Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2012-034 November 16, 2012

5D Covariance Tracing for Efficient Defocus and Motion Blur

Laurent Belcour, Cyril Soler, Kartic Subr, Nicolas Holzschuch, and Fredo Durand

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

5D Covariance Tracing for Efficient Defocus and Motion Blur

LAURENT BELCOUR[1], CYRIL SOLER[2], KARTIC SUBR[3], NICOLAS HOLZSCHUCH[2], and FREDO DURAND[4]

[1]Grenoble University
[2]INRIA
[3]University College London
[4]MIT CSAIL

The rendering of effects such as motion blur and depth-of-field requires costly 5D integrals. We dramatically accelerate their computation through adaptive sampling and reconstruction based on the prediction of the anisotropy and bandwidth of the integrand. For this, we develop a new frequency analysis of the 5D temporal light-field, and show that first-order motion can be handled through simple changes of coordinates in 5D. We further introduce a compact representation of the spectrum using the covariance matrix and Gaussian approximations. We derive update equations for the 5×5 covariance matrices for each atomic light transport event, such as transport, occlusion, BRDF, texture, lens, and motion. The focus on atomic operations makes our work general, and removes the need for special-case formulas. We present a new rendering algorithm that computes 5D covariance matrices on the image plane by tracing paths through the scene, focusing on the single-bounce case. This allows us to reduce sampling rates when appropriate and perform reconstruction of images with complex depth-of-field and motion blur effects.

Categories and Subject Descriptors:

Additional Key Words and Phrases: global illumination, Fourier analysis, motion blur

1. INTRODUCTION

Photo-realistic effects such as depth of field and motion blur require heavy computation because they involve intricate integrals over a 5D domain composed of the image, lens, and time. A large number of samples is needed to avoid noise due to the variance of the integrand. However, the variation of the integrand is usually not arbitrary, and in particular, it can exhibit strong anisotropy in 5D because radiance often varies slowly along some directions. For example, an object with a rough BRDF only exhibits small variations along the angular direction. This can be expressed in terms of the Fourier spectrum of radiance: the frequency spectrum of the integrand often does not have energy in all five dimensions and, even in the directions where it has energy, it is often band-limited. Recent work has leveraged the frequency content of radiance for the faster rendering of individual effects such as depth of field [Soler et al. 2009], motion blur [Egan et al. 2009], soft shadows [Egan et al. 2011] and directional occlusion [Egan et al. 2011]. However, these solutions are limited in scope because the general derivation of spectrum prediction equations is hard. We introduce a new approach that can predict the frequency effect of most aspects of light transport in dynamic scenes in a unified manner. We present a new rendering technique that predicts and leverages these properties to reduce sampling rate and perform appropriate reconstruction for efficient high-quality rendering.

We predict the band-limited nature of radiance in the 5D domain of space, angle, and time. Key to our approach is to focus on individual local interactions such as occlusion, reflection, and motion. In particular, we show that any first-order motion can be handled using a 5D change of coordinates that expresses interactions in the static frame of the moving object. This allows us to handle moving light sources, occluders, and receivers in a unified manner. Because we focus on atomic operations, formulae are easy to derive for the full 5D spectrum. Arbitrary configurations and light paths can be handled by chaining operations together.

The storage and computation of a full 5D spectrum is, however, prohibitive. We argue that meaningful information about a spectrum are relative to the distribution of energy in the frequency domain. Information such as the principal directions of the spectrum and the extent of the spectrum along those directions (or bandwidth) are critical as they give us clues to the behavior of the function in the primal domain. Previous methods [Durand et al. 2005; Soler et al. 2009; Egan et al. 2009] tried to characterize the bandwidth of the signal. But some signals might have an infinite bandwidth while still having most of its energy concentrated in a finite portion of the domain. To permit a more compact and flexible characterization of the spectrum, we study its variance. We complete this characterization by using the correlation of the signal that gives its orientation in the frequency domain. All this information is gathered in the 5D *covariance matrix* of the signal in Fourier space. The key element of this matrix is that we are able to translate light transport operators on the spectrum into simple linear operators over the covariance matrix. We also show that this characterization inherently results in approximating the spectrum of the signal by a 5D Gaussian, that is possibly degenerate if the matrix is not full rank.

We present a new rendering algorithm that traces covariance matrices in the scene to predict the 5D spectrum at the image plane. We adapt the local 5D sampling rate locally in the image plane according to this prediction, and perform appropriate reconstruction using sheared 2D filters. Our technique can handle complex phenomena with depth of field and motion blur simultaneously. We focus on single-bounce paths (*i.e.* lens-object-light) but the technique has potential for arbitrary light paths.

Our paper makes the following contributions:

—A first-order 5D frequency analysis of light transport that handles the rotation and translation components of arbitrary motions using a simple change of coordinates.

—A compact characterization of the variation and anisotropy of the radiance field by the 5×5 covariance matrix of its spectrum, inspired by a Gaussian approximation.

—A set of equations for updating the 5D covariance matrix through local light interactions such as transport, occlusion, glossy reflection, texture mapping, lens integration, and object motion.

—A derivation of the sampling rate and of a 2D filter based on the 5D covariance matrix. The derived filters account for the blur induced by motion blur and depth-of-field.

—A rendering algorithm that is able to reconstruct effects such as depth of field and motion blur, by first tracing covariances and then performing adaptive sampling in 5D on the image plane.

1.1 Related work

Our work is related to sparse and low-rank approximations of light transport, frequency analysis of light transport, adaptive sampling methods, and methods that provide depth-of-field and motion blur.

1.1.1 *Sparse and low-rank approximations of light transport.* Existing techniques analyze the low-rank nature of the light transport operator in the context of pre-computed light transport. Mahajan et al. [2007] presented an experimental study of the dimensionality of light transport for cast shadows and other phenomena such as glossy reflection. Lessig and Fiume [2010] studied the rank of the transport operator and showed that in many situations the effective dimension of the operator is lower than that of the space it operates on. Pre-computed radiance transport has used this property extensively. The transport operator—rather than the signal itself—is compressed, making it efficient for computing many images. Dimensionality is usually reduced using principal component analysis [Sloan et al. 2003]. This *a posteriori* analysis contrasts with our method that is based on an *a priori* bandwidth prediction.

Matrix row-column sampling is another way of exploiting the low effective dimensionality of light transport in scenes illuminated by many light sources. The low-rank property of the transfer matrix is leveraged by clustering light source positions [Hašan et al. 2007]. This approximation is performed by selecting some of its rows and columns. Multidimensional Lightcuts [Walter et al. 2006], proceeds by clustering emitter and receiver points following a perceptual metric to adapt to the local dimensionality of light transport.

In Adaptive wavelet rendering [Overbeck et al. 2009] exploited the low-rank of light transport, by rendering directly in wavelet space, where the transport operator is sparse. This method exhibits low noise, since the reconstructed signal only contains the computed frequency bands, but operates in image space. Our work seeks to *predict* the smoothness of not only the image, but the full 5D integrand at each pixel. Application of compressive sensing [Sen and Darabi 2011a] to rendering allows to estimate the wavelet transform of the image during rendering using fewer coefficients.

1.1.2 *Frequency analysis of the transport operator.* We build on recent approaches that have studied the frequency aspects of light transport in static scenes, e.g. [Ramamoorthi and Hanrahan 2001; Durand et al. 2005; Ramamoorthi et al. 2005]. They presented *a priori* frequency analysis to perform adaptive sampling and appropriate reconstruction, reduce the number of samples needed for advanced rendering, and effectively share samples across pixels when possible. Egan et al. [2011] applied such an analysis and sheared reconstruction to shadow fields and directional occlusion [Egan et al. 2011]. To efficiently ray trace images with depth of field, Soler et al. [2009] proposed to adaptively sample primary rays by predicting image bandwidth and per-pixel variance of incoming light. They used a sampled representation of the spectrum which is both expensive and prone to noise. Instead, we use a covariance representation that is compact and stable. Egan et al. [2009] studied motion in 3D space-time defined in image coordinates. They focus on a set of specific cases because they derive end-to-end bandwidth formulae. In contrast, we address the full 5D case that includes spatial, angular, and temporal aspects in the scene, and focus on atomic operations to achieve generality.

1.1.3 *Adaptive sampling.* Our goal to use fewer samples where the integrand is smoother follows a long line of adaptive sampling methods, e.g. [Whitted 1980; Mitchell 1987; 1991; Kirk and Arvo 1991]. Multidimensional adaptive sampling [Hachisuka et al. 2008] adapts sampling in the full 5D domain, according to variance estimated from the samples themselves. They perform anisotropic reconstruction by analyzing a local structure tensor of the light field. Although we aim at optimally sampling as well, we inherently differ from this work in both the way we estimate the variance and in the way we estimate the reconstruction filter, both of which we do based on the predicted local frequency content of the illumination rather than *a posteriori* computation.

The first-order analysis of lighting, shading, and shadows [Ramamoorthi et al. 2007] permit gradient-based adaptive sampling. Derivatives convey less information than the raw shape of the spectrum. In particular, the first order analysis of the light field only provides a conservative measure of non-smoothness and does not capture anisotropy, which means that these techniques might be oversampling in some cases.

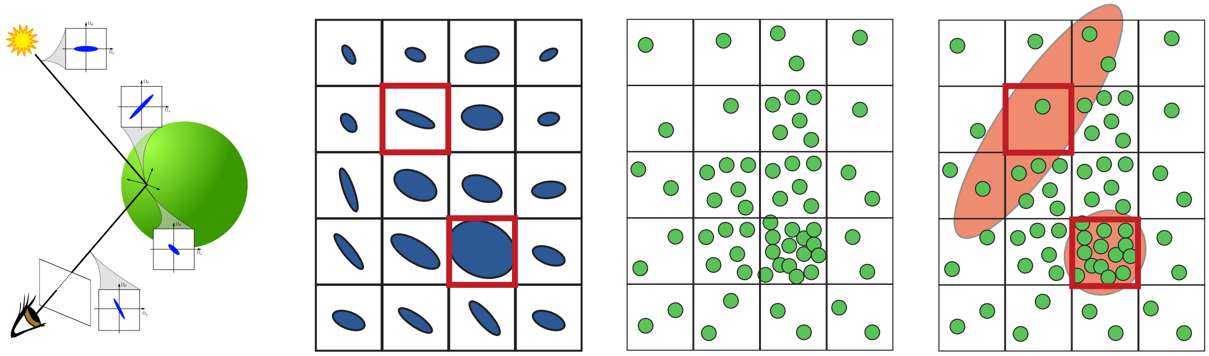
1.1.4 *Filtering and reconstruction.* Recently, Lehtinen et al. [2011] proposed a reconstruction method for motion blur and depth of field from sparse sampling of the 5D light field that uses speed and depth information to reproject samples into each pixel in order to obtain an approximate dense sampling. This method neglects the variation a sample might have along its reprojection line which might lead to artifacts when reconstructing glossy surfaces.

Sen and Darabi [2011b] proposed to filter the noise of Monte Carlo rendering by looking at the correlation between the variation of the samples value and the noise used to generate them. But this method cannot perform adaptive sampling since it studies only the correlation between noise and sample values.

2. OVERVIEW

Our algorithm adapts the number of 5D incident radiance samples, used in numerical integration, across the image pixels for the simulation of depth of field and motion blur. It then uses appropriate reconstruction filters to effectively share the radiance samples across pixels and provides high-quality images with reduced sample counts. Both the sampling rate and the reconstruction filters follow a new prediction of the local 5D frequency content. Central to our approach is an anisotropic Gaussian approximation of the 5D spectrum based on the covariance matrix for compactness and efficiency.

In a first pass (Fig. 1(a)), we trace a number of light paths and propagate the covariance matrix of the 5D spectrum in the neighborhood of each ray. At each interaction (e.g., reflection by a BRDF, transport, occlusion), we update this matrix according to new simple atomic operators. In particular, we handle dynamic



(a) Propagation of 5D covariance matrices

(b) Per pixel 5D covariance matrices

(c) 5D Samples are distributed on the image plane using the density derived from the 5D covariance matrix.

(d) 2D reconstruction

Fig. 1: *Overview of our proposed algorithm. (a) In a first step, we trace one-bounce light paths and propagate* 5×5 *covariance matrices from the light source to the camera. (b) The matrices are stored in image space and determine the sampling rate per pixel. (c) We compute the number of samples needed at each pixel and store the estimated incident radiance at each of them in a 2D buffer. (d) Finally, we render the image using 2D reconstruction kernels (red ellipses), designed using the covariances, to gather contributions of relevant radiance estimates.*

scenes with simple local changes of coordinates in 5D. This provides us with bandwidth and anisotropy information about the 5D spectrum reaching the lens for each pixel (Fig. 1(b)). Based on this spectrum approximation, we compute the required per-pixel 5D sampling rate following a variant of the Nyquist-Shannon theorem adapted to Monte Carlo integration.

In the second pass, we use the above sampling rates and compute the appropriate number of radiance samples for each pixel, sampling across the pixel area, the lens aperture and the shutter interval. This results in a number of 5D samples across the image (Fig. 1(c)).

We then determine, for each pixel, the correct 2D reconstruction filter (Fig. 1(d), in red), as a function of the stored 5D covariance matrices. Pixels that have a smaller spectrum (e.g. the upper left one) use a lower sampling rate and a correspondingly larger reconstruction filter, thereby sharing samples with neighbors. Pixels with higher frequency content (e.g. the lower-right one) use a higher sampling rate and a smaller reconstruction kernel. An important contribution of this paper is that we show that reconstruction can be performed in 2D alone, that is, the weights depend only on the x, y coordinates across the image plane, independent of the lens and time coordinates. This allows us to store and access the radiance samples in a 2D data structure rather than a 5D one.

