IEEE/EG International Symposium on Volume Graphics (2010), pp. 1–8
R. Westermann and G. Kindlmann (Editors)

Depth-Enhanced Maximum Intensity Projection

J. Díaz and P. Vázquez

MOVING Group, Universitat Politècnica de Catalunya

Abstract

The two most common methods for the visualization of volumetric data are Direct Volume Rendering (DVR) and Maximum Intensity Projection (MIP). Direct Volume Rendering is superior to MIP in providing a larger amount of properly shaded details, because it employs a more complex shading model together with the use of user-defined transfer functions. However, the generation of adequate transfer functions is a laborious and time costly task, even for expert users. As a consequence, medical doctors often use MIP because it does not require the definition of complex transfer functions and because it gives good results on contrasted images. Unfortunately, MIP does not allow to perceive depth ordering and therefore spatial context is lost. In this paper we present a new approach to MIP rendering that uses depth and simple color blending to disambiguate the ordering of internal structures, while maintaining most of the details visible through MIP. It is usually faster than DVR and only requires the transfer function used by MIP rendering.

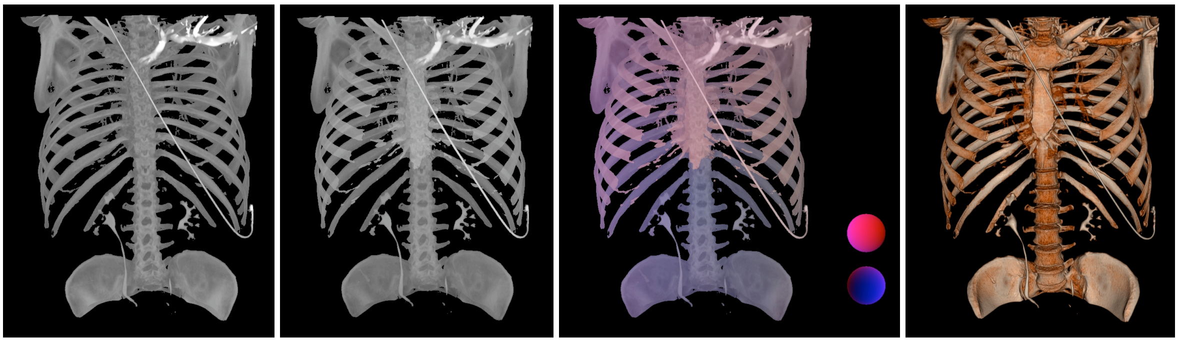
Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithms

1. Introduction

Volume graphics have evolved rapidly during the last decade, hand to hand with graphics hardware evolution. Nowadays, we may render larger models, with more complex transfer functions and in sophisticated layouts such as by combining multimodal data sources. Although this has lead to an increase of problems addressed, such as surgery planning, surgery assistance, and so on, several, simple rendering techniques, are still widely used amid a large bunch of new rendering methods that include complex transfer functions, focus+context visualization, or exploded views. Concretely, Maximum Intensity Projection (MIP) [MHG00], has the advantage of not requiring a complex Transfer Function definition and is useful for datasets where the interesting features have high intensity values, often obtained through the use of contrast elements. Unfortunately, MIP rendering does not provide enough depth features in order to maintain contextual information, and therefore, often users do not distinguish from a front and a back view unless some manipulation is allowed.

Some attempts have addressed the problem of loss of context in MIP, such as the method by Heidrich *et al.* [HMS95] for polygon rendering, or the one by Bruckner and Gröller [BG09]. In the first case, the actual depth of rendered structures is changed in order to improve depth perception. In the second, the rendering algorithm itself is modified, leading to a technique more similar to DVR, but that requires less effort in transfer function definition. Our approach lies somewhere between those two techniques: We improve the perception of depth by changing how intensity is accumulated, though our images are similar to the original MIP technique. This produces results like the ones obtained by Heidrich *et al.*, but without the need to extract isosurfaces. We directly render the model using a classical GPU raycasting. A second modification we added over MIP is the use of a sphere-based color mapping. This further improves the perception of depth because points at different depths are tinged with different colors. We show this two effects compared with MIP and DVR in Figure 1. In order to evaluate our results, we have carried out a user study that shows that the rendering obtained with our technique effectively improves context and depth perception.

