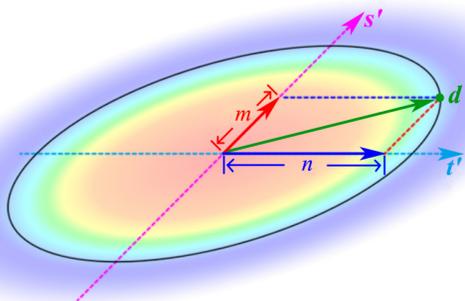


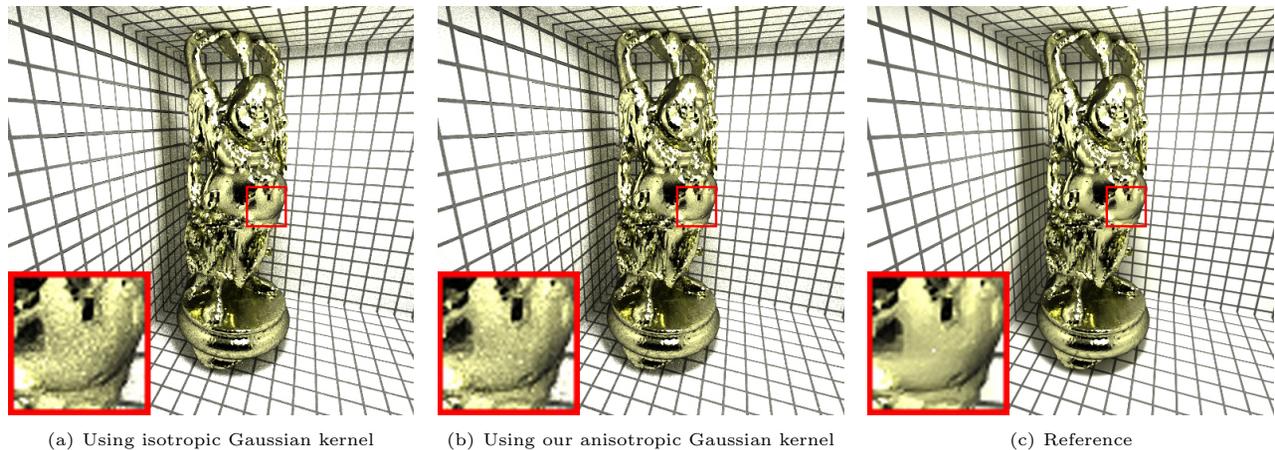
Fig. 3 Illustration of linear combination.

3770 3.4GHz CPU and 8GB memory. In our implementation, anisotropic filtering is only applied to density estimation for anisotropic eye paths. For other types of eye paths, e.g., eye ray starting from the viewpoint directly hits a diffuse surface, our method degenerates to use isotropic filtering in density estimation instead. Note that our method focuses on improving the effectiveness of density estimation and hence is perpendicular to other photon mapping improvements such as progressive photon mapping [12, 15]. To obtain consistent results, we use the progressive photon mapping framework to produce final rendering results. In each iteration, we reduce the radius of density estimation with a reduction parameter  $\alpha = 2/3$ .

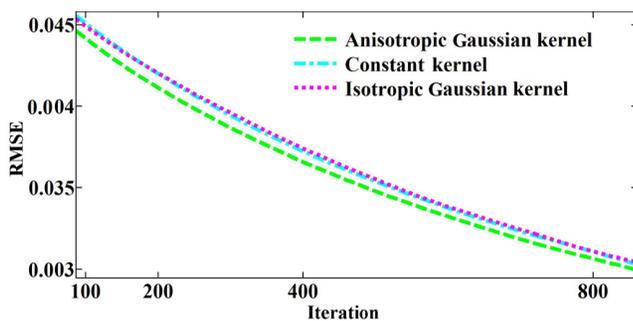
**Comparison.** To demonstrate the efficiency of our anisotropic filtering kernel, we compare the rendering results of our anisotropic filtering kernel to those of an isotropic kernel and a constant kernel, using several scenes with anisotropic glossy materials. In our comparison among different methods, only the filtering kernels are different while other parameters such as the image resolution, the number of photons and the number of iterations keep the same. We compare the results visually, as well as measuring root mean square errors (RMSE) to ground truth.

Figure 4 shows an anisotropic glossy buddha with anisotropic ratio of 10 inside a diffuse textured box. As shown in the enlarged images in Fig. 4, our proposed anisotropic filtering kernel produces less noise than an isotropic filtering kernel with equal rendering time. Note that the time required for constructing the anisotropic filter in our method is negligible compared to other steps in photon mapping, hence, our method won't incur additional costs. Figure 5 shows the RMSE of the rendering results of a constant kernel, an isotropic Gaussian kernel, and our anisotropic Gaussian kernel. Our method consistently gives the smallest error.

