EUROGRAPHICS 2001 / A. Chalmers and T.-M. Rhyne
(Guest Editors)

Volume 20 (2001), Number 3

Fast Visualization of Object Contours by Non-Photorealistic Volume Rendering

Balázs Csébfalvi[†],
Vienna University of Technology, Austria, `http://www.cg.tuwien.ac.at/home/`,

Lukas Mroz, Helwig Hauser[‡],
VRVis Research Center, Vienna, Austria, `http:/www.vrvis.at/`,

Andreas König, and Eduard Gröller[§]
Vienna University of Technology, Austria, `http://www.cg.tuwien.ac.at/home/`

Abstract

In this paper we present a fast visualization technique for volumetric data, which is based on a recent non-photorealistic rendering technique. Our new approach enables alternative insights into 3D data sets (compared to traditional approaches such as direct volume rendering or iso-surface rendering). Object contours, which usually are characterized by locally high gradient values, are visualized regardless of their density values. Cumbersome tuning of transfer functions, as usually needed for setting up DVR views is avoided. Instead, a small number of parameters is available to adjust the non-photorealistic display. Based on the magnitude of local gradient information as well as on the angle between viewing direction and gradient vector, data values are mapped to visual properties (color, opacity), which then are combined to form the rendered image (MIP is proposed as the default compositing stragtegy here). Due to the fast implementation of this alternative rendering approach, it is possible to interactively investigate the 3D data, and quickly learn about internal structures. Several further extensions of our new approach, such as level lines are also presented in this paper.

Key words: interactive volume rendering, non-photorealistic rendering, shear-warp projection.

1. Introduction

In the last two decades various volume rendering methods were proposed for investigating internal structures contained in large and complex data sets. Depending on the data of interest and the characteristics of the structures to be visualized different rendering models can be applied.

Traditionally, so called photorealistic approaches dominate the field of volume visualization. The representation of objects within a 3D data set by means of iso-surfaces, for example, which themselves are approximated by a large collection of polygons each (cf. marching cubes[14]), as well as the use of transfer functions to map density values to visual properties, which in turn are composed to the final image by the use of alpha-blending along viewing rays (cf. direct volume rendering[3, 13]), are just two prominent examples.

One special challenge of most volume rendering approaches is the specification of parameters such that the resulting images are in fact providing useful insights into the objects of interest. With iso-surface rendering, for example, the specification of proper iso-values is crucial for the quality of the visualization. When direct volume rendering (DVR) is used, the specification of useful transfer functions has been understood as the most demanding and difficult part[11, 6, 7, 1]. Therefore, techniques, which do not require a lot of parameter tuning, while still conveying useful information about the data set of interest, proved to be useful, especially when applied during the exploration phase of data investigation, i.e., when the data should be understood in general.

[†] `csebfalvi@cg.tuwien.ac.at`
[‡] `mroz@vrvis.at, hauser@vrvis.at`
[§] `{koenig|groeller}@cg.tuwien.ac.at`

An interesting experience from working with applications in the field of volume rendering, which is especially important for the work presented in this paper, is that (during investigation) volumetric data often is interpreted as being composed of distinct objects, for example, organs within medical data sets. For a useful depiction of 3D objects, often boundary surfaces are used as a visual representation of the objects. This is mostly due to the need to avoid visual clutter as much as possible. Even in direct volume rendering, when transfer functions are used which also depend on gradient magnitudes[13], visible structures are significantly related to object boundaries. This is a direct consequence of the fact that voxels, which belong to object boundaries, typically exhibit relatively high values of gradient magnitude, i.e., values of first-order derivative information.

In contrast to photorealistic approaches, which stick to (more or less accurate) models of the illumination of translucent media[15], non-photorealistic techniques allow to depict user-specified features, like regions of significant surface curvature, for example, regardless of any physically-based rendering. Recently, also non-photorealistic rendering techniques, which originally have been proposed for computer graphics in general[20, 12, 5], have been proposed for volume rendering[4, 23], definitely extending the abilities for the investigation of 3D data. Ebert and Rheingans[4], for example, give a broad spectrum of various non-photorealistic visual cues (boundary enhancement, sketch lines, silhouettes, feature halos, etc.) to be easily integrated within the volume visualization pipeline. Also very interesting, other specialized techniques for the representation of object surfaces have been proposed recently like 3D LIC which is based on an eigenvalue/-vector analysis of surface curvature as proposed by Interrante[10], or the representation of surface points by the use of small geometric objects as proposed by Saito[21].

In this paper, we present a non-photorealistic rendering technique for volumetric data, which does not depend on data values, but on the magnitude of gradient information, instead. Thereby, 3D structures, which are characterized by regions exhibiting significant changes of data values, become visible without being obstructed by visual representations of rather continuous regions within the 3D data set. We discuss the importance of high-quality gradient reconstruction, which in our case is performed by a sophisticated method based on linear regression[16]. We also present two interactive techniques which are based on the reduction of the data to be processed. Due to our simplified NPR model only a small part of the volume contributes to the generated image. Therefore, interactive frame rates can be achieved without using any specialized hardware.

