

Scattered Data Interpolation in N-Dimensional Space

VACLAV SKALA

Department of Computer Science and Engineering
University of West Bohemia, Faculty of Applied Sciences
Univerzitni 8, CZ 306 14 Plzen
Czech Republic
skala@kiv.zcu.cz <http://www.VaclavSkala.eu>

Abstract: - Radial Basis Functions (RBF) interpolation theory is briefly introduced at the “application level” including some basic principles and computational issues. The RBF interpolation is convenient for un-ordered data sets in n-dimensional space, in general. This approach is convenient especially for a higher dimension $N > 2$ conversion to ordered data set, e.g. using tessellation, is computationally very expensive. The RBF interpolation is not separable and it is based on distance of two points. The RBF interpolation leads to a solution of a Linear System of Equations (LSE) $\mathbf{Ax} = \mathbf{b}$. There are two main groups of interpolating functions: “global” and “local”. Application of “local” functions, called Compactly Supporting Functions (CSF), can significantly decrease computational cost as they lead to a system of linear equations with a sparse matrix. The RBF interpolation can be used also for image reconstruction, inpainting removal, for solution of Partial Differential Equations (PDE) etc.

Key-Words: - RBF interpolation, radial basis function, image reconstruction, incremental computation, RBF approximation.

1 Introduction

Interpolation is one of the most frequent operations used in computational techniques. Several techniques have been developed for data interpolation, but they expect some kind of data “ordering”, e.g. structured mesh, rectangular mesh, unstructured mesh etc. The typical example is a solution of partial differential equations (PDE) where derivatives are replaced by differences and rectangular mesh is used in the vast majority of cases. Nevertheless in many engineering problems, data are not ordered and they are scattered in n-dimensional space, in general. Usually, in technical applications the scattered data are tessellated using triangulation but this approach is quite prohibitive for the case of n-dimensional data interpolation because of the computational cost. An interesting technique is n dimensional data interpolation using Radial Basis Functions (RBF). The RBF interpolation is computationally more expensive because interpolated data are not ordered, but offers quite interesting applications with acceptable computational cost, e.g. solution of partial differential equations, image reconstruction, neural networks, fuzzy systems, GIS systems, optics and interferometry etc.

2 Problem Formulation

Interpolation is very often used and mostly linear interpolation is used in technical applications. Let us analyze first different types of data to be processed. Also there is a question whether the Euclidean space representation is the best for computing and engineering applications. It is well known that the division operation is very dangerous in numerical computations and causes severe problems in numerical methods. Also it is known that computations can be made in the projective extension of the Euclidean space [20][21][23][27]. The projective formulation of numerical problems leads to very interesting questions, e.g. an explicit solution of LSE is equivalent to the cross-product. Why the division operation in the Gauss-Seidel or similar methods is needed? [Ska, Ondra]. The projective space representation and the principle of duality also helps to solve some problems more efficiently [19][20][25]. Also Non-rational uniform B-Splines (NURBS) are actually curves or surfaces defined using the projective extension of the Euclidean space.

In the following we use the Euclidean space representation to explain the fundamental principles. We will explore incremental computation of RBF as well.

3 Data Classification

Before analyzing methods for interpolation, it is reasonable to classify data to be processed. It seems to be a simple task, but let us explore it more deeply. Generally, the data can be represented by:

1. Coordinates, e.g. by points $\{\mathbf{x}_i\}_1^M$ in computer graphics, which forms triangular mesh in E^2 , or scalar values, e.g. representing temperatures etc.,
2. coordinates and associated values $\{(\mathbf{x}_i, \mathbf{h}_i)\}_1^M$, e.g. coordinates of points \mathbf{x}_i and associated values \mathbf{h}_i with each point.

The dimensionality of a vector of coordinates $dim(\mathbf{x}_i) = M$, i.e. $\mathbf{x}_i = [x_1, \dots, x_M]^T$, while the dimensionality of a vector of values $dim(\mathbf{h}_i) = p$, i.e. $\mathbf{h}_i = [h_1, \dots, h_p]^T$.

