REVISTA INVESTIGACIÓN OPERACIONAL VOL. 37 , NO 2,xx/xx, 2016

A PRACTICAL USE OF RADIAL BASIS FUNCTIONS INTERPOLATION AND APPROXIMATION

Vaclav Skala[1]
Department of Computer Science and Engineering
Faculty of Applied Sciences
University of West Bohemia, Univerzitni 8
CZ30614 Plzen, Czech Republic

ABSTRACT

Interpolation and approximation methods are used across many fields. Standard interpolation and approximation methods rely on "ordering" that actually means tessellation in d-dimensional space in general, like sorting, triangulation, generating of tetrahedral meshes etc. Tessellation algorithms are quite complex in d-dimensional case. On the other hand, interpolation and approximation can be made using meshfree (meshless) techniques using Radial Basis Function (RBF). The RBF interpolation and approximation methods lead generally to a solution of linear system of equations. However, a similar approach can be taken for a reconstruction of a surface of scanned objects, etc. In this case this leads to a linear system of homogeneous equations, when a different approach has to be taken.

In this paper we describe novel approaches based on RBFs for data interpolation and approximation generally in d-dimensional space. We will show properties and differences of "global" and "Compactly Supported RBF (CSRBF)", run-time and memory complexities. As the RBF interpolation and approximation naturally offer smoothness, we will analyze such properties as well as approaches how to decrease computational expenses. The proposed meshless interpolation and approximation will be demonstrated on different problems, e.g. inpainting removal, restoration of corrupted images with high percentage of corrupted pixels, digital terrain interpolation and approximation for GIS applications and methods for decreasing computational complexity.

KEYWORDS: Radial basis function, RBF, interpolation, approximation, numerical methods

MSC: 68W25, 65D05

1. INTRODUCTION

Objects in computer graphics are usually defined as a surface model using a surface description, e.g. polygonal meshes, parametric patches or as a volumetric model using computer solid geometry, etc. Available hardware is optimized for triangular meshes processing. Recently a surface of time varying objects was represented by a triangular mesh with a constant connectivity. It enables to make effective data representation, compression, transmission, decompression and rendering of such models. In the discrete case, volumetric models are mostly considered, like CT and MRI images, standard techniques like marching cubes or tetrahedra are used and data are represented in regular structured meshes.

This paper describes representations, manipulation, compression and reduction of meshless (meshfree) representation. As the meshless techniques are easily scalable to higher dimensions and can handle spatial scattered data and spatial-temporal data as well, they can be used in many engineering and economical computations, etc. Polygonal representations (tessellated domains) are used in computer graphics and visualization as a surface representation and for surface rendering. In time varying objects a surface is represented as a triangular mesh with constant connectivity. This approach led to new algorithms for simplification and compression of dynamic meshes with constant connectivity [22] - [24]. The compression is actually based on the algorithm for surface extraction of implicitly defined objects [5]- [7]. The presumption that a surface is given as a polygonal mesh (actually as a triangular mesh) with constant connectivity has lead to quite effective algorithms for dynamic mesh compression [25].

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 implementation grows significantly with dimensionality and problems with robustness might be expected as well.

[1] http://www.VaclavSkala.eu

Presented at IWOR workshop
March 2015, Habana,Cuba

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. That 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) interpolation and approximation methods will be described, but they are not separable.

2. MESHLESS INTERPOLATION

Meshless (meshfree) methods are based on the idea of Radial Basis Function (RBF) interpolation [2], [27], [28], which is not separable, but they are easily extensible for d-dimensional case. RBF based techniques are easily scalable to d-dimensional space and do not require tessellation of the geometric domain and offers smooth interpolation naturally. In general, meshless techniques lead to a solution of a linear system equations (LSE) [7], [8] with a full or sparse matrix.
Generally, meshless methods for scattered data can be split into two main groups in computer graphics and visualization:

- "implicit" – surface reconstruction results to an implicit function representation, i.e. $F(\boldsymbol{x}) = 0$, e.g. $F(x, y, z) = 0$ in the case of a surface representation in E^3 – this problem is actually originated from the implicit function modeling [16] approach
- "explicit" – interpolation or approximation results to a functional representation, i.e. $F(\boldsymbol{x}) = h$, e.g. a height map in E^2 – 2 1/2D, i.e. . $h = F(x, y)$. However, there is a severe problem – an iso-curve, resp. iso-surface extraction

where: $\boldsymbol{x}$ is a point representated generally in d-dimensional space and h is a scalar value or a vector value.
The RBF interpolation is based on computing of the distance of two points in the d-dimensional space and it is defined by a function:

