SIMON FRASER UNIVERSITY
THINKING OF THE WORLD

Toward High-Quality Gradient Estimation on Regular Lattices

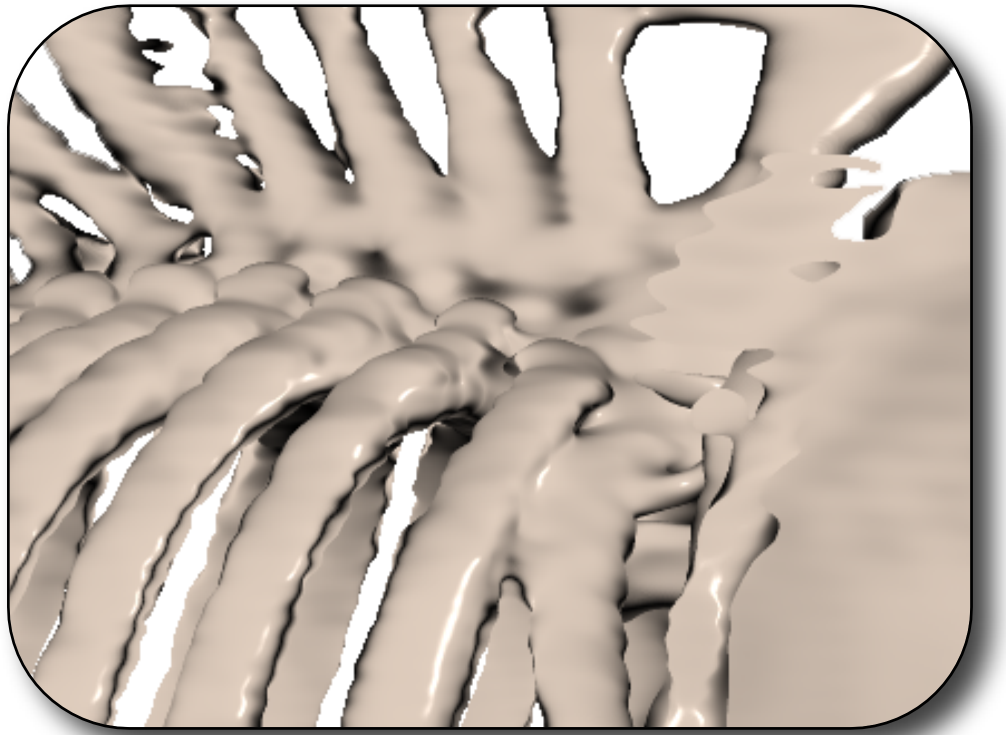
Zahid Hossain[†], Usman R. Alim[*], and Torsten Möller[*]

[†]Stanford University [*]Graphics, Usability, and Visualization (GrUVi) Lab.
Simon Fraser University

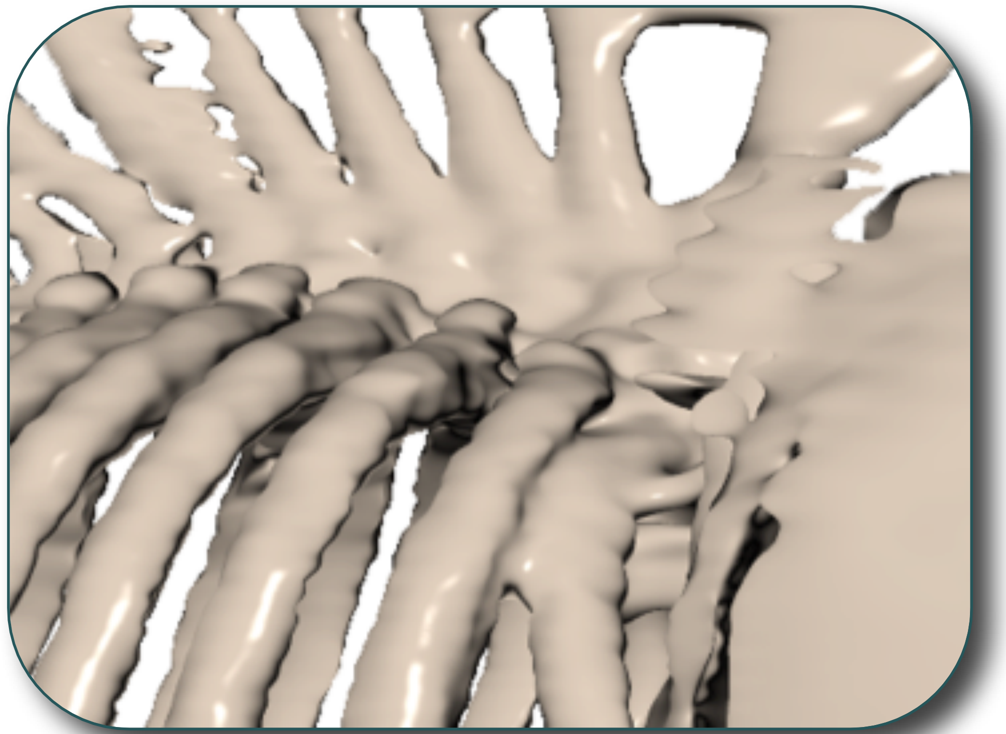
gruvi graphics + usability + visualization

