



**HAL**  
open science

# Adaptive Hierarchical RBF Interpolation for Creating Smooth Digital Elevation Models

Joachim Pouderoux, Ireneusz Tobor, Jean-Christophe Gonzato, Pascal Guitton

► **To cite this version:**

Joachim Pouderoux, Ireneusz Tobor, Jean-Christophe Gonzato, Pascal Guitton. Adaptive Hierarchical RBF Interpolation for Creating Smooth Digital Elevation Models. Proceedings of the Twelfth ACM International Symposium on Advances in Geographical Information System 2004, 2004, United States. pp.232–240. hal-00308008

**HAL Id: hal-00308008**

**<https://hal.science/hal-00308008>**

Submitted on 20 Jan 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Adaptive Hierarchical RBF Interpolation for Creating Smooth Digital Elevation Models

Joachim Pouderoux<sup>\*</sup>  
pouderou@labri.fr

Ireneusz Tobor  
tobor@labri.fr

Jean-Christophe Gonzato  
gonzato@labri.fr

Pascal Guitton  
guitton@labri.fr

IPARLA (LaBRI-INRIA) Project  
Université Bordeaux 1  
351, cours de la Libération  
33405 Talence, France

## ABSTRACT

This paper presents a fast algorithm for smooth digital elevation model interpolation and approximation from scattered elevation data. The global surface is reconstructed by subdividing it into overlapping local subdomains using a perfectly balanced binary tree. In each tree leaf, a smooth local surface is reconstructed using radial basis functions. Finally a hierarchical blending is done to create the final  $C^1$ -continuous surface using a family of functions called Partition of Unity. We present two terrain data sets and show that our method is robust since the number of data points in the Partition of Unity blending areas is explicitly specified.

## Categories and Subject Descriptors

G.1.2 [Numerical Analysis]: Approximation of Surfaces and Contours

## General Terms

Algorithms

## Keywords

DEM, contours, interpolation, approximation, scattered data, GIS, grid, computational geometry

## 1. INTRODUCTION

This paper deals with the creation of Digital Elevation Models (DEM) from elevation samples more particularly for processing in Geographical Information Systems (GIS) applications. Originally, mathematician researchers based their work on signal reconstruction from irregular sampling models. More and more point based

<sup>\*</sup>Corresponding author.

models are manipulated for various applications (GIS, virtual reality, video games, etc.). A massive emergence of point based acquisition peripherals: 3D scanners, satellites, radar, etc creates these models.

DEMs can be seen as regular grids of elevation. They are used in many applications such as environment, territory development, virtual reality and video games, geology, hydrography, civil and military engineering, etc. Sources for recent elevation models come from variable sources such as dual satellite images or Synthetic Aperture Radar (SAR). Sources for DEM are, for example, the United States Geological Survey (USGS) in the United States or the Institut Géographique National (IGN) in France. When working on older documents, DEM can be created using contour lines – isolines – extracted from scanned topographic maps [1, 25, 26] or provided by operators such as USGS in the form of Digital Line Graphs (DLG). The problem with DEM reconstruction is to find a specific interpolation model which can produce a continuous surface in a reasonable computational time for a great number of elevation samples. Moreover, the resulting smooth DEM should be as close as possible to the real terrain even if, of course, it is not possible to reproduce the earth complexity only from sparse elevation data.

In this paper, we propose to use a Radial Basis Function (RBF) scheme to interpolate elevation samples. These sparse samples can be extracted by subsampling contour lines. According to Franke [9], RBF are known to be one of the sharpest and stablest methods to solve scattered data approximation and interpolation problems. We propose to enhance this model by an adapted Partition of Unity – PoU – approach which permits to prevent a combinatorial explosion to solve the RBF equation system.

The paper is organized as follows. In section 2, related works developed about the contour lines and scattered data interpolation domain before focusing on previous works about RBF are described. Then, in section 3 we present the RBF principle enhanced by our hierarchical PoU algorithm for limiting computing costs.

Finally, an application to this DEM interpolation from terrain data – irregularly sampled or known on contour lines – is presented in section 4. Results in term of time consumption and visual aspects applied to 3D-terrain reconstruction are shown. The comparison is made by varying the number of known elevation points and intrinsic parameters of our algorithm.

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GIS'04, November 12–13, 2004, Washington, DC, USA.  
Copyright 2004 ACM 1-58113-979-9/04/0011 ...\$5.00.

## 2. RELATED WORKS

Many approaches have been proposed for terrain interpolation based on two incoming data types: scattered elevation samples of terrain, and contour lines which can also be seen as a subset of the first one. In this section we first quickly review the most used contour lines interpolation techniques. Then, we focus on previous works related to the scattered data interpolation problem.

### 2.1 Contour lines interpolation

Many research works have been done on the specific contour line interpolation question. One of the first methods is based on the inverse distance weighted averaging [22, 23] but it generates a lot of artifacts and the resulting model is not guaranteed to be  $C^1$ -continuous. A commonly used family of methods studied in [5, 7, 11, 12] uses Partial Differential Equations (PDE) models such as the Laplacian or the so-called thin-plate model solved using iterative methods. The best known thin-plate implementation is computed by TOPOGRID [16] provided by the commercial software ArcInfo and performs a first pass to compute ridge lines and streams in order to create more realistic DEMs. However, the major drawbacks of these methods are the computing time and the terracing and ringing artifacts they generate. Another algorithm based on a front propagation of euclidian or geodesic distances is presented by Soille in [24]. This is a fast method but the created DEMs are not smooth on contour lines and peaks and pits are not interpolated and remain flat areas. More recently, Gousie et al. [13] propose to iteratively compute intermediate contours between each contour line pairs. Peaks and pits are considered separately by computing a Hermite spline that follows the slope trend. The final DEM is obtained after one or more smoothing passes, but according to the authors, some artifacts remain and the surface has a tendency to be stratified.

### 2.2 Scattered data interpolation

The subject of scattered data interpolation is a very extensive one. Excellent overviews can be found in [10, 19, 21]. In this section, we present the most popular techniques insisting on methods involving RBF.

