doi:10.2498/cit.1000916

Influence of Blending Functions on Volumetric Cloud Images

Bogdan Lipuš and Nikola Guid

Faculty of Electrical Engineering and Computer Science, University of Maribor, Maribor, Slovenia

Various blending functions can be applied in the volumetric modelling of clouds. This paper analyzes the impact of five different blending functions (linear, cubic, Wyvill-1, Wyvill-2, and cosine) on cloud shape, structure, and area. We have found that there are only negligible differences among them. Therefore, we can conclude that the simple linear blending function satisfies all the desired properties for generating cloud images by volumetric modelling.

Keywords: computer graphics, volumetric modelling, cloud modelling, natural phenomena

1. Introduction

Volume modelling is a useful technique for modelling natural phenomena, such as clouds, which are semi or totally transparent and do not have precisely distinguished boundaries. One way to define cloud is to use implicit primitives, which are connected as a whole using blending (or field) functions. They altogether represent the cloud macrostructure [3]. In addition, the cloud micro-structure is madeup of procedural functions, such as the Perlin noise function and the fractal turbulence function [4,7]. An interior of the semitransparent volume must be determined in volume modelling. The blending functions provide a smooth density blending among the implicit primitives, which can obtain, in general, various structures.

This paper is based on our work [5], which presents a new approach for combining implicit blending functions. Our method gives better results than Ebert's summation [3,4]. As Ebert, we also only used the Wyvill-1 blending function [8].

In literature regarding implicit surface modelling, there are many different blending functions for satisfying continuity conditions among implicit objects. Firstly, we have chosen the Wyvill-2 blending function, and a simple cubic function [2]. Naturally, there are also other convenient blending functions [1]. Thus, we have tried to apply cosine blending functions, which also satisfy the aforementioned requirements. Finally, we have also tested the simple linear blending function, although it does not fulfill the required conditions regarding implicit surface modelling.

Thus, our research only considers five different blending functions. We analyze their influence on the cloud shape, structure, and area. We discover that the simple linear blending function also provides a smooth blending of cloud density. This phenomenon can be explained by the nature of the volume rendering, which accumulates density field values along a ray. In this case, we do not consider the continuity conditions on the surface level, but on the solid level.

This paper is organized as follows: Section 2 presents the various blending functions with their properties and combining techniques. In Section 3, we introduce various evaluation criteria and present the results of this evaluation. Finally, Section 4 concludes the paper.

2. Blending Functions

In volume rendering, blending functions are used for the continuous connection of density fields within the implicit primitives, as can be seen in Figure 1. Here, we blend a skeleton

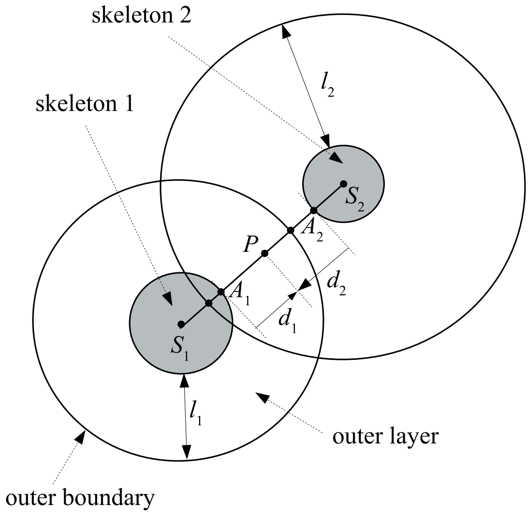

Figure 1. Two implicit primitives.

area, where the density is 1, and an outer boundary, where the density falls to 0. The blending function b_i of i-th primitive is a function of the distance d_i (i.e. the distance $\overline{A_iP}$ between the boundary of i-th implicit skeleton and the point P) and the width l_i of the outer layer (field). Without loss of generality, we can use the normalized distance x_i instead of the actual distance d_i $(x_i = d_i/l_i)$.

