Hardware-Accelerated Illustrative Medical Surface Visualization with Extended Shading Maps

Christian Tietjen, Roland Pfisterer, Alexandra Baer, Rocco Gasteiger, and Bernhard Preim

Otto-von-Guericke University, Magdeburg, Germany
tietjen@isg.cs.uni-magdeburg.de

Abstract. In this paper, we introduce a new framework for the illustrative visualization of medical surface data. In most visualization frameworks, only light intensity is used to determine the surface shading. The analysis of medical textbooks reveals more complex shading approaches. The parameters of these approaches are mapped to different *Shading Maps*, which may be weighted and flexibly combined. We discuss the use of high-level attributes to simplify the specification. The resulting *Shading Map* is used as a lookup to determine the final intensity at a certain area. For this purpose, the rendering is accomplished on GPU by using OpenGL's Framebuffer Objects. This framework may be useful for interactive educational systems or for medical record printings.

1 Introduction

For illustrative visualizations, it is essential to convey the rich information on the object's shape and their relations to each other. Illustrators use shading to support shape perception, to guide the observer's attention and to achieve a balanced overall impression. We analyzed anatomical drawings to identify composition parameters and to adopt these to 3D surfaces. Parameters may describe local attributes of the objects surface or global attributes, like the relationship of different objects to each other.

The paper at hand discloses additional or alternative parameters for shading, regardless of the viewing direction, with the best idea of the scene. *Shading* in the further course describes the general use of brightness for displaying objects, as opposed to pure illumination shading. Every parameter is mapped onto one *Shading Map*, which may be weighted and combined. The resulting Shading Map is used as a lookup for the final intensity at a certain area. Therefore, a scene consisting of several medical surface objects may be rendered with stippling or hatching techniques, for example. It is also possible to control other properties like transparency, hue or saturation.

Since more freedom in parameterization leads to a greater demand of user interaction, we propose a simplification of the parameterization effort. Our framework is based on a two-level approach: at the higher level, the user may adjust four high-level attributes. At the elementary level, 20 parameter adjustments and weightings are available for fine-grained control.

This framework may be useful for interactive educational systems or for medical record printings. In most clinical offices still black-and-white printers dominate. Thus, an improved visualization for printings is helpful. Also, educational systems require motivational aspects in order to introduce the system or to explore the anatomical datasets. To render the scene in real-time, all Shading Maps are computed and combined on the GPU by using Framebuffer Objects.

2 Related Work

Bruckner et al. [1] present the concept of style transfer functions for illustrative volume rendering. Their image-based lighting model uses sphere maps to represent illustrative rendering styles. Tietjen et al. [2] introduce a combination of volume-, surface- and object-based line rendering. Svakhine et al. [3] build up a system to semi-automatically create visualizations of volume data in a variety of illustration motifs. The motif specification contains a compact set of core parameters. Ritter et al. [4] developed a pure black-and-white rendering of vascular structures to display shadow-like depth indicators and distance information for intra-operative use.

A couple of concepts exist for illustrative shading techniques. Akers et al. [5] developed a system to generate comprehensible images based on several photos. For non-photorealistic rendering of 3D objects, Hamel [6] suggests a lighting model consisting of different weighted illustrative shading techniques and simulates raking light by darkening surface areas with high curvature. Lee and Hao [7] optimize the overall lighting of a scene by using several local light sources. Shacked and Lischinski [8] propose a method to automatically determine the values of various lighting parameters by optimizing a perceptual quality metric. A similar approach is presented by Gumhold [9]. Both concepts are not designed for interactive exploration.

Yuan et al. [10] perform all non-photorealistic rendering operations in a geometry-image domain. In our approach, we need to compute the optimal shading per view, so our work is more related to Saito and Takahashi [11]. They make use of rendering buffers, so called G-buffers, to generate illustrative images including simple hatching techniques. Luft et al. [12] utilize the difference between the original depth buffer content and a low-pass filtered copy to enhance the perceptual quality of images. Rusinkiewicz et al. [13] investigate a non-photorealistic shading model based on adjusting the effective light position for different areas of the surface. Scheuermann and Hensley [14] present an efficient algorithm to compute image histograms in real-time entirely on the GPU. For the emulation of atmospheric depth, Ebert and Rheingans [15] suggest to interpolate non-linear between the original color and the background color value according to the depth value.