The rest of the paper is organized as follows: The next section reviews the previous work. Section 3 introduces our



(a) MIP (b) DEMIP (c) DEMIP with Color (d) DVR

Figure 1: *Comparison of MIP-based rendering vs our approach (DEMIP and DEMIP with color) for the model of the thorax. Rightmost image provides a DVR rendering of the same model. The spheres in the third image show respectively the front and back spherical mapping that is applied to the model in order to aid in the correct interpretation of the spatial distribution of the elements.*

method. Section 4 discusses the obtained results. We conclude in Section 5 where we sum up and analyze possible future research.

2. Related Work

The correct perception of depth in volume rendering is often complex. Even DVR techniques often suffer from poor depth cues because the datasets commonly have a large number of fine overlapping structures. Thus, some techniques such as halos [EBRI09, BG07, SE03] or improved shadowing [PM08, RKH08] have been developed in order to improve the understanding of complex models. For a deep work on depth perception in Computer Graphics, the interested reader can refer to the early experiments by Wanger *et al.* [WFG92], or the more recent Pfautz's work [Pfa02].

For Maximum Intensity Projection the perspective is even worse, because the use of the maximum operator does not provide any context cue. However, as stated previously, the use of MIP has other advantages such as the limited effort required to create a good transfer function together with the fact that it provides a good understanding of the structures represented by high signal intensities [ME05]. Several attempts have been carried out in order to improve the perception of depth in MIP rendering. Heidrich *et al.* [HMS95] improve the perception of depth in a polygon rendering application. The core idea is to extract the required iso-surfaces and transform the geometry in order to replace a polygon's depth by the value of its iso-surface. Then, MIP is approximated by displaying the z-buffer in grayscale. Local Maximum Intensity Projection (LMIP) [SPN*98] overcomes the limitations of MIP by selecting local maximum values along optical rays instead of the maximum values. These local maximum values are values over a user-defined threshold. If no value above this threshold is found, the pixel is rendered using the maximum computed value.

Ropinski *et al.* [RSH06] do improve depth perception in angiography by a set of techniques such as edge enhancement or changing the color of the rendered structures. The color change (pseudo chromadepth) is based in the fact that the lens of the eye refracts colored light with different wavelengths at different angles. As a result, blue objects are perceived as being far away, while red objects are perceived as being closer.

Bruckner and Gröller [BG09] propose MIDA, a rendering method in between DVR and MIP. It does not require complex transfer function definition but it also provides occlusion cues similar to DVR. Other scientists have addressed the problem of the automatic or semi-automatic generation of transfer functions ([KD98]) which is orthogonal to our approach because we want to obtain a depth-enhanced MIP-like result (almost) without loss of details. Chan *et al.* [CWM*09] modify DVR based on perception principles. Their main objective is to improve the final quality, by evaluating the perception of the semi-transparent layer from the visibility, shape, and transparency aspects. However, this approach is mainly focused for DVR, not MIP.

Our method is in the spirit of Heidrich *et al.* [HMS95], as we modify the weighting of iso-surfaces according to their depth, but this is done in realtime using a GPU-based volume ray casting. Thus, we do not require the previous iso-surface extraction. We obtain visually similar results to their system. We combine this method with a technique similar to the pseudo chromadepth by Ropinski *et al.* [RSH06]. In our case, the color is mapped using an auxiliary sphere that shows the color map. This sphere can be manipulated by the user (i. e. rotated) and the color distribution in the sphere is mapped to the object. This way the user may use the sphere to infer the spatial distribution of the structures of interest if the previous cues where not enough to disambiguate their position. We may use a similar color combination to that of