If $x_i = 0$, the point P is on the skeleton's boundary, if $x_i = 1$, it is on the outer boundary of the outer layer of i-th implicit skeleton. The following properties of the blending function $b_i(x_i)$ are desirable [3]:

1. Its initial value is $b_i(0) = 1$ and the final value is $b_i(1) = 0$.
2. It has zero first derivatives by $x_i = 0$ and $x_i = 1$ (i.e. $b'_i(0) = b'_i(1) = 0$).
3. It passes through the point $P(0.5, 0.5)$.
4. It is symmetrical around the point $P(0.5, 0.5)$. This symmetry requires the validity of the following equation: $b_i(x_i) = 1 - b_i(1 - x_i)$.

These properties can be fully or partially satisfied by most of the blending functions (see Table 1). We have studied five various functions: the linear, the cubic, the Wyvill-1 [8], the Wyvill-2 [2], and the cosine one. All functions are defined within the interval $[0, 1]$. There are also some other blending functions (see [1]).

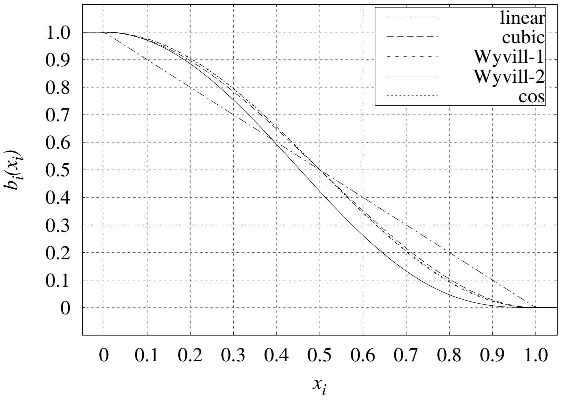

Figure 2. Various blending functions: linear, cubic, Wyvill-1, Wyvill-2, and cosine.

Figure 2 shows all five mentioned blending functions. The cubic, the Wyvill-1, and the cosine blending functions are minimally different. Moreover, the Wyvill-1 and the cosine blending functions vary by only 1%. The values of the Wyvill-2 [2] blending function decreases much more quickly than the aforementioned group of blending functions. The linear blending function varies noticeably from the others. Wyvill [8] proposed a blending function (called Wyvill-1) to avoid the calculation

Blending function	Equation	Properties			
		1	2	3	4
linear	$1 - x$	+	−	+	+
cubic	$2x^3 - 3x^2 + 1$	+	+	+	+
Wyvill-1	$-\frac{4}{9}x^6 + \frac{17}{9}x^4 - \frac{22}{9}x^2 + 1$	+	+	+	−
Wyvill-2	$(1 - x^2)^3 = 1 - 3x^2 + 3x^4 - x^6$	+	+	−	−
cos	$\frac{\cos(\pi x)}{2} + \frac{1}{2}$	+	+	+	+

Table 1. Blending functions.

x_i	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
linear	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.0
cubic	1.0	0.9720	0.8960	0.7840	0.6480	0.5	0.3520	0.2160	0.1040	0.0280	0.0
Wyvill-1	1.0	0.9757	0.9052	0.7950	0.6554	0.5	0.3441	0.2035	0.0927	0.0231	0.0
Wyvill-2	1.0	0.9703	0.8847	0.7536	0.5927	0.4219	0.2621	0.1327	0.0467	0.0069	0.0
cos	1.0	0.9755	0.9045	0.7939	0.6545	0.5	0.3455	0.2061	0.0955	0.0245	0.0

Table 2. Blending functions values.

of square root, which is used for the normalized Euclidean distance x between two points. The Wyvill-1 blending function goes through the point $P(0.5, 0.5)$, but is only approximately symmetrical around this point. The Wyvill-2 does not even pass through the mentioned point ($b(0.5) = 27/64 = 0.41875$), therefore it is also non-symmetrical. Blending function values are showed in Table 2.

