

# Efficient opacity specification based on feature visibilities in direct volume rendering

Yunhai Wang<sup>1,2,3</sup>, Jian Zhang<sup>1†</sup>, Wei Chen<sup>4</sup>, Huai Zhang<sup>2</sup> and Xuebin Chi<sup>1</sup>

<sup>1</sup>Computer Network Information Center, Chinese Academy of Sciences, China

<sup>2</sup>Graduate University of Chinese Academy of Sciences, China

<sup>3</sup>Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

<sup>4</sup>State Key Lab of CAD&CG, Zhejiang University, China

---

## Abstract

*Due to 3D occlusion, the specification of proper opacities in direct volume rendering is a time-consuming and unintuitive process. The visibility histograms introduced by Correa and Ma reflect the effect of occlusion by measuring the influence of each sample in the histogram to the rendered image. However, the visibility is defined on individual samples, while volume exploration focuses on conveying the spatial relationships between features. Moreover, the high computational cost and large memory requirement limits its application in multi-dimensional transfer function design.*

*In this paper, we extend visibility histograms to feature visibility, which measures the contribution of each feature in the rendered image. Compared to visibility histograms, it has two distinctive advantages for opacity specification. First, the user can directly specify the visibilities for features and the opacities are automatically generated using an optimization algorithm. Second, its calculation requires only one rendering pass with no additional memory requirement. This feature visibility based opacity specification is fast and compatible with all types of transfer function design. Furthermore, we introduce a two-step volume exploration scheme, in which an automatic optimization is first performed to provide a clear illustration of the spatial relationship and then the user adjusts the visibilities directly to achieve the desired feature enhancement. The effectiveness of this scheme is demonstrated by experimental results on several volumetric datasets.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

---

## 1. Introduction

Direct volume rendering has proven to be an effective means of discovering meaningful features in volumes. By specifying appropriate opacities for extracted features, the exterior and interior features can be simultaneously revealed in a semi-transparent manner. However, in practice it is rather difficult and time-demanding to specify appropriate opacities, due to 3D occlusion between features. The main reason is that interaction in the transfer function domain is mainly guided by careful observation of changes in the rendered image [KKH01]. Decisions based on this kind of adjustment are subjective and view-dependent. More-

over, there is no quantitative metric to measure the influence of each feature to the rendered image. While most of the previous research focused on how to extract features [RBS05, MWCE09, WCS\*10], very little attention has been paid to quantitative analysis of the visibility of classified features in the rendered image.

Recently, visibility histograms [CM11] have been introduced to guide transfer function design, which represent the contribution of each sample in the histogram to the rendered image. This metric is computed by measuring the visibility of each sample in each view-oriented slice and then adding the visibility to the corresponding bin in the histogram. As the dimension of the histogram increases, its computational performance decreases drastically. Meanwhile, the histogram space is often classified into several continuous

---

† Corresponding author, the first two authors contributed equally.

regions, of which each region represents a feature [KKH01]. The user is more concerned with the influence of each feature to the rendered image. Therefore, calculating the visibility for each sample seems to be unnecessary, especially for multi-dimensional transfer function design.

In this paper we propose a new metric, *feature visibility*, that reveals the visibility of each classified feature. Computing the visibility of a feature is simple and fast. For any number of extracted features, their corresponding feature visibilities can be obtained with one additional rendering pass. By representing feature visibilities with a bar chart, the influence of each feature on the rendered image can be intuitively perceived. As shown in Fig. 1, when the other features are occluded by the skin, only the skin's visibility is larger. In contrast, the opacity of the skin is lower than those of the teeth and the vessel. The manipulation of the opacity transfer function becomes more intuitive when the feature visibility is incorporated into the interaction.

Even when guided by this metric, obtaining satisfactory classification results for some complicated datasets still requires many interactions. To further reduce user workload, we introduce a new interaction scheme that allows the user to set the visibility for each feature instead of directly modulating the opacity. An optimization procedure is then invoked to generate the opacity of each feature automatically. This visibility-based opacity specification method is effective. In addition, we introduce *summarized visibility* to make the optimization result work well for other viewpoints.

Since the optimization procedure is independent of the definition of transfer functions, the visibility based opacity specification can be applied to all types of current transfer function design modes, e.g. the gradient magnitude modulated transfer function. As a result, a visibility-driven opacity optimization can always be applied to an existing transfer function to provide a clear spatial depiction. The user can then continue the iteration loop between the transfer function design and the visibility based opacity optimization. We call this process *visibility-driven volume exploration*, which equips volume exploration with a perceivable and objective guidance and eases the user interaction.

The contributions of this paper are as follows:

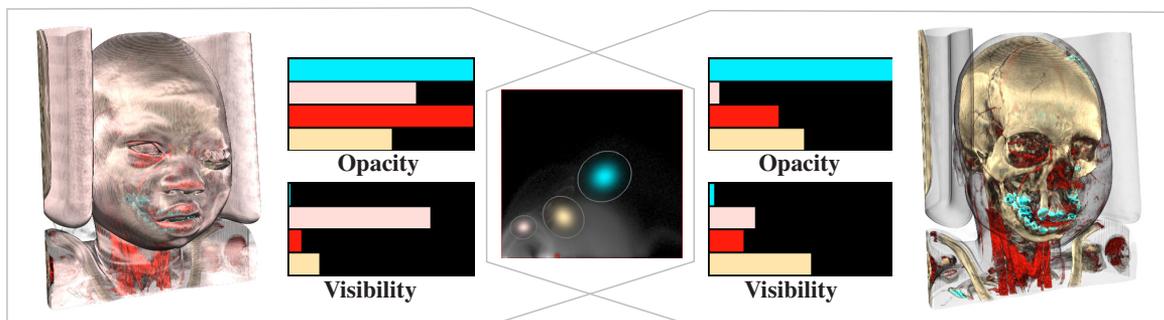
- A new metric, *feature visibility*, that measures the contribution of each feature to the rendered image, and an efficient algorithm to calculate it (section 3).
- A feature visibility based opacity specification approach that allows the user to directly specify the desired visibility for the feature of interest (section 4).
- A two-step feature visibility driven volume exploration scheme that improves the effectiveness of the volume exploration (section 5).