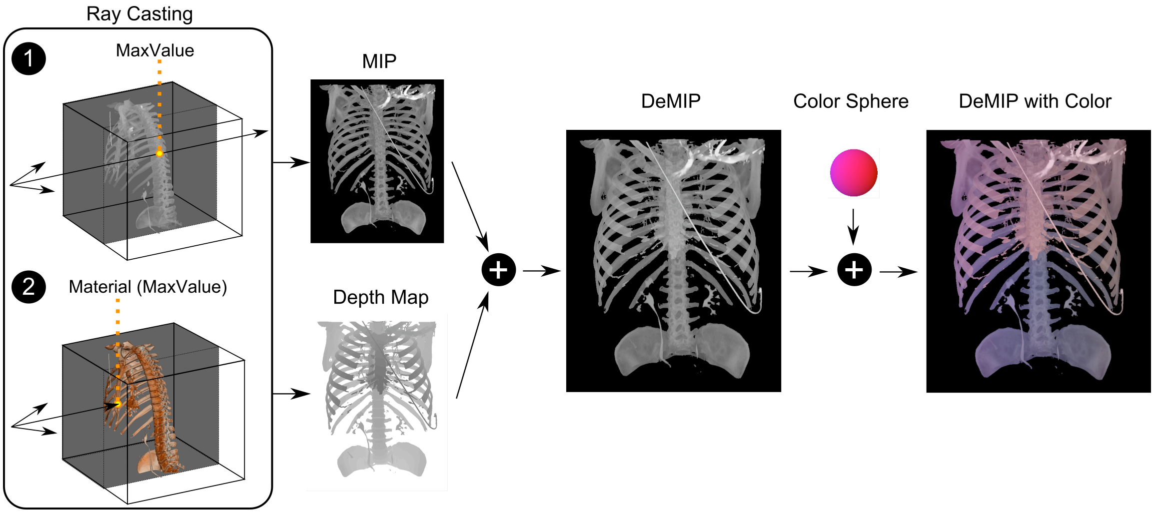


Figure 2: *Fragment shader overview. The fragment shader traces two rays: the first one determines the maximum intensity. Then, it looks for the first point where the same material is found. Finally, the point is shaded taking into account the new depth, and eventually, the color is tinged using the color sphere.*

Ropinski *et al.* (see Figure 7-left) or a different color map (see Figure 7-center and right) with similar results because the depth perception is reinforced by the knowledge of the color distribution the user obtains by looking at the auxiliary sphere.

3. Depth-Enhanced Maximum Intensity Projection

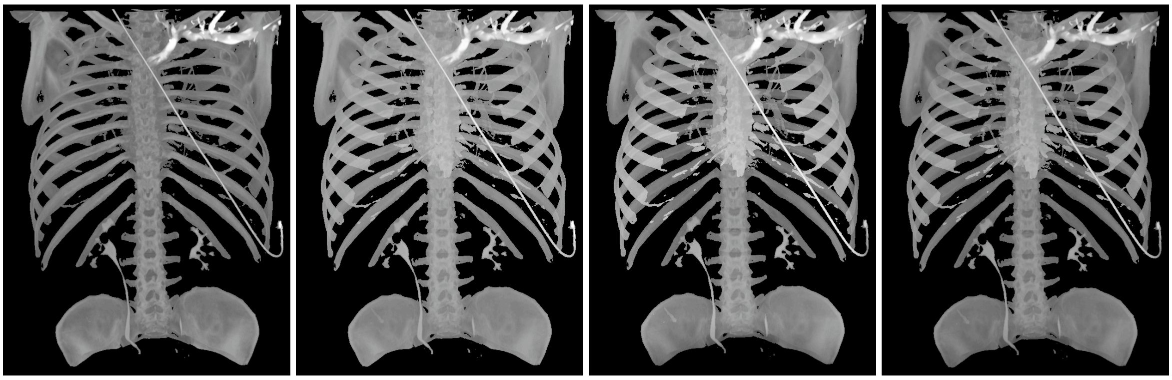
Large, complex structures, are often difficult to interpret when using MIP rendering. Spatial comprehension is often impaired by the lack of occlusions. Even in the presence of spatial binocularity, the perception of depth is often challenged by inconsistencies in size and disparity. As a conclusion, depth is sometimes judged inaccurately. Our work combines two tools in order to improve the depth perception in MIP rendering: occlusion revealing and spatial color-coding. In the first case, we modify the shading color according to depth in order to make occlusions visible. The second element is a fragment tinging function that changes the color conforming to the 3D sampling position.

3.1. Overview

Our algorithm basically consists on a two-pass GPU volume ray-casting executed in a single fragment shader. In a first step, rays are traced in order to obtain a MIP image. At each ray sample, shading is calculated by applying the MIP transfer function to the density value. The resulting value is the *material*. Once the ray reaches the maximum intensity, the material at this point is stored (step 1 in Figure 2). With this value, a second ray is traced, following the same path. When the new ray reaches the same material that caused the maximum intensity again, the depth of this ray is stored and the ray is terminated (step 2 in Figure 2). The idea behind this approach is to look for a structure with the same *material* than the one that produced the maximum intensity, and change its shading in order to make the occlusion visible. When the reached point is the same (i. e. there are not previous occlusions), the rendering will be the same than with MIP. Otherwise, the MIP value is weighted with the depth, and, eventually, a color. As a means to detect the same material, we actually look for a value *around* the value of interest with the use of a threshold. Differently, we may not detect the same material even if it exists due to the discrete sampling and this causes noise in the images. Changing this factor may enhance different structures.

