Accurate and Efficient Filtering using Anisotropic Filter Decomposition

Cyril Soler, Mahdi M. Bagher, Derek Nowrouzezahrai

Accurate and Efficient Filtering using Anistropic Filter Decomposition

Cyril Soler, Mahdi Mohammad-Bagher, Derek Nowrouzezahrai

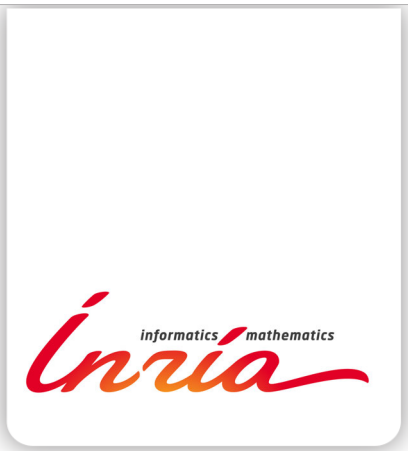
RESEARCH REPORT

N° 8349

August 2013

Project-Teams Maverick

ISSN 0249-6399 ISRN INRIA/RR--8349--FR+ENG



Accurate and Efficient Filtering using Anistropic Filter Decomposition

Cyril Soler*†, Mahdi Mohammad-Bagher‡†, Derek Nowrouzezahrai§

Project-Teams Maverick

Research Report n° 8349 — August 2013 — 19 pages

Abstract: Efficient filtering remains an important challenge in computer graphics, particularly when filters are spatially-varying, have large extent, and/or exhibit complex anisotropic profiles. We present an efficient filtering approach for these difficult cases based on an *isotropic filter decomposition* (IFD). By decomposing complex filters into linear combinations of b simpler, displaced isotropic kernels, and precomputing a compact prefiltered dataset, we are able to interactively apply any number of—potentially transformed—filters to a signal. Our performance scales linearly with the *size of the decomposition* $s \ll n$ (i.e. $\mathbf{O}(s)$ time), not the size n nor the dimensionality of the filter, and our prefiltered data requires $\mathbf{O}(bn)$ storage, comparing favorably to the state-of-the-art. We apply IFD to interesting problems in image processing and realistic rendering.

Key-words: Filtering, Compressive Sensing, Rendering

* Footnote for first author
† Shared foot note
‡ Footnote for second author
§ etc

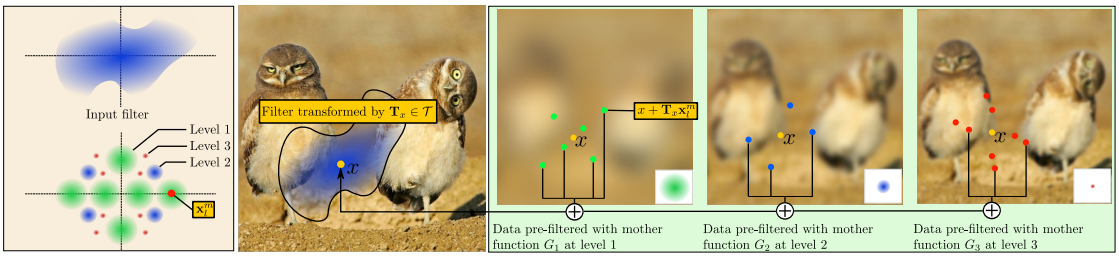
RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Filtrage Efficace par Decomposition Isotrope

Résumé : Les opérations de filtrage en synthèse/analyse d'images sont coûteuses à effectuer lorsque les filtres varient spatialement, sont très étendus et/ou très anisotropes. Nous présentons dans ce cas précis une méthode pour rendre le filtrage efficace, basée sur une décomposition du filtre en une combinaison linéaire de filtres isotropes, en translation. Le coût de notre méthode est linéaire par rapport au nombre de filtres utilisés dans la décomposition, et ne dépend pas de la taille des données filtrées. Nous en présentons différentes applications, en analyses d'images et en rendu.

Mots-clés : Filtrage, *Compressive Sensing*, Rendu



Filter decomposed over $\{G_l^m\}$ Input signal f IFD Filtering method

Figure 1: Illustrating the IFD for 2D images (Section 4.1). We decompose an input filter into a sum of displaced kernels G_l that are invariant under transforms $\mathcal{S}$ (here, 2D rotations). This allows us interactive filter the input signal with any transformed (by $\mathbf{S_x} \in \mathcal{S}$) copy of the filter, all with only a small number of lookups into prefiltered data. We similarly apply our theory to spherical filtering (Section 4.2)

1 Introduction

Filtering is a fundamental operation in many computer graphics domains including image processing, realistic rendering, and geometry reconstruction. Many filtering approaches assume a spatially-constant filter, simplifying filtering to a convolution, or target only a subset of filters such as those with small extent or limited profiles (e.g. Gaussians). We consider filters with spatially-varying behavior, arbitrary size, and potentially anisotropic shape.

We improve upon traditional trade-offs between filtering performance, storage cost, and accuracy by developing a new theory of *isotropic filter decomposition* (IFD) that permits the application of complex filters with *linear* time complexity in the size of our decomposition, and linear storage complexity in the number of frequency bandwidth levels used in the decomposition. We provide two concrete realizations of our theory to 2D image and spherical image filters, and we apply IFD to problems in image-processing and interactive realistic rendering.

Our approach proceeds in three steps, detailed for the general case of d-dimensional filtering with arbitrary spatially-varying filters, in Section 3: first, we devise a set of $b \ll n$ isotropic "mother" kernels of various frequency bandwidths (with exactly one kernel per bandwidth level); next, we prefilter our signal once with these kernels (in $\mathbf{O}(bn \log n)$ time in 2D and $\mathbf{O}(bn)$ on the sphere)and store the resulting prefiltered signals (with $\mathbf{O}(bn)$ storage) for use during filtering; finally, after decomposing an arbitrarily complex filter into an s-term ($b \leq s \ll n$) expansion of displaced copies of the b mother kernels, we can apply any transformed version (i.e. translated/rotated/scaled) of the filter to the signal using only s (constant-time) lookups into the prefiltered dataset.

We build atop our theory in Section 3 to detail the process and practical considerations of implementing IFDs in Section 4. We discuss the properties and specific choices of isotropic kernels for different domains and show that, in real-world scenarios, decomposing filters as weighted sums of these (displaced) kernels can be accomplished with techniques ranging in complexity from simple linear mappings (i.e. matrix multiplications) to compressive sensing (in order to optimize e.g. for sparsity). We detail applications of IFD to image processing and interactive realistic rendering in Section 5. See Figure 1 for an overview in the image domain.

We present the following contributions:

- We develop a general theory of filtering (in any d-dimensional space) based on *isotropic filter decomposition*, a compact, transform-invariant representation with controllable error.

- We provide two practical IFD realizations for 2D and spherical image domains, detailing several design principles and numerical tools to guide IFD development in other domains.
- We apply IFDs to applications in interactive image editing and environment map shading with realistic BRDFs.
- Filtering with IFDs reduces to only a few lookups into compact prefiltered data, allowing us to accurately apply complex filters with high performance and modest storage requirements.

2 Previous Work

The development of efficient and accurate filtering approaches is a long-standing problem, in many fields, with several decades of prior work. We focus on techniques that most closely motivate our approach, including recent high-performance filtering systems.

The naïve application of a filter to a signal of size n, where we assume n to be "unfolded" across the dimensionality of the signal/filter (i.e. an $\sqrt{n} \times \sqrt{n}$ image in 2D), has time complexity $\mathbf{O}(n^2)$. The *fast Fourier transform* (FFT) permits an optimization to $\mathbf{O}(n \log n)$ time for periodic domains. We consider the application of transformed (i.e. translated/rotated/scaled) *spatially-varying* filters with arbitrary size and shape, developing an $\mathbf{O}(s)$ time algorithm where s, the size of the IFD, is considerably smaller than n.

