Improvement of Some Interpolation Methods for Terrain Reconstruction from Scattered Data

Róbert Bohdal
Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics
Mlynská dolina
Slovakia, 842 48
Bratislava
robert.bohdal@fmph.uniba.sk

ABSTRACT

Using GPS modules it is easy to obtain 3D data for areas that have not been digitized yet. Such terrain data are usually not arranged in a grid, and therefore we have to use scattered data interpolation methods. The aim of the paper is to create a digital terrain model from 3D data using modifications of known methods. Sibson interpolation method is often used when we need to interpolate large data sets. This method has low memory requirements, it is sufficiently fast, but creates undesired surface artefacts. Our aim is to have the resulting interpolation surface as similar as possible to the original surface. We have decided to replace the heights at specified points by local functions. We use biquadratic and bicubic polynomials, Hardy's multiquadrics and thin plate spline as local functions. In the paper, we have evaluated the time requirements and the accuracy with which the interpolated area matches the actual 3D data on 2 terrain samples (the Little Carpathians and a small part of the Little Carpathians).

Keywords

Digital terrain model, Radial basis functions, Inverse distant weights, Thin plate spline, Natural neighbours, Sibson interpolation

1 INTRODUCTION

Digital terrain model (DTM) can be used in many areas and applications. It is commonly used in urban planning, hydrology, geosciences, for investigating soil erosion, modelling of movement of avalanches, army applications, graphics information systems, creation of topographic maps and similarly. DTM can be understood as a 3D representation of a part of the Earth surface in digital format. It is commonly created from a large amount of 3D points in the form of surfaces, which are created using interpolation and approximation functions. These 3D points are obtained, for example, using *stereophotogrammetry*, *laser scanning* and *radargrametry*.

Creating of digital terrain from scattered data is still an interesting area of research. This is suggested by several comparative studies that we can find currently in literature.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

In our research, we have decided to use local interpolants, which replace the specified height points that are used in creating the terrain model. We want to improve the accuracy of methods using the weighted average and remove shape artefacts that these methods produce in the final models. We want to take advantage of their algorithmic simplicity and their speed of calculation. In conclusion, we show that the use of local functions is meaningful, even at the cost of slightly increased computational time. We consider *radial basis functions* (RBF) to be the best local functions, which are currently used in many areas of research.

2 RELATED WORK

There are several methods that can be used to create a DTM from scattered data. In most cases interpolation methods are used. It is then possible to calculate the height value at any point of the considered area.

The most commonly used methods are based on triangulation entry points, the weighted average methods, methods using radial basis functions and others. Less-known techniques use, for example, dividing the input data into areas using local interpolants or neural networks. There are also approaches that use a combination of several methods.

For the methods using *triangular irregular networks* (TIN), one first constructs a suitable triangulation from the input data, for example, *Delaunay triangulation*. Methods which use a piecewise continuous interpolant can be used afterwards, for example, Bezier patches in case of *Clough-Tocher* [Hug04] or *Powell-Sabin* [PS77] method. Further, one can use a triangular network method called *triangle based blending* (TBB) [GS13, DG02, Ami02], which uses the weighted average of three functions from a preselected class of local functions interpolating the corresponding vertex of the triangle and its nearest neighbours. Triangular network is also used by a method described earlier [HZDS01], in which one first constructs a uniform triangular network, and local approximation polynomials (Bezier triangular patches) are calculated using the method of least squares. The resulting spline function is then created using a combination of these polynomials with the use of Bernstein-Bezier smoothness conditions. Among the methods using triangular networks, we can also use a method of the *natural neighbours* (Sibson's interpolant) [DG02], which uses Voronoi diagram, which is the dual graph to the Delaunay triangulation. The function value (height at the searched point) is given by the weighted average of local height values of the relevant vertices.

Among the methods using only the weighted average, *Shepard's method* also known as the *Inverse Distance Weighting* (IDW) is the most famous [AAAC05, GG13, GS13, Hu13, Hug04]. Since Shepard's method does not give good results, its modification is used more often. The modification uses local interpolant calculated by the method of least squares [GS13].

To create a model of the terrain, RBF methods are probably the ones used the most often [AAAC05, CL12, CL13, GG13, GS13, Hu13, Hug04, MS16, PGTG04, SS09]. *Thin plate splines* (TPS) and *Hardy's multiquarics* (HMQ) are the most famous from this class of functions. However, the disadvantage of these methods is that for their calculation it is necessary to solve a system of equations. If the number of input points is big, we use methods that produce a final interpolation surface using a local interpolant. In [PGTG04], there is a procedure which in the first step recursively splits the input area into an overlapping sub-regions using k-d trees. The second step is calculating the functional value as a weighted average of two functions recursively enumerated in the respective sub-regions. For a large number of data points, we can use a RBF approximation given in [MS16]. The method uses a determination of significantly fewer so-called referent points, which together with the given points create an overdetermined system of equations. This system of equations is then solved by the method of least squares to obtain the unknown coefficients of the resulting interpolation function.

For the needs of creating a model of the terrain, we often use a geostatistics method called *Kriging* [GS13, Hug04]. It is based on predicting the value of a function at a given point using the weighted average of points in the neighbourhood of the calculated point.

From less-known methods for the terrain construction, it is necessary to mention also neural networks of type MLP, *Support Vector Machine Regression* and *Neural Networks* in [ON15] or genetic algoritms in [BSS14].

3 METHODS

The creation of a digital terrain model from scattered data points can be easily solved using suitable interpolation methods. In our case, we have focused on the modification of known methods, using local interpolants, which are used instead of the height values. As local interpolants, we choose thin plate splines and Hardy's multiquadrics [Isk03] and also well-known cubic and quadratic polynomials of two variables. As a further option, we choose the replacement of height values by planes, while their normal vector is calculated as a gradient of the local interpolation function.