Motivation

Primary: Lighting in volume rendering



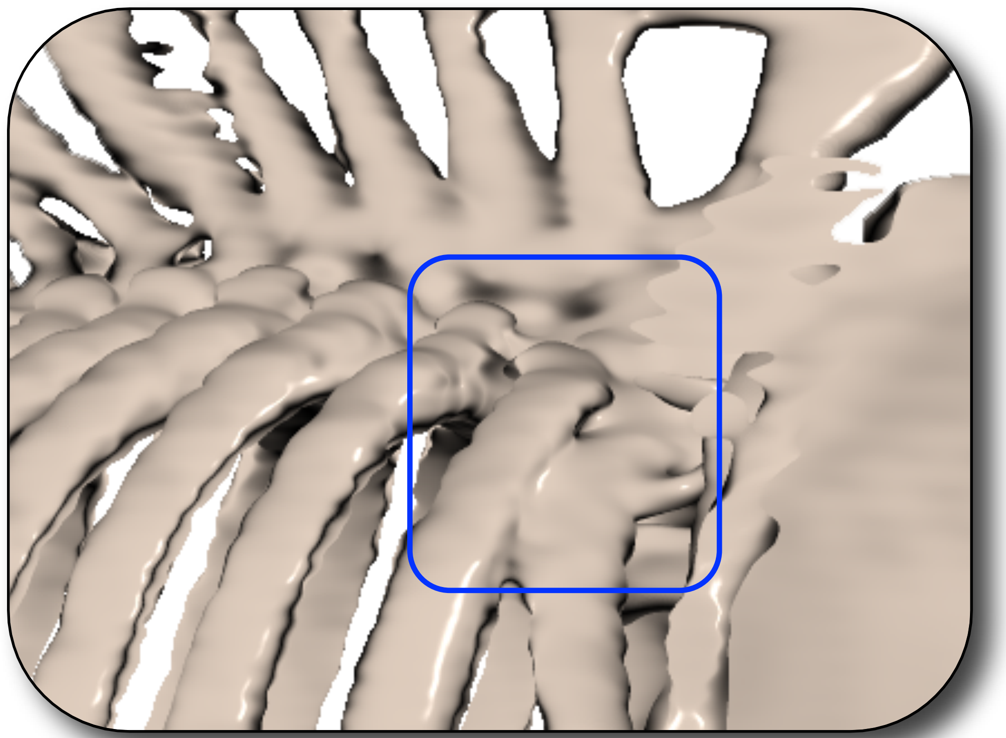
finite differencing



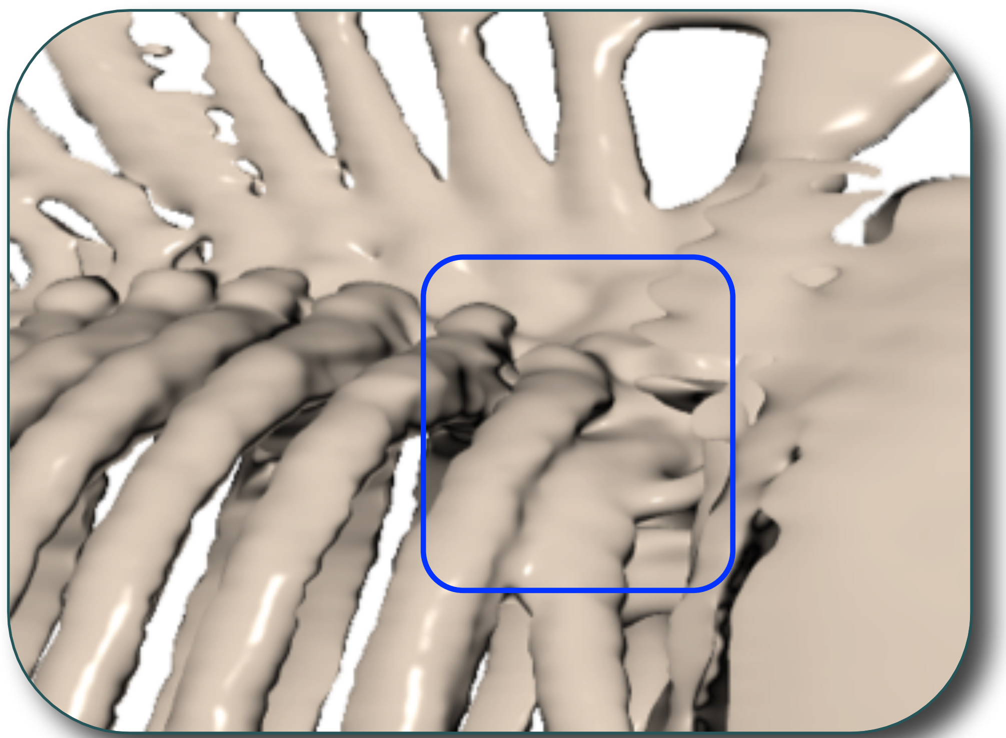
orthogonal projection

Motivation

Primary: Lighting in volume rendering



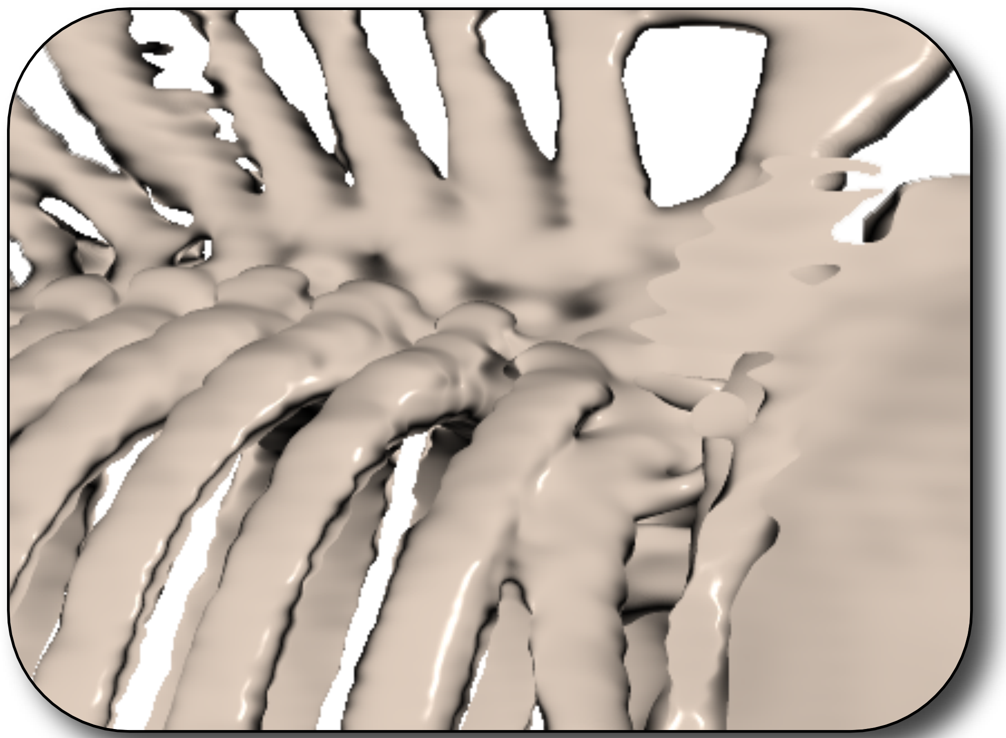
finite differencing



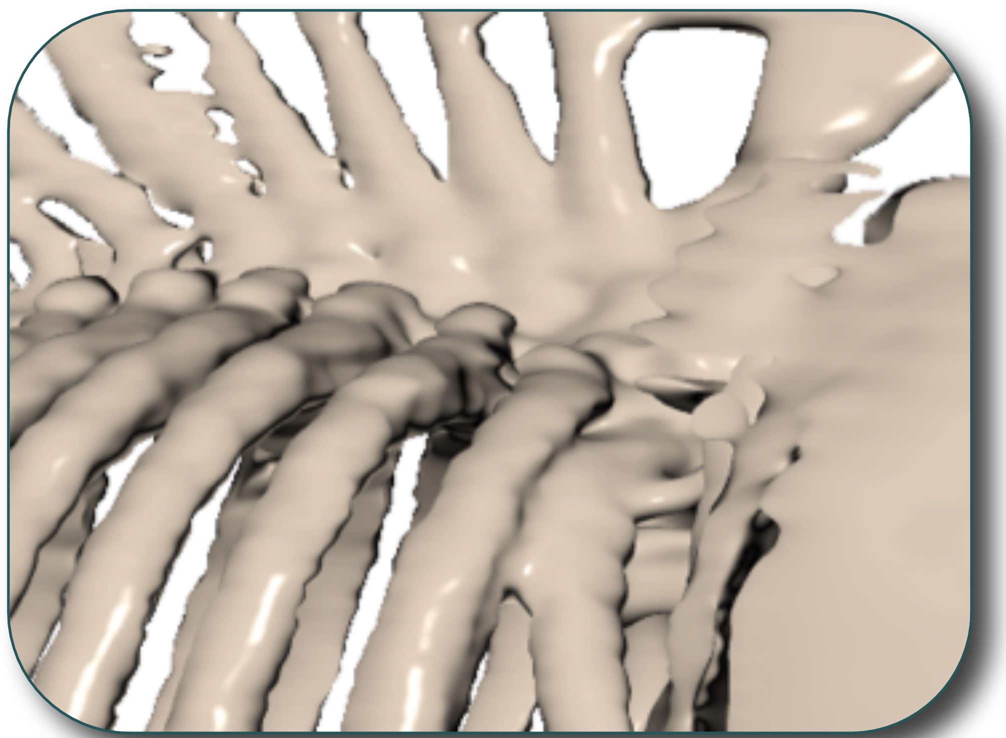
orthogonal projection

Motivation

Primary: Lighting in volume rendering



finite differencing



orthogonal projection

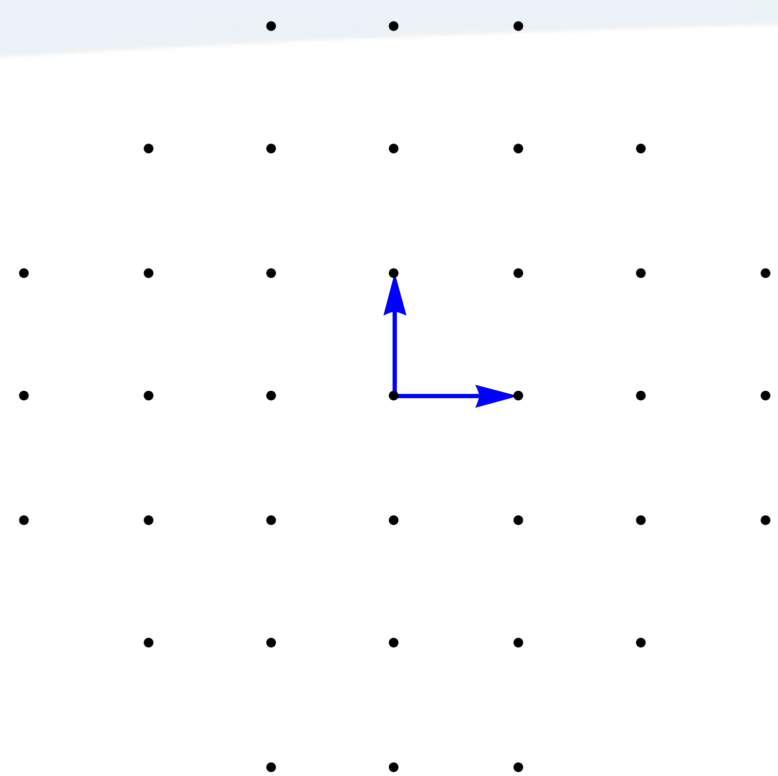
Secondary: Numerical methods for PDEs

Outline

1. Motivation ✔
2. Two Novel Gradient Estimation Frameworks
    a. Taylor Series Framework
    b. Approximation Spaces
3. Comparison + Results
4. Conclusion

Taylor Series Framework

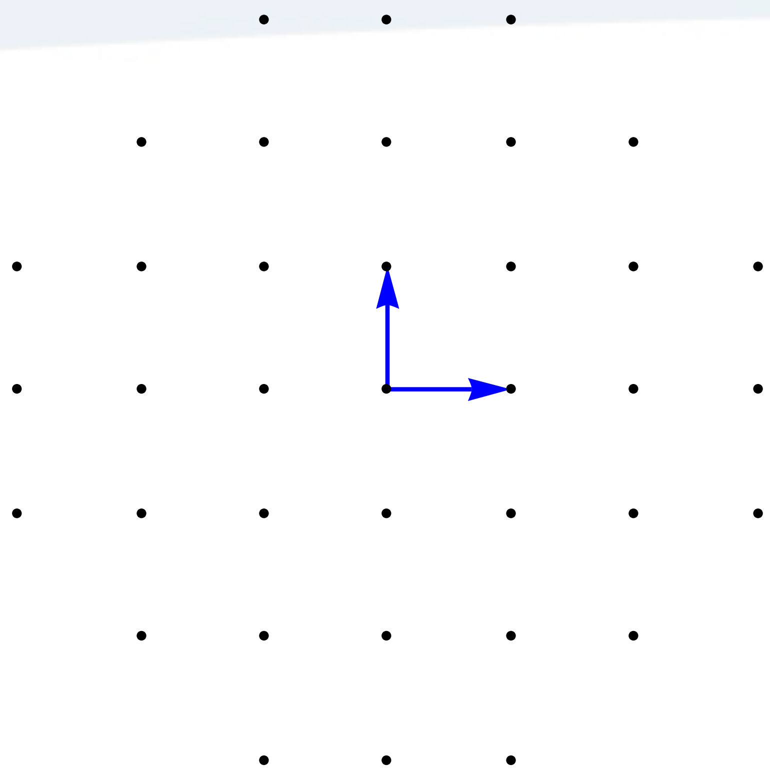
Background



Cartesian lattice
- Axis aligned finite differences
- Higher-order filters [Möller *et al.* 1997]

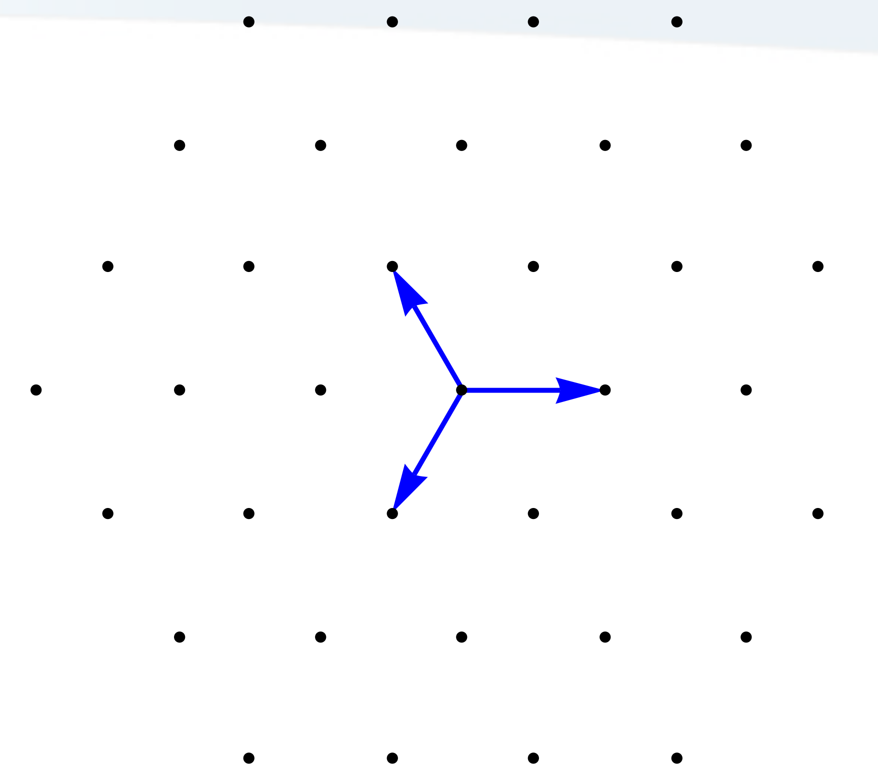
Background

Finite difference method for arbitrary lattices?