Gaussian-based filters are an important special case that includes standard Gaussian filtering, bilateral filtering [26], non-local means [3]. Burt [5] presented an efficient approximate algorithm for image processing with time complexity $\mathbf{O}(n)$, albeit for Gaussians of restricted extent. More recently, Chen et al. [8] introduced a low-memory, GPU accelerated approximate bilateral filtering system. Adams and colleagues [2, 1] consider high-dimensional Gaussian-based filtering and proposed novel data-structures to reduce filtering time complexity to $\mathbf{O}(dn' \log n')$ and $\mathbf{O}(d^2 n')$, where d is the dimensionality of the filter/signal, all with modest memory requirements (here n' is the size of a *single dimension* of the signal, as opposed to the "unfolded" size n). We also require modest memory usage for our prefiltered signal representation that, once computed, allows us to apply any number of (different) filters to a signal, all with time complexity linear in our decomposition size $s \ll n$.

Hierarchical filtering based on Laplacian Pyramids [6], wavelet decompositions [15], and bilateral filter pyramids [10] all decompose input signals according to scale or frequency to accelerate filtering. Convolution pyramids [9] design kernels for a multiscale filtering transform. As a special case, separable high-dimensional filters can be factored into outer products of lower-dimensional filters. For non-separable filters, a singular value decomposition can be used to approximate the outer product factorization [20].

We combine hierarchical representations with factorization into a filter decomposition theory based on isotropic kernels, with varying frequency bandwidths, aligned about a set of displacement vectors.

Unlike e.g. the subsampling operations used to accelerate hierarchical techniques [5], our decomposition remains numerically stable and *transform-invariant* (e.g. translation-, rotation- and scale-invariant). Transform-invariance is an important property in many filtering applications, for example when filtering spherical environment maps with complex BRDF kernels; here, without rotational-invariance, disturbing temporal and spatial artifacts can arise in the final shaded image.

Our approach is conceptually similar to Freeman and Adelson's steerable filters (SFs) [11]. SFs decompose (potentially anisotropic) filters exactly as a sum of oriented anisotropic filters, whereas we decompose using displaced copies of several identical isotropic filters. This allows us to first prefilter our signal using only each (canonically oriented) isotropic kernel and secondly

apply arbitrary spatially-varying filters by simply sampling into the prefiltered data. The benefits of our reduced amount of prefiltered data grow with the size and dimensionality of the domain. We can exchange performance and memory for accuracy, all with quantifiable error, by reducing the number of terms in our decomposition. Lastly, SFs only operate on band-limited filters whereas we handle filters with arbitrary bandwidth, extent and profile.

Gastal and Oliveira [12] evaluate filters at carefully chosen sample positions and interpolate these values in order to perform approximate high-dimensional filtering with time complexity $\mathbf{O}(dn)$. We factorize our filters as a sum of s displaced copies of b isotropic kernels however, unlike Gastal and Oliveira, any well-distributed set of displacement vectors will yield a suitable decomposition. We can optionally optimize these displacements to bias the decomposition towards higher-performance (sparsity) and higher-accuracy (better discrepancy of the displacements). Our approximation error is also easily quantifiable and, for band-limited filters, we can converge with a provably finite b (proportional to the bandlimit).

We derive a special-case of IFD for *spherical* signals in Section 4.2, which are used in several domains of computer graphics including image synthesis and geometry processing. Our spherical IFD generalizes recent work on zonal harmonic factorization [18], where we choose the zonal harmonic (ZH) subset of the spherical harmonics (SH) basis as our isotropic kernel set. We apply our spherical filtering to the problem of interactive rendering, where a view-rotated cosine-weighted BRDF at each pixel acts as an anisotropic filter on the incoming radiance distribution, resulting in the final shaded pixel intensity [21, 22].

3 Theory: Isotropic Filter Decomposition

We present our theory on filtering multi-dimensional signals using an *isotropic decomposition* of complex, arbitrarily sized, potentially anisotropic linear filters. Unlike many existing techniques, we allow spatially-varying filters that vary depending on the locations of the signal they are applied to. This complicates the problem and invalidates the often utilized, spatially-constant, definition of filtering where the application of a filter $g(\mathbf{x})$ to a signal $f(\mathbf{x})$ can be expressed *for all* $\mathbf{x}$ *in the domain* using a simple convolution as $\widehat{f}(\mathbf{x}) = (f \otimes g)(\mathbf{x})$.

Spatially-varying filtering is exposed by arbitrary transformations (e.g. translations, rotations, scales, etc.) applied to the filter during integration against the signal. We begin with an explicit formulation of this generalized filtering operation before defining and explaining how to use IFD for filtering.

Problem Definition. We define the operation of filtering as the double-product integration of a signal f and a filter g, potentially transformed by transform $\mathbf{T_x}$ instanced from a space of transforms $\mathcal{T}$ (i.e. translations, scales, and/or rotations), shifted across the integration domain Ω by a transform $\mathbf{S_x}$ from a transform group $\mathcal{S}$. The subscripts of $\mathbf{T_x}$ and $\mathbf{S_x}$ indicate their (potential) spatial-variance (i.e. dependence on $\mathbf{x}$). The filtered signal is thus

$$\widehat{f}(\mathbf{S_x}, \mathbf{T_x}) \quad = \quad \int_\Omega f(\mathbf{y})\; g\left(\mathbf{T_x}^{-1}\; \mathbf{S_x}^{-1}\; \mathbf{y}\right)\; d\mathbf{y} \tag{1}$$

The filtered value is by nature a function of the *transform* $\mathbf{S_x}$. Most of the time $\mathbf{S_x}$ is a rotation or translation that maps a fixed vector to a varying point $\mathbf{x} \in \Omega$, and so we permit a slight abuse in notation and often denoting $\widehat{f}(\mathbf{S_x})$ with $\widehat{f}(\mathbf{x})$. Traditional spatially-constant and untransformed filtering-as-convolution formulations correspond to having $\mathbf{T_x}$ be the identity and $\mathbf{S_x}$ a d-dimensional translation such that $\mathbf{T_x}(0) = \mathbf{x}$ (for $\dim(\Omega) = d$).

Computing Equation 1 is hard since g depends on the transform $\mathbf{T_x}$, which in turn varies spatially (on $\mathbf{x}$), and there is no (efficient) way to compute $\widehat{f}(\mathbf{x})$ other than e.g. brute-force numerical integration. This is the problem our work will resolve.

Isotropic Filter Decomposition (IFD). We decompose filters $g(\mathbf{x})$, across individual frequency bands l, as a weighted sum of s (displaced) isotropic kernels $G_l^m(\mathbf{x})$. Each displaced "basis kernel" in frequency band-l is a transformed duplicate of a *mother kernel* for band-l, $G_l(\mathbf{x})$: $G_l^m(\mathbf{x}) = G_l\left((\mathbf{S}_l^m)^{-1}\,\mathbf{x}\right)$, where $\mathbf{S}_l^m \in \mathcal{S}$ is the duplication transform. Mother kernels must additionally satisfy *isotropic-invariance*, meaning

$$\forall \mathbf{T} \in \mathcal{T} \quad G_l(\mathbf{x}) = G_l(\mathbf{T}\,\mathbf{x}) \ .$$

The displacement and isotropy properties permit a very efficient filtering formulation, which we detail below in Equation 3, and the full *isotropic filter decomposition* of $g(\mathbf{x})$ is written as

$$g(\mathbf{x}) = \sum_{l=1}^{b} \sum_{m} \lambda_l^m \ G_l^m(\mathbf{x}) \ . \tag{2}$$

We will present choices for the isotropic kernels G_l^m in the 2D image (Section 4.1) and spherical (Section 4.2) domains, as well as methods for performing the decomposition (i.e. solving for the λ_l^m).

