Intuitive Transfer Function Editing Using Relative Visibility Histograms

Shengzhou Luo*, John Dingliana†

Graphics Vision and Visualisation Group (GV2), Trinity College Dublin, Ireland

ABSTRACT

We introduce the concept of a relative visibility histogram, which represents the difference between the global visibility distribution across the full volume and the local visibility distribution within a user-selected region in the viewport. From this measure we can infer what the user intends to select when they click on a specific region in the viewport and use this result to directly modify the relevant parts of the transfer function. The approach is lightweight compared to similar techniques and performs in real-time.

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1 INTRODUCTION

A recurring challenge in volume visualization, is defining effective transfer functions (TF), which assign color and opacity (alpha value) to specific data ranges for visualization. Due to the non-linear relationship between the transfer function and the resultant rendering, the process of editing transfer functions is often counter-intuitive, typically necessitating a trial-and-error process. This may be addressed using an output sensitive approach where the user can more directly control the appearance of the visualization, without explicit knowledge of the transfer function.

In this paper we propose a technique which enables us to infer a user’s intended changes to the visualization, when they click or select a region in the rendered image of a 3D volume data set. The is done by weighting the data in the selected region based on the proportion of materials visible to the user within that region. We introduce the concept of a relative visibility histogram, derived from the relationship between the global visibility and the local visibility of data in the user-selected region. Based on this weighting, the user can directly modify colors and opacity of the volume data, in a manner analogous to painting a 3D scene. Compared to other similar techniques, our approach is relatively lightweight, requiring only intermediate information about visibility of data samples. It is thus simple to implement and performs in real-time.

2 RELATED WORK

The visibility of a sample refers to the alpha contribution of a sample to the final image, taking into account the degree to which it is occluded by other samples. This can be computed during ray-casting as the difference between the accumulated alpha of a sample and the accumulated alpha of the previous sample along a ray in the view direction [2]. Correa et al. presented the general notion of visibility histograms [1] which represent the distribution of visibility over intensity ranges in a volume rendering image. Wibel et al. [5] found that the user usually perceives features at a screen position with the highest visibility along a ray and exploited this information for volume picking. Guo et al. [3] proposed a sketch-based manipulation technique for volume visualization based on clustering of attributes such as depth, visibility, alpha and intensity. Guo and Yuan [4] described a sketch-based technique to specify local transfer functions for topology regions using contour trees.

3 RELATIVE VISIBILITY HISTOGRAMS

Global visibility histograms [1] represent the visibility distribution of all the voxels in the viewport. Local visibility histograms represent the local visibility distribution for the voxels that contribute to a region of interest in the final image. Let \( H \) denote the global visibility histogram and \( H_L \) denote the local visibility histogram, both are normalized by dividing each value by the sum of all the values in each histogram. Then the relative visibility histogram \( H_R \) is defined as the difference between \( H_L \) and \( H \) divided by the maximum of the absolute value in the difference. \( H_R = H_L / \max(\text{abs}(H_L)) \), where \( H_L = H_L - H \). The relative visibility histogram is scaled to the range \([-1, 1]\) by dividing by the maximum absolute value in the histogram.

Fig. 1 shows a nucleon dataset, its associated transfer function and sample modifications using our technique. The global visibility histogram is shown in Fig. 2(a) and the local visibility histogram for the region of interest (the rectangle in inverted color) is shown in Fig. 2(b). The relative visibility histogram is show in Fig. 2(c).

In order to smooth the histogram, we apply a Gaussian kernel to \( H_r \) and then scale it to the range \([-1, 1]\). So the smoothed relative visibility histogram is \( H_G = H_R / \max(\text{abs}(H_R)) \), where \( H_G = \text{Gaussian}(H_R, n, \sigma) \), \( n \) is the size and \( \sigma \) is the standard deviation of the Gaussian kernel (see Fig. 2(d)). Henceforth, this smoothed histogram \( H_G \) will be referred to as the relative visibility histogram, and \( H_G(i) \), which is the value \( H_G \) at intensity \( i \), will be

\*e-mail:huos@scss.tcd.ie
\†e-mail:john.dingliana@scss.tcd.ie

![Figure 1](image1.png)

![Figure 2](image2.png)

![Figure 2](image3.png)
We have implemented our approach with a GPU-Raycast volume renderer with the visibility computation and rendering implemented on a GPU using CUDA. The implementation is quite lightweight, and achieves real-time performance at 36 to 40 frames per second on a computer equipped with an Intel Xeon E3-1246 v3 CPU and a NVIDIA Quadro K4200 graphics card.

Fig. 3(left) displays the result of both applying color blue and adjusting alpha of the TF in Fig. 1(b). Note that the intensity ranges with initial red color in the middle of the transfer function have been blended with blue and have become purple. Similarly, Fig. 3(right) shows the result of applying yellow and adjusting alpha. Here, the intensity ranges in the middle have become orange after blending with the color yellow. In both cases, the alpha of the relevant parts of the transfer function are increased and the alpha of the less relevant parts are decreased in order to emphasize the materials of interest.

5 RESULTS AND CONCLUSIONS

We have implemented our approach with a GPU-Raycast volume renderer with the visibility computation and rendering implemented on a GPU using CUDA. The implementation is quite lightweight, and achieves real-time performance at 36 to 40 frames per second on a computer equipped with an Intel Xeon E3-1246 v3 CPU and a NVIDIA Quadro K4200 graphics card.

Fig. 3(left) displays the result of both applying color blue and adjusting alpha of the TF in Fig. 1(b). Note that the intensity ranges with initial red color in the middle of the transfer function have been blended with blue and have become purple. Similarly, Fig. 3(right) shows the result of applying yellow and adjusting alpha. Here, the intensity ranges in the middle have become orange after blending with the color yellow. In both cases, the alpha of the relevant parts of the transfer function are increased and the alpha of the less relevant parts are decreased in order to emphasize the materials of interest.

4 COLOR AND ALPHA EDITING

When a user selects a region of interest on the rendered image of the volume, a single pass of volume ray casting is done to calculate the global visibility histogram for the whole volume and the local visibility histogram for the selected region in the view-port. From this we calculate the relative visibility histogram, which provides a measure of visible materials within the selected region. This is used to infer features that the user intends to edit in the visualization. More precisely, \( H_G \) is used as a weighting function to blend the colors or opacities in the original transfer function with a user-selected target color or alpha value.

The user-selected target color is blended with the original transfer function for intensity ranges that have positive values in the relative visibility histogram as below:

\[
C_i = \begin{cases} 
C_i + H_G(i)(C_s - C_i) & \text{if } H_G(i) > 0 \\
C_i & \text{otherwise}
\end{cases}
\]

where \( H_G(i) \) denotes the relative visibility at intensity \( i \) in \( H_G \), \( C_s \) is the user-selected target color and \( C_i \) the color of intensity \( i \) in the original transfer function.

Similarly, the alpha \( (A_i) \) of the transfer function is increased in intensity ranges that have positive relative visibility values, and decreased for ranges with negative relative visibility values, as follows:

\[
A_i = \begin{cases} 
A_i + H_G(i)(1 - A_i) & \text{if } H_G(i) > 0 \\
A_i - H_G(i)(0 - A_i) & \text{otherwise}
\end{cases}
\]

Note that the color blending and alpha blending operations can be applied separately. Fig. 1(c) displays the result of only applying the color blending to the volume rendering of the nucleon data set in Fig. 1(a), and Fig. 1(d) displays the result of only applying the alpha blending to the original.

REFERENCES