Illustrative, texture-based shading techniques are presented by [16, 17]. Baer et al. propose a novel approach for frame-coherent and seamless scalable stippling.

Praun et al. introduce a real-time system for rendering of hatching strokes over arbitrary surfaces.

3 Shading Arrangement in Medical Textbooks

Numerous publications deal with the creation of meaningful illustrations by using graphical elements, like stipples or strokes. Less attention was payed to the shading arrangement of such black-and-white illustrations. Therefore, this section analyzes the arrangement of shading in medical textbooks.

3.1 Shape Modeling

In contrast to a photographer, the illustrator has the possibility to arrange the lighting proportions to achieve an optimal perceptibility of the structure's shape and surface. For a detailed analysis of these methods, we refer to Hodges [18].

Conventional and Reflected Lighting: A light source from the top left is used corresponding to the perception of the daylight. In reality, *reflected light* ensures that shadow territories appear not completely black. The *reflected light* is used as scatter lighting to make structures apparent that are not directly lighted.

Plateau Lighting: *Plateau lighting* is suitable to emphasize the silhouette and for figure-ground-separation. This results in an enhanced shape perception. For roundish structures, the center appears brighter and border areas are darker.

Backlighting: In contrast to the *plateau lighting* the *backlighting* separates an object from a dark background. The *backlighting* has a lower effect than the *plateau lighting* and may be used at the same time.

Raking Light: Details are optimally recognizable when the light is approximately tangential to the surface, although the *raking light* does not affect the overall impression of a realistic main light source.

For the illustration of the object's surface, additional techniques are used, which are not related to the incidence of light. *Ridges and valleys* may be especially highlighted to improve the conventional lighting methods.

3.2 Depth Cueing and Contrast Enhancement

Because a 2D illustration has no third dimension, illustrators are using optical hints to visualize the depth of the scene more clearly. In nature, the scattering of light and fog are the reason that farther objects appear brighter and with low contrast. The *atmospheric perspective* gives a rough idea of depth and clarifies the whole scene, because fore- and background are optically separated.

Furthermore, the distance between overlapping objects may be emphasized by mapping a *shadow* onto the rear object. This *shadow* seems to be caused by a light source from the direction of the observer. The shadow is independent of the *main light* source and is only applied locally. Thus, it differs from a shadow in reality. Additionally, the front object is brightened at the boundaries to aid the contrast to the rear object, similar to the *backlighting*.

In addition, a *high contrast range* makes illustrations appear more dynamic and increases the amount of perceived information at the same time. Therefore, the whole contrast range should be utilized.

4 Emulation of Shading Arrangements

Techniques like stippling or hatching are based on the placement of black stipples or strokes to create an impression of gray values. Thus, the actual gray value of one resulting pixel is black or white, but for every pixel a target gray value must be determined. Typically, the stippling or hatching renderer computes the final gray value based on a local lighting model. In the following, a system for a flexible computation of an illustrative shading is described.

4.1 Requirement Analysis

The goal of medical illustrations is to depict the human anatomy expressively. The emphasis is therefore placed on the illustration of shape and spatial relationships. Since the computer generated illustrative shading for 3D scenes is an extension of the medical textbook illustrations, the already discussed techniques are well suited.

To adopt the computer generated shading to different geometric models, it is necessary that the different shading arrangements are provided as parameters in an appropriate user interface. Since the parameters may have different impact on the visualization, a differentiated examination of their involvement for the final rendering is necessary.

This results in a high-dimensional parameter space. Therefore, a guided interaction is essential which provides a good trade-off between illustrative freedom and simplicity of use.

4.2 Realization of the Extended Shading

The algorithms for computing the single parameters pose different computational requirements. The whole pipeline is completely passed to the GPU to achieve interactive frame rates. The concept of Framebuffer Objects (FBO) is used [19]. Every parameter is mapped on one FBO. To commit additional static values to the GPU, e. g. curvature, user vertex attributes provided by OpenGL are used.

Lighting is computed per fragment and therefore for every point of the surface. However, histogram operations have to be computed for the whole image. Hence, the local brightness of a surface is dependent on the surrounding. It is not possible to render the brightness, as customary, in a normal fragment shader. Therefore, we suggest to compute the brightness distribution in a separate process and make it available in the final rendering process.

