Article

Application of Gaussian Radial Basis Functions for Fast Spatial Imaging of Ground Penetration Radar Data Obtained on an Irregular Grid

Krzysztof Jasek, Mateusz Pasternak , Witold Miluski, Jarosław Bugaj * and Michał Grabka

Faculty of Electronics, Military University of Technology, 00-908 Warsaw, Poland; krzysztof.jasek@wat.edu.pl (K.J.); mateusz.pasternak@wat.edu.pl (M.P.); witold.miluski@wat.edu.pl (W.M.); michal.grabka@wat.edu.pl (M.G.)

* Correspondence: jaroslaw.bugaj@wat.edu.pl; Tel.: +48-261-839-223

Abstract: Spatial imaging of ground penetrating radar (GPR) measurement data is a difficult computational problem that is time consuming and requires substantial memory resources. The complexity of the problem increases when the measurements are performed on an irregular grid. Such grid irregularities are typical for handheld or flying GPR systems. In this paper, a fast and efficient method of GPR data imaging based on radial basis functions is described. A compactly supported modified Gaussian radial basis function (RBF) and a hierarchical approximation method were used for computation. The approximation was performed in multiple layers with decreasing approximation radius, where in successive layers, increasingly finer details of the imaging were exposed. The proposed method provides high flexibility and accuracy of approximation with a computational cost of $N \cdot \log(N)$ for model building and $N \cdot M$ for function evaluation, where N is the number of measurement points and M is the number of approximation centres. The method also allows for the control smoothing of measurement noise. The computation of one high-quality imaging using 5000 measurement points utilises about 5 s on an Intel Core i5-7200U CPU 2.5 GHz, 8 GB RAM computer. Such short time enables real-time image processing during field measurements.

Keywords: ground penetrating radar data imaging; irregular grid of the measurement; real-time imaging; gaussian radial basis function; hierarchical approximation method

Citation: Jasek, K.; Pasternak, M.; Miluski, W.; Bugaj, J.; Grabka, M. Application of Gaussian Radial Basis Functions for Fast Spatial Imaging of Ground Penetration Radar Data Obtained on an Irregular Grid. *Electronics* **2021**, *10*, 2965. <https://doi.org/10.3390/electronics10232965>

Academic Editor: Reza K. Amineh

Received: 24 October 2021

Accepted: 24 November 2021

Published: 28 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Ground penetrating radars (GPRs) are electromagnetic systems for soil subsurface structure imaging. To obtain imaging that is precise enough for correct interpretation beside of specific electrical system the respectively accurate positioning, data must be registered. Software that is commonly used for processing the obtained data set commonly assumes that the measurements were made on a simple rectangular grid. The GPR operators usually try to fulfil such requirements to obtain the best possible imaging using the software.

Unfortunately, there are many situations where the requirements cannot be fulfilled. Typical situations take place in handheld systems where measurement points are located randomly toward a curved trajectory. In general, the problem concerns all scanning echolocation systems where spatial distribution of measurement points is random but dense enough. A separate issue is the accurate positioning method.

When the measurements cannot be made on a simple rectangular grid due to handheld operation or inaccessibility of some locations, as a result the data from irregularly located measurement points are registered. Meshless approximation methods should then be used for spatial imaging. At present, a variety of meshless methods are used in geosciences for digital terrain model generation, such an inverse distance weighting (IDW) [1,2], triangulated irregular network (TIN) with linear interpolation [3,4], natural neighbour (NN) [5,6] radial basis functions (RBFs) [7] and kriging [8]. The implementation

of these algorithms can be found many different applications e.g., Ref. [9]. They can now be found on geophysical packages, such as Surfer [10], Geosoft Oasis Montaj Geoscience [11] and TerrSet [12]. However, all of the packages are geared towards solving slightly different physical problems than GPR data analysis. First, with typical geophysical surveys, there are relatively equidistant points, except for some local areas. In the case of handheld GPR, there are densely distributed points along the curved scanning path, and the distances between successive paths are very large compared to the distances between successive survey points. This causes problems with setting up a computational mesh and ill-conditioning of many classical methods. Another problem arises from the large amount of measurement data. A single point measurement generates hundreds of numbers corresponding to echo values coming from different depths. This requires data pre-processing to prepare the input data for geophysical packages. All this makes the available computational packages, which are valuable for the final analysis of the results, unable to fulfil their role during field measurements, where real-time processing and simplicity of use are required.