The earliest technique in this research field is known as the Shepard method [22]. The  $C^0$ -continuous interpolant is based on inverse distance weighting average of data. This is a global method where all data are used for the evaluation of each unknown point. Sibson [23] introduced a variant of this method by modifying the weighted function to take into account only points present in a disc of a fixed radius centered on the evaluated point.

Another popular method constructs an interpolant as a piecewise union of patches (usually low degree multivariate polynomials) joined together with a certain continuity. Bartles [2] used this approach with spline and Bezier patches. These patches are extensively used in the area of the geometric design and give a user freedom to model and change the shape of the object. Haber et al. [14] propose a  $C^1$ -continuous surface generated from triangular Bezier patches.

A more recent method proposed by Lee et al. [18] uses multilevel B-splines. These splines are introduced to compute a  $C^2$ -continuous surface through a set of irregularly spaced points using a coarse-to-fine hierarchy of control lattices. According to Bertram et al. [4], a drawback of this method is that the global refinement of lattices is inefficient when accurately reproducing small local features. In [4], authors also use B-splines in a hierarchical way. They combine adaptive clustering, based on quadrees with piecewise polynomial least square approximations. These polynomials are combined to a B-spline surface with multiple knots at the cluster

boundaries. Finally, a knot removal algorithm is used to combine the fitted patches into a global and smooth surface.

Another approach, that Franke [9] identified as the most accurate and stable one to scattered data interpolation, is to define the interpolation function as a linear combination of radially symmetric basis functions, each centered on a data point. The unknown coefficients are determined by solving a linear system of size  $n$ . Unfortunately, since this method has a global support, the linear system is made of a matrix a symmetric and full matrix. That means that a direct solution of the system is in  $O(n^3)$  which is not acceptable for large data sets with thousands of points.

To overcome this problem, Wendland [28] uses compactly supported RBFs and the linear system becomes sparse. This algorithm has been further improved by Kojekine et al. [17] by organizing the sparse matrix into a band-diagonal sparse matrix which can be solved more efficiently by using iterative methods. Unfortunately, the radius of support has to be chosen globally, which means that the method is not robust against highly non-uniformly distributed point sets where the density of the samples may vary significantly. Another approach has been proposed by Carr et al. [6]. It is based on the fast evaluation of RBF proposed by Beatson et al. [3] using fast multipole methods. Unfortunately, the far field expansion has to be done separately for every basis function, and is very complex to implement.

In a theoretical paper, Wendland [29] proposed another improvements of radial basis functions reconstruction method based on the PoU and spatial decomposition into regular cells. This idea has been recently applied for implicit surface reconstruction by Ohtake et al. [20] and Tobor et al. [27].

In this paper, we propose an efficient model based on this previous approach which combines RBF and a hierarchical PoU reconstruction.

## 3. METHOD DESCRIPTION

Our algorithm for an efficient reconstruction of terrain is based on the well-known RBF interpolation and PoU integration in order to compose small local solutions into a global one. An improvement of PoU (called hierarchical) optimizing the reconstruction and the evaluation time is also proposed here.

There are two stages in the reconstruction scheme: reconstruction of the function  $f$  and its evaluation at a point  $\mathbf{p}$ . The PoU-based reconstruction being quasi-independent from geometry partitioning, we propose a new partition scheme that minimizes the time of each of the two stages. First it seems clear that the minimal reconstruction time is reached if all the local subdomains have an equal number of points. Second, the  $f(\mathbf{p})$  evaluation time is shorter if the partitioning scheme allows a fast search of the domains containing  $\mathbf{p}$ . Our reconstruction and evaluation scheme optimizes these two processing times. This scheme also allows more control over robustness and stability of the numerical solution.

We begin with a presentation of the theoretical background of our method, then, our new hierarchical approach of reconstruction and evaluation scheme is detailed. Finally a short complexity analysis is done.

### 3.1 Theoretical background

#### 3.1.1 RBF reconstruction

Given the set of  $N$  pairwise distinct points  $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$  of dimension 2:  $\mathbf{p}_k = \{p_k^x, p_k^y\} \in \mathbb{R}^2$ , and the set of values  $\{h_1, \dots, h_N\}$ , we want to find a function  $f: \mathbb{R}^2 \rightarrow \mathbb{R}$  satisfying the constraints

$$f(\mathbf{p}_i) = h_i \quad i = 1 \dots N. \quad (1)$$

As such a problem having an infinite number of solutions, the variational techniques with RBF give a solution that minimizes an “energy” or “smoothness” of the reconstructed function. Duchon [8] has showed that the simplest smoothing function is a linear combination of rotation-invariant basis functions

$$f(\mathbf{p}) = \sum_{i=1}^N \omega_i \phi(\|\mathbf{p}, \mathbf{p}_i\|) + \pi(\mathbf{p}). \quad (2)$$

We denote here  $\|\mathbf{p}_i, \mathbf{p}_j\|$  the Euclidean distance,  $\omega_i$  the linear combination weights (to be computed),  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  a conditionally positive definite basis function, and  $\pi$  a polynomial of degree  $m$  depending on the choice of  $\phi$ .