Filtering with IFD. Substituting the IFD of g into Equation 1 and performing algebraic simplification yields our *IFD filtering equation*:

$$\begin{aligned} \widehat{f}(\mathbf{x}) &= \sum_{l,m} \lambda_l^m \int_{\mathbf{y}\in\Omega} f(\mathbf{y})\ G_l((\mathbf{S}_l^m)^{-1}\mathbf{T}_\mathbf{x}^{-1}\mathbf{S}_\mathbf{x}^{-1}\mathbf{y}))\ d\mathbf{y} \\ &= \sum_{l,m} \lambda_l^m (f \otimes G_l)(\mathbf{S_x}\mathbf{T_x}\mathbf{S}_l^m) \end{aligned}$$

where $\widehat{f_l} = f \otimes G_l$ is the convolution (over the group $\mathcal{S}$) of the signal with each (per-band) isotropic *mother kernel*. In practice, this convolution can be efficiently computed with e.g. FFT.

We note from Equation 3 that, once the $\widehat{f_l}$ functions are computed, evaluating the filtered signal $\widehat{f}$ at any location $\mathbf{x}$, and for **any transform $\mathbf{T_x}$**, simply requires the evaluation (and weighting) of $\widehat{f_l}$ at the transformed and displaced evaluation location $\mathbf{T_x}\ \mathbf{S}_l^m\ \mathbf{x}$. We illustrate this process for 2D images in Figure 1.

Choosing mother kernels G_l that satisfy the IFD displacement and isotropy conditions can be challenging, especially in the context of traditional function bases (e.g. wavelets, Fourier, sinusoids, etc.), albeit manageable (as we show in Section 4). We need only satisfy these, and no other, mathematical conditions; for example, while orthogonality of the G_l^m would aid in computing the weights λ_l^m, it is not strictly required (not the case for e.g. SFs [11]). These weights λ_l^m depend only on the filter g (in its canonical frame).

Section 4 will provide a concrete methodology for choosing the mother kernels G_l^m, solving for the decomposition weights λ_l^m, and performing filtering. If we precompute the $\widehat{f_l}$ for all l (a total of b terms) using e.g. FFT, our pretabulation time complexity is $\mathbf{O}(bn \log n)$ ($\mathbf{O}(bn)$ on the sphere) with storage $\mathbf{O}(bn)$, and filtering has time complexity $\mathbf{O}(s)$ where s is the total number of terms in e.g. Equation 2.

4 Designing Isotropic Filter Decompositions

We detail two example realizations of our theory of efficient filtering using IFD from Section 3, first in the 2D image domain (Section 4.1) and then in the spherical domain. Figure 1 overviews the entire procedure in the case of 2D images. The high-level advantages of filtering with IFD are two-fold: the reduced cost of pre-filtering signals with the mother kernels G_l at bandwidth levels l ($\mathbf{O}(bn \log n)$ time in 2D and $\mathbf{O}(sn)$ time on the sphere, and $\mathbf{O}(bn)$ storage for both), and the ability to efficiently and accurately apply complex spatially-varying filters using only $\mathbf{O}(s)$ look-ups into the pre-filtered data.

4.1 Isotropic Filter Decomposition for 2D Signals

We detail the process of IFD filtering 2D images, including: choosing mother kernels G_l that satisfy the isotropy and band-limited properties, computing isotropic kernels G_l^m and their displacement transforms $\mathbf{S}_\mathbf{x}$, solving for the IFD weights λ_l^m, and using the IFD to apply complex spatially-varying image filters. Here our domain $\Omega = \mathbb{R}^2$ is 2D plane (we treat color channels independently) with $\mathbf{x} \equiv (x, y)$, and we take $\mathcal{S} = \mathcal{R}_{2\mathrm{D}}$ to be the space of in-plane image rotations and $\mathcal{T} = \mathcal{T}_{2\mathrm{D}}$ to be the space of in-plane translations. Apart from isotropy and localization in frequency-space, another desirable property (especially for image processing) is spatial-locality. In an IFD, this property will cause sparsity in the weights λ_l^m because of the compactness of the isotropic kernels G_l^m.

2D Mother Kernels. Unlike the spherical domain (see Section 4.2), to our knowledge there are no non-trivial yet orthogonal 2D bases built atop displaced copies of rotationally symmetric functions. Rotationally symmetric wavelets [19] are a close candidate, but this basis is only invariant for a discrete set of rotations. Radial basis functions (RBFs) [4] are another possibility, however they do not generally exhibit compact spatial support and any filter decomposition based on them would yield mostly non-zero coefficients. Another option is a Mixture of Gaussians (MoG) that can be constructed to combine the frequency-locality of Gaussians with spatial-locality. For this reason, we choose MoGs as our 2D image IFD representation, with Gaussian mother kernels $G_l(\mathbf{x}) = e^{(-\|\mathbf{x}\|/\sigma_l)^2}$ with bandwidth σ_l, as summarized below.

	Frequency localization	Spatial localization	Rotational symmetry	Orthogonal basis
Fourier basis	✔	✘	✘	✔
2D Wavelets	✔	✔	✘	✔
RBFs	✘	✘	✔	✘
MoG	✔	✔	✔	✘

These mother kernel functions are clearly isotropic and invariant under transforms in $\mathcal{T}$, as required by IFD. While individually localized in space and frequency, we still require a MoG of these kernels that covers the entire domain Ω while remaining compact *across* frequencies and spatial locations. In other words, we need to carefully select bandwidths σ_l, and displacement transforms $\mathbf{S}_l^m$ (i.e. 2D vectors $\mathbf{x}_l^m \in \mathcal{S}$) for each frequency band l, to reconstruct any filter g with as few non-zero IFD weights λ_l^m as possible. Ideally, the IFD process for determining these unknowns should be efficient and permit trade-offs between filtering performance (i.e. sparsity of the weights) and the fitting error.

Isokernel Placement. We propose an isotropic kernel bandwidth distributions that doubles in frequency with each level l (similarly to Gaussian pyramids [5]) and, at each bandwidth level

l, we choose a distribution of displacement vectors $\mathbf{x}_l^m$ with density equal to the Nyquist limit of the bandwidth l; concretely, each band l will contain 4^l displacement vectors arrange on a $2^l \times 2^l$ uniform grid. The resulting isotropic IFD kernels are

$$G_l^m(\mathbf{x}) = e^{-(\|\mathbf{x}-\mathbf{x}_l^m\|/\sigma_l)^2} \text{ with } \sigma_l = 2^{-l} \ . \tag{3}$$

Figure 2 illustrates the displacement grid $\mathbf{x}_l^m$ and an arbitrary isotropic kernel $G_{l'}^{m'}(\mathbf{x})$ for $l = \{1, 2, 3\}$.