A method to get around this problem is to create custom software. All methods of interpolation and approximation of multidimensional data boil down to solving systems of matrix equations. Due to the complexity of computational procedures, ready-made libraries of numerical procedures, which contain optimised implementations of solvers for linear algebraic equation systems, are extremely helpful. One such package is ALGLIB [13], which also has procedures for interpolation and approximation of multidimensional data.

This paper presents a method of visualising spatial GPR data using RBFs. A modified Gaussian function was used as the RBF, so that it assumed a value of zero at the given distance from the centre. This method provides ease of use, flexibility and high interpolation accuracy. It also allows for the suppression of measurement noise.

2. Radial Basis Functions (RBFs)

A radial basis functions (RBFs), ϕ , is a function whose value depends only on the distance from the origin or some other fixed point, $\mathbf{c}$, called a centre, so that $\phi(\mathbf{x}) = \phi(\|\mathbf{x} - \mathbf{c}\|)$. The distance, $r = \|\mathbf{x} - \mathbf{c}\|$, is usually Euclidean; although, other metrics are sometimes used. Since the average distance between points can vary greatly depending on the problem being studied, it is convenient to use the concept of the RBF radius, R_0 . It is the shape parameter that defines the behaviour of the function. Using the radius, the scaled distance can be defined as $r_s = r/R_0$ and used to calculate RBFs.

There are two main groups of basis functions: global RBFs and compactly supported RBFs (CS-RBFs) [14]. Global RBFs have non-zero values at any distance from the centre; although, some of them can drop quickly with distance. CS-RBFs are positive only within a certain radius from the centre and are zero outside of it.

Scattered data adjusting with CS-RBFs caused faster and simpler calculation as a linear equations system has a sparse matrix. Nevertheless, approximation using CS-RBFs is susceptible to the density of the approximated scattered data and to the selection of shape parameter. Typical examples of local RBFs are shown in Table 1.

Table 1. Examples of compactly supported RBFs; “+” means value zero out of the scaled distance $<0, 1>$ [15].

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

Typical global RBFs are Gauss, $\exp(-r_s^2)$, inverse quadratic, $(1 + r_s^2)^{-1}$, and inverse multiquadric, $(1 + r_s^2)^{-1/2}$. Along with increasing radius, r , RBFs are monotonically

decreases, strictly positive definite, infinitely differentiable and convergent to zero. Another globally well-known RBFs is thin plate spline (TPS), $r^{2k} \ln(r)$, $k \in N$, without shape parameter and which is disparate while increasing radius. TPS has a singularity at the origin, which is removable for the function, and its first derivative but this singularity is not removable for the second derivative of TPS [16]. In the case of a great number of points, global RBFs cause an ill-conditioned linear system of equations (LSE) with a dense matrix, which can lead to convergence problems.

The ALGLIB package implements both kinds of RBFs: the global Gaussian function and compactly supported. The classical Gaussian function takes small values already at a distance of about $3R_0$ from the center and can easily be modified to be compactly supported. ALGLIB implementation of such function is expressed by the following exponential form:

$$\exp\left(1 - \frac{r^2}{R_0^2} - \frac{1}{1 - r^2/(3R_0)^2}\right) \quad (1)$$

RBF (1) differs little in shape from the Gaussian function while having all the advantages of a compactly supported function. It reaches zero at a distance of $R_0 = 3$. The shapes of both functions are shown in Figure 1.

Figure 1. Shapes of global and compactly supported Gaussian RBFs.

Radial basis functions can be used to process scattered data from GPR. Two types of approaches are used for this purpose: RBF interpolation or approximation.

3. RBF Interpolation

In the interpolation process, we are looking for a function $f(\mathbf{x})$ that at N measuring points $\mathbf{x} = \{\mathbf{x}\}_1^N$, $\mathbf{x} = (x_1, x_2)^T$ obtains the same values as the measured $\mathbf{h} = \{h\}_1^N$. RBF interpolation is based on determining the distance of two points (in the d -dimensional space in general) and is provided by the following form:

$$f(\mathbf{x}) = \sum_{j=1}^N w_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) \quad (2)$$

where the interpolating function $f(\mathbf{x})$ is represented as a sum of N RBFs, each centered at a different data point, $\mathbf{x}_j$, and weighted by an appropriate coefficient, w_j . For the given dataset, this leads to an LSE, as follows:

$$f(\mathbf{x}_i) = \sum_{j=1}^N w_j \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) \quad (3)$$

which can be represented in matrix form as the following:

$$\mathbf{A}\mathbf{w} = \mathbf{h} \quad (4)$$

where $\mathbf{A} = \{A_{i,j}\} = \{\phi(\mathbf{x}_i - \mathbf{x}_j)\}$ is the $N \times N$ symmetric matrix, the vector $\mathbf{w} = (w_1, w_2, \dots, w_N)^T$ is the vector of unknown weights, and $\mathbf{h} = (h_1, h_2, \dots, h_N)^T$ is the vector of values in the given points. After solving the system of linear equations, the interpolated value at point $\mathbf{x}$ is computed using Equation (2). The disadvantage of RBF interpolation is the large and usually ill-conditioned design matrix of the LSE.

