Reconstruction of Gradient in Volume Rendering

Željka Mihajlović, Leo Budin
University of Zagreb
Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{zeljka.mihajlovic, leo.budin}@fer.hr

Nikola Guid
University of Maribor
Faculty of Electrical Engineering and Computer Science
Smetanova 17, 2000 Maribor, Slovenia
guid@uni-mb.si

Abstract– **This paper deals with reconstruction of the gradient in the volume element space that is required for shading in the volume rendering. The accuracy of the surface reconstruction in the volume rendering is very important task as well as reconstruction accuracy of the normal vector. The central difference operator is often used as approximation of the gradient. The gradient reconstruction is an essential operation for determination of illumination and it has a great impact on the appearance of the reconstructed surface of an object. Our reconstruction of the gradient is based on the cubic approximation or interpolation B-splines.**
Volume rendering is visualization technique with widespread applicability in many fields, from the medical visualization to the engine visualization. Artefacts introduced by the surface reconstruction or the gradient reconstruction are of the crucial importance for perception of the projected object.
KEY WORDS – gradient, reconstruction, B-splines, volume rendering

1 Introduction

One of the most popular visualization technique is volume rendering. In the volume rendering, rays are cast in the volume element space. Volume data are sampled on the regular rectilinear grid and the reconstruction is required at arbitrary positions along the ray. Therefore, the reconstruction at an arbitrary position in the three-dimensional space according to the set of the discrete samples is required. The best solution is to find a continuous function, defined with a set of discrete samples. If one finds the continuous function, calculation of the new samples (resampling), or calculation of the derivative at any position is easy. For two or three dimensional extension, separability of the reconstruction filter is important property.

In the signal processing field, a conversion from the discrete representation to the continuous representation of the signal is usually applied by lowpass filtering. In computer graphics, the continuous curve is defined with the set of points. Essentially, it is the same problem in both fields, but the solutions are different. B-splines are the most popular in computer graphics and interpretation of the B-splines in the signal processing field brings a new light to the reconstruction problem.

In computer graphics, conversions from continuous to discrete representation, and vice versa, are inherited in many algorithms. The display devices usually have discrete coordinates and objects can be represented using discrete or continuous coordinates. Therefore, sampling or resampling is necessary in almost every procedure and it causes appearance of the alias artefacts on the result. Rotation of the image, texture mapping and ray tracing are some examples where sampling or resampling is typical. Therefore, reconstruction is one of fundamental operations in computer graphics.

To reconstruct a value at the arbitrary position, various reconstruction approaches can be applied. Much work has been done towards the design of reconstruction filters and error characterization [2], [3], [5]. Simple approaches are nearest neighbour or trilinear interpolation, but continuity of the reconstructed function is only C^0 and C^1, respectively. Better approaches for reconstruction are cubic spline, e. g. BC-splines introduced by Mitchell and Netravali [5], Catmull-Rom spline [6], or approximation and interpolation B-splines [7], [8]. Research of the reconstruction filters is done for one or two-dimensional case [1-3], [6-8]. The volumetric space is three-dimensional space, so extension should be made. In the volume visualization, one of the fundamental operations is reconstruction of the continuous function from a given set of samples.

Conversion from discrete to the continuous representation, or resampling, in the volume rendering occurs at several levels in the procedure, so aliasing artefacts superpose during the procedure and finally reveal in the resulting image. A discrete number of rays is cast in the volume element space, at the discrete positions on the ray, reconstruction is done, and according to the discrete samples, values are calculated. Therefore, conversion between the continuous and discrete representation is significant. Our attention is focused only on the reconstruction of the function value and gradient at the discrete positions along the ray. Our goal is to enhance the influence due to this reconstruction on the result.

According to the discrete samples in the volume element space, it is best to consider real-valued function f(x, y, z) defined over real domain $\boldsymbol{R}^3$. From the continuous representation, one can find the derivatives at arbitrary position.

