How many rays do we need?

Christian Lessig
Otto-von-Guericke-Universität Magdeburg
Objective
Objective

Rectilinear transport in known scene
Objective

Rectilinear transport in known scene
Objective

Rectilinear transport in known scene

\[ \hat{f} \in L_2([0, 1]^2) \]

\[ f \in \mathcal{H} \subset L_2, \dim(\mathcal{H}) < \infty \]
Objective

Rectilinear transport in known scene

\[ G \]

(\text{close-to-}) optimal for

\[ \| \hat{f} - f \| < \epsilon \]

\[ \hat{f} \in L_2([0, 1]^2) \]

\[ f \in \mathcal{H} \subset L_2, \dim(\mathcal{H}) < \infty \]
Objective

Rectilinear transport in known scene

\[
G \quad \overset{\text{minimal # of rays for}}{\longrightarrow} \quad \| \hat{f} - f \| < \epsilon
\]

\[
\hat{f} \in L_2([0, 1]^2)
\]

\[
f \in \mathcal{H} \subset L_2, \text{dim}(\mathcal{H}) < \infty
\]
Objective

Rectilinear transport in known scene

$G$

minimal # of rays for

$\| \hat{f} - f \| \lesssim N^{-\alpha}$

$\hat{f} \in L_2([0, 1]^2)$

$f \in \mathcal{H} \subset L_2, \dim(\mathcal{H}) < \infty$
Objective

Rectilinear transport in known scene

\[
G \quad \text{minimal # of rays for} \quad \| \hat{f} - f \| \lesssim N^{-\alpha}
\]

- Superresolution
- Progressive
- Varying resolution
- Parallelizable

\[
\hat{f} \in L_2([0, 1]^2)
\]

\[
f \in \mathcal{H} \subset L_2, \dim(\mathcal{H}) < \infty
\]
Computational costs

\[ C = N \cdot C_p \]

- \( N \): Number of pixels
- \( C_p \): Cost per pixel
Computational costs

\[ C = N \cdot C_p \]

- \( C \): Cost
- \( N \): Number of pixels
- \( C_p \): Cost per pixel

(last bounce)
Number of “pixels”
Number of “pixels”
Number of “pixels”

compression

$\epsilon_{rel} = 6.44 \times 10^{-4}$

100%

8.48%
Number of “pixels”
Number of “pixels”

- wavelets (quasi)
- curvelets
- shearlets
- contourlets

\[ \| \hat{f} - f \| \lesssim N^{-\alpha} \]
Number of “pixels”
Number of “pixels”

\[ f(x) = \sum_{i \in U \subset r\mathbb{Z}^2} f_i \chi_i(x) \]
Number of “pixels”

\[ f(x) = \sum_{i \in U \subseteq r\mathbb{Z}^2} f_i \chi_i(x) \]
Number of “pixels”

\[ f(x) = \sum_{i \in U \subseteq r\mathbb{Z}^2} f_i \chi_i(x) \]

\[ f(x) = \sum_{i \in \mathcal{I}} f_i \psi_i(x) \]
Number of “pixels”

\[ f(x) = \sum_{i \in U \subset \mathbb{R}^2} f_i \chi_i(x) \]

\[ f(x) = \sum_{i \in \mathcal{I}} f_i \psi_i(x) \approx (2^{-l} k, 2^l \xi) \]
Number of “pixels”

\[
f(x) = \sum_{i \in U \subset \mathbb{R}^2} f_i \chi_i(x) \]

\[
f(x) = \sum_{i \in \mathcal{I}} f_i \psi_i(x) \approx (2^{-l}k, 2^l \xi)\]
Number of “pixels”

\[ f(x) = \sum_{i \in U \subset r\mathbb{Z}^2} f_i \chi_i(x) \]

\[ f(x) = \sum_{i \in \mathcal{I}} f_i \psi_i(x) \approx (2^{-l}k, 2^l \xi) \]
Number of “pixels”

\[ f(x) = \sum_{i \in U \subset r\mathbb{Z}^2} f_i \chi_i(x) \]

Hierarchical: progressive

\[ f(x) = \sum_{i \in I} f_i [\psi_i(x)] \approx (2^{-l}k, 2^l \xi) \]
Number of “pixels”

Locality: varying resolution

Hierarchical: progressive

\[ f(x) = \sum_{i \in U \subseteq r\mathbb{Z}^2} f_i \chi_i(x) \]

\[ f(x) = \sum_{i \in \mathcal{I}} f_i \psi_i(x) \approx (2^{-l} k, 2^l \xi) \]
Number of “pixels”
Number of “pixels”
Number of “pixels”