Figure 6 shows a scene with a teapot, a fork, and a spoon, and the anisotropic ratios are 10, 3.3, and 3.3, respectively. The surrounding objects are diffuse textures. Figure 7 shows a scene with anisotropic glossy pans and pots whose anisotropic ratios are 10. In both scenes, our method achieves better visual quality and lower RMSE than using isotropic kernel with equal rendering time.



**Fig. 4** Anisotropic happy buddha. (a) and (b) Photon mapping results using isotropic and anisotropic density estimation kernels, respectively; (c) reference.



**Fig. 5** RMSE curve. We plot the RMSE of the rendered images with respect to the number of iterations used in progressive photon mapping in rendering the buddha scene. Notice that our method consistently gives the smallest error.

## 6 Conclusions

In this work, we have presented an anisotropic filtering kernel for density estimation. The kernel takes into consideration of the anisotropic BRDFs on the eye path. It is derived from the ASG representation of anisotropic BRDFs. Through experiments and comparisons, our proposed anisotropic kernel is demonstrated to produce lower rendering error without incurring additional costs in rendering anisotropic scenes.

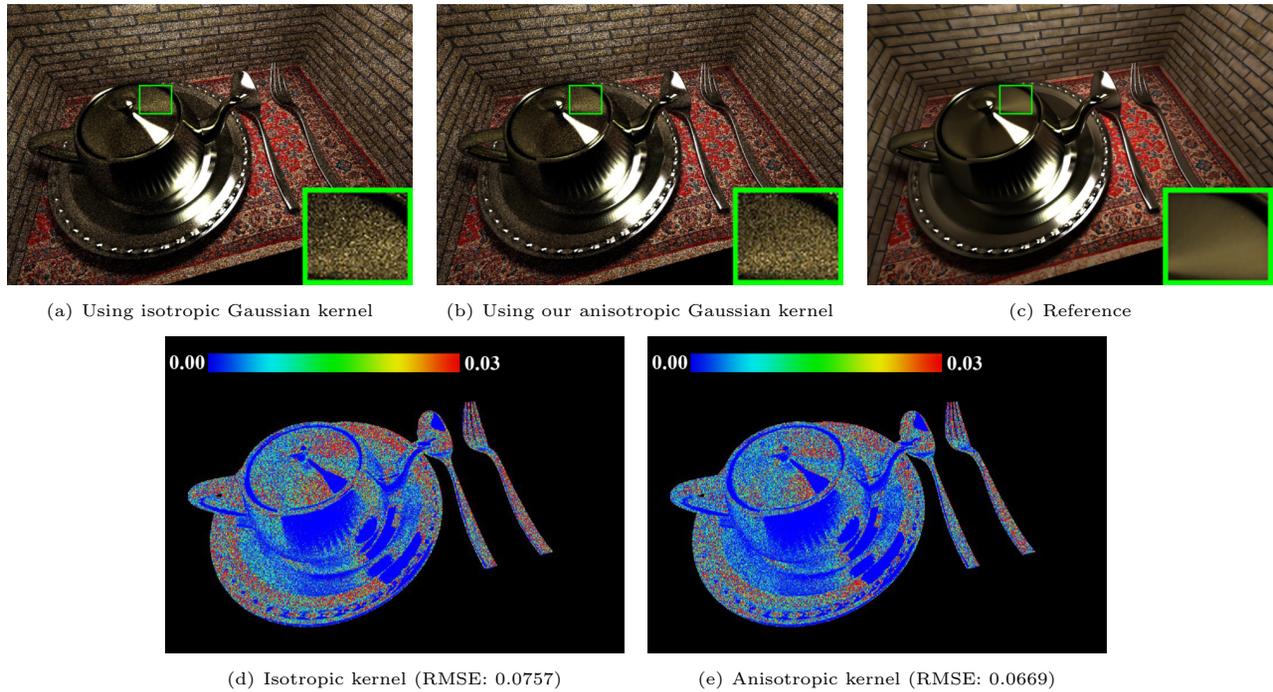
## Acknowledgements

We thank the anonymous reviewers for their valuable comments. This work was supported by the National High-tech R&D Program of China (No. 2012AA011802) and the National Natural Science Foundation of China (No. 61170153).