It can be seen that those two cases are quite different cases if an interpolation is to be used. Also data can be

- hierarchical
- non-hierarchical

or

- adaptive to some physical phenomena
- non-adaptive

and

- dynamic (t-variant) in coordinates \mathbf{x}_i or in values \mathbf{h}_i or both! or
- static

- | | |
|--------------|----------------|
| - Un-ordered | - Scattered |
| | - Clustered |
| - Ordered | - Unstructured |
| | - Structured |
| | - Non-regular |
| | - Semi-regular |
| | - Regular |

Table 1: A simple classification of data

In the case of un-ordered data, mostly some tessellation techniques like triangularization in the E^2 case or tetrahedronization in the E^3 case are used and generally an unstructured mesh is obtained.

The semi-regular mesh is obtained just in the case when data are ordered in a rectangular grid and Delaunay triangulation is used. It should be noted that this is a very unstable situation, as due to some small shifts in coordinates, the tessellation can be totally changed.

Interpolation techniques on “ordered” data sets are well known and used in many packages.

Let us explore how to interpolate values \mathbf{h}_i in the given un-ordered $\{(\mathbf{x}_i, \mathbf{h}_i)\}_1^M$ data set. Of course,

there is a theoretical possibility to use a tessellation in order to get an ordered unstructured mesh, but this process is computationally very expensive as the computational complexity of the tessellation grows with the dimension N non-linearly and complexity of the implementation grows as well.

On the other hand, there are interpolation techniques applicable for un-ordered data sets. One of such technique is based on Radial Basis Functions (RBF) which is especially convenient for the interpolation in the n -dimensional space. The RBF interpolation based on radial basis functions is quite simple from a mathematical point of view.

Let us consider the case, when \mathbf{h}_i are scalar values for RBF interpolation explanation. The RBF interpolation is based on computing of the distance of two points in the N -dimensional space and is defined by a function

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

$$r_j = \|\mathbf{x} - \mathbf{x}_j\|$$

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

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

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

$$h_i = f(\mathbf{x}_i) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|) + P_k(\mathbf{x}_i)$$

$$i = 1, \dots, M$$

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

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

$$= \sum_{j=1}^M \lambda_j \varphi_{i,j} + \mathbf{a}^T \mathbf{x}_i + a_0$$

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

and additional conditions are applied:

$$\sum_{j=1}^M \lambda_j = 0 \quad \sum_{j=1}^M \lambda_j \mathbf{x}_j = \mathbf{0}$$

It can be seen that for N -dimensional case a system of $(M + N + 1)$ LSE has to be solved, where M is a number of points in the dataset and N is the dimensionality of data.

For $N=2$ vectors \mathbf{x}_i and \mathbf{a} are given as $\mathbf{x}_i = [x_i, y_i]^T$ and $\mathbf{a} = [a_x, a_y]^T$. Using the matrix notation we can write for 2-dimensions:

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

$$\begin{bmatrix} \mathbf{B} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} \quad \mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\mathbf{a}^T \mathbf{x}_i + a_0 = a_x x_i + a_y y_i + a_0$$

It can be seen that 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 \mathbf{B} is “full”, if “local” functions (Compactly supported RBF – CSRBF) are used, the matrix \mathbf{B} can be sparse.

The radial basis functions interpolation was originally introduced by [5] by introduction of multiquadric method in 1971, which he 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 by [9] and [12] introduced Compactly Supported RBF (CSRBF) as

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

where: $P(r)$ is a polynomial function and q is a parameter.

Theoretical problems with stability and solvability were solved by [6] and [13]. 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 \mathbf{A} of the LSE is full and for large M is becoming ill conditioned and problems with convergence can be expected. On the other hand if the CSRBFs are taken, the matrix \mathbf{A} is becoming relatively sparse, i.e. computation of the LSE will be faster, but we

need to carefully select the scaling factor and the final function tends to be “blobby” shaped.

“Global“ functions	$\phi(r)$
Thin-Plate Spline (TPS)	$r^2 \log r$
Gauss function	$\exp(-(\epsilon r)^2)$
Inverse Quadric (IQ)	$1/(1+(\epsilon r)^2)$
Inverse multiquadric (IMQ)	$1/\sqrt{1+(\epsilon r)^2}$
Multiquadric (MQ)	$\sqrt{1+(\epsilon r)^2}$

Table 1 Typical example of “global” functions”

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

Table 2 Typical examples of “local” functions - CSRBF [13]

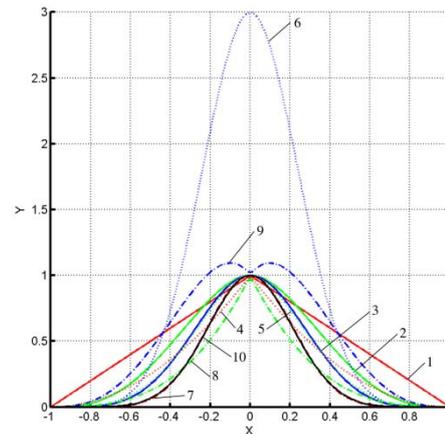


Figure 1 Geometrical properties of CSRBF [13]

Tab.2 presents typical examples of CSRBFs. They are defined for the interval $<0, 1>$, but for the practical use a scaling is used, i.e. the value r is multiplied by a scaling factor α , where $0 < \alpha < 1$.

