Edge detection methods based on RBF interpolation

Daniela Schenone
joint work with Lucia Romani and Milvia Rossini

Università degli Studi di Milano–Bicocca

daniela.schenone@unimib.it

February 21, 2018
The aim of our work is to propose a new edge detection method based on radial basis function interpolation.
The aim of our work is to propose a new edge detection method based on radial basis function interpolation.

Why RBFs?
The aim of our work is to propose a new edge detection method based on radial basis function interpolation.

Why RBFs?

- meshless method
Aim

The aim of our work is to propose a new edge detection method based on radial basis function interpolation.

Why RBFs?

- meshless method
- work on non-gridded data
Aim

The aim of our work is to propose a new edge detection method based on radial basis function interpolation.

Why RBFs?

- meshless method
- work on non-gridded data
- easily extended to multivariate problems
Kernels and RBFs

A symmetric kernel

\[ K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}, \]

is translation invariant on \( \mathbb{R}^d \) if there exists a function \( \Phi : \mathbb{R}^d \rightarrow \mathbb{R} \) such that

\[ K(x, y) = \Phi(x - y) \quad \forall \quad x, y \in \mathbb{R}^d. \]
Kernels and RBFs

A symmetric kernel

\[ K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}, \]

is translation invariant on \( \mathbb{R}^d \) if there exists a function \( \Phi : \mathbb{R}^d \rightarrow \mathbb{R} \) such that

\[ K(x, y) = \Phi(x - y) \quad \forall \quad x, y \in \mathbb{R}^d. \]

If \( K \) is also radial, i.e. there exists a scalar function \( \varphi : [0, +\infty) \rightarrow \mathbb{R} \) such that

\[ K(x, y) = \varphi(\|x - y\|), \quad \text{where} \quad \| \cdot \| \quad \text{is a norm}, \]

the function \( \Phi \) is called **Radial Basis Function (RBF)**.
Kernels $K$ on $\mathbb{R}^d$ can be scaled by a positive factor $\varepsilon$ in the following way

$$K(x, y; \varepsilon) = K\left(\frac{x}{\varepsilon}, \frac{y}{\varepsilon}\right) \quad \forall \ x, y \in \mathbb{R}^d.$$
Kernels $K$ on $\mathbb{R}^d$ can be scaled by a positive factor $\varepsilon$ in the following way

$$K(x, y; \varepsilon) = K\left(\frac{x}{\varepsilon}, \frac{y}{\varepsilon}\right) \quad \forall x, y \in \mathbb{R}^d.$$ 

**Definition**

Let $K$ be a kernel on $\mathbb{R}^{d+1}$. If a scale function $c : \mathbb{R}^d \rightarrow (0, \infty)$ is given, we define a **Variably Scaled Kernel** or **VSK** on $\mathbb{R}^d$ by

$$K_c(x, y) = K\left((x, c(x)), (y, c(y))\right) \quad \forall x, y \in \mathbb{R}^d.$$
Popular RBFs: Multiquadric and Wendland

<table>
<thead>
<tr>
<th>RBF Type</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiquadric</td>
<td>$\sqrt{1 + r^2}$</td>
</tr>
<tr>
<td>$C^2$-Wendland</td>
<td>$(1 - r)^{\frac{4}{4}}(4r + 1)$</td>
</tr>
<tr>
<td>Scaled multiquadric</td>
<td>$\sqrt{\varepsilon^2 + r^2}$</td>
</tr>
<tr>
<td>Scaled $C^2$-Wendland</td>
<td>$(1 - \varepsilon r)^{\frac{4}{4}}(4\varepsilon r + 1)$</td>
</tr>
</tbody>
</table>

To scale the support of a Wendland function we use the basic function $\varphi_\varepsilon(x) = \varphi(\varepsilon x)$, where $\varphi$ is the one of the definition. Specifically the support radius of $\varphi_\varepsilon$ is $\frac{1}{\varepsilon}$. 