Cartesian lattice

- Axis aligned finite differences
- Higher-order filters [Möller *et al.* 1997]



Arbitrary Lattices

- Non-separable filters
- Need a multidimensional analysis
- Extension of [Möller *et al.* 1997]

Taylor Expansion

1. Convolution of lattice samples with a discrete filter

$$f^{\Delta}[\boldsymbol{k}] := (f * \Delta)[\boldsymbol{k}] = \sum_{\boldsymbol{m} \in \mathbb{Z}^s} f(h\boldsymbol{L}\boldsymbol{m})\Delta[\boldsymbol{m} - \boldsymbol{k}]$$

2. Substitute the multi-dimensional Taylor expansion...

$$f(h\boldsymbol{L}\boldsymbol{m}) = \sum_{\boldsymbol{n} \in \mathbb{N}^s} \frac{(h\boldsymbol{L}\boldsymbol{m} - \boldsymbol{x})^{\boldsymbol{n}}}{\boldsymbol{n}!} D^{\boldsymbol{n}} f(\boldsymbol{x}) \quad \text{where} \quad D^{\boldsymbol{n}} := \frac{\partial^{|\boldsymbol{n}|}}{\partial x_1^{n_1} \ldots \partial x_s^{n_s}}$$

Taylor Expansion

1. Convolution of lattice samples with a discrete filter

$$f^{\Delta}[\boldsymbol{k}] := (f * \Delta)[\boldsymbol{k}] = \sum_{\boldsymbol{m} \in \mathbb{Z}^s} f(h\boldsymbol{L}\boldsymbol{m})\Delta[\boldsymbol{m} - \boldsymbol{k}]$$

samples

2. Substitute the multi-dimensional Taylor expansion...

$$f(h\boldsymbol{L}\boldsymbol{m}) = \sum_{\boldsymbol{n} \in \mathbb{N}^s} \frac{(h\boldsymbol{L}\boldsymbol{m} - \boldsymbol{x})^{\boldsymbol{n}}}{\boldsymbol{n}!} D^{\boldsymbol{n}} f(\boldsymbol{x}) \quad \text{where} \quad D^{\boldsymbol{n}} := \frac{\partial^{|\boldsymbol{n}|}}{\partial x_1^{n_1} \ldots \partial x_s^{n_s}}$$

Taylor Expansion

1. Convolution of lattice samples with a discrete filter

$$f^{\Delta}[\boldsymbol{k}] := (f * \Delta)[\boldsymbol{k}] = \sum_{\boldsymbol{m} \in \mathbb{Z}^s} f(h\boldsymbol{L}\boldsymbol{m})\Delta[\boldsymbol{m} - \boldsymbol{k}]$$

derivative filter

2. Substitute the multi-dimensional Taylor expansion...

$$f(h\boldsymbol{L}\boldsymbol{m}) = \sum_{\boldsymbol{n} \in \mathbb{N}^s} \frac{(h\boldsymbol{L}\boldsymbol{m} - \boldsymbol{x})^{\boldsymbol{n}}}{\boldsymbol{n}!} D^{\boldsymbol{n}} f(\boldsymbol{x}) \quad \text{where} \quad D^{\boldsymbol{n}} := \frac{\partial^{|\boldsymbol{n}|}}{\partial x_1^{n_1} \ldots \partial x_s^{n_s}}$$

Taylor Expansion

1. Convolution of lattice samples with a discrete filter

$$f^{\Delta}[\boldsymbol{k}] := (f * \Delta)[\boldsymbol{k}] = \sum_{\boldsymbol{m} \in \mathbb{Z}^s} f(h\boldsymbol{L}\boldsymbol{m})\Delta[\boldsymbol{m} - \boldsymbol{k}]$$

scaling parameter

2. Substitute the multi-dimensional Taylor expansion...

$$f(h\boldsymbol{L}\boldsymbol{m}) = \sum_{\boldsymbol{n} \in \mathbb{N}^s} \frac{(h\boldsymbol{L}\boldsymbol{m} - \boldsymbol{x})^{\boldsymbol{n}}}{\boldsymbol{n}!} D^{\boldsymbol{n}} f(\boldsymbol{x}) \quad \text{where} \quad D^{\boldsymbol{n}} := \frac{\partial^{|\boldsymbol{n}|}}{\partial x_1^{n_1} \ldots \partial x_s^{n_s}}$$

Taylor Expansion

1. Convolution of lattice samples with a discrete filter

$$f^{\Delta}[\boldsymbol{k}] := (f * \Delta)[\boldsymbol{k}] = \sum_{\boldsymbol{m} \in \mathbb{Z}^s} f(h\boldsymbol{L}\boldsymbol{m})\Delta[\boldsymbol{m} - \boldsymbol{k}]$$

lattice matrix

2. Substitute the multi-dimensional Taylor expansion...

$$f(h\boldsymbol{L}\boldsymbol{m}) = \sum_{\boldsymbol{n} \in \mathbb{N}^s} \frac{(h\boldsymbol{L}\boldsymbol{m} - \boldsymbol{x})^{\boldsymbol{n}}}{\boldsymbol{n}!} D^{\boldsymbol{n}} f(\boldsymbol{x}) \quad \text{where} \quad D^{\boldsymbol{n}} := \frac{\partial^{|\boldsymbol{n}|}}{\partial x_1^{n_1} \ldots \partial x_s^{n_s}}$$

Taylor Expansion

1. Convolution of lattice samples with a discrete filter

$$f^{\Delta}[\boldsymbol{k}] := (f * \Delta)[\boldsymbol{k}] = \sum_{\boldsymbol{m} \in \mathbb{Z}^s} f(h\boldsymbol{L}\boldsymbol{m})\Delta[\boldsymbol{m} - \boldsymbol{k}]$$

2. Substitute the multi-dimensional Taylor expansion...

$$f(h\boldsymbol{L}\boldsymbol{m}) = \sum_{\boldsymbol{n} \in \mathbb{N}^s} \frac{(h\boldsymbol{L}\boldsymbol{m} - \boldsymbol{x})^{\boldsymbol{n}}}{\boldsymbol{n}!} D^{\boldsymbol{n}} f(\boldsymbol{x}) \quad \text{where} \quad D^{\boldsymbol{n}} := \frac{\partial^{|\boldsymbol{n}|}}{\partial x_1^{n_1} \ldots \partial x_s^{n_s}}$$

Taylor Expansion

1. Convolution of lattice samples with a discrete filter

$$f^{\Delta}[\boldsymbol{k}] := (f * \Delta)[\boldsymbol{k}] = \sum_{\boldsymbol{m} \in \mathbb{Z}^s} f(h\boldsymbol{L}\boldsymbol{m})\Delta[\boldsymbol{m} - \boldsymbol{k}]$$

2. Substitute the multi-dimensional Taylor expansion...

$$f(h\boldsymbol{L}\boldsymbol{m}) = \sum_{\boldsymbol{n} \in \mathbb{N}^s} \frac{(h\boldsymbol{L}\boldsymbol{m} - \boldsymbol{x})^{\boldsymbol{n}}}{\boldsymbol{n}!} D^{\boldsymbol{n}} f(\boldsymbol{x}) \quad \text{where} \quad D^{\boldsymbol{n}} := \frac{\partial^{|\boldsymbol{n}|}}{\partial x_1^{n_1} \ldots \partial x_s^{n_s}}$$

component-wise

Taylor Expansion

1. Convolution of lattice samples with a discrete filter

$$f^{\Delta}[\boldsymbol{k}] := (f * \Delta)[\boldsymbol{k}] = \sum_{\boldsymbol{m}\in\mathbb{Z}^s} f(h\boldsymbol{L}\boldsymbol{m})\Delta[\boldsymbol{m}-\boldsymbol{k}]$$

2. Substitute the multi-dimensional Taylor expansion...

$$f(h\boldsymbol{L}\boldsymbol{m}) = \sum_{\boldsymbol{n}\in\mathbb{N}^s} \frac{(h\boldsymbol{L}\boldsymbol{m}-\boldsymbol{x})^{\boldsymbol{n}}}{\boldsymbol{n}!} D^{\boldsymbol{n}} f(\boldsymbol{x}) \quad \text{where} \quad D^{\boldsymbol{n}} := \frac{\partial^{|\boldsymbol{n}|}}{\partial x_1^{n_1}\dots\partial x_s^{n_s}}$$

Taylor Expansion

...and we obtain

$$f^{\Delta}[\boldsymbol{k}] = \sum_{\boldsymbol{n} \in \mathbb{N}^s} D^{\boldsymbol{n}} f(h\boldsymbol{L}\boldsymbol{k}) \cdot a^{\Delta}_{\boldsymbol{n}}$$

Taylor Expansion

...and we obtain

$$f^{\Delta}[\boldsymbol{k}] = \sum_{\boldsymbol{n}\in\mathbb{N}^s} D^{\boldsymbol{n}} f(h\boldsymbol{L}\boldsymbol{k}) \cdot a_{\boldsymbol{n}}^{\Delta}$$

where the Taylor coefficient is given by the linear system

$$a_{\boldsymbol{n}}^{\Delta} := \frac{h^{|\boldsymbol{n}|}}{\boldsymbol{n}!} \sum_{\boldsymbol{m}\in\mathbb{Z}^s} (\boldsymbol{L}\boldsymbol{m})^{\boldsymbol{n}} \cdot \Delta[-\boldsymbol{m}]$$

Taylor Expansion

...and we obtain

$$f^{\Delta}[\boldsymbol{k}] = \sum_{\boldsymbol{n}\in\mathbb{N}^s} D^{\boldsymbol{n}} f(h\boldsymbol{L}\boldsymbol{k}) \cdot a^{\Delta}_{\boldsymbol{n}}$$

$$\begin{aligned} f[\boldsymbol{k}] = {} & a_{0,0} f(h\boldsymbol{L}\boldsymbol{k}) + \\ & a_{1,0}\frac{\partial f}{\partial x}(\cdot) + a_{0,1}\frac{\partial f}{\partial y}(\cdot) + \\ & a_{1,1}\frac{\partial^2 f}{\partial x \partial y}(\cdot) + a_{2,0}\frac{\partial^2 f}{\partial x^2}(\cdot) + a_{0,2}\frac{\partial^2 f}{\partial y^2}(\cdot) + \\ & \dots \end{aligned}$$

where the Taylor coefficient is given by the linear system

$$a^{\Delta}_{\boldsymbol{n}} := \frac{h^{|\boldsymbol{n}|}}{\boldsymbol{n}!} \sum_{\boldsymbol{m}\in\mathbb{Z}^s} (\boldsymbol{L}\boldsymbol{m})^{\boldsymbol{n}} \cdot \Delta[-\boldsymbol{m}]$$