4 Incremental computation

As for many applications, the number of points is high and some data are to be deleted and new inserted, it is not possible to recompute the whole LSE due to computational complexity. In this case the incremental computation of RBF is to be used. The algorithm itself is simple [22-incremental] and can be simply described as follows:

Let us consider a matrix \mathbf{Q} of $(M+1) \times (M+1)$ and a matrix \mathbf{A} of $M \times M$ in the following block form:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^T & c \end{bmatrix}$$

Then the inverse of the matrix \mathbf{Q} applying the rule above can be written as:

$$\mathbf{Q}^{-1} = \begin{bmatrix} \left(\mathbf{A} - \frac{1}{c} \mathbf{b} \mathbf{b}^T\right)^{-1} & -\frac{1}{c} \mathbf{A}^{-1} \mathbf{b} \\ -\frac{1}{k} \mathbf{b}^T \mathbf{A}^{-1} & \frac{1}{k} \end{bmatrix}$$

where: $k = c - \mathbf{b}^T \mathbf{A}^{-1} \mathbf{b}$

We can easily simplify this equation if the matrix \mathbf{A} is symmetrical as:

$$\begin{aligned} \boldsymbol{\xi} &= \mathbf{A}^{-1} \mathbf{b} & k &= c - \boldsymbol{\xi}^T \mathbf{b} \\ \mathbf{M}^{-1} &= \frac{1}{k} \begin{bmatrix} k \mathbf{A}^{-1} + \boldsymbol{\xi} \otimes \boldsymbol{\xi}^T & -\boldsymbol{\xi} \\ -\boldsymbol{\xi}^T & 1 \end{bmatrix} \end{aligned}$$

where: $\boldsymbol{\xi} \otimes \boldsymbol{\xi}^T$ means the tensor multiplication. It can be seen that all computations needed are of $O(M^2)$ computational complexity.

It means that we can compute an inverse matrix incrementally with $O(M^2)$ complexity instead of $O(M^3)$ complexity required originally in this specific case. It can be seen that the structure of the matrix \mathbf{Q} is “similar to the matrix of the RBF specification [28].

5 RBF Approximation

The RBF interpolation relies on solution of a LSE $\mathbf{A} \mathbf{x} = \mathbf{b}$ 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 \mathbf{A} is full, while if the “local” functions are used (CSRBF), the matrix \mathbf{A} is sparse.

However, in visualization applications it is necessary to compute the final function $f(\mathbf{x})$ many many times and even for already computed λ_i values, the computation of $f(\mathbf{x})$ is too expensive. Therefore it is reasonable to significantly “reduce” the dimensionality of the LSE $\mathbf{A} \mathbf{x} = \mathbf{b}$. 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.

$$\begin{aligned} f(\mathbf{x}_i) &= \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x}_i - \boldsymbol{\xi}_j\|) + \mathbf{a}^T \mathbf{x}_i + a_0 \\ h_i &= f(\mathbf{x}_i) \quad i = 1, \dots, N \end{aligned}$$

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.

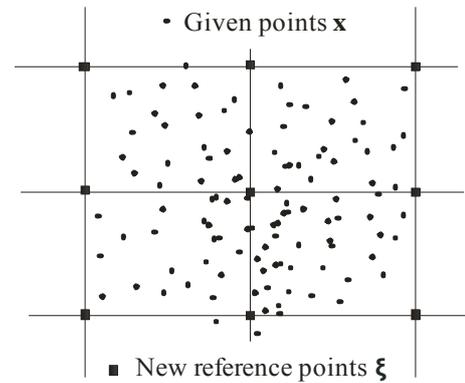


Figure 2. RBF approximation and points' reduction

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 $f(\mathbf{x})$ 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 10^4 !**

This formulation leads to a solution of a linear system of equations $\mathbf{A} \mathbf{x} = \mathbf{b}$ 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 $\mathbf{Ax} = \mathbf{b}$ significantly and can be solved by the Least Square Method (LSM) as $\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$ or Singular Value Decomposition (SVD) can be used.