2. Contour Rendering

In this section we present the non-photorealistic rendering (NPR) model which we use to display the contours of the objects within the volumetric data set. The following characteristics of the rendering model in use are crucial:

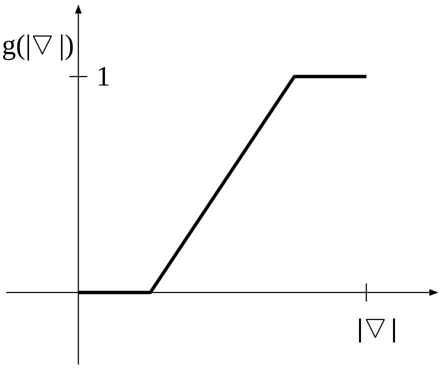


Figure 1: *The $g(|\nabla|)$ windowing function.*

- no a priori knowledge about the data values
- enhancement of internal details
- just a few rendering parameters, fast fine tuning
- support of optimization for fast previewing

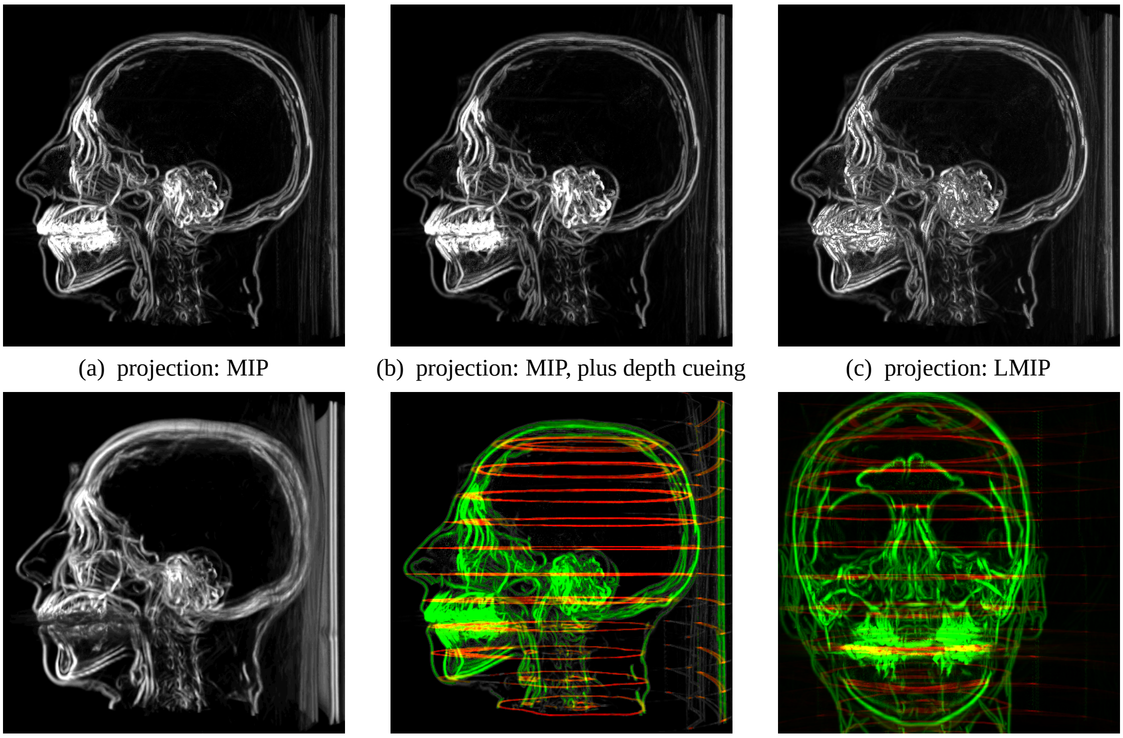
Since we need a model, which is independent of data values, as a general tool for volumes of various origins, the data values themselves should not directly contribute to the final result. Instead of the original data values, a function $g(|\nabla|)$ of gradient magnitudes is used (note, that for the sake of simplicity '∇' is used here as a shortcut for '∇data_i') in order to characterize the "surfaceness" of a voxel.

This function can be defined as a windowing function by default determining the range of interest in the domain of gradient magnitudes (Figure 1). Our model enhances the voxels with higher value of function g emphasizing the surfaces of different objects.

