

Efficient and Accurate Spherical Kernel Integrals using Isotropic Decomposition

C.Soler, M.Bagher, D.Nowrouzezahrai

INRIA / Grenoble University Microsoft Corporation University of Montreal

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Outline



Motivation

- Algebra of rotated zonal bases
- Applications

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Spherical kernel integrals





Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Spherical kernel integrals





Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Spherical kernel integrals





Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Spherical kernel integrals







Soler, Bagher, Nowrouzezahrai

Spherical kernel integrals... are expensive!





$$\widehat{f}(\mathbf{R}) = \int_{\Omega} k(\mathbf{R}^{-1}\omega) f(\omega) d\omega$$

pre-computation?

- rotation space is huge! One simply cannot tabulate R
- storage is a deal-breaker for GPU
- what if k also "depends" on R?

Soler, Bagher, Nowrouzezahrai

Spherical kernel integrals... are expensive!





pre-computation?

- rotation space is huge! One simply cannot tabulate R
- storage is a deal-breaker for GPU
- what if k also "depends" on R?
- use a basis of functions?

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Motivation Rotated zonal bases Applications SIGGRAPH 9219 SIQ 20 Desired characteristics for a "good basis" KDBE Sparsity S.Gaussians S.Wavelets 0 S.Harmonics

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

 Sparsity
 Consistency

 S.Gaussians
 Image: Consistency

 S.Wavelets
 Image: Consistency

S.Harmonics

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals



Motivation Rotated zonal bases Applications

Motivation Rotated zonal bases Applications
Desired characteristics for a "000



Desired characteristics for a "good basis"



	Sparsity	Consistency	Orthogonality	
S.Gaussians	<	\otimes	0	
S.Wavelets	✓	\checkmark		
S.Harmonics	\otimes	\checkmark		(And)

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Desired characteristics for a "good basis"



	Sparsity	Consistency	Orthogonality	Filter-friendly
S.Gaussians	~	${igwedge}$	0	
S.Wavelets	√	\checkmark		0
S.Harmonics	\mathbf{O}	\checkmark		0

Soler, Bagher, Nowrouzezahrai

Desired characteristics for a "good basis"



	Sparsity	Consistency	Orthogonality	Filter-friendly
S.Gaussians	~	${igwedge}$	0	
S.Wavelets	-	\checkmark		Ø
S.Harmonics	\otimes	\checkmark		0
Isotropic Spherical Decomposition	\otimes	\checkmark		

Isotropic Spherical Decomposition (ISD)



 $\hat{f}(\mathbf{R}) = \sum_{m} \lambda_{m} \int_{\Omega} \phi_{m}(\mathbf{R}^{-1}\,\omega) f(\omega) d\omega$

 ϕ_m

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Isotropic Spherical Decomposition (ISD)



 \mathbf{z}_m





Soler, Bagher, Nowrouzezahrai

 ϕ_m

Isotropic Spherical Decomposition (ISD)



 \mathbf{z}_m





Soler, Bagher, Nowrouzezahrai

 ϕ_m

Efficient and Accurate Spherical Kernel Integrals

Isotropic Spherical Decomposition (ISD)



 \mathbf{z}_{m_j}

$\hat{f}(\mathbf{R}) = \sum_{m} \lambda_m (\phi_m \otimes f)(\mathbf{R} \mathbf{z}_m)$



Soler, Bagher, Nowrouzezahrai

 ϕ_m

Efficient and Accurate Spherical Kernel Integrals

Isotropic Spherical Decomposition (ISD)



 \mathbf{z}_m

$$\hat{f}(\mathbf{R}) = \sum_{m} \lambda_m (\phi_m \otimes f)(\mathbf{R} \mathbf{z}_m)$$



use isotropic basis functions
 Reduces pre-convolution space by one dimension!

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Isotropic Spherical Decomposition (ISD)







- use isotropic basis functions
 Reduces pre-convolution space by one dimension!
 make them rotated copies of each others
- make them rotated copies of each others

Soler, Bagher, Nowrouzezahrai

Isotropic Spherical Decomposition (ISD)



$$\hat{f}(\mathbf{R}) = \sum_{m} \lambda_m (\phi \otimes f)(\mathbf{R} \mathbf{z}_m)$$



- use isotropic basis functions
 Reduces pre-convolution space by one dimension!
 make them retated copies of each others
- make them rotated copies of each others

Soler, Bagher, Nowrouzezahrai

Isotropic Spherical Decomposition (ISD)



$$\hat{f}(\mathbf{R}) = \sum_{m} \lambda_m (\phi \otimes f)(\mathbf{R}\mathbf{z}_m)$$



use isotropic basis functions
 Reduces pre-convolution space by one dimension!

 make them rotated copies of each others
 Reduces number of pre-convolution by one order!
 Removes the need for a rotation formula!