Taylor Expansion

...and we obtain

$$f^{\Delta}[\boldsymbol{k}] = \sum_{\boldsymbol{n} \in \mathbb{N}^s} D^{\boldsymbol{n}} f(h\boldsymbol{L}\boldsymbol{k}) \cdot a^{\Delta}_{\boldsymbol{n}}$$

$$f[\boldsymbol{k}] = a_{0,0} f(h\boldsymbol{L}\boldsymbol{k}) +$$ order 0

$$a_{1,0} \frac{\partial f}{\partial x}(\cdot) + a_{0,1} \frac{\partial f}{\partial y}(\cdot) +$$ order 1

$$a_{1,1} \frac{\partial^2 f}{\partial x \partial y}(\cdot) + a_{2,0} \frac{\partial^2 f}{\partial x^2}(\cdot) + a_{0,2} \frac{\partial^2 f}{\partial y^2}(\cdot) +$$ order 2

$$\ldots$$

where the Taylor coefficient is given by the linear system

$$a^{\Delta}_{\boldsymbol{n}} := \frac{h^{|\boldsymbol{n}|}}{\boldsymbol{n}!} \sum_{\boldsymbol{m} \in \mathbb{Z}^s} (\boldsymbol{L}\boldsymbol{m})^{\boldsymbol{n}} \cdot \Delta[-\boldsymbol{m}]$$

Taylor Expansion

...and we obtain

$$f^{\Delta}[\boldsymbol{k}] = \sum_{\boldsymbol{n} \in \mathbb{N}^s} D^{\boldsymbol{n}} f(h\boldsymbol{L}\boldsymbol{k}) \cdot a^{\Delta}_{\boldsymbol{n}}$$

$$f[\boldsymbol{k}] = a_{0,0} f(h\boldsymbol{L}\boldsymbol{k}) +$$ order 0

$$1 = a_{1,0}\frac{\partial f}{\partial x}(\cdot) + a_{0,1}\frac{\partial f}{\partial y}(\cdot) +$$ order 1

$$a_{1,1}\frac{\partial^2 f}{\partial x \partial y}(\cdot) + a_{2,0}\frac{\partial^2 f}{\partial x^2}(\cdot) + a_{0,2}\frac{\partial^2 f}{\partial y^2}(\cdot) +$$ order 2

$$\ldots$$

where the Taylor coefficient is given by the linear system

$$a^{\Delta}_{\boldsymbol{n}} := \frac{h^{|\boldsymbol{n}|}}{\boldsymbol{n}!} \sum_{\boldsymbol{m} \in \mathbb{Z}^s} (\boldsymbol{L}\boldsymbol{m})^{\boldsymbol{n}} \cdot \Delta[-\boldsymbol{m}]$$

Taylor Expansion

...and we obtain

$$f^{\Delta}[\boldsymbol{k}] = \sum_{\boldsymbol{n}\in\mathbb{N}^s} D^{\boldsymbol{n}} f(h\boldsymbol{L}\boldsymbol{k}) \cdot a^{\Delta}_{\boldsymbol{n}}$$

$$f[\boldsymbol{k}] = a_{0,0} f(h\boldsymbol{L}\boldsymbol{k}) + \quad \text{order 0}$$

$$1 = a_{1,0}\frac{\partial f}{\partial x}(\cdot) + a_{0,1}\frac{\partial f}{\partial y}(\cdot) + \quad \text{order 1}$$

$$a_{1,1}\frac{\partial^2 f}{\partial x \partial y}(\cdot) + a_{2,0}\frac{\partial^2 f}{\partial x^2}(\cdot) + a_{0,2}\frac{\partial^2 f}{\partial y^2}(\cdot) + \quad \text{order 2}$$

$$\ldots$$

where the Taylor coefficient is given by the linear system

$$a^{\Delta}_{\boldsymbol{n}} := \frac{h^{|\boldsymbol{n}|}}{\boldsymbol{n}!} \sum_{\boldsymbol{m}\in\mathbb{Z}^s} (\boldsymbol{L}\boldsymbol{m})^{\boldsymbol{n}} \cdot \Delta[-\boldsymbol{m}]$$

known coefficients

unknown filter weights

Implementation

- Linear system is often not full rank
- Find a suitable solution by:
  a. Imposing symmetry/anti-symmetry in the filter geometry
  b. Minimizing error due to higher order terms
- Optimal support for a given order?

Approximation Spaces

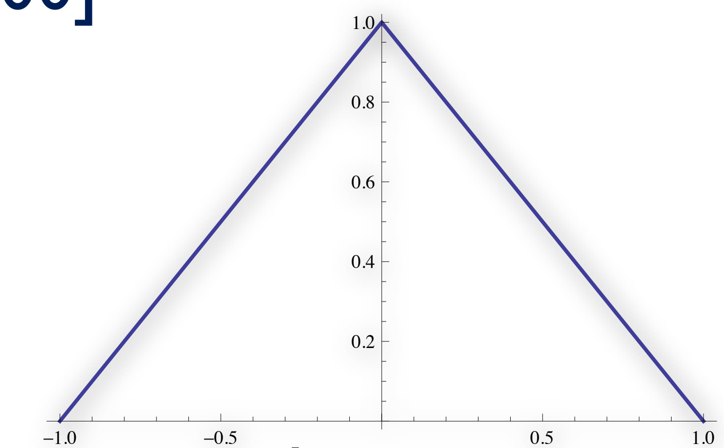
Background

Approximation space generated by shifts of a kernel

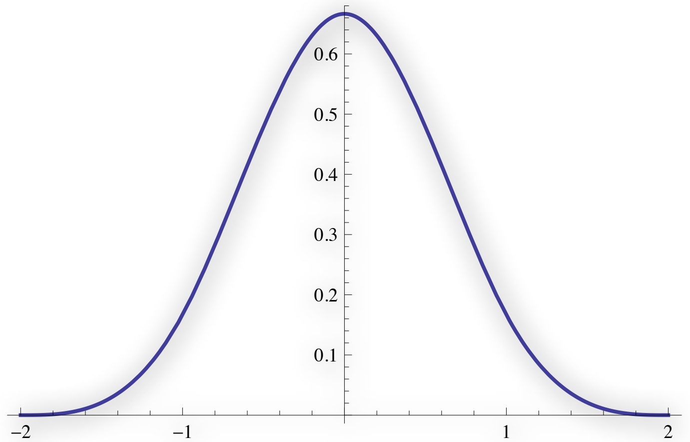
$$V_{\mathcal{L}_h}(\varphi) := \left\{ s(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^s} c[\boldsymbol{k}]\varphi(\frac{\boldsymbol{x}}{h} - \boldsymbol{L}\boldsymbol{k}) \;:\; c[\boldsymbol{k}] \in l_2(\mathbb{Z}^s) \right\}$$

Function reconstruction from discrete measurements

- Sampling, interpolation, approximation [Unser 00]
- Quantitative analysis [Blu *et al.* 99]



linear



cubic

Background

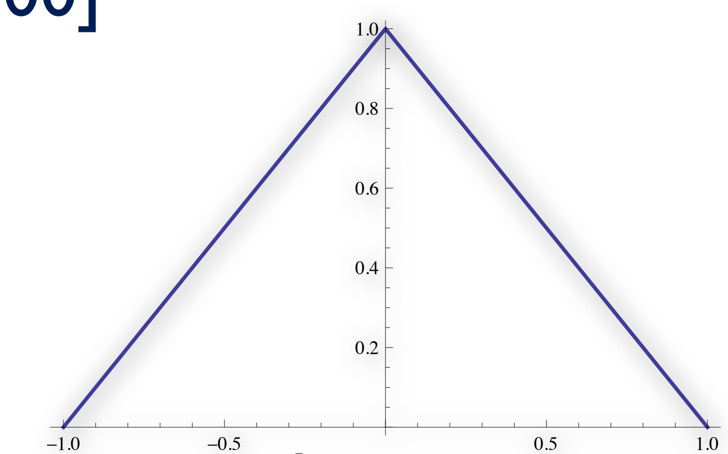
Gradient approximation in a shift-invariant space?

Approximation space generated by shifts of a kernel

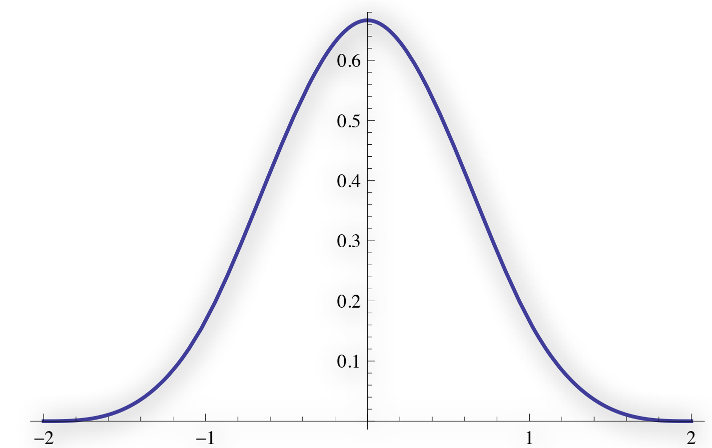
$$V_{\mathcal{L}_h}(\varphi) := \left\{ s(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^s} c[\boldsymbol{k}]\varphi(\frac{\boldsymbol{x}}{h} - \boldsymbol{L}\boldsymbol{k}) \ : \ c[\boldsymbol{k}] \in l_2(\mathbb{Z}^s) \right\}$$

Function reconstruction from discrete measurements

- Sampling, interpolation, approximation [Unser 00]
- Quantitative analysis [Blu *et al.* 99]

