RBF Interpolation and Approximation of Large Span Data Sets

Vaclav Skala
Department of Computer Science and Engineering
Faculty of Applied Sciences
University of West Bohemia
306 14 Plzen, Czech Republic
http://www.VaclavSkala.eu

Abstract—**This contribution presents a new analysis of properties of the Radial Bases Functions (RBF) interpolation and approximation related to data sets with a large data span. The RBF is a convenient method for scattered d-dimensional interpolation and approximation, e.g. for solution of partial differential equations (PDE) etc. The RBF method leads to a solution of linear system of equations and computational complexity of solution is nearly independent of a dimensionality of a problem solved. However, the RBF methods are usually applied for small data sets with a small span of geometric coordinates.**

In this paper, we show influence of polynomial reproduction mostly used in RBF interpolation and approximation methods in the context of large span data sets. The experiments made proved expected theoretical results.

I. Introduction

INTERPOLATION and approximation techniques are used in solutions of many engineering problems. However, the interpolation and approximation of unorganized scattered data is still a severe problem. The standard approaches are based on tessellation of the domain in x, y or x, y, z spaces using, e.g. Delaunay triangulation etc. This approach is applicable for static data and t-varying data, if data in the time domain are "framed", i.e. given for specific time samples. However, it leads to increase of the dimensionality, i.e. from triangulation in E^2 to triangulation in E^3 or from triangulation in E^3 to triangulation in E^4 etc. It results into significant increase of triangulation complexity and complexity of a triangulation algorithm implementation. On the contrary, interpolations based on Radial Basis Functions (RBF) offer several significant advantages:

- RBF formulation leads to a solution of a linear system of equations, i.e. $\boldsymbol{Ax} = \boldsymbol{b}$
- RBF interpolation is applicable to d-dimensional problems and does not require tessellation of the definition domain
- RBF interpolation and approximation is especially convenient for scattered data interpolation, including interpolation of scattered data in time
- RBF interpolation and approximation are smooth by a definition
- RBF interpolation can be applied for interpolation of scalar fields and vector fields as well, which can be used for scalar and vector fields visualization
- if Compactly Supported RBFs (CSRBF) are used, sparse matrix data structures can be used as the matrix $\boldsymbol{A}$ is sparse, which decreases memory requirements significantly.

However, there are some weak points of the RBF application in real problems solution, e.g.:

- there is a real problem with robustness and reliability of the RBF computation due to low conditionality of the matrix $\boldsymbol{A}$ of the system of linear equations, especially if "global" RBFs are to be used
- numerical stability and representation if interpolation or approximation is to be applied over a large span of x, y, z values, i.e. if values are spanned over several magnitudes
- problems with memory management as the memory requirements are of $O(N^2)$ complexity, where N is a number of points in which values are given
- computational complexity of a solution of the LSE, which is $O(N^3)$, resp. $O(kN^2)$, where k is a number of iteration if iterative method is used, but k is relatively high, in general.
- problems with unexpected behavior at borders of geometrical objects

There are many contributions solving some issues of the RBF interpolation and approximation available. Numerical tests are mostly made using some standard testing functions and restricted domain span, mostly taking interval $< 0,1 >$ or similar. However, in many physically based applications, the span of a domain is high, usually over several magnitudes and large data sets need to be processed.

As the meshless techniques are easily scalable to higher dimensions and can handle spatial scattered data and scattered spatial-temporal data as well, they can be used in many engineering and economical computations, etc. Nowadays, polygonal representations (tessellated domains) are used in

This work was not supported by National Science Foundation GACR, project No.GA 17-05534S

DOI 10.1109/MCSI.2017.44 pp.212-218 ISBN 978-1-5386-2820-1/17

computer graphics and visualization as a surface representation and for surface rendering nearly exclusively. In time varying objects, a surface is represented as a triangular mesh with a constant connectivity.

On the other hand, all polygonal based techniques, in the case of scattered data, require tessellations, e.g. Delaunay triangulation with $O\left(N^{\lfloor d/2+1 \rfloor}\right)$ computational complexity (the worst case) for N points in d-dimensional space or another tessellation method. The complexity of tessellation algorithms implementation grows significantly with dimensionality and severe problems with robustness might be expected as well.

In the case of data visualization smooth interpolation or approximation on unstructured meshes is required, e.g. on triangular or tetrahedral meshes, when physical phenomena is associated with points, in general. This is quite a difficult task especially if smoothness of interpolation is needed. However, it is a natural requirement in physically based problems.

Interpolations methods used are usually separable, i.e. interpolation can be made along selected axis followed by another along the second axis etc. In the following meshless (meshfree) interpolations and approximation methods will be described, but they are not separable.

II. Meshless Interpolation

Meshless (meshfree) methods are based on the idea of Radial Basis Function (RBF) interpolation [2], [22], [23], [16], which is not separable. RBF based techniques are easily scalable to d-dimensional space and do not require tessellation of the geometric domain and offer smooth interpolation naturally. In general, meshless techniques lead to a solution of a linear system equations (LSE) [4], [5] with a full or sparse matrix.

Meshless methods for scattered data can be split into two main groups in computer graphics and visualization:

- "implicit" – $F(\boldsymbol{x}) = 0$, i.e. $F(x, y, z) = 0$ used in the case of a surface representation in E^3, e.g. surface reconstruction resulting into an implicit function representation. This problem is originated from the implicit function modeling [14] approach,
- "explicit" – $F(\boldsymbol{x}) = h$ used in interpolation or approximation resulting into a functional representation, e.g. a height map in E^2 – 2&1/2D,

where: $\boldsymbol{x}$ is a point representated generally in d-dimensional space, e.g. in the case of 2-dimensional case $\boldsymbol{x} = [x, y: 1]^T$ (expressed in the projective space notation) and h is a scalar value or a vector value associated with the point $\boldsymbol{x}$.