The traditional way of iso-surface enhancement is to use a transfer function, where the opacities are weighted by the gradient magnitudes[13]. This solution has the following drawbacks:

- time consuming transfer function specification
- limited number of iso-surfaces rendered at the same time
- internal details like cavities can be easily missed
- ray-casting-based rendering is computationally expensive

Considering these disadvantages we do not follow the idea of rendering several iso-surfaces. Instead of this we propose a non-photorealistic visualization model. Our goal is to give a first 3D impression about the content of the volume preferably without missing important features. Therefore, only the silhouette lines of iso-surfaces are rendered in order to avoid the hiding of important details. Previously this idea has been used only for single iso-surfaces defined by a density threshold and reconstructed by the "marching cubes" algorithm. The silhouette enhancement can be adapted to volumes, where the contours of all the objects can be rendered at



(a) projection: MIP
(b) projection: MIP, plus depth cueing
(c) projection: LMIP
(d) opacity modulation
(e) additional level lines
(f) front view

Figure 2: *Different options of using non-photorealistic rendering for the visualization of object contours within volumes.*

the same time providing images which are rich in details. In our model we use a view-dependent function $s(P,V)$ which assigns higher weights to voxels belonging to an object contour:

$$s(P,V) = (1 - |\nabla(P) \cdot V|)^n, \qquad (1)$$

where P is the position of a given voxel and vector V is the viewing direction and n is an exponent controlling the sharpness of the contour.

There are several opportunities of using weighting functions g and s in calculating the pixel colors corresponding to the viewing rays. For example, an intensity value for a sample position P can be calculated using the following formula:

$$\begin{aligned} I(P,V) &= g(|\nabla(P)|) \cdot s(P,V) \\ &= g(|\nabla(P)|) \cdot (1 - |\nabla(P) \cdot V|)^n \qquad (2) \end{aligned}$$

After having these view-dependent intensities calculated at the sample points along a viewing ray we can use maximum intensity projection (MIP), thus the highest sample intensity is assigned to the given ray. Note that, the classification according to function $I(P,V)$ results in a sparse volume significantly reducing the number of voxels which contribute to the final image. This approach has two advantages. There is no visual overload (unlike in direct volume rendering, when several iso-surfaces are rendered at the same time) and this property can also be exploited in the optimization of the rendering process as we will show it in the further discussion. Figure 2a shows the CT scan of a human head rendered using the contour enhancement function $I(P,V)$, where one can see the contours of the skull, the skin, and the brain at the same time.

If simple maximum intensity projection is used, depth information is not apparent in the image. The spatial impression can be improved by using simple depth cueing where the intensities of sample points are weighted according to their distance from the viewpoint. Figure 2b shows an image generated with this depth-cueing extension. Here for instance, the contours of the ears are better recognizable than in Figure 2a.

Even if depth cueing is used it can happen that some higher intensity contours hide the weaker contours which are closer to the viewer. This might be confusing in correct interpretation of the image. In medical imaging practice usually local maximum intensity projection (LMIP) is applied instead of traditional MIP in order to avoid such problems[22]. The idea is to project the first local maximum intensity onto the image plane, which is higher than a predefined threshold (Figure 3).

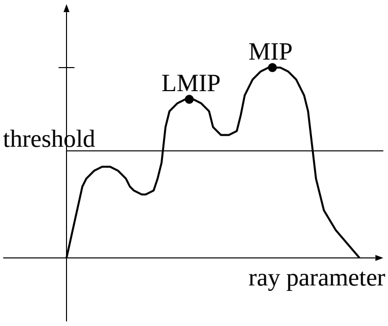


Figure 3: *Local maximum intensity projection (LMIP).*

This technique can be adapted also to the view-dependent intensities defined in our method. Figure 2c shows an image rendered using the LMIP of our contour enhancing intensity function $I(P,V)$.

Another alternative of keeping the depth information is to assign the values of function g as colors and opacities modulated by the view-dependent intensity function $I(P,V)$ (Equation 2) to the sample points and to perform an alpha-blending rendering.

This approach demonstrated in Figure 2d emphasizes rather the contours of low-curvature surfaces since for each high intensity pixel there are several high opacity samples along the corresponding viewing ray resulting in high accumulated opacity. Therefore, the fine details are less visible. This version is appropriate for data sets which contain many high frequency regions. In this case only the main characteristic contours are enhanced.

The basic contour projection approach can be extended by also rendering some additional characteristic lines like level lines. For instance, every twentieth slice is selected in the volume and independently from the viewing direction the values of function g are assigned as intensities to the voxels of the selected slices. These assigned intensities are rendered by maximum intensity projection as well but using a different color channel in order to distinguish the level lines from the object contours (Figure 2e). Of course these level lines can be perpendicular to any arbitrary direction as well.

Figure 2e clearly illustrate that the additional characteristic lines improve the spatial impression without hiding the important features. After having the image generated the user can interactively and separately scale the intensity of different color channels. It is also possible to simply switch off the display of additional characteristic lines.