This algorithm is made possible by a novel frequency analysis of time-varying light transport and a new Gaussian approximation of the 5D spectrum based on the covariance matrix. In particular, we show that maintaining the covariance matrix is simple for a variety of light interactions. Covariance is a powerful source of information to represent the distribution of energy in different directions in 5D and predict where the signal is strongly anisotropic and low bandwidth.

3. 5D FREQUENCY ANALYSIS OF LIGHT TRANSPORT

We extend the 4D local lightfield parametrization used by Durand et al [2005] to include time in the fifth dimension, for handling moving objects. We denote the local light field and its spectrum using $\ell(\mathbf{x}, \mathbf{\Theta}, t)$ and $\hat{\ell}(\omega_{\mathbf{x}}, \omega_{\mathbf{\Theta}}, \omega_t)$ respectively where $\mathbf{x}$ denotes (x, y) — that span the plane orthogonal to the direction of propagation — and $\mathbf{\Theta}$ denotes (θ, ϕ) — angles measured from x and y respectively.

We analyze the 5D spectrum of the local temporal light field $\ell(x, y, \theta, \phi, t)$ in the neighborhood of a ray as it propagates through the scene. We build on the 4D analysis by Durand et al. [2005] for static interactions and add the time dimension to handle motion. We first show that interactions such as occlusion, transport, and BRDF are similar to their static counterparts, except for the fifth dimension, and that the incoming light field potentially contains temporal energy. We then derive the case with motion and show that a simple change of coordinates suffices, which corresponds to shears in space-time and angle-time.

3.1 Static 4D interaction

At the temporal scale we are interested in, light interacts instantaneously. Effects such as occlusion, transport, and reflection by a BRDF only concern light rays that share the same temporal coordinate t. They can be treated independently for different 4D time slices of the temporal light field. This means that we can apply the formulae derived by Durand et al. [2005] for the 4D static components of our local temporal light fields.

The 4D static operators on the spectrum involve shear, convolution, and multiplication. Table I summarizes the transformations on local light field spectra in static scenes, due to transport processes.

4D convolutions in the primal, such as those required for shading, become 5D convolutions with a kernel that has the same 4D component as before and is infinitely thin along time, since shading only happens with rays at the same instant. In the Fourier domain, this means that the 5D spectrum of the BRDF is constant along the time frequency dimension, assuming that the BRDF is not time-varying.

Multiplications in the primal are necessary for textures and occlusion. If the texture is constant across time, this means that its Fourier transform is a Dirac along that direction. Time-varying textures can be handled by computing their 3D FFT. Occlusion by blockers that are locally static also corresponds to a convolution by a spectrum that is the 4D spectrum of blockers times a Dirac in time. Moving occluders are handled using the change of coordinates described in Section 3.2.

Lens shears. As we want to perform lens integration to obtain depth-of-field, we need to characterize a lens operator for the frequency analysis theory. Previously, Soler et al. [2009] proposed such an operator but they looked at integration of the light field by the entire lens. Instead, we would like to obtain the non-integrated light field on the sensor to achieve the depth-of-field effect afterwards.

We add another operation for static light fields to handle the effect of a small thin lens. We build on results from paraxial optics [Gerrard and Burch 1975] to derive this operation. We assume that the lens is small enough so that ray coming to the camera can be parametrized with respect to the center of the lens and that the angle between the central ray direction and the sensor-lens axis is small enough. This hypothesis restricts our analysis to lenses without fish-eye effects. Our theory would handle such lenses at the expense of additional projections onto the lens' surface and sensor.

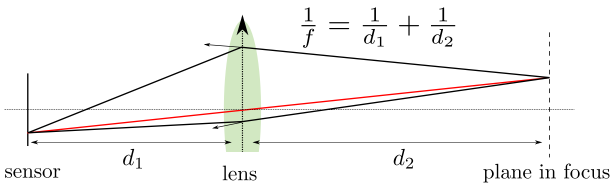


Fig. 2: *At the lens, the incoming rays undergo a shift as well as a shear due to lens refraction. Finally, there is transport through free space to the sensor.*

Given an incoming local light-field at the lens $l(\mathbf{x}, \mathbf{\Theta}, t)$ we want to characterize the local light-field at the sensor position oriented along the central direction of the sensor and lens $l'(\mathbf{x}, \mathbf{\Theta}, t)$. First, when the light field passes throught the lens, its direction is changed due to refraction, by the two interfaces it crosses. Since the lens is assumed to be thin, travel can be neglected between the two interfaces and the curvature shear caused by the interfaces produces a cumulated curvature shear of parameter $\kappa = 1/f$, where f is the focal length of the lens. Please refer to Gerrard and Burch [1975], Chapter II.4.1, for more details. Then, the light leaves the lens and travels to the sensor, making the light field in or out of focus depending on the spatial position:

$$\ell'(\mathbf{x}, \mathbf{\Theta}, t) = \ell\big(\mathbf{x} - d_1(\mathbf{\Theta} + \frac{1}{f}\mathbf{x}), \mathbf{\Theta} + \frac{1}{f}\mathbf{x}, t\big) \qquad (1)$$

where f is the focal length of the lens, and d_1 is the distance from the outgoing central position on the lens to the sensor position and d_2 is the distance of the plane in focus from the lens. Figure 2 presents notations for the different quantities involved.

Example. Given a point in focus (with coordinate $[\mathbf{x}, \mathbf{\Theta}]$ in the local light-field), the local coordinates at the lens will be:

$$\begin{bmatrix} \mathbf{x}_l \\ \mathbf{\Theta}_l \end{bmatrix} = \begin{bmatrix} \mathbf{x} - d_2\mathbf{\Theta} \\ \mathbf{\Theta} \end{bmatrix} \qquad (2)$$

Using Equation 1 and Equation 2, we can write the spatial component at the sensor as:

$$\begin{aligned} \mathbf{x}_s &= \mathbf{x} - d_2\mathbf{\Theta} - d_1\big(\mathbf{\Theta} + \frac{1}{f}(x - d_2\mathbf{\Theta})\big) \\ &= \mathbf{x}(1 + \frac{1}{f}) \qquad (3) \end{aligned}$$

The influence of the angle $\mathbf{\Theta}$ on the final spatial component of the local light-field at the sensor cancels out, indicating that the point is indeed in focus.

Transport process	**Transformation to spectrum**
Travel (free space)	Angular shear proportional to distance d
Occlusion	Spatial convolution with blocker spectrum β
Reflection	Angular bandlimiting by reflectance spectrum ρ
Curvature	Space-angle shear
Texture	Convolution with texture spectrum
Lens	Space-angle shear, angle-space shear and phase shift

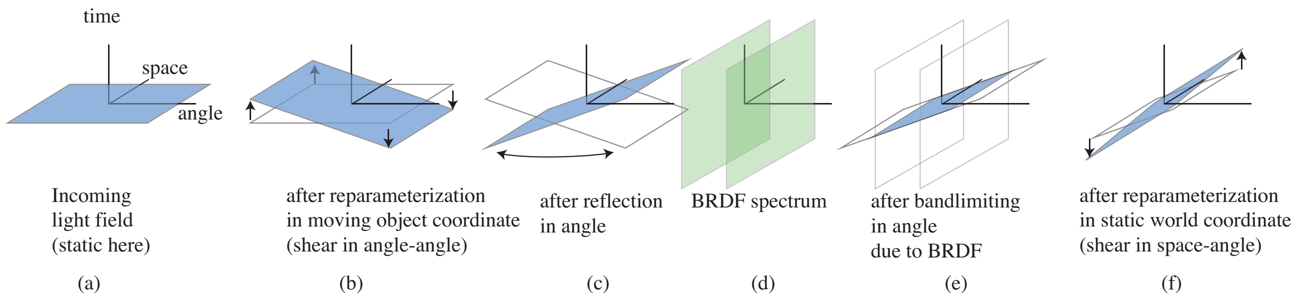
Table I. : *Summary of transformations to spectra in static scenes. These are the natural extension of the 4D operators of Durand et al. [2005] and our new lens operator with a constant in the time dimension.*

3.2 Change of coordinates for motion

Motion in the scene can result in complex interactions. Imagine a deforming reflector receiving light from a rotating light source, partially blocked by a translating occluder. Previous approaches such as Egan et al. [2009] have sought to derive end-to-end equations and have necessitated special cases for different configurations. Another option would be to extend local operators such as BRDF shading, taking into account the motion of the receiver and other elements, and how it affects normals and other aspects. This would unfortunately lead to complex formulae, not unlike full end-to-end approaches.

Instead, we simplify the handling of motion by reducing the problem to simple local transformations in the coordinate system of each moving object. The incoming temporal light field potentially has complex temporal components, but this is not a problem because the transformation itself is simple.

In order to treat each interaction locally as static, we change the parametrization of the light field to the coordinate system of the local moving object. This follows the traditional Galilean principle that all motion is relative. For illustration, consider the reflection of a fixed light source by a rotating, glossy plane. We show the corresponding spectrum operations in Figure 3. The incoming light field spectrum has no temporal component because the light is static (Figure 3(a)). We reparametrize the incoming temporal light field in the static frame of the moving reflector (Figure 3(b)). In this new coordinate system, the light field spectrum has temporal energy, corresponding to the opposite motion of the reflector (angle-time shear). We then handle the reflection using static operators [Durand et al. 2005], including incidence angle, cosine term, curvature handling, and BRDF bandlimiting (Figure 3(c-e)). This



Incoming light field (static here)

(a)

after reparameterization in moving object coordinate (shear in angle-angle)

(b)

after reflection in angle

(c)

BRDF spectrum

(d)

after bandlimiting in angle due to BRDF

(e)

after reparameterization in static world coordinate (shear in space-angle)

(f)

Fig. 3: *Frequency analysis of BRDF shading for a rotating glossy plane. We model shading as a static operation (c-e) in the coordinate system of the moving receiver. The changes of coordinates occur in (b) and (f). The spectrum of the static incoming light (a) has a temporal component (b) in the coordinate system of the moving receiver.*

provides us with the outgoing temporal light field in the frame of the moving object. We finally perform another change of coordinate (angle-time shear with the opposite direction for motion) to express it in the static world frame (Figure 3(f)).

We have discussed a static incoming light field for illustration purposes only, and an incoming light field with a complex temporal spectrum is treated the same way through a change of coordinates. Other types of motion and interactions are handled similarly, expressing the local temporal light field in the frame of a moving light source or blocker.

Equations for the coordinate changes. We consider a first-order approximation of motion at the location of the central ray. Let the object have translational velocity $\mathbf{v} = (v_x, v_y, v_z)$ and angular velocity $r = (r_\theta, r_\phi, r_\psi)$. Velocities are expressed in the local frame of the incident temporal light field with rotations corresponding to the x and y axis of the plane parameterizing our local light field, as well as the rotation by r_ψ around the central ray.

The linearization of motion leads the to simple formula:

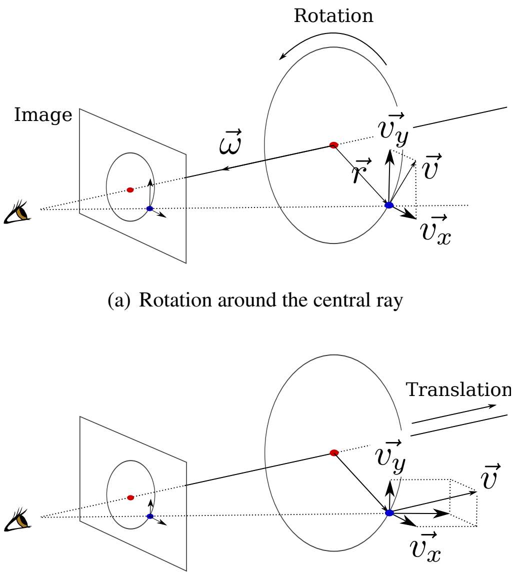
$$\ell'(\mathbf{x}_{st}, \mathbf{\Theta}_{st}, t) = \ell(\mathbf{x} - (v_x, v_y)t, \mathbf{\Theta} - (r_\theta, r_\phi)t, t) \qquad (4)$$

where $\mathbf{x}_{st}$ and $\mathbf{\Theta}_{st}$ denote the spatial and anglular coordinates in the moving object's local frame respectively. A derivation of this property can be found in Appendix A . The equation reveals that the local light field undergoes a shear in space-time due to translation and in angle-time due to local rotation. We define the corresponding transformation in the Fourier domain using an operator, $\mathcal{G}$.

$$\mathcal{G}(\hat{\ell}) \;\equiv\; \hat{\ell}(\omega_{\mathbf{x}}, \omega_{\mathbf{\Theta}}, \omega_t + \mathbf{v}\omega_{\mathbf{x}} + r\omega_{\mathbf{\Theta}}) \qquad (5)$$

$\mathcal{G}$ is a forward motion operator that expresses the light field spectrum in the local frame of the moving object.

The z component of the velocity and the in-plane rotation r_ψ do not affect our first-order analysis. This is because they only create second- order effects that multiply different variables such as angle and time for the in-plane rotation. Our method can, however, handle objects that globally rotate. Consider the case of a textured disk that rotates around its center (Figure 4(a)). The local blur is tangential and proportional to the distance to the center. This means that there is no blur at the exact center of the disk. If we move away from the center, the local motion considered by our technique is re-centered to the point of interest, and includes both a (v_x, v_y) translation component and an in-plane rotational component r_ψ. The local blur is mostly caused by the translational component, as predicted by our first-order model. A similar situation occurs when an object is moving towards a viewer. In the center, there is no blur, but away from it, blur can be analyzed as caused by a (v_x, v_y) translation since the ray going from the viewpoint to the plane is not orthogonal anymore (see Figure 4(b)).