Evaluation of the spatial gradient in two dimension space is important for an edge detection. Reconstruction of a derivative in the volume rendering is required for shading. Alias artefacts due to the reconstruction of the derivative also superpose to the final image. Bentum [1] presents an analysis of gradient estimators in frequency domain for cubic spline, but he does not consider interpolation and approximation B-splines. Although reconstruction can be correct, error in derivative reconstruction has high influence on the perception of the projected object.

In Section 2, we review some essential properties of continuous B-spline functions. The central difference operator and derivative of B-spline are described in

Section 3. Finally, multidimensional extension and results are presented in Sections 4 and 5.

2 B-splines

For interpolation or approximation curves and surfaces in computer graphics, B-splines are the most important. Recently [7], [8], B-splines are described in signal processing field as the direct and indirect B-splines transform.

The direct B-spline transform determines the spline coefficients required for interpolation. It is analogous to finding the control points (coefficients) for interpolation B-spline. The indirect one determines continuous function through a given set of points based on the calculated spline coefficients [7]. It is analogous to finding the approximation curve for a given set of points.

When it is important to interpolate a given set of samples, the direct B-spline should be applied. In a presence of noise in the input signal, low pass prefiltering is required. In that case, we can omit the direct B-spline transform, because the indirect B-spline transform approximates the input sequence and lowpass filtering is included.

For a given set of $(n+1)$ control points $\mathbf{r}_i$, the approximation B-spline curve $\mathbf{p}(t)$ is:

$$\mathbf{p}(t) = \sum_{i=0}^{n} \mathbf{r}_i \, \mathrm{N}_{i,k}(t), \tag{2.1}$$

where $\mathrm{N}_{i,\,k}$ are the basis or blending functions of degree k. For the sequence of the points $\mathbf{r}_i$, this formula defines the continuous curve, where $\mathbf{p}(t)$ is a point on the curve, for a given parameter t. Without lost of generality, we will focus our attention on the cubic case. The volumetric data are sampled on the regular, uniform and rectilinear grid. The B-spline with the uniform parameterization is a special case of B-spline that is appropriate for our problem.

The cubic uniform B-spline is a special case of (2.1) where i-th segment is:

$$\mathbf{p}_i(t) = \mathbf{T}_3 \cdot \mathbf{M}_{B,3} \cdot \mathbf{R}_3 \tag{2.2}$$

$\mathbf{T}_3$ is the parameter matrix, $\mathbf{M}_{\mathrm{B},3}$ is a matrix with constant coefficients (transformation matrix) and $\mathbf{R}_3$ is matrix of four control points. Equation (2.2) in detail is:

$$\mathbf{p}_i(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{r}_{i-1} \\ \mathbf{r}_i \\ \mathbf{r}_{i+1} \\ \mathbf{r}_{i+2} \end{bmatrix} \tag{2.3}$$

For the cubic B-spline, only four points have an effect on each segment.

Notice that for $t = 0$ and for $t = 1$:

$$\begin{aligned} \mathbf{p}_i(0) &= \frac{1}{6}\begin{bmatrix} 1 & 4 & 1 & 0 \end{bmatrix} \mathbf{R}_3 \\ \mathbf{p}_i(1) &= \frac{1}{6}\begin{bmatrix} 0 & 1 & 4 & 1 \end{bmatrix} \mathbf{R}_3 \end{aligned} \tag{2.4}$$

From the equations (2.4), we can see that at the boundary of i-th segment only three points are involved, but the curve does not interpolate any of them.

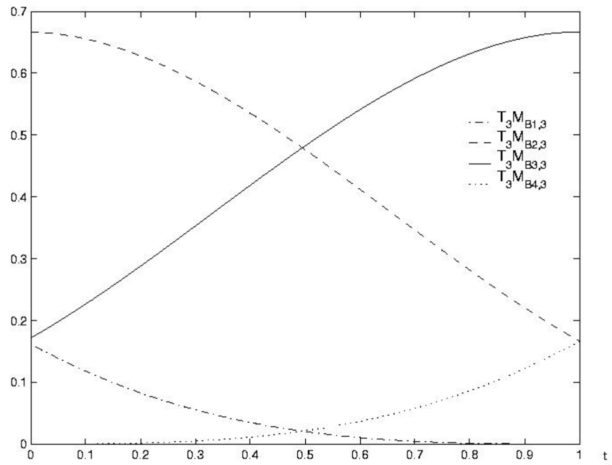

Fig. 1. Cubic bases function.