Let us have a set $\mathscr{P}$ of N mutually different input points $\mathscr{P} = \{\mathbf{p}_1[p_1^x, p_1^y], \ldots, \mathbf{p}_N[p_N^x, p_N^y] \mid \mathbf{p}_i \in \mathbb{R}^2\}$ with height values $h_i \in \mathbb{R}$, for $i = 1, \ldots, N$. We search for such function $f : \mathbb{R}^2 \to \mathbb{R}$, for which the interpolation condition is true:

$$f(\mathbf{p}_i) = h_i,\ i = 1, \ldots, N. \tag{1}$$

3.1 Inverse Distance Weighted (IDW)

The simplest form of IDW interpolation is called Shepard's method. Shepard defined his interpolating function $f(\mathbf{x})$ with argument $\mathbf{x} \in \mathbb{R}^2$ to be the weighted average of the heights h_i [HL93]:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \omega_i(\mathbf{x}) h_i. \tag{2}$$

Weight functions $\omega_i(\mathbf{x})$ from formula (2) can be expressed as:

$$\omega_i(\mathbf{x}) = \frac{\sigma_i(\mathbf{x})}{\sum_{j=1}^{N} \sigma_j(\mathbf{x})},$$

where $\sigma_i(\mathbf{x}) = ||\mathbf{x} - \mathbf{p}_i||^{-\mu_i}$, for $\mu_i > 0$. The parameter μ_i allows to control the shape of the final surface in the neighbourhood of the interpolated points. The standard value for this parameter is $\mu_i = 2$.

The global character of this method can be made local by multiplying the weighted function $\omega_i(\mathbf{x})$ by the mollifying function [HL93]:

$$\lambda_i(\mathbf{x}) = \left(1 - \frac{\sigma_i(\mathbf{x})}{R_i}\right)_+^{\mu_i}, \text{ where } R_i > 0.$$

For example, we can set the radius R_i to $\frac{D}{2}\sqrt{\frac{N_w}{N}}$, where D is the maximum distance between arbitrary two points of the set $\mathscr{P}$ and $N_w = 19$ [TH10].

3.2 Radial Basis Functions (RBF)

Radial basis functions have gained immense popularity in the multi-dimensional interpolation of scattered data. They are simple to implement, and they generate an interpolation surface with a sufficient smoothness.

We can write the interpolation function $f(\mathbf{x})$ in the following form [HL93]:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i R(\|\mathbf{x}-\mathbf{p}_i\|) + \sum_{k=1}^{l} c_k \Phi_k(\mathbf{x}), \quad (3)$$

where $\Phi_k(\mathbf{x}) \in \pi_m^2,\ l = \dim(\pi_m^2) = \binom{m-1+2}{2}$. Symbol π_m^d denotes a linear space containing all polynomials over the field $\mathbb{R}$ with d variables and a degree at most $m-1$. Functions $R(\|\mathbf{x}-\mathbf{x}_i\|)$ are radial basis functions with an argument expressing the euclidean distance between points $\mathbf{x}$ and $\mathbf{x}_i$.

Unknown coefficients $\boldsymbol{\lambda} = (\lambda_1,\ldots,\lambda_N)^\mathsf{T}$ and $\mathbf{c} = (c_1,\ldots,c_l)^\mathsf{T}$ in relation (3) are given by solving a system of equations:

$$\begin{pmatrix} \mathbf{A} & \mathbf{P} \\ \mathbf{P}^\mathsf{T} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{h} \\ \mathbf{0} \end{pmatrix}, \quad (4)$$

where $\mathbf{A}_{i,j} = R(\|\mathbf{p}_i - \mathbf{p}_j\|)$, $\mathbf{P}_{i,k} = \Phi_k(\mathbf{p}_i)$ and $\mathbf{h} = (h_1,\ldots,h_N)$, for $i,j = 1,\ldots,N$ and $k = 1,\ldots,l$. Due to the fact that both RBFs described below are conditional positive definite [Fas07], the system of equations (4) has a solution if the points $\mathbf{x}_i$ are non-collinear.

3.2.1 Thin Plate Splines (TPS)

Thin plate splines belong to the class of polyharmonic splines:

$$R_{d,m}(\|\mathbf{x}\|) = R_{d,m}(r) = \begin{cases} r^{2m-d} & \text{if } d \text{ is odd} \\ r^{2m-d}\log(r) & \text{if } d \text{ is even} \end{cases}$$

The name is derived from a relation, in which we search for the minimum of an integral describing the distribution of so-called bending energy on an infinitely thin elastic plate. According to [Isk03], it is possible to write the interpolation function in the form:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i R_{d,m}(\|\mathbf{x}-\mathbf{p}_i\|) + \sum_{|\boldsymbol{\alpha}|<m} c_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}, \quad (5)$$

where $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_d)$ is so-called *multi-index* and $\mathbf{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1}\cdots x_d^{\alpha_d}$, $|\boldsymbol{\alpha}| = \alpha_1 + \ldots + \alpha_d$, $\alpha_k \in \mathbb{N}_0^d$. After substituting $d = m = 2$ (dimension of space $\mathbb{R}^2$) we get a standardly used interpolation function:

$$f(\mathbf{x}) = f(x,y) = \sum_{i=1}^{N} \lambda_i r_i^2 \log(r_i) + c_1 + c_2 x + c_3 y, \quad (6)$$

where $r_i = \|\mathbf{x}-\mathbf{p}_i\| = \sqrt{(x-p_i^x)^2 + (y-p_i^y)^2}$.

3.2.2 Hardy's Multiquadrics (HMQ)

This method is very similar to the previous method, but it uses different RBFs, and for $d = 2$ it does not have a polynomial term. For our interpolation problem, we get the following interpolation function:

$$f(\mathbf{x}) = f(x,y) = \sum_{i=1}^{N} \lambda_i \sqrt{r_i^2 + c^2}. \quad (7)$$

Value c changes the shape of the resulting interpolation surface. In general, a smaller value of the parameter c creates so-called "sharp extremes" in the graph of the function, while its greater value "smoothes" the function. In literature, there are several ways of how to suitably choose it [HL93]:

- $c = 0.815d$, where d is the average distance between the points $\mathbf{p}_i$ of set $\mathscr{P}$ to their closest neighbours,
- $c = 1.25\frac{D}{n}$, where D is the average of the smallest circle, which contains all points of the set $\mathscr{P}$,
- $c = \sqrt{\frac{1}{10}\max_{i,j}\|\mathbf{p}_i - \mathbf{p}_j\|}$,
- $c = \sqrt{\frac{3}{5}\min_{i,j}\|\mathbf{p}_i - \mathbf{p}_j\|}$.

