

Viewpoint Information Channel for Illustrative Volume Rendering

Abstract

This paper introduces a volume rendering framework based on the information channel constructed between the volumetric data set and a set of viewpoints. From this channel, the information associated to each voxel can be interpreted as an ambient occlusion value that allows to obtain illustrative volume visualizations. The use of the voxel information combined with the assignation of color to each viewpoint and non-photorealistic effects produces an enhanced visualization of the volume data set. Voxel information is also applied to modulate the transfer function and to select the most informative views.

Keywords: Illustrative visualization, ambient occlusion, viewpoint selection

1. Introduction

In the last decades many different strategies have been proposed to visualize and explore volume data sets efficiently. One of the main challenges is to obtain renderings that adapt the appearance of the data to the specific task satisfying user requirements. For instance, in medical visualization, for diagnosis and pre-operative planning, we seek realistic renderings with transfer functions designed to show the inherent structures within a given volume. On the other hand, in educational environments, non-photorealistic techniques are preferred since they are able to simplify data, producing clearer images than traditional photorealistic methods. Moreover, the demand of interactivity when exploring volume data has led to the development of new strategies to accelerate the rendering process. In this context, focus+context and viewpoint-based strategies improve the exploration efficiency by directing the users to the most informative parts of the data. GPU-based implementations which exploit hardware capabilities have been also proposed.

In this paper, we propose a volume visualization system based on the

information channel defined between the voxels of a volume data set and a set of viewpoints. This channel is obtained from the reversion of the viewpoint channel defined in Viola et al. [26]. Thus, instead of analyzing how a viewpoint sees the volume data set, we focus on how a voxel “sees” the viewpoints. The shared information of each voxel with the set of visible viewpoints is interpreted as a visibility quality descriptor of a voxel that provides a natural ambient occlusion value [14, 30].

The proposed framework results in a flexible system for producing realistic and non-photorealistic renderings in an automatic way. The use of the voxel information combined with the assignation of color to each viewpoint and non-photorealistic effects produces an enhanced visualization of the volume data set. Voxel information is also applied to modulate the transfer function in order to focus on or highlight the most informative parts of the data set. Finally, a new viewpoint selection measure based on voxel information is introduced and compared with other information-theoretic viewpoint measures. The proposed framework has been partially implemented using Compute Unified Device Architecture (CUDA)¹, allowing to exploit the capabilities of modern GPUs.

The paper is organized as follows. Section 2 describes previous work related to volumetric shadowing, volume illustration, and viewpoint selection. Section 3 introduces an information channel which enables us to calculate the information associated to each voxel. Section 4 presents different visualization applications that can be derived from the voxel information. Section 5 defines a viewpoint quality measure based on voxel information. Finally, Section 6 presents the conclusions and future work.

2. Related Work

In this section, we describe the main areas of research dealt with in this paper.

2.1. Volume Shadowing and Illustrative Techniques

Although the integration of global illumination effects in direct volume rendering enhances volume data interpretation, its high computational cost overcomes its application. Different strategies have been proposed to simulate these effects preserving interactive frame rates.

¹<http://www.nvidia.com/cuda>

Behrens and Ratering [1] integrated in a texture-based volume renderer shadow maps that store the intensities computed according to light and transfer function conditions. Shadows can also be computed with half-angle slicing, either by using simultaneous slicing for rendering and shadow computation [13], or combined with splatting [29]. Other strategies are based on the ambient occlusion technique introduced by Landis [14], a simplified version of the obscurances illumination model [30]. Stewart [21] introduced vicinity shading that simulates illumination of isosurfaces by taking into account neighboring voxels. This is achieved by computing an occlusion volume and storing it in a shading texture that is accessed during rendering. Tarini et al. [23] refined this model to increase the performance. Wyman et al. [28] presented a method that supports the simulation of direct lighting, shadows and interreflections by storing pre-computed global illumination in an additional volume to allow viewpoint, lighting and isovalue changes. Ropinski et al. [18] computed a local histogram for each voxel from the voxel neighbourhood, by accumulating intensities weighted by inverse squared distances. These local histograms can be combined interactively with the user-defined transfer function to give an effect similar to local ambient lighting. Hernell et al. [10] obtained the incident light intensity, arriving at a voxel, by integrating the attenuated transfer function density for each voxel and within a sphere surrounding it. Ruiz et al. [19] introduced an obscurance-based volume rendering framework where obscurances include color bleeding effects without additional cost.

Illustrative techniques are suitable for emphasizing certain features or properties while omitting or greatly simplifying less important details [17]. The most popular styles, such as stippling, hatching, and silhouettes are from the pen-and-ink family [5, 16]. To incorporate illustrative effects in a volume renderer, Kindlmann et al. [12] utilized curvature-based transfer functions. Hauser et al. [9] proposed the two-level volume rendering concept which allows focus+context visualization of volume data. Different rendering styles, such as direct volume rendering and maximum intensity projection, are used to emphasize objects of interest while still displaying the remaining data as context. Viola et al. [27] introduced importance-driven volume rendering, where features within the volumetric data are classified according to object importance. Bruckner et al. [3] presented context-preserving volume rendering, where the opacity of a sample is modulated by a function of shading intensity, gradient magnitude, distance to the eye point, and previously accumulated opacity.

2.2. Viewpoint Information Channel

Automatic selection of the most informative viewpoints is a very useful focusing mechanism in visualization of scientific data, guiding the viewer to the most interesting information of the data set. Best view selection algorithms have been applied to computer graphics domains, such as scene understanding and virtual exploration [20, 25], and volume visualization [2, 4, 22, 26]. Shannon’s information measures, such as entropy and mutual information, have been used in these fields to measure the quality of a viewpoint from which a given scene is rendered. Viewpoint entropy, first introduced in [24] for polygonal models, has been applied to volume visualization in [2, 22]. In particular, Bordoloi and Shen [2] obtained the goodness of a viewpoint from the entropy of the visibility of the volume voxels. Viola et al. [26] proposed a viewpoint information channel and used the viewpoint mutual information to automatically determine the most expressive view on a selected focus. A unified information-theoretic framework for viewpoint selection, ambient occlusion, and mesh saliency for polygonal models has been presented in [6, 7]. Next we review the definitions of viewpoint information channel, viewpoint mutual information, and viewpoint entropy in volume visualization.

To select the most representative or relevant views of a volume data set, a viewpoint quality measure, the *viewpoint mutual information*, was defined [26] from an *information channel* $V \rightarrow Z$ between the random variables V (input) and Z (output), which represent, respectively, a set of viewpoints \mathcal{V} and the set of objects (or voxels) \mathcal{Z} of a volume data set (see Figure 1(a)). Viewpoints are indexed by v and voxels by z . The capital letters V and Z as arguments of $p(\cdot)$ are used to denote probability distributions. For instance, while $p(v)$ denotes the probability of a single viewpoint v , $p(V)$ denotes the input probability distribution of the set of viewpoints.