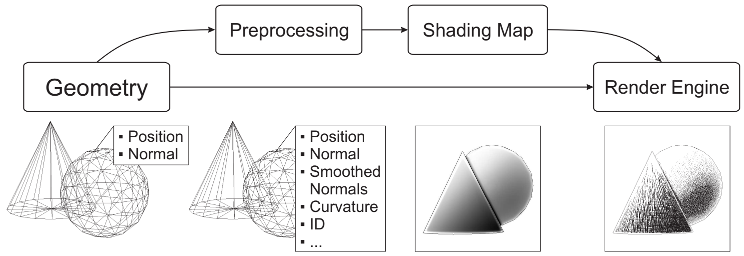


Fig. 1. View-independent values are mapped onto the surface and rendered into shading maps. The combined shading maps are used for brightness for the final rendering.

Some parameters are computed more efficiently in object space, e. g., lighting and curvature, others in image space (like histogram operations). For the combination of both spaces it is necessary to transfer them into a uniform space. Therefore, the image space is appropriate, because the needed object parameters are computable in image space, according to the G-buffers [11].

All static values are computed once in object space, to save computation for the single rendered frames. The combination of all parameters yields to the so-called extended shading map: a view of the scene with current camera position and screen resolution, indicating the brightness for every single pixel. The final renderer reads the brightness value per pixel and produces the actual rendering using the desired technique, like stippling, hatching or color saturation (Fig. 1).

According to the G-buffers, for each parameter an individual parameter buffer is generated and stored in a FBO. Some of these buffers are generated directly by reading the hardware-buffers (e. g., z-buffer), others are computed after a further image processing.

4.3 Buffer Combination

The sequence used in the combination of all parameter buffers significantly influences the resulting image. In the following equations parameter buffer P and shading map M are regarded as 2D arrays with corresponding weights $w = [0, 1] \in \mathbb{R}$. The addition of two parameter buffers $P = P_i + P_j$ is performed element-wise:

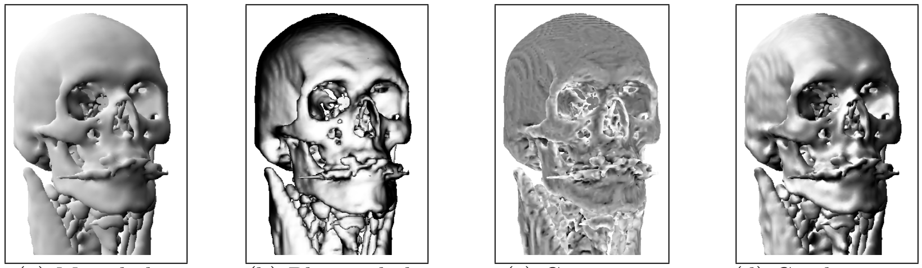
$$\forall x, y : P(x, y) = P_i(x, y) + P_j(x, y) \tag{1}$$

The product $P = w \cdot P_i$ is accordingly:

$$\forall x, y : P(x, y) = w \cdot P_i(x, y) \tag{2}$$

The first parameters, that are *conventional*, *reflected*, *plateau* and *raking light*, as well as *surface shape* and *curvature*, represent the distribution of brightness. According to [5, 6], the parameter buffers are added up to a weighted sum:

$$M = \frac{1}{\sum w_i} \cdot \sum w_i \cdot P_i \tag{3}$$



(a) Main light (b) Plateau light (c) Curvature (d) Combination

Fig. 2. Combination of the surface representation parameter: (d) is the weighted sum of (a), (b) and (c), and an additional histogram equalization.

The resulting shading map M is normalized and subsequently modified by a *histogram equalization* to enhance the contrast (Fig. 2(d)). The equalization has to be performed at this stage, because all of the following parameters only change the local brightness. If they are added before the equalization, the local changes affect the whole scene. For example, the *atmospheric perspective* would also darken the foreground.

The following parameter buffers need individual considerations due to the different characteristics. Therefore, the sequence of modifications has to be regarded. In the following, the result M' accords to M in the subsequent equation.