Left: $C^2$-W. Right: MQ with $\varepsilon = 0.1$. 

Left: $C^2$-W. Right: MQ with $\varepsilon = 0.1$. 

Daniela Schenone (UniMiB)
Edge detection with RBFs
February 21, 2018 5 / 28
Interpolation with RBFs

If \( \varphi \) is a positive definite radial function, the **interpolant**

\[
s_{f, \varphi}(x) = \sum_{i=1}^{N} \lambda_i \varphi(||x - x_i||)
\]

of a set of data \( \{x_i, f_i\}_{i=1}^{N} \), **exists** and is **unique**

The coefficients \( \lambda_i \) are obtained solving the linear system

\[
M\lambda = f,
\]

where \( M \) has elements \( M_{ij} = \varphi(||x_i - x_j||). \)
Study of coefficients: WORK IN PROGRESS

From 1 and 2 we have a formula for the cardinal expansion coefficients

\[ \lambda_k = \frac{1}{(2\pi)^{3/2}} \int_{0}^{2\pi} \frac{e^{ik\xi}}{\sum_{j=-\infty}^{\infty} \hat{\varphi}(|\xi + 2\pi j|)} d\xi, \]  

(1)

if \( \varphi \) is a positive definite radial function and the expression at the denominator is never null.

\[ \sum_{j=-\infty}^{\infty} \hat{\varphi}(|\xi + 2\pi j|) \]

\[ 1 \leq j \leq \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varepsilon \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ \int_{0}^{2\pi} \]

\[ 2\pi \]

\[ \hat{\varphi} \]

\[ \xi \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]

\[ 1 \]

\[ 2\pi \]

\[ j \]

\[ \left\lfloor \frac{1}{\varepsilon} \right\rfloor \]

\[ \varphi_{\varepsilon}(h) \]

\[ \cos(h \xi) \]

\[ \varphi_{\varepsilon}(0) \]
Study of coefficients: WORK IN PROGRESS

From \(^1\) and \(^2\) we have a formula for the cardinal expansion coefficients

\[
\lambda_k = \frac{1}{(2\pi)^{3/2}} \int_0^{2\pi} \frac{e^{ik\xi}}{\sum_{j=-\infty}^{\infty} \hat{\varphi}(|\xi + 2\pi j|)} d\xi, \tag{1}
\]

if \(\varphi\) is a positive definite radial function and the expression at the denominator is never null.

This is relevant for studying of the behaviour of interpolation coefficients near jump discontinuities, in fact we can express any kind of discrete data as a linear combination of cardinal functions.

---


Study of coefficients: WORK IN PROGRESS

From 1 and 2 we have a formula for the cardinal expansion coefficients

\[
\lambda_k = \frac{1}{(2\pi)^{3/2}} \int_0^{2\pi} \frac{e^{ik\xi}}{\sum_{j=-\infty}^{\infty} \hat{\varphi}(|\xi + 2\pi j|)} d\xi, \tag{1}
\]

if \( \varphi \) is a positive definite radial function and the expression at the denominator is never null.

This is relevant for studying of the behaviour of interpolation coefficients near jump discontinuities, in fact we can express any kind of discrete data as a linear combination of cardinal functions.

Especially, we would like to have a closed expression for the coefficient of the Wendland cardinal interpolant. For \( \varphi_\varepsilon \)

\[
\lambda_k = \frac{1}{2\pi} \int_0^{2\pi} \frac{\cos(k\xi)}{\sum_{h=1}^{\lfloor \frac{1}{\varepsilon} \rfloor} (2\varphi_\varepsilon(h)\cos(h\xi)) + \varphi_\varepsilon(0)} d\xi.
\]

---

An iterative adaptive edge detection method based on MQ RBF interpolation has been proposed in [Jung et al., 2011].

---

An iterative adaptive edge detection method based on MQ RBF interpolation has been proposed in [Jung et al., 2011].