Figure 4 (see color plates) shows internal organs of a human body rendered by our combined technique. Without taking care about the different density ranges the contours of the skin, the lungs, the spine, and the pelvis are visualized at the same time. The branches of the bronchia inside the lungs are also visible. These are typical low density cavities which can be hardly rendered using the conventional transfer function based ray casting.

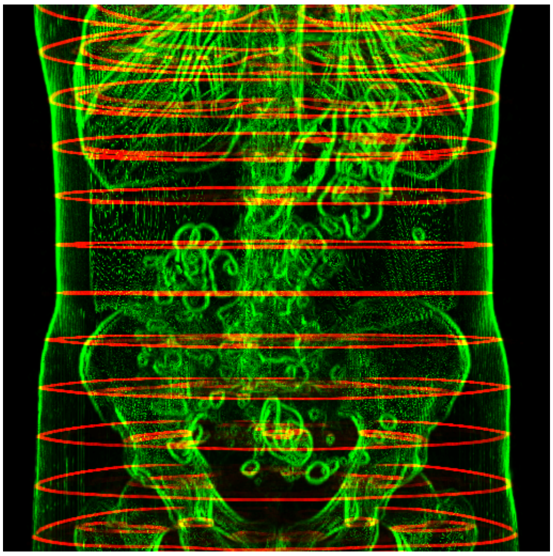


Figure 4: *Non-photorealistic visualization of internal organs.*

Figure 2f shows also the rendering of such internal cavities, where generating the front view of the human head the frontal sinuses are clearly visible.

For rendering the CT scan of human vertebrae containing screws, which is depicted in Figure 5 (see color plates) it would be confusing to use any additional characteristic lines. Note that, the visualization of the screws would require quite some time to find an appropriate transfer function if traditional direct volume rendering was applied.

3. Gradient Estimation

Usually in volume rendering the conventional method of central differences is used for gradient estimation. Since it takes only a narrow voxel neighborhood into account it causes typical staircase artifacts. Contour enhancing as it is described in this paper is very sensitive to accuracy in gradient directions, therefore a more sophisticated gradient estimation is required.

We use the results of our previous work [2]. The gradient estimation method presented there is based on linear regression, where in a local neighborhood the density function is approximated by a 3D hyperplane. The error of the approximation is defined as a mean square penalty function which is evaluated at each neighboring voxel position using a ring off weighting of error contributions. The minimization of the equation system leads to a computationally efficient convolution. Since the surface inclination is estimated from

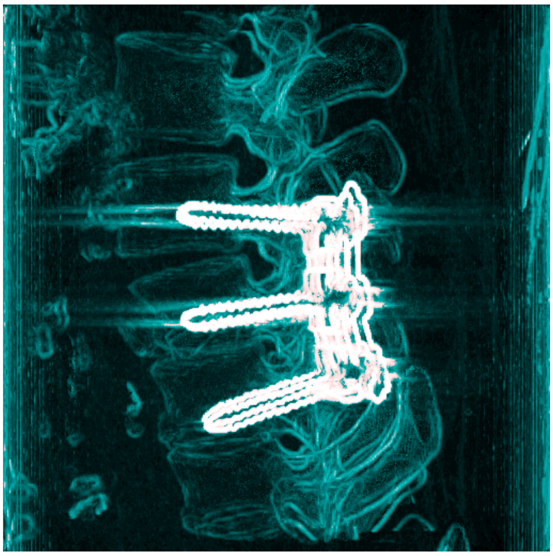


Figure 5: *Non-photorealistic rendering of human vertebrae with screws inside.*

a larger voxel neighborhood we obtain a smooth gradient function.

4. Rendering

Two different implementations of the non-photorealistic method have been used for generating the images shown in this paper. A high-quality, perspective ray caster (Figure 2, 4, and 5) and an interactive shear-warp based implementation (Figure 8) which is discussed below in detail.

When the lighting model from Equation 2 is used, the usual spatial cues which are offered by conventional rendering with a physics-based lighting model are not available. If in addition MIP is used to obtain pixel colors, depth information is also lost due to the inherent lack of occlusion within MIP images. The ability to interactively modify the viewing parameters and to change the view-point becomes crucial for understanding the result of the visualization process.

A precondition to achieve interactive rendering performance for volume visualization without special volume rendering hardware like a VolumePro board[24] is the ability to rapidly exclude from rendering parts of the volume which do not contribute to the resulting image. Classical volume rendering is based on compositing the contributions of samples along rays according to their opacities. In this case either "empty", entirely transparent regions, or parts of the volume located within high-opacity objects which are occluded by outer parts of the object can be omitted. Skipping of empty regions is usually performed using either a hierarchical approach (for example based on octrees[26]) or by encoding the distance to the closest non-transparent structure at each sample within the volume[25]. The skipping of non-contributing data within high-opacity regions can be done using front-to-back rendering and early ray termination.