Isotropic Spherical Decomposition (ISD)



$$\hat{f}(\mathbf{R}) = \sum_{m} \lambda_m (\phi \otimes f)(\mathbf{R} \mathbf{z}_m)$$



Soler, Bagher, Nowrouzezahrai

Outline



Motivation

- Algebra of rotated zonal bases
- Applications

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

The rotated zonal harmonic basis



introduced by [Nowrouzezahrai et al.'2012]



Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

The rotated zonal harmonic basis

introduced by [Nowrouzezahrai et al.'2012]



Soler, Bagher, Nowrouzezahrai



The rotated zonal harmonic basis

introduced by [Nowrouzezahrai et al.'2012]



Soler, Bagher, Nowrouzezahrai



The rotated zonal harmonic basis

introduced by [Nowrouzezahrai et al.'2012]



Soler, Bagher, Nowrouzezahrai



The rotated zonal harmonic basis

introduced by [Nowrouzezahrai et al.'2012]

$$z_l^m = y_l^0 \circ \mathbf{R}$$
 such that

- needs 2l + 1 directions to span $\{y_l^m\}_{-l \le m \le l}$
- directions z_m need to be all distinct, but can be chosen at random!
- need to do the same for each degree I

Soler, Bagher, Nowrouzezahrai Eff

Efficient and Accurate Spherical Kernel Integrals



 $\mathbf{R}\mathbf{z}_m = \mathbf{z}$

1 - RZH Span the same space than SH



$$z_l^m(\omega) = (y_l^0 \circ R_m)(\omega) = \underbrace{\sqrt{\frac{4\pi}{2l+1}}}_{\pi_l} \sum_m y_l^m(\omega) y_l^m(R_m \mathbf{z})$$

⇒ linear dependency between RZH/SH coefficients:

 $\mathbf{C}_l = M_l \Lambda_l$

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

1 - RZH Span the same space than SH



$$z_l^m(\omega) = (y_l^0 \circ R_m)(\omega) = \underbrace{\sqrt{\frac{4\pi}{2l+1}}}_{\pi_l} \sum_m y_l^m(\omega) y_l^m(R_m \mathbf{z})$$

⇒ linear dependency between RZH/SH coefficients:

$$\mathbf{C}_l = M_l \Lambda_l$$

 \Rightarrow if M_l is invertible, $\{z_l^m\}$ and $\{y_l^m\}_{-l \le m \le l}$ span the same space

Soler, Bagher, Nowrouzezahrai

1 - RZH Span the same space than SH



$$z_l^m(\omega) = (y_l^0 \circ R_m)(\omega) = \underbrace{\sqrt{\frac{4\pi}{2l+1}}}_{\pi_l} \sum_m y_l^m(\omega) y_l^m(R_m \mathbf{z})$$

⇒ linear dependency between RZH/SH coefficients:

 $\mathbf{C}_l = M_l \Lambda_l$

 \mathbf{C}_l

 M_{l}

 Λ_i

 \Rightarrow if M_l is invertible, $\{z_l^m\}$ and $\{y_l^m\}_{-l \le m \le l}$ span the same space

Soler, Bagher, Nowrouzezahrai

1 - RZH Span the same space than SH



$$z_l^m(\omega) = (y_l^0 \circ R_m)(\omega) = \underbrace{\sqrt{\frac{4\pi}{2l+1}}}_{\pi_l} \sum_m y_l^m(\omega) y_l^m(R_m \mathbf{z})$$

⇒ linear dependency between RZH/SH coefficients:

$$\mathbf{C}_l = M_l \Lambda_l$$

 \Rightarrow if M_l is invertible, $\{z_l^m\}$ and $\{y_l^m\}_{-l \le m \le l}$ span the same space



