

# Extending Neural Networks for B-Spline Surface Reconstruction

G. Echevarría, A. Iglesias, and A. Gálvez\*

Department of Applied Mathematics and Computational Sciences, University of Cantabria, Avda. de los Castros, s/n, E-39005, Santander, Spain  
`iglesias@unican.es`

**Abstract.** Recently, a new extension of the standard neural networks, the so-called *functional networks*, has been described [5]. This approach has been successfully applied to the reconstruction of a surface from a given set of 3D data points assumed to lie on unknown Bézier [17] and B-spline tensor-product surfaces [18]. In both cases the sets of data were fitted using Bézier surfaces. However, in general, the Bézier scheme is no longer used for practical applications. In this paper, the use of B-spline surfaces (by far, the most common family of surfaces in surface modeling and industry) for the surface reconstruction problem is proposed instead. The performance of this method is discussed by means of several illustrative examples. A careful analysis of the errors makes it possible to determine the number of B-spline surface fitting control points that best fit the data points. This analysis also includes the use of two sets of data (the training and the testing data) to check for overfitting, which does not occur here.

## 1 Introduction

The problem of recovering the 3D shape of a surface, also known as *surface reconstruction*, has received much attention in the last few years. For instance, in [9, 20, 21, 23, 24] the authors address the problem of obtaining a surface model from a set of given cross-sections. This is a typical problem in many research and application areas such as medical science, biomedical engineering and CAD/CAM, in which an object is often known by a sequence of 2D cross-sections (acquired from computer tomography, magnetic resonance imaging, ultrasound imaging, 3D laser scanning, etc.).

Another different approach consists of reconstructing surfaces from a given set of data points (see, for example, [10, 14, 15, 19, 22, 27]). In this approach, the goal of the surface reconstruction methods can be stated as follows: *given a set of sample points  $X$  assumed to lie on an unknown surface  $U$ , construct a surface model  $S$  that approximates  $U$* . This problem has been analyzed from several points of view, such as parametric methods [3, 4, 11, 27], function reconstruction [7, 28], implicit surfaces [19, 26], B-spline patches [22], etc.

---

\* This research was supported by the CICYT of the Spanish Ministry of Education (projects TAP1998-0640 and DPI2001-1288).

One of the most striking and promising approaches to this problem is that based on neural networks. After all, artificial neural networks have been recognized as a powerful tool for learning and simulating systems in a great variety of fields (see [8] and [12] for a survey). Since the behavior of the brain is the inspiration behind the neural networks, these are able to reproduce some of its most typical features, such as the ability to learn from data. This feature makes them specially valuable for solving problems in which one is interested in fitting a given set of data. For instance, the authors in [10] propose to fit surfaces through a standard neural network. Their approach is based on training the neural network to learn the relationship between the parametric variables and the data points. A more recent approach can be found in [13], in which a Kohonen neural network [16] has been applied to obtain free-form surfaces from scattered data. However, in this approach the network is used exclusively to order the data and create a grid of control vertices with quadrilateral topology. After this pre-processing step, any standard surface reconstruction method (such as those referenced above) has to be applied. Finally, a very recent work using a combination of neural networks and PDE techniques for the parameterization and reconstruction of surfaces from 3D scattered points can be found in [2].

It should be remarked, however, that the neural network scheme is not the “panacea” for the surface reconstruction problem. On the contrary, as shown in [17], some situations might require more sophisticated techniques. Among them, an extension of the “neural” approach based on the so-called *functional networks* has been recently proposed [5, 17]. These functional networks are a generalization of the standard neural networks in the sense that the weights are now replaced by neural functions, which can exhibit, in general, a multivariate character. In addition, when working with functional networks we are able to connect different neuron outputs at convenience. Furthermore, different neurons can be associated with neural functions from different families of functions. As a consequence, the functional networks exhibit more flexibility than the standard neural networks [5]. The performance of this new approach has been illustrated by its application to fit given sets of data from Bézier [17] and B-spline tensor-product surfaces [18].

