COMPACT REPRESENTATION OF RANGE IMAGING SURFACES

B. Li
Manchester Metropolitan Univ.
Dept. of Computing and Mathematics
Manchester, UK, M1 5GD

Q. Meng and H. Holstein
Univ. of Wales, Aberystwyth
Dept. of Computer Science
Aberystwyth, UK, SY23 3DB

ABSTRACT

Range images of complex geometry presented by large point data sets almost always yield surface reconstruction imperfections. We propose a novel compact and complete mesh representation for non-uniformly sampled noisy range image data using an adaptive Radial Basis Function network. The network is established using a heuristic learning strategy. Neurons can be inserted, removed or updated iteratively, adapting to the complexity and distribution of the underlying data. This flexibility is particularly suited to highly variable spatial frequencies, and is conducive to data compression with network representations. Experiments confirm the performance advantages of the network when applied to 3D point-cloud surface reconstruction.

Index Terms— Geometric modelling, Neural network applications, Image representations

1. INTRODUCTION

Digitisation techniques such as range scanning devices are widely used for 3D model acquisition. Laser scanners can acquire millions of irregularly distributed point-clouds on digitised surfaces of objects. However, sampling a complex geometry from a specific viewpoint almost always yields surface reconstruction imperfections, in particular “holes”. Faithful compact reproduction of incomplete noisy meshes is a ubiquitous problem for visualisation and engineering modelling.

A common way to complete surface is to fill each individual hole with a smooth surface patch that meets the boundary conditions. Such an approach could work well for small holes in relation to the geometric variation in the surface. Sophisticated surface reconstruction demands a functional representation residing in a low dimensional and, therefore, computationally manageable feature space. Inspired by advances in radial basis functions (RBFs) for solving function approximation and pattern classification problems [1, 2], implicit modelling with RBFs has recently emerged as an effective technique in computer graphics for 3D modelling [3, 4]. The RBFs have good global generalisation property in function approximation, have simple topology and can accommodate data on irregular grids. These properties make RBFs well suited for interpolating scattered data distributed across large, irregular holes. However, problems of computational cost in high dimensional parameter space and approximation accuracy remain, particularly for locally sharp features, since RBF solutions are global in nature. Consequently, fitting and evaluating RBFs by means of nonlinear least squares optimisation for large data sets is challenging for most present PCs.

In this paper, we present a new approach that adapts the advantages of the RBFs to the framework of neural networks. The hybrid RBF-network method aims to satisfy the desirable criteria of functional representation compactness, robustness to data noise, and adaptivity to highly variable spatial densities and data complexity. The novelty of the proposed RBF network is that neurons can be located and adjusted iteratively according to the distribution and complexity of the underlying data. This flexibility is particularly suited to cases of highly varying spatial density and provides the possibility of compact network representations. Furthermore, we report a greedy learning algorithm that uses a neighbourhood extended Kalman filter (NEKF), leading to significant computation reduction in the training process.

2. THE ADAPTIVE RBF NETWORK

The network has one hidden layer with $K$ Gaussian RBF functions $\phi_k(x)$. The network output $f(x)$ takes the form

$$f(x) = a_0 + \sum_{k=1}^{K} a_k \phi_k(x)$$  \hspace{1cm} (1)

where $\phi_k(x) = \exp \left( -\frac{1}{\sigma_k^2} \| x - \mu_k \|^2 \right)$. $\| . \|$ denotes the Euclidean norm, $\mu_k$ locates the centre of the $k$th neuron, and $\sigma_k$ (standard deviation) indicates its width (coverage). $a_k$ is its weight connecting to the output and $a_0$ is the bias term.

To achieve network adaptivity to the underlying data, we developed an adaptive RBF network algorithm benefitted from the self-organising learning model of the resource-allocating network (RAN) [5, 6]. At each learning step: 1) the network grows one neuron, where necessary, based on the “novelty” of the input observation; 2) if the “novelty” criteria are not
satisfied, resulting in no newly added neuron at this instance, a subset of neuron parameters are adjusted in full dimension in accordance with a neighbourhood extended Kalman filter (NEKF) algorithm; 3) network compaction is obtained by pruning “pseudo” neurons that consistently make little contribution to the network output. The remainder of this section describes the learning process in these three stages.

2.1. Network growth

In the case of surface reconstruction from point clouds, the training set \( T = \{x_n, z_n\}_{n=1}^N \) consists of normalised and randomised 3D point data. The \( N \) pairs of 2D points \( x_n = [x_n, y_n]^T \) are network inputs having associated \( z_n \) as outputs.

The network starts with no hidden unit. It grows by inserting a new neuron if the current observation \( \{x_n, z_n\} \) satisfies the following three novelty conditions:

**Condition 1:** the input of the observation is sufficiently far away from all existing neurons:

\[
d_n = \|x_n - \mu_{near}\| > \eta_n
\]  

(2)

where \( \mu_{near} \) is the neuron centre nearest to the current input \( x_n \), and \( \eta_n \) indicates the scale of neuron resolution in network.