Although *backlighting* and *depth shadow* are lighting techniques, they are added separately, because their impact should be limited to the object's boundaries. An inclusion of these parameters into the weighted sum would result in a lower contrast, as explained above. Instead, the *backlighting* is added to the assembled shading map and the result is clamped to $[0, 1]$:

$$M' = \underset{[0,1]}{clamp}\,(M + w \cdot P) \tag{4}$$

The depth shadow buffer should only influence pixels in shadow regions. According to [12], the buffer values lie in the range $[-1, 0]$. With the following equation, the brightness is shifted to 0 in shadow regions and other regions are left unattended:

$$M' = (1 + w \cdot P) \cdot M \tag{5}$$

The *atmospheric perspective* is realized by contrast reduction according to [15]. In addition, the *background intensity* $I_b \in [0, 1]$ is freely adjustable:

$$M' = (1 - w_{ap} \cdot P_{ap}) \cdot M + w_{ap} \cdot P_{ap} \cdot I_b \tag{6}$$

The integration of *feature lines* is carried out at last, because the lines should not be effected by any other parameter. Additionally, the thickness of the *feature lines* may be varied by a weight. Due to the functionality of edge filters, the buffer has to be inverted after weighting. The feature lines are added by the minimum operation to ensure their display on bright areas and to avoid the brightening of dark areas:

$$M' = \min(1 - w \cdot P, M) \tag{7}$$

It is also possible to convey the *atmospheric perspective* by varying the *feature line* thickness. The size of the edge filter is chosen according to the value of the *atmospheric perspective.*

5 Parameter Adjustment and Simplification

All specified parameters will affect the visualization. Again, the impact of a parameter is controlled by a weight and possibly adjustments. Thus, to obtain the final visualization, 20 adjustments and weights have to be tuned (Fig. 3). Therefore, default values and the summary of parameters to high-level attributes have to be determined.

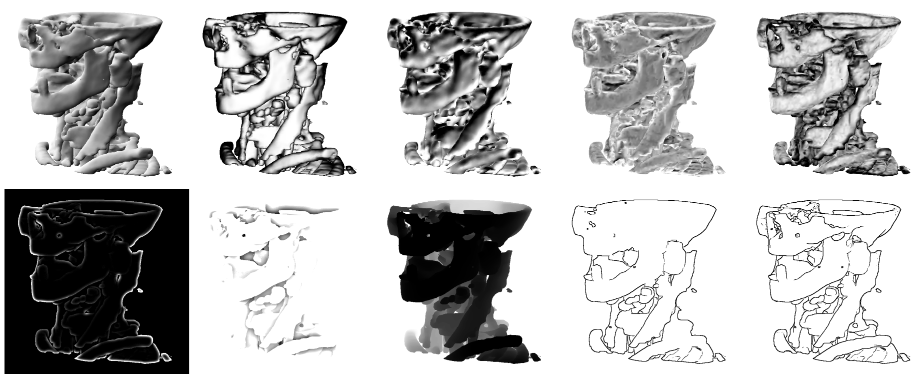
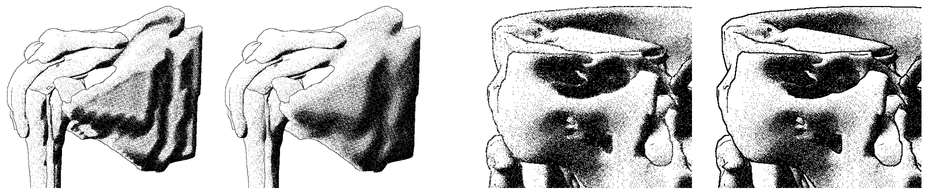


Fig. 3. From top left to bottom right: conventional, reflected, and plateau lighting, raking light, surface shape, curvature, backlighting (black background), depth shadow (increased by 1), atmospheric perspective, silhouettes (inverted), feature lines (inverted).

5.1 Parameters for the Shading Arrangement

So far, we introduced some parameters for illustrative shading. In the following, for every parameter a realization is described and the adjustment is discussed. To support a flexible parameterization, no combined light models are used, as recommended by [7]. Instead, the single lighting techniques are applied separately to achieve an efficient parameter controlling.

The *conventional and reflected lighting* is achieved by a modified diffuse lighting according to [20], where no separate parameter for the *reflected light* is necessary. The direction of the *main light* is selectable. Additionally, the usage of



(a) Detail frequency. (b) Feature lines.

Fig. 4. (a) A shoulder dataset with strong artifacts. Smoothing removes artifacts, but features also disappear. (a) Slightly attenuated feature lines fit better into the overall rendering.

normal smoothing is possible to get a softer shading. The *plateau lighting* has no adjustments and corresponds to diffuse lighting from the viewing direction. Local lighting according to [13] emulates *raking light*. The direction is parallel to the *main light*. The parameters are *toon shading* and *detail frequency*.