The calculation of the coefficients, $\mathbf{w}$, for all measuring points (interpolation) is, in many cases, computationally expensive. However, for good-quality spatial imaging, it is often enough to have a much smaller number of coefficients, especially if the measurements are made densely. Then, the computed function, $f(\mathbf{x})$, does not take exactly experimental values, but the errors may be so small that they can be completely accepted. This approach is called the RBF approximation [17].

The simplest approach to solving the approximation problem is the least square method (LSM). We consider a modified interpolation formula, as follows:

$$f(\mathbf{x}) = \sum_{j=1}^M w_j \phi(\|\mathbf{x} - \mathbf{c}_j\|) \quad (5)$$

where $\mathbf{c}_j$ is not the measuring point but the reference point of the grid on which we want to perform the calculations. The mesh contains M nodes and may be irregular in general; although, computationally convenient rectangular meshes are usually used. The number of nodes can be many times smaller than the number of measurement points ($M \ll N$), which speeds up the calculations accordingly.

For the dataset, Equation (5) gives an overdetermined LSE, as follows:

$$f(\mathbf{x}_i) = \sum_{j=1}^M w_j \phi(\|\mathbf{x}_i - \mathbf{c}_j\|) \quad i = 1, \dots, N \quad (6)$$

The linear system of Equation (6) can be represented as the matrix Equation (4). In the expanded form:

$$\begin{bmatrix} \phi_{1,1} & \cdots & \phi_{1,M} \\ \vdots & \ddots & \vdots \\ \phi_{i,1} & \cdots & \phi_{i,M} \\ \vdots & \ddots & \vdots \\ \phi_{N,1} & \cdots & \phi_{N,M} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} h_1 \\ \vdots \\ h_i \\ \vdots \\ h_M \end{bmatrix} \quad (7)$$

This LSE can be solved by the LSM as $\mathbf{A}^T \mathbf{A} \mathbf{w} = \mathbf{A}^T \mathbf{h}$, singular value decomposition, etc. Reducing the number of points speeds up the $\mathbf{w}$ computation by a factor of $O((M/N)^3)$ as the LSE computational is of $O(M^3)$ [18].

This method can have problems with stability and solvability. Therefore, the RBF approximant (5) is usually extended by the polynomial function $P_k(\mathbf{x})$ of degree k , as follows:

$$f(\mathbf{x}) = \sum_{j=1}^M w_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) + P_k(\mathbf{x}) \quad (8)$$

In practice, a linear polynomial is the following, or a constant term is used:

$$f(\mathbf{x}) = \sum_{j=1}^M w_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) + P_k(\mathbf{x}) \quad (9)$$

For the linear term, the matrix equation can be rewritten as the following:

$$\begin{bmatrix} \phi_{1,1} & \cdots & \phi_{1,M} & x_1 & y_1 & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & 1 \\ \phi_{i,1} & \cdots & \phi_{i,M} & x_i & y_i & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & 1 \\ \phi_{N,1} & \cdots & \phi_{N,M} & x_N & y_N & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_M \\ a_x \\ a_y \\ a_0 \end{bmatrix} = \begin{bmatrix} h_1 \\ \vdots \\ h_i \\ \vdots \\ h_N \end{bmatrix} \quad (10)$$

For two-dimensional space, a linear system of N equations in $M + 3$ variables must be solved, where N is the number of points in the given dataset and M is the number of references. Such system is overdetermined again and can also be solved by the same methods. If global RBFs are used, matrix $\mathbf{A}$ is dense, while in the case of CS-RBF this matrix can be sparse.

If LSE is solved directly, the matrix equation $\mathbf{A}^T \mathbf{A} \mathbf{w} = \mathbf{A}^T \mathbf{h}$ must be solved. The $\mathbf{A}^T \mathbf{A}$ matrix is ill-conditioned, and for large M the system of linear equations is difficult to solve [19]. Moreover, the matrix size is $2M \times 2M$. This means that the memory requirements are not acceptable even for medium data sets. For real applications of the RBF approximation, we need to decrease memory requirements significantly. There are various approaches to solving these problems, the most important of which are listed below.