3.2. Depth weighting

The key issue for our method is a good evaluation of the distance of the point to shade to the near plane. We could consider the OpenGL depth of the ray at the point of interest. However, since we use perspective projections, this value is not very accurate and yields poor results. Because we work in a Virtual Reality environment, efficiency is an important issue. Therefore, we evaluate several cheap approximations to the exact depth: First, the number of ray steps before it is terminated. Unfortunately, the classical GPU volume ray casting algorithm traces rays from the boundary of the bounding box. Therefore, the number of steps covered is not proportional to the distance to the viewer. Finally, we evaluate the distance by adding the initial ray start position OpenGL depth to the number of steps traced. This yields a value that is correctly mapped to zero at the near



(a) MIP (b) DEMIP, OpenGL depth (c) DEMIP, ray length inside bounding box (d) DEMIP, exact depth

Figure 3: *Comparison of MIP-based rendering vs the DEMIP algorithm using different depth computation methods. Left image shows the MIP Rendering result. Columns second to fourth show the different ways to compute depth: OpenGL depth value, number of ray steps inside the bounding box, and exact depth, respectively.*

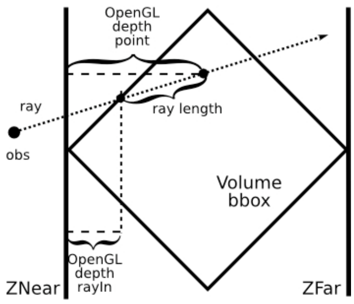


Figure 4: *Depth computation methods: The first approach uses* OpenGL depth*, the second, the* ray length *inside the bounding box, and the third, the one that yielded the best results, uses* OpenGL depth rayIn + ray length.

plane, and almost equal to the exact distance (actually, the images generated using this depth are visually indistinguishable from the ones using the exact depth). This results in a shader slightly simpler than the one required to compute the exact value. However, no clear performance gains were observed. The results using these different methods are shown in Figure 3, and the elements used to evaluate the depth are depicted in Figure 4. This method is superior to the previous two. The second method, while yielding good static views, produces artifacts near the boundary of the bounding box, noticeable when rotating the object. Given a certain the MIP color, the final color is calculated using the depth ($depth$) and the weight (d_w) we apply to the depth value:

$$color = MIP_{color} * (1 - d_w) + 2 * d_w * (1 - depth) \quad (1)$$

If $d_w = 0$, the result we obtain is the MIP color. The two parts of the formula have been added this way in order to slightly lighten the regions closer to the user and slightly darken the regions far from the observer. In most cases, assigning a value of around 0.15 to d_w is enough for enhancing the occlusions (while keeping the details) of the structures of interest, though the user may change this value if it does not fulfill his or her needs for a certain model. Figure 5 shows the effect of d_w.

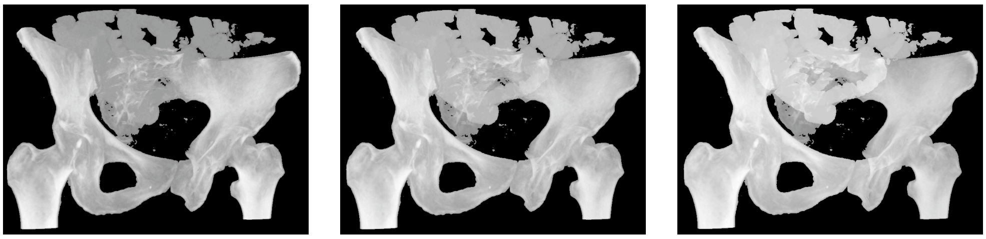
3.3. Color Sphere

We add a second element that provides a color cue for the improvement of depth perception. In this case, what we do is to use the 3D position of the iso-surface determined by the second ray casting in order to tinge the MIP color using a spherical texture map. The spherical map is initially textured using a color for the front region and a different color for the back part, such as red and blue, respectively. These colors are interpolated across the surface of the sphere. Given a 3D point, we build a vector with initial point at the center of the bounding box and endpoint at the 3D point. Then, we intersect the bounding sphere with a ray whose direction is the one indicated by this vector, and take the color of the texture map ($color_{sph}$) to modify the color computed with the previous algorithm ($color_{DEMIP}$). The color change is performed using the following formula:

$$color = color_{DEMIP} * (1.0 - sph_w) + color_{sph} * sph_w, \quad (2)$$

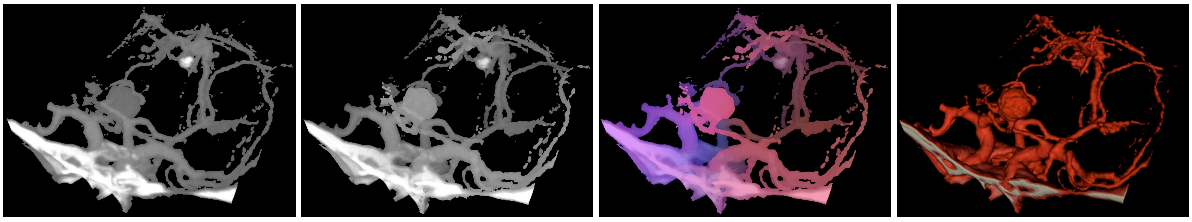
where sph_w is the weigh we give to the sphere color. We may see the result of the spherical-based color mapping in Figure 6. Note how the color sphere further improves over DEMIP and the perception of the model orientation is easier to infer (the correct orientation can be appreciated in the DVR-generated image at the right column).

The spherical map is represented in screen with a 3D



(a) MIP: $d_w = 0$ (b) DEMIP: $d_w = 0.15$ (c) DEMIP: $d_w = 0.3$

Figure 5: *Effect of the depth weight (d_w) for the hip model. Using $d_w = 0$ we obtain regular MIP.*



(a) MIP (b) DEMIP (c) DEMIP with color (d) DVR

Figure 6: *Comparison of MIP-based rendering vs DEMIP and DEMIP with color for a model containing an aneurysm. The rightmost image shows the DVR rendering for comparison purposes.*

sphere that can be rotated by the user. This way, possible ambiguities can be resolved without the need of moving the model, because the user can see the color map applied to the sphere and then analyze how the visible structures are colored. We also provide a small set of predefined color mappings, as shown in Figure 7, for the user to select the most adequate (or the colors her or she sees better).

4. Results and Discussion

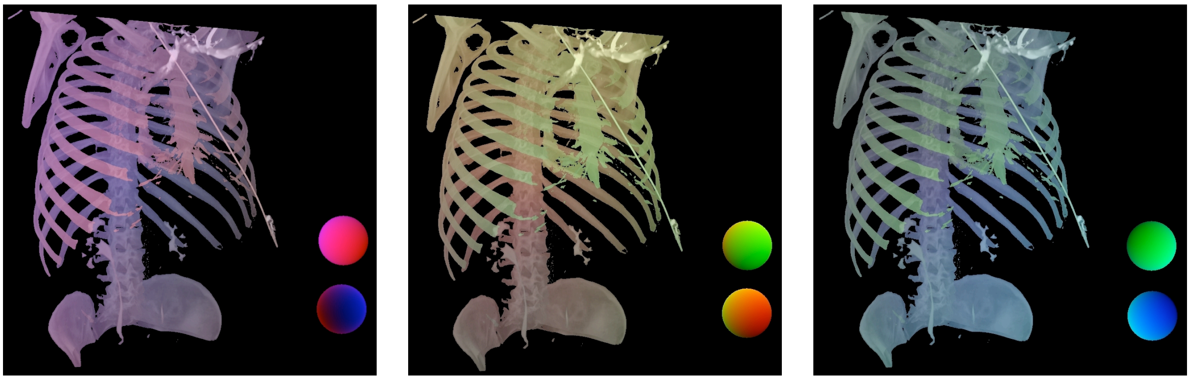
Model	Resolution	MIP	DEMIP	MIDA	DVR
Jaw	$512^2 \times 40$	64.7	53.8	48.1	49.8
Aneurysm	$512^2 \times 120$	76.2	54.5	43.9	45.8
Abdomen	$512^2 \times 171$	59.4	51.8	50.1	53.4
Head	$512^2 \times 485$	28.9	25.8	17.3	17.6
Body	$512^2 \times 512$	28.1	24.1	26.6	27.1

Table 1: *Comparison of framerates between our algorithm, DEMIP, and the classical MIP algorithm. We also provide timings for MIDA and DVR for reference purposes. Timings are in frames per second, taken with one sample per voxel, as more samples do not visually affect image quality in MIP or DEMIP. Viewport size is 512×512.*