(a) Rotation around the central ray

(b) Translation along the central ray

Fig. 4: *(a) We only need to account for tangential rotation since the radial component can be considered as translatory motion in a first order analysis. (b) Similarly, motion along the ray direction is approximated by a translation in the tangent plane.*

The static operators are applied on $\mathcal{G}(\hat{\ell})$ and the result is then transformed back into the global static frame using a reverse time operator, to obtain the transformed temporal local light field. The reverse time operator is essentially the same transform as $\mathcal{G}$, but with negative translational and rotational velocities, and is therefore equal to its inverse $\mathcal{G}^{-1}$.

In the Fourier domain, the shears are time-space and time-angle for translation and rotation respectively. Previous work on motion blur [Levin et al. 2008; Egan et al. 2009] also used shears, but in the 3D time space defined by image coordinates, and based on 2D translational velocity. In contrast, our transformation is in the 5D space of temporal light fields and exhibits both translation and rotation.

3.3 Time-varying signals

Some of the signals involved might have built-in temporal content, for example, a movie projector or a TV. The frequency content of such source signals can be directly incorporated by considering their discrete Fourier Transform.

4. COVARIANCE MATRICES AND GAUSSIAN APPROXIMATION

In this section, we describe a compact formulation to track the extent of local variation and anisotropy of temporal light-fields in 5D space: the *covariance matrix*.

We compute the 5D covariance matrix of the local light field's Fourier amplitude spectrum, and use it to measure local variation and anisotropy. We define this matrix Σ such that the entry at the i^{th} row and j^{th} column is:

$$\Sigma_{i,j} \equiv \int\limits_{\mathbf{z}\in\Omega} \langle \mathbf{z}, \mathbf{e}_i \rangle \, \langle \mathbf{z}, \mathbf{e}_j \rangle \, \left| \hat{\ell}(\mathbf{z}) \right| \, \mathrm{d}\mathbf{z}. \tag{6}$$

Here $\langle ., . \rangle$ denotes an inner product, $\mathbf{e_i}$ and $\mathbf{e_j}$ are members of the canonical basis of the Fourier domain $\Omega = \Omega_x \times \Omega_y \times \Omega_t \times \Omega_p \times \Omega_t$, and $\left| \hat{\ell}(\mathbf{z}) \right|$ is the amplitude of the local light field spectrum. The diagonal elements in Σ represent the variances of the amplitude of the spectrum along each dimension and $\Sigma_{i,j}$, $i \neq j$ represent the covariances between the i^{th} and j^{th} dimensions. The covariance of the spectrum [1] conveys important information such as the extent of variation and anisotropy.

A conceptually simpler interpretation is that *we approximate the amplitude spectrum of the local light field using a 5D Gaussian* defined entirely by the above covariance matrix: $g(\mathbf{z}) = e^{-\mathbf{z}^T \Sigma_g^{-1} \mathbf{z}}$, with $\Sigma_g(i,j) = \mathbf{e_j}^T \Sigma \mathbf{e_i}$ .This representation is close to Heckbert's elliptical Gaussian filters [Heckbert 1989].

Some transport operators clamp energy in subspaces of the spectrum, causing the covariance matrix to contain a few null Eigenvalues. The Gaussian is then degenerate, and the concept less elegantly fits our purpose. Nevertheless, the equivalence between covariance and (possibly degenerate) Gaussians will be useful to derive some formulae, particularly during convolution of spectra.

In the remainder of this section, we explain how the 5D covariance representation of the light field spectra can be updated through transport processes such as motion, free space transport, occlusion and reflection using simple matrix operations.

4.1 Background

A convenient property of covariance matrices is that updating the covariance matrix of a signal when the domain is transformed by a linear operator is straightforward: Let Σ be the covariance matrix of the signal, and Σ' the covariance matrix of the signal after the transformation of its domain by a 5D matrix M. We have:

$$\Sigma' = |M| M^T \Sigma M \tag{7}$$

We provide a proof of this property in Appendix B. We will use this property to update the covariance matrix after changes of coordinates due to general motion, reflection, and transport through free space.

We also need to be able to handle convolutions for occlusion and reflection (texture as well as cosine term). Although there is no general formula, for zero-centered Gaussians, convolution is equivalent to summing up the covariance matrices. That is, given two Gaussians with covariance matrices Σ_1 and Σ_2, the covariance matrix of their convolution is:

$$\Sigma_{1 \otimes 2} = \Sigma_1 + \Sigma_2 \tag{8}$$

We adopt this as an approximation of the covariance matrix of the convolution of two spectra.

The covariance of the sum of spectra also has a simple expression: since spectra are zero-centered, the covariance of the sum is the sum of the covariances. If spectrum h is defined as $h(\mathbf{x}) = \alpha f(\mathbf{x}) + \beta g(\mathbf{x})$, then the covariance matrix of h is:

$$\Sigma_h = \alpha \Sigma_f + \beta \Sigma_g \tag{9}$$

4.2 Motion

Object motion is handled identically for all processes – light sources, occlusion and reflection – through a change of coordinates.

Let Σ be the covariance of the incident local temporal light field. Let G be the matrix operator transforming the local light field in the static coordinate system of the moving object. According to Equation 7 the covariance of the signal in the moving frame is (See 0**??**)

$$\Sigma_{\mathrm{st}} = G^T \Sigma G \tag{10}$$

Next we apply all necessary transformations that happen in the moving frame (e.g. reflectance) updating Σ_{st} to Σ'_{st}. Finally we convert back the covariance into the world coordinate system by transforming Σ'_{st} back into the global static frame:

$$\Sigma' = (G^{-1})^T \Sigma'_{\mathrm{st}} G^{-1} \tag{11}$$

4.3 Transport through free space

The local light field spectrum undergoes a shear in angle during transport of distance d through free space. This is a linear transform of the space by a matrix T_d defined in Table II. Using Equation 7, the covariance Σ' after transport is

$$\Sigma' = T_d^T \; \Sigma \; T_d \tag{12}$$

4.4 Occlusion

The spectrum resulting from occlusion is the convolution of the incident spectrum and the blocker's spectrum. The convolution of two Gaussians is given in Equation 8 and we estimate the resulting covariance matrix by summing the incoming covariance matrix

[1] Since we approximate the local light field, Equation 6 adopts the definition of covariance of functions that is used to study the relationship along the different input dimensions. This differs from the more common usage of covariance is to study relationships between random variates.

(a) Time transform matrix

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ v_x & v_y & r_\theta & r_\phi & 1 \end{bmatrix}$$

(b) Travel matrix

$$T_d = \begin{bmatrix} 1 & 0 & -d & 0 & 0 \\ 0 & 1 & 0 & -d & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(c) Scale matrix

$$S_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(d) Curvature matrix

$$C_\eta = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -\eta_x & 0 & 1 & 0 & 0 \\ 0 & -\eta_y & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(e) Symmetry matrix

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(f) Lens matrix

$$L_{f,d} = \begin{bmatrix} 1 & 0 & -d & 0 & 0 \\ 0 & 1 & 0 & -d & 0 \\ \frac{1}{f} & 0 & 1 & 0 & 0 \\ 0 & \frac{1}{f} & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(g) Occlusion matrix

$$\Sigma_B = \begin{bmatrix} O_{x,x} & O_{x,y} & 0 & 0 & 0 \\ O_{x,y} & O_{y,y} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(h) Cosine term matrix

$$\Sigma_C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C & 0 & 0 \\ 0 & 0 & 0 & C & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Table II. : *Linear time-transformation matrices (a) G and G^{-1} (corresponding to $-v_x, -v_y, -r_\theta, -r_\phi$) perform transformations into and out of the static coordinate frame;(b) T_d performs an angular shear due to travel in free space; (c) S_α performs a spatial scale due to the incoming or outgoing cosines; (d) $C_\eta$is a spatial shear that accounts for local curvature during reparametrization for reflection; (e) M performs the symmetry of the signal to express it in the reflected frame; (f) $L_{f,d}$ performs the camera transformation. Non-linear transformation matrices: (g) Σ_B is the occlusion matrix that represent the transform of the blocker visibility function; (h) Σ_C is the differential cosine irradiance matrix.*

and the covariance of the blocker spectrum. The transformed covariance due to occlusion, Σ'_{st}, is the sum of the covariance before occlusion, Σ_{st}, and the covariance of the blocker spectrum, Σ_B:

$$\Sigma'_{\text{st}} = \Sigma_{\text{st}} + \Sigma_B \qquad (13)$$

4.5 Reflection

Shading is performed as a composition of the transformations due to the following processes [Durand et al. 2005], which we further describe below: a foreshortening (and alignment) along the incident ray; a spatial shear due to curvature; a mirror reflection; a convolution due to the irradiance cosine term; a multiplication by the spectrum of the BRDF; a second curvature shear; and finally a different foreshortening along the reflected ray.

Foreshortening for incident ray. To handle reflections, we need to project our light field into the local frame of the object. This process is also done when we project the local light field on the outgoing direction. Those two projections are handled by combining a rotation R_z around the local direction to align the local x of the local light field with the object's surface, and a scale along the y direction by the inverse cosine of the angle θ between the incoming direction and the normal $\mathbf{n}$. Later on, the covariance matrix might need to be further rotated around the normal, to match the coordinate system of the next component such as texture, curvature or BRDF. Notations are summarized in Figure 5.

The scale S_α is anisotropic in space since the foreshortening is purely in the plane containing the normal and incident (or reflected) ray. We have:

$$\Sigma'_{\text{st}} = S_\alpha^T\, R_z^T\, \Sigma_{\text{st}}\, R_z\, S_\alpha \qquad (14)$$

where $\alpha = 1/\cos\theta$ is the scaling factor, θ is the incident (or reflected) angle. Matrices R_z and S_α are shown in Table II.

Curvature. The effect of local curvature is a spatial shear of the local light field spectrum and remains the same for the covariance.

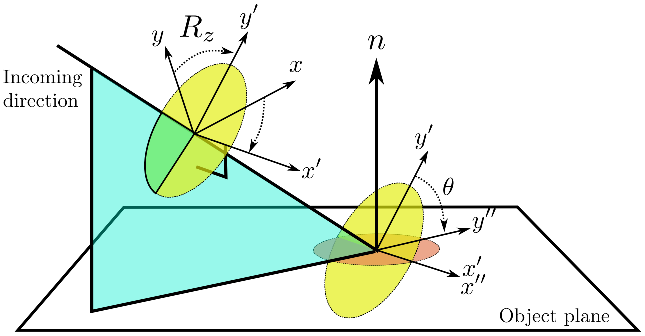


Fig. 5: *Successive transformations to convert the incoming light field (with axes x, y), into the frame of the object (with axes x'', y''): a rotation R_z to align x with the object's plane and a scale along y' by $\frac{1}{\cos\alpha}$.*

Let R_η be the rotation matrix that aligns the frame of the light field with the directions of principal curvature, we have:

$$\Sigma'_{\text{st}} = C_\eta^T\, R_\eta^T\, \Sigma_{\text{st}}\, R_\eta\, C_\eta \qquad (15)$$

where C_η is shown in Table II, and (η_x, η_y) represents the principal eigenvalues of the local curvature tensor.

Mirror reflection. The expression of the local light field in the reflected direction implies a reflection along the angular dimensions. The angular covariance terms are therefore negated, leaving the angular variance terms untouched:

$$\Sigma'_{\text{st}} = M^T\, \Sigma_{\text{st}}\, M \qquad (16)$$

where M is shown in Table II.

Cosine term. Before shading, the incoming spectrum is convolved with the spectrum of the clamped cosines of incident directions. Following Equation 8, the corresponding operation on the covariance is an addition of the covariance matrices of the spectrum of the clamped cosine and that of the light field's spectrum:

$$\Sigma'_{\mathrm{st}} = \Sigma_{\mathrm{st}} + \Sigma_C \tag{17}$$

where Σ_C is shown in Table II.

Shading. Shading by the BRDF is a convolution in the primal, and an angular product of the spectrum with the spectrum of the BRDF in the Fourier domain.

When we consider time- and space-invariant BRDFs, their spectrum only has energy along the two angular dimensions. In the primal domain, they have the shape of a 2D Gaussian multiplied by Diracs in the other three dimensions. The covariance matrix of the spectrum is therefore not invertible, although well defined.

In the scalar case, the multiplication of zero-mean Gaussians results in a Gaussian whose covariance is the harmonic mean of the inputs. In the 5D case, we need to do a summation of the pseudo-inverse of the covariance matrix by the pseudo-inverse of the BRDF frequency spectrum, and invert the result:

$$\Sigma'_{st} = (B^{+} + \Sigma_{st}^{-1})^{-1} \tag{18}$$

where B^{+} is the pseudo inverse of the covariance matrix of the BRDF's spectrum. We give a full proof of this in appendix C.

Texture. The effect of texture is to add spatial frequencies. Specifically, the transformed spectrum is a convolution of the incident spectrum with the local spatial spectrum of the texture. As with occlusion, we sum the incoming spectrum's covariance and the covariance of the texture Σ_T, after possibly rotating it into the same coordinate system using a rotation R_t:

$$\Sigma'_{\mathrm{st}} = R_t^T \Sigma_{\mathrm{st}} R_t + \Sigma_T \tag{19}$$

4.6 Lens

We described the lens operation in terms of local light-fields in Section 3.1. The operator being a linear transform of the space, the operation on the covariance matrix becomes:

$$\Sigma'_{\mathrm{st}} = L_{f,d}^T \Sigma_{\mathrm{st}} L_{f,d} \tag{20}$$

Where the matrix L is described in the Appendix 0**??**.

4.7 Putting it all together

To illustrate the power of our theory, we carry out the analysis of an intriguing case. Consider a spherical mirror rotating around one of its axes ω. It is well known that, despite the motion, the reflection remains perfectly sharp even for finite exposure times. Indeed, our theory predicts the same.