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

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:

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

where: λ_j are weights to be computed. Due to some stability issues, usually a polynomial $P_k(\boldsymbol{x})$ of a degree k is added to the formula:

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

For a practical use, the polynomial of the 1st degree is used, i.e. linear polynomial $P_1(\boldsymbol{x}) = \boldsymbol{a}^T \boldsymbol{x} + a_0$, in many applications. So the interpolation function has the form:

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{M} \lambda_j \, \varphi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) + \boldsymbol{a}^T \boldsymbol{x}_i + a_0 = \sum_{j=1}^{M} \lambda_j \, \varphi_{i,j} + \boldsymbol{a}^T \boldsymbol{x}_i + a_0$$

$$h_i = f(\boldsymbol{x}_i) \qquad i = 1, \dots, M$$

and additional conditions are applied:

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

It can be seen that 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 $\boldsymbol{x}_i$ and $\boldsymbol{a}$ are in the form $\boldsymbol{x}_i = [x_i, y_i]^T$ and $\boldsymbol{a} = [a_x, a_y]^T$, we can write :

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

Presented at IWOR workshop
March 2015, Habana,Cuba

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{P} \\ \boldsymbol{P}^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} \qquad \boldsymbol{Ax} = \boldsymbol{b} \qquad \boldsymbol{a}^T \boldsymbol{x_i} + a_0 = a_x\, x_i + a_y\, y_i + a_0$$

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. TPS ($\varphi(r) = r^2 lg\, r$), are used the matrix $\boldsymbol{B}$ is "full", if "local" functions (Compactly supported RBF – CSRBF) are used, the matrix $\boldsymbol{B}$ can be sparse.
The radial basis functions interpolation was originally introduced in [8] by introduction of multiquadric method in 1971, which was called Radial Basis Function (RBF) method. Since then many different RFB interpolation schemes have been developed with some specific properties, e.g. [7] 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 [27] and Compactly Supported RBFs (CSRBF) were introduced as:

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

where: $P(r)$ is a polynomial function and q is a parameter. Theoretical problems with stability and solvability were solved in [7]. Generally, there are two main groups of the RBFs:

- "global" – a typical example is TPS function
- "local" – Compactly supported RBF (CSRBF)

If the "global" functions are taken, the matrix $\boldsymbol{A}$ of the LSE is full and for large M. The LSE 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 scaling factor α and the final function might tend to be "blobby" shaped.

Table 1. Typical example of "global" functions

"Global" functions $\phi(r)$			
Thin-Plate Spline (TPS)	$r^2 \lg r$	Inverse Quadric (IQ)	$1/\sqrt{1+\varepsilon r^2}$
Gauss function	$e^{-\varepsilon r^2}$	Multiquadric (MQ)	$\sqrt{1+\varepsilon r^2}$

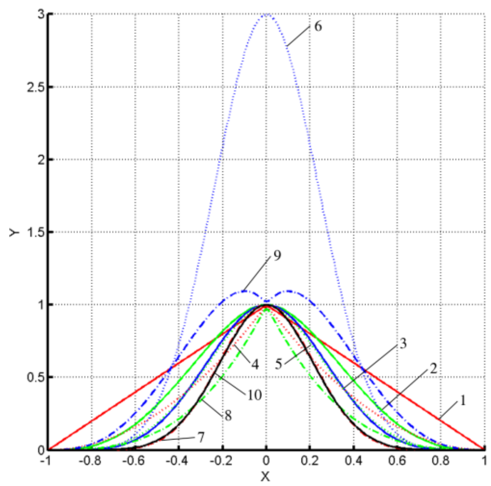


Figure 1. Geometrical properties of CSRBF

Tab.2 presents typical examples of CSRBFs and Fig.1 presents functions behavior geometrically.