The RBF interpolation is based on a distance of two points computation in the d-dimensional space. It is defined as:

$$f(\boldsymbol{x}) = \sum_{j=1}^{M} \lambda_j \,\varphi\left(\left\|\boldsymbol{x} - \boldsymbol{x}_j\right\|\right) = \sum_{j=1}^{M} \lambda_j \,\varphi\left(r_j\right) \qquad (1)$$

and

$$\begin{aligned} r_j = \left\|\boldsymbol{x} - \boldsymbol{x}_j\right\|_2 &\stackrel{\text{def}}{=} \sqrt{\left(x - x_j\right)^2 + \left(y - y_j\right)^2} \\ &= \sqrt{\left(x - x_j\right)^2 + \left(y - y_j\right)^2 + (1-1)^2} \end{aligned} \qquad (2)$$

where: λ_j are weights to be computed, M is the number of points given.

It means that for the given data set $\{\langle \boldsymbol{x}_i, h_i \rangle\}_1^M$, where h_i are associated values to be interpolated and $\boldsymbol{x}_i$ are domain coordinates, we obtain a linear system of equations:

$$\begin{gathered} h_i = f(\boldsymbol{x}_i) = \sum_{j=1}^{M} \lambda_j \,\varphi\left(\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|\right) + P_k(\boldsymbol{x}_i) \\ i = 1, \ldots, M, \quad \boldsymbol{x} = [x, y: 1]^T \end{gathered} \qquad (3)$$

Due to some stability issues, usually a polynomial $P_k(\boldsymbol{x})$ of a degree k is added. For a practical use, the polynomial of the 1st degree is used, i.e. linear polynomial $P_1(\boldsymbol{x}) = \boldsymbol{a}^T\boldsymbol{x}$ in many applications. Therefore, the interpolation function has the form:

$$\begin{aligned} f(\boldsymbol{x}_i) &= \sum_{j=1}^{M} \lambda_j \,\varphi\left(\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|\right) + \boldsymbol{a}^T\boldsymbol{x}_i \\ &= \sum_{j=1}^{M} \lambda_j \,\varphi_{i,j} + \boldsymbol{a}^T\boldsymbol{x}_i \\ h_i &= f(\boldsymbol{x}_i) \qquad i = 1, \ldots, M \end{aligned} \qquad (4)$$

and additional conditions are to be applied:

$$\sum_{j=1}^{M} \lambda_i \boldsymbol{x}_i = \boldsymbol{0} \qquad (5)$$

i.e.

$$\sum_{j=1}^{M} \lambda_i x_i = 0 \qquad \sum_{j=1}^{M} \lambda_i y_i = 0 \qquad \sum_{j=1}^{M} \lambda_i = 0 \qquad (6)$$

Now, for d-dimensional case a system of $(M + d + 1)$ LSE has to be solved, where M is a number of points in the dataset and d is the dimensionality of data. For $d = 2$, vectors $\boldsymbol{x}_i = [x_i, y_i: 1]^T$ and $\boldsymbol{a} = \left[a_x, a_y: a_0\right]^T$, we can write :

$$\begin{bmatrix} \varphi_{1,1} & \ldots & \varphi_{1,M} & x_1 & y_1 & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \varphi_{M,1} & \ldots & \varphi_{M,M} & x_M & y_M & 1 \\ x_1 & \ldots & x_M & 0 & 0 & 0 \\ y_1 & \ldots & y_M & 0 & 0 & 0 \\ 1 & \ldots & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_M \\ a_x \\ a_y \\ a_0 \end{bmatrix} = \begin{bmatrix} h_1 \\ \vdots \\ h_M \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (7)$$

This can be rewritten in the matrix form as:

$$\begin{gathered} \begin{bmatrix} \boldsymbol{B} & \boldsymbol{P} \\ \boldsymbol{P}^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} \\ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \qquad \boldsymbol{a}^T\boldsymbol{x}_i = a_x\, x_i + a_y\, y_i + a_0 \end{gathered} \qquad (8)$$

For the two-dimensional case and M points given a system of $(M + 3)$ linear equations has to be solved. If "global" functions, e.g. $\varphi(r) = r^2 lg\, r$ or $\varphi(r) = e^{-(\epsilon r)^2}$, are used, then the matrix $\boldsymbol{B}$ is "full", while if "local" functions (Compactly Supported RBF – CSRBF) are used, the matrix $\boldsymbol{B}$ can be sparse. The RBF interpolation was originally introduced by the multiquadric method in 1971 [5], which was called Radial Basis Function (RBF) method. Since then many different RFB interpolation schemes have been developed with some specific properties, e.g. [4] uses $\varphi(r) = r^2 lg\, r$, which is called Thin-Plate Spline (TPS), a function $\varphi(r) = e^{-(\epsilon r)^2}$ was proposed in [22]. The CSRBFs were introduced as:

DOI 10.1109/MCSI.2017.44 pp.212-218 ISBN 978-1-5386-2820-1/17

$$\varphi(r) = \begin{cases} (1-r)^q \, P(r), & 0 \le r \le 1 \\ 0, & r > 1 \end{cases} \qquad (9)$$

where: $P(r)$ is a polynomial function and q is a parameter. Theoretical problems with numerical stability were solved in [4]. In the case of global functions, the linear system of equations is becoming ill conditioned and problems with convergence can be expected. On the other hand if the CSRBFs are taken, the matrix $\boldsymbol{A}$ is becoming relatively sparse, i.e. computation of the LSE will be faster, but we need to carefully select the shape factor α (which can be "tricky") and the final function might tend to be "blobby" shaped, see Tab.1. and Fig.1.