- Number of pixels: #pixels
- Number of indoor non-zero elements: #nnz indoor
- Number of foliage non-zero elements: #nnz foliage

Graph showing the relationship between the number of pixels and super-resolution.
Number of “pixels”

\[ f(x) = \sum_{i \in U \subset r \mathbb{Z}^2} f_i \chi_i(x) \]

\[ f(x) = \sum_{i \in \mathcal{I}} f_i \left[ \psi_i(x) \right] \approx \left( 2^{-l} k, 2^l \xi \right) \]
Number of “pixels”

\[ f(x) = \sum_{i \in U \subset r\mathbb{Z}^2} f_i \chi_i(x) \]

\[ f(x) = \sum_{i \in \mathcal{I}} f_i \psi_i(x) \approx (2^{-l}k, 2^l \xi) \]
Number of "pixels"

\[ f(x) = \sum_{i \in U \subset r\mathbb{Z}^2} f_i \chi_i(x) \]

depends on image function

\[ f(x) = \sum_{i \in L} f_i \psi_i(x) \approx (2^{-l}k, 2^l \xi) \]
Computational costs

\[ C = N \cdot C_p \]

- \( C \): Computational costs
- \( N \): Number of "pixels"
- \( C_p \): Cost per pixel
Computational costs

\[ C = N \cdot C_p \]

- Use sparse, adaptive image representation

Number of “pixels” \( N \)

Cost per pixel \( C_p \)
Cost per “pixel”
Cost per "pixel"
Cost per “pixel”
Cost per “pixel”
Cost per “pixel”
Cost per “pixel”
Cost per "pixel"

\[ \{ f(x_i) \} \in \mathbb{R}^m \quad \overset{A}{\longrightarrow} \quad \{ f_i \} \in \mathbb{R}^n \]
Cost per “pixel”

\[
\{ f(x_i) \} \in \mathbb{R}^m
\]

\[
f(x_j) = \sum_{i \in I} f_i \phi_i(x_j)
\]

\[
\{ f_i \} \in \mathbb{R}^n
\]
Cost per “pixel”

\[ f(x_j) = \sum_{i \in I} f_i \phi_i(x_j) \]
Cost per “pixel”

\[ f(x_j) = \sum_{i \in I} f_i \phi_i(x_j) \]

\[ \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{pmatrix} = \begin{pmatrix} \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_n(x_1) \\ \phi_2(x_2) & \phi_3(x_2) & \cdots & \phi_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n-1}(x_m) & \phi_n(x_m) \end{pmatrix} \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} \]
Cost per "pixel"

\[ f(x_j) = \sum_{i \in I} f_i \phi_i(x_j) \]

\[
\begin{pmatrix}
    f(x_1) \\
    \vdots \\
    f(x_m)
\end{pmatrix}
= 
\begin{pmatrix}
    \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_n(x_1) \\
    \phi_2(x_2) & \phi_3(x_2) & \cdots & \phi_n(x_2) \\
    \vdots & \vdots & \ddots & \vdots \\
    \phi_{n-1}(x_m) & \phi_n(x_m)
\end{pmatrix}
\begin{pmatrix}
    f_1 \\
    \vdots \\
    f_n
\end{pmatrix}
\]
Cost per “pixel”

\[ \{ f(x_i) \} \in \mathbb{R}^m \]

\[ A \]

\[ \bar{f}(x_i) = B \bar{f}_i \]

\[ \{ f_i \} \in \mathbb{R}^n \]
Cost per “pixel”

\[ \{ f(x_i) \} \in \mathbb{R}^m \]

\[ \tilde{f}_i = B^+ \tilde{f}(x_i) \]

\[ \tilde{f}(x_i) = B \tilde{f}_i \]

\[ \{ f_i \} \in \mathbb{R}^n \]
Cost per “pixel”

\[
\{ f(x_i) \} \in \mathbb{R}^m
\]

\[
\tilde{f}(x_i) = B \tilde{f}_i
\]

\[
\tilde{f}_i = B^+ \tilde{f}(x_i)
\]

\[
\{ f_i \} \in \mathbb{R}^n
\]
Cost per “pixel”

Example: pixel basis

\[
B = \begin{pmatrix}
\phi_1(x_1) & \phi_2(x_1) & \cdots \\
\phi_2(x_2) & \phi_3(x_1) & \cdots \\
\vdots & \vdots & \ddots \\
\cdots & \cdots & \cdots & \phi_{n-1}(x_m) & \phi_n(x_m)
\end{pmatrix}
\]
Cost per "pixel"