Table 2. Typical examples of “local” functions - CSRBF

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	$(1-r)_+^7(16r^2+7r+1)$

The compactly supported RBFs are defined for the interval $r \in \langle 0\ ,\ 1\ \rangle$, but for the practical use a scaling is used, i.e. the value r is multiplied by a scaling factor α, where $\alpha > 0$.
In the case of surface reconstruction from scattered spatial data results is an implicit function $F(\boldsymbol{x}) = 0$. This situation is a little bit more complicated, as the matrix $\boldsymbol{A}$ is generally symmetric, semi-definite or positively definite and the equation $\boldsymbol{Ax} = \boldsymbol{0}$ would have only a trivial solution $\boldsymbol{x} = \boldsymbol{0}$, in general. In this case a surface is considered as an oriented one and additional off-set points are added expecting that a distance in those points is δ. Usually, additional points are given in the normal vector direction, i.e. $+\boldsymbol{n}$ and $-\boldsymbol{n}$ and matrix size is increased by factor 9, i.e. $3n \times 3n$, where n is a number of the given points [3], [14]. Also as number of points might be very high subdivision techniques are used [11].
Meshless techniques are primarily based on approaches mentioned above. 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 [12] etc. Also subdivision or hierarchical methods are used to decrease sizes of computations and increase robustness [15], [21]. Meshless interpolation and approximation techniques are also used in engineering problem solutions, nowadays, e.g. partial differential equations [9], surface modeling [11], surface reconstruction of scanned objects [3], [19], reconstruction of corrupted images [28], etc. More generally, meshless object representation is based on specific interpolation or approximation techniques [1], [2], [9], [17], [19] and [27].
Spatio-temporal data are usually considered as “framed” or “synchronized” in time. The first difficulty is distance computing as distance of two points $\boldsymbol{x_1} = (x_1, y_1, z_1, t_1)$ and $\boldsymbol{x}_2 = (x_2, y_2, z_2, t_2)$ is usually taken as

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 + \gamma^2(t_2 - t_1)^2}$$

where $\gamma = 1$. It is incorrect as we are adding values in [m] and [s]. Therefore γ must be of [m/s], etc.
As the scattered spatio-temporal data are naturally scattered in time as well, i.e. they are not “framed”, meshless methods enable to solve spatio-temporal not “framed” interpolation, i.e. scattered in time,, manipulation and representation in a more consistent way.
Meshless computational methods are the most progressively developing methods in many fields ranging from computational sciences and visualization to computer graphics and manipulation with geometrical models. This progress is given by technological progress as growing computational power enables to solve large problems, which seems to be hard to manage using tessellations, interpolation polygonal meshes of large data sets with higher dimensionality.
However, the *computational complexity* in the meshless methods actually covers complexity of tessellation itself and interpolation and approximation techniques. 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 matrix is sparse and computationally and memory requirements can be decreased significantly and special data structures must be developed to obtain efficient computation.
On the other hand in the case of visualization of physical phenomena, 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 to decrease computational complexity significantly by adding virtual points in the place of interest and use analogy of least square method modified for the RBF representation case. According to experiments made, this approach is quite promising and offers a significant speed up.
Due to CSRBF representation the space 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 architecture is use, e.g. GPU etc., it will lead to fast computation as well. This approach was experimentally verified for scalar and vector data used in visualization of physical phenomena.

3. 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 processed. If the "global" functions are used, the matrix $\boldsymbol{A}$ is full, while if the "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 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 dimensionality of the LSE $\boldsymbol{Ax} = \boldsymbol{b}$. Of course, we are now changing the interpolation property of the RBF to approximation, i.e. the values computed do not pass the given values exactly.
Probably the best way is to formulate the problem using the Least Square Error approximation. Let us consider the formulation of the RBF interpolation again.

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{M} \lambda_j \, \varphi(\|\boldsymbol{x}_i - \boldsymbol{\xi}_j\|)$$

$$h_i = f(\boldsymbol{x}_i) \qquad i = 1, \dots, N$$

where: $\boldsymbol{\xi}_j$ are not given points, but points in a pre-defined "virtual mesh" as only coordinates are needed (there is no tessellation needed). This "virtual mesh" can be irregular, orthogonal, regular, adaptive etc. For simplicity, let us consider the two-dimensional squared (orthogonal) mesh in the following example. Then the $\boldsymbol{\xi}_j$ coordinates are the corners of this mesh. It means that the given scattered data will be actually "re-sampled", e.g. to the squared mesh.