In spite of these good results, the previous scheme is very limited in practice because the sets of data were fitted by means of Bézier surfaces in both cases. This is a drastic limitation because, in general, the Bézier scheme is not longer used for practical applications. The (more flexible) piecewise polynomial scheme (based on B-spline and NURBS surfaces) is usually applied in surface modeling and industry instead. The present paper applies this recently introduced functional network methodology to fit sets of given 3D data points through B-spline surfaces. In Sect. 2 we briefly describe the B-spline surfaces. Then, in Sect. 3 the problem to be solved is introduced. Application of the functional network methodology to this problem is described in Sect. 4. Sect. 5 reports the results obtained from the learning process for different examples of surfaces as well as a careful analysis of the errors. It includes the use of two sets of data (the training and the testing data) to check for overfitting. As we will show, this

analysis makes it possible to determine the number of B-spline surface fitting control points that best fit the data points. Finally, Sect. 6 closes with the main conclusions of this work.

## 2 Some Basic Definitions

In this section we give some basic definitions required throughout the paper. A more detailed discussion about B-spline surfaces can be found in [25].

Let  $\mathcal{S} = \{s_0, s_1, s_2, \dots, s_{r-1}, s_r\}$  be a nondecreasing sequence of real numbers called *knots*.  $\mathcal{S}$  is called the *knot vector*. The *i*th B-spline basis function  $N_{ik}(s)$  of order  $k$  (or degree  $k - 1$ ) is defined by the recurrence relations

$$N_{i1}(s) = \begin{cases} 1 & \text{if } s_i \leq s < s_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

and

$$N_{ik}(s) = \frac{s - s_i}{s_{i+k-1} - s_i} N_{i,k-1}(s) + \frac{s_{i+k} - s}{s_{i+k} - s_{i+1}} N_{i+1,k-1}(s) \quad (2)$$

for  $k > 1$ . With the same notation, given a set of three-dimensional *control points*  $\{\mathbf{P}_{ij}; i = 0, \dots, m; j = 0, \dots, n\}$  in a bidirectional net and two knot vectors  $\mathcal{S} = \{s_0, s_1, \dots, s_r\}$  and  $\mathcal{T} = \{t_0, t_1, \dots, t_h\}$  with  $r = m + k$  and  $h = n + l$ , a B-spline surface  $\mathbf{S}(s, t)$  of order  $(k, l)$  is defined by

$$\mathbf{S}(s, t) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{P}_{ij} N_{ik}(s) N_{jl}(t), \quad (3)$$

where the  $\{N_{ik}(s)\}_i$  and  $\{N_{jl}(t)\}_j$  are the B-spline basis functions of order  $k$  and  $l$  respectively, defined following (1) and (2).

## 3 Description of the Problem

In this section we describe the problem we want to solve. It can be stated as follows: we look for the most general family of parametric surfaces  $\mathbf{P}(s, t)$  such that their isoparametric curves (see [6] for a description)  $s = \tilde{s}_0$  and  $t = \tilde{t}_0$  are linear combinations of the sets of functions:  $\mathbf{f}(s) = \{f_0(s), f_1(s), \dots, f_m(s)\}$  and  $\mathbf{f}^*(t) = \{f_0^*(t), f_1^*(t), \dots, f_n^*(t)\}$  respectively. In other words, we look for surfaces  $\mathbf{P}(s, t)$  such that they satisfy the system of functional equations

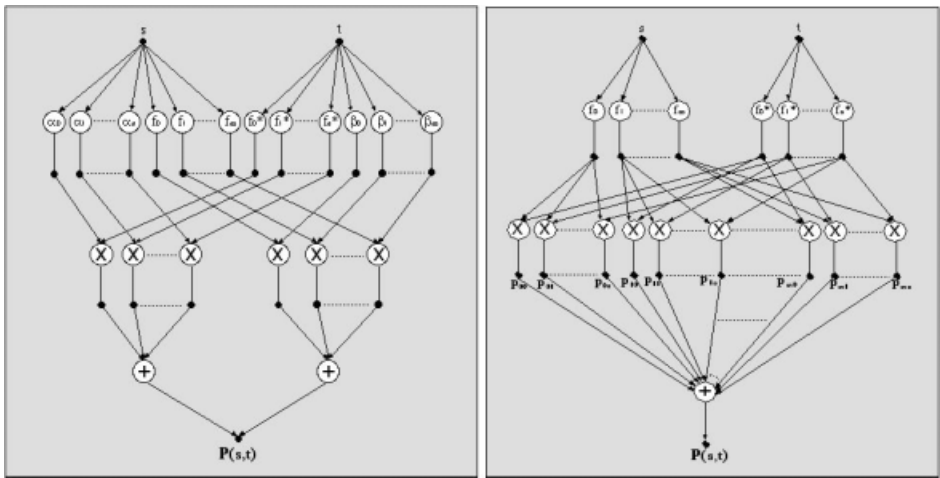
$$\mathbf{P}(s, t) \equiv \sum_{j=0}^n \alpha_j(s) f_j^*(t) = \sum_{i=0}^m \beta_i(t) f_i(s) \quad (4)$$