The information channel $V \rightarrow Z$ is characterized by a probability transition matrix (or conditional probability distribution) which determines, given the input, the output probability distribution (see Figure 1(b)). The main elements of this channel are the following:

- The transition probability matrix $p(Z|V)$, where each conditional probability $p(z|v)$ is given by the quotient $\frac{vis(z|v)}{vis(v)}$, where $vis(z|v)$ is the visibility of voxel z from viewpoint v and $vis(v) = \sum_{z \in \mathcal{Z}} vis(z|v)$ is the captured visibility of all voxels over the sphere of directions. The visibility $vis(z|v)$ of a voxel z is considered as the contribution of this

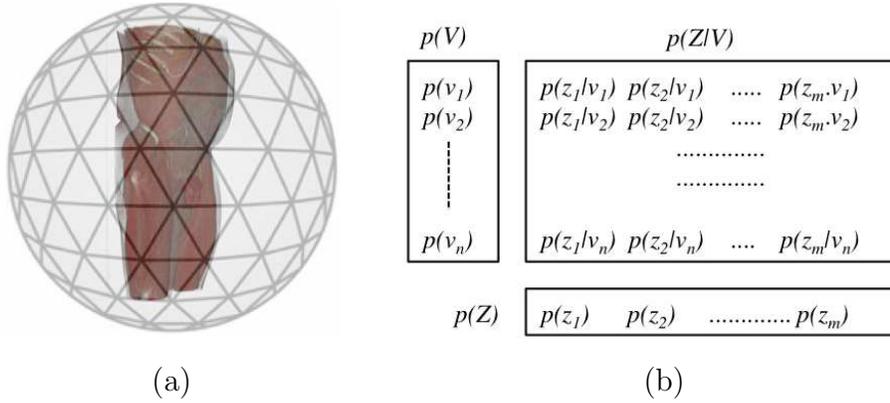


Figure 1: Viewpoint information channel. (a) Sphere of viewpoints of a voxel model. (b) Probability distributions of channel $V \rightarrow Z$.

voxel to the final image as rendered from viewpoint v (see Section 3). Conditional probabilities fulfill that $\sum_{z \in \mathcal{Z}} p(z|v) = 1$.

- The input probability distribution $p(V)$ is given by the probabilities of selecting each viewpoint, where an element $p(v)$ of this probability distribution can be interpreted as the importance of viewpoint v . In this paper, $p(V)$ is obtained from the normalization of the captured visibility of the data set over each viewpoint. Thus, $p(v) = \frac{vis(v)}{\sum_{v \in \mathcal{V}} vis(v)}$ expresses how much volume is visible from viewpoint v . In [26], uniform importance was assigned to each viewpoint.
- From $p(V)$ and $p(Z|V)$, the output probability distribution $p(Z)$ is given by

$$p(z) = \sum_{v \in \mathcal{V}} p(v)p(z|v), \quad (1)$$

which expresses the average visibility of each voxel.

The degree of *dependence* or *correlation* between a set of viewpoints \mathcal{V} and the volume data set \mathcal{Z} , expressed by the *mutual information* (MI) between V and Z , is given by

$$\begin{aligned} I(V; Z) &= \sum_{v \in \mathcal{V}} p(v) \sum_{z \in \mathcal{Z}} p(z|v) \log \frac{p(z|v)}{p(z)} \\ &= \sum_{v \in \mathcal{V}} p(v) I(v; Z), \end{aligned} \quad (2)$$

where

$$I(v; Z) = \sum_{z \in \mathcal{Z}} p(z|v) \log \frac{p(z|v)}{p(z)} \quad (3)$$

is defined as the *viewpoint mutual information* (VMI), which measures the degree of dependence between the viewpoint v and the set of voxels. Note that $I(v; Z)$ is not properly a mutual information, but the contribution of viewpoint v to mutual information $I(V; Z)$. In this framework, the most *representative* viewpoint is defined as the one that has minimum VMI. Low values correspond to more independent views, showing the maximum possible number of voxels in a balanced way. The term ‘balance’ is used here to express that the visibility distribution $p(Z|v)$ of v is similar to $p(Z)$. This similarity is expressed by the Kullback-Leibler distance between $p(Z|v)$ and $p(Z)$ (see [6, 26]). This distance is zero when $p(Z|v) = p(Z)$. On the other hand, high values of $I(v; Z)$ mean a high dependence between viewpoint v and the object, indicating a highly coupled view (for instance, between the viewpoint and a small number of voxels with low average visibility). In [26], it has been shown that one of the main properties of VMI is its robustness to deal with any type of discretisation or resolution of the volume data set. The same behavior can be observed for polygonal data [6].

From the viewpoint information channel, the *viewpoint entropy* (VE) of viewpoint v is defined by

$$H(Z|v) = - \sum_{z \in \mathcal{Z}} p(z|v) \log p(z|v). \quad (4)$$

VE measures the degree of uniformity of the visibility distribution $p(Z|v)$ at viewpoint v . The best viewpoint is defined as the one that has maximum VE, that would be obtained when a certain viewpoint can see all the voxels with the same projected visibility. On the other hand, minimum VE would be obtained when most of the visibility is captured from few voxels. In the volume rendering field, Bordoloi and Shen [2] and Takahashi et al. [22] adapted the polygonal viewpoint entropy [24] to a volume data set. Some examples of views corresponding to the maximum and minimum values of VMI and VE are shown in Section 5 (Figure 12).

3. Voxel Information

As we have seen in Section 2.2, the information associated with each viewpoint (VMI) is obtained from the definition of the channel between the

sphere of viewpoints and the voxels (or objects) of the volume data set. In this section, the voxel information is defined from the reversed channel $Z \rightarrow V$, so that Z is now the input and V the output. The probability distributions of this channel are shown in Figure 2(a). Note that MI is invariant to the reversion of the channel: $I(V; Z) = I(Z; V)$. The idea of reversing the channel was introduced in [7] for polygonal models, together with the computation of the information associated to a polygon.

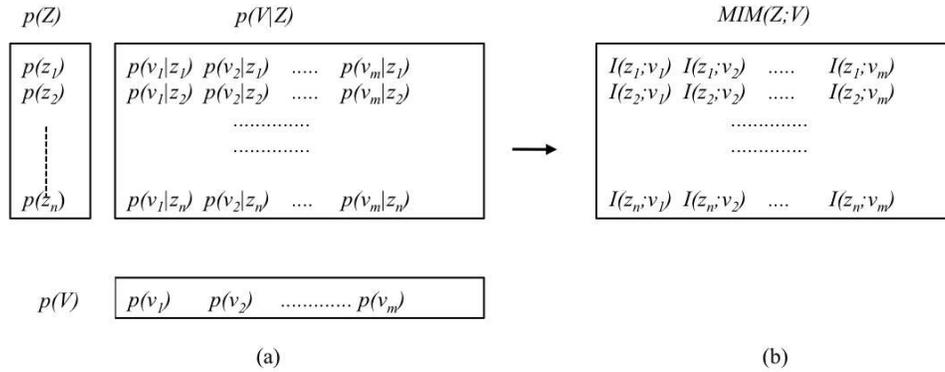


Figure 2: (a) Probability distributions of channel $Z \rightarrow V$, used to compute the voxel mutual information. (b) The elements of matrix $MIM(Z; V)$ are given by $I(z_i; v_j) = p(v_j|z_i) \log \frac{p(v_j|z_i)}{p(v_j)}$ and used to calculate the color ambient occlusion in Section 4.2.

