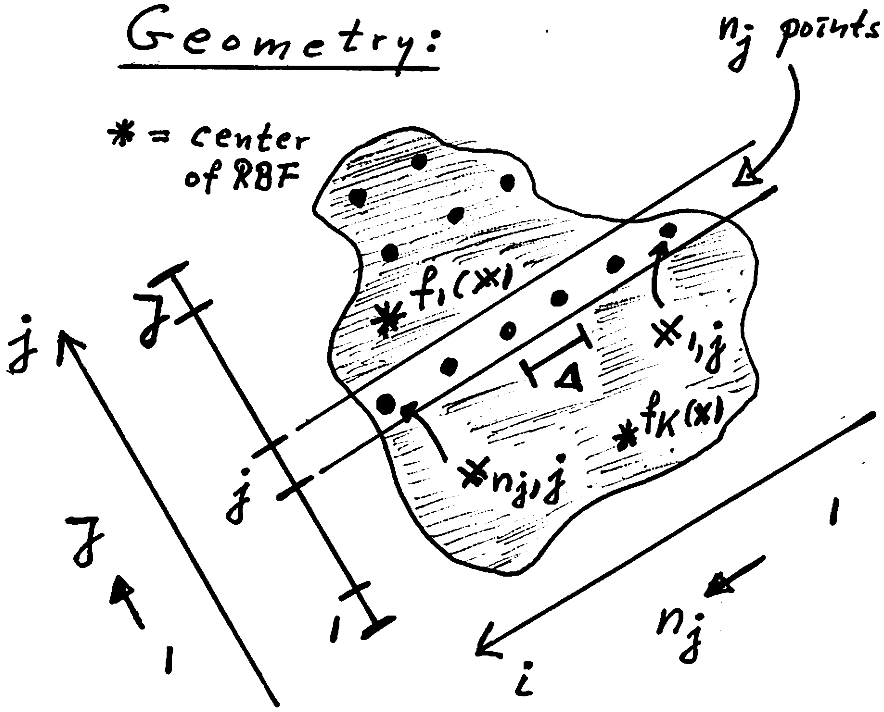
Bernd Hamann
Stratovan

1/29/2018

■ RBFs and Reconstruction – Cont'd.

Geometry:



- $RBFs = \{f_k(\mathbf{x})\}_{k=1}^{K}$
- "Reconstruction points" $= \{\mathbf{x}_{i,j}\}_{\substack{j=1,\dots J \\ i=1,\dots n_j}}$
- Notation:
  $\mathbf{x} = (x, y)$
  $\mathbf{x}_{i,j} = (x_{i,j}, y_{i,j})$
- RBFs: $f_1(\mathbf{x}), \dots, f_K(\mathbf{x})$
- Typical resolutions:
  i) 720 angles (of projection)
  ii) $J = 1024$
  iii) special case: $n_1 = \dots = n_J =: n$
- Reconstruction:

$$\boxed{\alpha(\mathbf{x}) = \sum_{k=1}^{K} c_k f_k(\mathbf{x})}$$

! ⇒ LINEAR SYSTEM:

$$\begin{array}{c} \uparrow \\ J \text{ rows} \\ \downarrow \end{array}
\begin{bmatrix} \sum_{i=1}^{n_1} f_1(\mathbf{x}_{i,1}) & \cdots & \sum_{i=1}^{n_1} f_K(\mathbf{x}_{i,1}) \\ \vdots & & \vdots \\ \sum_{i=1}^{n_J} f_1(\mathbf{x}_{i,J}) & \cdots & \sum_{i=1}^{n_J} f_K(\mathbf{x}_{i,J}) \end{bmatrix}
\begin{bmatrix} c_1 \\ \vdots \\ c_K \end{bmatrix} = \begin{bmatrix} \overline{C}_1 \\ \vdots \\ \overline{C}_J \end{bmatrix}$$

← K columns → $K \ll J$

BH

Bernd Hamann
Stratovan

1/29/2018

■ RBFs - Cont'd.

- LINEAR SYSTEM for ONE projection/imaging angle:

$$M c = \overline{C}$$ (see previous page)

- A LINEAR SYSTEM of the form $Mc = \overline{C}$ results FOR EVERY ANGLE. Thus:
  - For every angle, a matrix M and a right-hand side $\overline{C}$ results.
  - For every angle, the set of "reconstruction points" x_{ij} is different, just like the right-hand side $\overline{C}$.

! - The centers '*' of all RBFs are the SAME LOCATIONS in space, regardless of angle.

!! - "Concatenating/integrating" all matrices M and right-hand sides $\overline{C}$ defines the final OVERDETERMINED SYSTEM for the K unknowns $\{c_k\}$.
Number of rows of final system: NoAngles $\cdot J$
e.g. $720 \cdot 1024$

- Solve final system $Mc = \overline{C}$ using least-squares approach: Solve $M^T M c = M^T \overline{C}$, where $M^T M$ is a **K-by-K** matrix!

- Complexity/cost of evaluation of function $f_k(x)$ depends on its domain/support.
  → Consider and use $f_k(x)$ with small local domain.

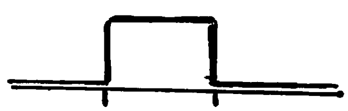
BH

- **RBFs - Cont'd.**

  - **Advantage:** RBF-based method <u>does not require a mesh</u> for reconstructions! Once the coefficients c_k in $\sum_k c_k f_k(\mathbf{x})$ are known, <u>any resolution and mesh can be used to discretize $\alpha(\mathbf{x})$</u>.

  - **Types of RBFs:** → Using <u>FINITE / INFINITE domains</u>
    → Lower or higher degrees of <u>CONTINUITY</u>

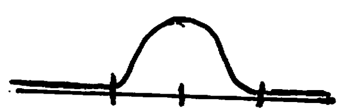
i) RBFs with <u>finite domain</u>:



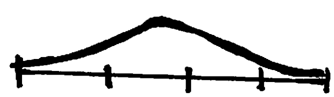
Constant

Linear

... <u>B-splines</u>

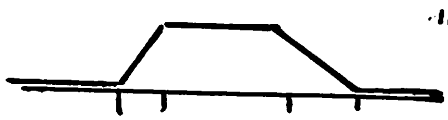


C¹-continuity
(Hermite)



C²-continuity
(cubic B-spline)

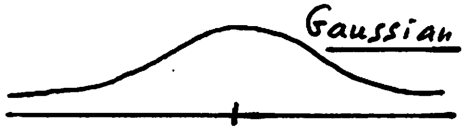
<u>Cubic splines</u>



"BLaC" — Blending of <u>L</u>inear & <u>c</u>onstant

ii) RBFs with <u>infinite domain</u>:

<u>Gaussian</u>



* center

, <u>Hardy's</u> multi-quadric/reciprocal multi-quadric <u>function</u>

$$f_k(x,y) = \left(R^2 + (x - x_k)^2 + (y - y_k)^2\right)^{+/- \frac{1}{2}}$$

BH