3.3 Triangle Based Blending (TBB)

This method belongs to a group of methods that use triangular irregular network $\mathscr{T}$ created from given points $\mathbf{p}_i$. Delaunay's triangulation is the most common method because it maximizes the minimum angle of triangles. There is a large number of optimal algorithms that construct this triangular net with $\mathscr{O}(n\log n)$ complexity. We can find one of these approaches in [BDH96]. First, we calculate for each point $\mathbf{p}_i$ from set $\mathscr{P}$ a biquadratic polynomial interpolating this point and its five nearest neighbours:

$$\begin{aligned} f_i(x,y) =& a_1(x-p_i^x)^2 + a_2(x-p_i^x)(y-p_i^y) + \\ & a_3(y-p_i^y)^2 + a_4(x-p_i^x) + a_5(y-p_i^y) + h_i. \end{aligned}$$

The unknown coefficients $a_1,\ldots,a_5$ are calculated from the condition that interpolates all 6 points. If we need to calculate the value of the height h for the point $\mathbf{x}[x,y]$ we have to find in which triangle Δijk of the triangulation $\mathscr{T}$ this point lies. Then the height h is calculated as the weighted average of three values of the corresponding local functions [Ami02]:

$$f(\mathbf{x}) = f(x,y) = w_i f_i(x,y) + w_j f_j(x,y) + w_k f_k(x,y), \quad (8)$$

where i,j,k are indices of vertices of triangle Δijk with vertices $\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k$ (see Figure 1). In Figure 1, black stars denote vertices, from which local interpolant $f_i(x,y)$ corresponding to the vertex $\mathbf{p}_i$ is calculated. Similarly, blue circles denote vertices giving interpolant $f_j(x,y)$

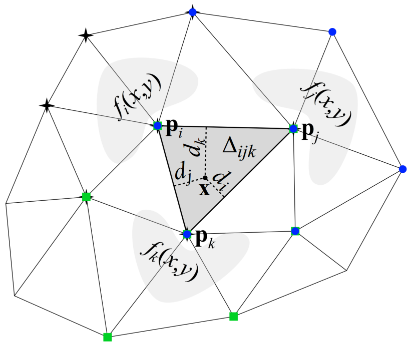


Figure 1: Local function is calculated by one of the triangle vertices and its adjacent vertices.

and green squares denote vertices creating local interpolant $f_k(x,y)$.

Smooth continuous transition between two triangles can be guaranted by calculating weights using relation:

$$w_i = d_i^r/(d_i^r + d_j^r + d_k^r),$$

where appropriate value for r is $r = 2$ or $r = 3$ and lengths d_i, d_j, d_k can be determined by using barycentric coordinates of the point $\mathbf{x}[x,y]$ of the triangle Δijk.

3.4 Natural Neighbours (NN)

This interpolation method belongs to the weighted average methods. To calculate the unknown height h at point $\mathbf{x}$, it uses a Voronoi diagram which can be constructed very effectively from Delaunay's triangulation by an algorithm with complexity $\mathcal{O}(n)$ [LH10]. We can simply say that a Voronoi diagram is the union of all Voronoi cells defined by the description:

$$\mathcal{V}(\mathbf{p}_i) = \{\mathbf{x} \in \mathbb{R}^2 \mid \|\mathbf{x} - \mathbf{p}_i\| < \|\mathbf{x} - \mathbf{p}_j\| \; \forall j \neq i\},$$

where $\mathbf{p}_i \in \mathcal{P}$.

Let us call *natural neighbours* of a point $\mathbf{p}_i$ such points $\mathbf{p}_j$, whose Voronoi cells $\mathcal{V}(\mathbf{p}_j)$ have a common edge with Voronoi cell $\mathcal{V}(\mathbf{p}_i)$. It is also possible to extend the previous definition for an arbitrary point $\mathbf{x} \in \mathbb{R}^2$ by including the point $\mathbf{x}$ in the set of given points $\mathcal{P}$, and then we create a new Voronoi diagram. Natural neighbours of the point $\mathbf{x}$ are all its natural neighbours in the newly created Voronoi diagram (see Figure 2, the grey polygon represents a cell in the newly created Voronoi diagram).

As in the TBB method, the unknown height value h is calculated using the weighted average:

$$f(\mathbf{x}) = \frac{\sum_{i=1}^{n} a_i h_i}{\sum_{i=1}^{n} a_i}, \qquad (9)$$

where h_i are heights of n natural neighbours of the point $\mathbf{x}$ and weights a_i are areas of the polygon that are taken from the area of the original Voronoi cell $\mathcal{V}(\mathbf{p}_i)$ after including the point $\mathbf{x}$ into the set $\mathcal{P}$. Detailed description of the algorithm that calculates these weights

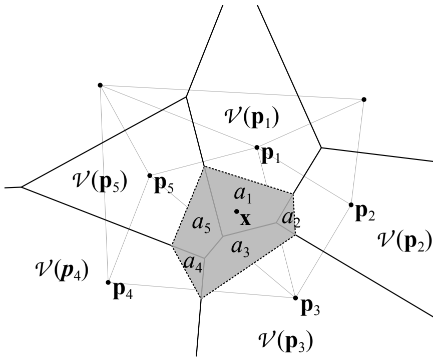


Figure 2: The original Voronoi diagram and the new Voronoi diagram which was created by including the point $\mathbf{x}$.

without creating the new Voronoi diagram can be found in [LH10].