## 2. Related Work

The use of multi-dimensional transfer functions has been shown to be an effective way to accurately classify materials for both scalar and multivariate data. By analyzing the 2D histogram of data values and gradient magnitudes, Kindlmann and Durkin [KD98] demonstrate that the arc structures of the 2D histogram correspond to the material boundaries. Further exploiting this behavior, Kniss et al. [KKH01] develop a set of direct manipulation widgets to define multidimensional transfer functions [Lev88]. To better characterize local structures, curvature has also been incorporated into the transfer function domain [KWTM03]. Despite the effectiveness of these approaches, they inherently lose the spatial information. To incorporate spatial relations into the transfer function design, Roettger et al. [RBS05] consider spatial information in the process of clustering 2D histograms. Recently, Correa and Ma introduce several spatial-aware properties to classify complex datasets in terms of spatial relationships among features, i.e., size [CM08], occlusion [CM09], and visibility [CM11]. For illuminating the global structure of the volume, Takeshima et al. [TTFN05] and Weber et al. [WDC\*07] introduce a set of topological attributes to design multi-dimensional transfer functions. However, as the dimension increases, finding an appropriate transfer function becomes more complicated.

Through analysis of the volume data, many methods have been proposed to simplify the creation of multi-dimensional transfer functions. Tzeng and Ma [TM04] use the ISODATA algorithm to cluster the multi-dimensional histograms. To effectively decompose the 2D histogram, Maciejewski et al. [MWCE09] apply a non-parametric density estimation technique to group voxels of similar features within the 2D histogram and Wang et al. [WCS\*10] analyze the histogram space with the Gaussian mixture model, which maximizes the likelihood of feature separation. Selver and Güzelis [SG09] introduce a self-generating hierarchical radial basis function network to initialize transfer functions for abdominal data. In order to provide intuitive user interfaces for the transfer function design, Tzeng et al. [TLM05] introduce a painting interface that derives high-dimensional transfer functions with a learning classifier and Rezk-Salama et al. [SKK06] present high-level user interfaces which allow the user to design transfer functions with semantics. Similar with them, our feature visibility is a high-level quantity, measuring the influence of each feature to the rendered images.

In all of the above, the quality of the transfer function evaluated by observing its resultant image is subjective. To make transfer function design more effective, several metrics have been proposed. Based on the perception principles, Chan et al [CWM\*09] introduce several image quality measures, such as *transmittance anchoring principle* (TAP), to enhance the perceived quality of semitransparent structures. The visibility histograms introduced by Correa and Ma [CM11] act as an immediate feedback to the user re-



**Figure 1:** Bar charts representing the visibilities of classified features in the CT facial deformity data ( $512 \times 512 \times 361$ ). After identifying four features shown as the ellipses in the 2D density vs. gradient magnitude histogram (middle), the user first sets a large opacity for the peach ellipse to observe the skin layer (left). We can see that the other features have small visibilities. After lowering the opacities of skin and vessel, the features visibilities are updated accordingly, where the visibility of the bones increases (right).

garding the contribution of each sample in the final resulting image. With this measure, the process of transfer function design becomes more intuitive and informative. However, computing a visibility histogram in the domain of multi-dimensional transfer functions will become memory intensive and costly to compute. In contrast, the proposed feature visibility is feature-oriented and its evaluation only requires one additional rendering pass for all features rather than the expensive visibility gathering from view-oriented slices. Furthermore, we provide a new method for opacity specification, where the user directly specifies the desired visibility for features of interest and the appropriate opacities are automatically obtained. This is similar to the perception-based transparency equalization [CWM\*09], which automatically corrects the rendering parameters based on the psychological principles. While this work focuses on the equalization of segmented layered structures, our scheme does not require any pre-processing. Moreover, the proposed feature visibility not only works as a guideline for transfer function design but can also be directly specified for the features of interest.

### 3. Feature Visibility

Feature visibility is inspired by the visibility histograms [CM11], which measure the accumulated opacity of each sample  $p$  to the eye position  $E$ :

$$VH(x) = A(x) \int_{p \in \Omega} \delta(p, x) e^{\int_p^E \tau(t) dt} dp \quad (1)$$

where  $\Omega$  is the volume of data,  $\tau(t)$  is the attenuation coefficient of a sample,  $A(x)$  is the opacity of the sample value  $x$ , and  $\delta(p, x)$  is a membership function:

$$\delta(p, x) = \begin{cases} 1 & \text{if } V(p) = x \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

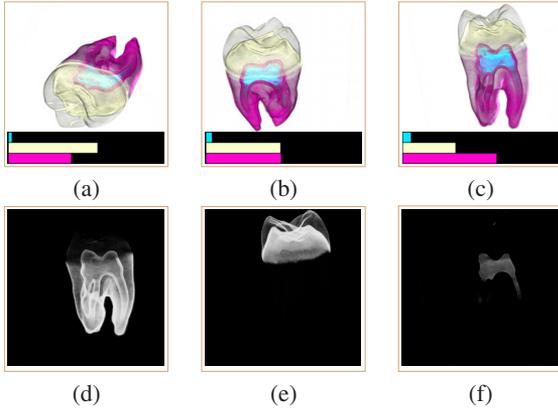
where  $V$  maps the voxel  $p$  to the transfer function domain. Accordingly, the computation complexity of visibility histogram is proportional to the number of histogram bins which increases exponentially with the dimension of transfer function domain. It will be memory intensive and time-consuming for multi-dimensional transfer functions [CM11]. On the other hand, calculating the visibility for each sample seems to be unnecessary. The volume classification is usually achieved by first specifying the regions of the transfer function that correspond to potential features and then setting appropriate opacities to these features, rather than individual samples. Thus the user is concerned with the influence of each feature to the rendered image and the proposed feature visibility is a high-level quantity.