This method is based on the observation that the absolute value of the MQ interpolation coefficients, with fixed shape parameter, is greater near jump discontinuities, while these local maxima do not appear if there is a null shape parameter near the discontinuities.

---

Literature: Iterative adaptive method (IA)

An iterative adaptive edge detection method based on MQ RBF interpolation has been proposed in [Jung et al., 2011]³.

This method is based on the observation that the absolute value of the MQ interpolation coefficients, with fixed shape parameter, is greater near jump discontinuities, while these local maxima do not appear if there is a null shape parameter near the discontinuities.

The idea of the algorithm is then to iteratively

An iterative adaptive edge detection method based on MQ RBF interpolation has been proposed in [Jung et al., 2011].

This method is based on the observation that the absolute value of the MQ interpolation coefficients, with fixed shape parameter, is greater near jump discontinuities, while these local maxima do not appear if there is a null shape parameter near the discontinuities.

The idea of the algorithm is then to iteratively

1. interpolate the data

---

Literature: Iterative adaptive method (IA)

An **iterative adaptive edge detection method** based on **MQ RBF interpolation** has been proposed in [Jung et al., 2011]³.

This method is based on the observation that the **absolute value** of the MQ interpolation **coefficients**, with fixed shape parameter, is greater near jump discontinuities, while these local maxima do not appear if there is a **null shape parameter** near the discontinuities.

The idea of the algorithm is then to iteratively

1. interpolate the data
2. find the coefficients greater than a certain tollerance level

---

Literature: Iterative adaptive method (IA)

An **iterative adaptive edge detection method** based on **MQ RBF interpolation** has been proposed in [Jung et al., 2011].

This method is based on the observation that the **absolute value** of the MQ interpolation **coefficients**, with fixed shape parameter, is greater near jump discontinuities, while these local maxima do not appear if there is a **null shape parameter** near the discontinuities.

The idea of the algorithm is then to iteratively:

1. interpolate the data
2. find the coefficients greater than a certain tolerance level
3. change to zero the correspondent shape parameter

---

Our method with variably scaled kernel (VSK1D)

The first method we propose for the jump detection in 1D is the iterative adaptive method with VSKs.

Instead of adapting the bases used for the interpolation with vanishing scale parameters, we use variably scaled kernels with step scale functions.
Our method with variably scaled kernel (VSK1D)

The first method we propose for the jump detection in 1D is the iterative adaptive method with VSKs.

Instead of adapting the bases used for the interpolation with vanishing scale parameters, we use variably scaled kernels with step scale functions.

The idea of our algorithm is to iteratively
1. interpolate the data
**Our method with variably scaled kernel (VSK1D)**

The first method we propose for the jump detection in 1D is the iterative adaptive method with VSKs.

Instead of adapting the bases used for the interpolation with vanishing scale parameters, we use **variably scaled kernels with step scale functions**.

The idea of our algorithm is to iteratively

1. interpolate the data
2. find the coefficients greater than a certain tolerance level in absolute value
Our method with variably scaled kernel (VSK1D)

The first method we propose for the jump detection in 1D is the iterative adaptive method with VSKs.

Instead of adapting the bases used for the interpolation with vanishing scale parameters, we use *variably scaled kernels with step scale functions*.

The idea of our algorithm is to iteratively

1. interpolate the data
2. find the coefficients greater than a certain tolerance level in absolute value
3. add a step function to the kernel in the points found at the previous step
Advantages of VSK1D method

- **Any radial kernel** can be used with our algorithm. $C^2$-Wendland RBF can conveniently replace MQ RBF as it is compactly supported and improves the stability of interpolation.

- The reconstruction for steep jumps is qualitatively better, as the VSKs reproduce the exact discontinuities instead of doing linear interpolation.

- For each pair (or triplet) of consecutive possible edge points identified in the same iteration, only their centroid is added to the edge, resulting in a thinner edge.
Advantages of VSK1D method

- **Any radial kernel** can be used with our algorithm. $C^2$-Wendland RBF can conveniently replace MQ RBF as it is compactly supported and improves the stability of interpolation.