3.5 Least Square Methods (LSM)

Methods of least squares create approximation surfaces that do not meet the interpolation condition (1). For our purposes, we use it to determine the gradient at the points $[\mathbf{p}_i, h_i] \in \mathbb{R}^3$. Bivariate polynomials are used the most often for terrain modelling using LSM:

$$f(\mathbf{x}) = f(x,y) = \sum_{k=0}^{m} \sum_{r+s=k} a_{rs} x^r y^s, \qquad (10)$$

where m is chosen polynomial degree ($m = 2$ or $m = 3$) and a_{rs} are unknown coefficients. To determine them, we need at least $(m+1)!$ given points $\mathbf{p}_i$. Unlike interpolation functions, which usually lead to solving a system of equations with a number of columns equal to the number of unknowns, in the method of least squares, we solve a system of equations with more equations than the number of unknowns. Consequently, the resulting surface cannot pass through the given entry points.

Let us calculate the bivariate polynomial for each point of $\mathbf{p}_i[p_i^x, p_i^y]$ and error e_i. Than we can create a system of equations:

$$h_i \approx \sum_{k=0}^{m} \sum_{r+s=k} a_{rs} (p_i^x)^r (p_i^y)^s + e_i, \; i = 1, \ldots, N, \qquad (11)$$

where $N \gg m$.

We search for such values of coefficients a_{rs} so that the following holds:

$$\sum_{i=1}^{N} e_i^2 \to 0$$

After rewriting the system of equations (11) into matrix form, we obtain:

$$\mathbf{h} \approx \mathbf{Pa} + \mathbf{e}. \qquad (12)$$

It is true that the sum of the squared errors $\sum_{i=1}^{N} e_i^2$ has a minimum for such vector of coefficients $\mathbf{a}$ that we can calculate using a system of normal equations [GR70]:

$$\mathbf{P}^\mathsf{T}\mathbf{h} = \mathbf{P}^\mathsf{T}\mathbf{P}\mathbf{a}$$

Vector of unknown coefficients **a** can be calculated from relation:

$$\mathbf{a} = (\mathbf{P}^\mathsf{T}\mathbf{P})^{-}\mathbf{P}^\mathsf{T}\mathbf{h},$$

where $(\mathbf{P}^\mathsf{T}\mathbf{P})^{-}$ is a pseudoinverse matrix to matrix $(\mathbf{P}^\mathsf{T}\mathbf{P})$. It is numerically convenient to use *singular value decomposition* (SVD) method while solving the system (12). We can find the SVD method, for example, in [GR70].

4 OUR APPROACH

Each of these interpolation methods has some drawbacks. RBF methods require solving systems of equations, which for a large number of input points results in high memory costs, very long calculation time and problems with numerical stability of the calculations. IDW methods, in addition to long calculation time, create unwanted artefacts (see Figure 7) in the shape of the resulting interpolation surface, which are present also in the TBB and NN methods.

Using local interpolation functions in this context is new. We have not found any study which uses local functions in interpolation methods for digital terrain model creation. In our comparison, we try to find such an interpolation method that has sufficient visual smoothness and does not suffer from shape artefacts. It should also have sufficient accuracy and a short time of calculation.

In this section, we give a procedure for finding close neighbours to a given point $\mathbf{p}_i$, which are necessary for constructing local interpolants. We will also introduce alterations to the methods using the weighted average, while we replace given height values h_i of points $\mathbf{p}_i$ by local functions $f_i(\mathbf{x})$.

4.1 Nearest Neighbours

Close neighbours of a point $\mathbf{p}_i$ can be found using Delaunay triangulation $\mathscr{T}$ created in-advance, because each vertex contains a pointer to all adjacent vertices when the triangulation is created. To determine local interpolants, we need to specify the minimum number of close neighbours of point $\mathbf{p}_i$. Without this condition, it is not possible to calculate all the unknown coefficients of local functions.

The procedure of finding these neighbours is shown in Figure 3. First, we find all adjacent vertices in the triangulation $\mathscr{T}$ to the vertex (point) $\mathbf{p}_i$, and we add them to the list of close neighbours. In Figure 3, they are marked by circles, and their index in the upper left corner has value 1 (level of the depth). If necessary, we add also the neighbouring vertices of these vertices to the list. They are marked by a star in the picture, and their index in the upper left corner shows depth value 2. We continue to the chosen level in this way. If the number of close neighbours does not achieve the necessary value, we find other vertices using the euclidean distance from vertex $\mathbf{p}_i$. In Figure 3, such vertices are labeled by green rectangles. To speed up the search by distance, we use a hash table in which all entry points $\mathbf{p}_i$ are assigned in advance.

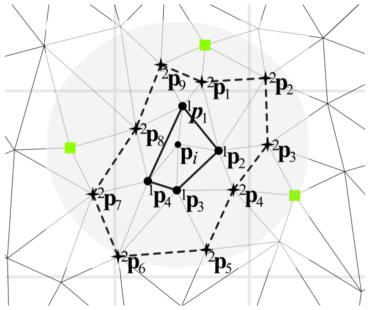


Figure 3: Selecting close neighbours of vertex $\mathbf{p}_i$ based on the neighbourliness and euclidean distance.

4.2 Modification of the Methods Using the Weighted Average

Replacing given height values by local interpolation functions allows the methods using the weighted average to make the resulting interpolation surface much more similar to the ideal (real) surface of the terrain.

Let us rewrite the expressions for interpolation functions $f(\mathbf{x})$ such that we use a local function $f_i(\mathbf{x})$ instead of the height h_i:

- We get the following expression for the IDW method (see relation (2)):

$$f(\mathbf{x}) = \sum_{i=1}^{N} \omega_i(\mathbf{x}) f_i(\mathbf{x}).$$

- The expression remains the same in the TBB methods (see relation (8)), but the original biquadratic polynomial is replaced by a general local function.
- For the method NN (relation (9)), we get the following expression:

$$f(\mathbf{x}) = \frac{\sum_{i=1}^{n} a_i f_i(\mathbf{x})}{\sum_{i=1}^{n} a_i},$$

  where n is number of natural neighbours of point $\mathbf{x}$.