2.1. Combining Blending Functions

It is wellknown that the macrostructure of a cloud is defined by some implicit primitives. We must combine individual density functions in order to calculate overall density function. The simplest technique, which was used in cloud modelling by [3,4], is a summation of n blending functions [8]. There are also other combining techniques presented by Wyvill [9] and Pasko [6]. These techniques are mainly used in implicit surface modelling.

In [5], it is reported that the summation technique can cause visual abnormalities in those areas where two or more implicit primitives intersect. These abnormalities have their origins in a higher value of density function than those within the implicit skeleton. Therefore, a new approach is presented (a corrected sum of the blending functions) to overcome these irregularities and to obtain better results than current techniques. It is realized by the following recursive formula:

$$\begin{aligned} \rho_1(\mathbf{p}) &= g_1(x_1) \\ \rho_i(\mathbf{p}) &= \rho_{i-1}(\mathbf{p}) + g_i(x_i) - \rho_{i-1}(\mathbf{p})g_i(x_i), \\ & \qquad i = 2, \ldots, n \end{aligned} \tag{1}$$

where $g_i(x_i) = w_i b_i(x_i)$ is a weighted blending function of i-th implicit primitive, w_i is the weight of i-th implicit primitive, b_i is the blending function of i-th implicit primitive (in general, b_i can be different, but usualy, they are the same for all i), and n is the number of implicit primitives. Figure 3 shows the combining of two Wyvill-1 blending functions using Equation 1.

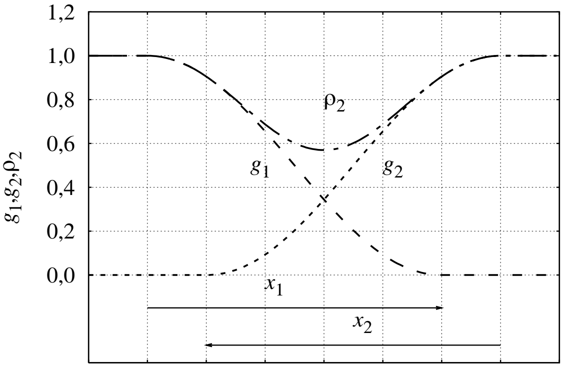

Figure 3. Combining of two Wyvill-1 blending functions.

3. Influence of the Various Blending Functions on the Cloud Structure, Shape and Size

In this section, various evaluation criteria are considered using the different blending functions. We use our new aforementioned technique for combining blending functions.

3.1. Pixel Intensity along a Line on a Cloud Image

Pixel intensity is examined along a line of the rendered image of two implicit primitives combining by two blending functions. The pixel intensity of the cloud can be studied along the line, as shown in Figure 4. We use 500 samples between two line endpoints and try to figure out whether we obtain any visible discontinuity on the inspection line using different blending functions. The number of samples is taken according to the chosen image resolution. As shown in Figure 4, no turbulence is used by generating a cloud image.

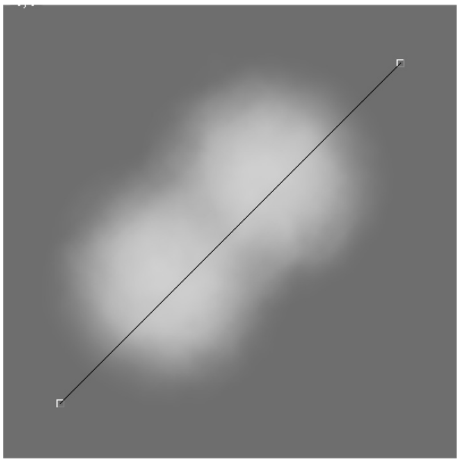

Figure 4. An inspection line on a cloud created by two implicit primitives (without turbulence).

