Approximation by NURBS curves with free knots

M. Randrianarivony G. Brunnett

Technical University of Chemnitz, Faculty of Computer Science
Computer Graphics and Visualization
Straße der Nationen 62, 09107 Chemnitz, Germany
Email: maharavo@informatik.tu-chemnitz.de

Abstract

This paper is concerned with approximating noisy samples by NURBS curves with special emphasis on free knots. We consider the knots as unknown parameters so as to find their optimal positions. The original problem which is linear with respect to the weights and the control points but is nonlinear with respect to the knots is reformulated such that the knots are the only variable set. We show how to set up the problem such that nonlinear optimization methods can be applied efficiently. This involves the introduction of penalizing terms in order to avoid undesired knot positions. We report on our implementation of the nonlinear optimization. The performance of our method are confirmed by several practical examples. The generalization to the surface case will be briefly described at the end.

1 Introduction

NURBS settings are appreciated in many theoretical analyses because they allow flexible description of both free form surfaces and usual geometries such as conic sections. Indeed, the set of rational functions is much larger than that of polynomial functions, so NURBS give mainly better approximation than their B-spline counterparts do. Another reason for the appreciation of NURBS is that it is supported by many softwares. For instance, OpenGL and ACIS ([9], [2]) have built-in commands for drawing NURBS by only giving the required parameters.

Our interest in the subject of approximation with NURBS is motivated by the application of reverse engineering. In reverse engineering, one is concerned with the automated generation of a CAD-model from a set of points digitized from an existing 3D object.

Since many real world objects have been constructed using both simple algebraic surfaces as well as free-form surfaces, NURBS surfaces appear to be a universal class for surface fitting in reverse engineering.

In this paper we consider the curve case as a preliminary investigation for the surface case. We assume that a sequence of noisy sample points $\mathbf{M}_i$ is given and we aim at reconstructing a NURBS curve $\mathbf{X}$ that approximates the points in a least-square sense.

It is well known (see [6]) that the choice of the knot vector of a spline (also called parameterization) has a tremendous influence on the result of the fitting procedure. For this reason, several suggestions for a reasonable knot spacing have been made (see works of Foley, Nielson, Lee which are referenced in [6]). However, for all these suggestions of knot spacings, examples can be found where these methods provide unsatisfactory results. Furthermore, these methods can only be applied for interpolatory splines while we are interested in spline approximation. Therefore, for reliable results, one has to treat the knots as unknowns in the approximation process.

In the context of polynomial B-splines, the use of free knots has already been investigated by several authors ([12], [7], [11]). For the case of NURBS, the following approaches have been taken: in [3], the author uses an iterative segment determination in order to establish the positions of the knots. The authors of [8] use an improved version of Polak-Ribiere algorithm in order to minimize some cost functional without trying to reduce the number of parameters.

In this paper, we report on our implementation of a general method for approximation by NURBS curves with free knots. In section 2, we provide the necessary preliminaries to state the approximation

problem. In section 3, we reformulate the problem such that nonlinear optimization methods can be applied efficiently. For this, we reduce the dimensionality of the minimization problem and introduce penalizing terms in order to avoid undesired knot positions. In section 4, we describe in detail our implementation of the nonlinear optimization that is based on the Levenberg-Marquardt method. We discuss the results obtained by this approach for several data sets and different knot spacings in section 5. We will describe briefly the similarity for the case of surfaces in the last section.