4. Model Construction

After selecting the basis function (and optionally the polynomial term), we calculate the weight coefficients, and thereby can build the model. There are several ways to approach the issue. Certain functions can be used for both global and compact basis functions [13]:

- “Straightforward” approach: using a dense solver to the linear system. Performs with any basis function, but it has low efficiency as the solution time increases with the number of points, N , as $O(N^3)$. Thus, this approach is limited to several thousand points.
- Preconditioning: preconditioning approximate cardinal basis function (ACBF) to accelerate convergence [20]. In this case, several non-local basis functions are constructed to build another local basis function (cardinal basis function). Usually, the solution itself only requires some of the total processing time, and most of it is spent on the construction of the new basis. In dense systems that we obtain with global basis functions, preconditioning allows for good performance with iterative solver to be achieved. In the case of CS-RBFs, compact functions become “more compact,” which allows us to use iterative solvers with quite a large radius, equal to two distances to the nearest neighbour.

Regarding compact basis functions, it is possible to use a sparse linear solver, both iterative or direct. Convergence of the iterative solver depends on system conditioning, what easily solve problems with a small radius (approximately equal to the distance to the nearest neighbour), but it will have difficulties with larger one. From the different side, it does not need additional memory. Problems with larger radii (up to 2–3 distances) can be solved by direct solver but it requires more memory to store Cholesky or LU apportionment of the system.

Dividing the original dataset into few smaller ones [21] is the next method used with compact functions, which allows for the problem of RBF construction to be tackled apart for above mentioned smaller parts, and next, the fractional model is put together into a larger, global one. However, composition of the model is imperfect, and the artifacts are nearly to the boundaries what causes that the problem with interpolation have to be tackled anew to correct the errors of the first model. As an outcome, we obtain iterative algorithms, which repeat the “separate-solve-compose-correct” loop several times before achieving the desirable accuracy.

The above described algorithms have one common attribute: they solve the original problem without any modifications. Nevertheless, by changing the problem characteristic slightly we can receive a good interpolant with repeatedly greater efficiency. For instance, we can use more functions with different radii, instead of N Gaussian functions with the same radii. The approximation difficulty can be tackled in a few steps by building a hierarchy of a precise model with decreasing radii. Such an approach has different terms: multiscale RBF, multilevel RBF, hierarchical RBF or multilayer RBF (RBF-ML). Its main asset is high efficiency and precision due to the hierarchy of easy to build application.

5. RBF-ML Interpolation Algorithm

In this paper, the RBF-ML algorithm implemented in the ALGLIB package was used. It has three parameters: the initial radius, R_0 , the number of layers, N_L , and the regularisation coefficient, λ . This algorithm builds a hierarchy of models with decreasing radii [13].

In the initial (optional) iteration, the algorithm builds a linear least squares model. The values predicted by the linear model are subtracted from the function values at the nodes, and the residual vector is passed to the next iteration. In the first iteration, a traditional RBF model with a radius equal to R_0 is built. However, it does not use a dense solver and does not try to solve the problem exactly. It solves the least squares problem by performing a fixed number (approximately 50) of LSQR [22] iterations. Usually, the first iteration is sufficient. Additional iterations will not improve the situation because with such a large radius, the linear system is ill-conditioned. The values predicted by the first layer of the RBF model are subtracted from the function values at the nodes, and, again, the residual vector is passed to the next iteration. With each successive iteration, the radius is halved by performing the same constant number of LSQR iterations, and the forecasts of the new models are subtracted from the residual vector.

In all subsequent iterations, a fine regularisation can be applied to improve the convergence of the LSQR solver. Larger values for the regularisation factor can help reduce data noise. Another way of controlled smoothing is to select the appropriate number of layers.

Figure 2 shows an example of B-scan and a method of approximation by a hierarchical model. Subsequent layers have a radius that is twice as small and explain the residues after the previous layer. As the radius decreases, the finer details of the B-scan are reproduced.

The hierarchical algorithm has several significant advantages:

- Gaussian CS-RBFs produce linear systems with sparse matrices, enabling the use of the sparse LSQR solver, which can work with the rank defect matrix;
- The time of the model building depends on the number of points, N , as $N \log N$ in contrast to simple RBF's implementations with $O(N^3)$ efficiency;
- An iterative algorithm (successive layers correct the errors of the previous ones) creates a robust model, even with a very large initial radius. Successive layers have smaller radii and correct the inaccuracy introduced by the previous layer;
- The multi-layer model allows for control smoothing both by changing the regularisation coefficient and by a different number of layers.