From the Bayes theorem $p(v, z) = p(v)p(z|v) = p(z)p(v|z)$, MI (see Equation 2) can be rewritten as

$$\begin{aligned}
 I(Z; V) &= \sum_{z \in \mathcal{Z}} p(z) \sum_{v \in \mathcal{V}} p(v|z) \log \frac{p(v|z)}{p(v)} \\
 &= \sum_{z \in \mathcal{Z}} p(z) I(z; V),
 \end{aligned} \tag{5}$$

where

$$I(z; V) = \sum_{v \in \mathcal{V}} p(v|z) \log \frac{p(v|z)}{p(v)} \tag{6}$$

is the contribution of voxel z to $I(Z; V)$ and is defined as the *voxel mutual information* (VOMI). This represents the degree of correlation between the voxel z and the set of viewpoints, and can be interpreted as the information

associated with voxel z . Analogous to VMI, low values of VOMI can correspond to voxels seen by a large number of viewpoints in a balanced way. That is, the lowest values of VOMI correspond to the voxels with conditional probability distribution $p(V|z)$ similar to $p(V)$. The opposite happens for high values.

Figure 3 shows for different data sets the VOMI maps computed using 42 viewpoints and colored using the thermal scale represented in Figure 3(e). Warm colors correspond to high VOMI values and cool colors to low ones.

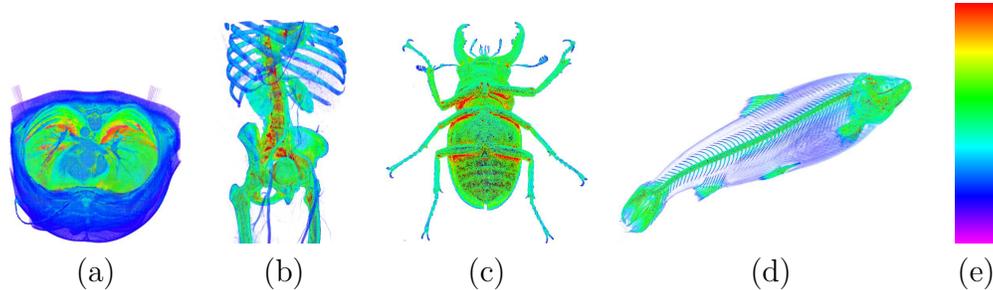


Figure 3: VOMI maps generated using 42 viewpoints for different models and transfer functions: (a) CT body, (b) CT body (skeleton), (c) CT beetle, and (d) CT salmon. Maps are colored using the thermal scale in (e).

The great potential of VOMI is that it allows varied interpretations that can be used in different visualization applications, such as volume illustration and viewpoint selection. Both will be discussed in detail in Sections 4 and 5, respectively.

In Figure 4 we represent the different steps required for computing the VOMI of a voxel model. The process starts classifying the volume data by defining a transfer function. Then, a ray casting is performed considering the volume data set centered in a sphere of viewpoints and the camera looking at the center of this sphere. For each viewpoint a histogram of visibilities is created and then used to estimate $p(Z|v)$. Using Equation 1 and the Bayes theorem, $p(Z)$ and $p(V|Z)$ can be obtained from both $p(V)$ and $p(Z|V)$. Finally, the VOMI map is obtained.

Since $p(Z|V)$ is a very huge matrix it cannot be stored in memory and we have to compute its rows, $p(Z|v)$, every time we need them. Thus, the VOMI computation requires executing the ray casting stage three times, to compute $p(V)$, then $p(Z)$, and then the VOMI map.

We have implemented the most costly computations of our framework

(ray casting, computation of $p(Z)$, and VOMI) using CUDA in order to speed up the process compared to a pure CPU implementation.

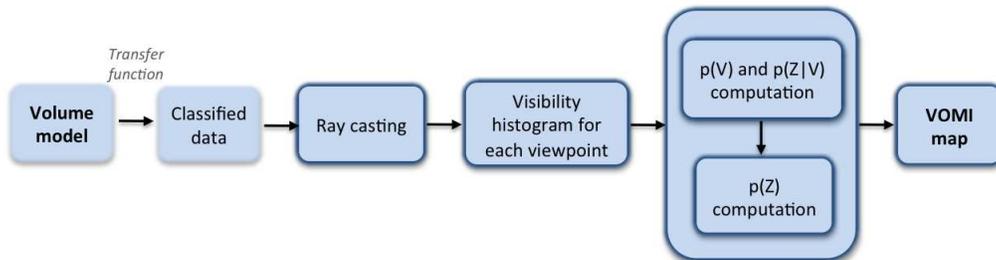


Figure 4: Overview of the VOMI pipeline.

The first and most complex part done in CUDA is the ray casting needed to compute $p(Z|v)$ for a viewpoint v . In this ray casting we need to fill a table with the visibility of each voxel from a given viewpoint, from which $p(Z|v)$ can be obtained. This visibility is a real value equal to the contribution of the voxel to the final image according to its opacity and also to the opacity of the preceding voxels in each ray that visits it [15]. For example, a voxel that is seen from one ray, that is fully opaque, and that is not occluded at all by any other voxel in this ray, has a visibility of 1. To update the visibility table, we need atomic operations in order to avoid race conditions (the same voxel may be visited by two or more neighboring rays), but unfortunately CUDA doesn't support atomic operations with floating point values. To overcome this limitation we multiply the visibility by a big constant (we have used 1 million) and truncate the result to obtain an integer that is atomically added to the visibility table; this is equivalent to working with fixed point precision. At the end of the process, we divide each value by the same constant to get real values. So, we have two kernels for this whole task: the first one does the ray casting and fills the integer table, and the second converts this table to a table of floating point numbers.

The other two parts that we have implemented in CUDA are fairly simple. One of them is the computation of the voxel probabilities $p(Z)$ (see Equation 1), with a kernel that just accumulates $p(v)p(Z|v)$ for a given viewpoint v . The other part is the computation of VOMI, where the kernel accumulates the term $p(v|z) \log \frac{p(v|z)}{p(v)}$. Remember that $p(v|z) = p(v) \frac{p(z|v)}{p(z)}$, for each voxel z and viewpoint v .

Table 1 shows the times to compute the VOMI map for different data

sets and transfer functions with 42 viewpoints. We also report the individual times to do the ray casting and fill the visibility histogram, accumulate $p(Z)$ and accumulate VOMI, each for 1 viewpoint, because these are the processing bottlenecks that we have implemented in CUDA. The total time is the result of adding the ray casting 3×42 times, plus $p(Z)$ 42 times, plus VOMI 42 times, plus additional costs (CPU, sincronizations, memory transfers, etc.).