where the sets of coefficients  $\{\alpha_j(s); j = 0, 1, \dots, n\}$  and  $\{\beta_i(t); i = 0, 1, \dots, m\}$  can be assumed, without loss of generality, as sets of linearly independent functions.

This problem cannot be solved with simple standard neural networks: to represent it in terms of a neural network, we should allow some neural functions to be different, while the neural functions in neural networks are always identical. Moreover, the neuron outputs of neural networks are different; however, in our scheme, some neuron outputs in the example are coincident. This implies that the neural networks paradigm should be generalized to include all these new features, which are incorporated into the functional networks (see [5]). To be more precise, our problem is described by the functional network in Fig. 1(left) which can be simplified (see [17] for details) to the expression:

$$\mathbf{P}(s, t) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{P}_{ij} f_i(s) f_j^*(t) \quad (5)$$

where the  $\mathbf{P}_{ij}$  are elements of an arbitrary matrix  $\mathbf{P}$ ; therefore,  $\mathbf{P}(s, t)$  is a tensor product surface. Eq. (5) shows that the functional network in Fig. 1(left) can be simplified to the equivalent functional network in Fig. 1(right).



**Fig. 1.** (left) Graphical representation of a functional network for the parametric surface of Eq. (4); (right) Functional network associated with Eq. (5). It is equivalent to the functional network on the left

This functional network is then applied to solve the surface reconstruction problem described in Sect. 1. In order to check the flexibility of our proposal, we have considered sets of 256 3D data points  $\{T_{uv}; u, v = 1, \dots, 16\}$  (from here on, the *training points*) in a regular  $16 \times 16$  grid from four different surfaces. The first one (*Surface I*) is a B-spline surface given by (3) with the control points listed in Table 1,  $m = n = 5$ ,  $k = l = 3$  and nonperiodic knot vectors (according to the classification used in [1]) for both directions  $s$  and  $t$ . The other three surfaces

**Table 1.** Control points used to define *Surface I*

$(x, y, z)$	$(x, y, z)$	$(x, y, z)$	$(x, y, z)$	$(x, y, z)$	$(x, y, z)$
(0, 0, 1)	(0, 1, 2)	(0, 2, 3)	(0, 3, 3)	(0, 4, 2)	(0, 5, 1)
(1, 0, 2)	(1, 1, 3)	(1, 2, 4)	(1, 3, 2)	(1, 4, 3)	(1, 5, 2)
(2, 0, 1)	(2, 1, 4)	(2, 2, 5)	(2, 3, 5)	(2, 4, 4)	(2, 5, 3)
(3, 0, 3)	(3, 1, 4)	(3, 2, 5)	(3, 3, 1)	(3, 4, 2)	(3, 5, 3)
(4, 0, 2)	(4, 1, 2)	(4, 2, 4)	(4, 3, 4)	(4, 4, 3)	(4, 5, 2)
(5, 0, 1)	(5, 1, 2)	(5, 2, 3)	(5, 3, 3)	(5, 4, 2)	(5, 5, 1)

(labelled as *Surface II*, *III* and *IV*) are explicit surfaces defined by the equations  $z = y^3 - x^3 - y^2 + x^2 + xy$ ,  $z = 2(x^4 - y^4)$  and  $z = \frac{0.8y^2 - 0.5x^2}{x^2 + y^2 + 0.1}$ , respectively.

In order to check the robustness of the proposed method, the third coordinate of the 256 three-dimensional points  $(x_p, y_p, z_p)$  was slightly modified by adding a real uniform random variable  $\epsilon_p$  of mean 0 and variance 0.05. Therefore, in the following, we consider points given by  $(x_p, y_p, z_p^*)$ , where

$$z_p^* = z_p + \epsilon_p \quad , \quad \epsilon_p \in (-0.05, 0.05). \quad (6)$$

Such a random variable plays the role of a measure error to be used in the estimation step to learn the functional form of  $\mathbf{P}(s, t)$ .