For multi-dimensional transfer functions, features are represented as subregions in the feature space. Usually, these regions are brushed by classification widgets, which can be overlapping or non-overlapping. For the  $i$ th feature, the opacity of its corresponding voxels is defined as [KPI\*03]:

$$\alpha_i(x) = \alpha_{\max i} \rho_i(x) \quad (3)$$

where  $x$  is a sampling point in the feature space,  $\alpha_{\max i}$  is the specified opacity which is the maximum within the associated classification widget.  $\rho(x)$  is a template function, such as a constant function, a linear function, or a Gaussian function. Its parameters can be interactively specified or determined with well designed feature extraction algorithms [WCS\*10, MWCE09]. After the feature extraction, the user has to specify proper opacity to each feature to explore features of interest. For convenience, in this paper we refer to the specification of  $\alpha_{\max i}$  as the opacity specification.

As mentioned above, the user concerns the influence of each feature on the rendered image. Hence, we first define



**Figure 2:** (a,b,c) The visibility bar charts from different viewpoints. Notice the increase and decrease of features visibilities, depends on the occlusion relation among them. (d,e,f) Illustration of the visibility images of three features in the rendered image (c). (d) The visibility image of dentine; (e) The visibility image of enamel; (f) The visibility image of pulp.

the absolute feature visibility for each feature:

$$\tilde{V}_i = \int_{s \in S} VI_i(s) ds \quad (4)$$

$$VI_i(s) = \int_0^{L_s} \alpha_i(V(p(\lambda_s))) e^{\int_0^{\lambda_s} \tau(t) dt} d\lambda_s \quad (5)$$

where  $S$  is the output image,  $s$  is the sample in  $S$  from which a ray is cast,  $p(\lambda_s)$  is a parametric representation of the ray, and  $L_s$  is the ray length.  $VI_i$  is the visibility image of the  $i$ th feature where  $VI_i(s)$  is the visibility of each pixel. Since the user concerns the relative influence, our proposed feature visibility is a relative variable:

$$V_i = \frac{\tilde{V}_i}{\sum_{j=1}^n \tilde{V}_j} \quad (6)$$

For  $n$  extracted features, the accumulated visibility  $VI_i(s)$  is discretized by means of the front-to-back composition at discrete intervals:

$$\alpha_j = \alpha_{j-1} + \sum_{i=1}^n \alpha_i(x)(1 - \alpha_{j-1}) \quad (7)$$

$$VI_{i,j} = VI_{i,j-1} + \alpha_i(x)(1 - \alpha_{j-1}) \quad (8)$$

where  $\sum_{i=1}^n \alpha_i(x)$  is the resulting opacity contributions from  $n$  classification widgets, and  $\alpha_{j-1}$  is the previously accumulated value for opacity. As such, the visibility images of  $n$  extracted features and the rendered image can be obtained simultaneously. Fig. 2(d,e,f) show the visibility images of three features in Fig. 2(c). To graphically represent the occlusion relation between different features, we associate feature visibilities to a bar chart. Fig. 1 shows an example in which the teeth are partially occluded by the skin. Lowering

the opacity of the skin leads to small increases in the teeth' visibility, while the visibilities of the vessel and bone are increased greatly.

In principle, feature visibility can be obtained as brushing the visibility histogram into several regions and summing the visibilities of samples in each region. If these regions are the same size and cover the entire histogram range, the bar chart of feature visibilities is a visibility histogram with small number of bins. However, the calculation of feature visibilities does not require the high dimensional visibility histogram. Instead, an efficient volume rendering algorithm can be employed to generate 2D visibility images using GPU and then simply transfer the image from GPU to CPU to calculate the feature visibilities.

Obviously, the proposed feature visibility is opacity-dependent and view-dependent. As shown in Fig. 1, the feature visibility depends on opacities of all features. Meanwhile, it varies with viewpoints, as shown in Fig. 2(a,b,c). From the viewpoint in Fig. 2(a) to the viewpoint in Fig. 2(c), the visibility of the dentine gradually increases while the enamel's visibility decreases.

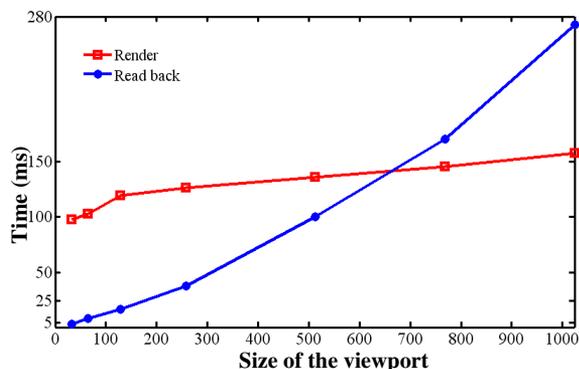
### 3.1. GPU-assisted Computation

Using the multiple render targets (MRT) extension, we can handle 8 color buffers simultaneously. Thus, in total we can render 32 visibility images with one additional rendering pass by packing 32 visibility images into 8 floating point RGBA textures. If the number of features exceeds 32, the visibility images can be obtain by multipass rendering. After obtaining these visibility images, they are read back to the CPU for accumulation.

With all other factors such as viewpoint, opacity and lighting are fixed, the viewport size is the unique factor that affects the performance of visibility evaluation. The procedure of visibility evaluation consists of three steps: rendering visibility images, reading back from GPU to CPU, integration on the CPU. Fig. 3 shows the time consumed on rendering and reading back visibility images for the CT facial deformity dataset with different viewport sizes. Due to the large size of this dataset, a ray casting through the data takes about 100ms on the NVIDIA GeForce GTX 260 video card. Furthermore, the parallel efficiency of the GPU shader also has slight influence on the rendering performance, shown in red in Fig. 3. The main impact of viewport size on performance is the time consumed in reading back of the visibility images, shown in blue in Fig. 3. The integration on the CPU can be finished in less than 5ms even for  $1024 \times 1024$  visibility images.