2 Problem setting and notations

2.1 NURBS curve

A nonuniform rational B-spline curve (NURBS curve) with weights $w_0, ..., w_n \in \mathbf{R}_+$ and control points $\mathbf{d}_0, ..., \mathbf{d}_n \in \mathbf{R}^3$ is given by:

$$\mathbf{X}(t) := \frac{\sum_{i=0}^{n} w_i \mathbf{d}_i N_i^k(t)}{\sum_{i=0}^{n} w_i N_i^k(t)}, \qquad (1)$$

where N_i^k is the usual B-spline basis ([4]). Since we are mainly interested in open curves, we will assume that N_i^k is define on a knot sequence

$$\theta_0 = ... = \theta_{k-1}, \theta_k, ..., \theta_n, \theta_{n+1} = ... = \theta_{n+k}.$$

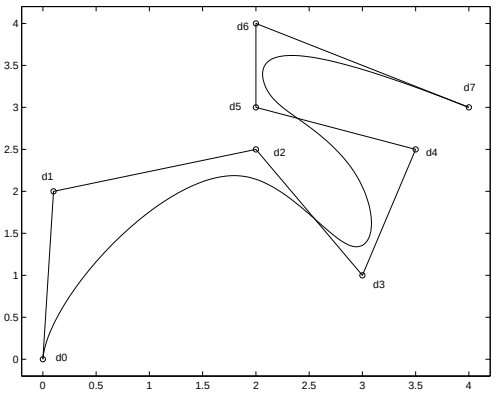

Figure 1: NURBS curve with $n = 7$, $k = 4$ $\mathbf{T} = (0, 0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1, 1)$ and $\mathbf{W} = (1.4, 0.5, 1.6, 1.8, 0.7, 1.9, 1.5, 0.9)$.

We will assume that

$$\theta_0 = 0\,, \quad \text{and} \quad \theta_{n+k} = 1\,.$$

In the sequel, we will denote:

$$\mathbf{W} := (w_0, ..., w_n) \qquad (2)$$
$$\mathbf{D} := (\mathbf{d}_0, ...\mathbf{d}_n) \qquad (3)$$
$$\mathbf{T} := (\theta_k, ..., \theta_n). \qquad (4)$$

An illustration can be found in Fig. 1.

2.2 Free knot problem

Suppose we are given a sequence of noisy samples $(t_i, \mathbf{M}_i)\ i = 0, ..., m$ with $k \ll m$. We want to find the NURBS curve $\mathbf{X}(t) = \mathbf{X}_{\mathbf{W},\mathbf{D},\mathbf{T}}(t)$ which fits these data best in a least square sense. Since we want to find the optimal positions of the knots, we put them as variables. That means, we have the following problem:

$$\min_{\mathbf{W},\mathbf{D},\mathbf{T}} \sum_{i=0}^{m} \left\| \mathbf{X}_{\mathbf{W},\mathbf{D},\mathbf{T}}(t_i) - \mathbf{M}_i \right\|^2 . \qquad (5)$$

This problem is too difficult to solve because of the nonlinear dependence of $\mathbf{X}$ on the parameters $(\mathbf{W}, \mathbf{D}, \mathbf{T})$. Furthermore, we need to add some constraints about the positivity of the weights. In the next section, we will show how to simplify this problem.

2.3 Brief recall of the fixed knot problem

If we are given a knot sequence $\mathbf{T}$, then the problem

$$\min_{\mathbf{W},\mathbf{D}} \sum_{i=0}^{m} \left\| \mathbf{X}_{\mathbf{W},\mathbf{D},\mathbf{T}}(t_i) - \mathbf{M}_i \right\|^2$$

will be referred to as fixed knot problem. It is investigated for example in [3] where it is shown to be equivalent to solving a linear system:

$$(A + \lambda B)\mathbf{y} = \lambda \mathbf{r}\,, \qquad (6)$$

where A and B are given in block structure:

$$A := \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},\ B := \begin{bmatrix} 0 & 0 \\ 0 & \tilde{B} \end{bmatrix}$$

$$A_{rs} := \begin{bmatrix} \Sigma \overline{N}_{00}^i \mathbf{Z}_i^{rs} & \cdots & \Sigma \overline{N}_{0n}^i \mathbf{Z}_i^{rs} \\ \vdots & & \vdots \\ \Sigma \overline{N}_{n0}^i \mathbf{Z}_i^{rs} & \cdots & \Sigma \overline{N}_{nn}^i \mathbf{Z}_i^{rs} \end{bmatrix} \qquad (7)$$