We have implemented our algorithm on an Intel Core 2 Duo CPU E8400 @ 3.0GHz equipped with 4 GB of RAM. The graphics card is a nVidia GeForce GTX 280 GPU with 1GB of RAM memory. Although some accelerations can be done to the MIP rendering ([ME05]), we use a regular GPU raycasting-based MIP, as we believe this will be a better framerate reference for most of the readers. Table 1 shows framerates for a set of models. Although timings strongly depend on the Transfer Function, MIP and DEMIP are generally faster than DVR because the computation at each sample is simpler. For models where the transfer function only shades isolated structures such as in the body model, DVR may be faster, because the more costly part of DVR, essentially the Phong shading, is only evaluated at a relative low number of points. In this case the cost of the second rays traced by the DEMIP algorithm is higher than the Phong shading of DVR. DEMIP is slower than MIP because there is an extra ray casting step, although we still have high framerates for all the models.

4.1. Single vs Multiple pass ray-casting

Initially, we considered a three-pass version of our algorithm: First, a MIP ray casting that stores in the alpha value of the color texture the material sampled when rays reached maximum intensity and were therefore terminated. Then, a second ray casting that traced the rays until reaching a material equal (similar up to an epsilon value) to the one in which the ray terminated. And then a third step that modified the color computed in the initial step using the depth computed in the second step. Unfortunately, though easier to program, this multiple pass rendering scheme is more costly than performing the whole work in a single fragment shader. The difference is small, specially when comparing the use of two or three rendering passes. However, the difference between the



(a) Red-Blue (b) Green-Red (c) Green-Blue

Figure 7: *Example of different pre-defined color maps applied to the DEMIP rendering in order to improve spatial distribution comprehension. Although Red-Blue can be suitable for a lot of users, some may perceive other color combinations better.*

algorithm in a single pass and two passes can vary approximately the 10% of the framerate, as shown in Table 2. The reason is difficult to evaluate. The key difference between the algorithm implemented in a single fragment shader and in two different steps is the 2D texture required to store the intermediate results. Probably, the generation of these 2D textures with the corresponding extra queries make the two rendering passes strategy more costly. But we may also consider than a large, complex shader may be more suitable for load distribution between GPU cores, as shown for GPU-based raytracing of polygonal scenes [AL09].

Model	1 pass	2 passes	3 passes
Jaw	53.8	49.7	49.4
Body	24.0	22.7	22.7
Head	25.8	25.0	24.9

Table 2: *Comparison of framerates with different implementations of DEMIP. The first one uses a complex fragment shader that traces the two rays. The second column shows the framerates for the implementation that uses two fragment shaders, one for the first MIP tracing, and another for the second ray and color composition. The rightmost column separates the final color composition in a different shader.*

4.2. Quality evaluation

In order to evaluate the results of our algorithm, we conducted an informal user study among a set of 12 people, that are either computer scientists or computer science students, most of them with Computer Graphics background. The users were asked to solve 6 perception problems, the first three were 2D problems were the next ones were stereo. The images were provided randomly, in order to avoid a learning effect. In the case of the stereo problems, DVR was provided first because it was the reference image, the other ones were shown in random order.

The first three problems are:

1. Determine if the body is facing towards or backwards the viewer.
2. Determine if the missing tooth belongs to the left or the right part of the mouth in a jaw model.
3. Determine if the head model is facing towards the observer or it is facing backwards.

For these three problems, the users were given a set of views that were generated using MIP, DEMIP with OpenGL depth, DEMIP with ray length depth, DEMIP with exact depth, and DEMIP with Color Sphere. For the case of the body model, shown in Figure 6, the results were clearly favorable to DEMIP-improved images. With the MIP version, the majority of the users chose the wrong answer, while all DEMIP-improved images with or without color sphere correctly disambiguated the orientation for almost all of them.

The other two problems show more interesting results. For the case of the head, MIP rendering completely confuses the users when given a static view, and although the use of DEMIP improves the result, only the Color Sphere completely disambiguates the orientation, as shown in Figure 8. The missing tooth problem is an example where the users first do not know where the hole belonging to the missing tooth is if they see the MIP image. All the DEMIP-improved images reduce uncertainty, notably DEMIP with exact depth. Best results are obtained, again, with the Color Sphere improvement (see Figure 9).