Example: pixel basis

\[ B = \begin{pmatrix}
\chi_1(x_1) & \chi_2(x_1) & \cdots \\
\chi_2(x_2) & \chi_3(x_1) & \cdots \\
\cdots & \cdots & \cdots \\
\cdots & \chi_{n-1}(x_m) & \chi_n(x_m)
\end{pmatrix} \]
Cost per “pixel”

Example: pixel basis

\[ B = \begin{pmatrix}
1 & 0 & \ldots \\
1 & 0 & \ldots \\
& \ddots & \ddots \\
\vdots & \vdots & \ddots \\
\vdots & 0 & 1 \\
\vdots & 0 & 1 \\
\end{pmatrix} \]
Cost per “pixel”

Example: pixel basis

\[ B = \begin{pmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & 0 \\ \vdots & 0 & 0 \end{pmatrix} \]
Cost per “pixel”

Example: pixel basis

\[ B^+ = \begin{pmatrix} \cdots \\ \cdots \\ \cdots \\ \cdots \end{pmatrix} \]
Cost per “pixel”

Example: pixel basis

\[ B^+ = \begin{pmatrix} 
1/N_1 & 1/N_1 & \cdots & 1/N_1 \\
& 1/N_2 & 1/N_2 & \cdots \\
& & \ddots & \ddots \\
& & & 1/N_n & 1/N_n 
\end{pmatrix} \]
Cost per “pixel”

Example: pixel basis

\[ B^+ = \begin{pmatrix} 
1/N_1 & 1/N_1 & \cdots & 1/N_2 & 1/N_2 \\
1/N_2 & 1/N_2 & \cdots & 1/N_n & 1/N_n \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
1/N_n & 1/N_n & \cdots & 1/N_1 & 1/N_1 \\
\end{pmatrix} \]

\[ f_i = \frac{1}{N_i} \sum_{j=1}^{N_i} f(x_j) \]
Cost per “pixel”

Example: wavelet basis

\[ B = \begin{pmatrix}
\psi_1(x_1) & \psi_2(x_1) & \ldots \\
\psi_2(x_2) & \psi_3(x_1) & \ldots \\
\vdots & \vdots & \ddots \\
\psi_{n-1}(x_m) & \psi_n(x_m)
\end{pmatrix} \]
Cost per “pixel”

Example: wavelet basis

\[ B = \begin{pmatrix} \vdots \\ \vdots \end{pmatrix} \]
Cost per “pixel”

Example: wavelet basis

\[ B = \begin{pmatrix} \vdots \end{pmatrix} \]

\[ B^+ \neq f_i = \frac{1}{N} \sum_{i=1}^{N} f(x_i) \psi(x_i) \]
Cost per “pixel”

Example: wavelet basis

\[ B = \begin{pmatrix} \vdots & \vdots \\ \vdots & \vdots \\ \end{pmatrix} \]

Multi-res structure: can be solved in \( O(n) \) time using multi-grid.
Cost per “pixel”

Example: wavelet basis

\[ B = \]

Multi-res structure: can be solved in \( O(n) \) time using multi-grid.
Computational costs

\[ C = N \cdot C_p \]

- Use sparse, adaptive image representation

\# Computational costs

\[ C = N \cdot C_p \]

- Use sparse, adaptive image representation
Computational costs

\[ C = N \cdot C_p \]

- Use sparse, adaptive image representation
- Use at least \( n \) samples
- Solve using multi-grid
Computational costs

\[ C = N \cdot C_p \]

- Use sparse, adaptive image representation
- Use at least \( n \) samples
- Solve using multi-grid

Number of "pixels"
Cost per pixel

minimize by

last bounce
Another perspective

\[ \{ f(x_i) \} \in \mathbb{R}^m \]

\[ \bar{f}_i = B^+ \bar{f}(x_i) \]

\[ \bar{f}(x_i) = B \bar{f}_i \]

\[ \{ f_i \} \in \mathbb{R}^n \]
Another perspective

\[
\{ f(x_i) \} \in \mathbb{R}^m
\]

\[
\bar{f}_i = B^+ \bar{f}(x_i)
\]

\[
\bar{f}(x_i) = B \bar{f}_i
\]

\[
\{ f_i \} \in \mathbb{R}^n
\]
Another perspective

$$\{ f(x_i) \} = \{ \delta_{x_i}(f) \}$$
Another perspective

\[ \{ f(x_i) \} = \{ \delta_{x_i}(f) \} \]
Another perspective

\[
\begin{align*}
\{ f(x_i) \} &= \{ \delta x_i(f) \} \\
\underbrace{f = \hat{f} + \epsilon} \\
f(x) &= L_{\text{out}}(x)
\end{align*}
\]
Another perspective