$$\mathbf{Z}_i^{(1,1)} := \mathbf{I} - c_i \mathbf{M}_i \mathbf{M}_i^T \quad (8)$$

$$\mathbf{Z}_i^{(1,2)} := c_i \mathbf{M}_i \quad (9)$$

$$\mathbf{Z}_i^{(2,1)} := c_i \mathbf{M}_i^T \quad (10)$$

$$\mathbf{Z}_i^{(2,2)} := 1 - c_i \quad (11)$$

$$\tilde{B} := \begin{bmatrix} \Sigma \overline{N}_{00}^i & \cdots & \Sigma \overline{N}_{0n}^i \\ \vdots & & \vdots \\ \Sigma \overline{N}_{n0}^i & \cdots & \Sigma \overline{N}_{nn}^i \end{bmatrix} \quad (12)$$

$$\begin{aligned} \mathbf{y} &:= [\bar{\mathbf{d}}_0, ..., \bar{\mathbf{d}}_n, w_0, ..., w_n]^T \\ \mathbf{r} &:= [\mathbf{0}, ..., \mathbf{0}, \Sigma N_0^k(t_i), ..., \Sigma N_n^k(t_i)]^T \\ \mathbf{I} &:= \text{identity matrix of order 3} \\ \bar{\mathbf{d}}_i &:= [w_i d_{ix}, w_i d_{iy}, w_i d_{iz}] \\ \overline{N}_{pq}^i &:= N_p^k(t_i) N_q^k(t_i) \\ c_i &:= 1/(1 + \mathbf{M}_i^2). \end{aligned}$$

In all these expressions, Σ is understood to be $\sum_{i=0}^{m}$ and λ is a positive constant which should be chosen large enough (see [3]) in order to ensure positivity of the weights.

3 Solving the free knot problem

The choice of the knot vector $\mathbf{T}$ has a large influence on the quality of the results of the curve fitting. It is well known that a bad placement of the knots may lead to overshooting effects that distort the shape of the curve. In this section, we describe our method of curve fitting that involves the determination of optimal knot positions.

3.1 Preparing the problem for nonlinear optimization

In the following, we set up the problem such that nonlinear optimization methods can be applied. According to section 2.3, for a given knot $\mathbf{T}$, we can solve the subproblem (6) in order to determine the corresponding weights $\mathbf{W}$ and the control points $\mathbf{D}$. In other words $\mathbf{W}$ and $\mathbf{D}$ are functions of $\mathbf{T}$ i.e.

$$(\mathbf{W}, \mathbf{D}) = (\mathbf{W}(\mathbf{T}), \mathbf{D}(\mathbf{T})).$$

Problem (5) is therefore simplified into:

$$\min_{\mathbf{T}} \sum_{i=0}^{m} \|\mathbf{X}_{\mathbf{W}(\mathbf{T}),\mathbf{D}(\mathbf{T}),\mathbf{T}}(t_i) - \mathbf{M}_i\|^2. \quad (13)$$

From now on, we will write only $\mathbf{X}(t_i)$ instead of $\mathbf{X}_{\mathbf{W}(\mathbf{T}),\mathbf{D}(\mathbf{T}),\mathbf{T}}(t_i)$ in order to simplify the notation. We have then

$$\min_{\mathbf{T}} \sum_{i=0}^{m} \|\mathbf{X}(t_i) - \mathbf{M}_i\|^2. \quad (14)$$