The basis functions $\mathrm{N}_{i,k}$ (t) of the degree k, from the formula (2.1), for the cubic case, become $\mathbf{T}_3\mathbf{M}_{\mathrm{B},3}$ in the formula (2.2). The basis function $\mathbf{T}_3\mathbf{M}_{\mathrm{B},3}$ is shown on Fig. 1 as four segments with respect to each control point $\mathbf{r}_{i-1}$, $\mathbf{r}_i$, $\mathbf{r}_{i+1}$, $\mathbf{r}_{i+2}$, where $t\in[0,1]$.

The cubic basis function is symmetric and it is the same for all segments i. The special case is only at the boundary of the sequence, thus some boundary condition should be applied. Phantom points could be added, or we can form closed curve by inserting the point $\mathbf{r}_\mathrm{n}$ as the first point $\mathbf{r}_{-1}$ and the point $\mathbf{r}_0$ as the last point $\mathbf{r}_{\mathrm{n}+1}$. In that case, even at the boundary, the basic function will be the same as (2.3).

We can represent the equation (2.3) in signal processing terms as convolution:

$$g^3(x) = y * \beta = \sum_{k=0}^{n} y(k)\beta^3(x-k), \tag{2.5}$$

where $y(k)$ is a sequence of the input points $\mathbf{r}_i$ and $\beta^3(x)$ is the filter kernel or the cubic B-spline basis function. In the signal processing, input is usually one dimensional $y(k)$, while, in computer graphics, $\mathbf{r}_i$ is a point in two or three dimensions, but we can take each coordinate; e. g. $x(t)$, separately.

The cubic B-spline approximation can be implemented as a FIR filter. The cubic B-spline basis function $\beta^3(x)$ is the filter kernel and it can be calculated from the equation (2.3) or it can be constructed recursively. For the cubic case, the filter kernel is [7]:

$$\beta^3(x) = \underbrace{\beta^0 * \ldots * \beta^0(x)}_{4\ times} \tag{2.6}$$

$\beta^0(x)$ is a centred, normalized, and rectangular pulse. The equation (2.6) is useful in efficiency optimization.

The same basis function $\beta^3(x)$ for all segments, even for boundary segments, enables the use of circular convolution.

The B-splines $\beta^i(x)$ of the degree 1 to 3 are shown on Fig. 2. For a larger degree of B-spline, the filter kernel support is wider. For the cubic B-spline, the kernel region of support is $[-2,2]$.

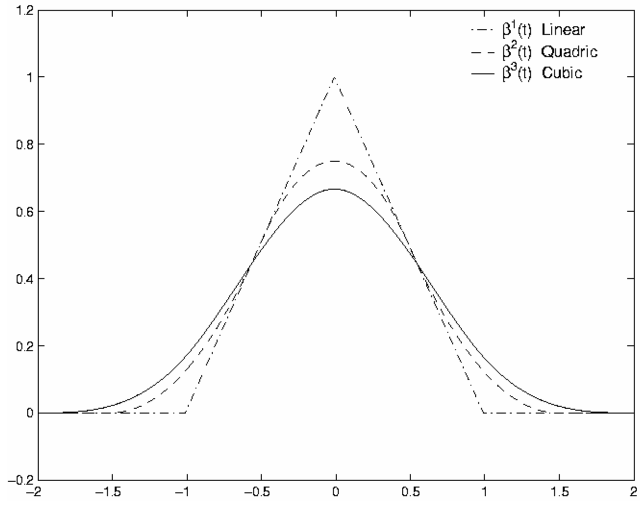

Fig. 2. B-splines of the degree 1 to 3.

The cubic B-spline basis function $\beta^3(x)$ is same as $\mathbf{T}_3\mathbf{M}_{B,3}$ in Fig. 1, except that four segments are shifted in appropriate intervals in the parameter space.