Nonetheless, the resulting visibility is insensitive to the viewport. This is because feature visibility is a relative variable, even in a small viewport the ratio of sampled voxels of each features is nearly the same as a large viewport. Thus,

we use a  $32 \times 32$  viewport to calculate the visibility. However, smaller viewport may introduce some aliasing effects for the features with high spatial frequency. Fortunately, we have not encountered this problem in our experiments.



**Figure 3:** The relation between the viewport size and the time consumed in rendering and reading back visibility images for the CT facial deformity dataset ( $512 \times 512 \times 361$ ).

#### 4. Feature visibility based Opacity Specification

In volume visualization, the user explores features of interest by assigning large opacities to them then continues manipulating other features' opacities. However, direct opacity specification is a challenging task since small adjustments to the opacity may result in dramatic changes in the result and usually adjustments on one feature affect appearances of other features. Fig. 1 shows an example, where the bone and teeth can be clearly revealed only when the opacity of the skin is lowered to a very small value.

Because feature visibility is a high-level quantity, we propose a new opacity specification paradigm, that is, the user specifies the desired visibility and the opacity is generated to produce an image with the desired visibility. This is a major difference from the semiautomatic transfer functions design with visibility histograms [CM11], in which the user can only set the initial transfer function not the visibility.

Since feature visibility is a normalized variable, the sum of all feature visibilities is always kept 1. To achieve focus+context effect, we allow the user to fix some feature visibilities and adjust the others. After setting the desired visibilities  $\bar{V}_i$  for features, the specification of the opacity set  $\Theta = \{\alpha_{\max 1}, \alpha_{\max 2}, \dots, \alpha_{\max n}\}$  can be cast into an optimization problem:

$$\operatorname{argmin}_{\Theta} \sum_{i=1}^n (V_i - \bar{V}_i)^2 \quad (9)$$

where  $V_i$  is the visibility of the  $i$ th feature defined in Equation 6, and  $\alpha_i$  satisfies the following constraints:

$$0 \leq \alpha_i \leq 1 \quad (10)$$

To minimize this energy function, we use an active set algorithm (ASA) [Pol69], which is an iterative method that distinguishes the set of active constraints from the inactive ones. ASA is a general optimization algorithm especially suited for solving box constrained optimization problems. Moreover, it can achieve global convergence with some loose assumptions. In other words, the iteration is guaranteed to converge to a local minimum no matter what the initial guess is. This property is very useful in the opacity optimization procedure where the user usually does not have a clear idea about what initial opacities are close enough to ensure convergence. The traditional conjugate gradient or steepest descent methods do not have this property. Realization of this algorithm requires the calculation of partial derivatives and an initial solution. Because it is not easy to find an analytical expression for the gradient in Equation 9, we calculate partial derivatives of the objective function using a second-order central difference scheme.

Meanwhile, we set the initial solution with the specified visibilities, because visibility is often positively correlated with opacity. With this configuration, the example in the optimization of Fig. 4(b) to Fig. 4(c) takes 38 steps to converge. In all the other experiments the optimization converges in less than 20 steps.

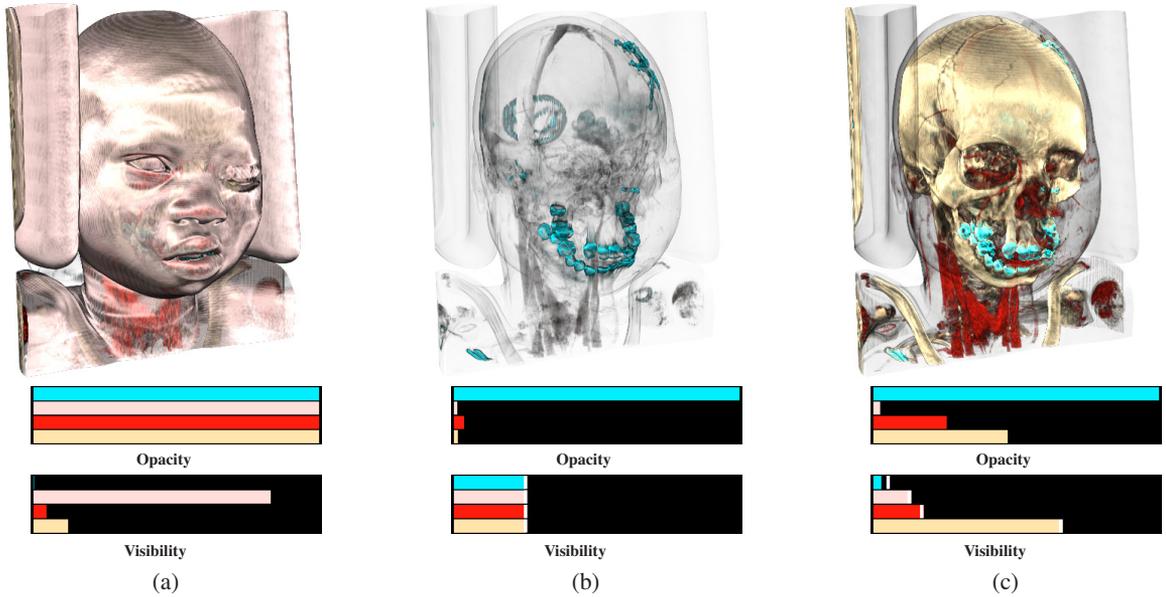
Because feature visibility is integrated on the spatial extents of its corresponding feature, a small-sized feature usually has a small visibility. Hence, setting the same specified visibility for each feature favors small-sized features. As shown in Fig. 4(b), large-sized features' opacities are greatly decreased to make the teeth region have the same visibility as other features. In order to make all features clear, the user can set a smaller specified visibility for the teeth (Fig. 4(c)).