	Ray casting (1 v)	$p(Z)$ (1 v)	VOMI (1 v)	Total (42 v)
Body	141.99 ms	13.09 ms	14.02 ms	29.716 s
Skeleton	108.09 ms	13.09 ms	13.41 ms	26.262 s
Beetle	158.93 ms	21.54 ms	19.92 ms	40.184 s
Salmon	114.65 ms	14.96 ms	14.78 ms	28.416 s

Table 1: Times to compute the VOMI maps for the data sets shown in Figure 3. First and second rows: body and skeleton ($256 \times 256 \times 415$); third row: beetle ($416 \times 416 \times 247$); fourth row: salmon ($336 \times 173 \times 511$). First column: mean time to do a ray casting and compute the visibility histogram for 1 viewpoint; second column: time to accumulate $p(Z)$ for 1 viewpoint; third column: time to accumulate VOMI for 1 viewpoint; fourth column: total time to compute the VOMI map for 42 viewpoints.

4. Illustrative Visualization using Voxel Information

In this section we describe how interactive visualization of realistic and non-photorealistic styles based on VOMI can be obtained.

4.1. Ambient Occlusion

A first application of the VOMI is by interpreting it as an ambient occlusion (AO) term [14, 30]. AO is a measure of the visibility around a voxel, but while classical AO takes into account only local visibility, VOMI considers the whole volumetric data visibility around a voxel, from viewpoints surrounding the volume. VOMI measures how this visibility is distributed between viewpoints. Thus, the more uniform the visibility the less important is which viewpoint we consider, meaning that the voxel is less interesting or informative. In that case the VOMI value is low. On the other hand, the less uniform the visibility, the more important is which viewpoint we consider, meaning that the voxel is more interesting or informative. In that case the VOMI value is high.

To obtain the AO of each voxel, the VOMI of all voxels has been normalized between 0 and 1 and subtracted from 1, because low values of VOMI,

represented in the grey map by values near 1, correspond to non-occluded or visible (from many viewpoints) voxels, while high values of VOMI, represented in the grey map by values near 0, correspond to highly occluded voxels.

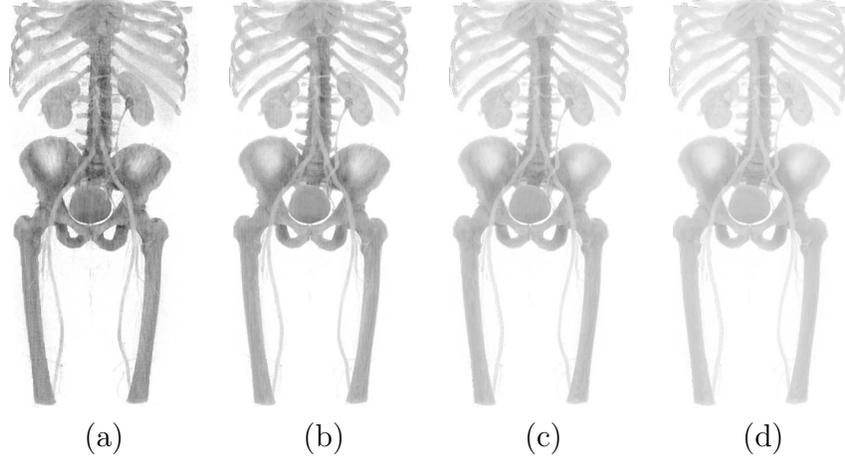


Figure 5: Ambient occlusion obtained from VOMI of the CT body model for 12, 20, 42, and 162 viewpoints.

In Figure 5 we show the results of computing AO using different number of viewpoints. Note that a sphere of 42 viewpoints provides enough quality for the AO maps, although 162 viewpoints is preferable for better quality.

In Figure 6, we compare the AO maps corresponding to the models of column (a) generated using different strategies. From column (b) to (e), respectively, we present the approaches by Landis [14], Stewart [21], Ruiz et al. [19], and finally our current proposed technique. Landis' approach is obtained by the application of the ambient occlusion technique [14] to volume rendering. Observe that, as expected, the all-or-nothing technique by Landis produces a too contrasted effect, due to too sharp transitions within the discrete set of occlusion values. On the other hand, Stewart's and Ruiz's methods generate smoother maps because of the continuous range of these values. These AO techniques take only in consideration the local occlusion of the voxel. The VOMI technique works in a different way since it considers the whole visibility, and thus occlusions, from the voxel to all viewpoints. This way, it integrates information of the whole volume with respect to the given voxel. This information will result in an AO map which will allow us to generate different volume rendering effects, as will be seen below. In

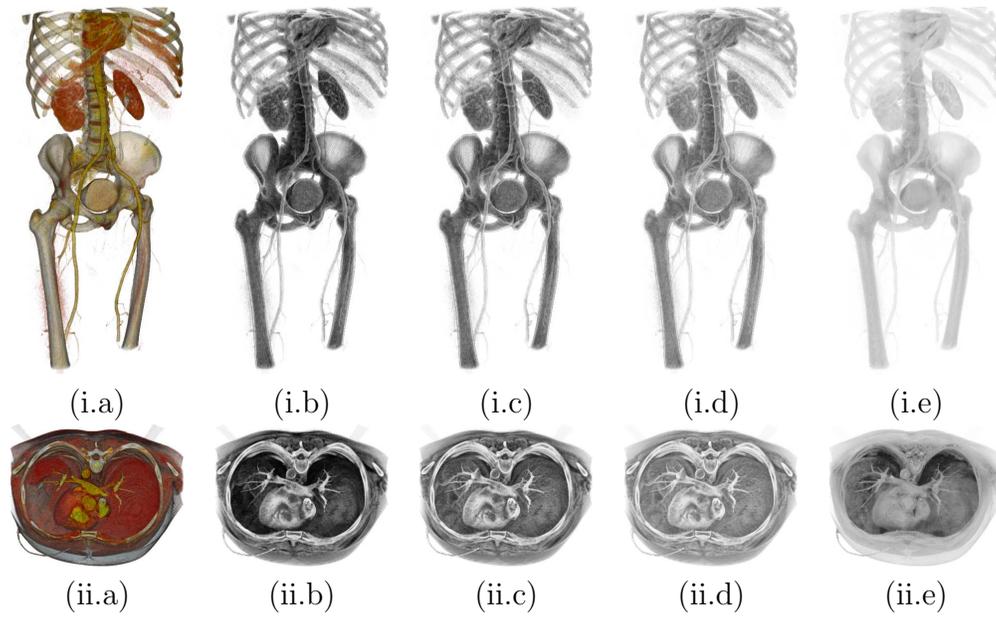


Figure 6: AO maps generated using (b) Landis', (c) Stewart's, (d) Ruiz's, and (e) VOMI approaches for the CT body model with (a) two different transfer functions.

the examples, we will consider Stewart’s method as representative of local occlusion methods.