Table 1. Typical examples of "local" functions – CSRBF ("+" means – value zero out of $\langle 0,1 \rangle$)

ID	Function	ID	Function
1	$(1-r)_+$	6	$(1-r)_+^6$ $(35r^2+18r+3)$
2	$(1-r)_+^3(3r+1)$	7	$(1-r)_+^8$ $(32r^3+25r^2+8r+3)$
3	$(1-r)_+^5$ $(8r^2+5r+1)$	8	$(1-r)_+^3$
4	$(1-r)_+^2$	9	$(1-r)_+^3(5r+1)$
5	$(1-r)_+^4(4r+1)$	10	$(1-r)_+^7(16r^2+7r+1)$

All CSRBFs are defined for a "normalized" interval $r \in \langle 0\,,\,1\,\rangle$, but for a practical use a scaling is used, i.e. the value r is multiplied by a shape factor α, where $\alpha > 0$.

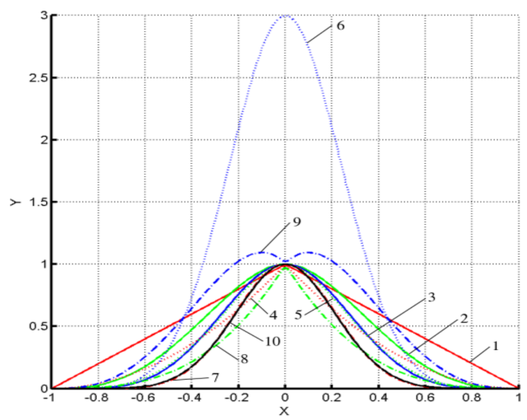

Figure 1. Geometrical properties of CSRBF

Meshless techniques are primarily based on approaches mentioned above. They are used in engineering problem solutions, nowadays, e.g. partial differential equations [6] surface modeling [8], surface reconstruction of scanned objects [3], [18] reconstruction of corrupted images [23], etc. More generally, meshless object representation is based on specific interpolation or approximation techniques [1][2][6][19][22].

The resulting matrix $\boldsymbol{A}$ tends to be large and ill-conditioned. Therefore, some specific numerical methods have to be taken to increase robustness of a solution, like preconditioning methods or parallel computing on GPU [11] etc. In addition, subdivision or hierarchical methods are used to decrease sizes of computations and increase robustness [14][20].

It should be noted, that *Computational complexity* of meshless methods actually covers complexity of tessellation itself and interpolation and approximation methods. This results into problems with large data set processing, i.e. numerical stability and memory requirements, etc.

If global RBF functions are considered, the RBF matrix is full and in the case of 10^6 of points, the RBF matrix is of the size approx.$10^6 \times 10^6$! On the other hand, if CSRBF used, the relevant matrix is sparse and computational and memory requirements can be decreased significantly using special data structures for sparse matrix representation.

On the other hand, in the case of physical phenomena visualization, data received by simulation, computation or obtained by experiments usually are oversampled in some areas and also numerically more or less precise. It seems possible to apply approximation methods and decrease computational complexity significantly by adding virtual points in the place of interest and use analogy of the least square method modified for the RBF case.

Due to CSRBF representation the space domain of data can be subdivided, interpolation, resp. approximation can be split to independent parts and computed more or less independently. This process can be also parallelized and if appropriate computational architecture is used, e.g. GPU etc. It will lead to faster computation as well. This approach was experimentally verified for scalar and vector data used in visualization of physical phenomena [12][17].

III. Meshless Approximation

The RBF interpolation relies on solution of a LSE $\boldsymbol{Ax} = \boldsymbol{b}$ of the size $M \times M$ in principle, where M is a number of the data to be processed. If "global" functions are used, the matrix $\boldsymbol{A}$ is full, while if "local" functions are used (CSRBF), the matrix $\boldsymbol{A}$ is sparse.

However, in visualization applications it is necessary to compute the final function $f(\boldsymbol{x})$ many times and even for already computed λ_i values, the computation of $f(\boldsymbol{x})$ is too expensive. Therefore it is reasonable to significantly "reduce" the size of the relevant LSE $\boldsymbol{Ax} = \boldsymbol{b}$. Of course, we are now changing the interpolation property of the RBF to RBF approximation, i.e. the values computed do not pass the given values exactly.

Simple approach

Probably the best way is to formulate the problem using the Least Square Method (LSM) approximation. Let us consider the modified formulation of the RBF interpolation, where M is a number of the given points [19].

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{P} \lambda_j \, \varphi(\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\|) \qquad (10)$$
$$h_i = f(\boldsymbol{x}_i) \qquad i = 1, \dots, M$$

DOI 10.1109/MCSI.2017.44 pp.212-218 ISBN 978-1-5386-2820-1/17

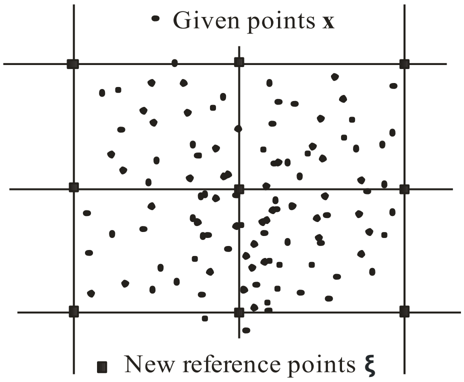

Figure 2. RBF approximation and points' reduction

where: $\boldsymbol{\xi}_j$ are not given points, but points in a pre-defined "virtual mesh" (in positions of area of interest etc.) as only coordinates are needed (there is no tessellation needed). This "virtual mesh" can be irregular, regular or adaptive etc. For a simplicity (*just for explanation purposes*), let us consider a two-dimensional squared (orthogonal) mesh, see Fig.2.