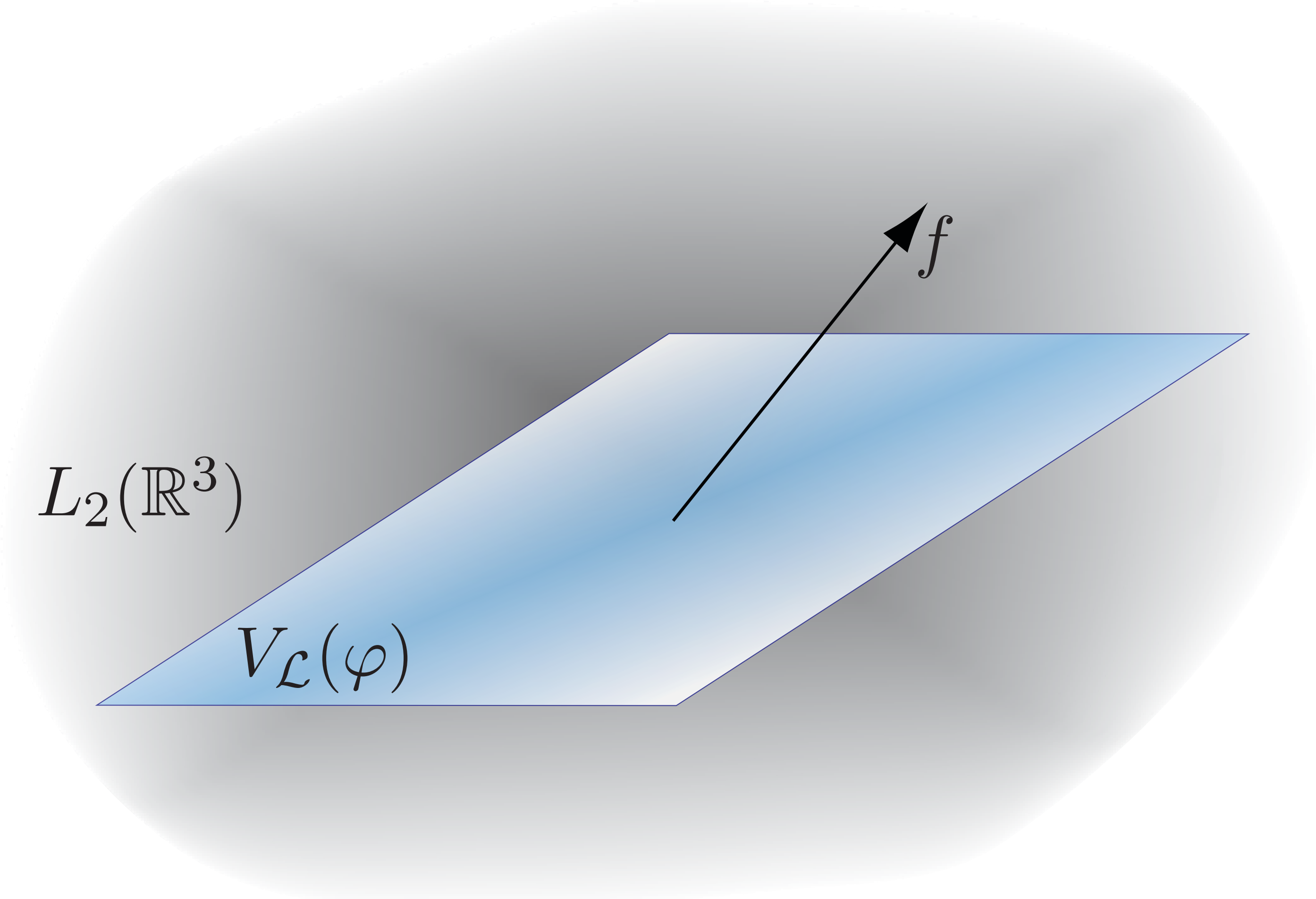


linear

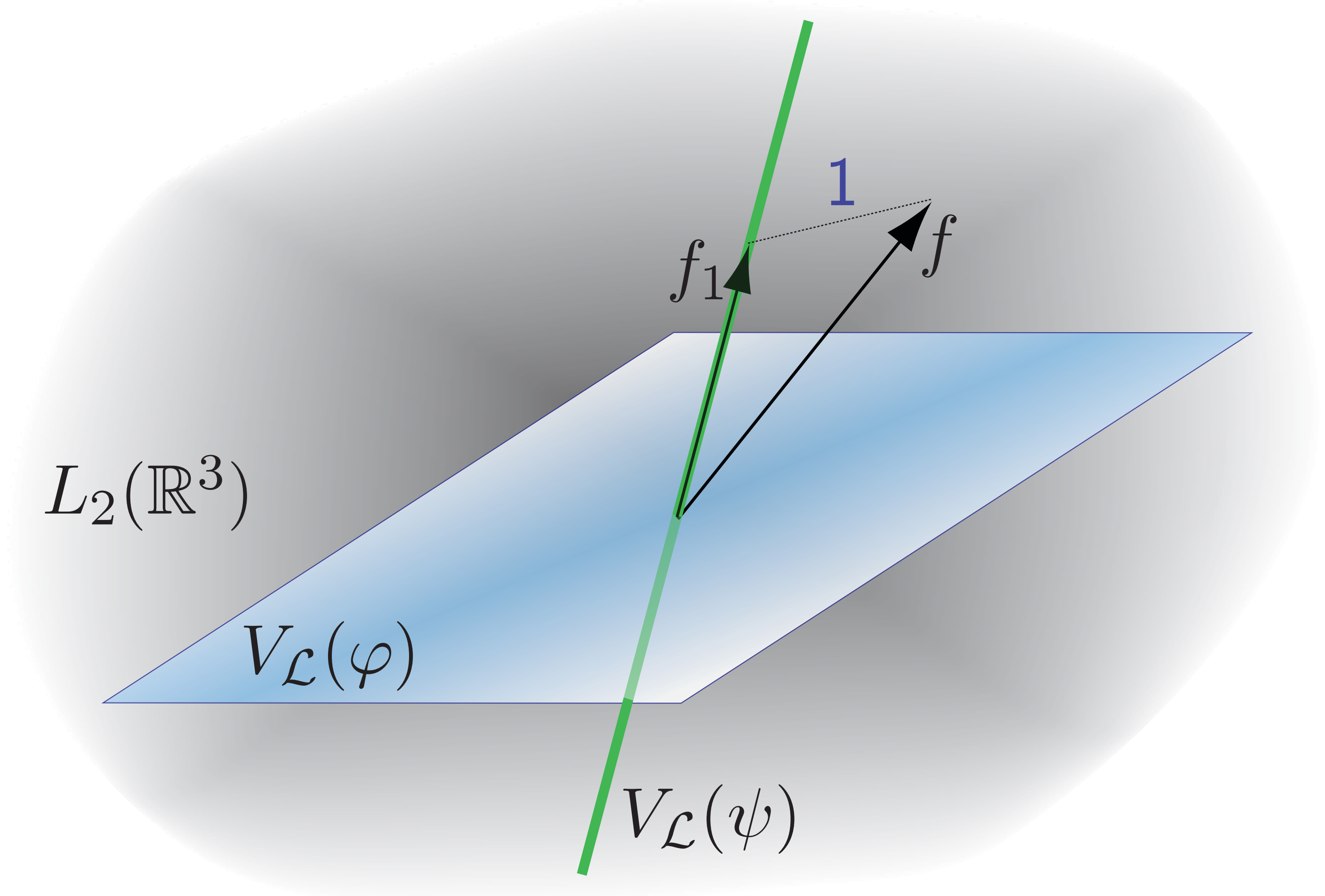


cubic

Two Stage Gradient Approximation



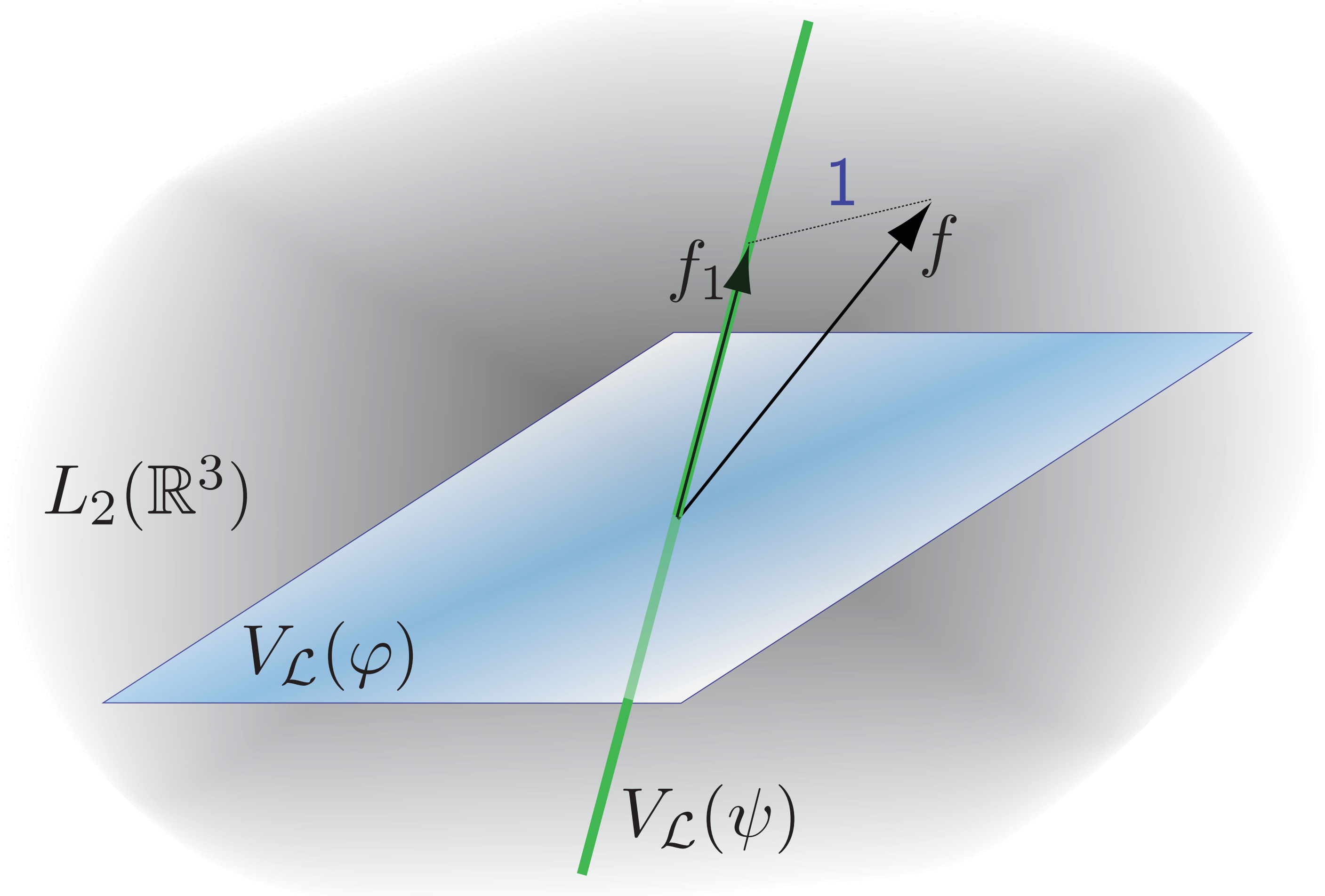
Two Stage Gradient Approximation



1. Approximate the function in an intermediate space

$$f_1(\boldsymbol{x}) = \sum_{\boldsymbol{k}} (f * p_1)[\boldsymbol{k}]\psi_{\boldsymbol{k}}(\boldsymbol{x})$$