The simplest effect is obtained considering the AO value as an ambient lighting AL term. In this case the color of a voxel z is obtained as

$$C(z) = AL(z) = k_i AO(z) C_{TF}(z), \quad (7)$$

where k_i is a constant factor that modulates the intensity of $AO(z)$ and $C_{TF}(z)$ is the pure color of the voxel as defined in the transfer function. In Figure 7 we illustrate the applications of the AO maps as an ambient lighting term, comparing the result of applying a local, classic ambient occlusion method [21] (Fig. 7(b)) with our approach (Fig. 7(c)). As we have commented above, VOMI takes into account the whole volume visibility, offering a more shaded result than local ambient occlusion. This is clearly visible in the skeleton. The overall information given by VOMI (Fig. 7(c)) produces better results than local ambient occlusion (Fig. 7(b)) with respect to the raw color information (Fig. 7(a)).

A different effect is obtained by adding the AO term to the local lighting equation, as in the global illumination case where ambient occlusion fakes indirect illumination [11]. Then, the final color of a voxel is obtained as

$$C(z) = (1 - w_{AO})((k_d N(z) \cdot L) C_{TF}(z) + k_s (N(z) \cdot H)^n) + w_{AO} AL(z) \quad (8)$$

where, k_d and k_s are the diffuse and specular lighting coefficients, $N(z)$ is the normal of the voxel, L is the light vector, H is the half-angle vector between L and the direction to the viewer, $AL(z)$ is the ambient lighting, and w_{AO} is the weight of the ambient occlusion in the final color.

In Figure 8 we illustrate the application of the AO maps as an additive term to the local lighting, comparing the result of applying Stewart’s method (Fig. 8(b)) with our approach (Fig. 8(c)). Here again, as in Figure 7, VOMI (Fig. 8(c)) produces better results with respect to the direct illumination image of column (Fig. 8(a)) than classic ambient occlusion (Fig. 8(b)). Observe that the overall features of the volume model are more distinguishable. Context information is better captured, giving an enhanced depth perception. This is clearly visible in the ribs: while in Figure 8(ii.b) all ribs appear in the same intensity due to the fact that local information is the same for all of them, in Figure 8(ii.c) the inner ribs are darker.

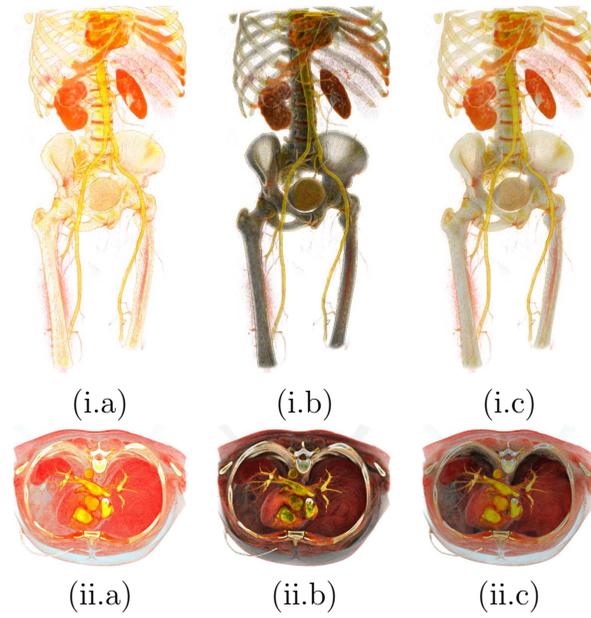


Figure 7: The original CT body model is shown (a) without illumination effects, and illuminated using AO computed with (b) Stewart's method and (c) VOMI, both applied as an ambient lighting term.

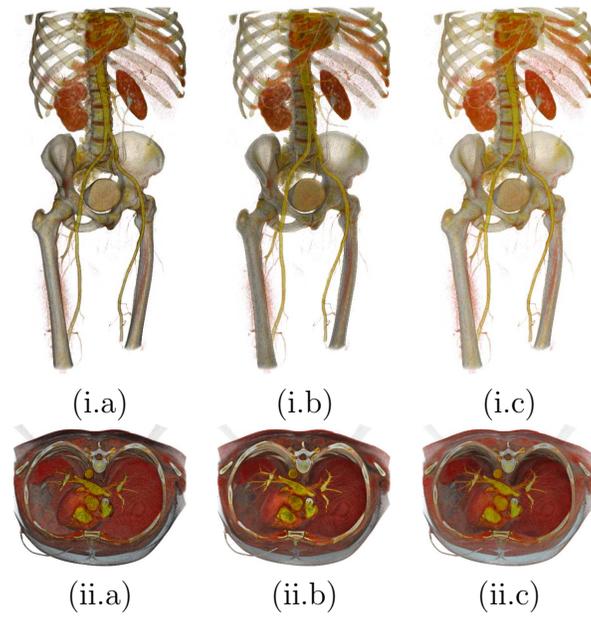


Figure 8: The original CT body model is shown with (a) local lighting, and with AO computed with (b) Stewart's method and (c) VOMI, both applied as an additive term.

4.2. Color Ambient Occlusion

Another effect that can be derived from the voxel information is the *color ambient occlusion*, which simulates the use of colored light sources at the different viewpoints. When all sources have the same color we recover the original AO. This can be obtained from the mutual information matrix and the color associated with each viewpoint. From Equation 5, we can consider that the mutual information matrix $MIM(Z; V)$ is constituted by the terms $I(z; v) = p(v|z) \log \frac{p(v|z)}{p(v)}$ (see Figure 2). Each term represents the shared information between voxel z and viewpoint v . The color ambient occlusion $CAO_\alpha(z; V)$ associated with the voxel z is defined by the scalar product of row z of matrix $MIM(Z; V)$ and the complement of a color vector $C(V)$ assigned to the set of viewpoints:

$$CAO_\alpha(z; V) = \sum_{v \in \mathcal{V}} I(z; v)(1 - C_\alpha(v)), \quad (9)$$

where α stands for each color channel, $C_\alpha(v)$ is the normalized vector for channel α , and $I(z; v)$ is a matrix element of $MIM(Z; V)$. After computing VOMI for each channel, the final color ambient occlusion is given by the combination of the color channel values. We can get a color vector by assigning certain colors to specific viewpoints and then interpolating the colors for the rest. In this way, a color is assigned to each viewpoint.

These relighting effects can be easily combined with other illustrative effects, such as color quantization, contours, and cool-and-warm. Figure 9 shows some of these effects applied to the CT body model considering different transfer functions. Figures 9(a-c) show, respectively, the AO map, the corresponding color ambient occlusion, and the AO map colored using a cool-and-warm technique [8]. Figures 9(d-f) show the use of color ambient occlusion combined with contours and color quantization.