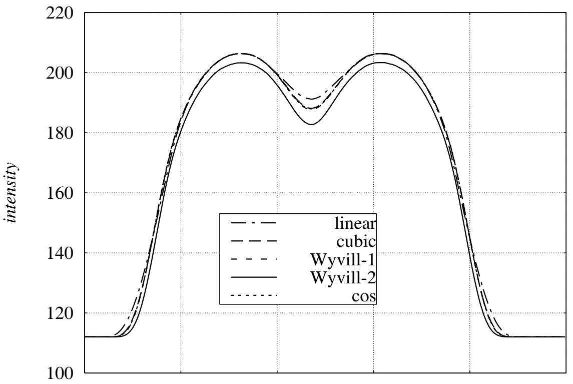

Figure 5. Pixel intensity curve along an inspection line for the different blending functions.

Figure 5 presents pixel intensity samples along the inspection line using different blending functions. We notice that all blending functions give very similar results. The intensity samples on the inspection line are influenced by two blending functions (see Figure 3). The values of the pixel intensity in the middle of the inspection line are the highest for the linear blending function, and the lowest for the Wyvill-2 blending function. In Figure 2, we see that the linear blending function is higher than the Wyvill-2 if x_i is in the interval $[\sim 0.4, 1.0]$. We cannot see any C^1 discontinuity following the values of the pixel intensity along the line. After our opinion, the main reason for this is the volume rendering, in which the density field values are accumulated along a ray. Therefore, the linear blending function also gives a continuous pixel intensity curve, although it possesses C^1 discontinuity at $x = 0$ and $x = 1$ (Figure 2).

Determination of intensity continuity can be done by calculating the differences between two adjacent intensity samples. Because the samples along the line on the image are obtained in the discrete manner, it is difficult to determine if some blending function causes C^1 discontinuity. Therefore, before the differences are calculated, the intensity samples are interpolated using the following recursive formula:

$$d_i^* = \frac{(d_i + \ldots + d_{i+k-1})}{k}, \; i = 1, \ldots, m-k+1 \qquad (2)$$

where d_i is i-th sample of the pixel intensity, k is the number of pixel samples to interpolate, and m is the number of samples. Figure 6 shows the differences obtained from interpolated intensity samples ($k = 15$) by the different blending functions.

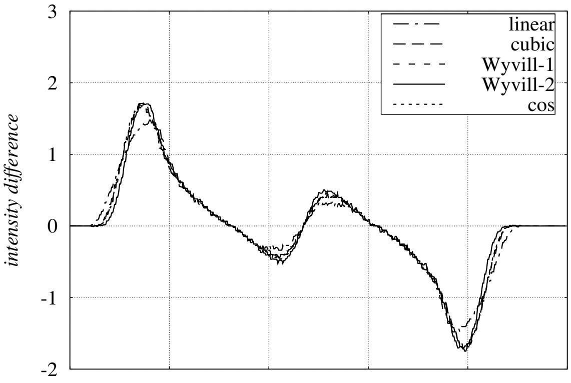

Figure 6. Pixel intensity differences along the inspection line for the different blending functions ($k = 15$).

All blending functions have similar behaviour, although we had firstly expected that the linear blending function would show C^1 discontinuity along the inspection line.

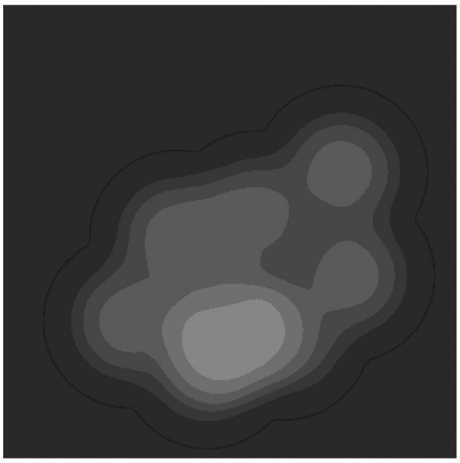

Figure 7a.