If Equation 2 is used for determining the color of a voxel, its potential influence on the resulting image depends on two factors: First, the windowed magnitude of the gradient at the voxel, and second on the angle between the gradient and the current viewing direction. Similarly to the definition of classical opacity transfer functions, the influence of gradient magnitude on a voxel changes only if the windowing function is modified. Voxels with $g(|\nabla(P)|) = 0$ can easily be skipped using one of the established techniques for empty space encoding. Most of the remaining voxels with a higher gradient magnitude also do not contribute any significant information to a contour image, due to the influence of the $(1 - |\nabla(P) \cdot V|)^n$ term. Unfortunately, this term is view dependent, making any of the conventional volume region skipping techniques infeasible.

In the following sections we will discuss two approaches for skipping non-contributing voxels within the volume. Their applicability is closely coupled to the used rendering technique. The method used to achieve interactive rendering is therefore outlined first.

4.1. Preprocessing

Effective rendering of volumetric data sets where voxel contributions depend on gradient magnitude, requires methods which are able to efficiently deal with "sparse" volumes. For example, potentially contributing voxels can be extracted from the volume and stored in a list. For each of those voxels, its position and relevant attributes, like data value or gradient direction and magnitude are stored. The voxels within the list are ordered to meet requirements of a specific rendering technique. For MIP for example, sorting and projecting the voxels by data values eliminates the need for maximum search and allows efficient skipping of voxels mapped to black[18]. For DVR, sorting the potentially contributing voxels by depth eliminates the need to process empty voxels while still maintaining a correct order of compositing[19]. Voxels from common sized (256^3) data sets stored in this way can be rendered at interactive frame rates using for example shear-warp based projection[8, 16, 17]. To achieve interactive frame rates, this method performs parallel projection only, using nearest neighbor interpolation for projecting voxels onto the base plane.

A quantization of gradient vectors to 12-16 bits allows to perform shading by a single table lookup using the gradient vector as an index. The effort for computing the lookup table is negligible. For scanned data sets even the quantization to only 12 bits per gradient still provides visually satisfactory results. Despite the reduction in accuracy by quantization, performing gradient reconstruction using a high-quality method is still recommended, to avoid staircase artifacts as introduced by using the method of central differences.

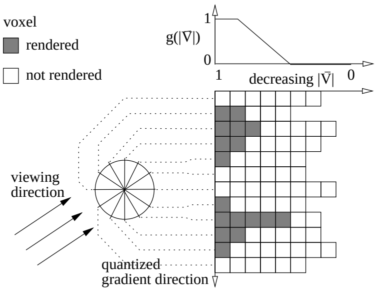


Figure 6: *Voxel ordering for MIP (2D). Voxels are grouped by (quantized) gradient direction. Within the groups, voxels are sorted by gradient magnitude. Only groups with $s(P,V) > \varepsilon$ are rendered, within a group rendering is stopped after the first voxel with $I(P,V) < \varepsilon$.*

4.2. Optimizations for MIP

In our case, voxel intensities are not constant data values from the volume as commonly used for MIP projection. Intensities $I(P,V)$ result from a function which depends on the viewing direction and gradient magnitude. This property makes a global pre-sorting of voxels by intensity impossible. However, proper ordering of the voxels can be used to group and efficiently skip groups of voxels mapped to black either by windowing of the gradient magnitude or by the influence of the current viewing direction.

For maximum intensity projection, the order of projecting voxels is not relevant as $max(a,b) = max(b,a)$. Thus, voxels do not have to be ordered and projected in spatial order. If we instead group voxels with the same or a similar gradient direction, we can exploit the fact, that voxels which are not part of a contour for the current viewing direction, are mapped to low intensity values. Entire groups of voxels with a similar gradient direction can be skipped, if the intensity $s(P,V)$ of a representative of this group is below some ε (see Figure 6). The quantization of gradient vectors for rendering leads to the required clustering of voxels into groups with the same gradient representation. For typical data sets, over 75% of all voxels can be skipped by exploiting just this scheme. Furthermore, within a group of voxels with the same gradient representation, voxels can be sorted by gradient magnitude. If projection of voxels within a group starts with voxels with the highest gradient magnitude, processing of the group can be stopped as soon as the first voxel with an intensity $I(P,V)$ below ε has been projected.

This arrangement of voxels allows to skip non-contributing parts of the data with high efficiency. The disadvantage of this optimization is the restriction of the compositing process to maximum intensity selection. Due to the arbitrary spatial ordering of the voxels, blending of voxel contributions is not feasible.