 \Rightarrow no approximation (beyond SH bandlimit) + consistency

Soler, Bagher, Nowrouzezahrai

2 - Spherical kernel integrals at much lower cost







Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals



Efficient and Accurate Spherical Kernel Integrals



Efficient and Accurate Spherical Kernel Integrals



Efficient and Accurate Spherical Kernel Integrals





Efficient and Accurate Spherical Kernel Integrals

3 - Pre-convolutions with Z-harmonics





"Pisa" envmap

*y*⁰₁₅

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

3 - Pre-convolutions with Z-harmonics





Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

3 - Pre-convolutions with Z-harmonics





Soler, Bagher, Nowrouzezahrai

3 - Pre-convolutions with Z-harmonics





Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

3 - Pre-convolutions with Z-harmonics





Soler, Bagher, Nowrouzezahrai

3 - Pre-convolutions with Z-harmonics





Soler, Bagher, Nowrouzezahrai

3 - Pre-convolutions with Z-harmonics is free





Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

4 - RZH basis is not orthogonal



Addition theorem:

$$\langle z_l^m, z_l^{m'} \rangle = P_l(\mathbf{z}_m \cdot \mathbf{z}_{m'})$$

cannot optimize sparsity by tweaking directions beyond I = 1.

• use the dual basis $\langle z_l^m, \tilde{z}_l^{m'} \rangle = \delta(m, m')$ so $\lambda_l^m = \langle f, \tilde{z}_l^m \rangle$



...or simply convert SH coefficients into RZH by product with M_l

orthogonality is never used during computation anyway

Soler, Bagher, Nowrouzezahrai Efficient and Accurate Spherical Kernel Integrals

5 - Extension to arbitrary kernels



- fit very naturally in the framework
- bandlimit approximation introduced

Soler, Bagher, Nowrouzezahrai

Outline



- Motivation
- Algebra of rotated zonal bases
- Applications

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Real-time rendering measured materials



$$I(x,\omega_0) = \int_{\Omega} E(\omega) \rho(x,\mathbf{R}_n\omega,\mathbf{R}_n\omega_0) \max(\cos heta,0) d\omega$$

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Real-time rendering measured materials



 $I(x,\omega_0) = \sum_{l,m} \lambda_l^m (\mathbf{R}_n \omega_0) (E \otimes y_l^0) (\mathbf{R}_n^{-1} \mathbf{z}_m)$

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Real-time rendering measured materials



 $I(x,\omega_0) = \sum_{l,m} \lambda_l^m (\mathbf{R}_n \omega_0) \pi_l \sum_k e_l^k y_l^k (\mathbf{R}_n^{-1} \mathbf{z}_m)$

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals



$I(x,\omega_0) = \sum_{l,m} \lambda_l^m (\mathbf{R}_n \omega_0) \pi_l \sum_k e_l^k y_l^k (\mathbf{R}_n^{-1} \mathbf{z}_m)$

Shader implementation:

- ▶ for each direction *z_m*
 - loop over *l* and *k*, and compute $y_l^k(\mathbf{R}_n^{-1}\mathbf{z}_m)$ (recursive formula)

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals



$$I(x,\omega_0) = \sum_{l,m} \lambda_l^m (\mathbf{R}_n \omega_0) \pi_l \sum_k \boldsymbol{e}_l^k \boldsymbol{y}_l^k (\mathbf{R}_n^{-1} \mathbf{z}_m)$$

Shader implementation:

- ▶ for each direction *z_m*
 - loop over *l* and *k*, and compute $y_l^k(\mathbf{R}_n^{-1}\mathbf{z}_m)$ (recursive formula)
- no rotation needed

 \Rightarrow still real-time at $I_{max} = 30 - 40$

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals



$$I(x,\omega_0) = \sum_{l,m} \lambda_l^m (\mathbf{R}_n \omega_0) \pi_l \sum_k e_l^k y_l^k (\mathbf{R}_n^{-1} \mathbf{z}_m)$$

Shader implementation:

- for each direction z_m
 - loop over *l* and *k*, and compute $y_l^k(\mathbf{R}_n^{-1}\mathbf{z}_m)$ (recursive formula)
- no rotation needed
 - \Rightarrow still real-time at $I_{max} = 30 40$

l_{max} = 7 80 fps

Yellow satin

Brushed Alum

Color changing paint 3

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals



$$I(x,\omega_0) = \sum_{l,m} \lambda_l^m (\mathbf{R}_n \omega_0) \pi_l \sum_k e_l^k y_l^k (\mathbf{R}_n^{-1} \mathbf{z}_m)$$