4.3. Focus + Context

The last application of VOMI is as a *focus+context strategy*. In this case, VOMI is interpreted as a measure of importance and is used to modulate the opacity of a transfer function. The focus of interest is considered as the most informative part of the volume. Then, the opacity of the most informative voxels is increased (or preserved) while the opacity of the least informative is reduced. This opacity modulation effect is driven by the following equation:

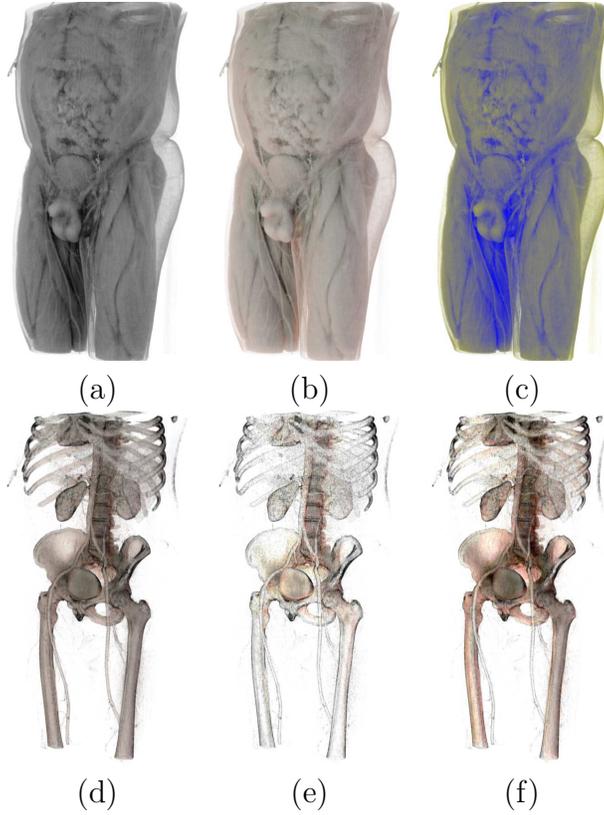


Figure 9: CT body model with different transfer functions and illustrative effects: (a) grayscale AO map, (b) color AO map, (c) cool-and-warm AO map, (d) color AO map with contours, (e) and (f) different color AO maps with contours and color quantization.

$$A'(z) = \begin{cases} A(z)k_l\overline{I(z;V)}, & \text{if } \overline{I(z;V)} < t_l \\ A(z)k_h\overline{I(z;V)}, & \text{if } \overline{I(z;V)} > t_h, \\ A(z), & \text{otherwise} \end{cases} \quad (10)$$

where $A(z)$ is the opacity of the voxel z before modulation, t_l and t_h are the low and high thresholds respectively, k_l and k_h are factors to regulate the effect of the modulation, $\overline{I(z;V)}$ is the normalized VOMI, and $A'(z)$ is the opacity of the voxel after modulation.

In the next figures we present the different effects that can be obtained varying the thresholds and factors, the transfer function, or the viewpoints considered for the visualization. In all the cases we have computed VOMI

using a set of 162 viewpoints. Figure 10 has been obtained modifying the thresholds and factors in order to emphasize a selected part of the model while preserving the context. Figures 10(a) and 10(d) correspond to the original CT body with $t_l = 0$ and $t_h = 1$ viewed from different viewpoints and with different transfer functions. In Figures 10(b) and 10(c) our target is the skeleton. As this is a highly occluded part, i.e., it has a high VOMI, to reach our objective, we have to decrease the opacity of less occluded parts, such as muscles, which have low VOMI. In Fig. 10(b) we obtain this effect by setting t_l to 0.5 and t_h , k_l and k_s to 1. In Fig. 10(c) we get a more extreme effect by changing k_l to 0.5, thus making less occluded parts even more transparent. In Fig. 10(e) we focus on the ribs, therefore, we want to make the muscles around them more transparent. We achieve this with $t_l = 0.3$, $t_h = 1$, $k_l = 0.1$, $k_h = 1$.

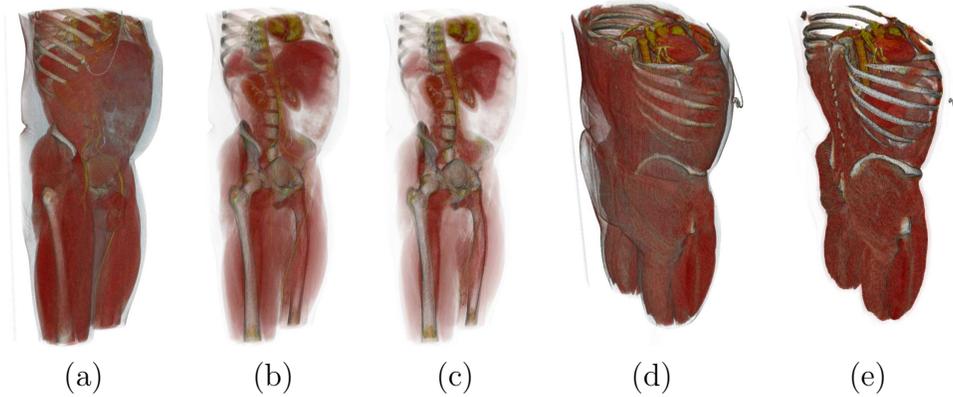


Figure 10: CT body model visualized with different transfer functions: (a) and (d) in their original states, (b) and (c) modulated from (a) by VOMI to emphasize the skeleton, (e) modulated from (d) by VOMI to emphasize the ribs.

VOMI depends on the transfer function used to visualize the model. In the previous example, we modulate the transfer function of the volume with the VOMI computed with that same transfer function. However, it is also possible to compute the VOMI with one transfer function and use it to modulate another one over the same model. To show this effect, we use the VOMI computed with the transfer function used in Figure 10(a), to modulate the one used in Figure 11(a). Figure 11(b) is obtained setting the parameters to the same values as in Figure 10(b). Since muscle in Fig. 10(a) is more transparent than in Fig. 11(a), modulating the opacity of the latter with the

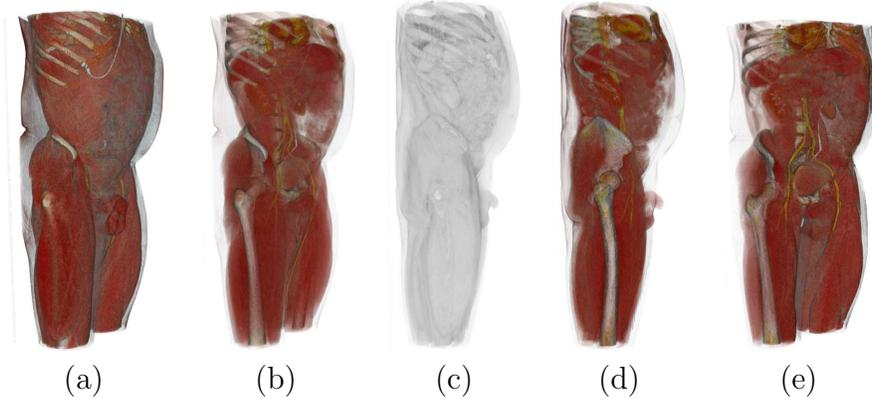


Figure 11: CT body model visualized with the transfer function: (a) of the original model, (b) modulated by VOMI computed with the transfer function used in Figure 10(a), (d) and (e) modulated by VOMI computed from the viewpoint in (c).