## 4 Applying the functional network

To solve the problem described in the previous section, the neural functions of the network must be estimated (learned) by using some minimization method. In functional networks, this learning process consists of obtaining the neural functions based on a set of data  $D = \{(I_i, O_i) | i = 1, \dots, n\}$ , where  $I_i$  and  $O_i$  are the  $i$ -th input and output, respectively, and  $n$  is the sample size. To this end, each neural function  $f_i$  is approximated by a linear combination of functions in a given family  $\{\phi_{i1}, \dots, \phi_{im_i}\}$ . Thus, the approximated neural function  $\hat{f}_i(\mathbf{x})$  becomes

$$\hat{f}_i(\mathbf{x}) = \sum_{j=1}^{m_i} a_{ij} \phi_{ij}(\mathbf{x}), \quad (7)$$

where  $\mathbf{x}$  are the inputs associated with the  $i$ -th neuron. In the case of our example, the problem of learning the above functional network merely requires the neuron functions  $x(s, t)$ ,  $y(s, t)$  and  $z(s, t)$  to be estimated from a given sequence of triplets  $\{(x_p, y_p, z_p), p = 1, \dots, 256\}$  which depend on  $s$  and  $t$  so that  $x(s_p, t_p) = x_p$  and so on. For this purpose we build the sum of squared errors function:

$$Q_\alpha = \sum_{p=1}^{256} \left( \alpha_p - \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} a_{ij} \phi_i(s_p) \psi_j(t_p) \right)^2 \quad (8)$$

where, in the present example, we must consider an error function for each variable  $x$ ,  $y$  and  $z$ . This is assumed by  $\alpha$  in the previous expression, so (8) must be interpreted as three different equations, for  $\alpha = x$ ,  $y$  and  $z$  respectively. Applying the Lagrange multipliers to (8), the optimum value is obtained for

$$\frac{\partial Q_\alpha}{2\partial a_{\gamma\mu}} = \sum_{p=1}^{256} \left( \alpha_p - \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} a_{ij} \phi_i(s_p) \psi_j(t_p) \right) \phi_\gamma(s_p) \psi_\mu(t_p) = 0 \quad (9)$$

$$\gamma = 0, 1, \dots, M-1 \quad ; \quad \mu = 0, 1, \dots, N-1.$$

On the other hand, a B-spline function is basically a piecewise polynomial function whose number of spans  $r$  is given by  $r = m + k - 1$ , where  $m$  and  $k$  are the number of the control points and the order, respectively. Hence, we need to make a decision between the following two possibilities:

- to fix the number of control points and to change the order of the B-spline or
- to fix the order and then change the number of the control points.

In this paper we have considered the second option: to fit the 256 data points of our examples we have used nonperiodic third-order B-spline basis functions  $\{N_{i3}(s)\}_i$  and  $\{N_{j3}(t)\}_j$ , that is, we have chosen  $\{\phi_i(s) = N_{i3}(s) | i = 0, 1, \dots, M-1\}$  and  $\{\psi_j(t) = N_{j3}(t) | j = 0, 1, \dots, N-1\}$  in (8). We remark that this choice is very natural: the B-spline functions are frequently used in the framework of both surface reconstruction and approximation theory. In particular, the third-order B-spline functions are the most common curves and surfaces in research and industry. Finally, nonperiodic knot vectors mean that we force the B-spline surfaces to pass through the corner points of the control net, a very reasonable constraint in surface reconstruction.

Therefore, we allow the parameters  $M$  and  $N$  in (9) to change. Of course, every different choice for  $M$  and  $N$  yields to the corresponding system (9), which must be solved. Note that, since third-order B-spline functions are used, the minimum value for  $M$  and  $N$  is 3. However, this value implies that the B-spline surface is actually a Bézier surface [1], so we have taken values for  $M$  and  $N$  from 4 to 8. Solving the system (9) for all these cases, we obtain the control points associated with the B-spline surfaces fitting the data. The results will be discussed in the next section.

## 5 Results

To test the quality of the model we have calculated the mean and the root mean squared (RMS) errors for  $M$  and  $N$  from 4 to 8 and for the 256 training data points from the four surfaces described in Sect. 3.