The *surface shape* is emphasized by ridges and valleys. *Curvature* is visualized according to [6] by darkening surface areas with high curvature. Both parameters need no adjustment.

Backlighting is extracted from the unsharping mask of the depth buffer according to [12], but only positive values are considered. The size of the kernel is variable. Like *backlighting*, the *depth shadow* is also extracted from the unsharping mask of the depth buffer, but only negative values are considered. The *atmospheric perspective* is computed by the depth buffer according to [15]. Adjustments relate to minimal and maximal depth, slope exponent as well as background intensity. *Silhouette and feature lines* are obtained by edge filtering of object and depth buffer according to [11]. The thickness of the lines is defined by filter type and dimension. *Histogram equalization* is chosen for contrast enhancement [14]. The equalization permits a frame-coherent rendering, in contrast to simple histogram stretching, and needs no adjustment.

5.2 Default Values

The adjustment may be improved by providing adequate default values for the parameters, which the user will likely accept in most cases.

Detail Frequency: Since we consider a parameterization for medical surfaces, the resolution and quality of the objects is an important aspect. Strong artifacts should result in a low *detail frequency* of the *raking light* and an increased *smoothing* (Fig. 4(a)). The *detail frequency* cannot be abridged and must always be adjusted.

Weighting of the Feature Lines: *Feature lines* should not be too prominent to keep the illustrative character and to fit better into the overall rendering (Fig. 4(b)). A favorable value is 80% of the full intensity.

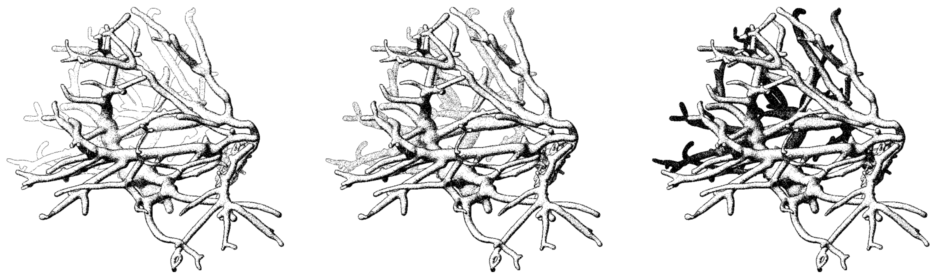


Fig. 5. A liver vessel tree with atmospheric perspective and different background intensities. The best attenuation is achieved on the left.

Direction of the Conventional Main Light: The *main light* shines from the top left to the lower right [18]. Because the main light vector is perpendicular to the viewing vector, the modified diffuse lighting is optimally used.

Weighting of the Raking Light: To maintain the impression of the *main light*, the *raking light* weight should be in a 1 : 2 ratio to *main and plateau light*. When this ratio is maintained, the *toon shading* adjustment alone is sufficient, thus, the parameterization is further simplified.

Background Intensity: According to medical textbooks, the background intensity of the *atmospheric perspective* is white, otherwise the background gains more attention (Fig. 5).

5.3 High-Level Attributes

Individual parameters may be combined to high-level attributes. For example, different parameters influence the smoothness of the surface and the border enhancement. By a combination of these parameters, the user's effort is reduced significantly to only one weight. The *detail frequency* is also kept as a high-level attribute, since it cannot be abridged and must always be adjusted.

Border Enhancement: Besides the *feature line* thickness, the ratio between *main and plateau light* affects the intensity of the object's border. The ratio may be combined with feature line thickness to one high-level attribute (Fig. 6).

Detail Amplitude: Details are emphasized by the toon shading factor of the *raking light* as well as the *curvature weight* and extenuated by *normal smoothing* of the *main light*. These parameters are combined to a detail amplitude (Fig. 7). The extremes correspond to high details and additional smoothing respectively. The medium setting is according to a normal lighting.

Spatial Effect: *Back light*, *depth shadow* and *atmospheric perspective* contribute to the spatial effect of a scene. Thus, their weights may be combined to one high-level attribute.