Figure 2. RBF-ML algorithm operation. Residues from successive layers are passed to the next iteration with a radius twice as small.

6. Experimental Results

An example worth highlighting are the raw results of the archeological survey made in Płońsk, Poland, between Przejazd and 19 Stycznia streets at the old graveyard. The

measurements were obtained using a 600 MHz impulse GPR system with bow-tie antennas. The investigated area was divided into subfragments, and the scans were realised on paths every 5 or 40 cm depending on terrain conditions. This gave about 200 scans for each subfragment. A drawing of one of the subfragments and an example of the one-path scanning result (B-scan) is shown in Figure 3.

Figure 3. Measurement plan and example of the one-path scanning result (B-scan). Red lines indicate scanning directions.

Every measurement point contains 256 numbers describing GPR echoes from subsequent depths (A-scan). The set of subsequent A-scans makes the B-scan. To each of the A-scan numbers, the spatial coordinates x, y are added to obtain a 3D array. This makes cross-sections called C-scans. To illustrate the operation of the proposed method, one C-scan from a 1 m depth was made.

In order to use the hierarchical algorithm, three parameters must be chosen: the initial radius, R_0 , the number of layers, N_L , in the model and the regularisation coefficient, λ . The main parameter affecting the quality of the approximation is the initial radius. Figure 4 shows the set of C-scans of 1 m depth for six different radius values. The other calculation parameters are $N_L = 3$, $\lambda = 0$, and the step of the mesh, which is equal to 5 cm.

Figure 4. C-scans for different radius values.

In the two first C-scans ($R_0 = 5$ and 10 cm), the empty lines along the scanning paths can be observed. An approximation of the areas between the paths starts properly from a radius value of $R_0 = 30$ cm because it is the minimal radius value for which the empty areas can be approximated with points coming from two adjacent paths. A further increase in radius sharpens the image and enhances detail, but it does not change the shape of the objects. For $R_0 \geq 50$ cm, these changes are almost imperceptible.

The value of the approximation radius decisively affects the calculation time. Figure 5 shows the calculation times of the model and function evaluation in the mesh nodes depending on R_0 . Obviously, the computation time of the points on the mesh depends on its density and is proportional to the number of mesh nodes. All calculations were made for a rectangular mesh with equidistant nodes in the x and y directions, with a step of 5 cm. Calculations were performed on a computer with Intel Core i5-7200U CPU 2.5 GHz, 8 GB RAM on a Windows 10 operating system.

Figure 5. Time of model construction (model) and evaluation (mesh) depending on the initial radius. M indicates the number of approximation centres.

The computation time increases linearly in intervals with increasing radius. For radii at which there is an increase in computational speed, the number of points analysed decreases. This is because the hierarchical RBF model removes excessive (too dense) nodes from the model. The support radius parameter R_{SUP} is used to evaluate which points need to be removed and which do not. If two points are less than $R_{SUP} \cdot R_{CURRENT}$ units of distance apart, one of them is removed from the model. The larger the support radius, the faster the model construction and evaluation. However, values that are too large cause models to be “bumpy.” Recommended values for the support radius coefficient are in the range of 0.1–0.4, with 0.1 being the default value.

In the analysed case, where the distances between adjacent B-scans are 40 cm, the corresponding value of the initial radius is approximately 50 cm.

The second important parameter included in the model is the number of layers. Table 2 shows the effect of the number of layers on the approximation errors for the three R_0 values. Example C-scans for $R_0 = 100$ cm are shown in Figure 6.

Table 2. The influence of the number of model layers on the approximation quality.

Layer Number	50 cm			100 cm			200 cm		
	E_{RMS}	E_{MAX}	M	E_{RMS}	E_{MAX}	M	E_{RMS}	E_{MAX}	M
1	4.93	40.68	2474	15.82	81.39	1682	25.17	137.13	1001
2	3.38	33.64	7316	4.94	40.63	3440	15.89	82.96	2122
3	1.98	15.34	12,158	3.38	33.61	8282	4.49	40.59	4138
4	0.00	0.00	17,000	1.98	15.34	13124	3.38	33.65	8980
5	0.00	0.00	21,842	0.00	0.00	17966	1.98	15.36	13,822
6	0.00	0.00	26,684	0.00	0.00	2208	0.00	0.00	18,664

Figure 6. C-scans for different numbers of model layers.

With the increase in the number of layers, we observe a drop in approximation errors down to zero, but this is done at the expense of the computation complexity (increase in M). By choosing larger initial radii, more layers are needed to obtain an accurate surface representation.