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

The high dimensional data can be approximated for visualization by RBF efficiently with a high flexibility as it is possible to add additional points of an 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.

6 Experimental Evaluation

The RBF interpolation is a very powerful tool for interpolation of data in N -dimensional space in general. In order to demonstrate the functionality the RBF, we have recently used RBF for reconstruction of damaged images by a noise or by inpainting. Also a surface reconstruction has been solved by the RBF interpolation well. Fig. 3 illustrates the power of the RBF interpolation [2][3][8][15][24][26][28].

The RBF interpolation gives quite good results even if the images are heavily damaged. The advantages of RBF interpolation over the other interpolations have been proved even though that the RBF interpolation causes some additional computational cost as the RBF is primarily targeted for scattered data interpolation.

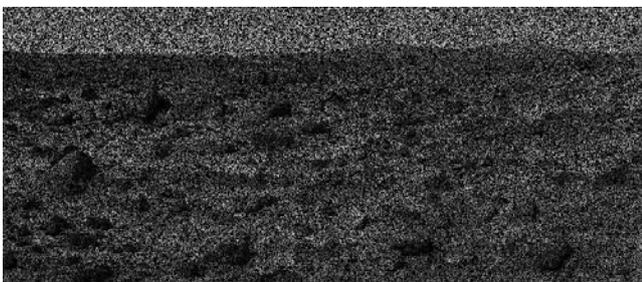


Figure 3a. Original image with 60% of damaged pixels [13]

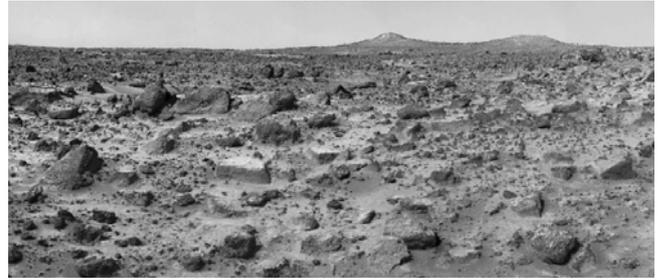


Figure 3b. Reconstructed image [13]

7 Conclusion

The radial basis functions (RBF) interpolation is a representative interpolation method for unordered scattered data sets. It is well suited approach for solving problems without meshing the data domain. RBF interpolations are used in many computational fields, e.g in solution of partial differential equations etc. The RBF interpolation formulation supports the N -dimensional space naturally.

This paper briefly describes a principle of the RBF incremental computation and shows the decrease of the computational complexity from approx. $O(M^3)$ to $O(M^2)$ for a point insertion and a point removal.

It also presents a method for “resampling” the data processed as the approximation is acceptable in many applications, namely in visualization. The approach enables to increase details for visualization by adding new points to the “virtual mesh”, if more details are needed. It is necessary to mention, that there is no mesh actually needed and only points of the “virtual mesh” need to be defined.

Acknowledgment

The author thanks to colleagues at the University of West Bohemia (UWB) in Plzen and at the VSB-Technical University in Ostrava 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, Lukas Loukota, Karel Uhlir, Jiri Zapletal and to Jan Hobza for his implementation on GPU and verification of the GPU based image reconstruction.

References:

- [1] Baxter, B.J.Ch.: The Interpolation Theory of Radial Basis Functions, PhD thesis, Trinity College, Cambridge University, U.K., 1992