Table 2 refers to *Surface I*. As the reader can appreciate, the errors (which, of course, depend on the values of  $M$  and  $N$ ) are very small, indicating that

the approach is reasonable. The best choice (indicated in bold in Table 2) corresponds to  $M = N = 6$ , as expected because data points come from a B-spline surface defined through a net of  $6 \times 6$  control points. In this case, the mean and the RMS errors are 0.0085 and 0.00071 respectively.

**Table 2.** Mean and root mean squared errors of the  $z$ -coordinate of the 256 training points from the *Surface I* for different values of  $M$  and  $N$

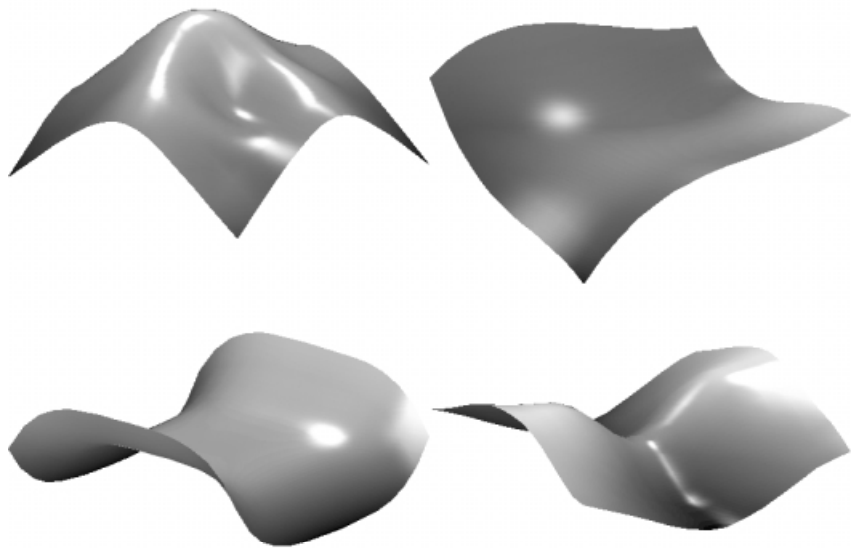
	$N = 4$	$N = 5$	$N = 6$	$N = 7$	$N = 8$
$M = 4$	0.1975 0.00919	0.1000 0.00798	0.0941 0.00762	0.0945 0.00764	0.0943 0.00763
$M = 5$	0.1229 0.00873	0.0939 0.00743	0.0885 0.00700	0.0888 0.00703	0.0886 0.00702
$M = 6$	0.0676 0.00528	0.0354 0.00265	<b>0.0085</b> <b>0.00071</b>	0.0115 0.00090	0.0093 0.00082
$M = 7$	0.0691 0.00547	0.0387 0.00301	0.0208 0.00163	0.0221 0.00172	0.0217 0.00168
$M = 8$	0.0678 0.00531	0.0356 0.00270	0.0117 0.00093	0.0139 0.00109	0.0131 0.00103

**Table 3.** Control points of the reconstructed *Surface I*

$(x, y, z)$	$(x, y, z)$	$(x, y, z)$	$(x, y, z)$	$(x, y, z)$	$(x, y, z)$
(0, 0, 1.0382)	(0, 1, 1.9897)	(0, 2, 3.047)	(0, 3, 2.9435)	(0, 4, 2.0411)	(0, 5, 0.9777)
(1, 0, 2.0048)	(1, 1, 2.9945)	(1, 2, 4.0228)	(1, 3, 1.9602)	(1, 4, 2.9981)	(1, 5, 2.0028)
(2, 0, 1.007)	(2, 1, 3.9777)	(2, 2, 4.9951)	(2, 3, 5.0357)	(2, 4, 3.9554)	(2, 5, 3.0221)
(3, 0, 2.9866)	(3, 1, 4.004)	(3, 2, 5.0283)	(3, 3, 0.9122)	(3, 4, 2.0926)	(3, 5, 2.968)
(4, 0, 2.0302)	(4, 1, 1.9729)	(4, 2, 4.0344)	(4, 3, 4.004)	(4, 4, 2.9637)	(4, 5, 2.0087)
(5, 0, 0.9757)	(5, 1, 2.0232)	(5, 2, 3.0047)	(5, 3, 3.009)	(5, 4, 1.9567)	(5, 5, 1.0423)

Table 3 shows the control points for the reconstructed *Surface I* corresponding to the best case  $M = N = 6$ . They were obtained by solving the system (9) with a floating-point precision and removing the zeroes when redundant. A simple comparison with Table 1 shows that the corresponding  $x$  and  $y$  coordinates are exactly the same, which is as expected because they were not affected by the noise. On the contrary, since noise was applied to the  $z$  coordinate, the corresponding values are not obviously the same but very similar, indicating that we have obtained a very good approximation. The approximating surface is shown in Fig. 2(top-left) and it is virtually indistinguishable from the original surface.