4.3 Local Interpolants

At first, we choose thin plate splines (paragraph 3.2.1) and Hardy's multiquadrics (paragraph 3.2.2) as the local interpolants in our tests. Secondly, we use bivariate polynomial:

$$f_i(x,y) = \sum_{k=0}^{m} \sum_{r+s=k} a_{rs}(x - p_i^x)^r (y - p_i^y)^s + h_i$$

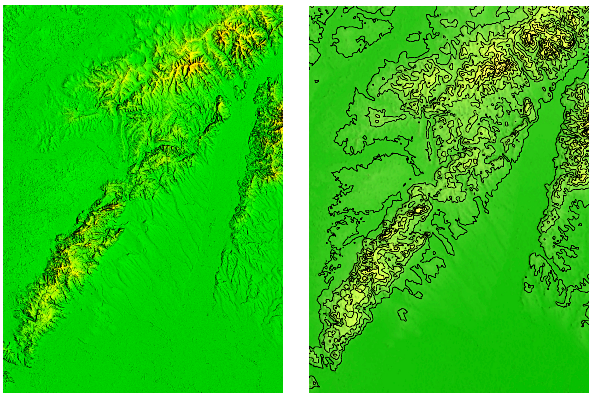


Figure 4: Original height map of the Little Carpathians is on the left, and the model created using TPS is on the right.

with $m = 2$ (biquadratic polynomial) and $m = 3$ (bicubic polynomial). Unknown coefficients a_{rs} are calculated from interpolation conditions $f_i(p_l^x, p_l^y) = h_l$, where $\mathbf{p}_l[p_l^x, p_l^y]$ are nearest neighbours of point $\mathbf{p}_i$.

Value of the height h_i at point $\mathbf{p}_i$ can be replaced by a relatively simple function of plane going through the point $\mathbf{p}_i[p_i^x, p_i^y, h_i]$ with expression:

$$f_i(x,y) = \frac{n_x}{n_z}(x - p_i^x) + \frac{n_y}{n_z}(y - p_i^y) + h_i,$$

while the normal vector $\mathbf{n}(n_x, n_y, n_z)$ is calculated from the gradient:

$$\mathbf{n}(n_x, n_y, n_z) = \left(\frac{\partial f(p_i^x, p_i^y)}{\partial x}, \frac{\partial f(p_i^x, p_i^y)}{\partial y}, -1 \right),$$

where $f(\mathbf{x})$ is any of the previously mentioned local functions, from which we can easily calculate the gradient.

5 TEST OF METHODS AND RESULTS

To test our modifications and compare different interpolation methods for creating digital terrain model, we have used a dataset of height points of the Little Carpathians obtained from the United States Geological Survey in SRTM format with a resolution of 1 arc second (30 meters). From this height map, we have created two files. The first contains the area region: $48.00°\text{N}, 17.00°\text{E} - 49.00°\text{N}, 18.00°\text{E}$, in figures and tables it is labelled with the name *Karpaty* (see Figure 4). The second file contains the area region: $48.15°\text{N}, 17.05°\text{E} - 48.20°\text{N}, 17.10°\text{E}$ labelled as *KarpatyCrop*.

For both files, we have created samples with $N = 2000$, 4000, 7000 and 10000 randomly selected points that were used to create the model of the terrain. We have also created a sample of $M = 20000$ test points to verify the accuracy of the model. We could not use a larger number of points for comparing interpolation methods because the TPS and HMQ methods require using matrices with a large number of nonzero elements. In this article, we present results only for a sample of $N = 10000$ points because of the limited space.

To create a digital terrain model, we have used not only all previously described methods, but also Powell-Sabin [PS77] and Clough-Tocher [Ami02] methods. However, we do not include these two methods in the results because we have not obtained for them an interpolation surface without unwanted artefacts, even though we have used the optimal normal vectors calculated from the gradient of the local TPS interpolation.

While evaluating the precision with which the model approximates the real terrain surface, we have used two statistical metrics:

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^{M} (f(x_j, y_j) - h_j)^2}{m}}$$

and

$$\text{Max Absolute Error} = \max_{j=1,\ldots,M} \{|f(x_j, y_j) - h_j|\}.$$

In addition to these metrics, we have been interested also in the visual smoothness, calculation time, memory demand and suitability for creating contours, which are used in topographic maps. Our results are shown in Table 1. Value *Accuracy rank* in the third column indicates the average rank of the given method, or group of methods (lower value is better). For a group of methods, we have always chosen the best candidate for the current sample of test points. Suitability sign "+/○/−" of accuracy is based on the accuracy rank. Similarly, suitability of the calculation time is based on the elapsed time in Table 2 and 3, memory demand is based on if large matrices are used in the algorithm and visual smoothness is decided visually using the obtained images (see Figure 7), depending on whether surfaces contain artefacts.

In the graphs, tables and pictures, we use the following abbreviations: lHMQ, lTPS, lQLS, lCLS denote using Hardy's multiquadrics, thin plate spline, biquadratic and bicubic polynomials as the local interpolant. Abbreviations gHMQ, gTPS, gQLS, gCLS denote using the relevant local functions while calculating the gradient of the tangent plane.

In the right part of Figure 4, we can see the terrain model of the Little Carpathians which has been calculated using the TPS method using 10000 points. This figure also demonstrates the suitability of this method for creating topographic maps with contour lines.

In Figure 7, we can see how using local functions in methods IDW, TBB and NN improves the shape of the resulting surface of the model of the terrain, and removes existing shape artefacts. In the top row, we can see at the same time the impact of improperly selected shape parameter in Hardy's multiquadrics, which leads to undesirable sharp points.

In Table 2 and Figure 5, we can see the evaluation of the accuracy of the resulting interpolation surfaces for data file *Karpaty*, and in Table 3 and Figure 6 for the data file *KarpatyCrop*. In the last column, we give in seconds the time necessary to calculate the height values for a grid 5463×8192 points. The evaluation time does not contain time for creating the hash table and the initial triangulation.

All methods have been tested on a desktop PC with Intel(R) Core(TM) i5-4670K CPU @3.40GHz processor with 8GB RAM.