#### 4.1. Multi-View Optimization

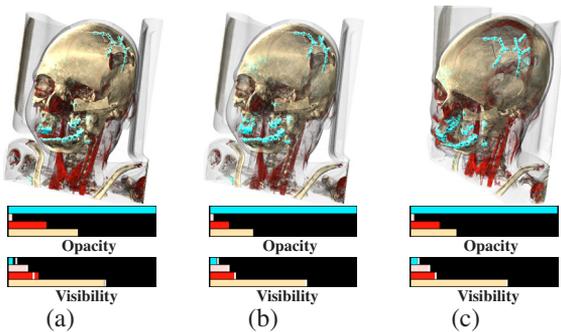
Because the visibility is view-dependent, the visibility based opacity specification cannot guarantee that all features have the same visibilities in other viewpoints. It is desirable that the obtained opacities can lead to the similar visibility under different viewpoints.

To achieve the visibility invariant multiple-view visualization, one of the most straightforward methods is to solve Equation 9 at each viewpoint. Fig. 5 shows the separate optimization results for two different viewpoints. Comparing Fig. 5(b) with Fig. 5(a), we see that the re-optimization result nearly achieves the desired visibilities and clearly shows the damaged regions near the teeth. However, the result in Fig. 5(b) is incoherent with the result in Fig. 4(c), which imposes some overhead on the user in the data exploration. This point is also demonstrated by the comparison of the optimization results in Fig. 5(b) and Fig. 5(c).

Allowing for small variations in the visibilities, we introduce the *summarized visibility* to accomplish this optimization.



**Figure 4:** The results of visibility based opacity specification for the CT facial deformity dataset. (a) The initial rendered image and its corresponding feature visibilities. (b) The obtained opacities: 0.999, 0.011, 0.035, and 0.013 for each feature and its resulting image and visibilities produced by specifying the same visibilities 0.25 for four features (white lines). (d) The resulting image, opacities, and feature visibilities produced by specifying the visibilities 0.132, 0.178, 0.027 and 0.663 for four features (white lines), respectively.



**Figure 5:** View-dependent nature of the visibility-based opacity specification for the CT facial deformity dataset. (a) The result produced by changing the viewpoint in Fig. 4(c), where the differences between the evaluated visibilities and the specified visibilities (white lines) are larger. (b) The optimization yields a better depiction of the region near the teeth, by decreasing the opacity of the vessel and bone. (c) The optimization result of another viewpoint where the damaged regions in the head are clearly shown.

tion, defined as:

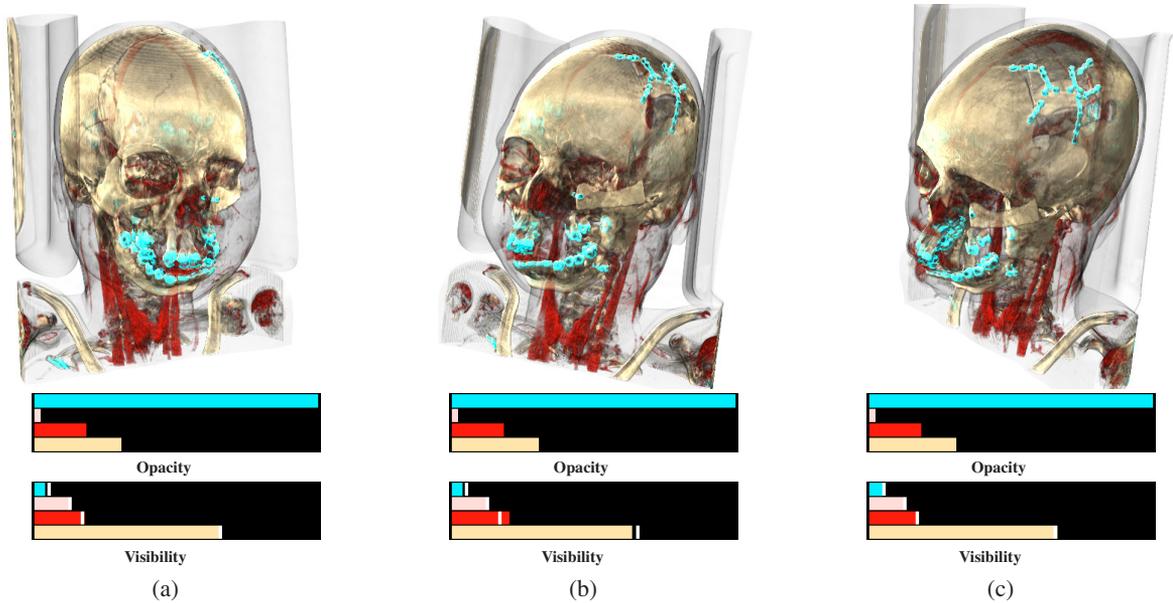
$$V_i = \mathcal{G}(V_i^1, V_i^2, \dots, V_i^m) \quad (11)$$

where  $m$  is the number of viewpoints,  $V_i^j$  is the visibility of the  $i$ th feature under the  $j$ th viewpoint, and  $\mathcal{G}$  is an operator, like maximum, minimum or mean. For convenience, we use the mean operator. To clearly reveal the spatial relationship among extracted features,  $m$  viewpoints can be selected by information-theoretic metrics [BS05, VFSG06] or manual specification. With this summarized visibility, the multi-view optimization can be performed as a single-view optimization. Fig. 6 shows the result of the optimization result of a summarized visibility for three specified directions. Although the optimized ones and the desired visibilities are not identical, the resulting images appropriately depict the damaged regions near the head and teeth.

## 4.2. Gradient Modulated Transfer Function

The definition of feature visibility is independent of transfer functions, and thus the optimization framework supports various transfer function design modes. Gradient magnitude opacity-modulation is an efficient way to highlight boundaries between materials [ER00]. It defines the opacity of each voxel as  $\alpha|\nabla N(P)|^k$ , where  $|\nabla N(P)|$  is the gradient at the sample point  $P$  and  $k$  controls how the opacity is scaled by the gradient magnitude. For convenience, we name the 2D transfer function without gradient modulation *standard* 2D transfer function.