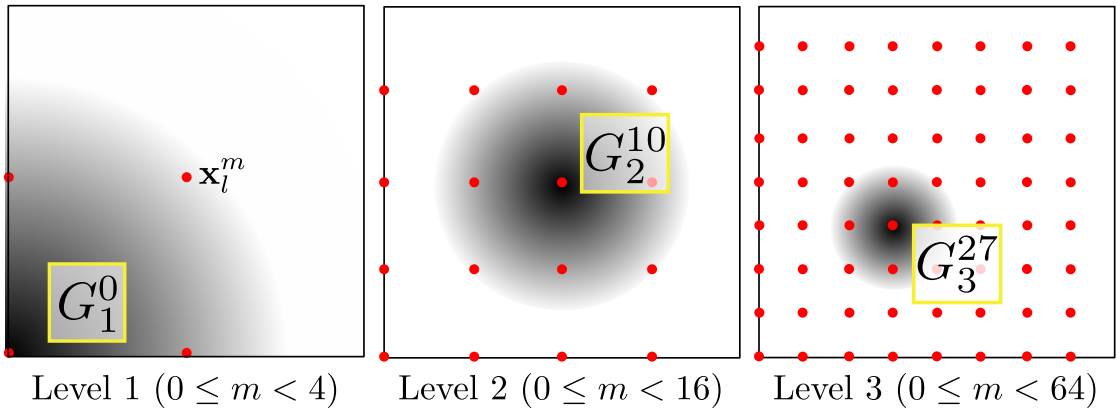


Figure 2: Displacements $\mathbf{x}_l^m$ for MoG kernels G_l^m used in our 2D image IFD, for frequency levels $l = \{1, 2, 3\}$.

IFD Weight Fitting. We now seek to decompose an arbitrary filter $g(\mathbf{x})$ as a linear combination of our isotropic kernels G_l^m. Since the G_l^m do not form an orthogonal basis, we choose an optimization approach for determining the IFD weights λ_l^m. The traditional approach would be to solve for the weight vector $\mathbf{\Lambda} = \{\lambda_0^0, \ldots, \lambda_b^{4^b}\}$ that minimizes an error-based objective function, such as e.g. $\underset{\mathbf{x}}{\text{argmin}} \| \sum_{l,m} \lambda_l^m G_l^m(\mathbf{x}) - g(\mathbf{x}) \|_{L_2}$. Since our choice of the isokernels $G_l^m(\mathbf{x})$ was (purposefully) dense in space and frequency, this L_2 optimization problem is likely to be under-constrained.

We exploit this to additionally solve for $\mathbf{\Lambda}$ that are also sparse: increased sparsity in $\mathbf{\Lambda}$ corresponds to the ability of our IFD to accurately represent a filter with as few coefficients as possible, which also yields a faster filtering implementation of Equation 3.

Recent work in compressive sensing [7, 27] has shown that optimizing the sparsity of a vector under linear constraints can be achieved by solving for $\mathbf{\Lambda}$ as

$$\underset{\mathbf{x}}{\text{argmin}} \| \sum_{l,m} \lambda_l^m G_l^m(\mathbf{x}) - g(\mathbf{x}) \|_{L_2} + \alpha \|\mathbf{\Lambda}\|_{L_1} \ , \tag{4}$$

where the first term maintains the fitting accuracy and the second term provably increases sparsity in Λ [7].

We solve this optimization problem using the SpaRSA algorithm [27]. Figure 3 illustrates an example of fitting an elongated Gaussian filter using our IFD, clearly demonstrating the sparsity in the IFD weights $\mathbf{\Lambda}$ as well as the fitting procedure's ability to continue reducing the L_2 component of the error even once an "optimal" sparsity is attained. Figure 4 illustrates the

convergence of the IFD procedure, both with of the final filtered image as well as in the filter approximation.

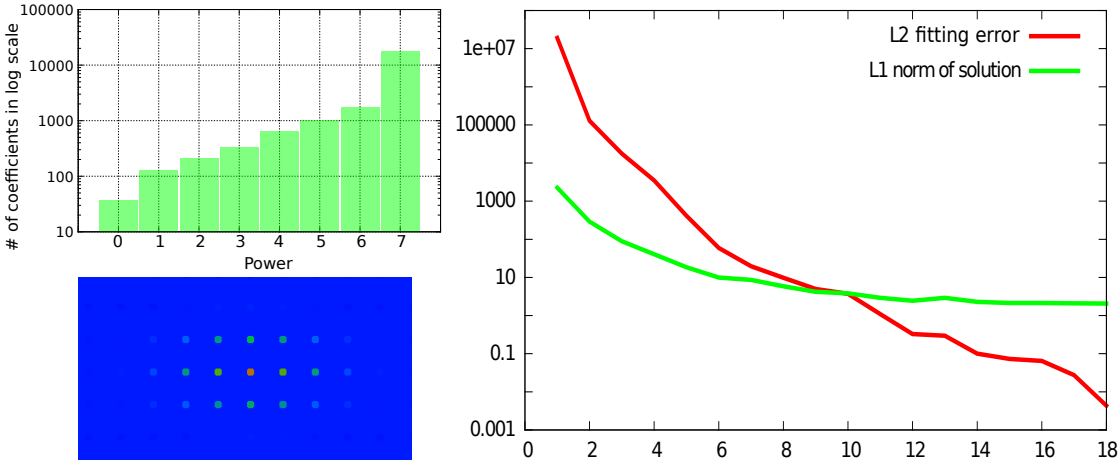


Figure 3: The log-scale histogram (*top left*) shows that most weight λ_l^m are small (weights in e.g. column five have magnitude between 10^{-6} and 10^{-5}). Due to the local support of the G_l^m, only local Gaussians end up being selected by SpaRSA (*botton left*).

4.2 Isotropic Filter Decomposition on the Sphere

We discuss the IFD process for filtering in the spherical domain, including: choosing mother kernels G_l and displaced isotropic kernels G_l^m, performing IFD by solving for the decomposition weights λ_l^m, and using the IFD to apply potentially-complex, spatially-varying spherical filters to spherical signals.

Here, the domain $\Omega = \mathcal{S}^2$ is the set of points on the unit sphere (all unit directions) $\mathbf{x} \equiv \omega = (\theta, \phi) = (x, y, z)$ where $x^2 + y^2 + z^2 = 1$, and we take $\mathcal{S} = \mathcal{R}$ to be the space of 3D rotations and $\mathcal{T} = \mathcal{R}_z$ to be the space of rotations around the z-axis.

Common approaches for representing spherical signals include spherical wavelets [24], frequency-space representations (e.g. SH), and spherical RBFs [16]. The first two do not fulfil isotropy requirements and, while the latter do, we are afforded an alternative capable of exact signal representation *and* rotational-invariance, built atop SH, that was not available to us in the 2D domain.

We build off SH for two reasons: first, because they are meet the frequency-localization we seek for our isotropic kernels $G_l^m(\omega)$ (and mother kernels $G_l(\omega)$); secondly, motivated by recent work on *zonal harmonic factorization* [18], we can isolate the isotropic *zonal harmonic* (ZH) subset of SH to serve as our mother kernel functions. We begin with a brief overview of SH before deriving a non-orthogonal basis that satisfies the IFD displacement and isotropy requirements.

Spherical Harmonics. Real SH basis functions are defined as:

$$y_l^m(\omega) = \begin{cases} \sqrt{2} N_l^m P_l^m(\cos\theta) cos(m\phi) & \text{if } m > 0 \\ N_l^m P_l^0(\cos\theta) & \text{if } m = 0 \\ \sqrt{2} N_l^{-m} P_l^{-m}(\cos\theta) sin(-m\phi) & \text{if } m < 0 \end{cases},$$

Figure 4: We demonstrate the convergence of IFD filtered images with an increasing number of terms s. Ground truth is computed numerically with a FFT. Our multi-frequency IFD MoG approximation (*bottom-right*) of an anisotropic kernel (*top-right*) obtained with the SpaRSA algorithm, and the reconstructed kernel and error using 200 coefficients.

where the basis functions are indexed by frequency band l and function index m, N_l^m is a normalization term, and P_l^m are the Associated Legendre Polynomials. A full order-N SH expansion of a function includes all bands $l<N$, and the $2l+1$ basis functions (with $-l\leq m\leq l$) in each band, for a total of $s=N^2$ terms.