3 Gradient

The central difference operator is often used as approximation of the gradient. If the distance between sampled points is one, the central difference operator in three-dimensional space is:

$$\begin{aligned}&\nabla \mathrm{f}(\mathbf{x}_i) = \nabla \mathrm{f}(x_i \quad y_i \quad z_i) \approx \\ &\Big\{\frac{1}{2}\big[\mathrm{f}(x_{i+1} \quad y_j \quad z_k) - \mathrm{f}(x_{i-1} \quad y_j \quad z_k)\big], \\ &\frac{1}{2}\big[\mathrm{f}(x_i \quad y_{j+1} \quad z_k) - \mathrm{f}(x_i \quad y_{j-1} \quad z_k)\big], \\ &\frac{1}{2}\big[\mathrm{f}(x_i \quad y_j \quad z_{k+1}) - \mathrm{f}(x_i \quad y_j \quad z_{k-1})\big]\Big\}\end{aligned} \quad . \tag{3.1}$$

In the previous chapter, we found the approximation B-spline function, for the given set of points, defined with the expression (2.3) or (2.5). The gradient of the function is defined with the first derivative. Using (2.3), we can find the expression for the derivative of the cubic B-spline:

$$\frac{\partial \mathbf{p}_i(t)}{\partial t} = \frac{\partial \mathbf{T}_3}{\partial t} \cdot \mathbf{M}_{B,3}\mathbf{R}_3$$

$$\begin{aligned}\frac{\partial \mathbf{p}_i(t)}{\partial t} &= \begin{bmatrix} t^2 & t & 1 \end{bmatrix} \frac{1}{2} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -4 & 2 & 0 \\ -1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{r}_{i-1} \\ \mathbf{r}_i \\ \mathbf{r}_{i+1} \\ \mathbf{r}_{i+2} \end{bmatrix} \\ &= \mathbf{T}_2\mathbf{M}_{B,d3}\mathbf{R}_3\end{aligned} \tag{3.2}$$

Notice that for $t = 0$ and for $t = 1$:

$$\begin{aligned}\frac{\partial \mathbf{p}_i(0)}{\partial t} &= \frac{1}{2}\begin{bmatrix} -1 & 0 & 1 & 0 \end{bmatrix}\mathbf{R}_3 \\ \frac{\partial \mathbf{p}_i(1)}{\partial t} &= \frac{1}{2}\begin{bmatrix} 0 & -1 & 0 & 1 \end{bmatrix}\mathbf{R}_3\end{aligned} \tag{3.3}$$

At the boundary of i-th segment, only two points are involved. At the discrete point, the gradient is the same as frequently used discrete approximation of a differentiator, like in formula (3.1). But, this expression is only combination of three one dimensional derivative filters of the approximation B-spline at the segments boundaries. In the formula (3.1), two approximations are made. The first one is approximation of the derivative that is defined in the continuous domain (3.2), with derivative defined only at the boundary (3.3). The second approximation is in combination of three one-dimensional derivatives, instead three partial derivatives of the continuous three-dimensional B-spline.
For our consideration, it is important to have the gradient defined over the continuous domain, so it would be better to use the continuous case (3.2).

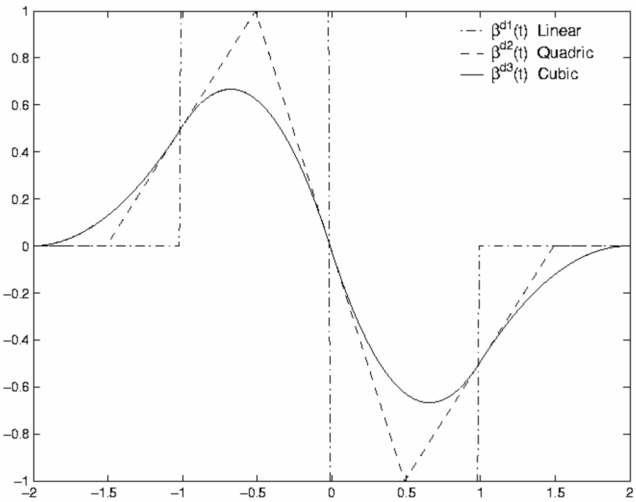

Fig. 3. Derivative filter kernels of the B-splines (the degree 1 to 3).