To adaptively enhance features, we allow the user to mod-



**Figure 6:** The obtained opacities and their resulting images produced by using the summarized visibility for the CT facial deformity dataset. Although the evaluated visibilities from three viewpoints are not completely equal to the specified ones, the resulting images not only depict the damaged regions but also preserve the coherence.

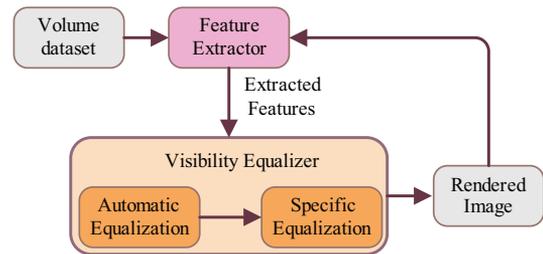
ulate different features. The opacity of the samples of the  $i$ th feature is defined as

$$\alpha_i(x) = \alpha_{\max_i} \rho_i(x) |\nabla N(P)|^{k_i} \quad (12)$$

where  $k_i$  is constant for the  $i$ th feature. For the traditional transfer function design, these kinds of transfer functions require the user to adjust the extra dimension for each feature. Fortunately, in our approach, the user interaction remains simple by specifying the desired visibility for each feature. Fig. 9 gives a comparison of the optimization results produced by the *standard* 2D transfer function and the gradient modulated 2D transfer function. Obviously, the gradient modulated 2D transfer function reveals the structure near the core in the supernova more clearly.

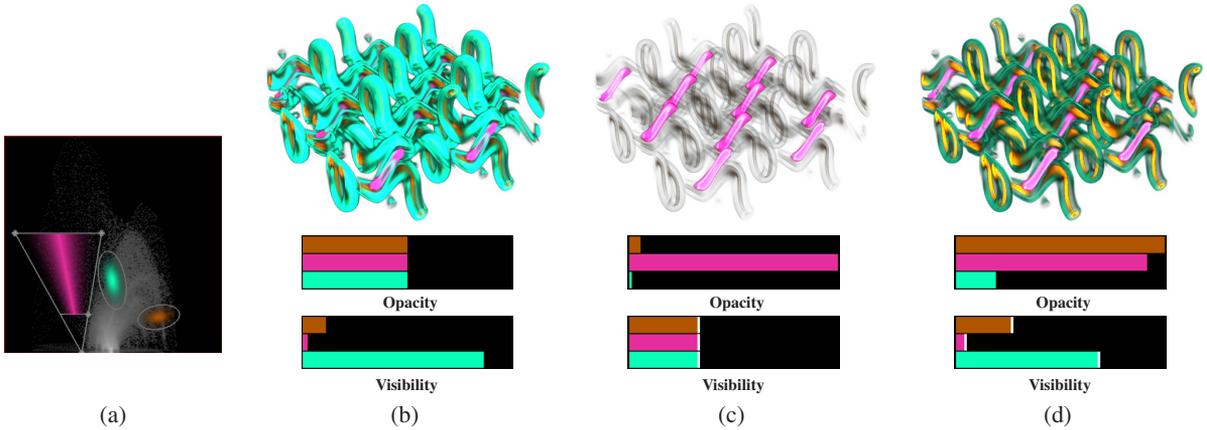
### 5. Feature Visibility driven Volume Exploration

Using an arbitrary classification approach such as the gradient analysis [KD98] or statistical analysis [MWCE09, WCS\*10], several meaningful features can be efficiently extracted from an input volume. The user can then explore these features by specifying appropriate opacities in two manners: visibility assisted and visibility based opacity specification. In the former, the opacity specification is guided by the visibility, which immediately indicates to the user the influence of each feature in the rendered image. Fig. 1 shows an example where the user can intuitively perceive the visibility change of each feature after lowering the opacity of the skin. However, finding appropriate opacities is still a trial-and-error process.



**Figure 7:** The pipeline of our volume exploration scheme.

We focus on the latter: visibility based opacity specification. Here, the features are already extracted by some algorithms [MWCE09, WCS\*10] or specified by other users, the user just wants to enhance features of interest. Fig. 7 shows the pipeline of this scheme, which consists of two steps: an automatic optimization and iterative specific optimization. The first is performed automatically by setting the specific feature visibility to  $1/n$  for each feature. Although its produced result may emphasize small-sized features, almost all features can be clearly shown. After gaining an overview from this result, the user iteratively adjusts the visibilities and optimizes the opacities. Based on the classified features, this exploration scheme help the user quickly find proper opacities to clearly reveal the spatial relationship among extracted features. If the user finds the classification result needs improvement after the optimization, he/she can



**Figure 8:** The optimization result for the Horseshoe Vortex dataset classified with various widgets. (a) The transfer function consists of three classification widgets. (b) The rendering result with the initial opacity specification. (c) The initial optimization result. (d) The optimization result produced by specifying the visibilities to be 0.269, 0.045, and 0.686 for the three features, respectively.

adjust the classification while the feature visibility can quantitatively measure the influence of each feature to the rendered image. Normally, the user can accomplish the volume exploration in several iterations.

### 5.1. Medical data

To demonstrate the effectiveness of our approach in surgery repair, we conduct an experiment on the CT facial deformity dataset. Initially, the user sets the opacities using a set of pre-defined values for four extracted features. Fig. 4(a) shows its resulting image and visibilities. We notice that the bone and a portion of the teeth are occluded by the skin. This point is also demonstrated by the visibility chart, where the skin's visibility is much larger than others. After applying the automatic optimization, Fig. 4(b) shows the overview of these four features. Compared to Fig. 4(a), these four features are visible but not clearly shown. The reason is that the size of the feature colored in cyan is very small, while its corresponding default visibility is fairly large. The user then decreases its specified visibility and increases the bone's visibility, yielding a more pleasing result (Fig. 4(c)) where the spatial relations of four features are clearly shown. However, we still can see something abnormal in the teeth region but not clearly in the current viewpoint. Some parts colored in cyan near the head cannot be seen.