For a reconstruction of topographic elevation data the best choice proposed by Hardy [15] is the multiquadric function:

$$\phi(r) = \sqrt{r^2 + \alpha^2} \quad (3)$$

where  $\alpha$  controls smoothness against fidelity to the data. The associated polynomial  $\pi$  of degree 1 is defined by:

$$\pi(\mathbf{p}) = c_0 + c_1 \mathbf{p}^x + c_2 \mathbf{p}^y \quad (4)$$

As we have an under-determined system with  $N + 3$  unknowns ( $\omega_i$  and  $c_0, c_1, c_2$ ) and only  $N$  equations, so-called natural additional constraints for the coefficients  $\omega_i$  are added in order to ensure orthogonality, so that

$$\sum \omega_i = \sum \omega_i \mathbf{p}_i^x = \sum \omega_i \mathbf{p}_i^y = 0 \quad (5)$$

The equations (1), (2), and (5) determine the following linear system:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (6)$$

$$\mathbf{A} = \begin{bmatrix} \Phi & P \\ P^T & 0 \end{bmatrix} \quad (7)$$

$$\Phi = [\phi(\|\mathbf{p}_i, \mathbf{p}_j\|)]_{\substack{i=1\dots N \\ j=1\dots N}}$$

$$P = [1 \ \mathbf{p}_i^x \ \mathbf{p}_i^y]_{i=1\dots N}$$

$$\mathbf{x} = [\omega_1, \omega_2, \dots, \omega_N, c_0, c_1, c_2]^T \quad (8)$$

$$\mathbf{b} = [h_1, h_2, \dots, h_N, 0, 0, 0]^T \quad (9)$$

The solution vector  $\mathbf{x}$  is composed of the weights  $\omega_i$  and the polynomial coefficients  $c_i$  for equation (2) and represents a solution of the interpolation problem given by (1).

### 3.1.2 Partition of unity principle

The main idea of the PoU method is to divide the global domain of interest into smaller overlapping subdomains where the problem can be solved locally. More precisely, the difficult global problem is decomposed into several smaller local problems and their local solutions are combined together by using weighting functions that act as smooth blending functions to obtain the global solution.

Notice that our new hierarchical approach requires the decomposition into two subdomains per hierarchical step. We introduce here only the simplest case; the extension on more subdomains being straightforward and out of the scope of this paper.

Consider the global domain  $\Omega$  and divide it into two overlapping subdomains  $\Omega_1$  and  $\Omega_2$  with  $\Omega = \Omega_1 \cup \Omega_2$  and  $\Omega_1 \cap \Omega_2 \neq \emptyset$ . On this set of subdomains  $\{\Omega_1, \Omega_2\}$ , we construct a partition of unity, i.e. a collection of non-negative functions  $\{\lambda_1, \lambda_2\}$  with limited support  $\text{supp}(\lambda_i) \subseteq \Omega_i$  and with  $\lambda_1 + \lambda_2 = 1$  in the entire domain  $\Omega$ .

For each  $\Omega_i$ , a set  $\mathcal{P}_i = \{\mathbf{p} \in \mathcal{P} \mid \mathbf{p} \in \Omega_i\}$  is defined, then a local reconstruction function  $f_i$  is computed. The global reconstruction

function  $f_{pou}$  is then defined as a combination of the local functions

$$f_{pou}(\mathbf{p}) = f_1(\mathbf{p})\lambda_1(\mathbf{p}) + f_2(\mathbf{p})\lambda_2(\mathbf{p}) \quad (10)$$

The condition  $\lambda_1 + \lambda_2 = 1$  is obtained from any other set of smooth functions  $\Lambda_1, \Lambda_2$  by a normalization procedure

$$\lambda_i(\mathbf{p}) = \frac{\Lambda_i(\mathbf{p})}{\Lambda_1(\mathbf{p}) + \Lambda_2(\mathbf{p})} \quad \text{with } i = 1, 2 \quad (11)$$

The weighting functions  $\Lambda_i$  have to be continuous at the boundary of the subdomains  $\Omega_i$ . We define the weighting functions  $\Lambda_i$  as the composition of a distance function  $D_i : \mathbb{R}^n \rightarrow [0, 1]$ , where  $D_i(\mathbf{p}) = 1$  at the boundary of  $\Omega_i$  and a decay function  $V : [0, 1] \rightarrow [0, 1]$ , i.e.  $\Lambda_i(\mathbf{p}) = V \circ D_i(\mathbf{p})$ .

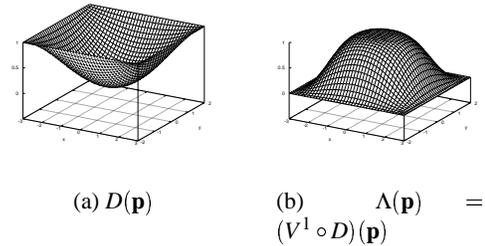
For a 2D axis-aligned box defined from the two opposite corners  $S$  and  $T$  we use the following distance function  $D_i$

$$D_i(\mathbf{p}) = 1 - \prod_{r \in \{x, y\}} \frac{4(\mathbf{p}_r - S_r)(T_r - \mathbf{p}_r)}{(T_r - S_r)^2}. \quad (12)$$

The choice of the decay function  $V$  determines the continuity between the local solutions  $f_i$  in the global reconstruction function  $f_{pou}$ . We suggest to use one of the following decay functions that were chosen by including some simple constraints similar to the construction of base spline functions ( $V(0) = 1, V(1) = 0, V'(0) = V'(1) = 0$ , etc.):

$$\begin{aligned} \text{continuity } \mathbb{C}^0 : & \quad V^0(d) = 1 - d \\ \text{continuity } \mathbb{C}^1 : & \quad V^1(d) = 2d^3 - 3d^2 + 1 \end{aligned}$$

The plots of the distance function  $D$  and the weighting functions are shown in Figure 1.



**Figure 1: Interpretation of the distance function  $D$  and weighting functions  $\Lambda$ .**

## 3.2 Our hierarchical approach

Our reconstruction algorithm is composed of two steps. First, the global domain of interest is subdivided recursively into overlapping local subdomains containing a quasi-equal number of points using a tree structure based mainly on the k-d principle (section 3.2.1). These overlapping regions are used into the PoU based process of local solution blending. Second, the local reconstruction functions in the leaf nodes is computed using variational techniques with RBF.

The evaluation of the global reconstruction function by a PoU blending of the local reconstruction functions is done in recursive manner and propagated bottom-up from the leaf nodes (section 3.2.2). The very last blending is made into root node that describes the global reconstruction function.