5.4 Effectiveness of the Manual Parameterization

The presented approaches aim at a more simple parameterization for the user. Therefore, it is essential to verify whether a simple parameterization could be

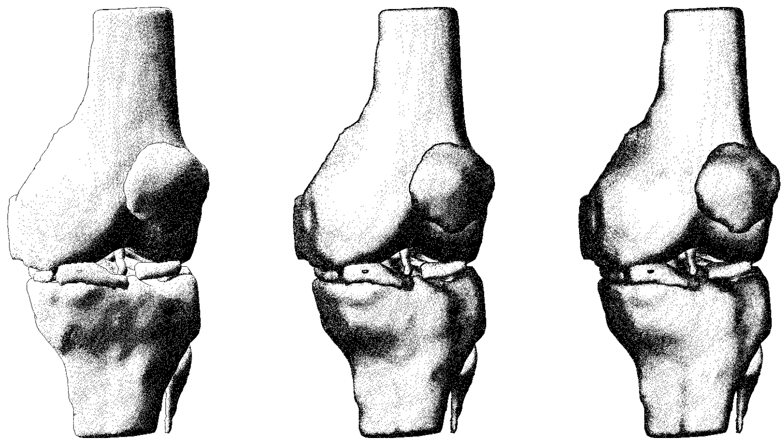


Fig. 6. Border enhancement of a knee dataset from left to right: main light decreases, feature line thickness and plateau light increases.

actually achieved. It is difficult to evaluate how long a user needs to achieve the desired shading, since the appreciated quality depends not only on the specific application example, but also on the personal aesthetic sense. Instead, we investigated the effectiveness of the parameterization, i. e., how goal-oriented the user may change the parameterization.

Since every modification at the shading is immediately visible in real-time, the impact of every single parameter is directly observable. In principle, the user may give a trial on the impact of all parameters. Without any simplification 20 parameter adjustments and weights are available and the contributions of the parameters partially reinforce or interfere. Thus, the application of default values and high-level attributes is highly recommended.

By using the high-level attributes, the parameterization is reduced to four settings. The procedure is additionally eased by the following strategy:

1. Border Enhancement
2. Detail Amplitude
3. Detail Frequency
4. Spatial Effect

The impact of these attributes is almost independent from each other. Due to the descriptive change of the comprehensive setting, even a novel user should be able to achieve a final setup very fast.

6 Conclusions and Future Work

We presented a new approach for an illustrative shading framework. For an efficient and flexible parameterization, we introduced single parameters and high-level attributes, which incorporate a wide variety of illustrative styles. The frame-

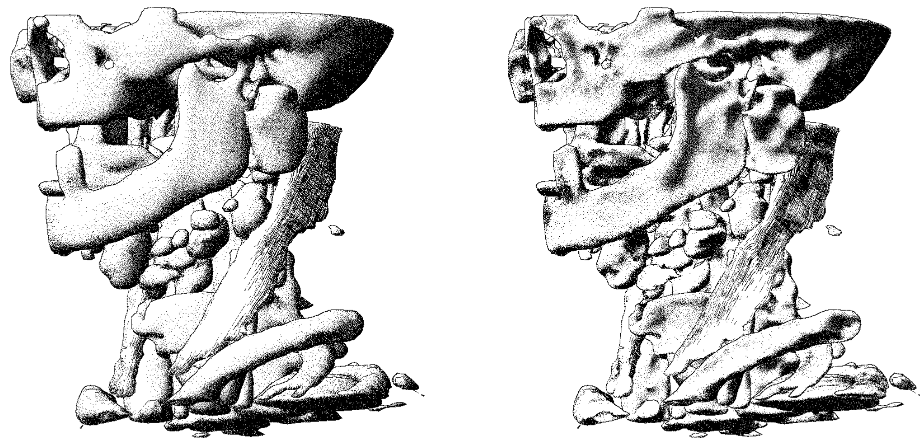


Fig. 7. Detail amplitude: on the left, only main light and smoothed normals are used. Rightwards, smoothing decreases, toon shading and curvature weight increases. For the muscles, a hatching technique was applied.

work is completely hardware-accelerated by the GPU using framebuffer objects. Thus, real-time rendering is achieved even for very complex geometric models.

We tested our framework with all the scenes presented in this paper with different viewport sizes (Tab. 1). The number of triangles ranges from $25K$ to $225K$. The testing environment was a workstation with $3.2GHz$ CPU, $1GB$ RAM and a NVidia GeForce 7900GS graphic card. The variation of the parameters has no impact on the frame rates. The frame rates are measured for the shading map generation process only. Stippling reduces the frame rate by 5%.

Table 1. Frames per second for different scenes and a viewport size of 512^2. Extended shading maps are compared to conventional shading.