**Condition 2:** the network prediction error for the current learning observation is significant:

\[
e_n = z_n - f(x_n), \quad |e_n| > E
\]  

(3)

where \( E \) denotes the desired accuracy of the network.

**Condition 3:** the prediction error within a sliding window \( W \) is significant:

\[
\sqrt{\frac{\sum_{i=n-(W-1)}^n e_i^2}{W}} > E_{rms}
\]  

(4)

where \( E_{rms} \) is the threshold to ensure smooth network growth.

The algorithm begins with \( \eta_n = \eta_{max}, \eta_{max} \) being chosen as the largest scale of interest in the input space. It decays exponentially according to

\[
\eta_n = \max\{\eta_{max} \gamma^n, \eta_{min}\}, \quad 0 < \gamma < 1,
\]  

(5)

until it reaches \( \eta_{min} \). The decay constant \( \gamma \) indicates the speed of the process, and \( \eta_{min} \) represents the desired neuron resolution of the network. The exponential decay of the distance criterion allows fewer neurons with large widths (smoother basis functions) initially. With increasing number of observations, more functions with smaller widths are recruited to refine the approximation.

When the three conditions are met, a new neuron is inserted into the current network of \( K \) units, with the following parameters at the position coincident at input \( x_n \):

\[
\begin{align*}
\mu_{K+1} &= x_n \\
a_{K+1} &= e_n \\
\sigma_{K+1} &= \psi \|x_n - \mu_{near}\|
\end{align*}
\]  

(6)

where \( \psi \) is an overlap factor between hidden units, and the number of units \( K + 1 \) is renamed as \( K \).

If an observation does not satisfy the novelty criteria Eq. (2) \( \sim \) Eq. (4), no new hidden unit is added. Instead, an EKF-based learning method, described below, is utilised to adjust the network parameters to best fit the current observation.

2.2. Network learning by neighbourhood EKF

Compared to classical least mean squares or gradient descent (GD) [5, 7] methods for updating parameters in nonlinear networks, extended Kalman filters (EKF) have been shown to be more compact and of better accuracy [8, 9]. However, the EKF usually updates all network parameters for all neurons. We therefore refer to it as global EKF (GEKF). The computational complexity of the GEKF is \( O(A^2) \) per learning step, where \( A \) denotes the number of parameters [8]. In our case, each neuron centre \( \mu = [\mu_x, \mu_y]^T \) is two dimensional, while width \( \sigma \) and weight \( \alpha \) are one dimensional. The computational cost of global EKF is therefore \( O((4K)^2) \) per learning step for updating all \( K \) hidden units.

To reduce computational cost, we utilise a local approach, called neighbourhood EKF (NEKF). At each learning step, only a subset of network parameters of \( B_n \) neighbour neuron(s) around the \( n \)th observation are updated. A neighbour neuron is selected, if it is 1) the nearest neighbour to the current observation; or 2) within a distance threshold \( D \) from the current observation. The computational cost of neighbourhood EKF is reduced to \( O((4B_n)^2) \). The minimum cost can be \( O(\text{constant}) \) in the extreme case when only the first criterion is applied \( (B_n = 1) \) to update the nearest neighbour at each learning step. Experimental results show that this improvement leads to considerable reduction of the computation load in network training compared to the GEKF or GD. It also obtains good accuracy, competitive with the GEKF and the GD, by consistently employing further-to-nearer refinement towards local fidelity.

2.3. Network pruning

The network size can become large when using only the above growth strategy, possibly leading to network overfit. To avoid this and obtain a compact network, a pruning strategy is applied [6]. Pruning removes those hidden units that make an insignificant contribution to the network output over a number of consecutive training observations. The pruning strategy is defined as follows:

For the current observation \( \{x_n, z_n\} \), compute the outputs of each hidden unit \( o_k^n \), \( k = 1...K \):

\[
o_k^n = a_k \exp \left(-\frac{\|x_n - \mu_k\|^2}{\sigma_k^2}\right),
\]  

(7)

Calculate the normalised output values \( r_k^n \) over the largest absolute hidden unit output:
If \( r_k^n < \rho \) for \( W \) consecutive observations, then the \( k \)th node is removed, where \( \rho \) is the pruning threshold.

3. EXPERIMENTAL RESULTS

We have implemented the proposed adaptive RBF network algorithm in C++. We tested the algorithm on 3D surface reconstruction from the range images obtained from the Signal Analysis and Machine Perception Laboratory (SAMPL) of Ohio State University [10]. The range images were acquired by a Minolta scanner at image pixel resolution of \( 200 \times 200 \). The scattered data are typically irregularly sampled with varying spatial densities. They contain measurement noise and holes.