- The reconstruction for **steep jumps** is **qualitatively better**, as the VSKs reproduce the exact discontinuities instead of doing linear interpolation.

![Graph](image)

MQ interpolant with vanishing parameters at the jump points (solid red). VSK interpolant with step scale function (dashed black). Exact function (solid green).
Advantages of VSK1D method

- **Any radial kernel** can be used with our algorithm. $C^2$-Wendland RBF can conveniently replace MQ RBF as it is compactly supported and improves the stability of interpolation.

- The reconstruction for steep jumps is qualitatively better, as the VSKs reproduce the exact discontinuities instead of doing linear interpolation.

- For each pair (or triplet) of consecutive possible edge points identified in the same iteration, only their centroid is added to the edge, resulting in a thicker edge.
Our Non-Iterative (N-I) method

The Non-Iterative algorithm is based on the fact that the local maxima of the absolute value interpolation coefficients identify the region where the edge is located.
Our Non-Iterative (N-I) method

The Non-Iterative algorithm is based on the fact that the local maxima of the absolute value interpolation coefficients identify the region where the edge is located.

The idea of our algorithm is to

1. interpolate the data
Our Non-Iterative (N-I) method

The Non-Iterative algorithm is based on the fact that the local maxima of the absolute value interpolation coefficients identify the region where the edge is located.

The idea of our algorithm is to

1. interpolate the data
2. compute the local maxima of the absolute value of coefficient expansion
Advantages of N-I method

- The N-I method is **faster** than the iterative methods, as \( \lambda \) is computed only once.
Advantages of N-I method

- The N-I method is **faster** than the iterative methods, as \( \lambda \) is computed only once.

- In 1D, the method provides **thinner edges** than IA method, as for each pair (or triplet) of consecutive possible edge points identified in the same iteration, only their centroid is added to the edge.
1D example

For the one dimensional numerical experimentation we use the following function

\[ f(x) = \begin{cases} 
(x + 2)^6 & -1 \leq x \leq -0.7 \\
(1 - x)^4 & -0.7 < x \leq -0.3 \\
(x + 2)^3 - 5 & -0.3 < x \leq 0 \\
\sin((7x - 2.1)^2) & 0 < x \leq 0.6 \\
-x & 0.6 < x \leq -0.8 \\
x^2 + 3 & 0.8 < x \leq 1 \\
0 & \text{elsewhere.}
\end{cases} \] (2)

The RBFs we use are the multiquadric with \( \varepsilon = 1 \) for the Jung method and \( C^2 \)-Wendland and \( \varepsilon = 1 \) for our methods.
(a) Original function and sampled random data. (b) Edge points obtained by IA1D. (c) Edge points obtained by VSK1D with $C^2$-W. (d) Absolute value of interpolation coefficients with $C^2$-W, local maxima in magenta. (e) Edge points obtained by N-I with $C^2$-W.
### Comparison

<table>
<thead>
<tr>
<th></th>
<th>IA1D</th>
<th>VSK1D with $C^2\text{-W}$</th>
<th>N-I1D with $C^2\text{-W}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iter.</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edge pts</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iter.</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edge pts</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edge pts</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Shepp-Logan phantom

Shepp-Logan phantom: zoom


We can compare the results with the SSIM, an index of structural similarity for binary images, where 1 means that the two images are identical. If we compare Canny and our method we obtain 0.9606. If we compare Canny to the IA method we obtain 0.9221.
(a) Original $512 \times 512$ pixel image. (b) Edges for IA2D dimension-by-dimension method. (c) Canny edge detection. (d) Absolute value of coefficients for dimension-by-dimension N-I method. (e) Edges for dimension-by-dimension N-I method: sum of rows and columns results.
(a) Original $512 \times 512$ pixel image. (b) Edges for IA2D dimension-by-dimension method. (c) Canny edge detection. (d) Absolute value of coefficients for dimension-by-dimension N-I method. (e) Edges for dimension-by-dimension N-I method: sum of rows and columns results.
(a) Original $512 \times 512$ pixel image. (b) Canny edge detection. (c) Absolute value of coefficients for dimension-by-dimension N-I method. (d) Edges for dimension-by-dimension N-I method with post-processing.
(a) Original $512 \times 512$ pixel image. (b) Edges for IA2D dimension-by-dimension method. (c) Canny edge detection. (d) Absolute value of coefficients for dimension-by-dimension N-I method. (e) Edges for dimension-by-dimension N-I method: sum of rows and columns results.
2D example: scattered data