VOMI map of the former makes the muscle more transparent than it would be with its own VOMI map.

Until now, we have always computed VOMI from a set of viewpoints uniformly distributed over the surface of a sphere, but it is also possible to compute it from a subset consisting of one of the viewpoints and its neighbours. This can be useful to emphasize a part of the volume seen from that viewpoint while preserving the rest. For instance, in order to emphasize the right hip and the femur we compute the VOMI map considering the viewpoint at the right side of the model and its neighbours. Figure 11(c) shows the obtained VOMI map, and Fig. 11(d) and Fig. 11(e) show the modulation of Fig. 11(a) using this map. Note how our target is emphasized.

5. Viewpoint Selection using Voxel Information

In this section we introduce a new viewpoint selection measure based on voxel information. Then we analyze the behavior of viewpoint entropy and viewpoint mutual information compared with the new measure.

Once we have calculated the information associated to the voxels of a volume data set, this information can be “projected” on a viewpoint in order to obtain its informativeness. This method has been previously used to select the most informative views for polygonal models [6]. The information projection over a viewpoint v can be done weighting the VOMI of voxel z by the transition probability $p(v|z)$ and summing over all voxels. Thus, the

informativeness (INF) of a viewpoint v is defined by

$$INF(v) = \sum_{z \in \mathcal{Z}} p(v|z)I(z; V). \quad (11)$$

This represents the total voxel information seen by each viewpoint. Thus, high values of INF will correspond to viewpoints which see a lot of voxel information, i.e., highly occluded parts of the model. In many cases, these parts with high voxel information values show relevant details of the model. On the other hand, low values of INF correspond to low voxel information that is usually associated with smooth changes in visibility and less detail. This will be seen in the examples.

As we have seen in Section 2.2, different information theoretic viewpoint measures have been introduced to select the “best” views. But the “goodness” of a view can not be separated of the pursued objective. Thus, if our objective is to see the maximum number of voxels, viewpoint entropy (Equation 4) can be the most appropriate measure. This is due to the fact that the maximum entropy would be obtained when all the voxels were seen with the same projected visibility. Minimum entropy would be obtained when only one voxel was visible. On the other hand, viewpoint mutual information (Equation 3) can be used to detect the most representative views. That is, the views that are most similar to the virtual view of the object obtained from the projection of all viewpoints. The main difference between VE and VMI is that, while VE is very sensitive to the resolution of the volumetric data set, VMI is very robust to deal with any type of segmentation [26]. Due to the regular discretization of the volume data set in voxels, the behavior of VE and VMI is not significantly different in the experiments shown below.

Figure 12 shows the views which capture the maximum and minimum VE, VMI, and INF. For each model, the first row corresponds to the “best” views (maximum VE, minimum VMI, and maximum INF) and the second row to the “worst” views (minimum VE, maximum VMI, and minimum INF). Observe the different behaviour of the presented viewpoint measures. While maximum VE and minimum VMI present a relatively similar behaviour, showing respectively the maximum number of voxels in a uniform way and the most representative view, maximum INF is devoted to show the maximum number of highly occluded voxels.

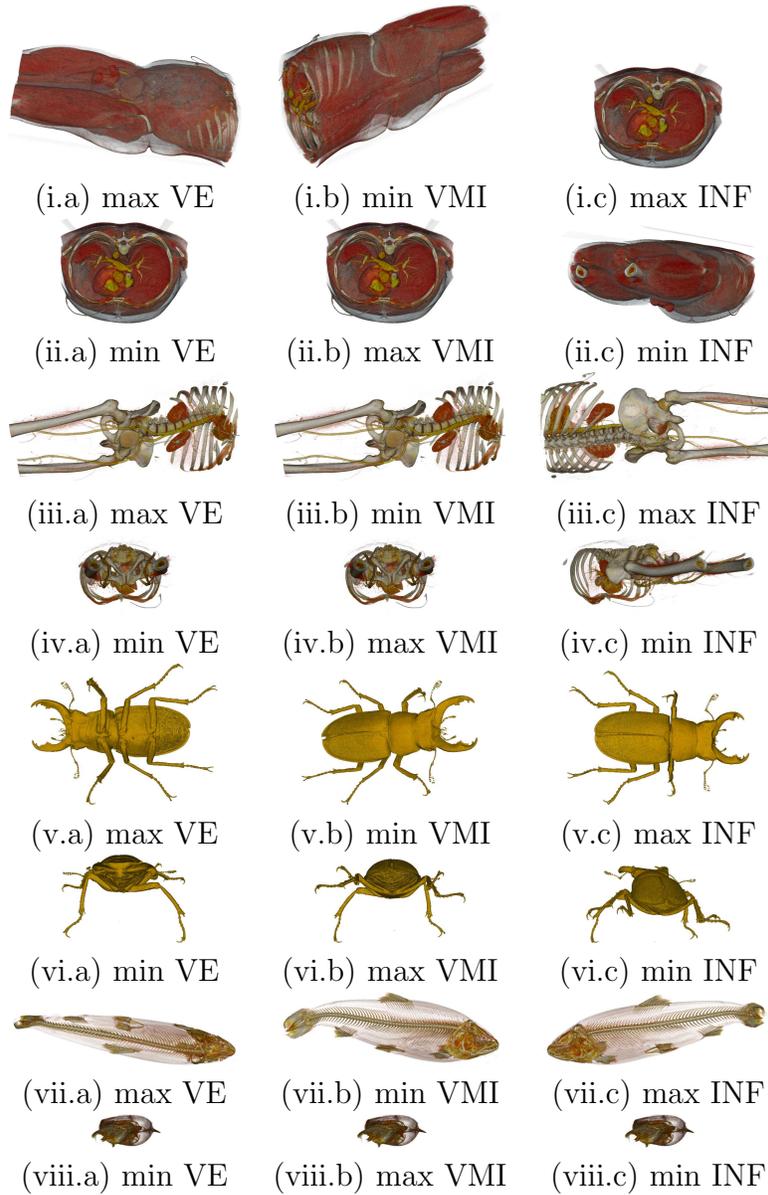


Figure 12: Selected viewpoints over a set of 162 with various models according to (a) viewpoint entropy, (b) viewpoint mutual information, and (c) informativeness.

6. Conclusions

In this paper, we have presented a viewpoint information channel to obtain illustrative renderings of volume data sets. The viewpoints and the voxels are the input and output distributions, respectively, for an information channel which is further defined by the visibility values of the voxels referred to each viewpoint. By reversing this channel we assign to each voxel an information value which can be shown to represent an ambient occlusion value. This quantity has been used for illustrative rendering purposes and combined with a lighting texture to further enhance the volume data. In addition, we have analyzed how the voxel information can be used to modulate transfer functions. Finally, we have used the voxel information to select the most informative viewpoints. As future work, we will discuss the relationship of the channel quantities with the transfer function and how the most representative or informative transfer functions could be obtained.