### 3.2.1 Binary domain decomposition

Starting from the unorganized point set  $\mathcal{P}$  and the entire domain of interest  $\Omega^0$  (generally defined by bounding box of  $\mathcal{P}$ ), the binary tree domain decomposition method adaptively subdivides  $\Omega^0$  using a kd-tree-like structure. The tree is built in a top-down recursive process starting from the root node, where the entire domain  $\Omega^0$  is subdivided into two overlapping subdomains  $\Omega_1^1$  and  $\Omega_2^1$  containing an near-equal number of points  $n_1^1$  and  $n_2^1$  in the respective point sets  $\mathcal{P}_1^1$  and  $\mathcal{P}_2^1$ . All subdomains  $\Omega_k^l$  at level  $l$  are themselves subdivided recursively into two overlapping subdomains  $\Omega_{k,1}^{l+1}$  and  $\Omega_{k,2}^{l+1}$  containing a near equal number of points  $n_{k,1}^{l+1}$  and  $n_{k,2}^{l+1}$  in the respective point sets  $\mathcal{P}_{k,1}^{l+1}$  and  $\mathcal{P}_{k,2}^{l+1}$ . The recursion is terminated, when the number of points in a subdomain falls below a threshold  $T_{leaf}$ .

Let us now give some insight how to subdivide a domain  $\Omega_k^l$ .

First, the minimum number of points  $n_o^l = \text{Card}(\mathcal{P}_{k,1}^{l+1} \cap \mathcal{P}_{k,2}^{l+1})$  in the overlapping zone  $\Omega_{k,1}^{l+1} \cap \Omega_{k,2}^{l+1}$  has to be specified explicitly as an *overlap quota*  $q \in ]0, 1[$  of the number of points  $n_k^l$ :

$$n_{o,k}^l = qn_k^l \quad (13)$$

Then, the minimum number of points  $n_k^{l+1}$  in the subdomains can be calculated as follows:

$$n_k^{l+1} = \left\lceil \frac{n_{o,k}^l + n_k^l}{2} \right\rceil \quad (14)$$

The extent of the two overlapping subdomains  $\Omega_{k,1}^{l+1}$  and  $\Omega_{k,2}^{l+1}$  is computed using a k-d tree principle. First, the longest axis of  $\Omega_k^l$  is determined. Second, we collect the point sets  $\mathcal{Q}_{k,1}^{l+1}$  (respectively  $\mathcal{Q}_{k,2}^{l+1}$ ) containing the  $n_k^{l+1}$  points with the lowest (respectively highest) values with respect to the longest axis. In practice, we rearrange the points  $\mathcal{P}_k^l$  with respect to their values of the longest axis and we take first (respectively last)  $n_k^{l+1}$  points: by setting  $i_1 = n_k^{l+1}$  and  $i_2 = n_k^l - n_k^{l+1} + 1$ , we rearrange the points so that  $\mathbf{p}_i < \mathbf{p}_{i_1}$  for  $1 \leq \mathbf{p}_i \leq \mathbf{p}_{i_1}$  and  $\mathbf{p}_i > \mathbf{p}_{i_2}$  for  $\mathbf{p}_{i_2} \leq \mathbf{p}_i \leq \mathbf{p}_{n^l}$ :

$$\mathcal{P}_k^l = \underbrace{\mathbf{p}_1, \dots, \mathbf{p}_{i_2-1}}_{\mathcal{Q}_{k,1}^{l+1}}, \underbrace{\mathbf{p}_{i_2}, \dots, \mathbf{p}_{i_1}, \mathbf{p}_{i_1+1}, \dots, \mathbf{p}_{n^l}}_{\mathcal{Q}_{k,2}^{l+1}} \quad (15)$$

Finally,  $\Omega_{k,1}^{l+1}$  and  $\Omega_{k,2}^{l+1}$  are defined by the bounding boxes of  $\mathcal{Q}_{k,1}^{l+1}$  and  $\mathcal{Q}_{k,2}^{l+1}$  and the points  $\mathcal{P}_{k,1}^{l+1}$ ,  $\mathcal{P}_{k,2}^{l+1}$  as the subsets of  $\mathcal{P}_k^{l+1}$  that is contained in  $\Omega_{k,1}^{l+1}$ ,  $\Omega_{k,2}^{l+1}$

Note that the  $\mathcal{P}^l \rightarrow \mathcal{Q}^{l+1} \rightarrow \Omega^{l+1} \rightarrow \mathcal{P}^{l+1}$  step is necessary if the initial point set  $\mathcal{P}$  is a subset of a regular grid and then the  $\mathbf{p}_{i_2-1}$  and  $\mathbf{p}_{i_2}$  from (15) can have the same coordinate on the partition axis.

Consider the resulting recursive algorithm for the binary tree domain decomposition shown in Algorithm 1 that has to be called using `Decompose( $\mathcal{P}$ ,  $\Omega$ )`. Finally, Figure 2 illustrates the first levels of the resulting binary tree with the corresponding overlapping subdomains.

### 3.2.2 Evaluation

For the evaluation of the global reconstruction function  $f$  at a point  $\mathbf{p}$ , the local reconstructions are blended together by recursively applying the PoU method bottom-up in the binary tree computed in the first reconstruction step (section 3.2.1). Starting from

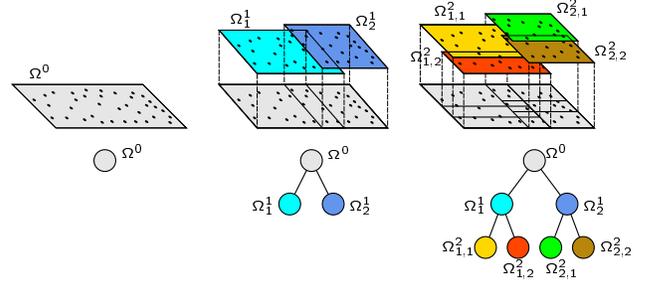
---

#### Algorithm 1 Decompose( $\mathcal{P}$ , $\Omega$ )

---

**Require:** points  $\mathcal{P}$ , domain  $\Omega$   
**Ensure:** binary tree  
 set  $n = \text{Card}(\mathcal{P})$   
**if**  $n > T_{leaf}$  **then**  
   set  $n_o = qn$   
   rearrange the points in  $\mathcal{P}$  according to the longest axis of  $\Omega$   
   determine the points  $\mathcal{P}_1, \mathcal{P}_2$   
   determine the subdomains  $\Omega_1, \Omega_2$   
   determine the points  $\mathcal{P}_1, \mathcal{P}_2$   
   Decompose( $\mathcal{P}_1, \Omega_1$ )  
   Decompose( $\mathcal{P}_2, \Omega_2$ )  
**else**  
   Calculate local RBF reconstruction for  $\mathcal{P}$   
**end if**