\[ f(x, y) = \text{franke}(x, y) - \frac{1}{2}(y \leq -\frac{1}{5} \sin(5x) + \frac{1}{2}) \]

(a) Sampled data \((2^{10} = 1024)\) in blue. (b) Coefficients of the Wendland interpolation, edge points in magenta. (c) 2D view of the data, in red the points recognized as edges, the red line is the polynomial interpolation of the edge points. (d) Function with sampled data in blue and magenta, the red lines are the edge.
2D example: scattered data

\[ f(x, y) = (x + y - \frac{1}{2})^3(x \leq y^3) + (x + y)^{1/3}(x^3 \leq y) + \frac{1}{10} \]

In this example the coefficients are computed with a partition of the unity method for RBFs.

(a) Function with sampled data \((2^{12} = 4096)\) in blue. (b) Coefficients of the RBF interpolation, with the ones corresponding to the edge points in magenta. (c) Function with sampled data corresponding to the edge points in magenta.
Thank you :)
References


VSK1D algorithm

input: \( \{(x_i, f_i)\}_{i=1}^{N}, c(x) = 0, S = \emptyset, \eta, \tau > 0 \)

1. compute \( \lambda \) by solving the system \( M\lambda = f \)
2. find \( S' = \{x_i : |\lambda_i s'_f, x(x_i)|/\max_j(|\lambda_j s'_f, x(x_j)|) > \eta\} \), \( S = S \cup S' \)
3. \( c^{\text{new}}(x) = c(x) + \sum_{x_i \in S'} (f_{i+1} - f_i) 1_{(x_i, x_N]}(x) \)
4. compute \( \lambda^{\text{new}} \)
5. if \( ||\lambda - \lambda^{\text{new}}|| > \tau \) do repeat 2. -- 4.

output: \( S \) (edge points)
N-I algorithm

input: \{((x_i, f_i))\}_{i=1}^{N}, \eta \geq 1

1. if 2D do divide the image in square sub-domains, apply
   2. to each, 3. globally

2. compute \(\lambda\) and \(\max_{\text{loc}}(|\lambda|)\)

3. \(S = \{x_i \in \max_{\text{loc}}(|\lambda|) : |\lambda_i| > \eta \ \text{mean}(|\lambda|)\}\)

output: \(S\) (edge points)
Popular RBFs: Multiquadric and Wendland functions

**Definition**

The **multiquadric functions** or **MQ** are defined as

$$
\Phi_{d,\varepsilon} : \mathbb{R}^d \ni x \mapsto \varphi_{\varepsilon}(||x||) \in \mathbb{R}
$$

where \( \varphi : [0, \infty) \ni r \mapsto \varphi_{\varepsilon}(r) = \sqrt{\varepsilon^2 + r^2} \in \mathbb{R}, \) \( \varepsilon \) is the scale parameter.

**Definition**

The **\( C^2 \)-Wendland functions** are compactly supported radial basis functions defined as

$$
\Phi_s : \mathbb{R}^d \ni x \mapsto \varphi(||x||) \in \mathbb{R},
$$

where \( \varphi : [0, \infty) \ni r \mapsto \varphi(r) = (1 - r)^{\ell+1}[(\ell + 1)r + 1] \in \mathbb{R} \) such that \( \ell = \lfloor s/2 \rfloor + 3. \) The function \( \Phi_s \) has its support in \( S^d(0, 1). \)