This problem still allows the presence of the situation where θ_i is not increasing. Therefore, we will modify this problem so that only knots with $\theta_k \leq \theta_{k+1} \leq ... \leq \theta_n$ may happen. By denoting:

$$\mathbf{X}(t) = (X_x(t), X_y(t), X_z(t)) \text{ and}$$
$$\mathbf{M}_i = (M_{ix}, M_{iy}, M_{iz}),$$

we have

$$\min_{\mathbf{T}} \sum_{i=0}^{m} (X_x(t_i) - M_{ix})^2 + (X_y(t_i) - M_{iy})^2 + (X_z(t_i) - M_{iz})^2.$$

By defining

$$\begin{cases} S_{3i} := X_x(t_i) - M_{ix} \\ S_{3i+1} := X_y(t_i) - M_{iy} \\ S_{3i+2} := X_z(t_i) - M_{iz}, \end{cases}$$

we obtain

$$\min_{\mathbf{T}} \sum_{i=0}^{3m+2} S_i^2. \quad (15)$$

We introduce now the function

$$R(x) := \begin{cases} 0 & \text{if} \quad x > 0 \\ (-x)^3 & \text{if} \quad x \leq 0, \end{cases}$$

and we define

$$R(\mathbf{T}) := R(\theta_{k+1} - \theta_k) + R(\theta_{k+2} - \theta_{k+1}) + ... + R(\theta_n - \theta_{n-1}). \quad (16)$$

Instead of (15), we will consider

$$\min_{\mathbf{T}} \sum_{i=0}^{3m+2} [S_i + \alpha R(\mathbf{T})]^2, \quad (17)$$

where α is a very large positive number. To understand the relationship between (15) and (17), we note the following two properties of equation (17).

- If we have $\theta_k \leq \theta_{k+1} \leq ... \leq \theta_n$, then $\theta_{k+r} - \theta_{k+r-1} \geq 0$ for all $r = 1, ..., n-k$ and therefore $R(\mathbf{T}) = 0$. Thus,

$$\sum_{i=0}^{3m+2} [S_i + \alpha R(\mathbf{T})]^2 = \sum_{i=0}^{3m+2} S_i^2 \,.$$

- If there is some r such that $\theta_{k+r} < \theta_{k+r-1}$, then $R(\theta_{k+r} - \theta_{k+r-1}) > 0$ and so $R(\mathbf{T})$ is nonzero. Because of our assumption that α is a very large number, we can expect that $\sum_{i=0}^{3m+2}[S_i + \alpha R(\mathbf{T})]^2$ is also very large.

Since we are searching for the minimum of $\sum_{i=0}^{3m+2}[S_i + \alpha R(\mathbf{T})]^2$, the preceeding two points show that a $\mathbf{T}$ with $\theta_{k+r} < \theta_{k+r-1}$ can never realize this minimum. That means that the integration of the trailing term in (17) penalizes those $\mathbf{T}$ with $\theta_{k+r} < \theta_{k+r-1}$.

In situation where it is desirable to have $|\theta_i - \theta_{i+1}| > \varepsilon$ for $i = k, ..., n-1$, we replace (16) by

$$R(\mathbf{T}) := R(\theta_{k+1} - \theta_k - \varepsilon) + R(\theta_{k+2} - \theta_{k+1} - \varepsilon) + ... + R(\theta_n - \theta_{n-1} - \varepsilon) \,. \tag{18}$$

3.2 Nonlinear optimization

By denoting $r(i, \mathbf{T}) := S_i + \alpha R(\mathbf{T})$ and $K = 3m + 2$, the curve fitting problem has the form of a usual nonlinear least square problem:

$$\min_{\mathbf{T}} \sum_{i=0}^{K} [r_i(i, \mathbf{T})]^2 \,.$$

Such a problem can be solved by nonlinear least square solvers like Levenberg-Marquardt and Gauss-Newton (see [10], [5]). Note that for each evaluation of the function $r(i, \mathbf{T})$, we need to solve the subproblem (6) in order to know the corresponding $\mathbf{W}(\mathbf{T})$, $\mathbf{D}(\mathbf{T})$. We note that the order of the linear system (6) is small. It does not depend on the number of data points. It depends exclusively on the degree n of the NURBS curves. Furthermore, taking into account that the support of N_i^k is $[\theta_i, \theta_{i+k})$, we conclude that the matrices in (7) are banded. More precisely, we have:

$$\forall \beta_i \quad \sum_{i=0}^{m} \overline{N}_{pq} \beta_i = 0 \quad \text{for } |p - q| \geq k.$$