Method	Accuracy	Accuracy rank	Calculation time	Memory demand	Visual smoothness
HMQ	+/−[1]	5.5	−	−	+/−[1]
TPS	+	5.3	−	−	+
NN	−	17.4	○/+	+	○
NN + plane	+	7.9	○/+	+	+
NN + local	+	4.1	○	+	+
IDW	−	22.3	−	+	−
IDW + plane	○	10.3	−	+	−
IDW + local	+	3.3	−	+	+
TBB	−	23.9	+	+	−
TBB + plane	○	12.5	+	+	−
TBB + local	+	4.9	+	+	○

Table 1: Suitability of using interpolation method. Symbol "+" represents suitability, "−" unsuitability and "○" average suitability of using a method.

6 CONCLUSION

We have shown that using local functions to the known methods (IDW, TBB and NN) for creating a digital terrain model significantly improves the visual smoothness of the resulting spline surface. It also increases the accuracy with which this surface approximates the actual surface of the terrain, and it suppresses undesired shape artifacts. With a suitable local function, we can even achieve results comparable with RBF methods, which have great memory and calculation requirements. At the same time, we have also pointed out that a wrong choice of the shape parameter in the HMQ method leads to a problematic surface shape.

The most appropriate method for creating the digital model, taking into account the computation time, accuracy, memory requirements and visual smoothness, is the method of Natural Neighbor with a local thin plate spline interpolant. As the second in order, we could use Triangle Based Blending method again with the local TPS interpolant, which is faster to calculate, but has worse visual smoothness.

An interesting finding is also the fact that the number of near vertices in LSM methods relates to the complexity of the terrain. For a rugged terrain, we have achieved better results when we used more points, and for a less rugged terrain when we used fewer points.

Method	Max Absolute Error	Mean Absolute Error	RMSE	Elapsed Time (s)
HMQ	184.857	10.726	20.199	6715
TPS	236.593	10.839	20.787	20930
NN - gQLS	185.654	11.221	21.090	95
NN - gCLS	204.537	11.570	21.838	107
NN - gHMQ	247.088	11.754	22.616	67
NN - gTPS	279.048	12.416	24.074	84
NN - lQLS	188.975	11.114	20.900	87
NN - lCLS	468.349	11.256	21.660	84
NN - lHMQ	205.663	10.781	20.332	86
NN - lTPS	210.575	10.732	20.495	171
IDW - gQLS	195.854	11.582	21.817	2541
IDW - gCLS	283.569	11.664	22.367	2532
IDW - gHMQ	259.551	11.762	22.775	2536
IDW - gTPS	298.943	12.436	24.255	3347
IDW - lQLS	203.549	11.669	21.666	2552
IDW - lCLS	500.426	20.895	39.873	2547
IDW - lHMQ	202.336	10.816	20.400	2616
IDW - lTPS	199.480	10.855	20.771	3603
TBB - gQLS	244.792	11.755	22.490	8
TBB - gCLS	271.387	11.918	22.991	7
TBB - gHMQ	257.001	11.992	23.461	8
TBB - gTPS	300.422	12.764	25.434	8
TBB - lQLS	245.993	11.565	22.098	8
TBB - lCLS	316.560	11.542	22.305	9
TBB - lHMQ	193.657	10.799	20.338	18
TBB - lTPS	201.006	10.780	20.562	21

Table 2: Accuracy and time of calculation for the tested methods for data file *Karpaty*.

7 FUTURE WORKS

In future work, we would like to focus on other ways to estimate parameter c, which occurs in some RBFs. In addition, we would like to verify the effect of selecting different neighbours on the accuracy of the interpolation surface and use compactly supported RBFs for calculation of a digital terrain model. Our testing algorithm has not used any accelerating techniques such as using parallelization or GPU, but it would be interesting to investigate the acceleration obtained using these techniques.

[1] Depends on the shape parameter c.

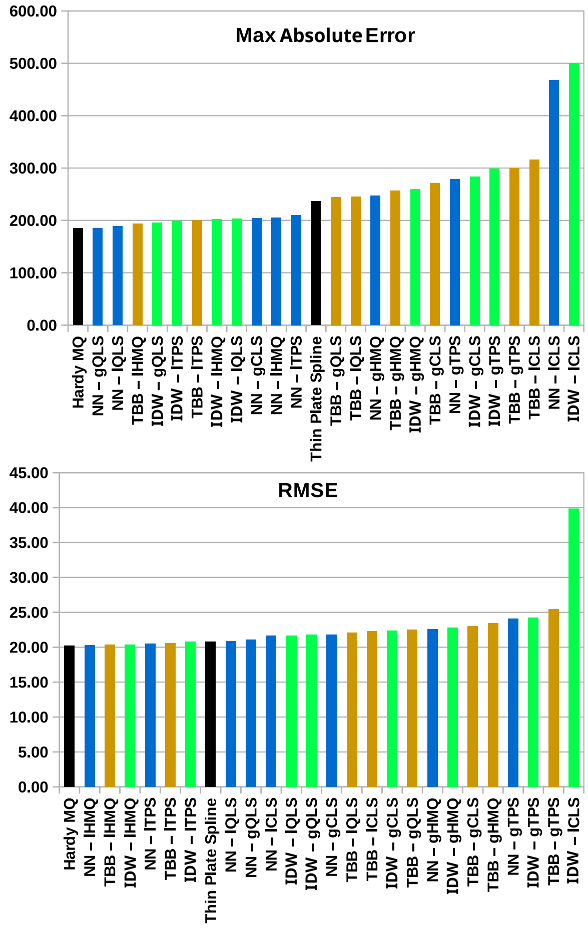


Figure 5: A plot showing the ranking of accuracy of individual methods for data file *Karpaty*.