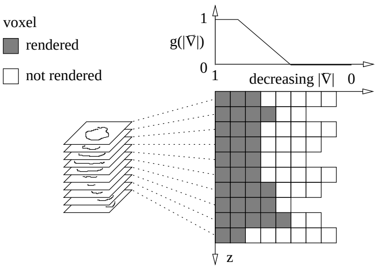


Figure 7: *Voxel ordering for back-to-front rendering: Voxels within each slice are sorted by gradient magnitude. Voxels which are mapped to 0 by $g(|\nabla|)$ can be skipped efficiently.*

4.3. Optimizations for Back-to-front Compositing (Blending)

To maintain full flexibility in the choice of compositing operations, like local MIP or alpha-blending, a spatially consistent ordering of the projected voxels has to be maintained. If shear-warp based projection is used, only the order in which slices of voxels are projected is relevant (consistently back-to-front or front-to-back). Within a slice of voxels, the ordering is not relevant as long as projections of neighboring voxels do not influence each other. Nearest neighbor interpolation for the projection to the base-plane meets this requirement. The grouping into slices has to be performed for each principal viewing axis -*x*, *y*, and *z* separately. Depending on the viewing direction, one of the three copies is used for rendering (the usual approach to shear-warp rendering).

Voxels with a gradient magnitude below a specific threshold do not provide a useful contribution to an image rendered using our model. Only about 25% of all voxels have a sufficiently high gradient magnitude, and are thus included into the extracted data structure, thus keeping the memory requirements at a reasonable level.

Within a slice, voxels are sorted according to gradient magnitude. During rendering, only voxels which are not mapped to black due to their gradient magnitude (see Figure 7) have to be considered. Voxels mapped to black due to the currently used $g(|\nabla|)$ are located at the end of the voxel list of the slice and can be efficiently skipped. Compared to the MIP-only ordering of voxels described in the previous section, significantly more voxels have to be rendered. Voxel skipping is only based on the “surfaceness” (i.e. gradient magnitude) of a voxel, but not on the view-dependent property of being part of a contour.

4.4. Interactive Rendering – Discussion

As usual for interactive rendering, a high frame rate is achieved by trading off rendering quality and accuracy against speed. The most important factor for the quality of contour images is the accuracy of gradient vectors (Section

	volume size	*voxels rendered*	*time*
head/mip	$256^2 \times 225$	102k	85ms
head/blend		366k	150ms
screws/mip	$256^2 \times 241$	337k	130ms
screws/blend		942k	270ms

Table 1: *Rendering Times*

3). Because of the gradient quantization, exponent n in Equation 1 can take values from a limited range. High values of n result in very sharp and thin contours. The quantization error of gradients close to a contour is therefore amplified and results in too bright or too dark voxel contributions. For a quantization to 12 bit as used by our implementation, an exponent around 4 provides a sufficiently narrow contour without producing disturbing artifacts (Figure 8).

Due to more efficient skipping of black voxels and a simpler compositing operation for projecting a voxel, rendering using MIP is faster (see Table 1) than when blending of voxel contributions is used. Although MIP allows to depict the most significant features of a volume (see Figure 8e, f), the lack of occlusion and depth information in MIP images may be a disadvantage. The high interactivity of non-photorealistic rendering using MIP compensates for this disadvantage by adding time as an additional degree of freedom for the visualization (i.e. interactive view-point changes).

More flexibility is gained by using opacity-modulated blending of voxel contributions for rendering. The rendering times are acceptable (Table 1), although slower than for MIP. Depending on the source of voxel opacity, different effects can be achieved. By setting the opacity equal to voxel intensity $I(P,V)$ (Figure 8a) an effect similar to MIP is achieved, with the difference, that occlusion and spatial ordering of the voxels is taken into account. Contours in areas with a higher gradient magnitude are depicted brighter than in areas with lower gradient magnitude. If opacity is derived from $g(|\nabla|)$ only, the resulting image displays a blended set of surfaces with lighted contours (Figure 8b)[17]. This approach can be also well used to enhance contours[4] in addition to Phong shading for surface rendering (Figure 8c). For segmented data which allows to distinguish between different objects, non-photorealistic methods can be easily combined with other rendering methods, for example with conventional surface rendering (Figure 8d).

The rendering times provided in Table 1 have been measured using a Java implementation of the algorithms on a PII/400MHz processor with Sun JDK 1.3 for Windows.

5. Conclusion

In this paper an interactive non-photorealistic volume rendering technique has been presented. It is proposed for fast viewing of different types of data sets without assuming any a priori knowledge about the density values. Our simplified visualization model does not require a time-consuming transfer function specification which is necessary in traditional direct volume rendering. It has been shown that using a simple contour enhancement approach different important details can be visualized simultaneously. This is hardly possible with rendering of several semi-transparent iso-surfaces.