Acknowledgements

This work has been supported by TIN2007-68066-C04-01 and TIN2007-67982-C02 of the Ministry of Education and Science (Spanish Government) and by the 2009 SGR 643 from the Generalitat of Catalunya. The visible human data set is courtesy of the Visible Human Project, National Library of Medicine, US. The stag beetle data set has been provided by Georg Glaeser, Vienna University of Applied Arts, Austria and Johannes Kastner, Wels College of Engineering, Austria. The salmon CT data set is the courtesy of Prof. Ole Brix, Christian Michelsen Research, Norway.

- [1] Behrens, U., Ratering, R., 1998. Adding shadows to a texture-based volume renderer. In: *VVS '98: Proceedings of the 1998 IEEE symposium on Volume visualization*. ACM, New York, NY, USA, pp. 39–46.
- [2] Bordoloi, U. D., Shen, H.-W., 2005. Viewpoint evaluation for volume rendering. In: *IEEE Visualization 2005*. pp. 487–494.
- [3] Bruckner, S., Grimm, S., Kanitsar, A., Groller, M. E., 2006. Illustrative context-preserving exploration of volume data. *IEEE Transactions on Visualization and Computer Graphics* 12 (6), 1559–1569.
- [4] Chan, M.-Y., Qu, H., Wu, Y., Zhou, H., 2006. Viewpoint selection for angiographic volume. In: *ISVC (1)*. pp. 528–537.

- [5] Csébfalvi, B., Mroz, L., Hauser, H., König, A., Gröller, E., 2001. Fast visualization of object contours by non-photorealistic volume rendering. *Computer Graphics Forum* 20 (3), 452–460.
- [6] Feixas, M., Sbert, M., González, F., 2009. A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Trans. Appl. Percept.* 6 (1), 1–23.
- [7] Gonzalez, F., Sbert, M., Feixas, M., 2008. Viewpoint-based ambient occlusion. *IEEE Computer Graphics and Applications* 28 (2), 44–51.
- [8] Gooch, A., Gooch, B., Shirley, P., Cohen, E., 1998. A non-photorealistic lighting model for automatic technical illustration. In: *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, New York, NY, USA, pp. 447–452.
- [9] Hauser, H., Mroz, L., Bisch, G.-I., Gröller, E., 2001. Two-level volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 7 (3), 242–252.
- [10] Hernell, F., Ljung, P., Ynnerman, A., 2007. Efficient ambient and emissive tissue illumination using local occlusion in multiresolution volume rendering. In: *Eurographics/IEEE-VGTC Symposium on Volume Graphics 2007*. pp. 1–8.
- [11] Iones, A., Krupkin, A., Sbert, M., Zhukov, S., 2003. Fast, realistic lighting for video games. *IEEE Computer Graphics and Applications* 23 (3), 54–64.
- [12] Kindlmann, G., Whitaker, R., Tasdizen, T., Möller, T., 2003. Curvature-based transfer functions for direct volume rendering: Methods and applications. In: *IEEE Visualization*. pp. 513–520.
- [13] Kniss, J., Premoze, S., Hansen, C., Ebert, D., 2002. Interactive translucent volume rendering and procedural modeling. In: *VIS '02: Proceedings of the conference on Visualization '02*. IEEE Computer Society, Washington, DC, USA, pp. 109–116.
- [14] Landis, H., 2002. Renderman in production. In: *Course notes of ACM SIGGRAPH*.

- [15] Levoy, M., May 1988. Display of surfaces from volume data. *Computer Graphics and Applications*, IEEE 8 (3), 29–37.
- [16] Lu, A., Morris, C. J., Ebert, D. S., Rheingans, P., Hansen, C., 2002. Non-photorealistic volume rendering using stippling techniques. In: *VIS '02: Proceedings of the conference on Visualization '02*. IEEE Computer Society, pp. 211–218.
- [17] Rheingans, P., Ebert, D., 2001. Volume illustration: Nonphotorealistic rendering of volume models. *IEEE Transactions on Visualization and Computer Graphics* 7 (3), 253–264.
- [18] Ropinski, T., Meyer-Spradow, J., Diepenbrock, S., Mensmann, J., Hinrichs, K. H., 2008. Interactive volume rendering with dynamic ambient occlusion and color bleeding. *Computer Graphics Forum (Eurographics 2008)* 27 (2), 567–576.
- [19] Ruiz, M., Boada, I., Viola, I., Bruckner, S., Feixas, M., Sbert, M., Aug 2008. Obscure-based volume rendering framework. In: *Proceedings of IEEE/EG International Symposium on Volume and Point-Based Graphics*. pp. 113–120.
- [20] Sokolov, D., Plemenos, D., Tamine, K., 2006. Methods and data structures for virtual world exploration. *The Visual Computer* 22 (7), 506–516.
- [21] Stewart, A. J., 2003. Vicinity shading for enhanced perception of volumetric data. In: *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*. IEEE Computer Society, pp. 355–362.
- [22] Takahashi, S., Fujishiro, I., Takeshima, Y., Nishita, T., 2005. A feature-driven approach to locating optimal viewpoints for volume visualization. In: *IEEE Visualization 2005*. pp. 495–502.
- [23] Tarini, M., Cignoni, P., Montani, C., 2006. Ambient occlusion and edge cueing for enhancing real time molecular visualization. *IEEE Transactions on Visualization and Computer Graphics* 12 (5), 1237–1244.
- [24] Vázquez, P. P., Feixas, M., Sbert, M., Heidrich, W., 2001. Viewpoint selection using viewpoint entropy. In: *Proceedings of Vision, Modeling, and Visualization 2001*. pp. 273–280.

- [25] Vázquez, P.-P., Feixas, M., Sbert, M., Heidrich, W., 2003. Automatic view selection using viewpoint entropy and its applications to image-based modelling. *Computer Graphics Forum* 22 (4), 689–700.
- [26] Viola, I., Feixas, M., Sbert, M., Gröller, M. E., 2006. Importance-driven focus of attention. *IEEE Transactions on Visualization and Computer Graphics* 12 (5), 933–940.
- [27] Viola, I., Kanitsar, A., Groller, M. E., 2004. Importance-driven volume rendering. In: *VIS '04: Proceedings of the conference on Visualization '04*. IEEE Computer Society, pp. 139–146.
- [28] Wyman, C., Parker, S. G., Shirley, P., Hansen, C. D., 2006. Interactive display of isosurfaces with global illumination. *IEEE Trans. Vis. Comput. Graph.* 12 (2), 186–196.
- [29] Zhang, C., Crawfis, R., 2003. Shadows and soft shadows with participating media using splatting. *IEEE Transactions on Visualization and Computer Graphics* 9 (2), 139–149.
- [30] Zhukov, S., Iones, A., Kronin, G., 1998. An ambient light illumination model. In: *Rendering Techniques*. pp. 45–56.