\[ \{ f(x_i) \} = \{ \delta x_i(f) \} \]

\[ f = \hat{f} + \epsilon \]

\[ \Rightarrow \delta x_i(f) \approx \delta x_i(\hat{f}) \]

\[ f(x) = L_{out}(x) \]
Another perspective

\[
\{ f(x_i) \} = \{ \delta x_i (f) \}
\]

\[
f = \hat{f} + \epsilon
\]

\[
\delta x_i (f) \approx \delta x_i (\hat{f})
\]

\(\delta_x\) is a continuous functional

\[
f(x) = L_{\text{out}}(x)
\]
Another perspective

\[
\{ f(x_i) \} = \{ \delta x_i (f) \}
\]

\[
f = \hat{f} + \epsilon
\]

\[
\Rightarrow \quad \delta x_i (f) \approx \delta x_i (\hat{f}) \quad \delta x \text{ is a continuous functional}
\]

\[
\Rightarrow \quad f \in \mathcal{H} \text{ with } \mathcal{H} \text{ a "reasonable" function space}
\]
Reproducing kernel Hilbert spaces

$\mathcal{H}(X)$ is Hilbert and $\delta_{x_i}(f)$ continuous, then
Reproducing kernel Hilbert spaces

$\mathcal{H}(X)$ is Hilbert and $\delta_{x_i}(f)$ continuous, then

$$\forall \bar{x} \in X, \exists k_{\bar{x}}(x) \in \mathcal{H}(X) : \left\langle k_{\bar{x}}(x), f(x) \right\rangle = f(\bar{x})$$
Reproducing kernel Hilbert spaces

\[ \mathcal{H}(X) \text{ is Hilbert and } \delta_{x_i}(f) \text{ continuous, then } \]

\[ \forall \bar{x} \in X, \exists k_{\bar{x}}(x) \in \mathcal{H}(X) : \left\langle k_{\bar{x}}(x), f(x) \right\rangle = f(\bar{x}) \]

reproducing kernel
Example: $\mathcal{H}([-1, 1]) = \text{span}_{n=1 \ldots N} \{ P_n(x) \}$
Example: $\mathcal{H}([-1, 1]) = \text{span}_{n=1\ldots N}\{P_n(x)\}$
Example: $\mathcal{H}([-1, 1]) = \text{span}_{n=1\ldots N}\{P_n(x)\}$
Example: $\mathcal{H}([-1, 1]) = \text{span}_{n=1 \ldots N} \{ P_n(x) \}$
Example: $\mathcal{H}([-1, 1]) = \text{span}_{n=1 \ldots N} \{ P_n(x) \}$
Example: $\mathcal{H}([-1, 1]) = \text{span}_{n=1 \ldots N} \{ P_n(x) \}$
Example: $\mathcal{H}([-1, 1]) = \text{span}_{n=1}^{N} \{ P_n(x) \}$
Example: \( \mathcal{H}([-1, 1]) = \text{span}_{n=1\ldots N} \{ P_n(x) \} \)
Reproducing kernel Hilbert spaces

$\mathcal{H}(X)$ is Hilbert and $\delta_{x_i}(f)$ continuous, then

$$\forall \bar{x} \in X, \exists k_{\bar{x}}(x) \in \mathcal{H}(X) : \left\langle k_{\bar{x}}(x), f(x) \right\rangle = f(\bar{x})$$
Reproducing kernel Hilbert spaces

\( \mathcal{H}(X) \) is Hilbert and \( \delta_{x_i}(f) \) continuous, then

\[
\forall \bar{x} \in X, \exists k_{\bar{x}}(x) \in \mathcal{H}(X) : \left\langle k_{\bar{x}}(x), f(x) \right\rangle = f(\bar{x})
\]
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\}, \lambda_i \in X \)
Reproducing kernel Hilbert spaces

Let $\{\lambda_i\}$, $\lambda_i \in X$ s. t.

$$\text{span}(k_{\lambda_i}(x)) = \mathcal{H}(X)$$
Example: $\mathcal{H}([-1, 1]) = \text{span}_{n=1}^{N} \{P_n(x)\}$
Example: $\mathcal{H}([-1, 1]) = \text{span}_{n=1 \ldots N} \{ P_n(x) \}$
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\}, \lambda_i \in X \) s. t.