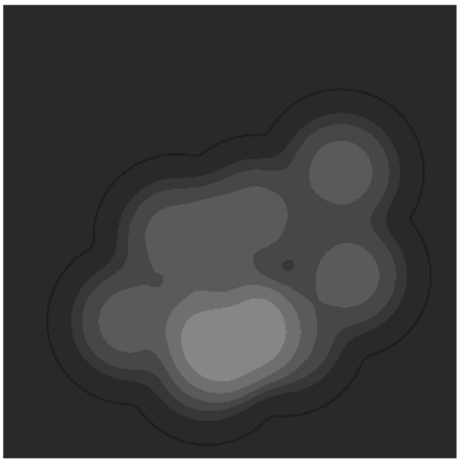

Figure 7b.

3.2. Histogram of Pixel Intensity Distribution

We also examined the distribution of the pixel intensities in a cloud's generated image using the different blending functions. The range of intensity values (i.e. 0 – 255) is divided into 16 bands (each band holds 16 different intensity values). Figure 8 presents the obtained histogram of the cloud image without turbulence. As can be seen, it is almost independent of the blending function. In our case, most of the intensity values are in the 7-th band (i.e. intensity values between 96 and 112).

Figure 7 shows the intensity distribution on the cloud image without turbulence and using nine implicit primitives, and the different blending functions. The range of intensity values is also divided into 16 bands, in the same manner as with the previous histogram. The highest inten-

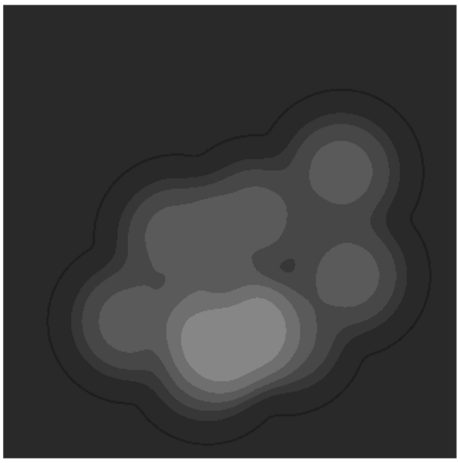

Figure 7c.

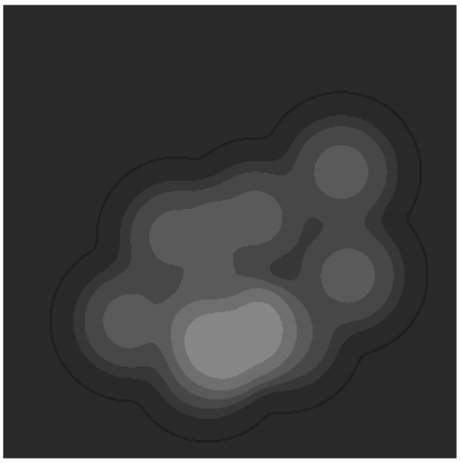

Figure 7d.

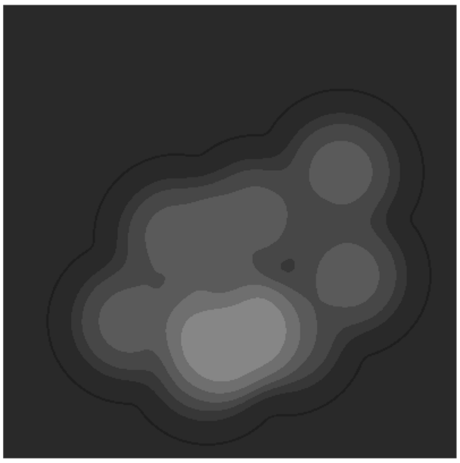

Figure 7e.