Each band l contains functions of fixed bandwidth in the (spherical) frequency domain, and the $2l+1$ band-l basis functions span the space of l-band-limited spherical functions. Unfortunately, we cannot directly utilize band-l basis functions to satisfy IFD since they are not all isotropic w.r.t. rotations around z, $\mathbf{R}_z \in \mathcal{T} = \mathcal{R}_z$, however Nowrouzezahrai et al. [18] show that the same band-l space is spanned by rotated copies of the band-l zonal harmonic y_l^0, which happens to be isotropic w.r.t. $\mathcal{T}=\mathcal{R}_z$ (circularly symmetric).

We thus choose band-l ZHs as our mother kernels: $G_l(\omega) = y_l^0(\omega)$.

IFD with Rotated Zonal Harmonics. Given our choice of mother kernel functions, the two remaining steps to complete the spherical IFD are: determining the duplication transforms $\mathbf{S}_l^m \equiv \mathbf{R}_l^m \in \mathcal{S}$ for our isotropic kernels $G_l^m(\omega) = y_l^0((\mathbf{R}_l^m)^{-1}\ \omega)$, and computing the decomposition weight λ_l^m of the IFD. For the latter task, we will see shortly that our choice of ZHs for the mother kernel will permit an analytic solution for the decomposition weights.

To simplify our exposition, we first consider a single band l and later extend our formulation (trivially) to order-N reconstructions.

Let $\mathbf{Z} = \{\mathbf{z}_l^0, \ldots, \mathbf{z}_l^{2l+1}\}$ be a set of $2l+1$ unit vectors in $\Omega\equiv\mathcal{S}^2$. We denote the 2D rotation that aligns direction $\mathbf{z}_l^i$ to z-axis as $\mathbf{R}_l^i \in \mathcal{S}$, which allows us to define our isotropic kernels as

$$G_l^m(\omega) = y_l^0\left(\mathbf{R}_l^m\ \omega\right) \quad \text{with} \quad \mathbf{R}_l^m = \left[\mathbf{e}_1^\mathbf{T}, \mathbf{e}_1^\mathbf{T}, (\mathbf{z}_l^m)^\mathbf{T}\right] \ . \tag{5}$$

Any non-degenerate distribution of directions $\mathbf{Z}$ yields an isotropic kernel basis (with elements as in Equation 5) that spans l-band-limited functions [18], satisfying IFD's frequency-localization, isotropy, and displacement properties.

Given the isotropic kernels G_l^m of our IFD (with their corresponding rotations direction $\mathbf{z}_l^m$), we need only determine weights λ_l^m to decompose our spherical filter g according to Equation 2. We first express the l-band-limited component g_l of g with its expansion in the band-l SH basis as $g_l(\omega) = \sum_{m=-l}^{l} c_l^m\ y_l^m(\omega)$. Here, the band-$l$ projection coefficients $c_l^m = \int_\Omega g(\omega)\ y_l^m(\omega) d\omega$ are computed using standard analytic or numerical integration, depending on the form of g (i.e. we support both analytic and tabulated filters; see Section 5.2).

Once (band-l) SH projection coefficients of the filter are computed, we devise a linear mapping between the c_l^m and IFD weights λ_l^m by leveraging the SH addition theorem [13] that expresses

rotated (band-l) ZHs as a weighted sum of (band-l) SH functions:

$$y_l^0(\mathbf{R}_l^m \omega) = \sqrt{4\pi/2l+1} \sum_m y_l^m(\mathbf{z}_l^m)\ y_l^m(\omega)\ . \tag{6}$$

Combining Equations 6 and 5 into a matrix equation across all m in band-l, we arrive at the following linear mapping:

$$\mathbf{c}_l = \mathbf{M}_l\ \mathbf{\Lambda}_l\ ,\ \text{with}\ (\mathbf{M}_l)_{ij} = \sqrt{4\pi/2l+1}\ y_l^i(\mathbf{z}_l^j)\ , \tag{7}$$

where $\mathbf{c}_l = [c_l^{-l}, \ldots, c_l^l]$ is a vector of g_l's band-l SH projection coefficients, and $\mathbf{\Lambda}_l = [\lambda_l^{-l}, \ldots, \lambda_l^l]$ is a vector of (unknown) IFD weights [18].

Reviewing spherical l-band-limited IFD, we begin by distributing $2l+1$ directions $\mathbf{z}_l^m$ (e.g. using low-discrepancy patterns on the sphere), and computing the elements of the matrix $\mathbf{M}_l$. The band-l SH projection coefficient vector $\mathbf{c}_l$ of the filter g is computed, and the IFD weights are solved as $\mathbf{\Lambda}_l = \mathbf{M}_l^{-1}\ \mathbf{c}_l$. The resulting band-$l$ IFD is

$$g_l(\omega) = \sum_{m=-l}^{l} \lambda_l^m\ G_l^m(\omega) = \sum_{m=-l}^{l} \lambda_l^m\ y_l^0(\mathbf{R}_l^m\ \omega)\ . \tag{8}$$

Note that, unlike the sampling locations of Gastal and Oliveira [12], we can choose any non-degenerate distribution of the $\mathbf{z}_l^m$ (guaranteeing a non-singular $\mathbf{M}_l$ [14]).

Extension to Order-N. The full $s = N^2$-term spherical IFD is

$$g(\omega) = \sum_{l=0}^{N-1} \sum_{m=-l}^{l} \lambda_l^m\ G_l^m(\omega) = \sum_l \sum_m \lambda_l^m\ y_l^0(\mathbf{R}_l^m\ \omega)\ , \tag{9}$$

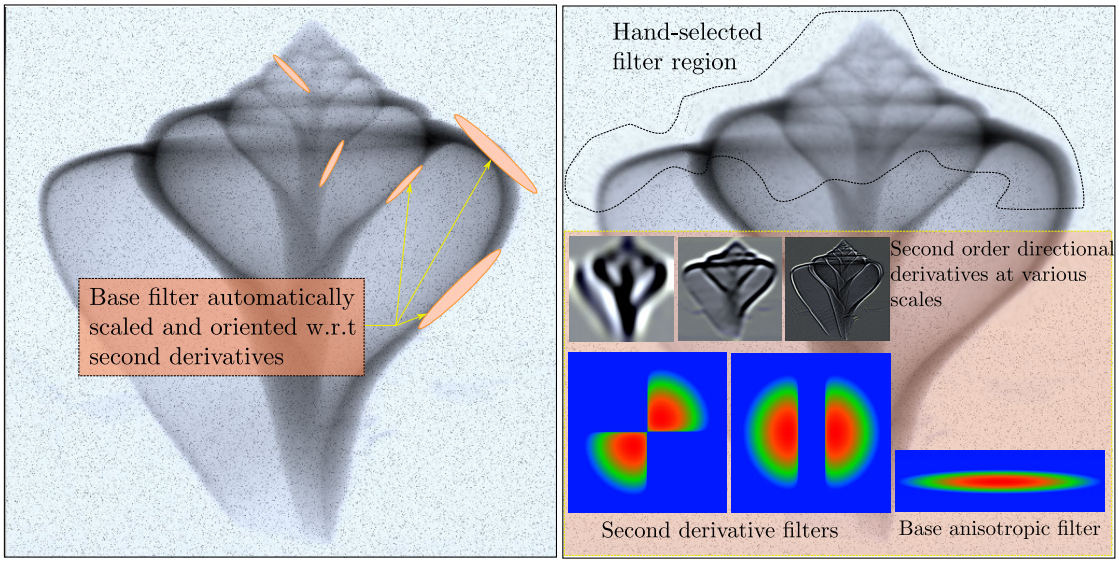
and can be easily formed as a simple combination across the band-l spherical IFDs (Equation 8) for every band $l < N$. We are able to perform this concatenation since each band-l isotropic kernel set spans an isolated frequency-bandlimited space and linear reconstruction across bands is L_2-optimal from the properties of SH.