The $\boldsymbol{\xi}_j$ coordinates are the nodes of this "virtual" mesh. It means that the given scattered data will be actually "re-sampled", e.g. to the "virtual" mesh.

In many applications, the given data sets are heavily over sampled. For fast previews, we can afford to "down sample" the given data set, e.g. for data visualization, WEB applications, etc.

Let us consider that for the visualization purposes we want to represent the final scalar field by P values instead of M and $P \ll M$. The reason is very simple as if we need to compute the function $f(\boldsymbol{x})$ in many points, the formula above needs to be evaluated many times. We can expect that the number of evaluation Q can be easily requested at $10^2\ M$ of points (new points) used for visualization.

If we consider that $Q \geq 10^2\ M$ and $M \geq 10^2\ P$ then

the speed up factor in evaluation can be easily about $\mathbf{10^4}$!

just for *one* function value evaluation

The formulation above leads to a solution of an over determined system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$ where number of rows $M \gg P$ number of unknown $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_P]^T$. The linear system of equations $\boldsymbol{Ax} = \boldsymbol{b}$. It can be solved by the Least Square Method (LSM) as $\boldsymbol{A}^T\boldsymbol{Ax} = \boldsymbol{A}^T\boldsymbol{b}$.

$$\begin{bmatrix} \varphi_{1,1} & \cdots & \varphi_{1,P} \\ \vdots & \ddots & \vdots \\ \varphi_{i,1} & \cdot\cdot & \varphi_{i,P} \\ \vdots & \ddots & \vdots \\ \varphi_{M,1} & \cdots & \varphi_{M,P} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_P \end{bmatrix} = \begin{bmatrix} h_1 \\ \vdots \\ \vdots \\ \vdots \\ h_M \end{bmatrix} \qquad \boldsymbol{Ax} = \boldsymbol{b} \tag{11}$$

When the system of LSE is solved, computation of one function value $f(x)$ will be sped-up by a factor ν:

$$\nu = {}^{M}/_{P} \tag{12}$$

It should be noted, that the computation of $\boldsymbol{\lambda}$ will be sped-up by a factor $O((M/P)^3)$ as LSE computational is of $O(M^3)$.

RBF with Lagrange Multipliers

Let us consider more general approach based on extreme finding with constrains described in [6]. Let us assume again:

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{M} \lambda_j\ \varphi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) \qquad i = 1, \dots, N \quad \boldsymbol{A\lambda} = \boldsymbol{f} \tag{13}$$

where $M \leq N$. We want to determine $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_M]^T$ minimizing a quadratic form $\frac{1}{2}\boldsymbol{\lambda}^T\boldsymbol{Q\lambda}$ with a linear constrains $\boldsymbol{A\lambda} - \boldsymbol{f} = \boldsymbol{0}$, where $\boldsymbol{Q}$ is a positive symmetric matrix. This can be solved using Lagrange multipliers $\boldsymbol{\xi} = [\xi_1, \dots, \xi_N]^T$, i.e. minimizing the expression:

$$\frac{1}{2}\boldsymbol{\lambda}^T\boldsymbol{Q\lambda} - \boldsymbol{\xi}^T(\boldsymbol{A\lambda} - \boldsymbol{f}) \tag{14}$$

i.e. $\boldsymbol{\lambda}$ and $\boldsymbol{\xi}$ are unknowns.

As the matrix $\boldsymbol{Q}$ is positive and symmetric, we obtain

$$\begin{aligned} \frac{\partial}{\partial\boldsymbol{\lambda}}\left(\frac{1}{2}\boldsymbol{\lambda}^T\boldsymbol{Q\lambda} - \boldsymbol{\xi}^T(\boldsymbol{A\lambda} - \boldsymbol{f})\right) &= \boldsymbol{Q\lambda} - \boldsymbol{A}^T\boldsymbol{\xi} = \boldsymbol{0} \\ \frac{\partial}{\partial\boldsymbol{\xi}}\left(\frac{1}{2}\boldsymbol{\lambda}^T\boldsymbol{Q\lambda} - \boldsymbol{\xi}^T(\boldsymbol{A\lambda} - \boldsymbol{f})\right) &= \boldsymbol{A}^T\boldsymbol{\lambda} - \boldsymbol{f} = \boldsymbol{0} \end{aligned} \tag{15}$$

In more compact matrix form we can write:

$$\begin{bmatrix} \boldsymbol{Q} & -\boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\xi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{f} \end{bmatrix} \tag{16}$$

As the matrix $\boldsymbol{Q}$ is symmetric and positive definite, block in matrix operations can be applied and we get:

$$\begin{aligned} \boldsymbol{\lambda} &= \boldsymbol{Q}^{-1}\boldsymbol{A}^T(\boldsymbol{AQ}^{-1}\boldsymbol{A}^T)^{-1}\boldsymbol{f} \\ \boldsymbol{\xi} &= (\boldsymbol{AQ}^{-1}\boldsymbol{A}^T)^{-1}\boldsymbol{f} \end{aligned} \tag{17}$$

As $\boldsymbol{A} = \boldsymbol{A}^T$ and it is invertible, computation can be further simplified. This approach is more robust, however also more computationally expensive.