---



**Figure 2: First steps of the binary tree domain decomposition into overlapping subdomains.**

the root node, at every intern node of the binary tree level  $l$  the reconstruction function  $f^l$  is given by the PoU blending of the local reconstruction functions of the two child nodes  $f_1^{l+1}$  and  $f_2^{l+1}$ :

$$f^l(\mathbf{p}) = \frac{f_1^{l+1}(\mathbf{p})\Lambda_1^{l+1}(\mathbf{p}) + f_2^{l+1}(\mathbf{p})\Lambda_2^{l+1}(\mathbf{p})}{\Lambda_1^{l+1}(\mathbf{p}) + \Lambda_2^{l+1}(\mathbf{p})} \quad (16)$$

The full recursive algorithm 2 illustrates the evaluation at a point  $\mathbf{p}$  by applying the recursive PoU blending and it has to be called by `Evaluate( $\mathbf{p}$ , 0)`.

---

#### Algorithm 2 Evaluate( $\mathbf{p}$ , $l$ )

---

**Require:** point  $\mathbf{p}$ , level  $l$   
**Ensure:**  $f(\mathbf{p})$   
**if**  $\mathbf{p}$  not in  $\Omega^l$  **then**  
   return 0  
**end if**  
**if**  $l$  is the leaf level **then**  
   return  $f_{rbf}(\mathbf{p})$   
**else**  
    $f_1 = \text{Evaluate}(\mathbf{p}, l+1)$  at the left child  
    $f_2 = \text{Evaluate}(\mathbf{p}, l+1)$  at the right child  
   compute  $\Lambda_1, \Lambda_2$  from children  
   return  $f = (f_1\Lambda_1 + f_2\Lambda_2)/(\Lambda_1 + \Lambda_2)$   
**end if**

---

## 3.3 Complexity analysis

The solution of the linear system (6) of size  $N$  requires  $O(N^3)$  floating point operations and  $O(N^2)$  core-memory cells. For one single evaluation the global RBF approach requires  $O(N)$  operations. Thus it is clear that direct methods are not suitable for a number of constraints greater than several thousands.

In our reconstruction scheme the binary tree height is about  $H = \log(N/T_{leaf})$ , the number of leaf nodes is about  $L = N/T_{leaf}$  and

each leaf contains at most  $T_{leaf}$  constraints. The reconstruction is composed of a subdivision stage which performs  $O(N \log N)$  floating point operations, and a RBF reconstruction stage in  $O(LT_{leaf}^3) = O(L) = O(N)$  operations.

The evaluation of  $f(\mathbf{p})$  can be done in bounded time due to the limited number of subdomains including  $\mathbf{p}$  where the evaluation of the local RBF function is done in a constant time.

## 4. RESULTS AND DISCUSSION

To illustrate our method we present two examples. The first one uses a downsampled full DEM model, the second involves a set of contour lines. In the following tables, we denote #leafs the number of tree leaves,  $t_{tree}$  the binary tree creation time in seconds,  $t_{rec}$  the local reconstruction time in seconds (RBF reconstruction into each leaf),  $t_{eval}$  the evaluation time of the global interpolation function for all grid nodes, and  $t_{total}$  the global interpolation time in seconds. For all our tests, we fixed empirically the overlapping quota to 20%. It is clear that if this quota is too small, some artefacts can appear in overlapped areas which not match the neighborhood data; if the quota is too high, the generated terrain becomes too smooth.

Note that all results presented in this section have been computed on a PC with an Intel Pentium IV 4GHz processor and with a 1.5GB memory.

### 4.1 Mount Washington

As a first experiment we used an USGS DEM of Mount Washington<sup>1</sup>. It consists in a  $932 \times 1384$  grid (=1,289,888 total points, see Fig. 4(a) and 4(b)). The elevations vary from 170 to 1913 meters.

The original model has been randomly downsampled to 43,419 points, that is 3.3% of the original (see Fig.4(c)). Remaining data points have been used to interpolate the terrain on the initial grid. Finally, the reconstructed DEM is compared with the original one using the Root Mean Square Error – RMSE. Numerical results using a smoothing factor  $\alpha = 2$  (note that  $\alpha$  does not influence the computing time) for our implementation are enumerated in table 1.

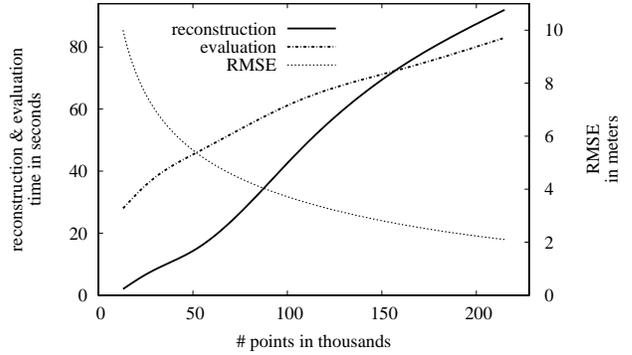
**Table 1: Numerical results for Mount Washington DEM, 43,419 data points (on 1,289,888 total points) with  $\alpha = 2$**

$T_{leaf}$	#leafs	$t_{tree}$	$t_{rec}$	$t_{eval}$	$t_{total}$	RMSE
1600	128	22	854	188	1066	5.04
800	256	20	374	157	531	5.05
600	512	16	170	100	287	5.05
200	2048	15	32	59	108	5.06
100	4096	18	16	46	81	5.09

We can see that for very different  $T_{leaf}$  values, RMSE values stay very closed. This is due to the fact that data points are quasi-regularly dispersed on the grid and in this case the result is independent of the  $T_{leaf}$  choice.