As a consequence, the matrices are sparse and therefore we need only to compute a few entries. On the other hand, we must note that the computation of one entry of this system involves all data points. The remedy to that problem is to assemble the following matrix and vector

$$F := \begin{bmatrix} I - c_0 \mathbf{M}_0 \mathbf{M}_0^T \\ I - c_1 \mathbf{M}_1 \mathbf{M}_1^T \\ ... \\ I - c_m \mathbf{M}_m \mathbf{M}_m^T \end{bmatrix} \,, \quad \mathbf{c} := \begin{bmatrix} c_0 \\ c_1 \\ ... \\ c_m \end{bmatrix} \tag{19}$$

only once and store them in arrays so that they do not need to be recomputed in subsequent computations. Note that F and $\mathbf{c}$ are independent of $\mathbf{T}$. They depend only on the initial data points. The only expressions which need to be updated in each iteration of the nonlinear optimization are the values of $N_i^j(t_i)$ which are mostly zero except for some few values. They can be computed recursively in a fast way (see [1]).

4 Numerical results

4.1 Performance of the algorithm

The numerical results in this paper have been computed with the Levenberg-Marquardt algorithm. Note that this algorithm is iterative and so it needs some initial guess. The initial guess that we have taken here is equidistant knots in which the approximating curve is very far from the true curve. The first numerical test that we perform is the reconstruction of W-form curve. We have 75 data which were added with random noise of amplitude 0.1. The curve is reconstructed with the help of the formerly described algorithm. The overall time for the reconstruction is 8 seconds. In Figure 2 we see a graphical illustration of the data, the initial curve and the reconstructed curve. The second test is a free-form curve. The time needed for the reconstruction is like in the first test. A graphical illustration can be found in Figure 3.

4.2 Iteration vs. error

The next test consists in investigating the error after each iteration. This test is for the W-form curve in which the exact knot position is $\mathbf{T} = (0.333333, 0.5, 0.666666)$. In Table 1, we see

the value of $\mathbf{T}$ for each iteration and the corresponding error. We note that we do not need so much iterations in practice in order to achieve a nice accuracy. Note also that the reconstructed curve is already graphically satisfactory after iteration 4.

Table 1: Error for each iteration.

Iter	$\mathbf{T}(0)$	$\mathbf{T}(1)$	$\mathbf{T}(2)$	Error
1	0.25000	0.50000	0.75000	0.055556
2	0.26471	0.50124	0.71852	0.040575
3	0.29438	0.50293	0.69673	0.023980
4	0.31266	0.50099	0.67307	0.009359
5	0.32722	0.49908	0.66601	0.002561
6	0.33331	0.49992	0.66668	0.000037

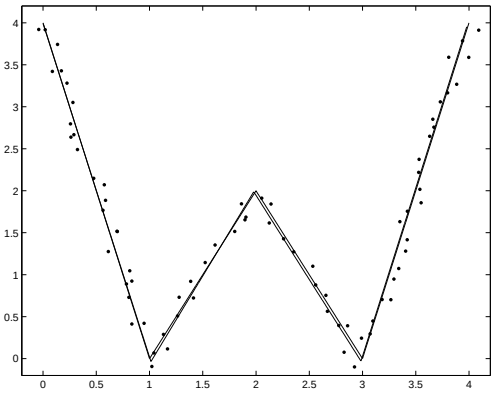

Figure 2: Initial curve, reconstructed curve and 75 samples with noise amplitude=0.1