\[
\text{span}(k_{\lambda_i}(x)) = \mathcal{H}(X)
\]
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\}, \lambda_i \in X \) s. t.

\[
\text{span}(k_{\lambda_i}(x)) = \mathcal{H}(X)
\]
Reproducing kernel Hilbert spaces

Let \( \{ \lambda_i \} \), \( \lambda_i \in X \) s. t.

\[
\text{span}(k_{\lambda_i}(x)) = \mathcal{H}(X)
\]

spanning set
Reproducing kernel Hilbert spaces

Let \( \{ \lambda_i \}, \lambda_i \in X \) s. t.

\[
\text{span}(k_{\lambda_i}(x)) = \mathcal{H}(X)
\]
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\}, \lambda_i \in X \) s. t.

\[
\text{span}(k_{\lambda_i}(x)) = \mathcal{H}(X)
\]

no spanning set
Example: $\mathcal{H}([-1, 1]) = \text{span}_{n=1\ldots N} \{ P_n(x) \}$
Example: $\mathcal{H}([-1, 1]) = \text{span}_{n=1\ldots N}\{P_n(x)\}$
Example: $\mathcal{H}([-1, 1]) = \text{span}_{n=1}^{N} \{P_n(x)\}$
Reproducing kernel Hilbert spaces

Let \{\lambda_i\}, \lambda_i \in X \text{ s. t.}

\[ \operatorname{span}(k_{\lambda_i}(x)) = \mathcal{H}(X) \]
Reproducing kernel Hilbert spaces

Let \( \{ \lambda_i \} \), \( \lambda_i \in X \) s. t.

\[
\text{span}(k_{\lambda_i}(x)) = \mathcal{H}(X)
\]

\( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\}, \lambda_i \in X \) s. t.

\[
\text{span}(k_{\lambda_i}(x)) = \mathcal{H}(X)
\]

\( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f_i k_{\lambda_i}(x)
\]
Reproducing kernel Hilbert spaces

\[ v = \sum_{i=1}^{2} v_i e_i \]
Reproducing kernel Hilbert spaces

\[ v = \sum_{i=1}^{2} \langle v, e_i \rangle e_i \]
Reproducing kernel Hilbert spaces

\[ v = \sum_{i=1}^{2} v_i u_i \]
Reproducing kernel Hilbert spaces

\[ v = \sum_{i=1}^{2} \langle v, u_i \rangle u_i \]
Reproducing kernel Hilbert spaces

\[ v = \sum_{i=1}^{2} \langle v, \tilde{u}_i \rangle u_i \]
Example: $\mathcal{H}([-1, 1]) = \text{span}_{n=1}^{N} \{ P_n(x) \}$
Example: $\mathcal{H}([-1, 1]) = \text{span}_{n=1}^{N} \{ P_n(x) \}$
Reproducing kernel Hilbert spaces

\[ v = \sum_{i=1}^{2} \langle v, \tilde{u}_i \rangle u_i \]
Reproducing kernel Hilbert spaces

\[ v = \sum_{i=1}^{2} \langle v, u_i \rangle \tilde{u}_i \]
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\} \), \( \lambda_i \in X \) s. t. \( \{k\lambda_i(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f_i k\lambda_i(x)
\]
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\} \), \( \lambda_i \in X \) s. t. \( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f_i \ k_{\lambda_i}(x) = \sum_{i=1}^{N} \left< f(y), \tilde{k}_{i}(y) \right> k_{\lambda_i}(x)
\]
Reproducing kernel Hilbert spaces

Let \{\lambda_i\}, \lambda_i \in X s. t. \{k_{\lambda_i}(x)\} forms a basis / frame for \mathcal{H}(X)

\[ f(x) = \sum_{i=1}^{N} f_i k_{\lambda_i}(x) = \sum_{i=1}^{N} \left\langle f(y), \tilde{k}_i(y) \right\rangle k_{\lambda_i}(x) \]

\[ = \sum_{i=1}^{N} \left\langle f(y), k_{\lambda_i}(y) \right\rangle \tilde{k}_i(x) \]
Reproducing kernel Hilbert spaces

Let \( \{ \lambda_i \}, \lambda_i \in X \) s. t. \( \{ k_{\lambda_i}(x) \} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f_i k_{\lambda_i}(x) = \sum_{i=1}^{N} \langle f(y), \tilde{k}_i(y) \rangle k_{\lambda_i}(x)
\]

\[
= \sum_{i=1}^{N} \langle f(y), k_{\lambda_i}(y) \rangle \tilde{k}_i(x)
\]
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\} \), \( \lambda_i \in X \) s. t. \( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f_{i} k_{\lambda_{i}}(x) = \sum_{i=1}^{N} \left< f(y), \tilde{k}_{i}(y) \right> k_{\lambda_{i}}(x) \]

\[
= \sum_{i=1}^{N} \left< f(y), k_{\lambda_{i}}(y) \right> \tilde{k}_{i}(x) = f(\lambda_{i})
\]
Reproducing kernel Hilbert spaces