Then, we started with stereo rendering. We gave the users three DVR images and next, some images using MIP, DEMIP, and DEMIP with color. They were asked to evaluate in a seven point Likert scale (1 strongly disagree, 7 strongly agree) whether they think the new stereo image helped to perceive the spatial distribution of structures as they were in

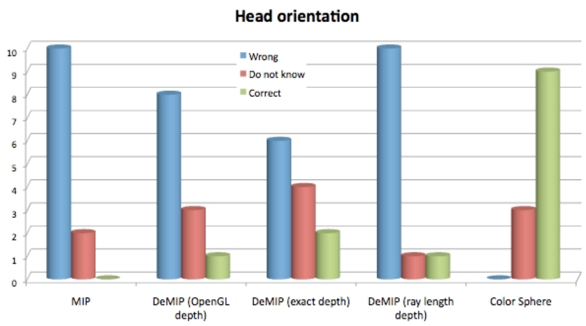


Figure 8: *Evaluation of the head orientation. While the DEMIP method with exact depth produces better results than MIP rendering, still most of the users do not correctly interpret the orientation of the skull. The combination of DEMIP with Color Sphere clearly disambiguates the view.*

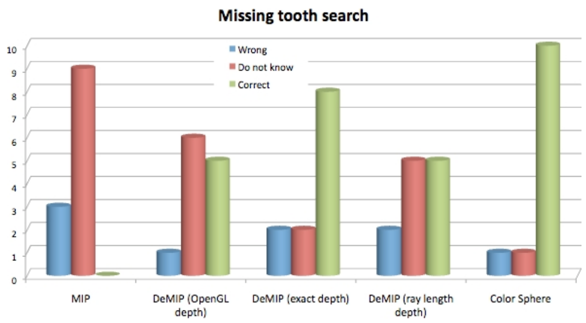


Figure 9: *Finding the missing tooth. In this example, the users clearly see that the tooth missing belongs to the left part of the jaw when using DEMIP or Color Sphere methods, while MIP rendering is completely ambiguous.*

DVR. Data was collected and analyzed estimating 95% confidence intervals for the means with the ANOVA analysis of the answers. We found the following conclusions: We may state with our test set, that with a confidence level of 95%, one may expect that the answer will be between the limits shown in Table 3. Thus, DEMIP and DEMIP with color are better rendering styles to perceive models orientation. Finally, we also asked other questions, briefly summarized:

- Is stereo better than mono for depth perception? There was unanimity in preferring stereo rendering.
- Do DEMIP or DEMIP plus Color Sphere lose information with respect to MIP? 60% said no, while 40% said yes.
- Which technique is the most helpful? More than 90% of the people preferred Color Sphere and more than 60% found DEMIP also useful.

5. Conclusions and Future Work

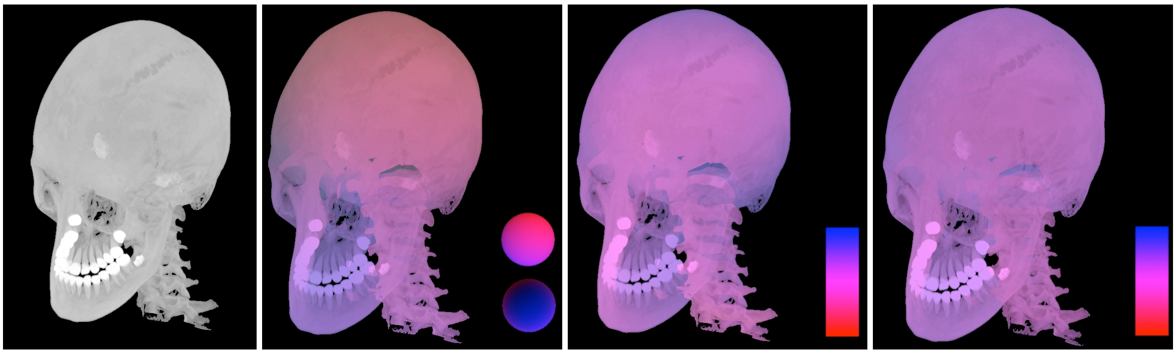
In this paper we presented a new technique tailored to improve depth perception in MIP rendering. Our algorithm adds two different visual cues: occlusion revealing and depth-based coloring. In the first case, we modify the MIP color in the presence of occluders with the same material than the ones at the point of maximum intensity. In the second case, the actual position of the shaded fragment is used to change its color using a supporting spherical map. The depth-based intensity change obtains visually similar results than previous approaches without the need to extract isosurfaces, and using directly the same Transfer Function as the MIP rendering (built using the window/level and optionally reducing the density values to visualize). We summarize the features of our method compared to others found in literature in Table 4. Our color change has similarities with the pseudo-chromadepth in [RSH06], although the motivation is different. In their case, the objective is to add a color that helps perceiving distances. In our case, the idea is to reinforce overall spatial distribution perception, and provide a visual reference that can be manipulated by the user. Hence, if the user moves the sphere, the mapping makes the colors in the model to be changed accordingly. This way, with the movement of the auxiliary sphere, the user can get further cues to infer the spatial positions of the visible structures. The user can also change the colors mapped in the sphere in order to select the ones that he or she finds more com-