By using the property of B-splines:

$$\beta^{d3}\left(x + \frac{1}{2}\right) = \frac{\partial \beta^3\left(x + \frac{1}{2}\right)}{\partial x} = \beta^2(x+1) - \beta^2(x), \tag{3.4}$$

the derivative of the input sequence $y(k)$ is from the equation (2.5):

$$\frac{\partial g^3(x)}{\partial x} = \sum_{k=0}^{n-1} \big(y(k) - y(k-1)\big)\beta^2\left(x - k + \frac{1}{2}\right) \tag{3.5}$$

Fig. 3 shows the derivative filter kernels $\beta^{di}(x)$ for the B-splines that are presented on Fig. 2.
For the cubic B-spline approximation the continuity of derivative filter kernel is C^2. The region of support for derivative filter kernel is same as for cubic B-spline.

4 Multidimensional extension

A bicubic uniform surface patch is defined with:

$$\mathbf{p}_i(t,u) = \mathbf{T}_3 \cdot \mathbf{M}_{B,3} \cdot \mathbf{R}_{3,3} \cdot \mathbf{M}_{B,3}^{\tau} \mathbf{U}_3^{\tau}, \tag{4.1}$$

where $\mathbf{T}_3$ is the parameter matrix, $\mathbf{M}_{B,3}$ is a matrix with constant coefficients, same as in (2.3), and $\mathbf{R}_{3,3}$ is two-dimensional net of control points:

$$\mathbf{R}_{3,3} = \begin{bmatrix} \mathbf{r}_{i-1,j-1} & \mathbf{r}_{i-1,j} & \mathbf{r}_{i-1,j+1} & \mathbf{r}_{i-1,j+21} \\ \mathbf{r}_{i,j-1} & \mathbf{r}_{i,j} & \mathbf{r}_{i,j+1} & \mathbf{r}_{i,j+21} \\ \mathbf{r}_{i+1,j-1} & \mathbf{r}_{i+1,j} & \mathbf{r}_{i+1,j+1} & \mathbf{r}_{i+1,j+21} \\ \mathbf{r}_{i+2,j-1} & \mathbf{r}_{i+2,j} & \mathbf{r}_{i+2,j+1} & \mathbf{r}_{i+2,j+21} \end{bmatrix}. \tag{4.2}$$

$\mathbf{U}_3$ is same as $\mathbf{T}_3$ except for parameter u. The gradient operator in two-dimensional space is:

$$\nabla \mathbf{p}_i(t,u) = \begin{bmatrix} \frac{\partial \mathbf{p}_i(t,u)}{\partial t} & \frac{\partial \mathbf{p}_i(t,u)}{\partial u} \end{bmatrix}, \tag{4.3}$$

where

$$\begin{aligned} \frac{\partial \mathbf{p}_i(t,u)}{\partial t} &= \mathbf{T}_2 \cdot \mathbf{M}_{B,d3} \cdot \mathbf{R}_{3,3} \cdot \mathbf{M}_{B,3}^{\tau} \mathbf{U}_3^{\tau} \\ \frac{\partial \mathbf{p}_i(t,u)}{\partial u} &= \mathbf{T}_3 \cdot \mathbf{M}_{B,3} \cdot \mathbf{R}_{3,3} \cdot \mathbf{M}_{B,d3}^{\tau} \mathbf{U}_2^{\tau} \end{aligned}. \tag{4.4}$$