We additionally share the displacement directions $\mathbf{z}_l^m$ *across* bands using the "lobe sharing" scheme presented by Nowrouzezahrai et al. [18], where the e.g. $-l \leq m \leq l$ directions for the band-l IFD are an identically indexed subset of the $-(l+1) \leq m \leq (l+1)$ directions from the band-$(l+1)$ IFD. We can similarly rewrite the system of equations in Equation 7 across all bands $l < N$, $\mathbf{f} = \mathbf{M}\ \mathbf{\Lambda}$, where the three terms are order-N generalizations of their band-l counterparts. As such, $\mathbf{M}$ (and $\mathbf{\Lambda}$) have block-diagonal structure with $(2l+1)\times(2l+1)$ matrix sub-blocks along the diagonal.

Efficient Spherical Filtering with IFD. Recall from Equation 3 that the input signal must be convolved (under group $\mathcal{S}$) against the IFD's mother kernel functions G_l in order to perform filtering. We (pre)filter the input signal f for each band-l in order to filter using Equation 3. This means we convolve the signal f with the band's mother kernel $G_l = y_l^0$. Since G_l is isotropic (by construction), the result of the convolution is a spherical function [1].

Our choice of mother kernel's affords an interesting accelerated $\mathbf{O}(n)$ time complexity algorithm for computing each of the b (pre)convolutions. Given the SH projection coefficients of *the*

[1] Here, similarly to Section 4.1, we abuse notation and interchange transforms in $\mathcal{S}$ with elements of Ω.



Original (noisy) image Example ongoing-filtering session

Figure 5: Our proof-of-concept user-guided image-smoothing tool applies an anisotropic Gaussian filters with a size and orientation computed on the fly based on an estimate of the local Hessian. Both the Hessian computation and the final anisotropic edge-respecting image smoothing are executed as filtering operations using IFD. The filtering is able to eliminate the noise in the image while maintaining the multi-scale details of the underlying image, all while the user interactively modifies the editing footprint scale.

signal, f_l^m, we can apply the SH convolution theorem [23] and compute the SH coefficients of the prefiltered function as

$$\widehat{f}_l^m = \sqrt{4\pi/2l+1} f_l^m \tag{10}$$

This interesting result means that preconvolving the signal against our RZH mother kernel corresponds to a simple per-band scaling of its frequency spectrum.

The preconvolved function can be evaluated directly (as necessary for filtering in Equation 3) by either pretabulating it (e.g. in a cubemap) or simply computing its SH expansion on the fly as

$$\widehat{f}_l(\omega) = \sum_m \widehat{f}_l^m \; y_l^m(\omega) = \sqrt{4\pi/2l+1} \sum_m f_l^m \; y_l^m(\omega) \; . \tag{11}$$

Drawing parallels to steerable filters, one of the works more closely related to ours, spherical SFs would amount to a mapping from SH basis functions onto (potentially rotated) SH basis functions, requiring N^2 (instead of N) preconvolved lookup functions.

We apply spherical IFDs to realistic interactive rendering in Section 5.2, where the environment lighting acts as a signal that is filtered by a spatially- and orientation-varying (in $\mathcal{T}$) BRDF.

5 Applications and Results

After a $\mathbf{O}(bn \log n)$ prefiltering stage ($\mathbf{O}(bn)$ in the spherical setting; requiring $\mathbf{O}(bn)$ storage), we are able to apply and reapply any number of complex filters (represented by their IFD) to

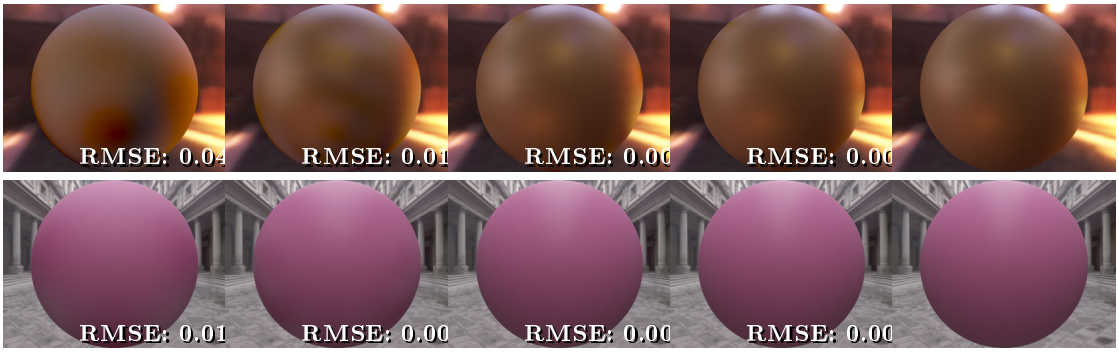


Figure 6: Convergence of spherical IFD shading for increasing $b = N; s = N^2$. Top to bottom: glossy (narrow) and diffuse BRDF filter.

the original signal. Our decomposition is transform-invariant (i.e. translation-, rotation-, scale-invariance) and filters can have arbitrary size/extent, be spatially-varying, arbitrarily oriented, and have anisotropic profiles. We demonstrate our approach on interactive image processing tasks and a common problem in realistic rendering: computing the shading of a scene with complex BRDFs and an environment light i.e. captured from the real world.

5.1 Image Editing and Anisotropic Texture Filtering

There are several possible applications of to image analysis/editing, such as computing local image descriptors as the response to densely rotated anisotropic filters, performing data-dependent image filtering operations (e.g. edge-aware anisotropic diffusion), or implementing efficient interactive filtering tools for image editing.

As a proof-of-concept we implement a custom hand-brush image editing tool that performs anisotropic filtering of an input image based on the estimate of the Hessian matrix of the image at the current pixel. The Hessian gives the principal directions of the second derivatives and, therefore, provides a meaningful estimate of the orientation and size of an spatially-varying, edge-respecting an anisotropic smoothing filter (we use a very anisotropic Gaussian kernel).

This tool uses our 2D image IFD filtering twice. First, a user selects/hovers over a region, and we use IFD to estimate the Hessian at the input location by applying a second-derivative filter at the user defined scale (see Figure 5); concretely, we decompose the two filters ($x - x$ and $x - y$ second derivatives), and we apply a rotated version of the $x - x$ filter to obtain the $y - y$ second derivative. Secondly, once computed, we perform an on-the-fly eigenanalysis of the (spatially-varying) Hessian in order to approximate the excentricity and orientation of our anisotropic Gaussian filter, which we also apply (after scaling and rotation) using IFD. An example of filtering session is depicted in Figure 5, and in our submission video. Since filtering performance does not depend on the original filter nor the size of the signals, our filtering tool runs interactively for images of any size and filters of arbitrary scale. This results in a seamless user experience.

5.2 Prefiltering for Realistic Interactive Rendering

We apply spherical IFDs to a difficult problem in interactive realistic rendering: shading scenes with environment lighting and spatially-varying, potentially-anisotropic reflectance (BRDFs).

Linear-cost Environment Map Shading with Arbitrary BRDFs. The reflection of distant spherical incident lighting $\omega \mapsto E(\omega)$ at a surface point x, with BRDF $\rho(x, \omega_o, \omega)$, towards a viewer ω_o is

$$L(x, \omega_o) = \int_{\mathcal{S}^2} E(\omega)\, \rho(x, \mathbf{R_n}\,\omega, \mathbf{R_n}\,\omega_o)\, \max(\cos\theta, 0)\, d\omega\,, \tag{12}$$

where the incident and outgoing directions ω and ω_o are expressed in global (world) coordinates, the BRDF ρ is parameterized in the local (surface) coordinate frame at x, and the rotation $\mathbf{R_n} \in \mathcal{S} \equiv \mathcal{R}$ transforms from global- to local-coordinates at x (where $\mathbf{n}$ is the normal at x and θ is the angle between $\mathbf{n}$ and ω).