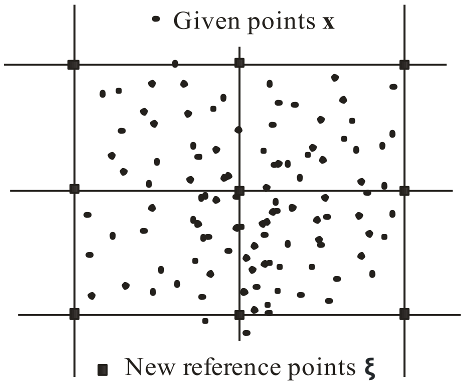


Figure 2. RBF approximation and points' reduction

In many applications the given data sets are heavily over sampled, or for the fast previews, e.g. for the WEB applications, we can afford to "down sample" the given data set. Therefore the question is how to reduce the resulting size of LSE.
Let us consider that for the visualization purposes we want to represent the final potential field in N-dimensional space 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 $\mathbf{10^4}$!

This formulation leads to a solution of a linear system of equations $\boldsymbol{Ax} = \boldsymbol{b}$ where number of rows $M \gg P$, number of unknown $[\lambda_1, \dots, \lambda_P]^T$. As the application of RBF is targeted to high dimensional visualization, it should be noted that the polynomial is not requested for all kernels of the RBF interpolation. But it is needed for $\varphi(r) = r^2 lg\, r$ kernel function (TPS). This reduces the size of the linear system of equations $\boldsymbol{Ax} = \boldsymbol{b}$ significantly and can be solved by the Least Square Method (LSM) as $\boldsymbol{A}^T\boldsymbol{Ax} = \boldsymbol{A}^T\boldsymbol{b}$ or the Singular Value Decomposition (SVD) method can be used.

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

The high dimensional data can be approximated for visualization by RBF efficiently with a high flexibility as it is possible to add additional points in the area of interest to the mesh. It means that a user can add some points to already given mesh and represent easily some details if requested. It should be noted that the use of LSM increases instability of the LSE in general.

4. MORE GENERAL APPROACH

Let us consider more general approach based on extreme finding with constrains given. Let us assume again

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

where $M \leq N$. We want to determine $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_M]^T$ minimizing a quadratic form

$$\frac{1}{2}\boldsymbol{\lambda}^T \boldsymbol{Q} \boldsymbol{\lambda}$$

with a linear constrains $\boldsymbol{A\lambda} - \boldsymbol{f} = \boldsymbol{0}$, where $\boldsymbol{Q}$ is positive and symmetric matrix. This can be solved using Lagrange multipliers $\boldsymbol{\xi} = [\xi_1, \dots, \xi_N]^T$, i.e. minimizing

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

i.e $\boldsymbol{\lambda} =$? and $\boldsymbol{\xi} =$?

So we are getting as the matrix $\boldsymbol{Q}$ is positive

$$\frac{\partial}{\partial \boldsymbol{\lambda}}\left(\frac{1}{2}\boldsymbol{\lambda}^T \boldsymbol{Q} \boldsymbol{\lambda} - \boldsymbol{\xi}^T(\boldsymbol{A\lambda} - \boldsymbol{f})\right) = \boldsymbol{Q\lambda} - \boldsymbol{A}^T \boldsymbol{\xi} = \boldsymbol{0}$$

$$\frac{\partial}{\partial \boldsymbol{\xi}}\left(\frac{1}{2}\boldsymbol{\lambda}^T \boldsymbol{Q} \boldsymbol{\lambda} - \boldsymbol{\xi}^T(\boldsymbol{A\lambda} - \boldsymbol{f})\right) = \boldsymbol{A}^T \boldsymbol{\lambda} - \boldsymbol{f} = \boldsymbol{0}$$

In more compact matrix form we can write

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

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

$$\boldsymbol{\lambda} = \boldsymbol{Q}^{-1}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^T)^{-1}\boldsymbol{f} \qquad \boldsymbol{\xi} = (\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^T)^{-1}\boldsymbol{f}$$

If $\boldsymbol{A} = \boldsymbol{A}^T$ and invertible, computation can be further simplified.

This approach is more robust, however also more computationally expensive.

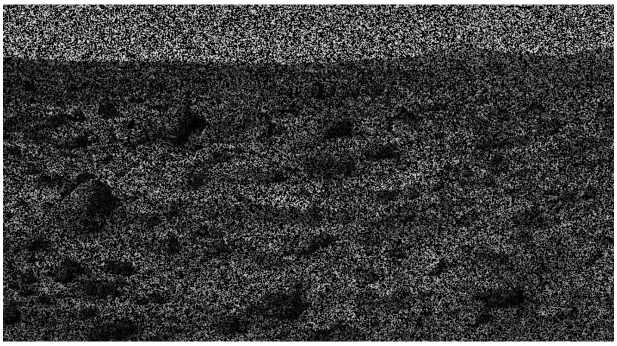
It should be noted, that if the Least Square Method (LSM) is used directly, i.e. $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}$ is to be solved directly, the $\boldsymbol{A}^T\boldsymbol{A}$ matrix is ill conditioned and for large n the system of linear equations is hard to solve.