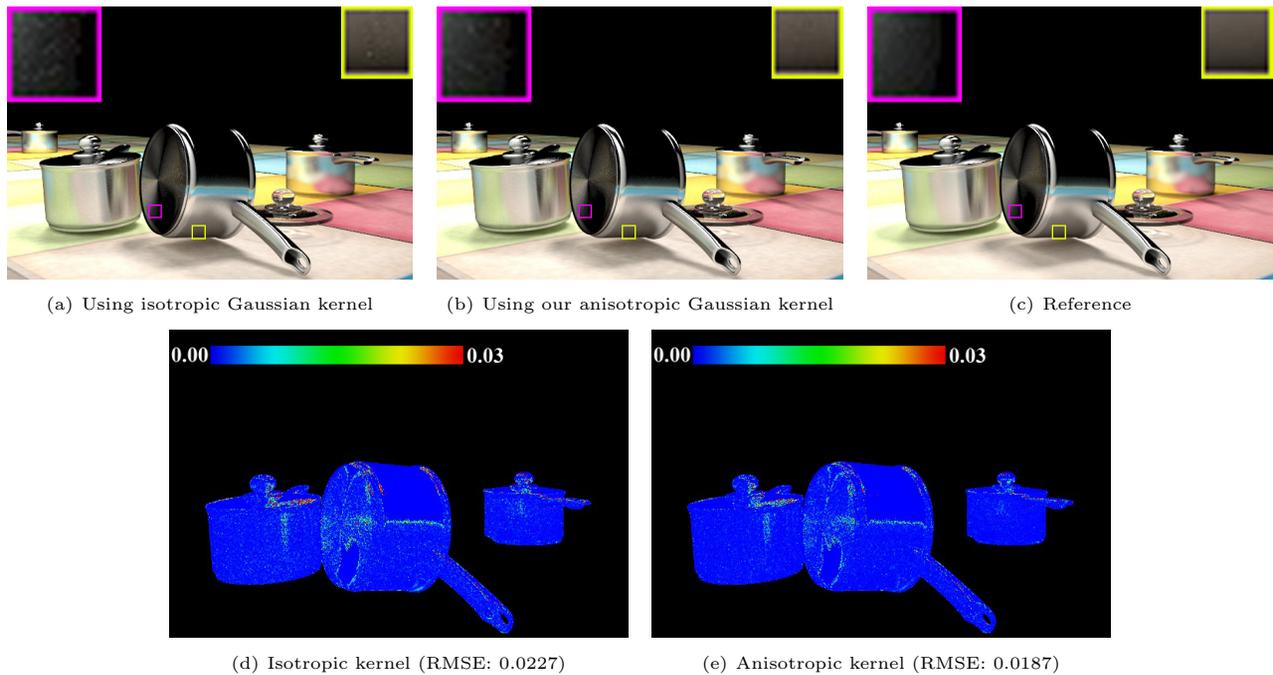
**Open Access** This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

## References

- [1] Jensen, H. W. Global illumination using photon maps. In: Proceedings of the eurographics workshop on Rendering techniques, 21–30, 1996.
- [2] Jensen, H. W. *Realistic Image Synthesis Using Photon Mapping*. Natick, MA, USA: A. K. Peters, Ltd., 2001.
- [3] Cook, R. L. Stochastic sampling in computer graphics. *ACM Transactions on Graphics* Vol. 5, No. 1, 51–72, 1986.
- [4] Cook, R. L.; Porter, T.; Carpenter, L. Distributed ray tracing. *ACM SIGGRAPH Computer Graphics* Vol. 18, No. 3, 137–145, 1984.
- [5] Roland, S. Bias compensation for photon maps. *Computer Graphics Forum* Vol. 22, No. 4, 729–742, 2003.
- [6] Walter, B. J. Density estimation techniques for global illumination. Ph.D. Thesis. Cornell University, 1998.
- [7] Schjøth, L. Anisotropic density estimation in global illumination. Ph.D. Thesis. University of Copenhagen, 2009.
- [8] Schjøth, L.; Frisvad, J. R.; Erleben, K.; Sporrang, J. Photon differentials. In: Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia, 179–186, 2007.
- [9] Xu, K.; Sun, W.-L.; Dong, Z.; Zhao, D.-Y.; Wu, R.-D.; Hu, S.-M. Anisotropic spherical Gaussians. *ACM Transactions on Graphics* Vol. 32, No. 6, Article No. 209, 2013.



**Fig. 6** Anisotropic teapot. (a) and (b) Photon mapping results using isotropic and anisotropic density estimation kernels, respectively; (c) reference; (d) and (e) RMSE visualization.



**Fig. 7** Anisotropic frying pan. (a) and (b) Photon mapping results using isotropic and anisotropic density estimation kernels, respectively; (c) reference; (d) and (e) RMSE visualization.

[10] Hachisuka, T.; Jarosz, W.; Bouchard, G.; Christensen, P.; Frisvad, J. R.; Jakob, W.; Jensen, H. W.; Kaschak, M.; Knaus, C.; Selle, A.; Spencer, B. State of the art in photon density estimation. In: ACM SIGGRAPH 2012 Courses, Article No. 6, 2012.

