Eurographics Conference on Visualization (EuroVis) (2014)
N. Elmqvist, M. Hlawitschka, and J. Kennedy (Editors)

Short Papers

VolVis Project: Individual Extensions

Isidoro Tamassia - 6054765

1. Extension 1: Maximum Intensity Difference Accumulation

The first extension that I implemented is the MIDA algorithm, as referenced in the original paper [1]. Essentially, MIDA aims to strike a balance between Direct Volume Rendering (DVR) and Maximum Intensity Projection (MIP) techniques. This balance is particularly beneficial in scenarios where the transfer function may not be optimally defined or adapted for the specific dataset. The process begins with the standard DVR compositing approach. However, MIDA's unique aspect is that it places greater emphasis on those parts of the ray where the maximum value exhibits more significant changes. Given a transfer function f, the value f_{P_i} of the sample at step i and the current maximum f_{max}, we define δ_i s.t.:

$$\delta_i = f_{P_i} - f_{max} \text{ if } f(P_i) > f_{max}$$
$$\delta_i = 0 \text{ otherwise}$$

And $\beta_i = 1 - \delta_i$. We then modify the compositing formula in the following way:

$$c_i = \beta_i c_{i-1} + (1 - \beta_i \alpha_{i-1})\alpha(f_{P_i})c(f_{P_i})$$
$$\alpha_i = \beta_i \alpha i - 1 + (1 - \beta_i \alpha_{i-1})\alpha(f_{P_i})$$

Through this adjustment, MIDA effectively modulates both color and opacity accumulated along a viewing ray, based on the intensity of the maxima within the ray. It's evident from the formula that sharp, discontinuous maxima induce more modulation compared to gradual increases in ray intensity.

The algorithm is implemented in the function `traceRayMIDA` and can be selected from the GUI as usual.

1.1. Test examples

An extreme case that I found out in which MIDA yields way better results than, for instance, standard DVR, is reported in Figures 1 and 2. In these examples, the transfer function has been adjusted to visualize the organs of the fish. However, in Figure 1 standard DVR is applied, which seems to not work at all, yielding an almost completely black result. Contrarily, Figure 2 shows how MIDA allows us to correctly visualize the parts of the volume we were looking for.

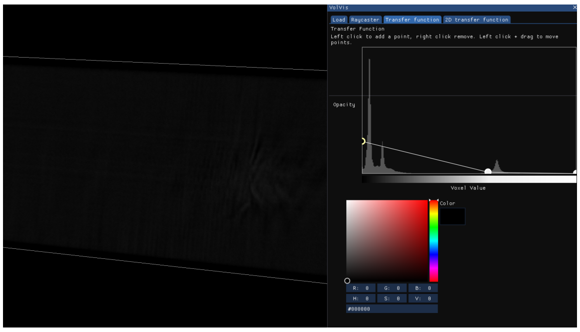

Figure 1: *carp8 rendering with DVR compositing*

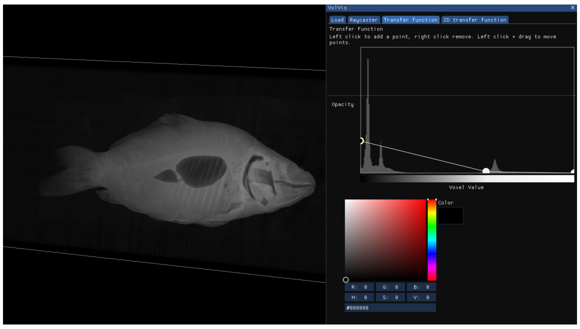

Figure 2: *carp8 rendering with MIDA compositing*

2. Extension 2: Smooth transitions between MIDA, DVR and MIP

We may be interested in a visualization that is closer to either DVR or MIP, but preserves some of the good properties of the more balanced MIDA. To do so, we can add another parameter $\gamma \in [0,1]$ [1]. Then we consider two cases:

2.1. $\gamma \leq 0$: MIDA to DVR

This is the simplest case, where we just modify the value of β according to this formulation:

$$\beta_i = 1 - \delta_i(1+\gamma) \text{ if } \gamma < 0$$
$$1 - \delta_i \text{ otherwise}$$

This way, lower values of γ will produce progressively more faded values, reaching the same result of DVR when $\gamma = -1$.

2.2. $\gamma > 0$: MIDA to MIP

To mitigate issues related to shading, here we follow a different approach by adopting linear interpolation between the accumulated color and opacity and that of the current maximum value encountered along the ray, using γ as the interpolation weight [1]. It can be shown that this yields a progressive reduction of shadows and darkened areas until reaching a result that is equivalent to MIP when $\gamma = 1$.