4.3 Initial guess

The number of the iterations needed for this method depend on the initial guess that we take. That is again due to the fact that this is an iterative method. Although two different initial guesses give the same results, they may need different time to run the algorithm because more iterations mean a longer time of execution. Two possible initial guesses are: equidistant knot sequence and chord length knot sequence. The latter is meant in the sense that the interval $[0, 1]$ is parted into subintervals in such a way that each corresponding curve part is proportional

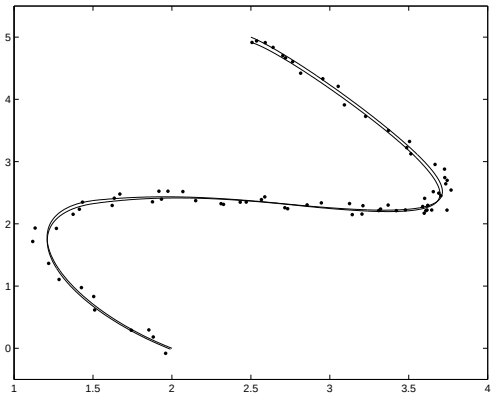

Figure 3: Initial curve, reconstructed curve and 75 samples with noise amplitude=0.1

to the length of the subinterval. Sometimes the first initial guess is good. For instance, the W-form curve (see Fig. 2) needs 6 iterations for equidistant initial guess. But it needs as many as 11 iterations for chord length initial guess to have the same results. The case of the curve in Fig. 3 is completely opposite to that. It needs 4 iterations for equidistant initial guess and 3 iterations for chord length initial guess. But as far as the final results are considered, both initial guesses give the same result. The difference can be found only in the number of iterations.

5 Similarity for surfaces

A nonuniform rational B-spline (NURBS) surface with weights $w_{i,j} \in \mathbf{R}_+$ and control points $\mathbf{d}_{i,j} \in \mathbf{R}^3$ $i = 0, \cdots, n_u$ $j = 0, \cdots, n_v$ is given by:

$$\mathbf{X}(u, v) := \frac{\sum_{i=0}^{n_u} \sum_{j=0}^{n_v} w_{i,j} \mathbf{d}_{i,j} N_i^{k_u}(u) N_j^{k_v}(v)}{\sum_{i=0}^{n_u} \sum_{j=0}^{n_v} w_{i,j} N_i^{k_u}(u) N_j^{k_v}(v)}, \tag{20}$$

where $N_i^{k_u}$ and $N_j^{k_v}$ are the usual B-spline basis ([4]) defined respectively on the knot sequences:

$$\begin{cases} \mu_i & i = 0, ..., n_u + k_u \\ \nu_j & j = 0, ..., n_v + k_v \end{cases}$$

In order to generalize the theory to the surface case, we introduce the following lexicographic ordering of the surface information:

$$\tilde{w}_{i(n_v+1)+j} := w_{i,j} \quad (21)$$

$$\tilde{\mathbf{d}}_{i(n_v+1)+j} := \mathbf{d}_{i,j} \quad (22)$$

$$\tilde{N}_{i(n_v+1)+j}(u,v) := N_i^{k_u}(u) N_j^{k_v}(v) \quad (23)$$

Note that the new expressions $\tilde{w}_s$, $\tilde{\mathbf{d}}_s$, $\tilde{N}_s(u,v)$ have only one index $s = 0, \cdots, n$ where $n := n_u(n_v+1) + n_v$, whereas the old ones $w_{i,j}$, $\mathbf{d}_{i,j}$, $N_i^{k_u}(u) N_j^{k_v}(v)$ have two indices $i = 0, \cdots, n_u$ $j = 0, \cdots, n_v$.

With the help of these new notations, the definition (20) becomes as simple as:

$$\mathbf{X}(u,v) = \frac{\sum_{s=0}^{n} \tilde{w}_s \tilde{\mathbf{d}}_s \tilde{N}_s(u,v)}{\sum_{s=0}^{n} \tilde{w}_s \tilde{N}_s(u,v)}, \quad (24)$$

We can notice right away that (24) looks very much like its curve counterpart (1).