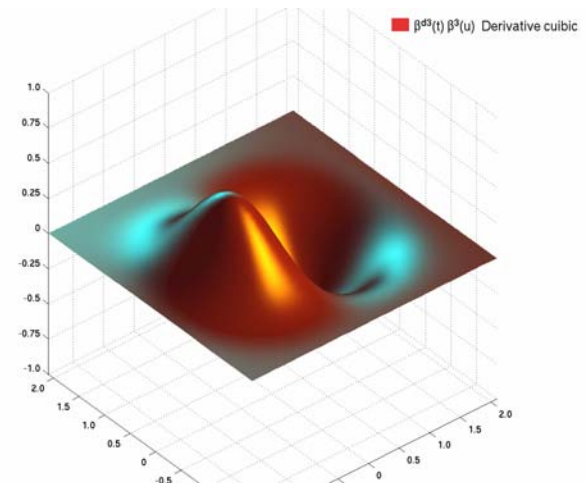

Fig. 4. Cubic derivative filter kernel of the B-spline in two dimensions.

For the parameter combination $t = 0$ and $u = 0$, the derivative is:

$$\frac{\partial \mathbf{p}_i(0,0)}{\partial t} = \frac{1}{2}\begin{bmatrix} -1 & 0 & 1 & 0 \end{bmatrix} \cdot \mathbf{R}_{3,3} \cdot \frac{1}{6}\begin{bmatrix} 1 \\ 4 \\ 1 \\ 0 \end{bmatrix}. \tag{4.5}$$

The matrix notation is inappropriate for an extension in three dimensions.

Let two dimensional mesh of uniformly spaced control points have $m \times n$ points. Two-dimensional circular convolution with the bicubic B-spline filter kernel is:

$$g^3(x, y) = \sum_{k=0}^{n} \sum_{l=0}^{m} y(k,l) \beta^3(x-k, y-l) \tag{4.6}$$

where we can compute two-dimensional basis B-spline or the filter kernel:

$$\beta^3(t,u) = \beta^3(t) \cdot \beta^3(u) \tag{4.7}$$

The region of support for this filter kernel is $[-2,2] \times [-2,2]$. The partial derivative filter kernel in the direction t for two-dimensional convolution is:

$$\frac{\partial \beta^3(t,u)}{\partial t} = \beta^{d3}(t) \cdot \beta^3(u) \tag{4.8}$$

The two-dimensional partial derivative filter kernel in one direction is shown on Fig. 4.
This concept could be expanded on three-dimensional case. The convolution form in three-dimensional space is:

$$g^3(x,y,z) = \sum_{k=0}^{n} \sum_{l=0}^{m} \sum_{j=0}^{p} y(k,l,j) \beta^3(x-k, y-l, z-j) \tag{4.9}$$

where the reconstruction filter kernel is:

$$\beta^3(t, u, w) = \beta^3(t) \cdot \beta^3(u) \cdot \beta^3(w) \tag{4.10}$$

Similarly as for two-dimensional case (4.8), we find the gradient of the B-spline function in three-dimensional space. The partial derivative filter kernel in the direction t for three-dimensional convolution is:

$$\beta^{t3}(t,u,w) = \frac{\partial \beta^3(t,u,w)}{\partial t} = \beta^{d3}(t) \cdot \beta^3(u) \cdot \beta^3(w) \tag{4.11}$$

The partial derivative of the reconstruction filter kernel for the parameter combination (t, u, w) defines the gradient at the certain position. We use this gradient for the illumination calculation.

5 Results

In volume rendering, reconstruction of the surface is required along the ray at arbitrary position, as well as reconstruction of the gradient. To improve the central difference operator (3.1), in our application, we use combination of three one dimensional derivatives (3.2) along each axis to find the gradient. The parameter combination (t, u, w) determines derivative calculation with formula (3.2) or (3.5) along (x, y, z) axis, respectively. We compare the improved central difference operator with reconstruction using the derivative of the three-dimensional approximation cubic B-spline.
Fig. 5. provides a comparison between the gradient estimation with the improved central difference operator (left) and the gradient that is achieved as the derivative of the B-spline function (4.11) (right). The reconstruction kernel for surface reconstruction, in both examples, is the same (4.10) (the approximation B-spline). The difference is obvious in perception of smoothness of the surface. Despite the same reconstruction procedure, in both

examples, reconstruction of the gradient has a drastic effect on the appearance of the surface.