We treat each spatially-varying view-slice of the cosine-weighted BRDF as a filter, $g(\omega) = \rho(x, \omega, \omega_o)\, \max(\cos\theta, 0)$, and the environment map as our signal $f(\omega) = E(\omega)$. Following the process outlined in Section 4.2, we choose $2N{+}1$ well-distributed directions $\mathbf{z}_l^m$—remember that directions are shared accross bands l—and compute IFD weights for an ensemble of filters (one for each BRDF slice, and each point x for spatially-varying BRDFs):

$$\rho(x, \omega, \omega')\, \max(\cos\theta, 0) = \sum_l \sum_m \lambda_l^m(x, \omega')\; G_l^m(\omega)\;. \tag{13}$$

Substituting Equation 13 into Equation 12, or applying Equation 2, yields the IFD equation for shading (we drop x for brevity, and we drop the l subscript on IFD displacement directions $\mathbf{z}_l^m \equiv \mathbf{z}^m$ to make explicit that "lobe sharing" permits us to store a single set of directions *across* frequency bands l):

$$\begin{aligned} L(\omega_o) &= \sum_{l=0}^{b-1} \sum_{m=-l}^{l} \lambda_l^m(\mathbf{R_n}\,\omega_o) \int_{\mathcal{S}^2} E(\mathbf{R_n^T}\,\omega)\; G_l^m(\omega)\; d\omega \\ &= \sum_{l,m} \lambda_l^m(\mathbf{R_n}\,\omega_o)\; \underbrace{\left(E \otimes y_l^0\right)}_{\widehat{f}_l(\omega) = \widehat{E}_l(\omega)} (\mathbf{R_n^T}\,\mathbf{z}^m) \\ &= \sum_{l,m} \sqrt{4\pi/2l{+}1}\; \lambda_l^m(\mathbf{R_n}\,\omega_o) \sum_{k=-l}^{l} e_l^k\, y_l^k(\mathbf{R_n^T}\,\mathbf{z}^m)\;, \end{aligned} \tag{14}$$

where the number of frequency bands $b{=}N$ and e_l^m are SH coefficients of the lighting E.

The (spatially- and view-varying) filter weights $\lambda_l^m(x, \omega')$ can be pre-tabulated for data-driven BRDFs (e.g. the MERL BRDF dataset [17]), or computed analytically from their analytic SH coefficients for phenomenological BRDFs e.g. Lambertian and Phong BRDFs. We illustrate rendering results for several environment maps with spatially-varying and anisotropic BRDFs (Figures 9 and 8) of varying glossiness/filter-extent (Figure 7), as well as demonstrating convergence on large filters with ground-truth computed using numerical integration (Figure 6).

6 Implementation and Discussion

Our implementations of IFDs for the applications in Section 5 can be almost fully described by following "naïvely" implementing algorithms according to the steps outlined in Section 4,

Figure 7: Spherical IFD with varying filter widths: we illustrate diffuse (wide; *top*) and glossy (narrower; *glossy*) filters for various scenes and lighting environments. All renderings were captured in real-time with performance ranging from 30 to 70 Hz.

Figure 8: We vary the BRDF filter anistropy from isotropic/diffuse (*left*) to anisotropic (*right*). IFD performance is independent of the filter profile, and all renderings were captured at 30-70 Hz.

ultimately leading to the implementation of IFD filtering using Equation 3. This simplicity permits trivial parallelization of the filtering algorithm, for example on the GPU (which we do for one of our applications). We summarize all additional implementation-specific details below before discussing our performance/memory requirements and scalability, and finally ending with a discussion on limitations.

Implementation Details. For our environment map shading with BRDF filters we implement Equation 14 in a GLSL shader. For data-driven BRDF filters (from the MERL BRDF dataset [17]) we pre-tabulate the IFD weights per (cosine-weighted) BRDF view-slice $\lambda_l^m(\omega_o)$, for each discrete BRDF, in a cubemap indexed by the view direction ω_o (in local surface coordinates). For analytic BRDFs (e.g. Lambertian, Phong) we hardcode analytic IFD weights. Equation 5 is not a numerically stable method for evaluating high-order SH basis functions. We designed stable recurrence formulae for these evaluations based on the principles suggested by

Figure 9: IFD supports spatially-varying filters: a texture modulates the anisotropy (*left*,*middle*) and albedo (*right*) of a complex BRDF model that combines diffuse (wide) and anisotropic BRDF profiles. Again, rendering (IFD filtering) is independent of the filter complexity and all images were captured at 30-70 Hz.

Sloan [25].

A subtle, albeit important, shader implementation trade-off exists: when evaluating the preconvolved signals $\widehat{f_l}(\omega)$ (Equation 11) in Equation 14, we can either pre-evaluate and store the functions in cubemaps (Equation 14, middle line) of or explicitly compute the band-l SH expansion (Equation 14, last line). This amounts to a texture-lookup vs. ALU opcode trade-off in the shader design, and we found that the pre-evaluated cubemaps were faster for our use-cases but that at higher-order the cubemap resolutions had to be increased to avoid numerical imprecisions.

Performance and Memory. All our results were captured on an Intel Xeon 1.2GHz with 24 GB of RAM and an NVIDIA GTX 670 with 2 GB of VRAM. We provide performance statistics both in Hz and *filtering operations per second* (FOPS), and our video results were all captured interactively. All of our applications are single-threaded and Table 1 details the performance and memory statistics of our shading application. In the case of anisotropic image

Interactive Environment Shading with BRDFs (Section 5.2)

$s =$	4^2	8^2	12^2	16^2
$n =$	6×128^2	6×256^2	6×256^2	6×512^2
Speed [Hz]	68	59	46	29
Speed [FOPS]	53.4M	46.4M	36.2M	22.8M
Mem. [MB]	6	42	96	672
Prefiltering [sec]	5	9	14	27

Table 1: Performance, memory and scalability statistics.

filtering, which was implemented in CUDA, every filtering result (in the paper and video) was computed in less than 1ms. Figures 4 and 5 illustrate various filtering results using this technique. For environment light shading with complex BRDFs, we notice a similar (sub)linear connection between s and performance, and we demonstrate convergence to ground truth filtering in Figure 6, and rendering results with spatially-varying and anisotropic BRDFs and varying BRDF glossiness (corresponding to different BRDF filter sizes) in Figures 7, 9 and 8.

Discussion and Limitations. Our choice to not require orthogonal displace isotropic kernels in the IFD is not without its shortcomings. In this case, most of the G_l^m do not integrate to 0, meaning that adding higher frequency terms could change the average value of the reconstructed filter.

There are many seemingly arbitrary choices in the design of our 2D image IFDs and, while our particular choice leads to a suitable representation, we are confident that our methodology can easily yield alternative suitable IFDs. Indeed, we have experimented with other options for mother kernels e.g. based on radial derivatives of Gaussians, polynomials in polar coordinates, etc. Each technique yielded an efficient and accurate IFD, and we opted to expose our final MoG choice due to its simplicity (both mathematically, and implementation-wise) and effectiveness. We make no claims on its optimality, and investigating the design of *optimal* IFDs for a fixed domain remains an interesting open direction of future research. In contrast, our choice of mother kernels in the spherical domain was motivated based on recent developments in spherical signal processing; however, our spherical IFD inherits the limitations of SH: as the angular extent of the filter reduces (increasing it angular frequency content), larger s are required to properly reconstruct the filter. Note, however, that this theoretical limitation is balanced in a practical sense as these smaller filters can be efficiently filtered using even the brute-force solution as the size of the non-zero regions in the filter become smaller. We gracefully handle the case of complex, spatially-varying anisotropic filters with *larger* extents, that are traditionally hard to compute numerically with any other technique.