To cross validate the model we have also used the fitted model to predict a new set of 1024 testing data points, and calculated the mean and the root mean squared (RMS) errors, obtaining the results shown in Table 4. The new results confirm our previous choice for  $M$  and  $N$ . A comparison between mean and



**Fig. 2.** (top-bottom, left-right) B-spline approximating surfaces of the surfaces labelled *Surface I*, *II*, *III* and *IV*, respectively. Their corresponding equations are described in Sect. 3

RMS error values for the training and testing data shows that, for our choice, they are comparable. Thus, we can conclude that no overfitting occurs. Note that a variance for the training data significantly smaller than the variance for the testing data is a clear indication of overfitting. This does not occur here.

**Table 4.** Mean and root mean squared errors of the  $z$ -coordinate of the 1024 testing points from the *Surface I* for different values of  $M$  and  $N$

	$N = 4$	$N = 5$	$N = 6$	$N = 7$	$N = 8$
$M = 4$	0.1118 0.00441	0.0943 0.00384	0.0887 0.00366	0.0889 0.00367	0.0889 0.00366
$M = 5$	0.10599 0.00422	0.0888 0.00363	0.0830 0.00342	0.0833 0.00343	0.0832 0.00342
$M = 6$	0.0649 0.00252	0.0341 0.00130	<b>0.0078</b> <b>0.00032</b>	0.0109 0.00042	0.0093 0.00038
$M = 7$	0.0668 0.00263	0.0381 0.00149	0.0203 0.00081	0.0216 0.00085	0.0213 0.00084
$M = 8$	0.0651 0.00253	0.0345 0.00133	0.0111 0.00043	0.0133 0.00051	0.0125 0.00049



A similar analysis was carried out for the other surfaces described in Sect. 3. The corresponding tables of results are not included here because of limitations of space. It is enough to say that the best choice of  $M$  and  $N$  for the *Surface II* is  $M = N = 4$ . The mean and RMS errors are 0.0052 and 0.00040 respectively for the training points and 0.0049 and 0.00019 for the testing points. The *Surface III* has a more complex shape so larger values of  $M$  and  $N$ , namely  $M = 6$  and  $N = 7$ , are required for the best fitting. For these values the mean and RMS errors are 0.0095 and 0.00073 respectively for the training points and 0.0090 and 0.00035 for the testing points. Finally, *Surface IV* is best fitted for  $M = N = 7$ . In this last case, the mean and RMS errors are 0.0143 and 0.00114 respectively for the training points and 0.0139 and 0.00055 for the testing points. The approximating B-spline surfaces are displayed in Fig. 2.

## 6 Conclusions

In this paper a powerful extension of neural networks, the so-called functional networks, has been applied to the surface reconstruction problem. Given a set of 3D data points, the functional network returns the control points and the degree of the B-spline surface that best fits these data points. We remark that the data points do not necessarily have to belong to a parametric surface. In fact, some examples of the performance of this method for both parametric and explicit surfaces have been given. A careful analysis of the error as a function of the number of the control points has also been carried out. The obtained results show that all these new functional networks features allow the surface reconstruction problem to be solved in several cases. Nevertheless, in order to assess the limitations of our proposal further research is required. This future work will be reported elsewhere.

## References

1. Anand, V.: Computer Graphics and Geometric Modeling for Engineers. John Wiley and Sons, New York (1993)
2. Barhak, J., Fischer, A.: Parameterization and reconstruction from 3D scattered points based on neural network and PDE techniques. IEEE Trans. on Visualization and Computer Graphics **7**(1) (2001) 1-16
3. Bolle, R.M., Vemuri, B.C.: On three-dimensional surface reconstruction methods. IEEE Trans. on Pattern Analysis and Machine Intelligence **13**(1) (1991) 1-13
4. Brinkley, J.F.: Knowledge-driven ultrasonic three-dimensional organ modeling. IEEE Trans. on Pattern Analysis and Machine Intelligence **7**(4) (1985) 431-441
5. Castillo, E.: Functional Networks. Neural Processing Letters **7** (1998) 151-159
6. Farin, G.E.: Curves and Surfaces for Computer-Aided Geometric Design (Fifth Edition). Morgan Kaufmann, San Francisco (2001)
7. Foley, T.A.: Interpolation to scattered data on a spherical domain. In: Cox, M., Mason, J. (eds.), Algorithms for Approximation II, Chapman and Hall, London (1990) 303-310