Two Stage Gradient Approximation

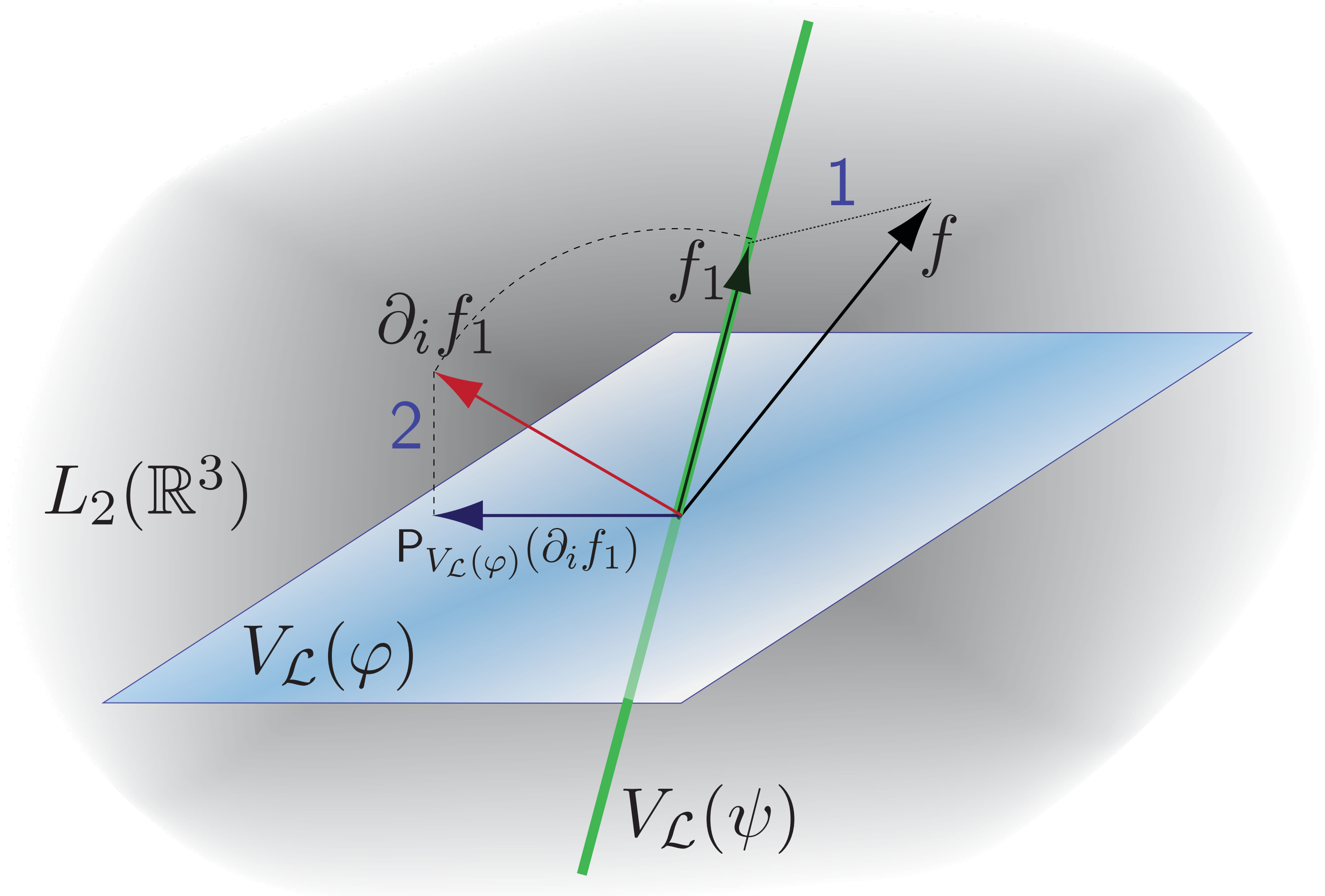


1. Approximate the function in an intermediate space

$$f_1(\boldsymbol{x}) = \sum_{\boldsymbol{k}} (f * p_1)[\boldsymbol{k}]\psi_{\boldsymbol{k}}(\boldsymbol{x})$$

Prefilter imposes interpolation constraints

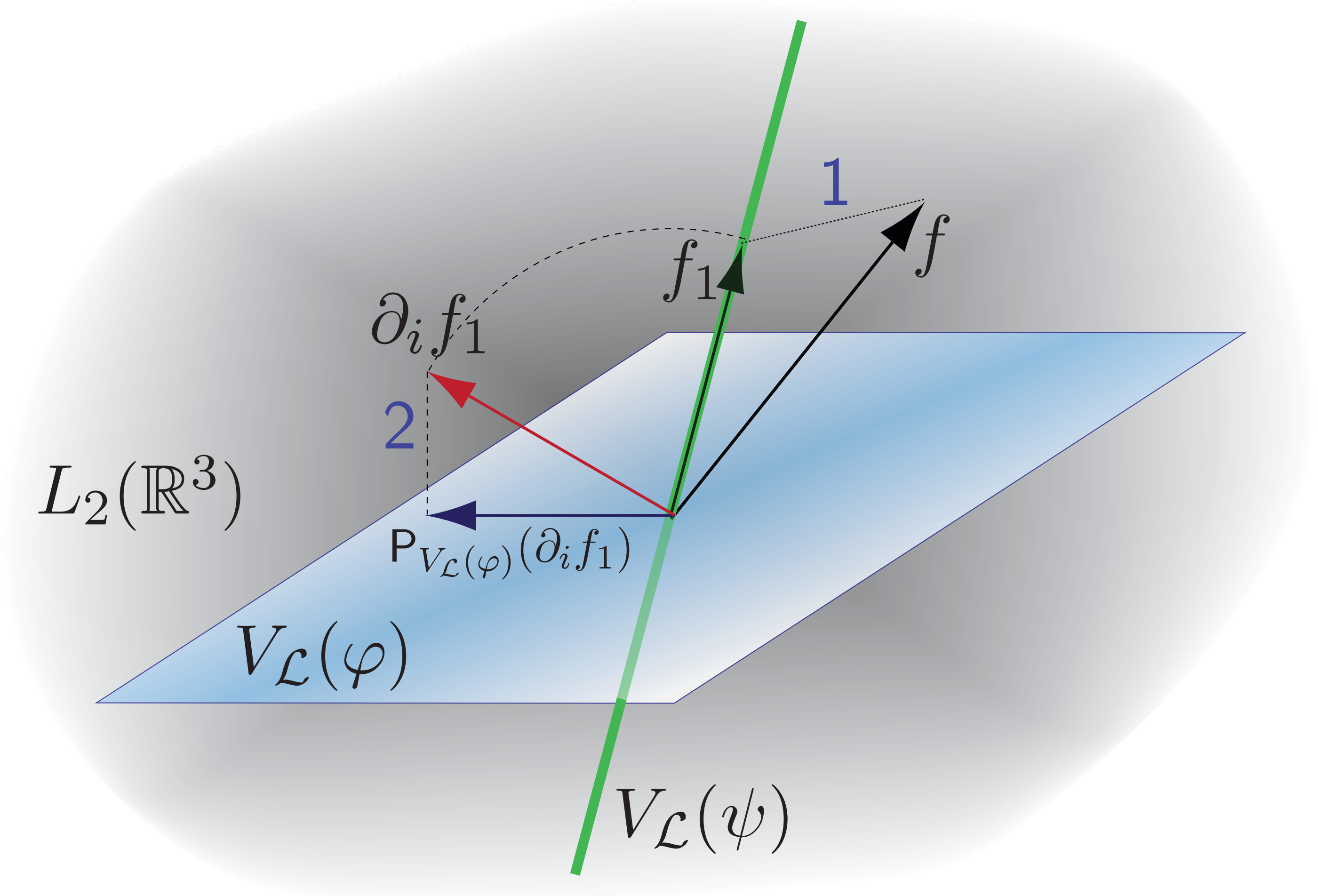
Two Stage Gradient Approximation



2. Orthogonally project the analytical derivative to the target space

$$f_{2,i}(\boldsymbol{x}) := \left(\mathsf{P}_{V_{\mathcal{L}(\varphi)}} \partial_i f_1\right)(\boldsymbol{x})$$
$$= \sum_{\boldsymbol{k}} ((f * p_1) * \mathring{d}_i)[\boldsymbol{k}]\, \varphi_{\boldsymbol{k}}(\boldsymbol{x})$$