It should be noted, that if the Least Square Method (LSM) is used directly, i.e. $\boldsymbol{A}^T\boldsymbol{Ax} = \boldsymbol{A}^T\boldsymbol{b}$ is to be solved directly, the $\boldsymbol{A}^T\boldsymbol{A}$ matrix is ill conditioned and for large M the system of linear equations is difficult to solve. In addition, selection of the $\boldsymbol{Q}$ matrix elements is not fully determined and depends on a user, actually. The advantage of this approach is that values of the matrix $\boldsymbol{A}$ have only a linear influence. It should be noted that the matrix size is $2M \times 2M$, where M is a number of points. It means that the memory requirements are no acceptable even for medium data sets. Also the cost of the value computation, i.e. computation of a value $f(\boldsymbol{x})$ for the given $\boldsymbol{x}$ is doubled.

For real applications of the RBF approximation, we need to decrease memory requirements significantly.

DOI 10.1109/MCSI.2017.44 pp.212-218 ISBN 978-1-5386-2820-1/17

Least Square Method with a Polynomial Reproduction

Let us consider again the overdetermined system:

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{M} \lambda_j \, \varphi(\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\|) + \boldsymbol{a}^T \boldsymbol{x}_i = \sum_{j=1}^{M} \lambda_j \, \varphi_{i,j} + \boldsymbol{a}^T \boldsymbol{x}_i \tag{18}$$

It can be rewritten in the matrix form as

$$\boldsymbol{A\lambda} + \boldsymbol{Pa} = \boldsymbol{f} \tag{19}$$

Now, we can define an error r of a solution as

$$r^2 = \|\boldsymbol{A\lambda} + \boldsymbol{Pa} - \boldsymbol{f}\|^2 = (\boldsymbol{A\lambda} + \boldsymbol{Pa} - \boldsymbol{f})^T(\boldsymbol{A\lambda} + \boldsymbol{Pa} - \boldsymbol{f}) \tag{20}$$

where:

$$\boldsymbol{Pa} = \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_m & y_m & 1 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_0 \end{bmatrix} \tag{21}$$

To minimize the error r the following conditions must be valid:

$$\frac{\partial r^2}{\partial \boldsymbol{\lambda}} = \boldsymbol{A}^T\boldsymbol{A\lambda} + \boldsymbol{A}^T\boldsymbol{Pa} - \boldsymbol{A}^T\boldsymbol{f} = \boldsymbol{0} \qquad \frac{\partial r^2}{\partial \boldsymbol{a}} = \boldsymbol{P}^T\boldsymbol{A\lambda} + \boldsymbol{P}^T\boldsymbol{Pa} - \boldsymbol{P}^T\boldsymbol{f} = \boldsymbol{0} \tag{22}$$

or in a matrix form as $\boldsymbol{Mx} = \boldsymbol{y}$, i.e.

$$\begin{bmatrix} \boldsymbol{A}^T\boldsymbol{A} & \boldsymbol{A}^T\boldsymbol{P} \\ \boldsymbol{P}^T\boldsymbol{A} & \boldsymbol{P}^T\boldsymbol{P} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{f} \\ \boldsymbol{P}^T\boldsymbol{f} \end{bmatrix} \tag{23}$$

The above presented formula leads to correct results [9]. However, it can be seen, that the values m_{ij} of the matrix $\boldsymbol{M}$ are influenced by:

- elements of the matrix $\boldsymbol{A}^T\boldsymbol{A}$, i.e. by used radial basis function and mutual positions of the given points
- elements of the matrix $\boldsymbol{P}^T\boldsymbol{P}$, i.e. by coordinates of the given points.

It is a significant problem if data sets with a large span are to be processed and the interval of $\boldsymbol{x}$ values, i.e. x, y, is high, as the values are squared due to $\boldsymbol{P}^T\boldsymbol{P}$ submatrix etc.

Let us analyze this property more in detail, now, in order to be able to estimate problems in real application use.

IV. Decomposition of RBF Interpolation

The RBF interpolation can be described in the matrix form as:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{P} \\ \boldsymbol{P}^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} \qquad \boldsymbol{a}^T \boldsymbol{x}_i = a_x \, x_i + a_y \, y_i + a_0 \tag{24}$$

where: $\boldsymbol{x} = [x, y{:}\, 1]^T$, the matrix $\boldsymbol{A}$ is symmetrical and semidefinite positive (or strictly positive) definite. Let us consider the Schur's complement (validity of all operation is expected):

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{CA}^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M/A} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}^{-1}\boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \qquad \boldsymbol{M/A} \stackrel{\text{def}}{=} \boldsymbol{D} - \boldsymbol{CA}^{-1}\boldsymbol{B} \tag{25}$$

where: $\boldsymbol{M/A}$ is the Schur's complement. Then the inversion matrix $\boldsymbol{M}^{-1}$ is defined as:

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{A}^{-1}\boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{M/A})^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{CA}^{-1} & \boldsymbol{I} \end{bmatrix} \tag{26}$$

Now, the Schur's complement can be applied to the RBF interpolation. As the matrix $\boldsymbol{M}$ is nonsingular, inversion of the matrix $\boldsymbol{M}$ can be used. Using the Schur's complement (as the matrix $\boldsymbol{D} = \boldsymbol{0}$) we get:

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{A}^{-1}\boldsymbol{P} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{M/A})^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{P}^T\boldsymbol{A}^{-1} & \boldsymbol{I} \end{bmatrix} \qquad \boldsymbol{M/A} \stackrel{\text{def}}{=} \boldsymbol{P}^T\boldsymbol{A}^{-1}\boldsymbol{P} \tag{27}$$

Then $\det(\boldsymbol{M}) \neq 0$, $\det(\boldsymbol{M/A}) \neq 0$ and $\det(\boldsymbol{M/A}) \neq 0$ as the matrices are nonsingular.