Consider the transformations undergone by the incident light field through reflection at an arbitrary point x on the sphere of radius r (See notations on Figure 6). The incident spectrum is first represented in the local tangent frame. This amounts to a rotation about the incident ray and a spatial scale by the incident cosine. Let Σ denote the covariance of the incident spectrum in the local tangent frame at the starting time of exposure $t = 0$. This transformation is independent of the exposure.

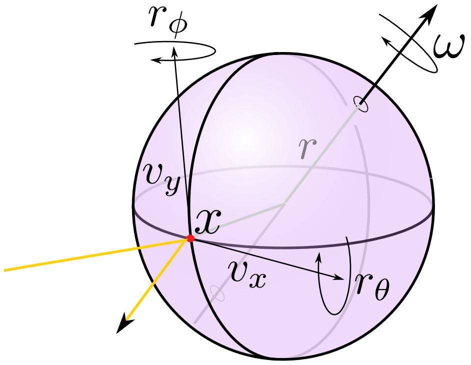


Fig. 6: *Notations for the rotating sphere example. A mirror sphere rotates around an axis ω. Our model correctly predicts the zero angular-time covariance of the light reflected at point x. $v_x, v_y r_\theta$ and $r\phi$ are respectively the spatial and angular velocities, in the local coordinate system at point x.*

The next step involves transforming Σ into the moving object's static coordinates, using G, where reflection can be treated as an instantaneous process. Following this, reflection involves a transformation due to curvature C_η (with $\eta = 1/r$), reparametrization in the direction of mirror reflection M, and a multiplication by the BRDF spectrum (constant for specular reflection). After reflection, the inverse transformations are applied for curvature and motion.

The resulting reflected covariance is given as $\Sigma' = R^T \Sigma R$ where:

$$R = G^{-1}\ C_\eta\ M\ C_{-\eta}\ G = \begin{bmatrix} 1 & 0 & -2\eta & 0 & 0 \\ 0 & 1 & 0 & -2\eta & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -2(r_\theta - \eta v_x) & -2(r_\phi - \eta v_y) & 1 \end{bmatrix}$$

v_x and v_y are the translational velocities and r_θ and r_ϕ are the angular velocities of the point of reflection. The time-angle covariance terms disappear if $r_\theta = \eta v_x$ and $r_\phi = \eta v_y$, which is indeed the case for all points on the sphere! This confirms that the reflection does not vary over time and remains sharp.

In the presence of texture on the sphere, since we add the covariance of the texture during reflection, the reflection contains time-space coefficients corresponding to the covariance of the texture. That is, for a finite exposure, a spinning sphere with texture is predicted to be blurry as expected.

5. TRACING COVARIANCE

In this section, we use the already derived update equations to estimate the local 5D covariance after integration on the image plane. For each pixel, we sample light paths (Section 5.1), and estimate the covariance matrix at the sensor by applying the operations on covariance matrices from the light to the camera (Section 5.2). We pay particular attention to handling occlusion (Section 5.3) since care is needed to determine when the local light field is partially occluded. Other update operations are straightforward. Finally we show how to accummulate covariance matrices (Section 5.4) to obtain an estimate of the matrix at each pixel.

5.1 Path tracing covariance to the camera

For each pixel, we trace a small number of rays (typically 10 to 100 depending on the scene's lighting complexity) to estimate covariance. Rays are cast from different lens locations and at different times and the resulting hit positions in the scene are connected to one of the light sources. Given a path, we propagate a covariance matrix from the light source to the lens. We initialize the matrix with the covariance of the light and update it according to the operators derived in Section 4.

Since the covariance matrix is symmetric, we only have to update 15 floating point values at each operation. We also keep track of the orientation of the tangent plane (two unit vectors) to be able to handle anisotropy.

5.2 Covariance of light sources

Diffuse area light sources have no variation in angle, and their spatial covariance depends on their size. For example, at rectangular light sources, we align the tangent plane with the principal direction of the light source and use the following formulae:

$$\Sigma_{1,1} = 2\,(\pi/s_x)^2,\;\; \Sigma_{1,2} = 0,\;\; \Sigma_{2,2} = 2\,(\pi/s_y)^2$$

where s_x, s_y are the sizes of the light along the principal directions. For environment maps, we compute the local angular covariance using a windowed Fourier transform on the 2D map of angles. Spotlights have angular content and the angular variance is inversely proportional to the half-angle of spread. We have not implemented light sources with temporally varying emissivities, such as video projectors, but they can be included by computing the DFT of their temporal emission. We do, however, handle light sources that are in motion. For moving light sources, we apply the motion matrix defined in Equation 5 after computing the covariance of the light source.

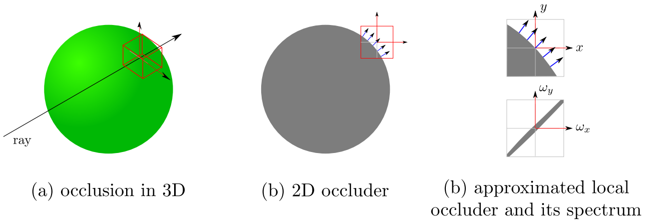
5.3 Occlusion

Partial occlusion of the local light field potentially introduces energy in the high frequencies of the magnitude spectrum. To account for this, we developed a volumetric data structure that stores local normal distributions to answer two queries: (a) does an occluder exists in the neighborhood of a path; and (b) if yes, what anisotropy does it introduce in the local light field spectrum with reference to its 2D local spatial frame.

Occlusion due to blockers is characterized, at a given location, by a 2D binary function along the spatial dimensions. The frequency content of this 2D function is captured by the silhouettes of the blockers, which can be deduced from the surface normals that are orthogonal to the direction of propagation (See Figure 7). We exploit this observation by storing 3D normal distributions, and using them to estimate occlusion caused on a local light field by extracting normal distributions in the plane orthogonal to the light path.

We use a regular 3D grid, where each voxel contains a 3 by 3 covariance matrix of the distribution of surface normals inside the corresponding voxel. This 3D covariance matrix is built by uniformly sampling normals on objects and accumulating 3D covariance contributions for each normal inside the voxel.

When a ray crosses a voxel "near" an obstacle we add the covariance matrix of the occluder to the covariance matrix of the spectrum



(a) occlusion in 3D (b) 2D occluder (b) approximated local occluder and its spectrum

Fig. 7: *When an occlusion is detected during ray tracing (a), we need to find an approximated equivalent 2D occluder (b), and compute its local spectrum (c). For that, we use the local distribution of normals of the occluder near the occlusion point in the tangent plane.*

(Equation 13). First, we rotate the voxel's 3D matrix to align it with the frame of the ray. Then, from this rotated matrix, we extract the 2D $x - y$ sub-matrix corresponding to the tangent frame of the ray. This 2D sub-matrix is the covariance matrix of normals projected to the $x - y$ plane of the local light field of the ray. We convert this 2D matrix into a 5D covariance matrix of the blocker, by padding other dimensions with zeros (See Figure 8).

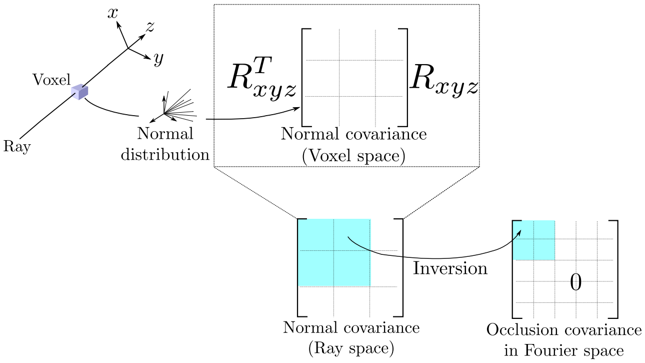

x
z
y
Voxel
Ray
Normal distribution
R_{xyz}^T
R_{xyz}
Normal covariance (Voxel space)
Inversion
0
Normal covariance (Ray space)
Occlusion covariance in Fourier space

Fig. 8: *We use a voxel grid to hand occlusion, and store normal distributions at each voxel. This information is used to estimate the local 2D spectrum of obstacles as they partially occlude a the local light field at points along the central light ray. The stored covariance matrix (of the local normal distribution) is first rotated to align it with the local frame of the ray and then the appropriate 2D sub-matrix (corresponding to the tangent frame of the ray) is projected to the XY plane of the local light field.*

If the normal distribution at each voxel is filtered to obtain a single normal direction, the resulting 5D occlusion matrix is equivalent to the half plane occlusion approximation used in Soler et al. [2009]. Furthermore, our approach enables the encoding of directional information for occlusion while seamlessly avoiding artifacts due to unwanted occlusions at either end of the ray. At the ends, where the distribution of normals is close to the normal at the intersection points, the occlusion covariance matrices automatically become null.

We handle moving occluders conservatively by filling the grid with positions of the moving object, sampled during the shutter interval. This conservative estimate of the occlusion through time allows us to store the same amount of data as for the static case.

5.4 Aggregating covariance due to multiple paths

Each light path provides us with a 5D covariance matrix that characterizes the local frequency spectrum. To obtain the combined covariance estimate at a given pixel position, we accumulate the various contributions. Since the covariance of two zero-centered functions is additive (Equation 9), we can build a Monte-Carlo estimate of the covariance matrix from the different samples Σ_i acquired at pixel p. We use this estimate of the covariance matrix to estimate the density and filter per pixel.

$$\Sigma \simeq \sum_{i \in p} \frac{E_i}{E} \Sigma_i \tag{21}$$

Where E_i is the radiance carried by sample i, E is the total radiance carried by samples for the current pixel.

6. RENDERING

Our rendering algorithm involves four steps as illustrated in Figure 1:

(1) We accumulate covariance per pixel by tracing a number of covariance light-paths using the theory described in Section 5.

(2) We compute, at each pixel, the required 5D sampling density, as well as a 2D reconstruction filter (Section 6.1).

(3) For each pixel, we sample its 5D domain with the determined sampling rate and store these 5D samples into a 2D xy-grid (Section 6.2).

(4) For each pixel, we sum the weighted contribution of all samples inside the 2D reconstruction filter, possibly using samples from nearby pixels (Section 6.3).

We provide an illustrative example in Figure 0**??** where the method is applied to a sphere lit by an environment map.

6.1 Estimating sampling density and reconstruction filters

2D bandwidth. In contrast to Egan et al. [2009], we perform reconstruction in the 2D image domain rather than the full 5D light field. This is made possible by a new derivation that considers the effect of the lens aperture and exposure interval as a multiplication followed by an integration, rather than a convolution. Our derivation might result in slightly higher sampling rates, but it alleviates the need to sample outside the shutter interval or lens. Furthermore, our approach avoids the need for higher-dimensional acceleration structures for efficient sample queries.

We first model the effect of integration over a finite shutter interval and lens aperture before considering sampling. The incoming radiance (Figure 9 (a)) is multiplied (windowed) by the shutter function (b), which corresponds to a convolution in the Fourier domain (c). Depth of field and motion blur result from the integral of this windowed radiance along time and aperture, which corresponds to slicing in the Fourier domain (d). The resulting 2D slice is the Fourier transform of the final image. In the case of motion blur, a longer shutter speed corresponds to a smaller kernel (b), and we get the intuitive behavior where longer shutter speeds lead to less high frequencies for moving parts in the final image. This 2D bandwidth is the key to deriving 2D reconstruction filters.

Sampling and reconstruction. We seek to compute the illumination for each pixel by sampling radiance in 5D (in the space of lens, image and time) and averaging these samples using an appropriate reconstruction kernel. Following Shannon, sampling creates replicas in the Fourier domain (Figure 9 (e)). The goal of accurate sampling is to prevent replicas of the spectrum from overlapping the useful signal. that characterizes the final image, as shown by Figure 9 (d)

We need to pack the Gaussian functions, which we approximate with ellipsoids of half axes lengths given by the matrix eigenvalues' square root. Critical sampling is obtained when the sampling distance along the eigenvectors is proportional to the square root of the eigenvalues' product (the product of sigmas), as illustrated in Figure 9.

The sampling density in 5D is consequently the product of twice the eigenvalues, that is 32 times the square root of the determinant of the covariance matrix, up to a constant k, the 5D volume of integration.

$$N = 32k\sqrt{|\Sigma\,|} \tag{22}$$

Our situation is different from that of traditional sampling because we are only interested in the final sliced spectrum. That is, our goal is to prevent replicas of the full spectrum in (c) (See Figure 9) to overlap the sliced spectrum in (d). We must consider the replicas of the full spectrum (c) and not that of the slice (d) because (c) is the only function we can access accurately with point sampling.

Since only the slice should not overlap, we can make the replicas half overlapping as shown in Figure 9.f. For the simple 2D case of our example, that means we only need half the samples predicted by sampling theory (since our minimum distance along the time axis is halved). In our general 5D case we can have 3 dimensions half overlapping leading to a 1/8 reduction of samples:

$$N = 4k\sqrt{|\Sigma\,|} \tag{23}$$

The reconstruction filter is then simply the inverse transform of the 2D slice shown in Figure 9 (d). It is further explained in Section 6.3.

6.2 Sampling radiance

For each pixel, we sample the 5D pixel's domain, with the number of samples from Equation 23. We store those samples in a 2D grid. The samples contain the integrated radiance (a number of secondary light rays are sent to integrate radiance per sample) for the given sub-pixel position, lens coordinate and time value. We compute this value using path tracing and importance sampling of light sources but other integration methods could be used (multiple importance sampling, Metropolis light transport, etc).

6.3 Reconstruction

To perform reconstruction, we sum the samples with weights based on their position in the image plane. We choose to use a Gaussian filter since it has a convenient formulation in both Fourier and primal spaces.