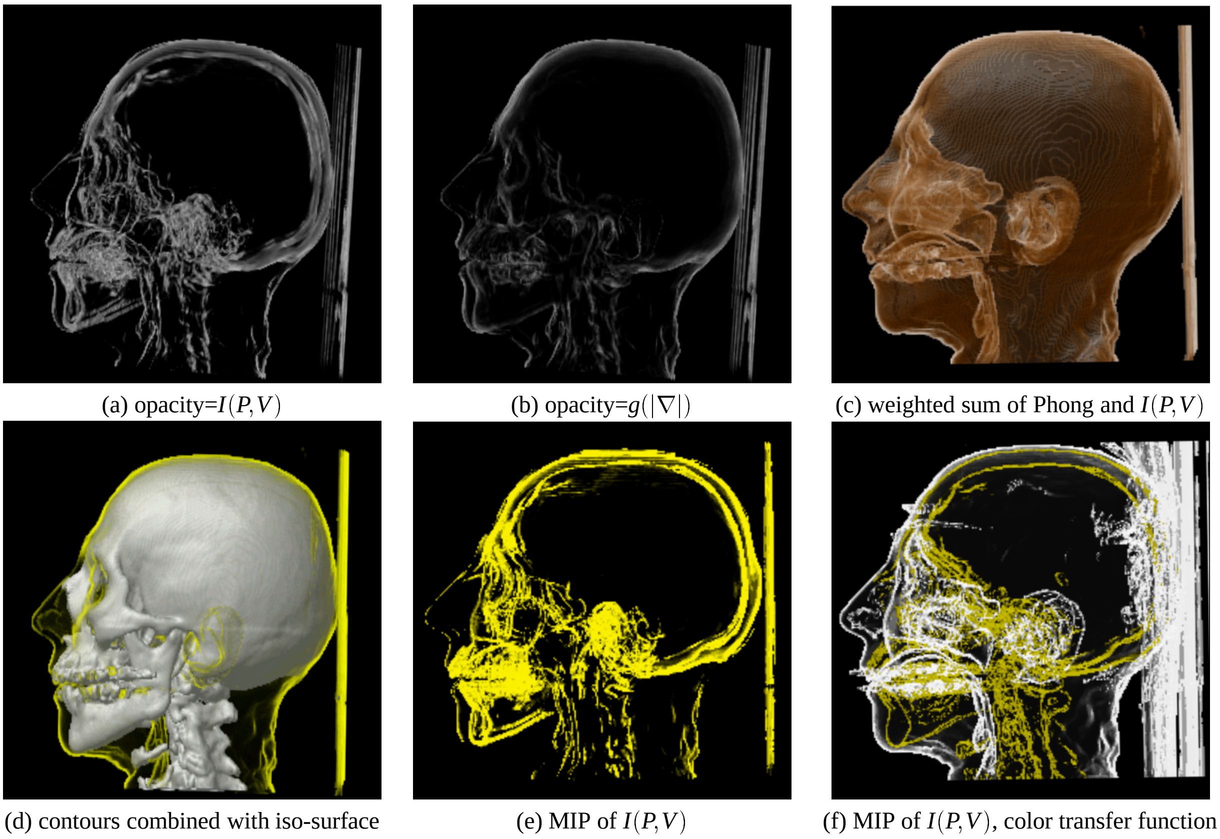
Our view-dependent voxel classification results in a sparse volume, therefore the number of voxels which contribute to the final image is significantly reduced. On one hand using this approach the visual overload can be avoided without missing the internal features, like low density cavities. On the other hand the data reduction can be exploited in the optimization of the rendering process. Since real-time rotation is very important in order to improve the spatial impression an interactive rendering technique has been presented as well. The main characteristic of the interactive approach is a reordering of voxels during a preprocessing step in a way, that voxels mapped to black during rendering due to their gradient magnitudes or due to the direction of their gradient vectors can be skipped with high efficiency. This is a pure software-based acceleration method which provides high frame rates even on low-end machines. It also supports the combination of traditional direct volume visualization and non-photorealistic volume rendering.

Acknowledgements

The work presented in this publication has been partly funded by the V^{is}M^{ed} project (`http://www.vismed.at`). V^{is}M^{ed} is supported by *Tiani Medgraph*, Vienna (`http://www.tiani.com`), by the *Forschungsförderungsfond für die gewerbliche Wirtschaft*, and by the Kplus program of the Austrian federal government. This work is also a part of the *BandViz* project (`http://bandviz.cg.tuwien.ac.at`), which is supported by FWF under project number P12811. The CT scan of the human head was obtained from the Chapel Hill Volume Rendering Test Dataset. The data was taken on the General Electric CT scanner and provided courtesy of North Carolina Memorial Hospital. The CT scan of human vertebrae was provided by *Tiani Medgraph*.

References

1. A. König and E. Gröller. Mastering Transfer Function Specification by using VolumePro Technology. *Tecnical Report TR-186-2-00-07, Institute of Computer Graphics and Algorithms, Vienna University of Technology*, 2000.