However, if RBF interpolation is used for larger data sets, there is a severe problem with robustness and numerical stability, i.e. numerical computability. Using the Schur's complement we can see, that:

$$\det(\boldsymbol{M}) = \det(\boldsymbol{A}) \, \det(\boldsymbol{M/A}) \tag{28}$$

and therefore

$$\det(\boldsymbol{M}^{-1}) = \frac{1}{\det(\boldsymbol{A})} \frac{1}{\det(\boldsymbol{M/A})} = \frac{1}{\det(\boldsymbol{A})} \frac{1}{\det(\boldsymbol{P}^T\boldsymbol{A}^{-1}\boldsymbol{P})} \tag{29}$$

Properties of the matrix $\boldsymbol{A}$ are determined by the RFB function used. The value of $\det(\boldsymbol{A})$ depends also on the mutual distribution of points. However, the influence of $\det(\boldsymbol{P}^T\boldsymbol{A}^{-1}\boldsymbol{P})$ is also significant as the value depends on the points mutual distribution due to the matrix $\boldsymbol{A}$ but also to points distribution in space, due to the matrix $\boldsymbol{P}$. It means that translation of points in space does have significant influence as well. Let us imagine for a simplicity that the matrix $\boldsymbol{A} = \boldsymbol{I}$ (it can happen if CSRBF is used and only one point is within the radius $r = 1$). Then the distance of a point from the origin has actually quadratic influence as the point position is in the matrices $\boldsymbol{P}^T$ and $\boldsymbol{P}$. There is a direct significant consequence for the RBF interpolation.

Let us conseder the RBF interpolation again:

$$f(\boldsymbol{x}) = \sum_{j=1}^{M} \lambda_j \, \varphi(\|x - \boldsymbol{x}_j\|) + P_k(\boldsymbol{x}) \tag{30}$$

where: the $P_k(\boldsymbol{x})$, $k = 1, 2$ is a quadratic polynomial:

$$P_1(\boldsymbol{x}) = a_0 + a_1 x + a_2 y \tag{31}$$

resp.

$$P_2(\boldsymbol{x}) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 \tag{32}$$

In the case of $\boldsymbol{A} = \boldsymbol{I}$, we get a matrix $\boldsymbol{P}^T\boldsymbol{P}$ of the size (3×3) and $\det(\boldsymbol{P}^T\boldsymbol{P})$ in the case of a linear polynomial $P_1(\boldsymbol{x})$ as:

$$\det(\boldsymbol{P}^T\boldsymbol{P}) = \begin{vmatrix} \sum_{i=1}^{M} x_i^2 & \sum_{i=1}^{M} x_i y_i & \sum_{i=1}^{M} x_i \\ \sum_{i=1}^{M} x_i y_i & \sum_{i=1}^{M} y_i^2 & \sum_{i=1}^{M} y_i \\ \sum_{i=1}^{M} x_i & \sum_{i=1}^{M} y_i & \sum_{i=1}^{M} 1 \end{vmatrix} \tag{33}$$

$$= n\left(\sum x_i^2 \sum y_i^2\right) - \sum y_i\,(\dots) + \sum y_i\,(\dots)$$

It means that points distribution in space and their distances from the origin play a significant role as the $\det(\boldsymbol{P}^T\boldsymbol{P})$ contains elements $\sum_{i=1}^{M} x_i^2$ and $\sum_{i=1}^{M} y_i^2$ in multiplicative etc. in the linear polynomial case.

If a quadratic polynomial $P_2(\boldsymbol{x})$ is used, the matrix $\boldsymbol{P}^T\boldsymbol{P}$ is of the size (6×6):

$$\boldsymbol{P}^T\boldsymbol{P} = \begin{bmatrix} x_1^2 & \cdots & x_M^2 \\ y_1^2 & \cdots & y_1^2 \\ x_1 y_1 & \cdots & x_M y_M \\ x_1 & \cdots & x_M \\ y_1 & \cdots & y_M \\ 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_1^2 & y_1^2 & x_1 y_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_M^2 & y_M^2 & x_M y_M & x_M & y_M & 1 \end{bmatrix} \tag{34}$$

and then:

$$\det(\boldsymbol{P}^T\boldsymbol{P}) = \det \begin{bmatrix} \sum_{i=1}^{M} x_i^4 & \sum_{i=1}^{M} x_i^2 y_i^2 & \cdots & \sum_{i=1}^{M} x_i^2 \\ \sum_{i=1}^{M} x_i^2 y_i^2 & \ddots & \cdots & \sum_{i=1}^{M} y_i^2 \\ \vdots & \vdots & \ddots & \\ \sum_{i=1}^{M} x_i^2 & \sum_{i=1}^{M} y_i^2 & \cdots & \sum_{i=1}^{M} 1 \end{bmatrix} \tag{35}$$

In the quadratic polynomial case, the $\det(\boldsymbol{P}^T\boldsymbol{P})$ contains elements $\sum_{i=1}^{M} x_i^4, \sum_{i=1}^{M} y_i^2, \dots, \sum_{i=1}^{M} 1$ in multiplicative, which brings even numerically worst situation as the matrix $\boldsymbol{P}^T\boldsymbol{P}$ contains small and very high values. As a direct consequence, eigenvalues will have large span and therefore the linear system of equations will become ill-conditioned.