Figure 3 confirms the theoretical time complexity of our algorithm. First, the original model was downsampled to several smaller point sets (i.e. from 1% to 20% of the original size). For each downsampled dataset the reconstruction function (with  $T_{leaf} = 100$  and  $\alpha = 2$ ) is computed and the DEM of original model size is evaluated. Finally, the RMS error is computed against the full dataset. Note also that RMSE decreases while the number of point increases.

<sup>1</sup>Available on the website of the Department of Geological Sciences at the University of Washington in Seattle: <http://duff.geology.washington.edu>



**Figure 3: Plots of time consuming analysis and RMSE. RBF reconstruction is linear and the global evaluation**

Fig.4(d) shows a 3D view of the interpolated terrain with parameters  $\alpha = 5$  and  $T_{leaf} = 800$ . We can see that most of the terrain features have been preserved even if the terrain is quite smooth.

### 4.2 Mount St. Helens

For this experiment, we used a set of contour lines representing the Mount Saint Helens volcano, in Washington State, after the eruption of May 18, 1980.

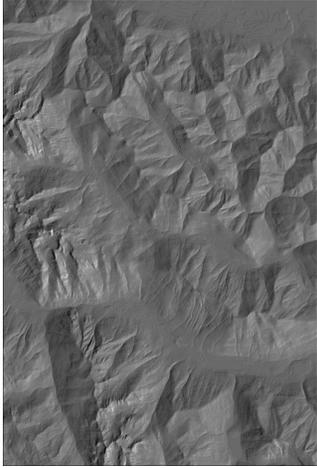
We use the full contour data which consists of 21 contours (21,442 data points) drawn on a  $449 \times 497$  grid (=223,153 total points, see Fig.5(a)). Contour line relative elevations vary from 20 to 740 meters. Table 2 shows numerical results for this data set.

**Table 2: Numerical results for Mount St Helens’s contour lines: 21,442 data points (grid is 223,153 total points)**

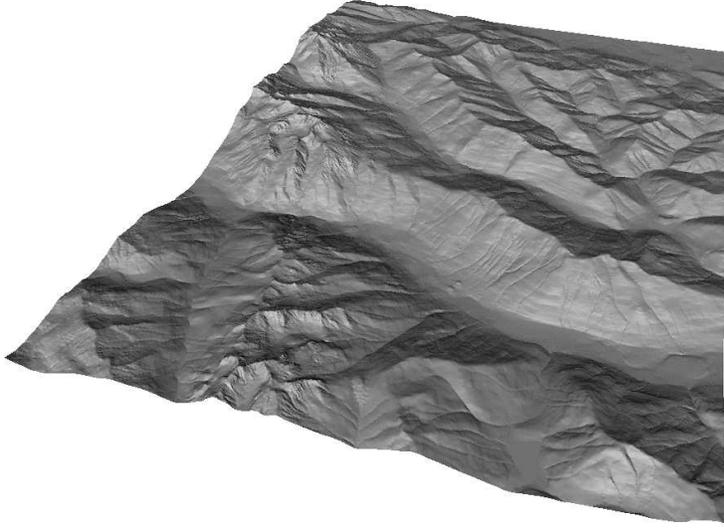
$T_{leaf}$	#leafs	$t_{tree}$	$t_{rec}$	$t_{eval}$	$t_{total}$
1200	32	4	89	13	107
800	64	2	41	10	53
400	128	3	17	7	28

Figures 5(b) and 5(c) show the interpolated model with two different smoothnesses with  $T_{leaf} = 800$ . With  $T_{leaf} = 100$ , the global interpolated model is quite correct but on the left of figure 5(d) we can see that some terrain areas are flat and have not been correctly reconstructed. This is due to the fact that some leaf areas contain many data points with almost the same elevation. That means that manual interventions are sometimes needed in order to interactively correct the interpolation parameters.

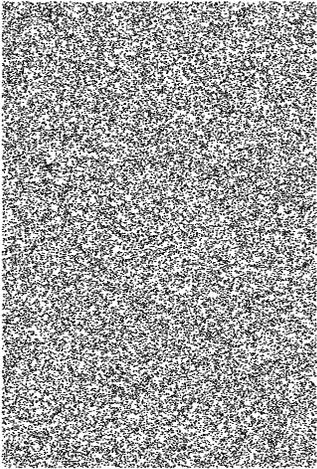
Next, we use randomly downsampled contour line data. Only 2,332 data points from the previous 21,442 points are conserved, that is about 10% (see Fig.6(a)). Computing times are summed up in table 3 and figure 6(b) depicts the resulting terrain with  $\alpha = 2$  and  $T_{leaf} = 800$ . Compared with figure 5(b), the obtained terrain conserves perfectly the global appearance and most of the details of the previous model with 10 times less of data. No oscillations can be observed between sample points, and this is even more true with higher values of  $\alpha$ . That means that this kind of downsampling is not a problem for reconstructing DEM from contour lines, and can be used to speed up the reconstruction process.