The optimal number of layers can be estimated from the relation $N_L = \text{round}[\ln(2R_0/\bar{r})/\ln(2)] + 2$, where $\bar{r}$ is the average distance between consecutive points in the B-scan. This selection ensures that the model is flexible enough to accurately reproduce the input data.

Another parameter that occurs in the hierarchical RBF model is the regularisation coefficient λ , which can take values of ≥ 0 . This parameter adds controllable smoothing to the problem, which may reduce noise. Specification of a non-zero lambda means that in addition to fitting error solver also minimise $\lambda[S''(\mathbf{x})]^2$, where $S''(\mathbf{x})$ is the second derivative of squared error S . Specification of an exactly zero value means that no penalty is added. The optimal lambda is problem-dependent and requires trial and error. The recommended λ starting value is in the of range 10^{-5} to 10^{-6} , which corresponds to a slightly noticeable smoothing of the function. A value of 0.01 usually means that strong smoothing is being used.

Calculation of nonlinearity penalty is costly: it results in a several-fold increase in the model construction time, but the evaluation time remains the same. In the example, the model building time increased by about 2.5-fold compared to the calculations for $\lambda = 0$

and did not depend on the value of $\lambda > 0$. However, since function smoothing can also be achieved by reducing the number of model layers, which reduces the computation time, the second solution is preferable.

7. Comparison of Gaussian CS-RBF with Other Approximation Methods

There are several methods of assessing the quality of the approximating function. The simplest is the root mean square error over N evaluation points x , as follows:

$$E_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{j=1}^N [f(x_j) - h(x_j)]^2} \quad (11)$$

where $f(x)$ is the evaluated function and $h(x)$ is a measurement value.

The second is leave-one-out-cross-validation (LOOCV) [23]. In practical problems of scattered data interpolation, the exact solution is most likely unknown. In such situations, it is not possible to calculate the exact RMS error. Cross-validation is a statistical approach. One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (training set), and validating the analysis on the other subset (validation set or testing set). To reduce variability, in most methods multiple rounds of cross-validation are performed using different partitions, and the validation results are combined (e.g., averaged) over the rounds to provide an estimate of the model's predictive performance [24].

Another evaluation method is benchmark tests that sample and reconstruct test functions, such as Franke's function [25].

GRP scans are a specific set of data for which we can determine an exact approximating function along the scan path, but large errors between paths are to be expected. LOOCV is not a good method for evaluating the quality of the approximating function since both training set and testing set will come from the scan path. Therefore, a custom test function that contains a point object and three longitudinal objects of different orientation and width was created (Figure 7). This function is described by the following formula:

$$f(x, y) = \sum_{i=1}^4 a_i \exp \left[\frac{(b_{1i}x + c_{1i})}{d_{1i}} + \frac{(b_{2i}y + c_{2i})}{d_{2i}} \right] \quad (12)$$

with the following coefficients:

i	a	b_1	c_1	d_1	b_2	c_2	d_2
1	0.75	60	−5	−4	60	−5	−4
2	0.75	100	−20	−10	9	−6	−10
3	0.5	9	−7	−4	30	−8	−4
4	0.2	9	−4	−11	9	−7	−1

For this test function, different RBF approximation methods available in Surfer [9] were compared. Model building was performed for a dataset typical of a B-scan GPR containing 4785 measurement points. Model evaluation was performed on the grid $x = 0, \dots, 1$, $y = 0, \dots, 1$, with a step of 0.02. Calculations were performed for different approximation radii. The E_{RMS} decreased with increasing radius to a value of around $R_0^2 = 2 \times 10^{-4}$, after which further increasing the radius slightly affected the quality of the test function representation. For such a radius, all E_{RMS} values were in the range of 2.1×10^{-3} to 2.5×10^{-3} , while E_{MAX} was in the range of 0.036–0.044. The model building times for compactly supported Gaussian RBFs and other RBFs implemented in Surfer are shown in Table 3.

Figure 7. Test function (12) used to evaluate the quality of interpolation of GPR data.

Table 3. Model building times for different RBFs at $R_0^2 = 0.002$.

RBF	CS Gaussian	Multiquadric	Inverse Multiquadric	Multilog	Natural Cubic Spline	TPS
t [s]	3.0	140.1	126.6	133.0	140.0	142.9

A similar analysis was performed for other models available in the Surfer package. Table 4 shows the model building times and model estimation errors according to increasing E_{RMS} . The calculations were performed for the default parameters, set in the Surfer program, because their changes did not bring significant improvement in the quality of approximation.