V. Decomposition of RBF Approximation

Decomposition for RBF approximation is analogous to the interpolation decomposition. Let us explore decomposition of the RBF approximation using the Schur's complement. Let us consider the system of linear equation for the RBF approximation in the form $\boldsymbol{Mx} = \boldsymbol{y}$:

$$\begin{bmatrix} \boldsymbol{A}^T\boldsymbol{A} & \boldsymbol{A}^T\boldsymbol{P} \\ \boldsymbol{P}^T\boldsymbol{A} & \boldsymbol{P}^T\boldsymbol{P} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{f} \\ \boldsymbol{P}^T\boldsymbol{f} \end{bmatrix} \tag{36}$$

Let us consider again the Schur's complement (validity of operations is expected and the matrix $\boldsymbol{D} \neq \boldsymbol{0}$)

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \dfrac{\boldsymbol{M}}{\boldsymbol{A}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}^{-1}\boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{37}$$

$$\boldsymbol{M}/\boldsymbol{A} \stackrel{\text{def}}{=} \boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}$$

In this case, for the RBF approximation we obtain:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{A} & \boldsymbol{A}^T\boldsymbol{P} \\ \boldsymbol{P}^T\boldsymbol{A} & \boldsymbol{P}^T\boldsymbol{P} \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ (\boldsymbol{P}^T\boldsymbol{A})(\boldsymbol{A}^T\boldsymbol{A})^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \dfrac{\boldsymbol{M}}{(\boldsymbol{A}^T\boldsymbol{A})} \end{bmatrix} \tag{38}$$

$$\begin{bmatrix} \boldsymbol{I} & (\boldsymbol{A}^T\boldsymbol{A})^{-1}(\boldsymbol{A}^T\boldsymbol{P}) \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$$

Then the matrix $\boldsymbol{M}^{-1}$ using the Schur's complement:

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{I} & -(\boldsymbol{A}^T\boldsymbol{A})^{-1}(\boldsymbol{A}^T\boldsymbol{P}) \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$$

$$\begin{bmatrix} (\boldsymbol{A}^T\boldsymbol{A})^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \left(\dfrac{\boldsymbol{M}}{(\boldsymbol{A}^T\boldsymbol{A})}\right)^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -(\boldsymbol{P}^T\boldsymbol{A}^{-1})(\boldsymbol{A}^T\boldsymbol{A})^{-1} & \boldsymbol{I} \end{bmatrix} \tag{39}$$

where:

$$\boldsymbol{M}/(\boldsymbol{A}^T\boldsymbol{A}) = \boldsymbol{P}^T\boldsymbol{P} - (\boldsymbol{P}^T\boldsymbol{A})(\boldsymbol{A}^T\boldsymbol{A})^{-1}(\boldsymbol{A}^T\boldsymbol{P}) \tag{40}$$

In the RBF approximation case, the matrix $\boldsymbol{D}$ is non-zero matrix $\boldsymbol{P}^T\boldsymbol{P}$. It can be seen, that if the matrix $\boldsymbol{A}^T\boldsymbol{A} \to \boldsymbol{I}$, then the Schur's complement

$$\boldsymbol{M}/(\boldsymbol{A}^T\boldsymbol{A}) \to \left(\boldsymbol{P}^T\boldsymbol{P} - (\boldsymbol{P}^T\boldsymbol{A})(\boldsymbol{A}^T\boldsymbol{P})\right)$$
$$= \boldsymbol{P}^T\boldsymbol{P} - (\boldsymbol{A}^T\boldsymbol{P})^T(\boldsymbol{A}^T\boldsymbol{P}).$$

It means, that the whole matrix $\boldsymbol{M}$ tends to be singular. It can be seen that the $\det(\boldsymbol{P}^T\boldsymbol{P})$ contains elements $\sum_{i=1}^{M} x_i^4$, $\sum_{i=1}^{M} y_i^2, \dots, \sum_{i=1}^{M} 1$ in multiplicative, too. This has a significant influence to the robustness of computation if small and high values of x_i and y_i occur in the data sets, or if they are from an interval with a large span.

If the values $(x_i, y_i) \in \langle -10^5, 10^5 \rangle \times \langle -10^5, 10^5 \rangle$, the value of $\det(\boldsymbol{P}^T\boldsymbol{P}) > \sum_{i=1}^{M} x_i^2 \sum_{i=1}^{M} y_i^2 > 10^{20}$ and the value of $1/\det(\boldsymbol{P}^T\boldsymbol{P}) < 10^{-20}$, in the case of the linear polynomial reproduction. It results into a situation when the matrix $\boldsymbol{M}^{-1}$ will be very "close" to singular.

In the case of quadratic polynomial $P_2(\boldsymbol{x})$, the situation gets even worst as $\det(\boldsymbol{P}^T\boldsymbol{P})$ contains elements $\sum x_i^4$ and $\sum y_i^4$ in multiplicative, i.e. $\det(\boldsymbol{P}^T\boldsymbol{P}) > \sum_{i=1}^{M} x_i^4 \sum_{i=1}^{M} y_i^4 > 10^{40}$.

This should be considered as a significant disadvantage of the RBF approximation used for large data spans.

However, it is recommended to use $P_0(\boldsymbol{x})$ as it "moves vertically" the data and improves the stability. Also, if $P_k(\boldsymbol{x})$ represents" a global data behavior, its usage can be recommended, as the functions $\varphi_{i,j}(r)$ actually make "additional off-set modulation".

VI. Experimental Results

The influence of polynomial reproduction was studied on synthetic and real data sets. The influence of a nonlinear polynomial was significant even for data with a smaller data span. As real data sets were used. Fig.3 presents a difference between original data and RBF approximation.

The experiments also proved that vertical "shift" of center of gravity of data increases the numerical stability as expected.