(a) opacity=$I(P,V)$

(b) opacity=$g(|\nabla|)$

(c) weighted sum of Phong and $I(P,V)$

(d) contours combined with iso-surface

(e) MIP of $I(P,V)$

(f) MIP of $I(P,V)$, color transfer function

Figure 8: *Interactive non-photorealistic rendering results.*

2. L. Neumann, B. Csébfalvi, A. König, E. Gröller. Gradient Estimation in Volume Data using 4D Linear Regression. *Computer Graphics Forum (Proceedings EUROGRAPHICS 2000)*, pages 351–358, 2000.

3. R.A. Drebin, L. Carpenter and P. Hanrahan. Volume rendering. *Computer Graphics (SIGGRAPH '88 Proceedings)*, 22, pages 65–74, 1988.

4. D. Ebert and P. Rheingans. Volume Illustration: Non-Photorealistic Rendering of Volume Data. *Proceedings of IEEE Visualization 2000*, pages 195–202, 2000.

5. G. Elber. Interactive Line Art Rendering of Freeform Surfaces. *Computer Graphics Forum (Proceedings EUROGRAPHICS '99)*, pages 1–12, 1999.

6. S. Fang, T. Bifflecome and M Tuceryan. Image-Based Transfer Function Design for Data Exploration in Volume Visualization. *IEEE Visualization '98*, pages 319–326, 1998.

7. I. Fujishiro, T. Azuma and Y. Takeshima. Automating Transfer Function Design for Comprehensible Volume Rendering Based on 3D Field Topology Analysis. *IEEE Visualization '99*, pages 467–470, 1999.

8. H. Hauser, L. Mroz, G.-I. Bischi and M. E. Gröller. Two-Level Volume Rendering - Fusing MIP and DVR. *IEEE Visualization 2000*, pp. 211–218, 2000.

9. K. H. Höhne, A. Pommert, M. Reimer, T. Schiemann, R. Schubert, U. Tiede, and W. Lierse. Anatomical Atlases Based on Volume Visualization. *IEEE Visualization 1992*, pp. 115–123, 1992.

10. V. L. Interrrante. Illustrating Surface Shape in Volume Data via Principal Direction-Driven 3D Line Integral Convolution. *Computer Graphics (SIGGRAPH '97 Proceedings)*, pages 109–116, 1997.

11. G. Kindlmann and J. W. Durkin. Semi Automatic Generation of Transfer Functions for Direct Volume Rendering. *IEEE Symposium on Volume Visualization '98*, pages 79–86, 1998.

12. J. Lansdown and S. Schofield. Expressive Rendering: A Review of Non-Photorealistic Techniques. *IEEE Computer Graphics and Applications*, Vol. 15, No. 3, pages 29–37, 1995.

13. M. Levoy. Display of surfaces from CT data.

IEEE Computer Graphics and Application, 8(5):29–37, 1988.

14. W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21, 4, pages 163–169, 1987.

15. N. Max. Optical Models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 1, No. 2, pp. 99–108, June 1995.

16. B. Csébfalvi, A. König and E. Gröller. Fast Surface Rendering of Volumetric Data. *Proceedings of Winter School of Computer Graphics*, 2000.

17. B. Csébfalvi and E. Gröller. Interactive Volume Rendering based on a "Bubble Model". *Proceedings of Graphics Interface*, 2001.

18. L. Mroz, A. König and E. Gröller. Real-Time Maximum Intensity Projection. *Data Visualization '99, Proceedings of the Joint EUROGRAPHICS - IEEE TCCG Symposium on Visualization*, pages 135–144, 1999.

19. L. Mroz, R. Wegenkittel and M. E. Gröller. Mastering interactive surface rendering for Java-based diagnostic applications. *IEEE Visualization 2000*, pp. 437–440, 2000.

20. T. Saito and T. Takahashi. Comprehensible Rendering of 3D Shapes. *Computer Graphics (SIGGRAPH '90 Proceedings)*, pages 197–206, 1990.

21. T. Saito. Real-time previewing for volume visualization. *Symposium on Volume Visualization*, pages 99–106, 1994.

22. Y. Sato, N. Shiraga, S. Nakajima, S. Tamura, and R. Kikinis. LMIP: Local Maximum Intensity Projection. *Journal of Computer Assisted Tomography*, Vol. 22, No. 6, 1998.

23. S. M. F. Treavett and M. Chen. Pen-and-Ink Rendering in Volume Visualization. *IEEE Visualization 2000*, pp. 203–209, 2000.

24. H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer and L. Seiler. The Volume Pro Real-Time Ray-casting System. *Proceedings of ACM SIGGRAPH'99*, pp. 251–260, 1999.

25. D. Cohen and Z. Sheffer. Proximity Clouds an Acceleration Technique for 3D Grid Traversal. *The Visual Computer* **10**(11), pp. 27–38, 1994.

26. P. Lacroute and M. Levoy. Fast Volume Rendering using a Shear-warp Factorization of the Viewing Transformation. *Proceedings of ACM SIGGRAPH '94*, pp. 451–459, 1994.