Given our covariance matrix Σ for current pixel, we compute the filter by slicing its inverse Σ^{-1} along the two spatial components

$$\Sigma_s^{-1} = \Sigma_{x,y}^{-1}$$

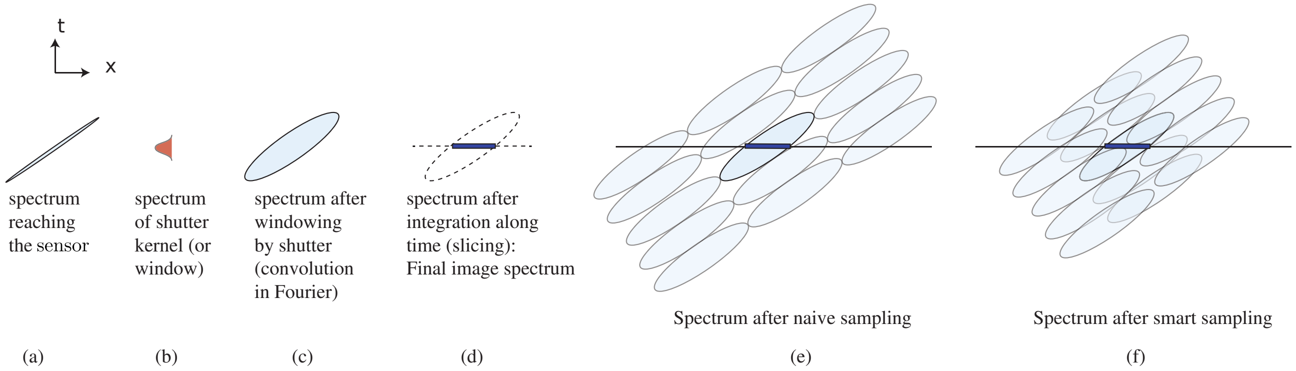

t
x
spectrum reaching the sensor
spectrum of shutter kernel (or window)
spectrum after windowing by shutter (convolution in Fourier)
spectrum after integration along time (slicing): Final image spectrum
Spectrum after naive sampling
Spectrum after smart sampling
(a)
(b)
(c)
(d)
(e)
(f)

Fig. 9: *Sampling and reconstruction, in two dimensions* (Ω_x, Ω_t) *for simplicity. For the spectrum reaching the sensor (a), we first convolve it (c) with the kernel of the shutter (b) (or the lens kernel). We are interested in the integration of the signal in the primal space. This is translated into the frequency domain into a slice of our convolved spectrum (d). We show that we can organize our samples (f) to allow us to save samples from the original packing (e).*

Then, to obtain the filter in image space, we need to apply an inverse Fourier transform. Since we are using a Gaussian filter, the covariance of the filter in the primal is the inverse of the covariance of the filter in Fourier space (up to a constant factor). Our reconstruction filter will therefore have the following formulation:

$$w_{\Sigma_s}(\mathbf{x}-\mathbf{p}) = e^{-\frac{1}{2}(\mathbf{x}-\mathbf{p})^T\Sigma_s(\mathbf{x}-\mathbf{p})} \tag{24}$$

6.4 Implementation details

Similar to Egan et al.[2009] we need to take into account the variation of covariance in a neighborhood (last paragraph of their appendix). In particular, the reconstruction for a pixel with a low-frequency prediction needs to gather samples from distant pixels. However, the spectrum estimate of these distant pixels could be very different, and in particular reveal high frequencies that cannot be handled by such a wide reconstruction filter. To avoid this, we only use samples from pixels whose filters overlap the pixel we are looking at.

We post-process the matrices in image space after the covariance tracing step in order to maintain some coherence and avoid outliers (low frequency with respect to the neighbors) that can occur when the number of covariance matrices sampled per pixel is not enough to correctly capture the frequency content. For this, we use a max filter on the determinant of the matrix. This post-process is in favor of high frequency covariance matrices and thus conservative. We apply this process per pixel and each matrix takes the max within its filter footprint.

In our implementation, we used a Gaussian shutter. Other kind of shutter can be used as long as the covariance of their spectrum is defined.

7. RESULTS

In this section, we present the results of our covariance tracing, adaptive sampling and reconstruction algorithms. First, we validate the use of covariance tracing with respect to measured covariance from a path tracer (Section 7.1). We show that we can predict the effects of motion, texture and occlusion using covariance matrices. Then, we perform *unit tests* (Section 7.2) to validate our algorithm for the different phenomena such as blur due to shallow depth-of-field, motion blur, occlusion, etc. Finaly, we demonstrate the efficacy of our algorithm in rendering complex scenes in comparison with a path tracer with multiple importance sampling (Section 7.3).

7.1 Validation of the covariance computation

We validated our covariance computation by comparing to ground truth covariance matrices. We designed a path tracer that records dense radiance samples in 5D space and computed the covariance component over the Fourier transform of this 5D function. Figure 10 presents the comparison of measured local light-fields (using a path tracer) from a light source reflecting on a diffuse striped plane moving orthogonally to the direction of the stripes. Our predicted covariance matrices are close to the measured quantities, and the correlation factors are correctly estimated [2].

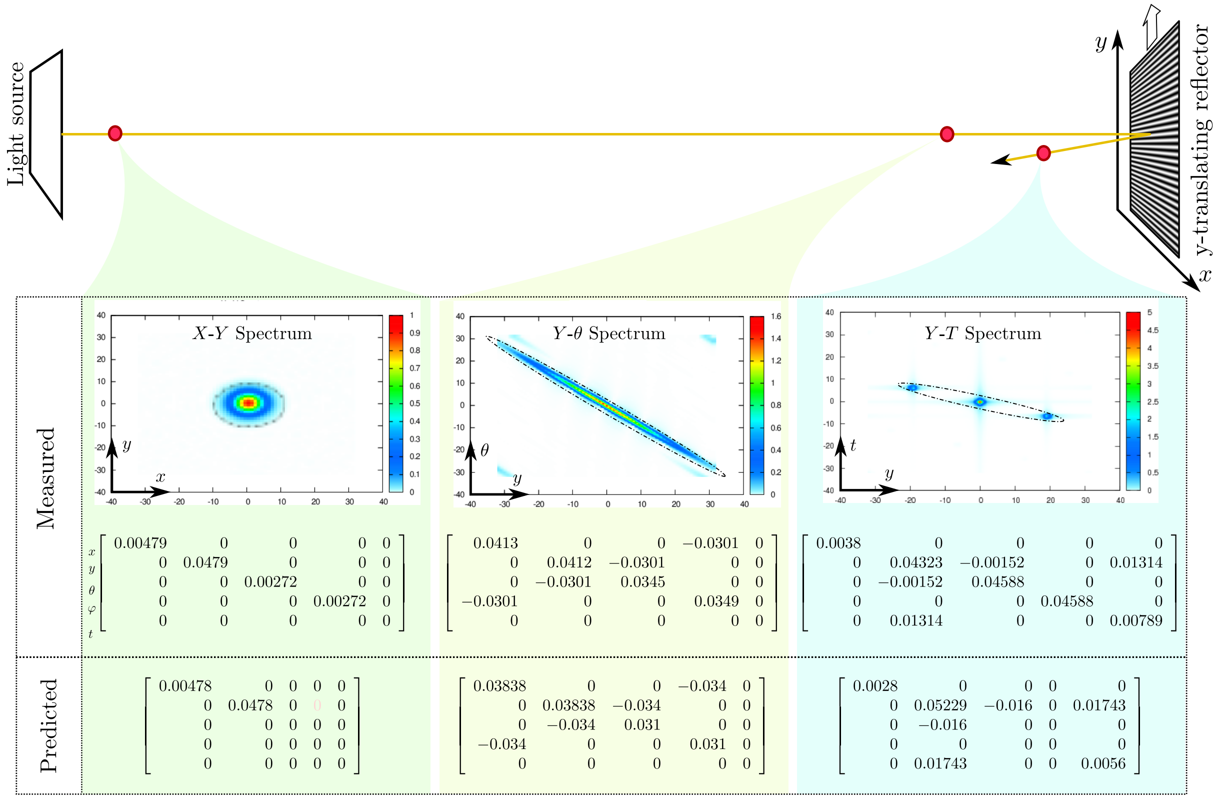
Figure 11 presents the comparison of measured local light-fields from a light source occluded by a rotated quad. We show that the occlusion grid correctly estimates the anisotropy of the local light-field after the blocker. It should be noticed that our predicted covariance matrix estimates a higher degree of occlusion anisotropy. This is because the windowed Fourier transform applied to compute the reference covariance introduces parasitic low frequencies in all directions at the same time.

Figure 12 presents an analysis of different regions on a soft shadow casted by a moving occluder. Our analysis correctly predicts regions where the shadow is influenced by the motion and regions where the motion doe not change the shadow. Note that the source used in Figure 12 is smaller than the one used in Figure 11 resulting in higher spatial covariance.

7.2 Unit tests

In this section, we propose to review some basic examples in order to view the effects handled by our theory. These examples are

[2] Differences between values are explained by the window we applied to compute the Fourier transform of measured light-fields and to their coarse resolutions.



$$\begin{array}{c}x\\y\\\theta\\\varphi\\t\end{array}\begin{bmatrix}0.00479 & 0 & 0 & 0 & 0\\0 & 0.0479 & 0 & 0 & 0\\0 & 0 & 0.00272 & 0 & 0\\0 & 0 & 0 & 0.00272 & 0\\0 & 0 & 0 & 0 & 0\end{bmatrix}\quad\begin{bmatrix}0.0413 & 0 & 0 & -0.0301 & 0\\0 & 0.0412 & -0.0301 & 0 & 0\\0 & -0.0301 & 0.0345 & 0 & 0\\-0.0301 & 0 & 0 & 0.0349 & 0\\0 & 0 & 0 & 0 & 0\end{bmatrix}\quad\begin{bmatrix}0.0038 & 0 & 0 & 0 & 0\\0 & 0.04323 & -0.00152 & 0 & 0.01314\\0 & -0.00152 & 0.04588 & 0 & 0\\0 & 0 & 0 & 0.04588 & 0\\0 & 0.01314 & 0 & 0 & 0.00789\end{bmatrix}$$

$$\begin{bmatrix}0.00478 & 0 & 0 & 0 & 0\\0 & 0.0478 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0\end{bmatrix}\quad\begin{bmatrix}0.03838 & 0 & 0 & -0.034 & 0\\0 & 0.03838 & -0.034 & 0 & 0\\0 & -0.034 & 0.031 & 0 & 0\\-0.034 & 0 & 0 & 0.031 & 0\\0 & 0 & 0 & 0 & 0\end{bmatrix}\quad\begin{bmatrix}0.0028 & 0 & 0 & 0 & 0\\0 & 0.05229 & -0.016 & 0 & 0.01743\\0 & -0.016 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0\\0 & 0.01743 & 0 & 0 & 0.0056\end{bmatrix}$$

Fig. 10: *A Comparison of predicted and measured covariance matrices of light field spectra for a toy scene involving a square light source with a Gaussian emission pattern, illuminating a textured reflector at a distance of 1m. The reflector is translating vertically at a speed of* $0.1 m.s^{-1}$. *The first row shows the sliced spectrum of 5D measurements at three positions (just after the source, just before the reflector, and just after the reflector), to be compared with the predicted covariance matrices in the following row.* **Notes:** *(1) In the measurements, we zeroed values below* 10^{-6} *for clarity. (2) some positive values in the measurements come from the windowing applied before the FFT, e.g. the* $\theta - \theta$ *and* $\varphi - \varphi$ *covariances in the 1st and 3rd steps. (3) At each step, we show slices with the most significant correlation, i.e. respectively the* $X - Y$, $Y - \theta$, *and* $Y - t$ *slices.*

provided as pedagological examples and so we do not compared to ground truth or focus on efficiency.

In Figure 13, we show how shininess and curvature of an object can affect the covariance, the respective reconstruction filters and the predicted number of samples required. The filters are correctly oriented along the surface, due to our handling of anisotropy.

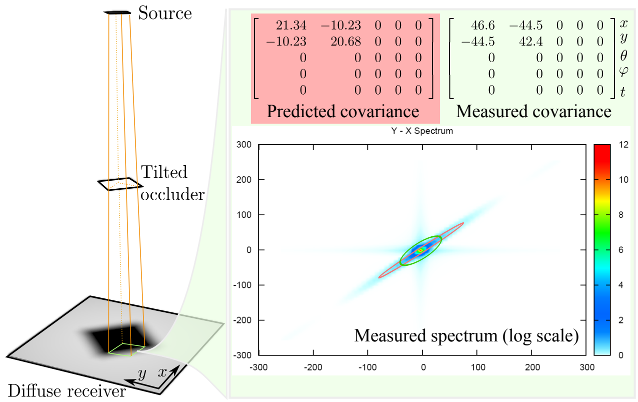
In Figure 14, we show how motion elongates reconstruction filters along the projected direction of the motion. In Figure 15, we illustrate that our algorithm exploits angular frequency to adapt the reconstruction of the depth-of-field effect. Both effects exploit the fact that slicing the 5D covariance matrix to produce 2D filters will stretch the filter in the direction of the shear (e.g. angular directions for the lens, direction of motion for time).

In Figure 16, we show the effect of occlusion on the estimate of filters. We underline the effect of the resolution of the occlusion grid on small scale elements. Our voxel-based method tends to overestimate occlusion and can possibly lead to oversampling of smooth regions if the resolution of the occlusion grid is not high enough.

7.3 Comparisons

We compared our algorithm implementation to a standard path tracer. The snooker scene (Figure 17) presents a billiard scene under the lighting of a sky environment map. The scene exhibits both low and high frequency materials (diffuse, glossy, and specular). Frequency information is computed using 15 covariance samples per pixel. We limited the maximum number of primary rays per pixel to 100 for our algorithm. We used a 200-wide voxel grid for the occlusion detection. We show a zoomed comparison between our algorithm and a path tracer in Figure 18 for an equal time of computation.