7 Conclusion and Future Work

We develop a general theory of filtering based on multi-scale isotropic displaced kernels. Unlike many other representations, ours has a combination of several important properties for efficient filtering: it is compact, transform-invariant, has a controllable error vs. performance behavior, has a modest footprint (that does not scale geometrically with the dimensionality of the signal), and is very efficient and trivially parallelizable.

With IFD, we can decompose spatially-varying filters with arbitrary domain, extent, anisotropy and transform (i.e. position/orientation/scale) with an s-term weighted sum of b displaced isotropic kernels. After computing b prefiltered signals, requiring $\mathbf{O}(bn \log n)$ time ($\mathbf{O}(bn)$ for the special-case of the sphere) and $\mathbf{O}(bn)$ storage, we can apply any number of filtering operations with any number of different filters in $\mathbf{O}(s)$ time, where $b \leq s \ll n$. Here note that filtering time grows linearly only with the size of our representation which is typically orders of magnitude smaller than the size of the signal. We detail the practical implementation of our theory in the 2D and spherical image domains, applying IFD filtering to difficult problems interactive image processing and realistic rendering.

Several interesting unanswered theoretical questions remain. Whether an IFD-like decomposition with can be devised with orthogonal filtering kernels remains an open theoretical question, and finding "optimal" isotropic kernels for different specific filtering domains is an open question.

In contrast, there are also many potential future applications of IFD. For example, the interactive rendering of anti-aliased mirror reflections: given a geometrically complex, metallic/mirror-like object that reflects its environment, as the viewer zooms-out of the scene each pixel will project onto a larger and more complex region of the object. The correct final pixel intensity should average all reflected scene points, corresponding to a warping of the perfect mirror reflection cone based on the warped pixel footprint projected onto the surface. Current solutions simply mipmap the environment texture and sample from coarser levels, resulting in temporally disturbing and incorrect reflection effects. With IFD, a canonical pixel footprint filter can be

decomposed and warped on-the-fly (at each pixel) in order to properly sample (from preconvolved version of) the environment texture. Other applications of IFD to e.g. 3D tomographic data filtering, volume rendering and even animation editing are interesting practical directions of future work.

A Isotropic Filter Decomposition

We derive Equation 3 by reporting Equation 2 in Equation 1:

$$\begin{aligned}
\widehat{f}(\mathbf{S_x}, \mathbf{T_x}) &= \sum_{l,m} \lambda_l^m \int_{\mathbf{y}\in\Omega} f(\mathbf{y})\ G_l^m(\mathbf{T_x}^{-1}\mathbf{S_x}^{-1}\mathbf{y}))\ d\mathbf{y} \\
&= \sum_{l,m} \lambda_l^m \int_{\mathbf{y}\in\Omega} f(\mathbf{y})\ G_l((\mathbf{S}_l^m)^{-1}\mathbf{T_x}^{-1}\mathbf{S_x}^{-1}\mathbf{y}))\ d\mathbf{y} \\
&= \sum_{l,m} \lambda_l^m (f \otimes G_l)(\mathbf{S_x}\mathbf{T_x}\mathbf{S}_l^m)
\end{aligned}$$

References

[1] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum (EG 2010 Proceedings*, 2010.

[2] Andrew Adams, Natasha Gelfand, Jennifer Dolson, and Marc Levoy. Gaussian kd-trees for fast high-dimensional filtering. *ACM Trans. Graph.*, 28(3):21:1–21:12, July 2009.

[3] A. Buades, B. Coll, and J. M. Morel. A Review of Image Denoising Algorithms, with a New One. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[4] Martin D. Buhmann. *Radial Basis Functions: Theory and Implementations*, volume Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Shaftesbury Road Cambridge, CB2 8BS, United Kingdom, 2003.

[5] P. J. Burt. Fast filter transform for image processing* 1. *Computer graphics and image processing*, 1981.

[6] Peter J. Burt and Edward H. Adelson. Readings in computer vision: issues, problems, principles, and paradigms. chapter The Laplacian pyramid as a compact image code, pages 671–679. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[7] E. J. Candès and T. Tao. Signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 8(59):1207—1223, 2005.

[8] Jiawen Chen, Sylvain Paris, and Frédo Durand. Real-time edge-aware image processing with the bilateral grid. *ACM Trans. Graph.*, 26(3), July 2007.

[9] Zeev Farbman, Raanan Fattal, and Dani Lischinski. Convolution pyramids. *ACM Trans. Graph.*, 30(6):175:1–175:8, December 2011.

[10] Raanan Fattal, Maneesh Agrawala, and Szymon Rusinkiewicz. Multiscale shape and detail enhancement from multi-light image collections. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), August 2007.

[11] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(9):891–906, September 1991.

[12] Eduardo S. L. Gastal and Manuel M. Oliveira. Adaptive manifolds for real-time high-dimensional filtering. *ACM TOG*, 31(4):33:1–33:13, 2012. Proceedings of SIGGRAPH 2012.

[13] E.W. Hobson. *The theory of spherical and ellipsoidal harmonics.* The University press, 1931.

[14] C. Lessig, T. de Witt, and E. Fiume. Efficient and accurate rotation of finite spherical harmonics expansions. *Journal of Computational Physics*, 231(2):243 – 250, 2012.

[15] Stphane Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way.* Academic Press, 3rd edition, 2008.

[16] K.V. Mardia and P.E. Jupp. *Directional Statistics.* Wiley Series in Probability and Statistics. Wiley, 2009.

[17] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Transactions on Graphics*, 22(3):759–769, July 2003.

[18] Derek Nowrouzezahrai, Patricio Simari, and Eugene Fiume. Sparse zonal harmonic factorization for efficient sh rotation. *ACM Transactions on Graphics*, 2012.

[19] John William Parker. *Band-Limited Wavelets with Rotational Symmetry.* PhD thesis, January 1994.

[20] Pietro Perona. Deformable kernels for early vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(5):488–499, May 1995.

[21] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 497–500, New York, NY, USA, 2001. ACM.

[22] Ravi Ramamoorthi and Pat Hanrahan. Frequency space environment map rendering. *ACM Trans. Graph.*, 21(3):517–526, July 2002.

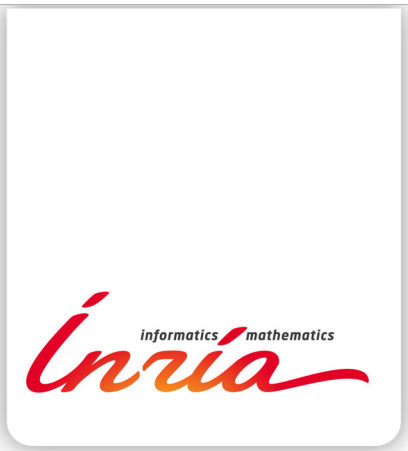
[23] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for reflection. *ACM Trans. Graph.*, 23(4):1004–1042, October 2004.

[24] Peter Schröder and Wim Sweldens. Spherical wavelets: efficiently representing functions on the sphere. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 161–172, New York, NY, USA, 1995. ACM.

[25] Peter-Pike Sloan. Stupid spherical harmonics (sh) tricks, 2008.

[26] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pages 839–, Washington, DC, USA, 1998. IEEE Computer Society.

[27] Stephen J. Wright, Robert D. Nowak, and Mário A. T. Figueiredo. Sparse reconstruction by separable approximation. *Trans. Sig. Proc.*, 57(7):2479–2493, July 2009.



RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399