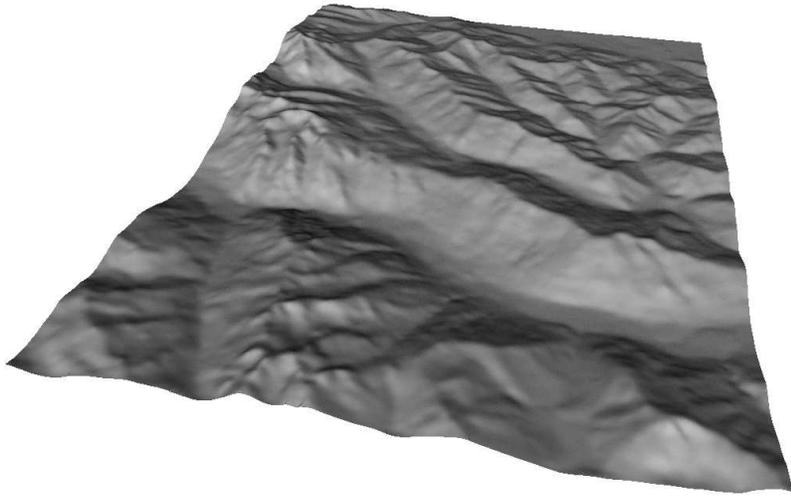
(a) Original DEM with 1,314,992 points



(b) 3D-view of the original DEM

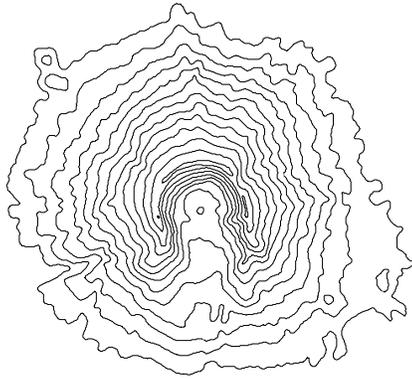


(c) DEM subsampled to 43,419 points (3.3% of initial set)

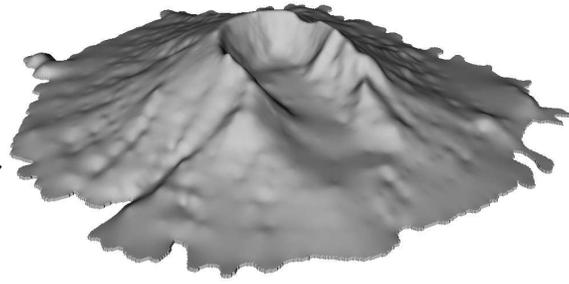


(d) 3D-view of the interpolated DEM with  $T_{leaf} = 800$  and  $\alpha = 5$ .  $t_{total} = 531s$

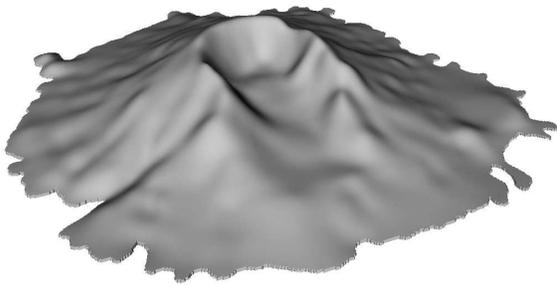
**Figure 4: Mount Washington DEM interpolation from scattered points**



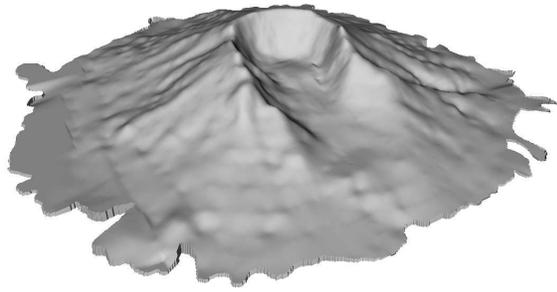
(a) Original contour lines set made of 21,442 points on the grid



(b) 3D-view of the interpolated DEM with  $\alpha = 2$ ,  $T_{leaf} = 800$  and  $t_{total} = 53s$

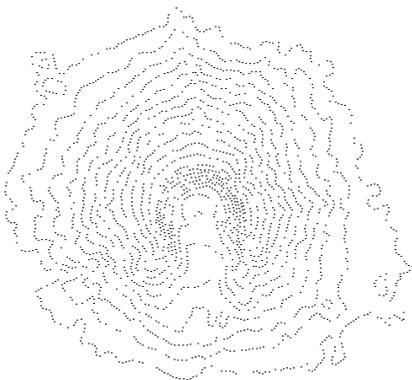


(c)  $\alpha = 10$  and  $T_{leaf} = 800$ .  $t_{total} = 53s$

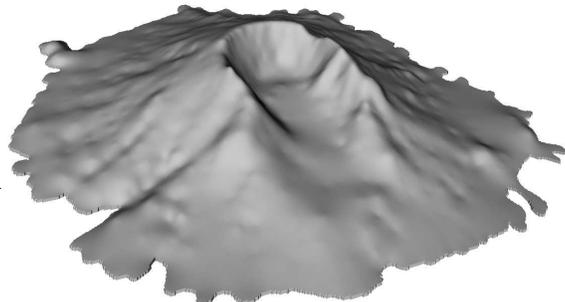


(d)  $\alpha = 2$  and  $T_{leaf} = 100$ .  $t_{total} = 10s$

**Figure 5: Mount St. Helens interpolation from contour lines**



(a) Contour lines resampled to 2,332 data points



(b)  $\alpha = 2$  and  $T_{leaf} = 800$ .  $t_{total} = 9s$

**Figure 6: Mount St. Helens interpolation from downsampled contour lines**

**Table 3: Numerical results for Mount St. Helens’s downsampled contour lines: 2,332 data points (grid is 223,153 total points)**

$T_{leaf}$	#leafs	$t_{tree}$	$t_{rec}$	$t_{eval}$	$t_{total}$
1200	4	0	8	7	16
800	8	0	3	5	9
400	16	0	1	3	5

## 5. CONCLUSION

We presented a new efficient and easy to implement algorithm for interpolating scattered data. It is based on Hardy’s multiquadric RBF and the Partition of Unity approach. The global domain is spatially decomposed into overlapping rectangles in a hierarchical manner, then, the local RBF reconstruction is computed and the local solutions are recursively blended together. The RBF approach is chosen to guarantee a robust solution even against non-uniform data. The PoU decomposition and blending is used to accelerate the process. It can be used to parallelize the reconstruction scheme on a multiprocessor system and to out-of-core processing in the case of a huge amount of data.