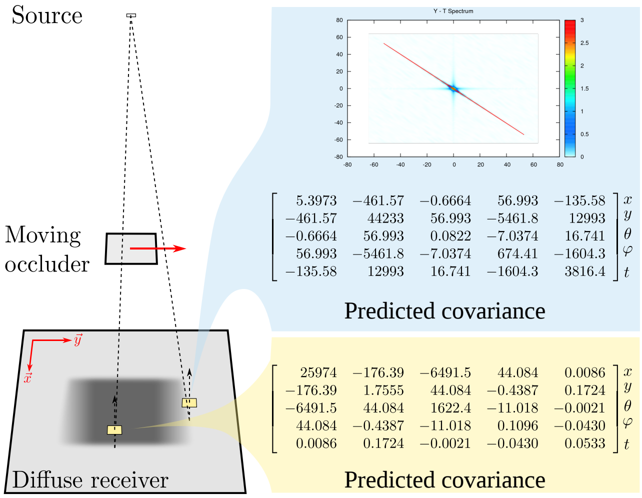
The helicopter scene (Figure 21) shows a toy lit by a square light source. The blades of the helicopter are rotating around its rotor's



$$\begin{bmatrix} 21.34 & -10.23 & 0 & 0 & 0 \\ -10.23 & 20.68 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{array}{c} \begin{bmatrix} 46.6 & -44.5 & 0 & 0 & 0 \\ -44.5 & 42.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} x \\ y \\ \theta \\ \varphi \\ t \end{array}$$

Predicted covariance   Measured covariance

Fig. 11: *A comparison of the predicted and measured covariances at the edge of the shadow caused by the square occluder. Our estimate correctly captures the anisotropy of the signal (the first eigenvector is* $(0.69, 0.71)$*). It is significantly more anisotropic (eigenvalues 89 and 0.05) than the measured covariance (eigenvalues 31.2 and 10.8), which we attribute to the effect of windowing in the measurement (see the bars along each axis in the spectrum image).*



$$\begin{bmatrix} 5.3973 & -461.57 & -0.6664 & 56.993 & -135.58 \\ -461.57 & 44233 & 56.993 & -5461.8 & 12993 \\ -0.6664 & 56.993 & 0.0822 & -7.0374 & 16.741 \\ 56.993 & -5461.8 & -7.0374 & 674.41 & -1604.3 \\ -135.58 & 12993 & 16.741 & -1604.3 & 3816.4 \end{bmatrix} \begin{array}{c} x \\ y \\ \theta \\ \varphi \\ t \end{array}$$

Predicted covariance

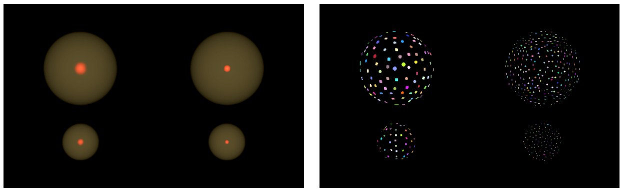
$$\begin{bmatrix} 25974 & -176.39 & -6491.5 & 44.084 & 0.0086 \\ -176.39 & 1.7555 & 44.084 & -0.4387 & 0.1724 \\ -6491.5 & 44.084 & 1622.4 & -11.018 & -0.0021 \\ 44.084 & -0.4387 & -11.018 & 0.1096 & -0.0430 \\ 0.0086 & 0.1724 & -0.0021 & -0.0430 & 0.0533 \end{bmatrix} \begin{array}{c} x \\ y \\ \theta \\ \varphi \\ t \end{array}$$

Predicted covariance

Fig. 12: *We analyze the effect of a moving occluder on our covariance estimator. Our estimator correctly depicts the anisotropy created by the motion of the occluding plane along the* **y** *axis. Occlusions by edges orthogonal to the motion bring high frequency to the temporal domain while occlusions by edges along the motion don't affect the temporal domain. We do not provide measured covariances because the windowing of the measured light-field introduced too much frequency to be compared against our prediction.*

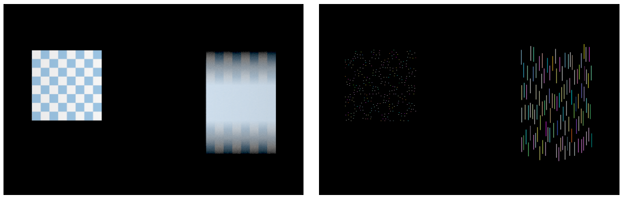
axis creating motion blur, while the textured background of the scene is out of focus. We used 10 light paths per pixel to estimate the covariance and a maximum of 200 samples per pixel for the reconstruction. Again, we compare our results with a path-traced image computed within the same amount of time.

We performed all computations on a Xeon W3520 at 2.66 GHz with 8GB of RAM. Our algorithm takes advantage of parallel computation for sampling both covariance rays and radiance rays.



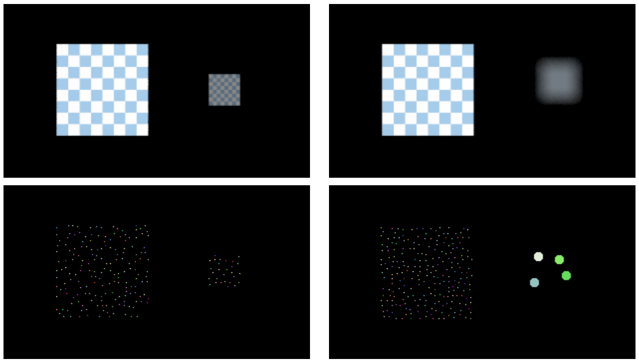
(a) Input scene   (b) Reconstruction filters

Fig. 13: *Four spheres of different size and shininess are lit by a square light source. The curvature and shininess of the spheres influence the reconstruction filter shapes. The filters align correctly with the geometry factor allowing us to filter samples along curvature.* ***Note:*** *we extract the maximum frequency of the BRDF for the covariance estimation.*



(a) Input scene   (b) Reconstruction filters

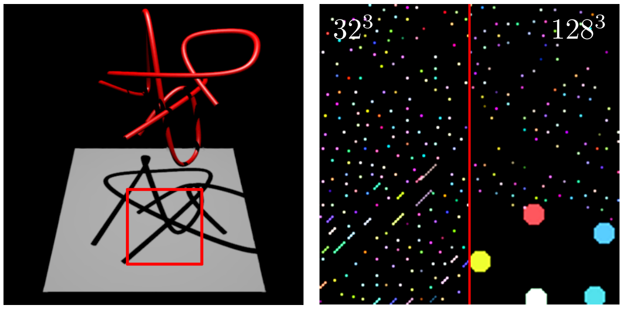
Fig. 14: *The motion of the planes influence the reconstruction filters shape (b) by elongating them in the direction of the motion.* ***Note:*** *the frequency of the texture has been set to its max to produce a comprehensive figure.*



(a) Small lens radius   (b) Larger lens radius

Fig. 15: *A comparison of two different aperture sizes, for the same camera configuration (position, direction and sensor size). For a small aperture (a), the depth of field is large (little defocus blur) and the filters are not changed. For a bigger aperture size (b), the depth of field is much shallower and and the filters are bigger.*

We emphasis the effect of the 5D windowing in Figure 20. Since the spatial $x - y$ windowing is a user parameter, it can be used to control the width of the filters in low frequency regions. We noticed that using too wide a filtering kernel leads to diffusion of high frequencies during the max filtering step. A trade-off must be made to obtain an optimal speed of our method.



(a) Input scene (b) Reconstruction Filters (red inset)

Fig. 16: *A complex occluder is casting a sharp shadow onto a diffuse plane. We present different resolutions of the occlusion grid for the red inset presented on the input scene (a). The example on the right half of (b) correctly depicts the contour of the shadow on the plane as the resolution of the grid is fine enough. On the contrary, the left half of (b) over-estimates the occlusion due to a coarse resolution of the grid.*

These scenes demonstrate that our method allows us to save computation time for the low frequency parts of the scene as shown in Table III.

Scene	Our (covariance tracing / reconstruction)	Reference
Snooker	25m (2m36 / 16s)	2h25m
Helicopter	29m (2m / 16s)	x

Table III. : *Timing comparison between our algorithm and our reference tracer for the snooker (Fig. 17) and for the helicopter scene (Fig. 21). The first column shows the time taken by our algorithm. Inside the brackets we show the covariance acquisition and the reconstruction timings. For the helicopter scene, we do not report the path tracer timing since we are doing an equal time comparison.*

7.4 Discussion

Comparison to Egan et al.. We differ from Egan et al. [2009] in that we consider the shutter effect as a multiplication by a unit pulse, whose size depends on the time the shutter stays open, followed by an integration along time in the primal domain. On the contrary, Egan *et al.* consider the shutter effect to be the convolution of the signal by the same unit pulse taken at discrete time (again in the primal space). As a result, in our final reconstruction, filters are not sheared in space-time like with Egan *et al.*'s approach, which dramatically simplifies implementation. This also means that there is no need to consider samples outside the shutter interval. The reconstruction filter can be sheared in the image plane if the 5D spectrum (Figure 9 (a)) is not isotropic. This is usually the case with moving objects where the spectrum reaching the lens (Figure 9 (a)) is sheared along the object motion.

Comparison to Soler et al. [2009]. Our approach has the same benefit as Soler *et al.*'s depth of field approach because we can take advantage of low bandwidth either along the lens or the image. However, their method relies on a two-step reconstruction whereas we do one single 2D reconstruction. As a result, their method can only exploit low bandwidth along either of the axes — angle or space — but not diagonally, while our method is able to achieve this as well. Furthermore, our method handles motion blur and general light paths.

Comparison to 2D post-processing. Our method is different from 2D post-processing solutions [Max and Lerner 1985; Potmesil and Chakravarty 1981] that blur a 2 or 2.5D image with spatially-varying filters according to depth or motion. The ressemblance exists only for purely diffuse planes. In general, our reconstruction filter is not simply the circle of confusion or motion vector: it takes into account complex effects in combination such as glossy highlights, occlusion, and lighting.

In a simple case such as the defocus of a diffuse object, the size of the reconstruction filter is exactly the size of the circle of confusion. But for more complex situations such as glossy objects or occlusion boundaries, the bandwidth becomes higher and the reconstruction filter is smaller.

8. LIMITATIONS

The main complication in computing the covariance matrix is the detection of partial occlusion. Since we use a voxel grid we add two major constraints: First, the locality of the frequency analysis is bound to the resolution of the voxel grid. This can result in over estimating the occlusion in scenes where tiny details occur (such as leaves on a tree). Second, the use of the voxel grid adds the cost of ray marching during the occlusion detection step. To retain the same locality in the occlusion detection for different scene sizes, the size of the voxel grid must be adapted, potentially increasing the computational cost of the frequency analysis.

Furthermore, our method might become ineffective in very complicated regions where effects such as occlusion and depth of field or motion blur co-occur. Since we use pre-integrated light samples, variance is only reduced by the number of samples gathered. For such regions, gathering from outside a pixel will create noticable aliasing and we obtain worse results than path tracing, with say stratified sampling, which decreases the variance in all dimensions simultaneously. However, since we are able to identify such confluences of complex effects, we could still use traditional path tracing for these pixels.

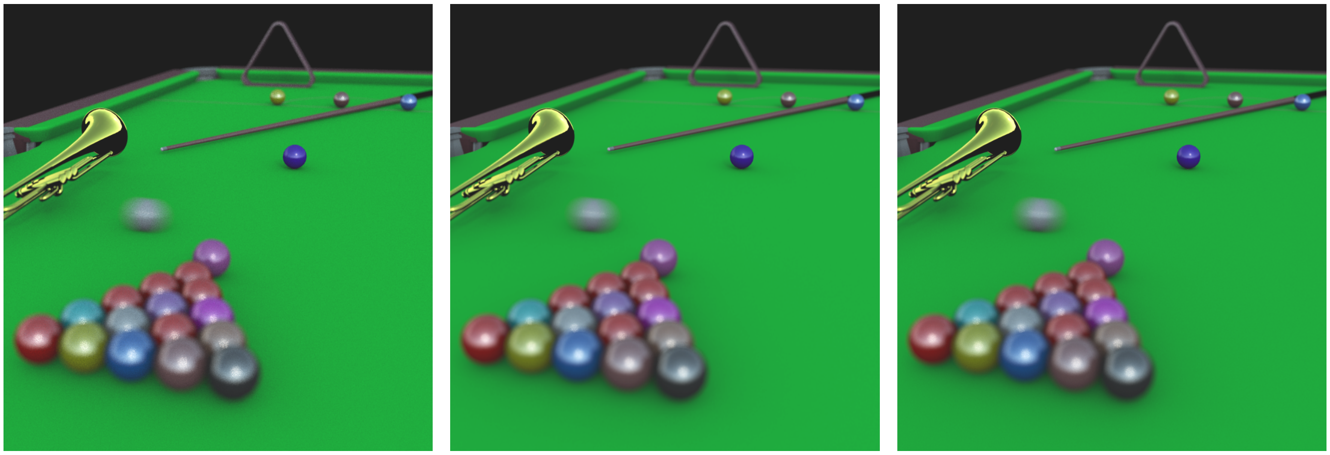
9. CONCLUSION

We have presented a new method to propagate frequency information along light paths. We demonstrated the use of the covariance matrix for adaptive sampling in a one bounce ray tracing algorithm, which permits the sharing of samples across pixels using a 2D reconstruction method. We validated the usability and the validity of our approximation against some ground truth covariance matrices.

We aim to further exploit the possibility of the covariance matrix in more general cases such as volumetric environments or complete global illumination algorithms such as path tracing methods.