Table 4. Model building times and estimation errors of the test function for Gaussian CS-RBF and other models included in the Surfer program.

Model	Gaussian CS-RBF	Minimum Curvature	Triangulation with Linear Interpolation	Kriging	IDP –Inverse Distance to a Power	Natural Neigh-Bor	Nearest Neigh-Bor	Local Polynomial
$t \times 10^3$ [s]	3.0	0.14	0.02	2.9	0.18	1.1	0.02	0.27
E_{RMS}	2.1	2.4	5.6	6.7	22	14	22	48
E_{MAX}	44	33	99	98	266	307	355	534

Based on the results obtained, it can be concluded that the Gaussian CS-RBF approximation provides a representation quality of test function comparable to the minimum curvature (CS) method and significantly better than the other methods implemented in Surfer. However, the model building time is clearly longer than for most of these methods. In contrast, within the radial basis function approximation (Table 3), Gaussian CS-RBFs provide almost 50 times faster computation speed than for global RBFs.

8. Conclusions

This paper investigates the feasibility of using RBF approximation to create C-scans in GPR measurements. For this purpose, hierarchical approximation with compactly supported Gaussian RBF was used. The method is a part of the linear algebra library ALGLIB [12]. The algorithm is optimised in terms of computation time and memory usage, which gives the ability to handle millions of points on an ordinary PC.

Calculations were carried out for real data, including c.a. 5000 measurement points, and the optimal input parameters were selected. It was found that for these parameters

the time needed to generate an image of one C-scan layer was about 5 s on a common computer. This is a relatively long time; however, it analyses data that often comes from several hours of field measurements. In addition, this time can be significantly reduced by settling for an image of slightly lower quality.

The use of Gaussian CS-RBFs enables multiple reductions in computation time compared to global RBFs with the same quality of surface representation. Compared to other commonly used methods, Gaussian CS-RBFs provide a very good approximation of the specific datasets that are obtained from GPR measurements. This is carried out at the cost of slightly higher computation time.

The presented hierarchical RBF approximation algorithm is one of the most efficient algorithms for processing large, scattered data sets. Its implementation, located in the ALGLIB library, enables simple software development, which can be successfully used to analyse GPR images.

Author Contributions: Conceptualization, M.P., K.J. and W.M.; methodology, K.J. and M.G.; software, K.J.; validation, M.P., W.M. and J.B.; formal analysis, K.J.; investigation, K.J., M.P. and W.M.; resources, K.J., M.G. and M.P.; data curation, K.J., M.P. and M.G.; writing—original draft preparation, K.J., M.P. and J.B.; writing—review and editing, K.J.; visualization, K.J. and M.P.; supervision, M.P.; project administration, M.P.; funding acquisition, M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data available on request from the authors.