Figure 7 (a – e). Intensity distribution with 16 bands on the cloud image with nine implicit primitives using the different blending functions: a) linear, b) cubic, c) Wyvill-1, d) Wyvill-2, and e) cosine.

sity values are in those areas where the implicit primitives are defined. We noticed that these areas are connected more closely with the linear blending function than with the Wyvill-2. This is also noticeable in Figure 5 where, at certain intensity values, the distance between the two peaks is longer with the Wyvill-2 blending function than with the linear one. The intensity distributions of the various blending functions are practically the same, only the distance (dark area in the middle of the cloud) is much bigger with the Wyvill-2 blending function (see Figure 7d).

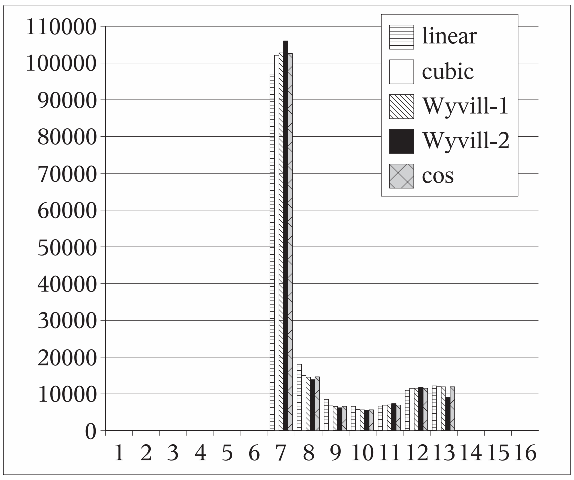

Figure 8. Histogram of pixel intensity distribution.

3.3. Cloud Area

In this section, we have researched how much the blending functions influence the cloud area. We have compared two cloud images generated with two different blending functions. Firstly, we detected the cloud area on both images. Because the cloud is brighter than the background sky, this procedure is trivial. Then, the following cloud similarity index is used:

$$r_i = \frac{N_{int}}{N_i} \tag{3}$$

where N_{int} is the number of pixels in the intersection area of both clouds and N_i is the number of pixels in i-th cloud area (see Figure 9). The indexes r_1 and r_2 indicate by how much the two cloud areas on the two images are similar to each other. The value of the index $r_i = 1$ indicates that the intersection area is all in i-th cloud area. If the indexes r_1 and r_2 are equal to 1, both cloud areas are equal. Table 3 presents cloud similarity indexes obtained by comparing cloud area 1 (generated by Wyvill-1 blending function) and cloud area 2 (generated by other blending functions). They prove the relationship among the blending functions, as shown in Figure 2.

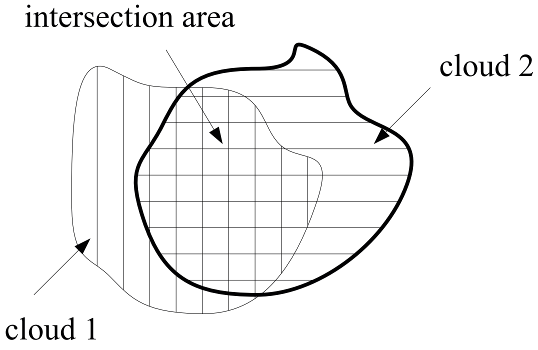

Figure 9. Comparison of two cloud areas from two different images.

r_1		r_2	
Wyvill-1	1.0	0.9088	**linear**
Wyvill-1	1.0	0.9887	**cubic**
Wyvill-1	0.9429	1.0	**Wyvill-2**
Wyvill-1	0.99996	0.9973	**cos**

Table 3. The indexes r_1 and r_2 obtained by comparing two cloud areas.

4. Conclusion

In this paper, we have evaluated the various blending functions using certain criteria to find out which blending function is the best. Our analysis has shown that there are minimal differences in the cloud model if the blending function is changed. We did not notice any C^1 discontinuity, which we originally expected for the linear blending function. The main reason for this is the volume rendering, in which the density field values are accumulated along a ray. Therefore, C^1 discontinuity is obscured. Continuity is, in general, also influenced by procedural functions with which we model the cloud micro-structure. In contrast to implicit surfaces, where the problem of continuity is important, it is negligible for volume modelling and rendering. Therefore, we can say that the properties of the blending functions written in Section 2 are too strong.