6 Conclusion

The main objective of this paper has been to derive the gradient of the three-dimensional B-spline function. The gradient is used as the normal vector for illumination calculation in volume rendering. We compare our result with the result achieved when the gradient approximation with the improved central difference operator is used. The artefacts due to the gradient reconstruction, introduce incorrect information about the surface that is reconstructed.
The artefacts, that appear when the improved central difference operator is used, confirm the importance of gradient reconstruction. We also show that the B-splines are suitable for surface reconstruction as well as for gradient reconstruction.

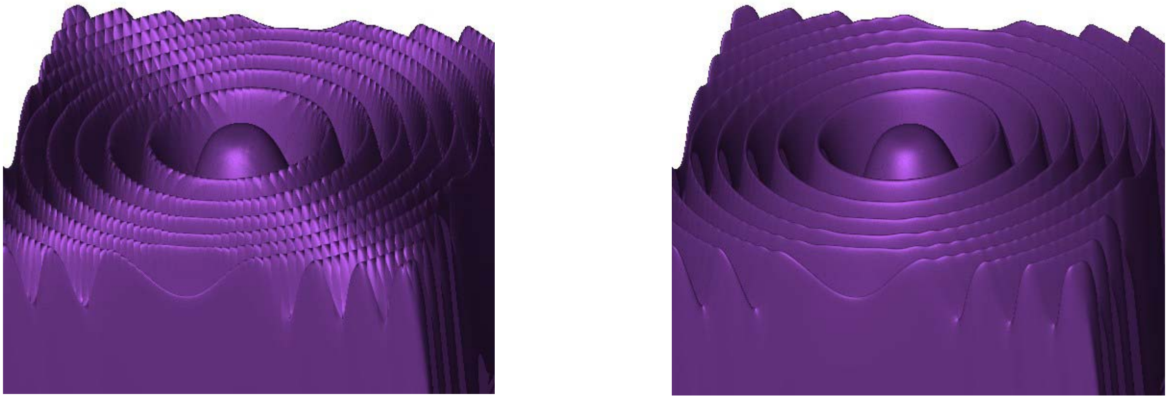

Fig. 5. Comparison between gradient estimation with the improved central difference operator and the gradient of the B-spline function.

References

[1] M. J. Bentum, B. A. Lichtenbelt, T. Malzbender, “Frequency Analysis of Gradient Estimators in Volume Rendering’”, IEEE Transactions on Visualization and Computer Graphics, Vol. 2, No. 3, September 1996, pp. 242-253.

[2] R. Machiraju and R. Yagel, “Reconstruction Error Characterization and Control: A Sampling Theory Approach”, IEEE Transactions on Visualization and Computer Graphics, Vol. 2, No. 4, December 1996, pp. 364-376.

[3] S. R. Marschner and R. J. Lobb, “An Evaluation of Reconstruction Filters for Volume Rendering”, Proc. Visualization '94, IEEE CS Press, October 1994, pp. 100-107.

[4] Ž. Mihajlović, A. Goluban, The B-spline Interpolation in Visualization, Journal of Computing and Information Technology, CIT, Vol. 7, No.3, September 1999, pp. 245-253.

[5] Ž. Mihajlović, L. Budin and Z. Kalafatić, “Volume Rendering with Least Squares Spline Reconstruction“, Proceedings of the 9 th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2002, Dubrovnik, Croatia, 2002, pp. 843-846.

[6] T. Moller, R. Machiraju, K. Mueller and R. Yagel, “Evaluation and Design of Filters Using a Taylor Series Expansion”, IEEE Transactions on Visualization and Computer Graphics, Vol. 3, No. 2, June 1997, pp. 184-199.

[7] M. Unser, A. Aldroubi and M. Eden, “B-spline Signal Processing: Part I - Theory”, IEEE Transactions on Signal Processing, Vol. 41, No. 2, February 1993, pp. 821-833.

[8] M. Unser, A. Aldroubi and M. Eden, “B-spline Signal Processing: Part II – Efficient Design and Applications”, IEEE Transactions on Signal Processing, Vol. 41, No. 2, February 1993, pp. 834-847.