Two Stage Gradient Approximation



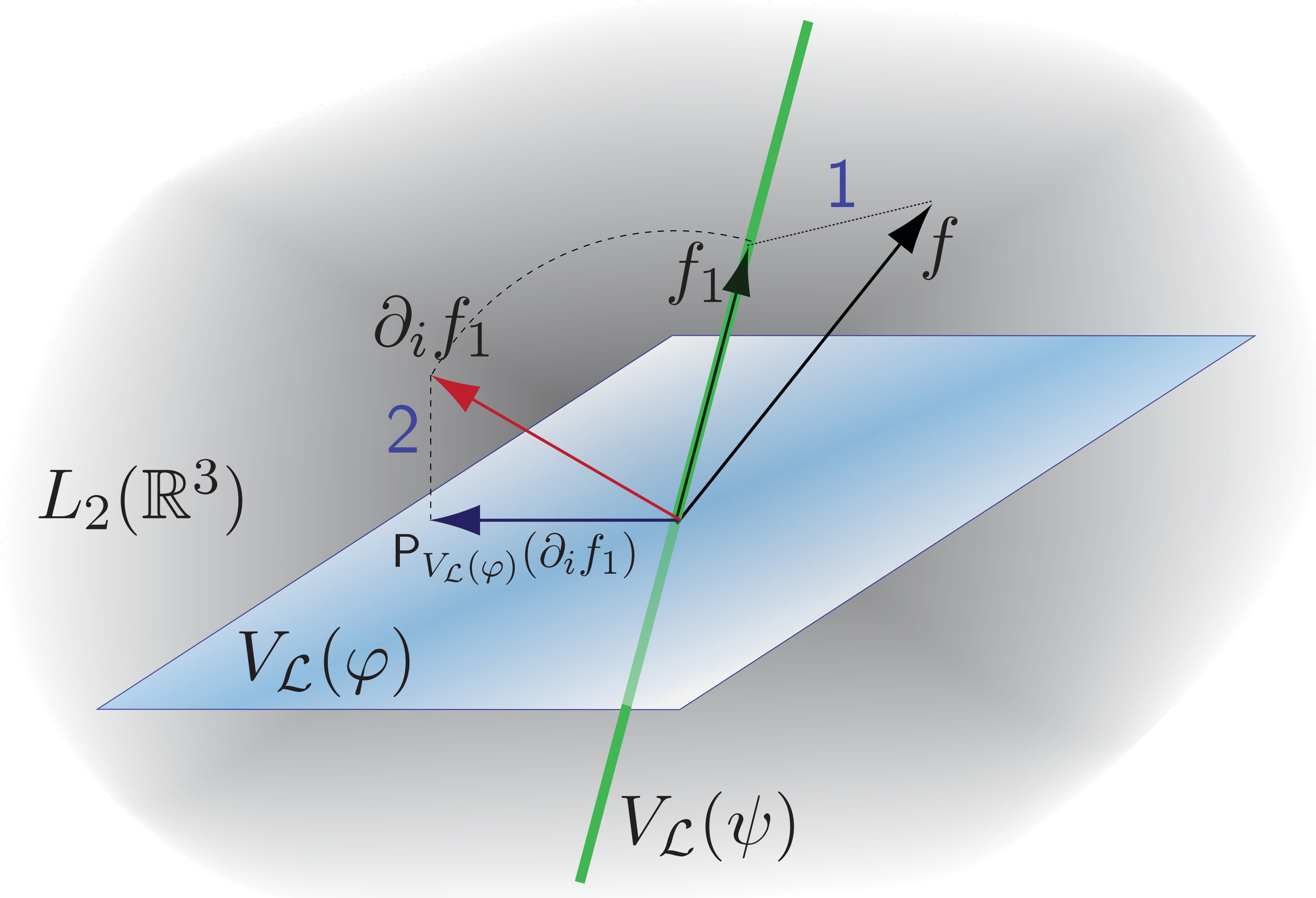
2. Orthogonally project the analytical derivative to the target space

$$f_{2,i}(\boldsymbol{x}) := \left(\mathsf{P}_{V_{\mathcal{L}(\varphi)}} \partial_i f_1\right)(\boldsymbol{x})$$
$$= \sum_{\boldsymbol{k}} ((f * p_1) * \mathring{d}_i)[\boldsymbol{k}]\, \varphi_{\boldsymbol{k}}(\boldsymbol{x})$$

Convolve with a <u>derivative</u> filter given by

$$\mathring{d}_i[\boldsymbol{l}] = \langle \partial_i \psi, \mathring{\varphi}_{\boldsymbol{l}} \rangle$$

Two Stage Gradient Approximation



2. Orthogonally project the analytical derivative to the target space

$$f_{2,i}(\boldsymbol{x}) := \left(\mathsf{P}_{V_{\mathcal{L}(\varphi)}} \partial_i f_1\right)(\boldsymbol{x})$$
$$= \sum_{\boldsymbol{k}} ((f * p_1) * \mathring{d}_i)[\boldsymbol{k}]\, \varphi_{\boldsymbol{k}}(\boldsymbol{x})$$

Convolve with a derivative filter given by

$$\mathring{d}_i[\boldsymbol{l}] = \langle \partial_i \psi, \mathring{\varphi}_{\boldsymbol{l}} \rangle$$

dual basis

Implementation

- Inner products easily computed
  ...using B-splines on CC, box-splines on BCC [Entezari *et al.* 2008]

- Filters are not compact
  ...implement in the Fourier domain during preprocessing

- Filter quality determined by the order of intermediate space
  ...choose a higher-order intermediate space [Alim *et al.* 2010]

Comparison + Results

Framework Comparison

Taylor Series	Approximation Spaces
Discrete kernel not used in filter design	**Continuous (feels discrete)** filters optimized for kernel
Compact filters can be implemented on the fly	**Filters have infinite support** need to compute a gradient volume
High quality tune by reducing the truncation error	**Superior quality** tune by choosing intermediate space

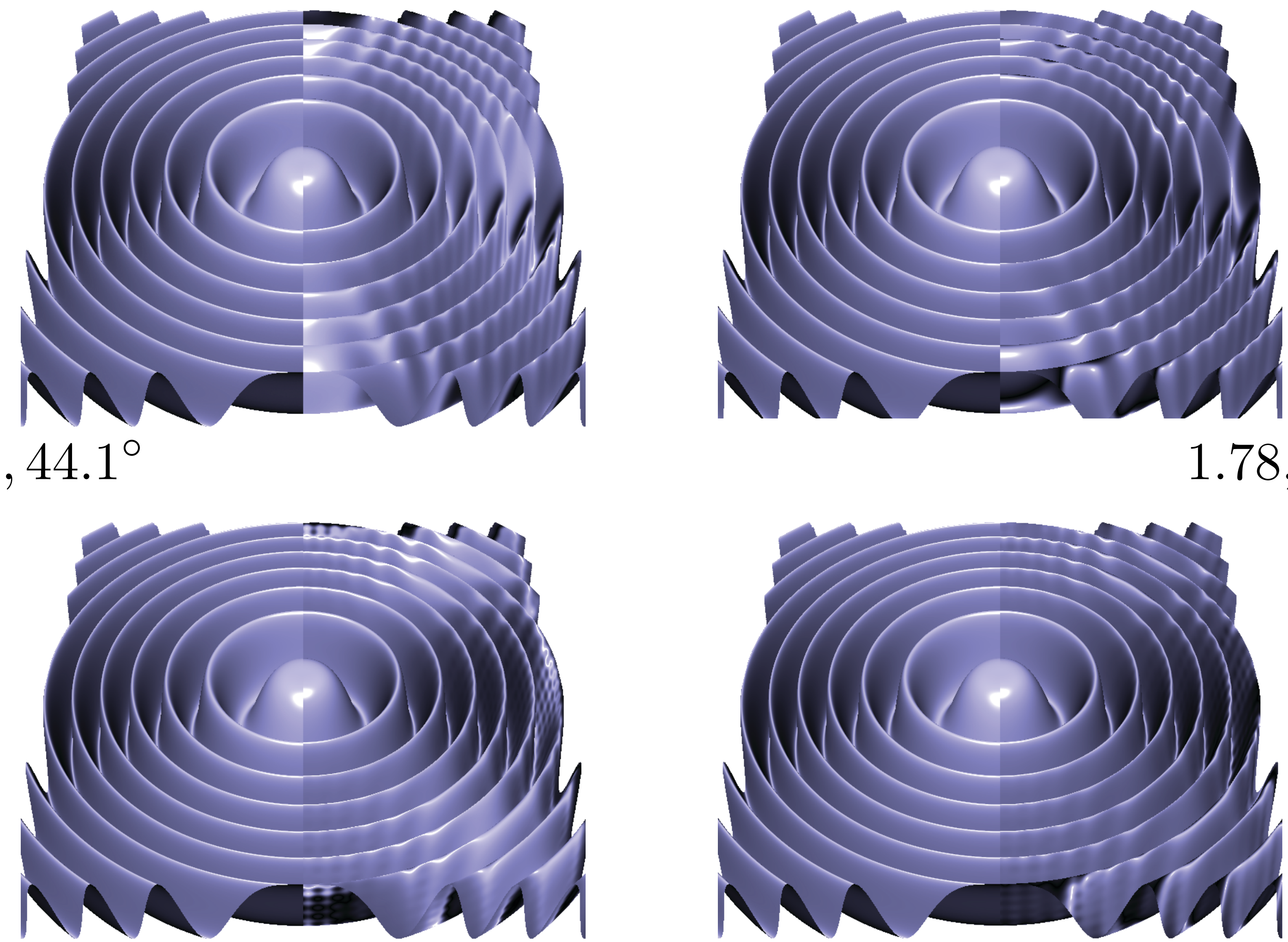
Quantitative Comparison

Fourth order filters

Taylor | OP

4-cd | *cc*

CC



$2.79, 44.1^\circ$

$1.78, 23.8^\circ$

BCC

$2.42, 28.7^\circ$ *OPT26*

QQ $\mathbf{1.38, 19.4^\circ}$

Qualitative Comparison

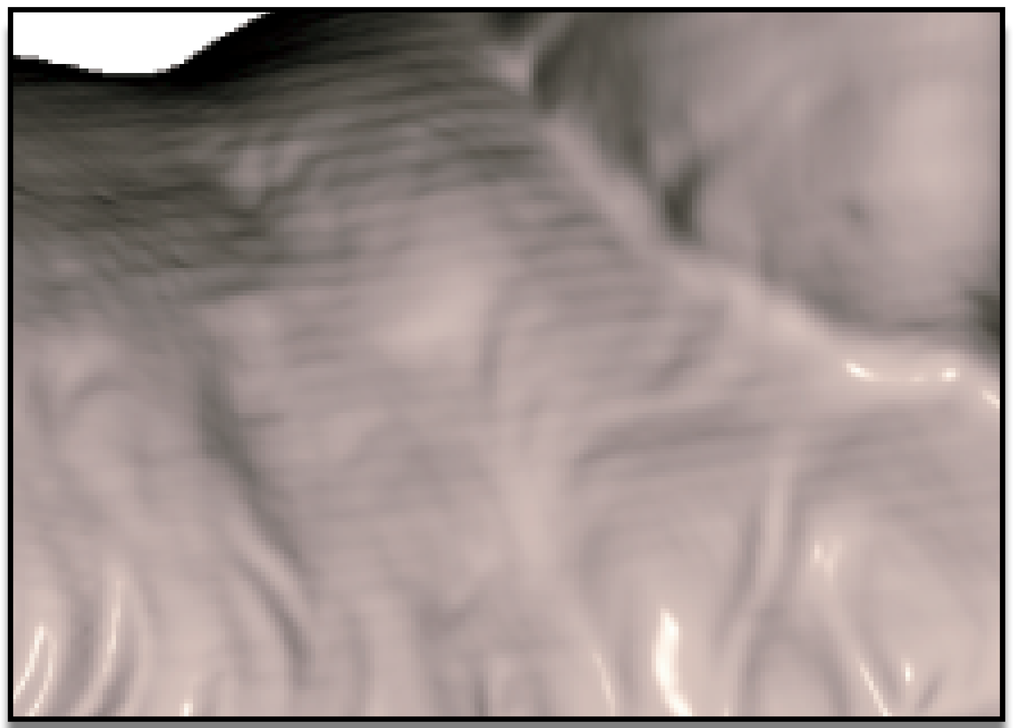
Second order filters on CC + linear interpolation