Acknowledgments: This work was financed by Military University of Technology under research project UGB 22-841 and UGB 22-852.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Jing, M.; Wu, J. Fast image interpolation using directional inverse distance weighting for real-time applications. *Opt. Commun.* **2013**, *286*, 111–116. [CrossRef]
- Chen, C.; Zhao, N.; Yue, T.; Guo, J. A generalization of inverse distance weighting method via kernel regression and its application to surface modeling. *Arab. J. Geosci.* **2015**, *8*, 6623–6633. [CrossRef]
- van Kreveld, M. Digital elevation models and TIN algorithms. In *Algorithmic Foundations of Geographic Information Systems*; CISM School 1996; Lecture Notes in Computer Science; van Kreveld, M., Nievergelt, J., Roos, T., Widmayer, P., Eds.; Springer: Berlin/Heidelberg, Germany, 1997; Volume 1340, Chapter 3. [CrossRef]
- Liang, S.; Wang, J. Geometric processing and positioning techniques. In *Advanced Remote Sensing*, 2nd ed.; Academic Press: Cambridge, MA, USA, 2020; Chapter 2; pp. 59–105. ISBN 9780128158265. [CrossRef]
- Ledoux, H.; Gold, C. An Efficient Natural Neighbour Interpolation Algorithm for Geoscientific Modelling. In *Developments in Spatial Data Handling*; Springer: Berlin/Heidelberg, Germany, 2005; Chapter 8; pp. 97–108. [CrossRef]
- Agarwal, P.K.; Beutel, A.; Mølhave, T. TerraNNI: Natural neighbor interpolation on 2D and 3D grids using a GPU. *ACM Trans. Spat. Algorithms Syst.* **2016**, *2*, 1–31. [CrossRef]
- Lodha, S.K.; Franke, R. Scattered Data Interpolation: Radial Basis and Other Methods. In *Handbook of Computer Aided Geometric Design*; Farin, G., Hoschek, J., Kim, M.-S., Eds.; North-Holland: Amsterdam, The Netherlands, 2002; Chapter 16; pp. 389–404. ISBN 9780444511041. [CrossRef]
- Papritz, A.; Stein, A. Spatial prediction by linear kriging. Spatial Statistics for Remote Sensing. In *Remote Sensing and Digital Image Processing*; Stein, A., Van der Meer, F., Gorte, B., Eds.; Springer: Dordrecht, The Netherlands, 2002; Volume 1, Chapter 6; pp. 83–113. ISBN 0-7923-5978-X. [CrossRef]
- Barmuta, P.; Gibiino, G.P.; Ferranti, F.; Lewandowski, A.; Schreurs, D. Design of Experiments Using Centroidal Voronoi Tessellation. *IEEE Trans. Microw. Theory Tech.* **2016**, *64*, 3965–3973. [CrossRef]
- A Basic Understanding of Surfer Gridding Methods. Available online: <https://support.goldensoftware.com/hc/en-us/articles/231348728-A-Basic-Understanding-of-Surfer-Gridding-Methods-Part-1> (accessed on 15 September 2021).
- Comparing Leapfrog Radial Basis Function and Kriging. Available online: <https://www.seequent.com/comparing-leapfrog-radial-basis-function-and-kriging/> (accessed on 15 September 2021).
- TerrSet 2020. Geospatial Monitoring and Modeling System. Available online: https://clarklabs.org/wp-content/uploads/2020/05/TerrSet_2020_Brochure-FINAL27163334.pdf (accessed on 15 September 2021).
- ALGLIB. Sergey Bochkanov. Available online: www.alglib.net (accessed on 25 November 2021).
- Wendland, H. Computational aspects of radial basis function approximation. *Stud. Comput. Math.* **2006**, *12*, 231–256. [CrossRef]

15. Skala, V. Fast interpolation and approximation of scattered multidimensional and dynamic data using radial basis functions. *WSEAS Trans. Math.* **2013**, *12*, 501–511.
16. Majdisova, Z.; Skala, V. Big geo data surface approximation using radial basis functions: A comparative study. *Comput. Geosci.* **2017**, *109*, 51–58. [[CrossRef](#)]
17. Majdisova, Z.; Skala, V. Radial basis function approximations: Comparison and applications. *Appl. Math. Model.* **2017**, *51*, 728–743. [[CrossRef](#)]
18. Skala, V. RBF Approximation of Big Data Sets with Large Span of Data. In Proceedings of the 2017 Fourth International Conference on Mathematics and Computers in Sciences and in Industry (MCSI), Corfu, Greece, 24–27 August 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 212–218. [[CrossRef](#)]
19. Skala, V. High dimensional and large span data least square error: Numerical stability and conditionality. *Int. J. Appl. Phys. Math.* **2017**, *7*, 148–156. [[CrossRef](#)]
20. Brown, D.; Ling, L.; Kansa, E.; Levesley, J. On Approximate Cardinal Preconditioning Methods for Solving PDEs with Radial Basis Functions. In *Engineering Analysis with Boundary Elements*; Leitao, V., Ed.; Elsevier: Amsterdam, The Netherlands, 2005; Volume 29, pp. 343–353. ISSN 0955-7997. [[CrossRef](#)]
21. Smolik, M.; Skala, V. Large scattered data interpolation with radial basis functions and space subdivision. *Integr. Comput. Aided Eng.* **2018**, *25*, 49–62. [[CrossRef](#)]
22. Paige, C.C.; Saunders, M.A. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.* **1982**, *8*, 43–71. [[CrossRef](#)]
23. Mishra, P.K.; Nath, S.K.; Sen, M.K.; Fasshauer, G.E. Hybrid Gaussian-cubic radial basis functions for scattered data interpolation. *Comput. Geosci.* **2018**, *22*, 1203–1218. [[CrossRef](#)]
24. Arlot, S.; Celisse, A. A survey of cross-validation procedures for model selection. *Statist. Surv.* **2010**, *4*, 40–79. [[CrossRef](#)]
25. Franke, R. *A Critical Comparison of Some Methods for Interpolation of Scattered Data*; Final Report; Defense Technical Information Center: Fort Belvoir, VA, USA, 1979; Available online: <https://calhoun.nps.edu/handle/10945/35052> (accessed on 15 April 2021).