10. ACKNOWLEDGMENTS

We thank all the reviewers of the MIT team, Charles de Rousiers, Jaakko Lehtinen and Kavita Bala for their insightful comments during the writing of this work. We thank also user vklidu from BlenderArtists.org forums for permission to use his helicopter scene. Frédo Durand acknowledges funding from the NSF.



(a) Equal time reference using 512 samples per pixel (25 minutes)

(b) Our algorithm (25 minutes)

(c) Reference with same maximum number of samples per pixel (3000 samples per pixel, 2 hours 25 minutes)

Fig. 17: *The snooker scene rendered using our algorithm at the center, with a path tracer using the maximum number of samples in a pixel used in our algorithm at the right (same quality), and using the same amounts of time as our algorithm at the left.*

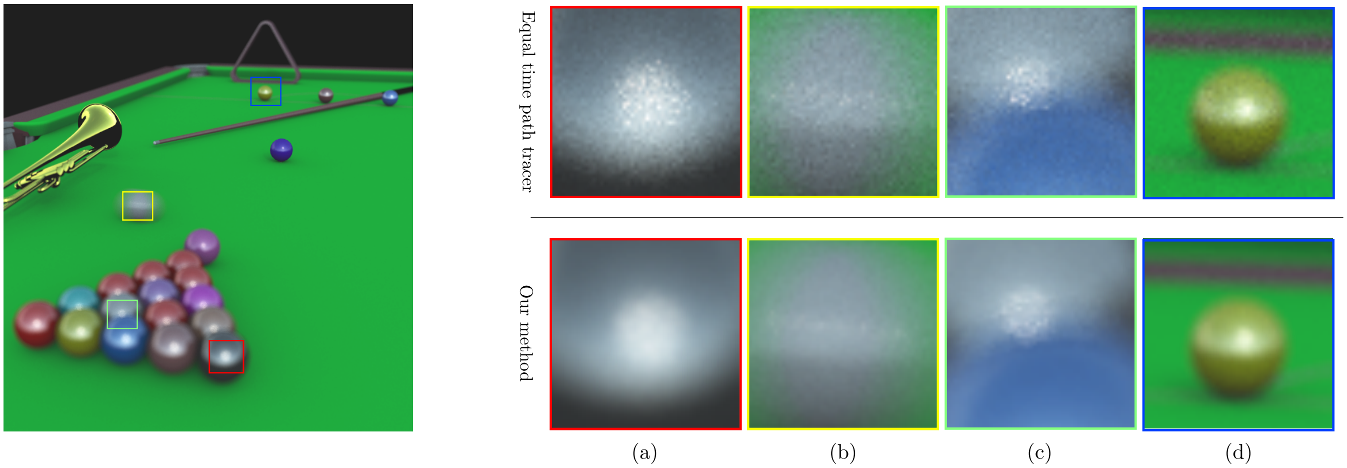


Fig. 18: *A close-up comparison for the snooker scene (Figure 17) between our algorithm and equal time path tracing. We present several regions where the effect of our adaptive sampling and reconstruction are perceptible. In the red inset (a) we demonstrate that we make the rendering of glossy out-of-focus surface converge faster. The yellow inset (b) show the rendering of motion-blur with a glossy surface. The green inset (c) emphasizes the rendering of complex combinations of effects such as depth-of-field occlusion of glossy surfaces. In this setting, we can see that our algorithm exhibits artifacts because of a limited number of samples. Finally the blue inset (d) shows the rendering of both out of focus diffuse green region of the pool and the glossy ball.*

REFERENCES

ARIKAN, O., FORSYTH, D. A., AND O'BRIEN, J. F. 2005. Fast and detailed approximate global illumination by irradiance decomposition. *ACM Trans. Graph. 24,* 3, 1108–1114.

BARSKY, B. A., HORN, D. R., KLEIN, S. A., PANG, J. A., AND YU, M. 2003. Camera models and optical systems used in computer graphics: Part II, image based techniques. In *International Conference on Computational Science and its Applications*.

BARSKY, B. A. AND KOSLOFF, T. J. 2008. Algorithms for rendering depth of field effects in computer graphics. In *12th WSEAS international conference on Computers*. 999–1010.

BOLIN, M. R. AND MEYER, G. W. 1995. A frequency based ray tracer. In *SIGGRAPH '95*. 409–418.

BOLIN, M. R. AND MEYER, G. W. 1998. A perceptually based adaptive sampling algorithm. In *SIGGRAPH '98*. 299–309.

CHAI, J.-X., TONG, X., CHAN, S.-C., AND SHUM, H.-Y. 2000. Plenoptic sampling. In *SIGGRAPH '00*. 307–318.

CHEN, M. AND ARVO, J. 2000. Theory and application of specular path perturbation. *ACM Trans. Graph. 19,* 4, 246–278.

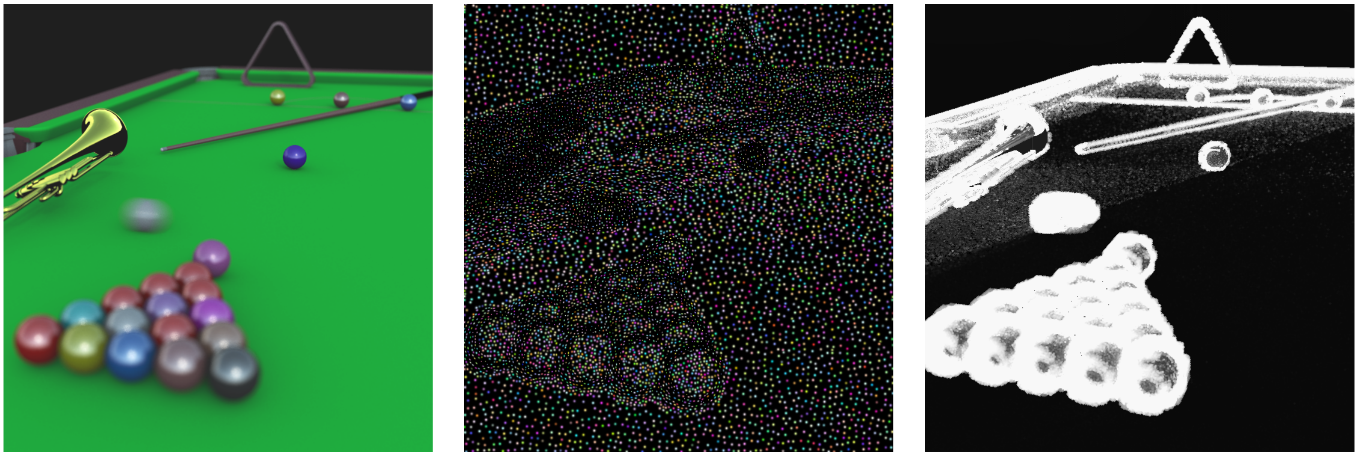
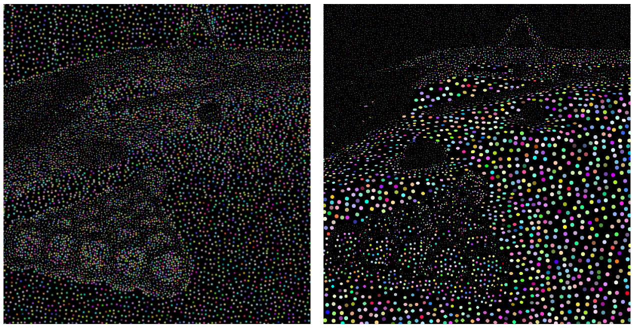


Fig. 19: *The snooker scene rendered using our algorithm. We present the filters used to perform the reconstruction and the estimated needed sampling rate. Note that we used a windowing of 3 pixel-wide for this example. We used a small window to avoid high frequency leaking into the diffuse region that will occur during the max filtering step. We present a view of the filters using a smaller windowing in figure 20.*



(a) Using a 3 pixel-wide window (b) Using a 10 pixels wide window

Fig. 20: *In this figure, we compare the spatial windowing parameter. Using a large spatial windowing kernel leads to a larger filters in smooth regions.*

Cook, R. L. 1986. Stochastic sampling in computer graphics. *ACM Trans. Graph. 5*, 51–72.

Cook, R. L., Carpenter, L., and Catmull, E. 1987. The reyes image rendering architecture. *SIGGRAPH Comput. Graph. 21*, 95–102.

Cook, R. L., Porter, T., and Carpenter, L. 1984. Distributed ray tracing. *SIGGRAPH Comput. Graph. 18,* 3, 137–145.

Dippé, M. A. Z. and Wold, E. H. 1985. Antialiasing through stochastic sampling. *ACM SIGGRAPH Computer Graphics 19,* 3, 69–78.

Durand, F., Holzschuch, N., Soler, C., Chan, E., and Sillion, F. X. 2005. A Frequency Analysis of Light Transport. *ACM Trans. Graph. 24,* 3, 1115 – 1126.

Egan, K., Durand, F., and Ramamoorthi, R. 2011. Practical filtering for efficient ray-traced directional occlusion. *SIGGRAPH Asia 2011*.

Egan, K., Hecht, F., Durand, F., and Ramamoorthi, R. 2011. Frequency analysis and sheared filtering for shadow light fields of complex occluders. *ACM Transactions on Graphics 30,* 2 (Apr.), 1–13.

Egan, K., Tseng, Y.-T., Holzschuch, N., Durand, F., and Ramamoorthi, R. 2009. Frequency analysis and sheared reconstruction for rendering motion blur. *ACM Trans. Graph. 28,* 3.

Fatahalian, K., Luong, E., Boulos, S., Akeley, K., Mark, W. R., and Hanrahan, P. 2009. Data-parallel rasterization of micropolygons with defocus and motion blur. In *Proceedings of the Conference on High Performance Graphics 2009*. HPG '09. ACM, New York, NY, USA, 59–68.

Ferwerda, J. A., Shirley, P., Pattanaik, S. N., and Greenberg, D. P. 1997. A model of visual masking for computer graphics. In *Siggraph '97*. 143–152.

Gerrard, A. and Burch, J. 1975. *Introduction to matrix methods in optics*. Wiley Series in Pure and Applied Optics. John Wiley.

Hachisuka, T., Jarosz, W., Weistroffer, R. P., Dale, K., Humphreys, G., Zwicker, M., and Jensen, H. W. 2008. Multidimensional adaptive sampling and reconstruction for ray tracing. *ACM Trans. Graph. 27,* 3, 33:1–33:10.

Hašan, M., Pellacini, F., and Bala, K. 2007. Matrix row-column sampling for the many-light problem. *ACM Trans. Graph. 26,* 3.

Heckbert, P. S. 1989. Master's thesis, CS Dept, UC Berkeley.

Hou, Q., Qin, H., Li, W., Guo, B., and Zhou, K. 2010. Micropolygon ray tracing with defocus and motion blur. In *ACM SIGGRAPH 2010 papers*. SIGGRAPH '10. ACM, New York, NY, USA, 64:1–64:10.

Igehy, H. 1999. Tracing ray differentials. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '99. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 179–186.

Kirk, D. and Arvo, J. 1991. Unbiased sampling techniques for image synthesis. *SIGGRAPH Comput. Graph. 25,* 4, 153–156.

Kolb, C., Hanrahan, P. M., and Mitchell, D. 1995. A realistic camera model for computer graphics. In *SIGGRAPH '95*. 317–324.

Kraus, M. and Strengert, M. 2007. Depth-of-field rendering by pyramidal image processing. *Computer Graphics Forum 26,* 3, 645–654.

Lee, S., Eisemann, E., and Seidel, H.-P. 2009. Depth-of-field rendering with multiview synthesis. *ACM Trans. Graph. 28,* 5, 134:1–134:6.

Lee, S., Eisemann, E., and Seidel, H.-P. 2010. Real-time lens blur effects and focus control. *ACM Trans. Graph. 29,* 4, 65:1–65:7.

Lehtinen, J., Aila, T., Chen, J., Laine, S., and Durand, F. 2011. Temporal light field reconstruction for rendering distribution effects. *ACM Trans. Graph. 30,* 4.

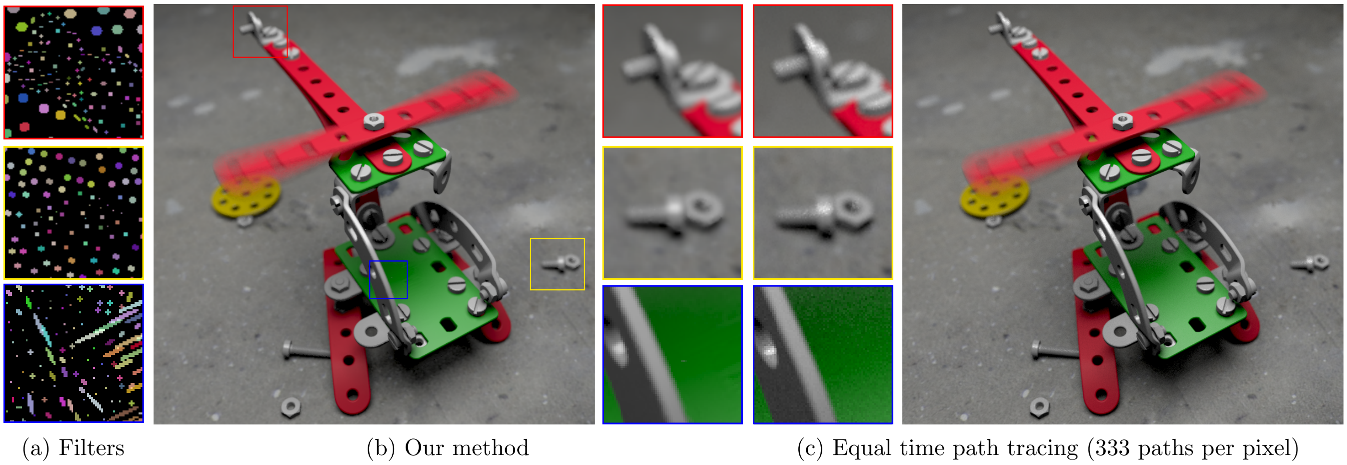


Fig. 21: *The helicopter scene rendered using our algorithm (a) and compared with an equal time path traced rendering (b). We used 10 light paths to estimate the covariance and used a maximum of 200 converged samples for the reconstruction.*

LESSIG, C. AND FIUME, E. 2010. On the effective dimension of light transport. *Comput. Graph. Forum 29,* 4, 1399–1403.