- [2] Bertalmio, M., Sapiro, G., Ballester, C., Caselles, V.: Image Inpainting, Proceedings of SIGGRAPH'00, Computer Graphics, New Orleans, 23-28, pp.417-424, 2000
- [3] Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R.: Reconstruction and representation of 3D objects with radial basis functions, Computer Graphics (SIGGRAPH 2001 proceedings), pp. 67-76, 2001.
- [4] Duchon, J.: Splines minimizing rotation-invariant semi-norms in Sobolev space", Constructive Theory of Functions of Several Variables, Springer Lecture Notes in Math, 21, pp. 85-100, 1977
- [5] Hardy, L.R.: Multiquadric equations of topography and other irregular surfaces, J. Geophy. Res., 76, pp. 1905-1915, 1971
- [6] Micchelli, C.A.: Interpolation of scattered data: distance matrix and conditionally positive definite functions, Constr. Approx., No.2, pp. 11-22, 1986
- [7] Jakobsson, S., Andersson, B., Edelvik, F.: Rational radial basis function interpolation with applications to antenna design Journal of Computational and Applied Mathematics, Vol. 233(4), pp. 889-904, December 2009.
- [8] Pan, R., Skala, V.: Implicit surface modeling suitable for inside/outside tests with radial basis functions, 10th International Conference on Computer Aided Design and Computer Graphics (CAD/Graphics), 2007
- [9] 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, Springer, 2011
- [10] Pan, R., Skala, V.: Continuous Global Optimization in Surface Reconstruction from an Oriented Point Cloud, Computer Aided Design, Vol.43, No.8, pp.896-901, Elsevier, 2011
- [11] Pan, R., Skala, V.: Surface Reconstruction with higher-order smoothness, The Visual Computer, on-line , Springer, 2011
- [12] Schagen, I.P.: Interpolation in Two Dimension – A New Technique, J. Inst. Maths Applics, 23, pp. 53-59, 1979
- [13] Uhlir, K., Skala, V.: Radial basis function use for the restoration of damaged images. In Computer vision and graphics. Dordrecht: Springer, pp. 839-844. 2006
- [14] Wang, Ch.C.L., Kwok, T.-H.: Interactive Image Inpainting using DCT Based Exemplar Matching, ISVC 2009, LNCS 5876, pp.709-718, 2009
- [15] Wendland, W.: Computational aspects of radial basis function approximation, in K. Jetter et al. (eds.) Topics in Multivariate Approximation and Interpolation , Elsevier, pp. 231-256, 2005.
- [16] Wright, G.B.: Radial Basis Function Interpolation: Numerical and Analytical Developments", University of Colorado, PhD Thesis, 2003.
- [17] Zapletal, J., Vanecek, P., Skala, V.: RBF-based Image Restoration Utilising Auxiliary Points, CGI 2009 proceedings, ACM, pp.39-44, 2009.
- [18] Ohtake, Y. Belyaev, A., Seidel, H.-P.: 3D Scattered Data Interpolation and Approximation with Multilevel Compactly Supported RBFs, Graphical Models, Vol.67, No.3., pp.150-165, 2005.
- [19] Skala, V.: Barycentric Coordinates Computation in Homogeneous Coordinates, Computers & Graphics, Elsevier, ISSN 0097-8493, IF= 0,787, Vol. 32, No.1, pp.120-127, 2008
- [20] Skala, V.: Geometric Computation, Duality and Projective Space, IW-LGK workshop proceedings, pp.105-111, Dresden University of Technology, 2011
- [21] Skala, V., Ondracka, V.: A Precision of Computation in the Projective Space, Recent Researches in Computer Science, 15th WSEAS Int.Conference on Computers, Corfu, pp.35-40, Greece, 2011
- [22] Skala, V.: Incremental Radial Basis Function Computation for Neural Networks, WSEAS Transactions on Computers, Issue 11, Volume 10, pp. 367-378, 2011.
- [23] Skala, V.: Robust Computation in Engineering, Geometry and Duality – TUTORIAL, 7th Int.Conf. on Systems, ICONS 2012, St. Gilles, Reunion Island, IARIA, 2011
- [24] Skala, V.: Radial Basis Functions for High Dimensional Visualization, VisGra - ICONS 2012, accepted for publication, Saint Gilles, Reunion Island, IARIA, 2012
- [25] Skala, V.: Interpolation and Intersection Algorithms and GPU, VisGra - ICONS 2012, i, Saint Gilles, Reunion Island, IARIA, 2012
- [26] Skala, V.: Radial Basis Functions: Interpolation and Applications- An Incremental Approach, Applied Mathematics, Simulation and Modeling - ASM 2010 conference, NAUN, Greece, pp.209-213, 2010
- [27] Skala, V.: Duality and Intersection Computation in Projective Space with GPU support, Applied Mathematics, Simulation and Modeling - ASM conference, NAUN, Greece, pp.66-71, 2010
- [28] Skala, V.: Progressive RBF Interpolation, Afrigraph 2010, pp.17-20, ACM, ISBN:978-1-4503-0118-3, 2010

WEB references

FastRBF: <http://www.farfieldtechnology.com/>.
<retrieved: 2012-02-02>