4. EXPERIMENTAL RESULTS

The presented approach has been taken in the following experiments:

- Reconstructions of images, where corrupted pixels are known. Application of RBF enabled to reconstruct image with more that 60% of corrupted pixels, see Fig.3
- Image inpainting removal, which is applicable e.g. in cultural heritage applications for restoration of old wall paintings corrupted, see Fig.4
- Reconstruction of nearly flat 3D objects using 2D scanner, see Fig.5
- Surface representation for GIS (Geographical Information Systems) applications using approximation instead of interpolation, see Fig.6

The meshless approximation approach can be especially used in data visualization applications, like visualization of potential or vector fields, as in visualization we need to obtain global information of the physical phenomena behavior with acceptable precision.




Original - 60% corrupted pixels Reconstructed image

Figure 3. Corrupted image reconstruction

Presented at IWOR workshop
March 2015, Habana,Cuba

Original image courtesy of Bertalmio, 2000 Inpainting removed

Figure 4. Inpainting removal

Figure 5. Coin scanned and 3D print of the reconstructed coin

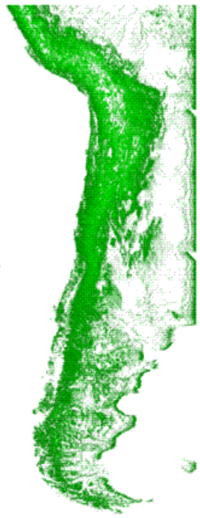


Figure 6. Approximation of 2&1/2D data

The above presented experimental results prove wide applicability of meshless interpolation and approximation methods.

6. SUMMARY AND FUTURE WORK

Interpolation and approximation methods using meshfree (meshless) representation have been described which are convenient for interpolation and approximation of scattered spatio-temporal data in d-dimensional space in general. There is no need to tessellate the data domain and meshless methods offer smoothness of the interpolated or approximated data naturally. Due to those properties the meshless methods are applicable in many areas, e.g. economical, geometrical, in engineering applications including solution of partial differential equations.

However there are many open problems related, especially issues related to robustness of numerical computations in the large data processing case. Problems related to large scattered spatio-temporal data will be explored in future.

Current and future research activities can be found at http://mesfree.zcu.cz

ACKNOWLEDGMENT

The author thanks colleagues at the University of West Bohemia (UWB) in Plzen for their critical comments and constructive suggestions, to anonymous reviewers for their critical view and comments that helped to improve the manuscript. Special thanks belong to former PhD and MSc. students at the UWB Vit Ondracka, Karel Uhlir, Jiri Zapletal and to Jan Hobza for his implementation on GPU and verification of the GPU based image reconstruction.

RECEIVED xxxx, 2015
REVISED xxxx 2015
Presented at IWOR workshop
March 2015, Habana,Cuba

REFERENCES

[1] ADAMS,B., OVSJANIKOV,M., WAND,M., SEIDEL,H.-P., GUIBAS,L.J. (2008): Meshless modeling of Deformable Shapes and their Motion. **ACM SIGGRAPH Symp. on Computer Animation**.

[2] BUHMANN,M.D.(2008): **Radial Basis Functions: Theory and Implementations**. Cambridge Univ.Press.

[3] CARR,J.C., BEATSOM,R.K., CHERRIE,J.B., MITCHELL,T.J., FRIGHT,W.R., FRIGHT,B.C. MCCALLUM,B.C, EVANS,T.R.(2001): Reconstruction and representation of 3d objects with radial basis functions. **Proceedings of SIGGRAPH'01**, pp. 67–76.

[4] CERMAK,M., SKALA,V.: Edge spinning algorithm for implicit surfaces. **Applied Numerical Mathematics**, ISSN 0168-9274, Elsevier, Vol.49, No.3-4, pp.331-342.

[5] CERMAK,M., SKALA,V.(2005): Polygonization of Implicit Surfaces with Sharp Features by the Edge Spinning. **The Visual Computer**, Springer Verlag, Vol.21, No4., ISSN 0178-2789, pp.252-264.