Shader implementation:

- for each direction z_m
 - loop over *I* and *k*, and compute $y_I^k(\mathbf{R}_n^{-1}\mathbf{z}_m)$ (recursive formula)
- no rotation needed
 - \Rightarrow still real-time at $I_{max} = 30 40$



Yellow satin

Brushed Alum

Color changing paint 3

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals



$$I(x,\omega_0) = \sum_{l,m} \lambda_l^m (\mathbf{R}_n \omega_0) \pi_l \sum_k e_l^k y_l^k (\mathbf{R}_n^{-1} \mathbf{z}_m)$$

Shader implementation:

- for each direction z_m
 - loop over *I* and *k*, and compute $y_{I}^{k}(\mathbf{R}_{n}^{-1}\mathbf{z}_{m})$ (recursive formula)
- no rotation needed
 - \Rightarrow still real-time at $I_{max} = 30 40$



 $I_{max} = 31$ 12 fps

Yellow satin

Brushed Alum

Color changing paint 3

Soler, Bagher, Nowrouzezahrai



$$I(x,\omega_0) = \sum_{l,m} \lambda_l^m (\mathbf{R}_n \omega_0) \pi_l \sum_k e_l^k y_l^k (\mathbf{R}_n^{-1} \mathbf{z}_m)$$

Shader implementation:

- for each direction z_m
 - loop over *l* and *k*, and compute $y_l^k(\mathbf{R}_n^{-1}\mathbf{z}_m)$ (recursive formula)
- no rotation needed
 - \Rightarrow still real-time at $I_{max} = 30 40$



Ref.

Yellow satin

Brushed Alum

Color changing paint 3

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Filtering anisotropic BRDF datasets



anisotropic BRDFs (e.g. MERL) are extremely noisy and sparse
 ⇒ cannot be used in a renderer unless filtered



Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Filtering anisotropic BRDF datasets



- anisotropic BRDFs (e.g. MERL) are extremely noisy and sparse
 ⇒ cannot be used in a renderer unless filtered
- we convolve them in 4D with a "bi-spherical" Gaussian \Rightarrow produces best L_2 bandlimited approximation



Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Filtering anisotropic BRDF datasets



- anisotropic BRDFs (e.g. MERL) are extremely noisy and sparse
 ⇒ cannot be used in a renderer unless filtered
- we convolve them in 4D with a "bi-spherical" Gaussian \Rightarrow produces best L_2 bandlimited approximation



Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Filtered mirror reflections





Distant lighting

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Nov 05, 2015

Filtered mirror reflections





Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Filtered mirror reflections





Isotropic spherical decomposition

Shader implementation:

Soler, Bagher, Nowrouzezahrai

- compute screen-space curvature buffer
- compute anisotropic Gaussian parameters
- use pre-computed isotropic decomposition of the Gaussian



Filtered mirror reflections





No Antialiasing
0.1msGPU-FXAA
0.1msOurs ($I_{max} = 15$)
2msRef 16 × 16
20ms

Shader implementation:

- compute screen-space curvature buffer
- compute anisotropic Gaussian parameters
- use pre-computed isotropic decomposition of the Gaussian

Efficient and Accurate Spherical Kernel Integrals

Filtered mirror reflections







No Antialiasing 0.1*ms*

GPU-FXAA 0.1*ms*



Ours $(I_{max} = 15)$ Ref 16×16 2ms 20ms

Shader implementation:

Soler, Bagher, Nowrouzezahrai

- compute screen-space curvature buffer
- compute anisotropic Gaussian parameters
- use pre-computed isotropic decomposition of the Gaussian

Numerical stability



Direction sharing can be a problem

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Numerical stability



- Direction sharing can be a problem
- Gibbs effect?

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Numerical stability



- Direction sharing can be a problem
- Gibbs effect?
- stable SH formula allows up to I_{max} = 80 (floats), I_{max} = 300 (doubles)

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals

Questions?



Resources:

source code (ylm+zlm lib)

- # apt-add-repository ppa:csoler-users/ylm
- # apt-get update
- # apt-get install libylm-dev libylm-tools

shaders and videos

http://artis.imag.fr/~Cyril.Soler/RZH/

Soler, Bagher, Nowrouzezahrai

Efficient and Accurate Spherical Kernel Integrals