In order to apply the RBF network, actual surface data stored in a \( 200 \times 200 \) matrix associated with their locations were normalised and re-sampled in a random order. Such a normalised random sequence was used as a training set \( T = \{ (x_n, y_n, z_n) | x_n, y_n, z_n \in [0, 1], n = 1...N \} \) of \( N \) 2D observation inputs \( x_n = [x_n, y_n]^T \) and outputs \( z_n \).

3.1. Implicit surface reconstruction and mesh repair

Figure 1 shows examples of surface reconstruction. Typical network parameters used in the experiments were: desired accuracy \( E = 0.01 \) (Eq. 3); \( E_{\text{rms}} = 0.01 \) in a sliding window \( W = 1000 \) (Eq. 4); largest scale of interest \( \eta_{\text{max}} = 0.4 \) in the normalised space; neuron resolution \( \eta_{\text{min}} = 0.01 \) for an average distribution density 1/200 of the range data. The decay constant was set as \( \gamma = 0.999 \) (Eq. 5) for an even sparse-to-dense spread of neurons in network. We set \( \psi = 0.7 \) for better performance in the presence of high spatial frequency and relatively big geometric variation in the images. The pruning threshold was set to \( \rho = 0.001 \) for 1000 consecutive observations (Eq. 8). From Fig. 1, we observe that Gaussian RBF networks provide extraordinary interpolation/extrapolation capabilities to: 1) smoothly reconstruct surfaces from non-uniformly sampled data with highly variable spatial frequencies; 2) smoothly blend and repair between raw noisy data to fill irregular holes that are large compared to the geometric variation in surfaces; 3) smoothly extend surfaces where there is no data by extrapolation.

Figure 2 shows the effectiveness of pruning employed to control the steady network growth and size in the angel reconstruction example.

Figure 3 shows the distribution, width and weight of neurons in the resulting network representation. Neurons distributed in 2D network space are represented by circles in normalised space at display ratio \( 3:1 \). For 2D intuitive visualisation associated with 3D geometry of the underlying data, we display the neurons in 3D space with their corresponding vertical \( z \) values estimated from the network. In Fig. 3(a), we used the radius of a circle to indicate neuron width. The maximum width is 0.33 for the angel example. In Fig. 3(b), we used the diameter of a circle to represent the absolute value of a neuron weight, with the maximum weight 0.92. We observe that distribution and density of neurons are highly adaptive to the complexity of the underlying data. Some neurons with large widths or high weights could often be generated initially for smooth bases within specified data ranges. Although there is an inherent tendency by the greedy algorithm to favour absorption of lower frequencies before higher ones, smaller neurons consistently refine the smoothness against fidelity to the local data so as to guarantee a coarse-to-fine optimality.

3.2. Data compression

During adaptive learning, neurons are added or removed according to their novelty and can be adjusted iteratively at full-dimension, hinting at high adaptivity to the underlying data.
These provide a reduction option allowing the vertices in raw surface data to be modelled compactly with fewer neurons in a network within a user specified accuracy. Table 1 presents network compression results from the reconstructions in Fig. 1, in which \( N \) stands for the number of 3D points in each raw range image, and \( K \) denotes the number of neurons in the generated network representation. Storage compression rate is calculated by \( 3N / 4K \), since each neuron has 4 parameters. The prediction error \( \bar{e} \) of the network is the average of absolute errors on all training points. We observe that the reconstructed angel image with relatively smooth geometric variation achieved high compression with good accuracy, while the valve with highly variable spatial frequencies on sharp edges and plane surfaces was relatively harder to model with a compact and accurate network structure.

<table>
<thead>
<tr>
<th>range image</th>
<th>num. of points ( N )</th>
<th>num. of neurons ( K )</th>
<th>compression rate</th>
<th>prediction error ( \bar{e} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>angel</td>
<td>14,089</td>
<td>1345</td>
<td>7.9</td>
<td>.0059</td>
</tr>
<tr>
<td>valve</td>
<td>10,145</td>
<td>1498</td>
<td>5.1</td>
<td>.0069</td>
</tr>
</tbody>
</table>

**4. CONCLUSION**

We presented an adaptive RBF network using a heuristic learning strategy to interpret non-uniformly sampled scattered data and to complete surfaces from noisy range images. The network is established by adaptively locating neurons accompanied by a pruning strategy, reflecting the fidelity of underlying data. The full dimensionality of network parameters, corresponding to location, weight and width for each neuron, is refined iteratively. Compared to approaches using pre-defined fixed network structures, the greedy learning strategy provides a significant flexibility that is particularly suitable for highly variable spatial frequencies. It consequently guarantees the possibility of data compactness while maintaining a desired global accuracy derived from only a subset of novelty points. Additionally, the neighbourhood EKF learning algorithm leads to a remarkable reduction of computation load. Experimental results demonstrate that the network approach offers new potential for surface interpretation, geometric formulation and compression.

**5. REFERENCES**