Method	Max Absolute Error	Mean Absolute Error	RMSE	Elapsed Time (s)
HMQ	10.535	0.672	1.151	6603
TPS	8.817	0.583	0.968	26203
NN - gQLS	10.267	0.723	1.216	168
NN - gCLS	11.345	0.742	1.241	138
NN - gHMQ	13.767	0.650	1.094	76
NN - gTPS	14.911	0.665	1.128	79
NN - lQLS	10.171	0.742	1.255	171
NN - lCLS	10.163	0.677	1.142	281
NN - lHMQ	11.131	0.744	1.275	100
NN - lTPS	8.925	0.650	1.109	164
IDW - gQLS	12.914	0.790	1.311	2433
IDW - gCLS	11.689	0.757	1.242	2473
IDW - gHMQ	11.791	0.684	1.122	2441
IDW - gTPS	11.638	0.690	1.139	3131
IDW - lQLS	11.627	0.789	1.315	2443
IDW - lCLS	9.967	0.706	1.169	2479
IDW - lHMQ	10.552	0.763	1.283	2481
IDW - lTPS	8.666	0.654	1.093	3399
TBB - gQLS	11.778	0.754	1.274	8
TBB - gCLS	11.242	0.739	1.242	9
TBB - gHMQ	10.396	0.652	1.093	9
TBB - gTPS	11.587	0.675	1.128	8
TBB - lQLS	10.779	0.769	1.306	12
TBB - lCLS	9.515	0.702	1.185	11
TBB - lHMQ	10.557	0.747	1.277	20
TBB - lTPS	9.117	0.669	1.122	50

Table 3: Accuracy and time of calculation for the tested methods for data file *KarpatyCrop*.

8 REFERENCES

[AAAC05] Aguilar, F., Agüera, F., Aguilar, M., and Carvajal, F. Effects of terrain morphology, sampling density, and interpolation methods on grid dem accuracy. Photogrammetric Engineering & Remote Sensing, 71(7), pp.805-816, 2005.

[Ami02] Amidror, I. Scattered data interpolation methods for electronic imaging systems: a survey. Journal of electronic imaging, 11(2), pp.157-176, 2002.

[BDH96] Barber, B., Dobkin, D., and Huhdanpaa, H. The quickhull algorithm for convex hulls. ACM Transactions on Mathematical Software (TOMS), 22(4), pp.469-483, 1996.

[BSS14] Bagheri, H., Sadjadi, Y., and Sadeghian, S. Exploring the Role of Genetic Algorithms and Artificial Neural Networks for Interpolation of Elevation in Geoinformation Models, pages 107-121. Springer International Publishing, 2014.

[CL12] Chen, C., and Li, Y. A robust method of thin plate spline and its application to DEM construction. Computers & Geosciences, 48, pp.9-16, 2012.

[CL13] Chen, C., and Li, Y. A robust multiquadric method for digital elevation model construction. Mathematical Geosciences, 45(3), pp.297-319, 2013.

[DG02] Dakowicz, M., and Gold, C. Visualizing terrain models from contours - plausible ridge, valley and slope estimation. In Proceedings of the International Workshop on Visualization and Animation of Landscape, 2002.

[Fas07] Fasshauer, G. Meshfree approximation methods with MATLAB. vol. 6, World Scientific, 2007.

[GG13] Garnero, G., and Godone, D. Comparisons between different interpolation techniques. ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 1(3), pp.139-144, 2013.

[GR70] Golub, G., and Reinsch, C. Singular value decomposition and least squares solutions, volume 14, pages 403-420. Springer, 1970.

[GS13] Gumus, K., and Sen, A. Comparison of spatial interpolation methods and multi-layer neural networks for different point distributions on a digital elevation model. Geodetski vestnik, 57(3), pp.523-543, 2013.

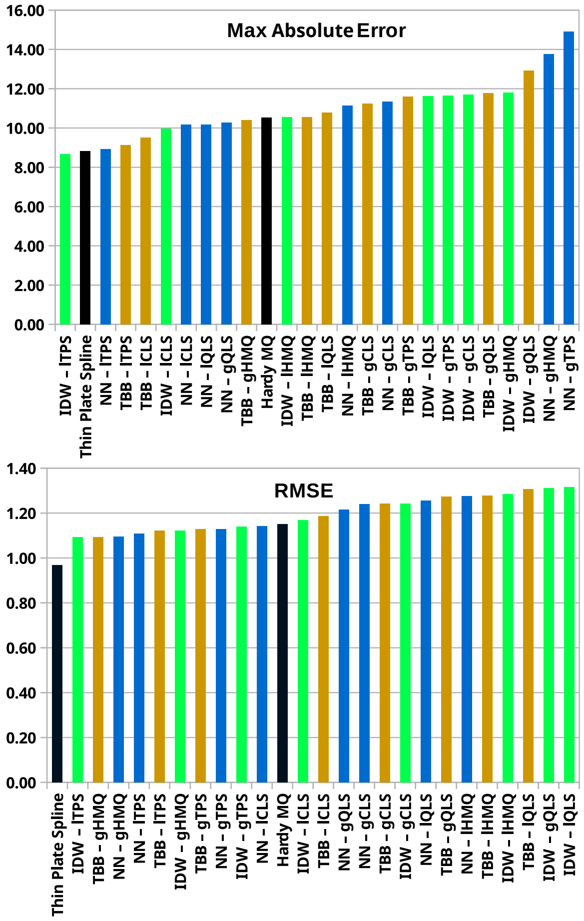


Figure 6: A plot showing the ranking of accuracy of individual methods for data file *KarpatyCrop*.

[HL93] Hoschek, J., and Lasser, D. Fundamentals of Computer Aided Geometric Design, pages 388-421. A K Peters, Wellesley, MA, 1993.

[Hu13] Hu, C. Comparison of the interpolation methods on digital terrain models. Master's thesis, Politecnico di Milano, 2013.

[Hug04] Hugentobler, M. Terrain Modelling with Triangle Based Free-Form Surfaces. PhD thesis, Mathematisch-naturwissenschaftlichen Fakultät der Universität Zürich, 2004.

[HZDS01] Haber, J., Zeilfelder, F., Davydov, O., and Seidel, H. Smooth approximation and rendering of large scattered data sets. In Proceedings of the Conference on Visualization '01, VIS '01, pp.341-348. IEEE Computer Society, 2001.