We have found that the linear blending function gives equally good results as, for example, the Wyvill-1. Furthermore, we can conclude that the different blending functions have hardly any noticeable effect on a cloud's appearance. The use of the various blending functions does not change the cloud image as much as the change in the procedural cloud model's parameters does.

In future work, we will study how a particular cloud model's parameter affects its appearance.

References

[1] C. Blanc, C. Schlick, Extended Field Functions for Soft Objects. *Implicit Surfaces '95*, Eurographics/ACM SIGGRAPH Workshop, 21–32.

[2] J. Bloomenthal, C. Bajaj, J. Blinn, M. P. Cani-Gascuel, A. Rockwood, B. Wyvill, G. Wyvill, *Introduction to Implicit Surfaces*, Morgan Kaufman Publishers, Inc., 1997.

[3] D. Ebert, Volumetric Modeling with Implicit Functions: A Cloud is Born. *SIGGRAPH 97 Visual Proceedings*, (1997), 147–156.

[4] D. S. Ebert, F. K. Musgrave, D. Peachey, K. Perlin, S. Worley, *Texturing and Modeling: A Procedural Approach*. Academic Press, second edition, 1998.

[5] B. Lipuš, N. Guid, A New Implicit Blending Technique for the Volumetric Modelling. *The Visual Computer*, **21**(1-2) (2005), 83–91.

[6] A. Pasko, V. Adzhiev, A. Sourin, V. Savchenko, Function Representation in Geometric Modeling: Concepts, Implementation and Applications. *The Visual Computer*, **11**(8) (1995), 429–446.

[7] K. Perlin, E. M. Hoffert, Hypertexture. *Computer Graphics*, **23**(3) (1989), 253–262.

[8] G. Wyvill, C. McPheeters, B. Wyvill, Data Structure for Soft Objects. *The Visual Computer*, **2**(4) (1986), 227–234.

[9] B. Wyvill, G. Wyvill, Field Functions for Implicit Surfaces. *The Visual Computer*, **5** (1989), 75–82.

Received: August, 2006
Revised: September, 2007
Accepted: October, 2007

Contact addresses:

Bogdan Lipuš
Laboratory for Computer Graphics
and Artificial Intelligence
Faculty of Electrical Engineering
and Computer Science
University of Maribor
Smetanova 17,
SI-2000 Maribor, Slovenia
e-mail: `bogdan.lipus@uni-mb.si`

Nikola Guid
Laboratory for Computer Graphics
and Artificial Intelligence
Faculty of Electrical Engineering
and Computer Science
University of Maribor
Smetanova 17,
SI-2000 Maribor, Slovenia

Bogdan Lipuš received his BSc, MSc, and PhD degrees in computer science from the Faculty of Electrical Engineering and Computer Sciences, University of Maribor, in 2000, 2003, and 2005, respectively. His research interests include computer graphics, computer-aided geometric design, and computer animation. In 2005, he joined Hermes Softlab, Ljubljana. Currently, he is a teaching assistant at the Faculty of Electrical Engineering and Computer Sciences, University of Maribor.

Nikola Guid is presently a full professor at the Faculty of Electrical Engineering and Computer Sciences, University of Maribor, Slovenia, and head of the Laboratory of Computer Graphics and Artificial Intelligence. His current research interests are computer graphics, computer aided geometric design, and geometric modeling. Guid received both a BSc degree in electronics and MSc degree in computer science from the University of Ljubljana, Slovenia and PhD in technical sciences from the University of Maribor. In 1991 and 1993, each time for a period of six months, he was doing at the Faculty of Engineering of the University of Udine, Italy. He is a member of the IEEE, the ACM, and Eurographics.