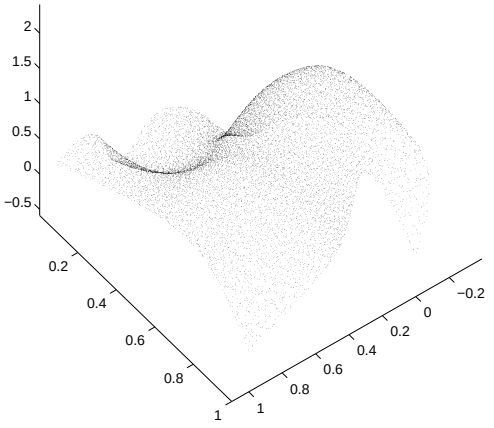

Figure 4: 12099 data points

The rest of the theory can be generalized in a similar way. In Fig. 4 and 5, we can illustrate the reconstruction of a surface from 12099 data points.

6 Future work

The most expensive part of this algorithm is the assembly of the matrices in (6). The assembly takes

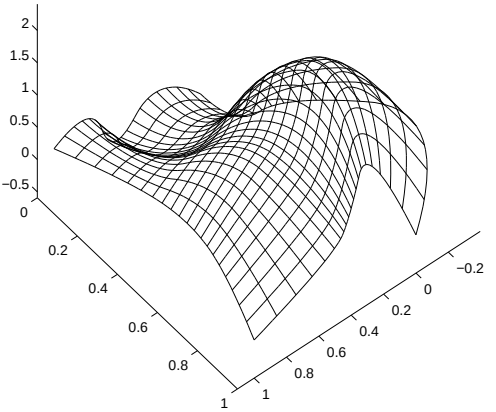

Figure 5: Reconstructed surface

in general more than 95% of the whole computational work. On the other hand, we can see that those matrices can be assembled in parallel because they consist only of sums of some terms which can be distributed on each processor. Our future work will deal with the parallelization of this algorithm in which each processor will have almost the same number of data points so as to ensure load balancing.

References

[1] C. de Boor, "*A practical guide to splines*", Springer, New York, 1978.

[2] J. Corney, "*3D modeling with ACIS kernel and toolkit*", John Wiley & sons, Chichester, 1997.

[3] B. Elsässer, "Approximation mit rationalen B-Spline Kurven und Flächen", *Ph.D. thesis*, Darmstadt, 1998.

[4] G. Farin, "*Curves and surfaces hoschekfor computer aided geometric design*", Academic Press, Boston, 2. ed., 1990.

[5] R. Fletcher, "*Practical methods of optimization*", John Wiley & Sons, Chichester, 2. ed., 1987.

[6] J. Hoschek, and D. Lasser "*Grundlagen der geometrischen Datenverarbeitung*", Teubner,

Stuttgart, 1989.

[7] D. Jupp, “Approximation to data by splines with free knots”, *SIAM J. Numer. Anal.*, pp. 328-343, vol 15, No. 2, 1978.

[8] P. Laurent-Gengoux, and M. Mekhilef, “Optimization of a NURBS representation”, *Computer Aided Design*, pp. 699-710, vol 25, No. 11, 1993.

[9] J. Neider, T. Davis, and M. Woo “*OpenGL programming guide*”, Addison-Wesley publishing company, Reading, 1994.

[10] J. Nocedal, and S. Wright “*Numerical Optimization*”, Springer Series in Operation Research, New York, 1999.

[11] T. Schütze, “Diskrete Quadratmittelapproximation durch Splines mit freien Knoten”, *Ph.D. thesis*, Dresden, 1998.

[12] H. Schwetlick, and T. Schütze, “Least squares approximation by splines with free knots”, *BIT*, pp. 361-384, vol 35, No. 3, 1995.