Note that the general method is independent of the local reconstruction method. Furthermore, the hierarchical PoU approach can be combined with other methods (such the Shepard’s [22] or Lee’s [18] methods), in order to improve them.

We have used visual aspects, sensitivity to parameters, timing requirements to test characteristics of our method. In term of visual aspects, the algorithm generates credible smooth terrains. Our reconstruction scheme has a linear behavior with respect to the size of initial data and near-constant evaluation time. The reconstruction is completely automatic, but the numerical stability can be improved by manually tuning the set of initial parameters of PoU blending.

Our algorithm is used to reconstruct smooth DEMs from unorganized elevation samples especially from contour lines. Obviously it is a general framework of reconstruction and it can be applied to many other interpolation problems for other scientific applications that process scattered data in two, tree, or more dimensions, for example hypsography, geology, finite element method, MRI medical imaging, shape transformations, image inpainting and more others.

## 6. REFERENCES

- [1] P. Arrighi and P. Soille. From scanned topographic maps to digital elevation models. In *Proceedings of Geovision’99*, 1999.
- [2] R. Bartles. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann Publishers, Inc, 1987.
- [3] R.K. Beatson and W.A. Light. Fast evaluation of radial basis functions: methods for two-dimensional polyharmonic splines. *Computational Mathematics and Applications*, 24(12):7–20, 1992.
- [4] M. Bertram, X. Tricoche, and H. Hagen. Adaptive smooth scattered-data approximation for large-scale terrain visualization. In *Proceedings of the symposium on Data visualisation 2003*, pages 177–184. Eurographics Association, 2003.
- [5] I. Briggs. Machine contouring using minimum curvature. *Geophysics*, (39):39–48, 1974.
- [6] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W.R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76. ACM Press, 2001.
- [7] J. Childs. Development of a two-level iterative computational method for solution of the Franklin approximation algorithm for the interpolation of large contour line data sets. Master’s thesis, Rensselaer Polytechnic Institute, Troy, NY 12180, may 2003.
- [8] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In Walter Schempp and Karl Zeller, editors, *Constructive Theory of Functions of Several Variables*, pages S. 85–100. Springer-Verlag, Berlin-Heidelberg, 1977. NAM-Bibliothek S/M/571.
- [9] R. Franke. Scattered data interpolation: Tests of some methods. *Mathematics of Computation*, 38(157):181–200, 1982.
- [10] R. Franke and G. M. Nielson. Scattered data interpolation and applications- a tutorial and survey. In H. Hagen and D. Roller, editors, *Geometric Modelling: Methods and Their Application*, pages 131–160. Berlin: Springer-Verlag, 1991.
- [11] W. R. Franklin. Elevation data operations. *Dagstuhl - Workshop on computational cartography*, 1996.
- [12] M. B. Gousie and W. R. Franklin. Converting elevation contours to a grid. *Eighth International Symposium on Spatial Data Handling (SDH)*, 1998.
- [13] M. B. Gousie and W. R. Franklin. Constructing a DEM from grid-based data by computing intermediate contours. In *Proceedings of the eleventh ACM international symposium on Advances in geographic information systems*, pages 71–77, New Orleans, Louisiana, USA, 2003. ACM Press.
- [14] J. Haber, F. Zeilfelder, O. Davydov, and H. P. Seidel. Smooth approximation and rendering of large scattered data sets. In *Proceedings of the conference on Visualization ’01*, pages 341–348, San Diego, California, 2001. IEEE Computer Society.
- [15] R. L. Hardy. Multiquadric equations of topography and other irregular surfaces. *J. Geophys. res*, 76:1905–1915, 1971.
- [16] M.F. Hutchinson. Calculation of hydrologically sound digital elevation models. In *Proceedings of the Third International Symposium on Space Data Handling*, pages 117–133, Columbus, Ohio, 1988. International Geographical Union.
- [17] N. Kojekine, I. Hagiwara, and V. Savchenko. Software tools using csrbfs for processing scattered data. *Computers & Graphics*, 27(2):311–319, 2003.
- [18] S. Lee, G. Wolberg, and S.Y. Shin. Scattered data interpolation with multilevel b-splines. *IEEE Transactions on Visualization and Computer Graphics*, 3(3):228–244, july–september 1997.
- [19] S. K. Lodha and R. Franke. Scattered data techniques for surfaces. In H. Hagen, G.M. Nielson, and F. Post, editors, *Proc. Dagstuhl Conf. Scientific Visualization*, pages 182–222, 1999.
- [20] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. In *Proceedings of SIGGRAPH’03*, pages 463–470, 2003.
- [21] L. L. Schumaker. Fitting surfaces to scattered data. In G.G. Lorentz, C.K. Chui, and L.L. Schumaker, editors, *Approximation Theory II*, pages 203–268, New York, 1976. Academic Press.

- [22] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM Press, 1968.
- [23] R. Sibson. *A Brief Description of Natural Neighbor Interpolation*, in *Interpreting Multivariate Data*, pages 21–36. V. Barnett, John Wiley and Sons, 1981.
- [24] P. Soille. Spatial distributions from contour lines: An efficient methodology based on distance transformation. *Journal of Visual Communication and Image Representation*, 2(2):138–150, june 1991.
- [25] S. Spinello and G. Greiner. Automatic contour line recognition from scanned topographic maps. Technical report, University of Erlangen, 2001.
- [26] S. Spinello and P. Guitton. Contour line recognition from scanned topographic maps. *Journal of WSCG*, 12(1-3), 2004.
- [27] I. Tobor, P. Reuter, and C. Schlick. Efficient reconstruction of large scattered geometric datasets using the partition of unity and radial basis functions. In *Journal of WSCG 2004*, volume 12, pages 467–474, 2004.
- [28] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree. *Advances in Computational Mathematics*, 4:389–396, 1995.
- [29] H. Wendland. *Fast evaluation of radial basis functions: Methods based on partition of unity. Approximation theory X: Abstract and classical analysis*, pages 473–483. Vanderbilt University Press, Nashville, 2002.