LEVIN, A., HASINOFF, S. W., GREEN, P., DURAND, F., AND FREEMAN, W. T. 2009. 4D frequency analysis of computational cameras for depth of field extension. *ACM Trans. Graph. 28,* 3, 97:1–97:10.

LEVIN, A., SAND, P., CHO, T. S., DURAND, F., AND FREEMAN, W. T. 2008. Motion-invariant photography. *ACM Trans. Graph. 27,* 3, 71:1–71:10.

LOVISCACH, J. 2005. Motion blur for textures by means of anisotropic filtering. In *Eurographics Symposium on Rendering*. 105–110.

MAHAJAN, D., SHLIZERMAN, I. K., RAMAMOORTHI, R., AND BELHUMEUR, P. 2007. A theory of locally low dimensional light transport. *ACM Trans. Graph. 26,* 3.

MAX, N. L. AND LERNER, D. M. 1985. A two-and-a-half-d motion-blur algorithm. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '85. ACM, New York, NY, USA, 85–93.

MITCHELL, D. 1987. Generating antialiased images at low sampling densities. In *SIGGRAPH '87*.

MITCHELL, D. P. 1991. Spectrally optimal sampling for distribution ray tracing. In *SIGGRAPH '91*. 157–164.

MYSZKOWSKI, K. 1998. The visible differences predictor: applications to global illumination problems. In *Rendering Techniques '98 (EG Workshop on Rendering '98)*. 223–236.

NG, R. 2005. Fourier slice photography. *ACM Trans. Graph. 24,* 3, 735–744.

OVERBECK, R. S., DONNER, C., AND RAMAMOORTHI, R. 2009. Adaptive wavelet rendering. *ACM Trans. Graph. 28,* 5, 140:1–140:12.

POTMESIL, M. AND CHAKRAVARTY, I. 1981. A lens and aperture camera model for synthetic image generation. *SIGGRAPH Comput. Graph. 15,* 3, 297–305.

POTMESIL, M. AND CHAKRAVARTY, I. 1983. Modeling motion blur in computer-generated images. *SIGGRAPH Comput. Graph. 17,* 3, 389–399.

QUINN, J. A., LANGBEIN, F. C., AND MARTIN, R. R. 2007. Low-discrepancy point sampling of meshes for rendering. In *Eurographics Symposium on Point-Based Graphics*. 19–28.

RAMAMOORTHI, R. AND HANRAHAN, P. 2001. A signal-processing framework for inverse rendering. In *SIGGRAPH '01*. 117–128.

RAMAMOORTHI, R., KOUDELKA, M., AND BELHUMEUR, P. 2005. A fourier theory for cast shadows. *IEEE Trans. Pattern Anal. Mach. Intell. 27,* 2, 288–295.

RAMAMOORTHI, R., MAHAJAN, D., AND BELHUMEUR, P. 2007. A first-order analysis of lighting, shading, and shadows. *ACM Trans. Graph. 26,* 1.

ROUSSELLE, F., KNAUS, C., AND ZWICKER, M. 2011. Adaptive sampling and reconstruction using greedy error minimization. In *ACM SIGGRAPH Asia 2011 papers*. SIGGRAPH Asia '011. ACM, New York, NY, USA.

SEN, P. AND DARABI, S. 2011a. Compressive rendering: A rendering application of compressed sensing. *IEEE Transactions on Visualization and Computer Graphics 17,* 4 (April), 487 – 499.

SEN, P. AND DARABI, S. 2011b. On filtering the noise from the random parameters in monte carlo rendering. *ACM Transactions on Graphics (TOG) to appear*.

SLOAN, P.-P., HALL, J., HART, J., AND SNYDER, J. 2003. Clustered principal components for precomputed radiance transfer. *ACM Trans. Graph. 22,* 3, 382–391.

SOLER, C., SUBR, K., DURAND, F., HOLZSCHUCH, N., AND SILLION, F. X. 2009. Fourier depth of field. *ACM Trans. Graph. 28,* 2.

STOKES, W. A., FERWERDA, J. A., WALTER, B., AND GREENBERG, D. P. 2004. Perceptual illumination components: a new approach to efficient, high quality global illumination rendering. *ACM Trans. Graph. 23,* 3, 742–749.

SUYKENS, F. AND WILLEMS, Y. 2001. Path differentials and applications. In *Rendering Techniques '01 (EG Workshop on Rendering)*. 257–268.

WALTER, B., ARBREE, A., BALA, K., AND GREENBERG, D. P. 2006. Multidimensional lightcuts. *ACM Trans. Graph. 25,* 3, 1081–1088.

WARD, G. J. AND HECKBERT, P. 1992. Irradiance gradients. In *EG Workshop on Rendering '92*. 85–98.

WHITTED, T. 1980. An improved illumination model for shaded display. *Commun. ACM 23,* 6, 343–349.

ZHOU, T., CHEN, J., AND PULLEN, M. 2007. Accurate depth of field simulation in real time. *Computer Graphics Forum 26,* 1, 15–23.

APPENDIX

A. MOTION SHEAR

We show that the transformation of motion can be seen as a shear in a first order approximation. We will provide demonstration for a 3D spectrum that is anisotropic in space and anisotropic in angle.

Given a point x_t on the tangent plane of the central ray and an angle θ_t, as shown in Figure 22 we want to estimate the new position x_{t+dt} and new angle θ_{t+dt} after a linear motion and rotation of the tangent plane. We note that θ is the angle covered by the rotation of the tangent plane during dt, y the tangential translation of the tangent plane during dt, and x_o the intersection between the tangent plane at $t + dt$ and the central ray at t.

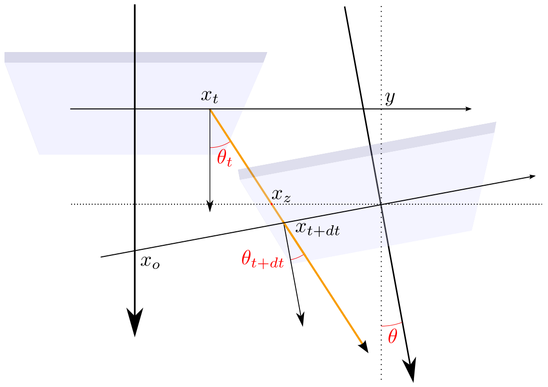


Fig. 22: *Shear from a light field frame to another.*

The new value of the angle is a simple addition $\theta_{t+dt} = \theta_t - \theta$.

We will decompose the derivation of x_{t+dt} into two steps. First we will find the intermediate position x_z after taking into account the translation only. Then we will infer the new position x_{t+dt} from x_z by looking at the rotation.

The intermediate position is the result of a spatial shear with a travel distance of $t\ v_z$ and a shift of distance y:

$$x_z = x_t - y + v_z\ t\ tan(\theta_t)$$

The last term is a second order term and we neglect it.

$$x_z \simeq x_t - y$$

The new position is found using the approximation that for very small angles θ_t, we can apply the law of cosines on the triangle defined by the intermediate point, the new point and the central position after motion. This approximation gives us:

$$x_{t+dt} \simeq \frac{x_z}{\cos(\theta_t)}$$

The formulation for x_{t+dt} is then :

$$x_{t+dt} \simeq \frac{x_t - y}{cos(\theta)}$$

Substituting the expression of y with the velocity multiplied by time, and putting it into a matrix formulation, we get :

$$\begin{bmatrix} x' \\ \theta' \\ t\prime \end{bmatrix} = \begin{bmatrix} \frac{1}{cos(\theta)} & 0 & -\frac{v_x}{cos(\theta)} \\ 0 & 1 & -v_\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ t \end{bmatrix}$$

We can further linearize the one over cosine term :

$$\frac{1}{cos(\theta_t)} \simeq 1$$

$$\begin{bmatrix} x' \\ \theta' \\ t\prime \end{bmatrix} = \begin{bmatrix} 1 & 0 & -v_x \\ 0 & 1 & -v_\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ t \end{bmatrix} \quad (25)$$

B. LINEAR TRANSFORMATION OF THE SIGNAL

In this section we derive the expression for the matrices to be used to update the covariance matrix of the light field's spectrum in each situation.

We want to compute the covariance of the function f expressed in a domain deformed by an invertible linear warping function $\mathcal{B}$, denoted as f', by the linear transform $\mathcal{B}$ with its associated matrix B. For any point $\mathbf{x}$ in the space, the value for the f in the space warped by $\mathbf{B}$ is :

$$f'(\mathbf{x}) = f \circ \mathcal{B}^{-1}(\mathbf{x})$$

We want to express the covariance matrix Σ' of f', based on the covariance matrix of Σ of f. From the definition we gave in section 4 of the covariance matrix :

$$\begin{aligned} \Sigma'_{i,j} &= \int_{\mathbf{x}\in\mathbb{R}^5} \langle \mathbf{x}, \mathbf{e}_i \rangle\ \langle \mathbf{x}, \mathbf{e}_j \rangle\ f'(\mathbf{x})\ \mathrm{d}\mathbf{x} \\ &= \int_{\mathbf{x}\in\mathbb{R}^5} \langle \mathbf{x}, \mathbf{e}_i \rangle\ \langle \mathbf{x}, \mathbf{e}_j \rangle\ f(\mathcal{B}^{-1}(\mathbf{x}))\ \mathrm{d}\mathbf{x} \end{aligned}$$

We apply the change of variables $B\mathbf{u} = \mathbf{x}$, to obtain the following integral formula :

$$\begin{aligned} \Sigma'_{i,j} &= \int_{\mathbf{u}\in\mathbb{R}^5} \langle B\mathbf{u}, \mathbf{e}_i \rangle\ \langle B\mathbf{u}, \mathbf{e}_j \rangle\ f(\mathbf{u})\ |B| \mathrm{d}\mathbf{u} \\ &= \int_{\mathbf{u}\in\mathbb{R}^5} \langle \mathbf{u}, B^T\mathbf{e}_i \rangle\ \langle \mathbf{u}, B^T\mathbf{e}_j \rangle\ f(\mathbf{u})\ |B| \mathrm{d}\mathbf{u} \quad (26) \end{aligned}$$

Now we expand the dot products :

$$\langle \mathbf{u}, B^T\mathbf{e}_i \rangle = \sum_k B_i^k \langle \mathbf{u}, \mathbf{e}_k \rangle$$

Injecting this into Equation 26 gives

$$\begin{aligned} \Sigma'_{i,j} &= |B| \sum_k \sum_l B_i^k B_j^l \int_{\mathbf{u}\in\mathbb{R}^5} \langle \mathbf{u}, \mathbf{e}_k \rangle\ \langle \mathbf{u}, \mathbf{e}_l \rangle\ f(\mathbf{u})\ \mathrm{d}\mathbf{u} \\ &= |B| \sum_k \sum_l B_i^k B_j^l \Sigma_{k,l} \\ &= |B| (B^T \Sigma B)_{i,j} \end{aligned}$$

Consequently, the matrix operator associated with the linear transformation $\mathcal{B}$ of the space is $\sqrt{|B|}B$.

C. BRDF MULTIPLICATION

BRDF multiplication is a band limiting process. In the most general situation, the 5D covariance matrix Σ of the signal doesn't have a full rank, and the calculation needs to be carefully conducted, approximating the signal by a combination of a Gaussian and Dirac functions.

Since Σ is always symmetric semi-definite and positive, it can be diagonalized with an orthogonal transform. Let Λ be the diagonal matrix of eigenvalues and U the matrix of eigenvectors:

$$\Sigma = U \Lambda U^T$$

Some of the eigenvalues λ_i might be null. Dirac along the associated eigenvector (since we always have energy in our light path, the DC term is never zero). Using the notation $\omega = (\omega_x, \omega_y, \omega_\theta, \omega_\phi, \omega_t)$:

$$f(\omega) = e^{-\omega^T U \Lambda^+ U^T \omega} \prod_j \delta_j(\omega)$$

Where Λ^+ is the pseudo inverse of Λ, and $\delta_j(\omega) = \delta(\langle \mathbf{e}_j, \omega \rangle)$

Because it has covariance only in the angular domain, the BRDF spectrum is well represented by a product of a 2D Gaussian along the θ and ϕ directions:

$$\text{brdf}(\omega) = e^{-\omega^T B \omega}$$

Where B has zeros everywhere except for the angular part of the matrix. Therefore, the product of the two functions is

$$\begin{aligned} f(\omega) \times \text{brdf}(\omega) &= e^{-\omega^T U \Lambda^+ U^T \omega} \prod_j \delta_j(\omega) e^{-\omega^T B \omega} \\ &= e^{-\omega'^T \Lambda^+ \omega'} e^{-\omega'^T U^T B U \omega'} \prod_j \delta_j(\omega) \\ &= e^{-\omega'^T (\Lambda^+ + U^T B U) \omega'} \prod_j \delta_j(\omega) \end{aligned}$$

The covariance matrix Σ' of this function is computed from the pseudo-inverse of $\Lambda^+ + U^T B U$:

$$\begin{aligned} \Sigma' &= ((U^T \Sigma U)^+ + U^T B U)^+ \\ &= U^T (\Sigma^+ + B)^+ U \end{aligned}$$