DOI 10.1109/MCSI.2017.44 pp.212-218 ISBN 978-1-5386-2820-1/17

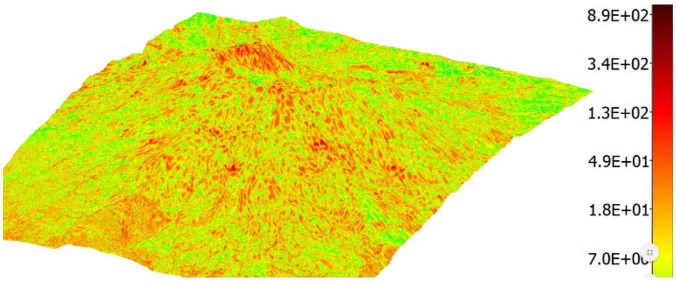

Figure 3. Difference of original data and RBF Least Square Error approximation of the St.Helens volcano 10 000 points used instead of 6 743 176 original points

VII. Conclusion

The RBF interpolation using compactly supported RBF (CSRBF) have several significant advantages over methods based on smooth interpolation made on triangulated space area. In this contribution some properties of the CSRBF interpolation and approximation methods have been presented from the "engineering" point of view and selected features related to robustness and stability of computation have been presented.

The presented founding are fundamental especially in the case of engineering and GIS related applications.

Deeper CSRBF analysis of sparse data structures used, space subdivision and speed up of computation will be explored in future.

Acknowledgment

The author would like to thank to colleagues and students at the University of West Bohemia, Plzen, for their comments and suggestions, comments and hints provided, to anonymous reviewers for their critical view and recommendations that helped to improve the manuscript.

References

[1] Adams,B., Ovsjanikov,M., Wand,M., Seidel,H.-P., Guibas,L.J.: Meshless modeling of Deformable Shapes and their Motion. ACM SIGGRAPH Symp. on Computer Animation, 2008

[2] Buhmann,M.D.: Radial Basis Functions: Theory and Implementations. Cambridge Univ.Press, 2008.

[3] Carr,J.C., Beatsom,R.K., Cherrie,J.B., Mitchell,T.J., Fright,W.R., Ffright,B.C. McCallum,B.C, Evans,T.R.: Reconstruction and representation of 3d objects with radial basis functions. Proc. SIGGRAPH'01, pp. 67–76, 2001.

[4] Duchon,J.: Splines minimizing rotation-invariant semi-norms in Sobolev space. Constructive Theory of Functions of Several Variables, LNCS 571, Springer, 1997.

[5] Hardy,L.R.: Multiquadric equation of topography and other irregular surfaces. Journal of Geophysical Research 76 (8), 1905-1915, 1971.

[6] Fasshauer,G.E.: Meshfree Approximation Methods with MATLAB. World Scientific Publishing, 2007.

[7] Lazzaro,D., Montefusco,L.B.: Radial Basis functions for multivariate interpolation of large data sets. Journal of Computational and Applied Mathematics, 140, pp. 521-536, 2002.

[8] Macedo,I., Gois,J.P., Velho,L.: Hermite Interpolation of Implicit Surfaces with Radial Basis Functions. Computer Graphics Forum, Vol.30, No.1, pp.27-42, 2011.

[9] Majdisova,Z., Skala,V.: A New Radial Basis Function Approximation with Reproduction, CGVCVIP 2016, pp.215-222, ISBN 978-989-8533-52-4, Portugal, 2016

[10] Majdisova,Z., Skala,V.: A Radial Basis Function Approximation for Large Datasets, SIGRAD 2016, pp.9-14, Sweden, 2016

[11] Nakata,S., Takeda,Y., Fujita,N., Ikuno,S.: Parallel Algorithm for Meshfree Radial Point Interpolation Method on Graphics Hardware. IEEE Trans.on Magnetics, Vol.47, No.5, pp.1206-1209, 2011.

[12] Pan,R., Skala,V.: A two level approach to implicit modeling with compactly supported radial basis functions. Engineering and Computers, Vol.27. No.3, pp.299-307, ISSN 0177-0667, Springer, 2011.

[13] Pan,R., Skala,V.: Surface Reconstruction with higher-order smoothness. The Visual Computer, Vol.28, No.2., pp.155-162, ISSN 0178-2789, Springer, 2012.

[14] Ohtake,Y., Belyaev,A., Seidel,H.-P.: A multi-scale approach to 3d scattered data interpolation with compactly supported basis functions. In: Proceedings of international conference shape modeling, IEEE Computer Society, Washington, pp 153–161, 2003.

[15] Savchenko,V., Pasco,A., Kunev,O., Kunii,T.L.: Function representation of solids reconstructed from scattered surface points & contours. Computer Graphics Forum, 14(4), pp.181–188, 1995.

[16] Skala,V: Progressive RBF Interpolation, 7th Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa, Afrigraph 2010, pp.17-20, ACM, ISBN:978-1-4503-0118-3, 2010

[17] Skala,V., Pan,R.J., Nedved,O.: Simple 3D Surface Reconstruction Using Flatbed Scanner and 3D Print. ACM SIGGRAPH Asia 2013 poster, 2013.

[18] Skala,V.: Projective Geometry, Duality and Precision of Computation in Computer Graphics, Visualization and Games. Tutorial Eurographics 2013, Girona, 2013

[19] Skala,V.: Meshless Interpolations for Computer Graphics, Visualization and Games. Tutorial Eurographics 2015, Zurich, 2015.

[20] Sussmuth,J., Meyer,Q., Greiner,G.: Surface Reconstruction Based on Hierarchical Floating Radial Basis Functions. Computer Graph. Forum, 29(6): 1854-1864, 2010.

[21] Wenland,H.: Scattered Data Approximation. Cambridge University Press, 2010.

[22] Wright,G.B.: Radial Basis Function Interpolation: Numerical and Analytical Developments. University of Colorado, Boulder. PhD Thesis,, 2003

[23] Zapletal,J., Vanecek,P., Skala,V.: RBF-based Image Restoration Utilizing Auxiliary Points. CGI 2009 proceedings, pp.39-44, 2009