8. Freeman, J.A.: *Simulating Neural Networks with Mathematica*. Addison Wesley, Reading, MA, (1994)
9. Fuchs, H., Kedem, Z.M., Uselton, S.P.: Optimal surface reconstruction from planar contours. *Communications of the ACM*, **20**(10) (1977) 693-702
10. Gu, P., Yan, X.: Neural network approach to the reconstruction of free-form surfaces for reverse engineering. *CAD*, **27**(1) (1995) 59-64
11. Hastie, T., Stuetzle, W.: Principal curves. *JASA*, **84** (1989) 502-516
12. Hertz, J., Krogh, A., Palmer, R.G.: *Introduction to the Theory of Neural Computation*. Addison Wesley, Reading, MA (1991)
13. Hoffmann, M., Varady, L.: Free-form surfaces for scattered data by neural networks. *Journal for Geometry and Graphics*, **2** (1998) 1-6
14. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. *Proc. of SIGGRAPH'92, Computer Graphics*, **26**(2) (1992) 71-78
15. Hoppe, H.: *Surface reconstruction from unorganized points*. Ph. D. Thesis, Department of Computer Science and Engineering, University of Washington (1994)
16. Kohonen, T.: *Self-Organization and Associative Memory* (3rd. Edition). Springer-Verlag, Berlin (1989)
17. Iglesias, A., Gálvez, A.: A new Artificial Intelligence paradigm for Computer-Aided Geometric Design. In: *Artificial Intelligence and Symbolic Computation*. Campbell, J. A., Roanes-Lozano, E. (eds.), Springer-Verlag, *Lectures Notes in Artificial Intelligence*, Berlin Heidelberg **1930** (2001) 200-213.
18. Iglesias, A., Gálvez, A.: Applying functional networks to fit data points from B-spline surfaces. In: *Proceedings of the Computer Graphics International, CGI'2001*, Ip, H.H.S., Magnenat-Thalmann, N., Lau, R.W.H., Chua, T.S. (eds.) IEEE Computer Society Press, Los Alamitos, California (2001) 329-332
19. Lim, C., Turkiyyah, G., Ganter, M., Storti, D.: Implicit reconstruction of solids from cloud point sets. *Proc. of 1995 ACM Symposium on Solid Modeling*, Salt Lake City, Utah, (1995) 393-402
20. Meyers, D., Skinnwer, S., Sloan, K.: Surfaces from contours. *ACM Transactions on Graphics*, **11**(3) (1992) 228-258
21. Meyers, D.: *Reconstruction of Surfaces from Planar Sections*. Ph. D. Thesis, Department of Computer Science and Engineering, University of Washington (1994)
22. Nilroy, M., Bradley, C., Vickers, G., Weir, D.:  $G^1$  continuity of B-spline surface patches in reverse engineering. *CAD*, **27**(6) (1995) 471-478
23. Park, H., Kim, K.: 3-D shape reconstruction from 2-D cross-sections. *J. Des. Mng.*, **5** (1997) 171-185
24. Park, H., Kim, K.: Smooth surface approximation to serial cross-sections. *CAD*, **28** (12) (1997) 995-1005
25. Piegl, L., Tiller, W.: *The NURBS Book* (Second Edition). Springer Verlag, Berlin Heidelberg (1997)
26. Pratt, V.: Direct least-squares fitting of algebraic surfaces. *Proc. of SIGGRAPH'87, Computer Graphics*, **21**(4) (1987) 145-152
27. Schmitt, F., Barsky, B.A., Du, W.: An adaptive subdivision method for surface fitting from sampled data. *Proc. of SIGGRAPH'86, Computer Graphics*, **20**(4) (1986) 179-188
28. Sclaroff, S., Pentland, A.: Generalized implicit functions for computer graphics. *Proc. of SIGGRAPH'91, Computer Graphics*, **25**(4) (1991) 247-250