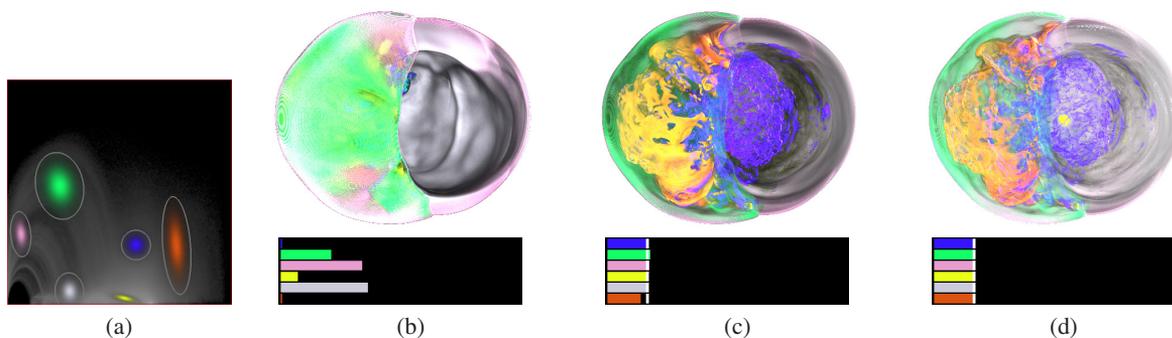
To inspect the damaged regions more clearly, the user first selects three other viewpoints and then obtains the optimization results for each viewpoint as shown in Fig. 5. Then, the user employs the summarized visibility to perform optimization to achieve coherent feature enhancement. Fig. 6 shows the coherent results where these damaged regions are still clearly shown. From this experiment, we can see that the summarized visibility not only helps the user obtain the de-

sired result but also preserves the coherence between neighboring viewpoints.

### 5.2. Simulated Data

To demonstrate the effectiveness of our approach, an experiment was conducted on the Horseshoe Vortex dataset ( $256 \times 256 \times 401$ ). This data consists of two layers of vortex tubes and several intersections of vortex tubes. Because the surface of the intersections is very thin and the inner layer is composed of vortices, we use the inverse triangular widget and elliptical widgets to capture them, respectively, as shown in Fig. 8(a). Fig. 8(b) shows the initial classification results after setting the opacity to 0.5 for each feature. We notice that interior vortex tubes and vortex intersections are occluded by the exterior vortex tubes. Fig. 8(c) shows the result of the automatic optimization. The resulting image clearly shows the inner vortex tubes and vortex intersections, while providing the outer shape of the features as a context. Fig. 8(d) shows the effect of shifting the visibility to the outer layer. The opacities are adapted interactively to reflect the change in the visibilities. From this experiment, we can see that the visibility based opacity assignment can work well with different classification widgets.

Although the spatial relationship between vortex tubes is very complicated, no strong occluder is present in the data. For some datasets produced by hydrodynamic simulations, some features of interest are occluded by the outer layer. An example is shown in Fig. 9(a), where the supernova dataset has been classified into six features: three outer layers, two turbulent structures, and an innermost core. Clearly revealing these features can help the scientist to understand the interaction between them. First, we use the standard 2D transfer function (the entropy and the gradient magnitude of



**Figure 9:** The automatic optimization results based on the standard 2D transfer function and the gradient modulated 2D transfer function for the supernova dataset. (a) The transfer function with six features. (b) The initial result produced by setting each feature's opacity to 1.0. (c) Setting uniform visibility 1/6 to each feature, the result can show the inner turbulent structures but not the core. (d) With the gradient modulated 2D transfer function, the optimization result clearly reveals the core and its surrounding turbulent structures.

entropy) to do optimization. Simply setting each feature's opacity to 1 (Fig. 9(b)), we only can see the three outer layers. After the automatic optimization, the resulting opacity specification provides better visibility to the inner structures (Fig. 9(c)). However, the part near the core is still not clearly revealed. When we use the gradient modulated 2D transfer function, the automatical optimization result (Fig. 9(d)) does a better job in depicting the inner features, especially the core.

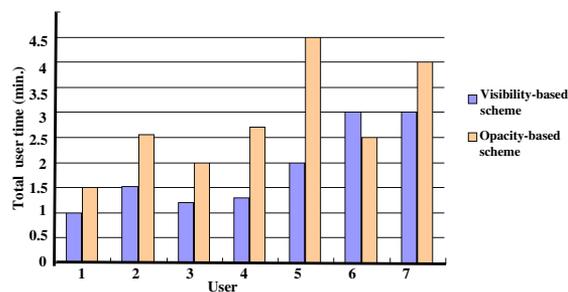
### 5.3. Evaluation and Discussion

To evaluate the usefulness of our method, we asked 7 unpaid graduate students to participate in our user study. Four of them are majored in scientific visualization, and others are in other research groups. Before the test, we explained the differences between traditional opacity-based and visibility-based volume exploration schemes and demonstrated how to use them in our system. The test did not start until they could easily explore the Horseshoe Vortex dataset (Fig. 8) without our help.

Next they were required to explore the CT facial deformity dataset in Fig. 4. We recorded the total interaction time needed to obtain the result in Fig. 4(c) starting from Fig. 4(a) by using these two different exploration schemes. The similarity between the image produced by the participant and Fig. 4(c) is measured with relative mean squared error. We consider the result satisfactory if the error is smaller than 10%. As shown in Fig. 10, most of participants spent at least 30% less time using the visibility-base exploration scheme compared with the time required by using the opacity-based scheme to generate similar results. All users said visibility-based exploration is more intuitive and the default optimization results provided a good starting point for further adjust-

ments. The user feedback suggested our system is an easy-to-use tool for volume exploration.

Nonetheless, the 6th user spent more time using visibility-based scheme than the opacity-based scheme as shown in Fig. 10. After analyzing his interaction, we found he spent more time in small fine tuning after obtaining a satisfactory result. From this case, we can see that our visibility-based exploration scheme is complementary to the traditional opacity-based exploration scheme.