[6] CERMAK,M., SKALA,V.(2007): Polygonization of disjoint implicit surfaces by the adaptive edge spinning algorithm. **Int.J.Computational Science and Engineering**, Vol.3, No.1, pp.45-52.

[7] DUCHON,J.(1997): Splines minimizing rotation-invariant semi-norms in Sobolev space. **Constructive Theory of Functions of Several Variables**, LNCS 571, Springer.

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

[9] FASSHAUER,G.E.(2007): **Meshfree Approximation Methods with MATLAB**. World Scientific Publishing.

[10] LAZZARO,D., MONTEFUSCO,L.B.(2002): Radial Basis functions for multivariate interpolation of large data sets. **Journal of Computational and Applied Mathematics**, 140, pp. 521-536.

[11] MACEDO,I., GOIS,J.P., VELHO,L. (2011): Hermite Interpolation of Implicit Surfaces with Radial Basis Functions. **Computer Graphics Forum**, Vol.30, No.1, pp.27-42.

[12] NAKATA,S., TAKEDA,Y., FUJITA,N., IKUNO,S.(2011): Parallel Algorithm for Meshfree Radial Point Interpolation Method on Graphics Hardware. **IEEE Trans.on Magnetics**, Vol.47, No.5, pp.1206-1209.

[13] PAN,R., SKALA,V.(2011): 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.

[14] PAN,R., SKALA,V.(2012): Surface Reconstruction with higher-order smoothness. **The Visual Computer**, Vol.28, No.2., pp.155-162, ISSN 0178-2789, Springer.

[15] OHTAKE,Y., BELYAEV,A., SEIDEL,H.-P.(2003): **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.

[16] SAVCHENKO,V., PASKO,A., OKUNEV,O., KUNII,T.L.(1995): Function representation of solids reconstructed from scattered surface points & contours. **Computer Graphics Forum**, 14(4), pp.181–188.

[17] SKALA,V.(2012): **Projective Geometry and Duality for Graphics**, Games and Visualization. Course SIGGRAPH Asia 2012, Singapore.

[18] SKALA,V., PAN,R.J., NEDVED,O.(2013): **Simple 3D Surface Reconstruction Using Flatbed Scanner and 3D Print**. ACM SIGGRAPH Asia 2013 poster.

[19] SKALA,V.(2013): **Projective Geometry, Duality and Precision of Computation in Computer Graphics, Visualization and Games**. Tutorial Eurographics 2013, Girona

[20] SKALA,V.(2015): **Meshless Interpolations for Computer Graphics**, **Visualization and Games**. Tutorial Eurographics 2015, Zurich.

[21] SÜßMUTH,J., MEYER,Q., GREINER,G.(2010): Surface Reconstruction Based on Hierarchical Floating Radial Basis Functions. **Computer Graph. Forum**, 29(6): 1854-1864.

[22] VÁŠA,L., SKALA,V.(2009): Combined Compression and Simplification of Dynamic 3D Meshes. The **Journal of Comp. Animation and Virtual Worlds**, Willey Interscience, Vol.20, No.4., pp.447-456.

[23] Váša,L., Skala,V.(2010): Cobra: Compression of the basis for PCA Represented Animations. **Computer Graphics Forum**, Vol.28, No.6, pp.1529-1540.

[24] VÁŠA,L., SKALA,V.(2011): A perception correlated comparison method for dynamic meshes. **IEEE Trans. on Visualization and Computer Graphics**, ISSN 1077-2626, Vol.17, No.2, pp.220-230.

[25] VÁŠA,L., RUS,J.(2012): Dihedral Angle Mesh Error: a fast perception correlated distortion measure for fixed connectivity triangle meshes. **Computer Graphics Forum**, Vol. 31(5).

[26] WENDLAND,H.(2010): **Scattered Data Approximation**. Cambridge University Press.

[27] WRIGHT,G.B.(2003): G. B. Wright. **Radial Basis Function Interpolation: Numerical and Analytical Developments**. University of Colorado, Boulder. PhD Thesis

[28] ZAPLETAL,J., VANECEK,P., SKALA,V.(2009): RBF-based Image Restoration Utilizing Auxiliary Points. **CGI 2009 proceedings**, pp.39-44

Presented at IWOR workshop
March 2015, Habana,Cuba