Let \( \{ \lambda_i \} \), \( \lambda_i \in X \) s. t. \( \{ k_{\lambda_i}(x) \} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f_i k_{\lambda_i}(x) = \sum_{i=1}^{N} \langle f(y), \tilde{k}_i(y) \rangle k_{\lambda_i}(x)
\]

\[
= \sum_{i=1}^{N} \langle f(y), k_{\lambda_i}(y) \rangle \tilde{k}_i(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x)
\]

\[
= f(\lambda_i)
\]
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\} \), \( \lambda_i \in X \) s. t. \( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x)
\]
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\}, \lambda_i \in X \) s. t. \( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x)
\]

reproducing kernel frame
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\}, \lambda_i \in X \) s. t. \( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x)
\]

reproducing kernel frame \( \iff \) sampling theorem
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\} \), \( \lambda_i \in X \) s. t. \( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f(\lambda_i) \hat{k}_i(x)
\]

reproducing kernel frame \( \Leftrightarrow \) sampling theorem

Shannon: \( f(x) = \sum_{i=-\infty}^{\infty} f(i) \text{sinc}(x - i) \)
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\}, \lambda_i \in X \) s. t. \( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x)
\]

reproducing kernel frame \( \iff \) sampling theorem

Shannon: \[
f(x) = \sum_{i=-\infty}^{\infty} f(i) \text{sinc}(x - i)
\]

\[
= \sum_{i=-\infty}^{\infty} \langle f(y), \text{sinc}(y - i) \rangle \text{sinc}(x - i)
\]
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\} \), \( \lambda_i \in X \) s. t. \( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x)
\]

reproducing kernel frame \( \iff \) sampling theorem
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\}, \lambda_i \in X \) s. t. \( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x)
\]

reproducing kernel frame \( \iff \) sampling theorem

- arbitrary locations
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\}, \lambda_i \in X \) s. t. \( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x)
\]

reproducing kernel frame \( \iff \) sampling theorem

- arbitrary locations
- non-bandlimited functions
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\} \), \( \lambda_i \in X \) s. t. \( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x)
\]

reproducing kernel frame \( \Leftrightarrow \) sampling theorem

- arbitrary locations
- non-bandlimited functions
- over-sampling
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\}, \lambda_i \in X \) s. t. \( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x)
\]

reproducing kernel frame \( \iff \) sampling theorem

- arbitrary locations
- non-bandlimited functions
- over-sampling
- arbitrary domains
Reproducing kernel Hilbert spaces

Let \( \{\lambda_i\} \), \( \lambda_i \in X \) s. t. \( \{k_{\lambda_i}(x)\} \) forms a basis / frame for \( \mathcal{H}(X) \)

\[
f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x)
\]

reproducing kernel frame \( \IFF \) sampling theorem

- arbitrary locations
- non-bandlimited functions
- over-sampling
- arbitrary domains
- optimal locations for setting
Reproducing kernel Hilbert spaces

\[ \{ f(x_i) \} \in \mathbb{R}^m \]

\[ \bar{f}_i = B^+ \bar{f}(x_i) \]

\[ \bar{f}(x_i) = B \bar{f}_i \]

\[ \{ f_i \} \in \mathbb{R}^n \]
Reproducing kernel Hilbert spaces

\[ f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x) \]
Reproducing kernel Hilbert spaces

\[ \{ f(x_i) \} \in \mathbb{R}^m \]

\[ \bar{f}_i = B^+ \bar{f}(x_i) \]

\[ \bar{f}(x_i) = B \bar{f}_i \]

\[ \{ f_i \} \in \mathbb{R}^n \]

\[ f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x) \]

\[ f(x) = \sum_{i=1}^{N} f_i \phi_i(x) \]
Reproducing kernel Hilbert spaces

\[ \{ f(x_i) \} \in \mathbb{R}^m \]

\[ \tilde{f}_i = B^+ \tilde{f}(x_i) \]

\[ \tilde{f}(x_i) = B \tilde{f}_i \]

\[ \{ f_i \} \in \mathbb{R}^n \]

\[ f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x) \]