[Isk03] Iske, A. Radial basis functions: basics, advanced topics and meshfree methods for transport problems. Rendiconti del Seminario Matematico, Polytechnic University of Turin, 61(3), pp.247-285, 2003.

[LH10] Liang, L., and Hale, D. A stable and fast implementation of natural neighbor interpolation, 2010.

[MS16] Majdisova, Z., and Skala, V. A radial basis function approximation for large datasets. In Proceedings of SIGRAD 2016, May 23rd and 24th, Visby, Sweden, number 127, pages 9-14. Linköping University Electronic Press, 2016.

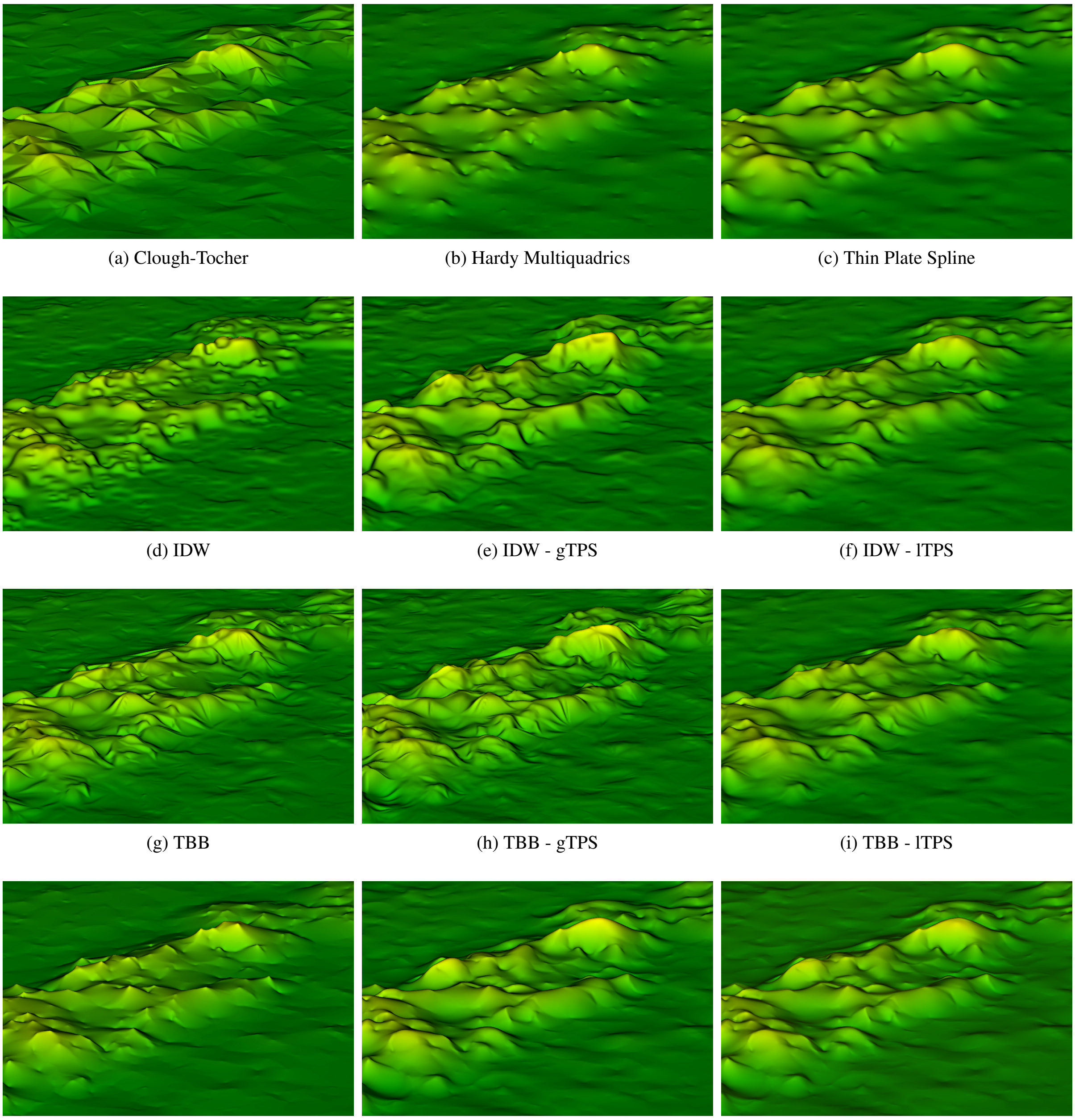
[ON15] Okwuashi, O., and Ndehedehe, C. Digital terrain model height estimation using support vector machine regression. South African Journal of Science, 111(9-10), pp.148-152, 2015.

[PGTG04] Pouderoux, J., Gonzato, J.-C., Tobor, I., and Guitton, P. Adaptive hierarchical rbf interpolation for creating smooth digital elevation models. In Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems, GIS '04, pages 232-240, New York, USA, 2004. ACM.

[PS77] Powell, M., and Sabin, M. Piecewise quadratic approximations on triangles. ACM Transactions on Mathematical Software (TOMS), 3(4), pp.316-325, 1977.

[SS09] Soycan, A., and Soycan, M. Digital elevation model production from scanned topographic contour maps via thin plate spline interpolation. Arabian Journal for Science and Engineering, 34(1), pp.121-134, 2009.

[TH10] Thacker, W., et al. Algorithm 905: SHEPPACK: Modified Shepard algorithm for interpolation of scattered multivariate data. ACM Transactions on Mathematical Software (TOMS), 37(3), 2010.



(a) Clough-Tocher (b) Hardy Multiquadrics (c) Thin Plate Spline

(d) IDW (e) IDW - gTPS (f) IDW - lTPS

(g) TBB (h) TBB - gTPS (i) TBB - lTPS

(j) NN (k) NN - gTPS (l) NN - lTPS

Figure 7: Removing shape artefacts while using local functions. Surfaces of the original methods are on the left, modifications using the tangent planes are in the middle, using local TPS is on the right.