Model	Fig. 2	Fig. 4(a)	Fig. 5	Fig. 6	Fig. 7
Triangles	147480	39863	225080	25224	94399
Ext. shading	10	15	7.5	20	14
Conv. shading	30	60	15	60	45

In the future, we intend to consider an *importance* parameter to further refine our visualizations. To simplify the parameterization, other interaction facilities will be explored, e. g., to represent a high-level attribute in a graphical editor. We want to overcome the limitation of opaque surfaces. So far, first tests brought good results by using depth peeling. A step towards an automatic parameterization would be a similar approach to [8]. Also, methods for other shading improvements used in illustrations, like composition, balance and harmony, could improve the quality of the presented images.

Acknowledgment We thank MeVis Research for providing advanced MeVisLab features. This work was supported by the Deutsche Forschungsgemeinschaft (DFG) (Priority Programme 1124, PR 660/3-2).

References

1. Bruckner, S., Gröller, M.E.: Style Transfer Functions for Illustrative Volume Rendering. Computer Graphics Forum **26**(3) (2007) 715–724
2. Tietjen, C., Isenberg, T., Preim, B.: Combining Silhouettes, Shading, and Volume Rendering for Surgery Education and Planning. In: EuroVis. (2005) 303–310, 335
3. Svakhine, N., Ebert, D.S., Stredney, D.: Illustration motifs for effective medical volume illustration. IEEE Computer Graphics & Applications **25**(3) (2005) 31–39
4. Ritter, F., Hansen, C., Dicken, V., Konrad, O., Preim, B., Peitgen, H.O.: Real-Time Illustration of Vascular Structures. IEEE TVCG **12**(5) (2006) 877–884
5. Akers, D., Losasso, F., Klingner, J., Agrawala, M., Rick, J., Hanrahan, P.: Conveying Shape and Features with Image-Based Relighting. In: IEEE Visualization. (2003) 349–354
6. Hamel, J.: A New Lighting Model for Comuter Generated Line Drawings. PhD thesis, University of Magdeburg (2000)
7. Lee, C.H., Hao, X.: Geometry-Dependent Lighting. IEEE TVCG **12**(2) (2006) 197–207
8. Shacked, R., Lischinski, D.: Automatic Lighting Design Using a Perceptual Quality Metric. Computer Graphics Forum **20**(3) (2001) 215–227
9. Gumhold, S.: Maximum Entropy Light Source Placement. In: IEEE Visualization. (2002) 275–282
10. Yuan, X., Nguyen, M.X., Zhang, N., Chen, B.: Stippling and Silhouettes Rendering in Geometry-Image Space. In: Eurographics Symposium on Rendering, Eurographics Association (2005) 193–200
11. Saito, T., Takahashi, T.: Comprehensible Rendering of 3-D Shapes. Computer Graphics **24**(4) (1990) 197–206
12. Luft, T., Colditz, C., Deussen, O.: Image Enhancement by Unsharp Masking the Depth Buffer. ACM Transactions on Graphics **25**(3) (2006) 1206–1213
13. Rusinkiewicz, S., Burns, M., DeCarlo, D.: Exaggerated Shading for Depicting Shape and Detail. ACM Transactions on Graphics **25**(3) (2006) 1199–1205
14. Scheuermann, T., Hensley, J.: Efficient Histogram Generation Using Scattering on GPUs. In: Proc. of the 2007 Symposium on Interactive 3D Graphics. (2007) 33–37
15. Ebert, D.S., Rheingans, P.: Volume Illustration: Non-Photorealistic Rendering of Volume Models. In: IEEE Visualization. (2000) 195–202
16. Baer, A., Tietjen, C., Bade, R., Preim, B.: Hardware-Accelerated Stippling of Surfaces Derived from Medical Volume Data. In: EuroVis. (2007) 235–242
17. Praun, E., Hoppe, H., Webb, M., Finkelstein, A.: Real-Time Hatching. In: SIGGRAPH. (2001) 579–584
18. Hodges, E.R.S.: The Guild Handbook of Scientific Illustration. Van Nostrand Reinhold (1989)
19. Juliano, J., Sandmel, J.: OpenGL Framebuffer Object Extension (December 2007)
20. Gooch, A., Gooch, B., Shirley, P., Cohen, E.: A Non-Photorealistic Lighting Model for Automatic Technical Illustration. In: SIGGRAPH. (1998) 447–452