change of basis

\[ f(x) = \sum_{i=1}^{N} f_i \phi_i(x) \]
Reproducing kernel Hilbert spaces

\[ f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x) \quad \text{change of basis} \quad f(x) = \sum_{i=1}^{N} f_i \phi_i(x) \]

\[ \{ f(x_i) \} \in \mathbb{R}^m \]

\[ \bar{f}_i = B^+ \bar{f}(x_i) \]

\[ \bar{f}(x_i) = B \bar{f}_i \]

\[ \{ f_i \} \in \mathbb{R}^n \]
Reproducing kernel Hilbert spaces

\[ \{ f(x_i) \} \in \mathbb{R}^m \]

\[ f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x) \]

\[ \tilde{f}_i = B^+ \tilde{f}(x_i) \]

\[ \tilde{f}(x_i) = B \tilde{f}_i \]

\[ \{ f_i \} \in \mathbb{R}^n \]

\[ f(x) = \sum_{i=1}^{N} f_i \phi_i(x) \]
Reproducing kernel Hilbert spaces

\[ \{ f(x_i) \} \in \mathbb{R}^m \]

\[ f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x) \]

\[ \tilde{f}_i = B^+ \tilde{f}(x_i) \]

sampling theorem for sparse image representation

\[ \tilde{f}(x_i) = B \tilde{f}_i \]

change of basis

\[ \{ f_i \} \in \mathbb{R}^n \]

\[ f(x) = \sum_{i=1}^{N} f_i \phi_i(x) \]
Reproducing kernel Hilbert spaces

\[ f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x) \]

\[ \bar{f}_i = B^+ \bar{f}(x_i) \]

\[ \bar{f}(x_i) = B \bar{f}_i \]

sampling theorem for sparse image representation

\[ f(x) = \sum_{i=1}^{N} f_i \phi_i(x) \]
Reproducing kernel Hilbert spaces

\[ f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x) \]

sampling theorem for sparse image representation

\[ \bar{f}_i = B^+ \bar{f}(x_i) \]
\[ \bar{f}(x_i) = B \bar{f}_i \]

\[ f(x) = \sum_{i=1}^{N} f_i \phi_i(x) \]

- optimal reconstruction kernels
Reproducing kernel Hilbert spaces

\[ f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x) \]

sampling theorem for sparse image representation

\[ \bar{f}_i = B^+ \bar{f}(x_i) \]
\[ \bar{f}(x_i) = B \bar{f}_i \]

\[ f(x) = \sum_{i=1}^{N} f_i \phi_i(x) \]

- optimal reconstruction kernels
- designed for non-bandlimited signals
Reproducing kernel Hilbert spaces

\[ f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x) \]

sampling theorem for sparse image representation

\[ \bar{f}_i = B^+ \bar{f}(x_i) \]
\[ \bar{f}(x_i) = B \bar{f}_i \]

\[ f(x) = \sum_{i=1}^{N} f_i \phi_i(x) \]

- optimal reconstruction kernels
- designed for non-bandlimited signals
- designed for non-uniform locations
Reproducing kernel Hilbert spaces

\[
f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x)
\]

\[
\bar{f}_i = B^+ \bar{f}(x_i)
\]

\[
\bar{f}(x_i) = B \bar{f}_i
\]

\[
f(x) = \sum_{i=1}^{N} f_i \phi_i(x)
\]

- optimal reconstruction kernels
- designed for non-bandlimited signals
- designed for non-uniform locations
- allows to optimize sampling locations
Reproducing kernel Hilbert spaces

\[ f(x) = \sum_{i=1}^{N} f(\lambda_i) \tilde{k}_i(x) \]

**sampling theorem for sparse image representation**

- optimal reconstruction kernels
- designed for non-bandlimited signals
- designed for non-uniform locations
- allows to optimize sampling locations
Recipe

1. Find (approximate) tight function space
2. Sample function
3. Construct reproducing kernel basis
Recipe
Reproducing kernel Hilbert spaces

$$\int_X f(x) \, dx$$
Reproducing kernel Hilbert spaces

\[ \int_{X} f(x) \, dx = \int_{X} \sum_{i=1}^{N} f(x_i) \tilde{k}_i(x) \, dx \]
Reproducing kernel Hilbert spaces

\[ \int_X f(x) \, dx = \int_X \sum_{i=1}^{N} f(x_i) \, \tilde{k}_i(x) \, dx \]

\[ = \sum_{i=1}^{N} f(x_i) \int_X \tilde{k}_i(x) \, dx \]
Reproducing kernel Hilbert spaces