This more flexible version of MIDA is implemented into the function `traceRayCombined`, and the user can control its behavior by changing the value of γ using the GUI.

2.3. Test examples

In Figure 3 we can see how selecting $\gamma = -0.8$, the image gets very dark compared to MIDA ($\gamma = 0$) but is still visible, differently from DVR (Figure 1, equivalent to setting $\gamma = -1$). On the other hand, in Figure 4 we set $\gamma = 0.4$, and we can see how the result approaches the one that would be the output of MIP (equivalent to setting $\gamma = 1$).

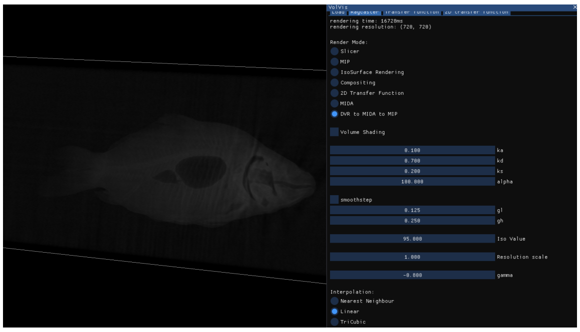

Figure 3: *carp8 rendering with modified MIDA, $\gamma = -0.8$*

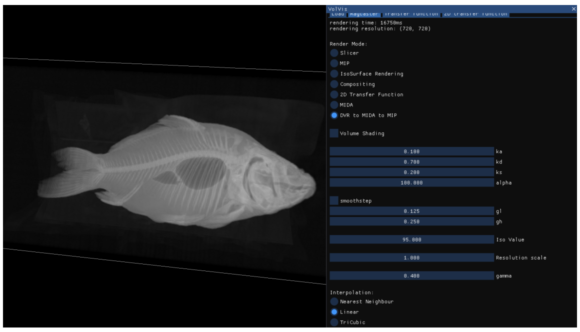

Figure 4: *carp8 rendering with modified MIDA, $\gamma = 0.4$*

3. Extension 3: Shading + Smooth-step

While employing the standard Phong shading, we address the issue of gradient noise in nearly homogeneous regions [1] by blending unshaded and shaded colors. This blending is governed by a weight calculated through a 'smooth-step' function, which varies between 0 and 1 as the gradient's magnitude $|\nabla f_{P_i}|$ moves from g_l to g_h. By selecting an appropriate $[g_l, g_h]$ range for each volume, we effectively diminish artifacts and noise. This shaded rendering, along with the smoothstep adjustment, is accessible via the GUI, allowing users to fine-tune g_l and g_h for different volumes.

3.1. Test examples

In Figure 5, standard DVR and Phong shading are applied to the `pig8` volume data, with the transfer function aimed at visualizing the pig's exterior. This results in considerable noise, particularly in the background, and inadequate shading in areas like the legs and the right side of the pig. Similarly, using standard MIDA with Phong shading does not improve the outcome. Conversely, Figure 6 demonstrates the effectiveness of integrating 'smoothstep' interpolation on the gradient's magnitude. This approach significantly reduces background noise and enhances the clarity of the pig's form, evident in the more accurately rendered legs and right side.

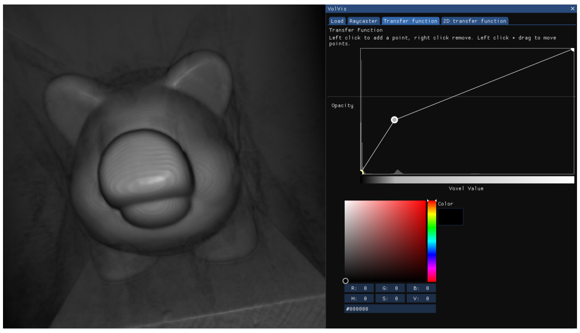

Figure 5: *pig8 rendering with DVR and Phong shading*

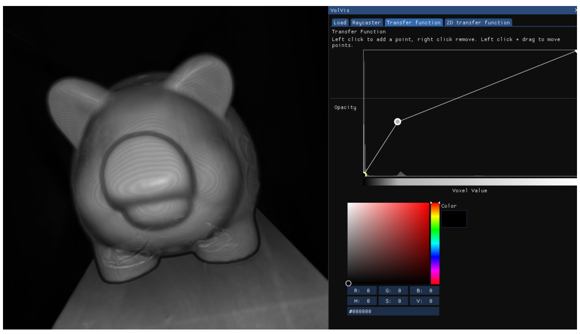

Figure 6: *pig8 with MIDA and smoothed Phong shading*

References

[1] Stefan Bruckner and M. Eduard Gröller. “Instant volume visualization using maximum intensity difference accumulation”. In: *Computer Graphics Forum* 28.3 (2009), pp. 775–782.