**Figure 10:** User interaction time using visibility-based exploration scheme and opacity-based scheme.

As discussed in Section 4, the optimization result favors small-sized features, which leads to the opacities of large-sized features are too small. This brings inconvenience for the user to understand the spatial relation of features, as demonstrated in Section 5.1 and 5.2. This is a drawback of the current scheme. A possible solution is to incorporate the image footprint into the definition of feature visibility.

Currently, a feature in the feature space is characterized by a template function  $\rho$ . In this sense, different  $\rho$  produces different classification results. A feature can also be defined

as a group of voxels, such as segmented objects. Extension of feature visibility to support more general feature definition is one of future work.

## 6. Conclusion

This paper proposes a new metric: *feature visibility* to measure the influence of each feature to the resulting image. With this metric, traditional transfer function design becomes more informative. To make this procedure more intuitive, the feature visibility based opacity specification is provided to allow the user to directly set the visibility for the features of interest instead of adjusting the opacity. We also introduce a two-step volume exploration scheme, in which an automatic optimization is first performed to provide a clear spatial relation and then the user is allowed to select features and specify the desired visibility for features of interest. This scheme has proven to be effective and efficient over various types of volumetric data, ranging from medical data to fluid simulation data.

## 7. Acknowledgment

This paper was partially supported by the Knowledge Innovation Project of Chinese Academy of Sciences (No. KGGX1-YW-13), 973 program of China (2010CB732504), National Natural Science Foundation of China (No. 60873123), and 863 project (No. 2010AA012402, No. 2010AA012301). The datasets are courtesy of General Electric, Xinliang Li, the Terascale Supernova Initiative, and the OsiriX Foundation.

## References

- [BS05] BORDOLOI U., SHEN H.: View selection for volume rendering. In *IEEE Visualization'05* (2005), pp. 487–494. 6
- [CM08] CORREA C., MA K.: Size-based Transfer Functions: A New Volume Exploration Technique. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1380–1387. 2
- [CM09] CORREA C., MA K.: The Occlusion Spectrum for Volume Visualization and Classification. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1465–1472. 2
- [CM11] CORREA C., MA K.: Visibility Histograms and Visibility-Driven Transfer Functions. *IEEE Transactions on Visualization and Computer Graphics* 17, 2 (2011), 192–204. 1, 2, 3, 5
- [CWM\*09] CHAN M., WU Y., MAK W., CHEN W., QU H.: Perception-based transparency optimization for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1283–1290. 2, 3
- [ER00] EBERT D., RHEINGANS P.: Volume illustration: non-photorealistic rendering of volume models. In *Proceedings of IEEE Visualization'00* (2000), pp. 195–202. 6
- [KD98] KINDLMANN G., DURKIN J.: Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of IEEE Symposium on Volume Visualization'98* (1998), pp. 79–86. 2, 7
- [KKH01] KNISS J., KINDLMANN G., HANSEN C.: Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of IEEE Visualization'01* (2001), pp. 255–262. 1, 2
- [KPI\*03] KNISS J., PREMOZE S., IKITS M., LEFOHN A., HANSEN C., PRAUN E.: Gaussian transfer functions for multi-field volume visualization. In *Proceedings of IEEE Visualization'03* (2003), pp. 65–72. 3
- [KWTM03] KINDLMANN G., WHITAKER R., TASDIZEN T., MOLLER T.: Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications. In *Proceedings of IEEE Visualization'03* (2003), pp. 513–520. 2
- [Lev88] LEVOY M.: Display of surfaces from volume data. *IEEE Computer Graphics and Applications* 8, 3 (1988), 29–37. 2
- [MWCE09] MACIEJEWSKI R., WU I., CHEN W., EBERT D.: Structuring Feature Space: A Non-Parametric Method for Volumetric Transfer Function Generation. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1473–1480. 1, 2, 3, 7
- [Pol69] POLYAK B.: The conjugate gradient method in extremal problems. *USSR Computational Mathematics and Mathematical Physics* 9, 4 (1969), 94–112. 5
- [RBS05] ROETTGER S., BAUER M., STAMMINGER M.: Spatialized transfer functions. In *Proceedings of IEEE/Eurographics Symposium on Visualization'05* (2005), pp. 271–278. 1, 2
- [SG09] SELVER M., GÜZELIS C.: Semi-automatic Transfer Function Initialization for Abdominal Visualization Using Self-Generating Hierarchical Radial Basis Function Networks. *IEEE Transactions on Visualization and Computer Graphics* 15, 3 (2009), 395–409. 2
- [SKK06] SALAMA C., KELLER M., KOHLMANN P.: High-Level User Interfaces for Transfer Function Design with Semantics. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 1021–1028. 2
- [TLM05] TZENG F., LUM E., MA K.: An Intelligent System Approach to Higher-Dimensional Classification of Volume Data. *IEEE Transactions on Visualization and Computer Graphics* 11, 3 (2005), 273–284. 2
- [TM04] TZENG F., MA K.: A cluster-space visual interface for arbitrary dimensional classification of volume data. In *Proceedings of IEEE/Eurographics Symposium on Visualization'04* (2004), pp. 17–24. 2
- [TTFN05] TAKESHIMA Y., TAKAHASHI S., FUJISHIRO I., NIELSON G.: Introducing topological attributes for objective-based visualization of simulated datasets. In *Proceedings of International Workshop on Volume Graphics'05* (2005), pp. 137–236. 2
- [VFSG06] VIOLA I., FEIXAS M., SBERT M., GROLLER M.: Importance-driven focus of attention. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 933–940. 6
- [WCS\*10] WANG Y., CHEN W., SHAN G., DONG T., CHI X.: Volume Exploration using Ellipsoidal Gaussian Transfer Functions. In *IEEE Pacific Visualization Symposium'10* (2010), pp. 25–32. 1, 2, 3, 7
- [WDC\*07] WEBER G., DILLARD S., CARR H., PASCUCCI V., HAMANN B.: Topology-controlled volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 13, 2 (2007), 330–341. 2