\[
\int_X f(x) \, dx = \int_X \sum_{i=1}^N f(x_i) \tilde{k}_i(x) \, dx
\]

\[
= \sum_{i=1}^N f(x_i) \int_X \tilde{k}_i(x) \, dx
\]

\[
\{ \sum_{i=1}^N w_i f(x_i) \}
\]
Reproducing kernel Hilbert spaces

\[ \int_X f(x) \, dx = \int_X \sum_{i=1}^{N} f(x_i) \tilde{k}_i(x) \, dx \]

\[ = \sum_{i=1}^{N} f(x_i) \int_X \tilde{k}_i(x) \, dx \]

\[ = \sum_{i=1}^{N} w_i \, f(x_i) \]
Recipe

1. Find (approximate) tight function space
2. Sample function
3. Construct reproducing kernel basis
Recipe

1. Find (approximate) tight function space
2. Sample function
3. Construct reproducing kernel basis
Tight function spaces
Tight function spaces

jump discontinuity: infinite frequency content
Tight function spaces

Jump discontinuity: infinite frequency content

\[ \hat{f}(\xi) \lesssim |\xi|^{-1} \]
Tight function spaces

Jump discontinuity: infinite frequency content
\[ \hat{f}(\xi) \lesssim |\xi|^{-1} \]

Smooth region:
\[ \hat{f}(\xi) \lesssim |\xi|^{-N}, \forall N \in \mathbb{N} \]
Tight function spaces

jump discontinuity: infinite frequency content
\[ \hat{f}(\xi) \lesssim |\xi|^{-1} \]

smooth region:
\[ \hat{f}(\xi) \lesssim |\xi|^{-N}, \forall N \in \mathbb{N} \]
Tight function spaces

jump discontinuity: infinite frequency content
\[ \hat{f}(\xi) \lesssim |\xi|^{-1} \]

smooth region:
\[ \hat{f}(\xi) \lesssim |\xi|^{-N}, \forall N \in \mathbb{N} \]
Tight function spaces

jump discontinuity: infinite frequency content
\[ \hat{f}(\xi) \lesssim |\xi|^{-1} \]

smooth region:
\[ \hat{f}(\xi) \lesssim |\xi|^{-N}, \forall N \in \mathbb{N} \]
Tight function spaces

jump discontinuity: infinite frequency content
\[ \hat{f}(\xi) \lesssim |\xi|^{-1} \]

smooth region:
\[ \hat{f}(\xi) \lesssim |\xi|^{-N}, \forall N \in \mathbb{N} \]
Tight function spaces

Jump discontinuity: infinite frequency content
\[ \hat{f}(\xi) \lesssim |\xi|^{-1} \]

Smooth region:
\[ \hat{f}(\xi) \lesssim |\xi|^{-N}, \forall N \in \mathbb{N} \]
Tight function spaces

Jump discontinuity: infinite frequency content
\[ \hat{f}(\xi) \lesssim |\xi|^{-1} \]

Smooth region:
\[ \hat{f}(\xi) \lesssim |\xi|^{-N}, \forall N \in \mathbb{N} \]
Tight function spaces

Jump discontinuity: infinite frequency content

\[ \hat{f}(\xi) \lesssim |\xi|^{-1} \]

Smooth region:

\[ \hat{f}(\xi) \lesssim |\xi|^{-N}, \forall N \in \mathbb{N} \]
Tight function spaces
Tight function spaces
Tight function spaces

\[ \xi \]

\[ x \]

© Christian Lessig, 2016
Tight function spaces
Tight function spaces
Tight function spaces
Tight function spaces
Tight function spaces
Tight function spaces
Tight function spaces
Tight function spaces

\[ f(x) = \sum_{i \in \mathcal{I}} f_i \psi_i(x) \approx (2^{-l}k, 2^l \xi) \]
Tight function spaces

\[ f(x) = \sum_{i \in I} f_i \psi_i(x) \approx (2^{-l} k, 2^l \xi) \]

depends on image function
Tight function spaces

\[ \phi \left( 2^l k, 2^l \phi \right) \]
Tight function spaces
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation
Image generation

1. Find (approximate) tight function space
2. Sample function
3. Construct reproducing kernel basis
Image generation
How many rays do we need?
How many rays do we need?

» As many as there are nonzero coefficients in the sparsest signal representation (times a small constant). «
How many rays do we need?

» As many as there are nonzero coefficients in the sparsest signal representation (times a small constant). «

Recipe:

1. Find (approximate) tight function space
2. Sample function
3. Construct reproducing kernel basis