[11] Arvo, J. Backward ray tracing. In: *Developments in Ray Tracing*, SIGGRAPH '86 Course Notes, Vol. 12, 259–263, 1986.

[12] Hachisuka, T.; Ogaki, S.; Jensen, H. W. Progressive photon mapping. *ACM Transactions on Graphics* Vol.

- 27, No. 5, Article No. 130, 2008.
- [13] Hachisuka T.; Jensen H. W. Stochastic progressive photon mapping. *ACM Transactions on Graphics* Vol. 28, No. 5, Article No. 141, 2009
- [14] Hachisuka, T.; Jarosz, W.; Jensen, H. W. A progressive error estimation framework for photon density estimation. *ACM Transactions on Graphics* Vol. 29, No. 6, Article No. 144, 2010.
- [15] Knaus, C.; Zwicker, M. Progressive photon mapping: A probabilistic approach. *ACM Transactions on Graphics* Vol. 30, No. 3, Article No. 25, 2011.
- [16] Kaplanyan, A. S.; Dachsbacher, C. Adaptive progressive photon mapping. *ACM Transactions on Graphics* Vol. 32, No. 2, Article No. 16, 2013.
- [17] Spencer, B.; Jones, M. W. Into the blue: Better caustics through photon relaxation. *Computer Graphics Forum* Vol. 28, No. 2, 319–328, 2009.
- [18] Veach, E. Robust Monte Carlo methods for light transport simulation. Ph.D. Thesis. Stanford University, 1998.
- [19] Vorba, J. Bidirectional photon mapping. In: The 15th Central European Seminar on Computer Graphics, Vol. 2, 2011. Available at <http://cgg.mff.cuni.cz/~jaroslav/papers/2011-bdpm/vorba2011-bdpm.pdf>.
- [20] Georgiev, I.; Krivánek, J.; Davidovic, T.; Slusallek, P. Light transport simulation with vertex connection and merging. *ACM Transactions on Graphics* Vol. 31, No. 6, Article No. 192, 2012.
- [21] Hachisuka, T.; Pantaleoni, J.; Jensen, H. W. A path space extension for robust light transport simulation. *ACM Transactions on Graphics* Vol. 31, No. 6, Article No. 191, 2012.
- [22] Kajiya, J. T. Anisotropic reflection models. *ACM SIGGRAPH Computer Graphics* Vol. 19, No. 3, 15–21 1985.
- [23] Ward, G. J. Measuring and modeling anisotropic reflection. *ACM SIGGRAPH Computer Graphics* Vol. 26, No. 2, 265–272, 1992.
- [24] Ashikhmin, M.; Shirley, P. An anisotropic phong BRDF model. *Journal of Graphics Tools* Vol. 5, No. 2, 25–32, 2000.
- [25] Marschner, S. R.; Jensen, H. W.; Cammarano, M.; Worley, S.; Hanrahan, P. Light scattering from human hair fibers. *ACM Transactions on Graphics* Vol. 22, No. 3, 780–791. 2003.
- [26] Marschner, S. R.; Westin, S. H.; Arbre, A.; Moon, J. T. Measuring and modeling the appearance of finished wood. *ACM Transactions on Graphics* Vol. 24, No. 3, 727–734, 2005.
- [27] Irawan, P.; Marschner, S. Specular reflection from woven cloth. *ACM Transactions on Graphics* Vol. 31, No. 1, Article No. 11, 2012.
- [28] Sadeghi, I.; Bisker, O.; De Deken, J.; Jensen, H. W. A practical microcylinder appearance model for cloth rendering. *ACM Transactions on Graphics* Vol. 32, No. 2, Article No. 14, 2013.
- [29] Cook, R. L.; Torrance, K. E. A reflectance model for computer graphics. *ACM Transactions on Graphics* Vol. 1, No. 1, 7–24, 1982.
- [30] Torrance, K. E.; Sparrow, E. M. Theory for off-specular reflection from roughened surfaces. *Journal of the Optical Society of America* Vol. 57, No. 9, 1105–1112, 1967.



**Fu-Jun Luan** is a Ph.D. student in the Computer Science Department at Cornell University. He received his B.S. degree from Tsinghua University in 2015. He works on computer graphics with a focus on physically-based rendering and material appearance modeling.



**Li-Fan Wu** received his B.Eng. degree from Tsinghua University in 2015. He is a Ph.D. student of University of California, San Diego. His research interests include realistic rendering and image synthesis.



**Kun Xu** is an associate professor in the Department of Computer Science and Technology, Tsinghua University. He received his Ph.D. degree from Tsinghua University in 2009. His research interests include realistic rendering and image/video editing.

Other papers from this open access journal are available at no cost from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.