Taylor

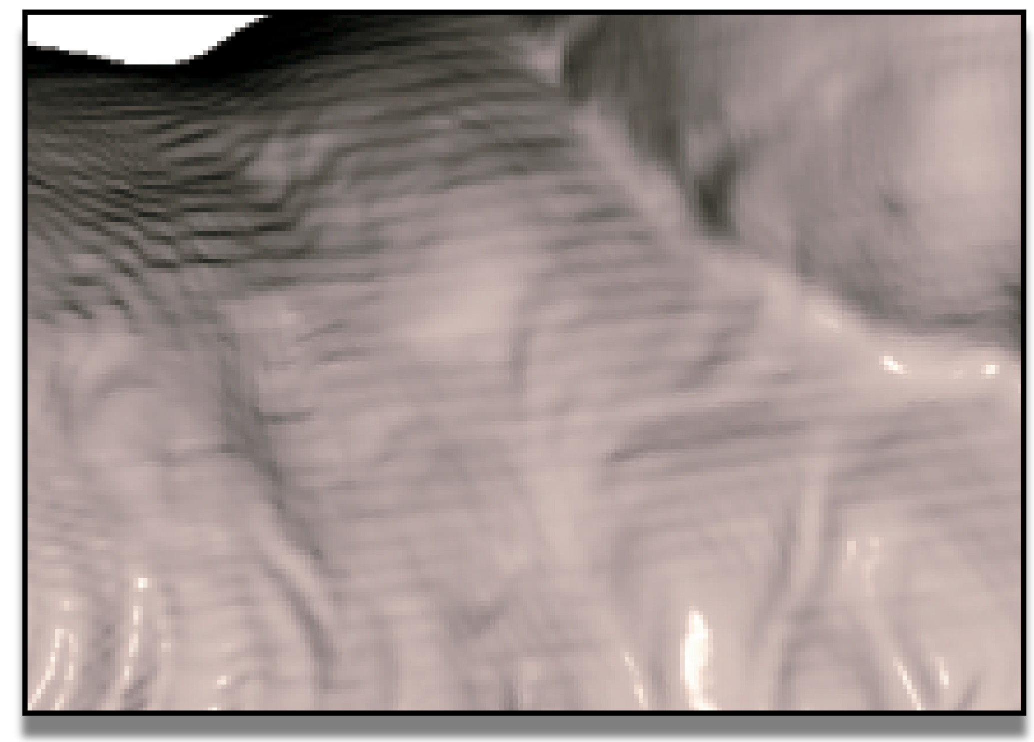
OP

Qualitative Comparison

Second order filters on CC + linear interpolation



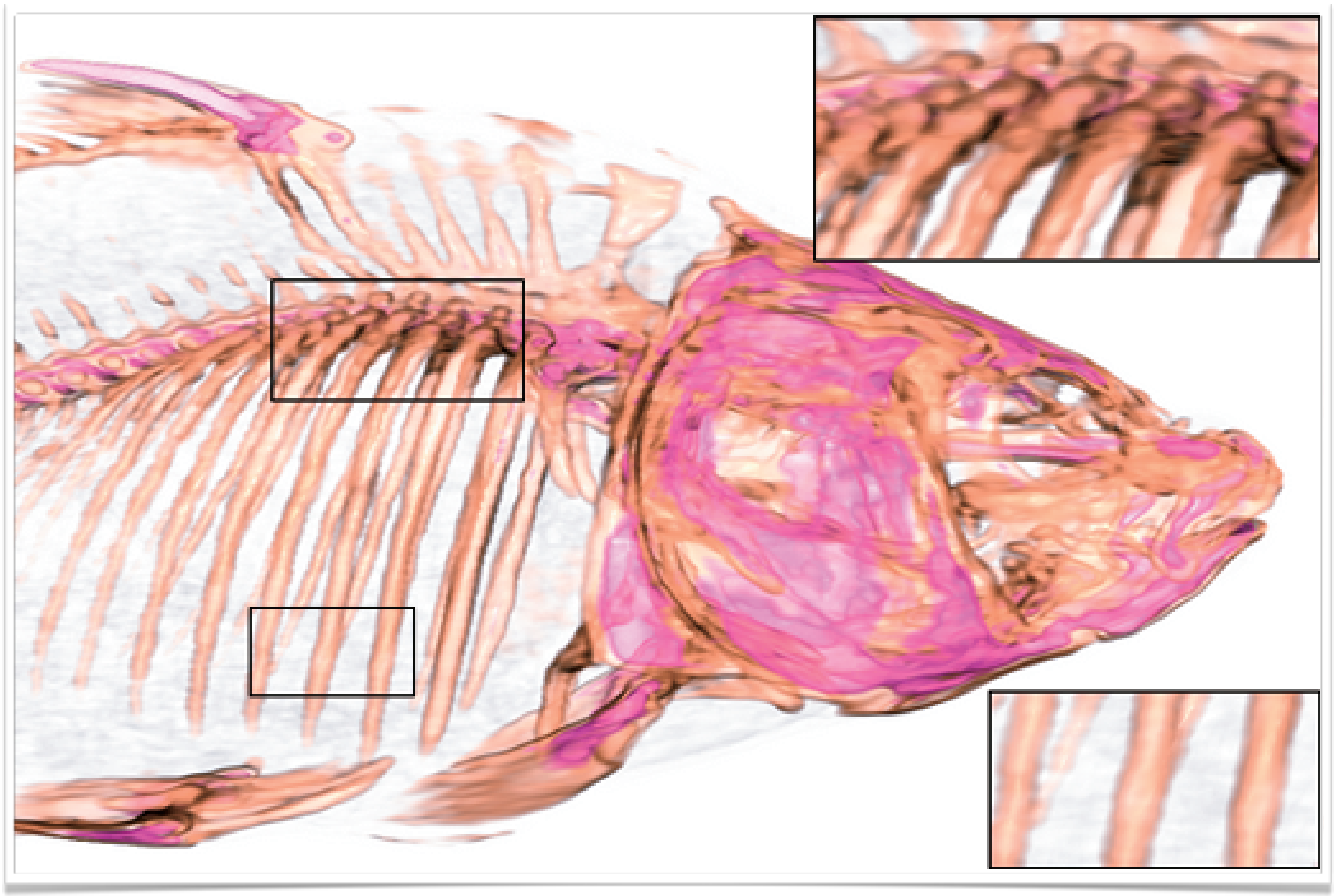
Taylor



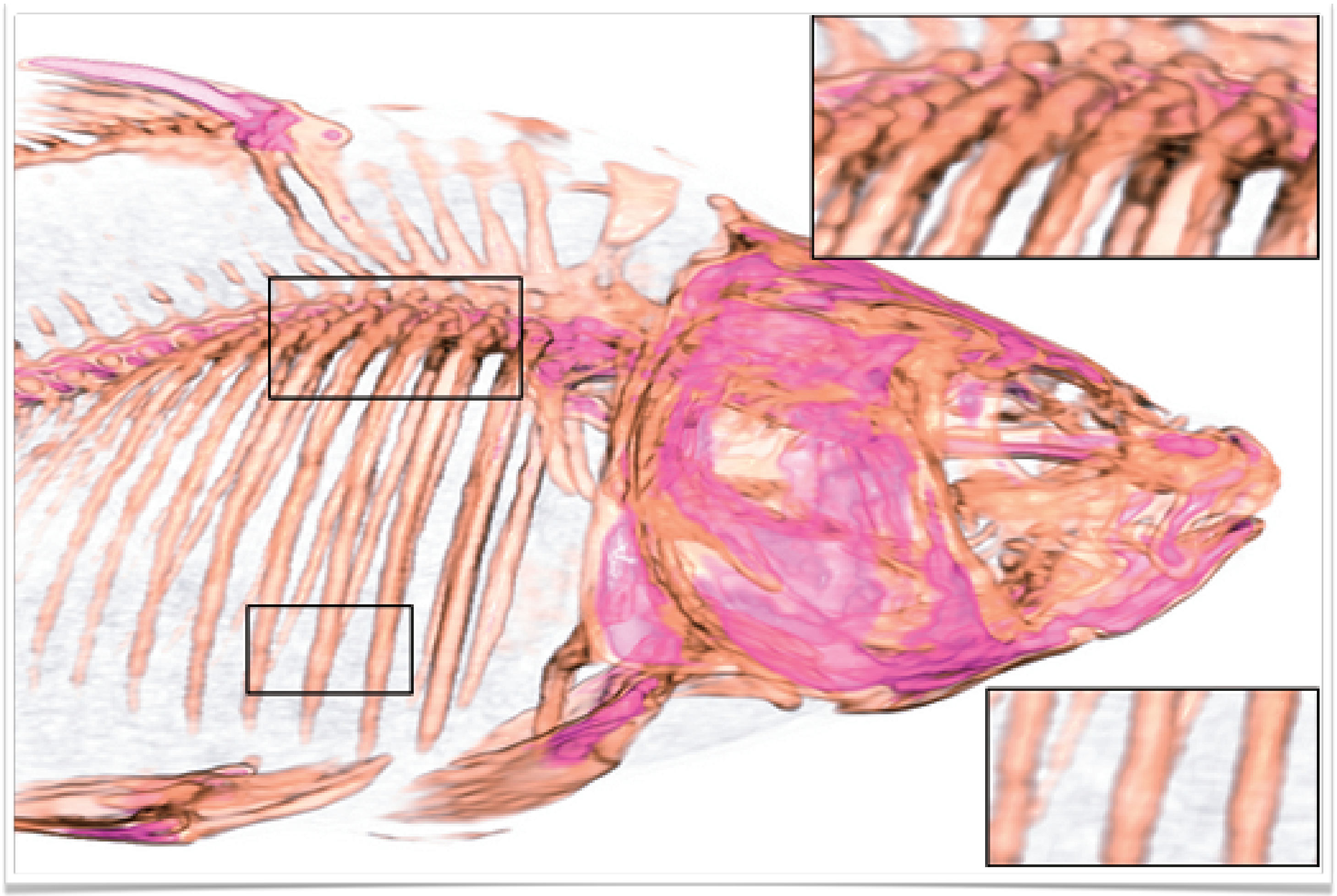
OP

Qualitative Comparison

Fourth order filters on BCC + quintic box-spline interpolation



Taylor



OP

Conclusion

Contributions

Two novel gradient estimation framework

- Taylor series framework for filter design
  ...easily extends to other types of filters

- Two-stage orthogonal projection framework
  ...easily handles other types of operators

Acknowledgements

Laurent Condat, *GREYC Lab*

Alireza Entezari, *University of Florida*

Dimitri Van De Ville, *University of Geneva*

Source code available at:

http://www.sfu.ca/~ualim/

NSERC CRSNG Natural Science and Engineering Research Council of Canada

Thank you for your attention