95%	**MIP**	**DEMIP**	**DEMIP with color**
Lower bound	2.33	3.66	4.99
Upper bound	3.50	4.83	6.16

Table 3: *95% confidence intervals for stereo rendering.*

Method	**Most suitable for**	**Features**
MIP	Contrasted models	Spatial context is lost
DSMIP	Contrasted models	Requires isosurface Polygon-based rendering
LMIP	Contrasted models	Improves contextual information Parameter requires tuning May occlude details
DEMIP	*Contrasted* models	*Improves contextual information* *Most details preserved* *Slower than MIP* *Faster than DVR*
MIDA	General volumes	Little TF definition effort Improves contextual information Occludes details
DVR	General volumes	Hard TF definition effort Improves contextual information Occludes details

Table 4: *Comparison of DEMIP method with other MIP-based methods. We also compare with DVR and MIDA, although their objectives are usually slightly different.*



(a) MIP (b) DEMIP with color (c) DEMIP with chromadepth (d) MIP with chromadepth

Figure 10: *Color mapping strategies: DEMIP with sphere color (b) and DEMIP with pseudo-chromadepth (c) show little differences, but the occlusion emphasis obtained by the depth-enhancement becomes notorious if we compare (b) or (c) against a MIP rendering enhanced by applying the pseudo chromadepth as described by Ropinski* et al.*, shown in (d).*

fortable. This can overcome possible color issues such as color blindness in the users. We believe this is a powerful tool thanks to its flexibility and because it does not require moving the object. In Figure 10 we compare our sphere-based color mapping and the pseudo-chromadepth applied to a depth-enhanced MIP (center images), and the simple MIP rendering with pseudo-chromadepth as described by Ropinski *et al.* [RSH06]. Note how the occlusions revealed by the depth-enhancement are not visible with the other method. In future we want to evaluate this approach with medical doctors and find ways to improve framerates.

Acknowledgments

The authors want to thank the reviewers for their useful comments. Project supported by TIN2007-67982-C02-01 of the Spanish Government. J. Díaz is supported by grant FPI BES-2008-002404.

References

[AL09] AILA T., LAINE S.: Understanding the efficiency of ray traversal on gpus. In *HPG '09: Proceedings of the Conference on High Performance Graphics 2009* (New York, NY, USA, 2009), ACM, pp. 145–149.

[BG07] BRUCKNER S., GRÖLLER E.: Enhancing depth-perception with flexible volumetric halos. *IEEE Transactions Visualization and Computer Graphics 13*, 6 (2007), 1344–1351.

[BG09] BRUCKNER S., GRÖLLER M. E.: Instant volume visualization using maximum intensity difference accumulation. *Computer Graphics Forum (Proceedings of EuroVis 2009) 28*, 3 (2009),

[CWM*09] CHAN M.-Y., WU Y., MAK W.-H., CHEN W., QU H.: Perception-based transparency optimization for direct volume rendering. *IEEE Trans. Vis. Comput. Graph. 15*, 6 (2009), 1283–1290.

[EBRI09] EVERTS M. H., BEKKER H., ROERDINK J. B. T. M., ISENBERG T.: Depth-dependent halos: Illustrative rendering of dense line data. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1299–1306.

[HMS95] HEIDRICH W., MCCOOL M., STEVENS J.: Interactive maximum projection volume rendering. In *VIS '95: Proceedings of the 6th conference on Visualization '95* (Washington, DC, USA, 1995), IEEE Computer Society, p. 11.

[KD98] KINDLMANN G., DURKIN J. W.: Semi-automatic generation of transfer functions for direct volume rendering. In *In IEEE Symposium on Volume Visualization* (1998), pp. 79–86.

[ME05] MORA B., EBERT D. S.: Low-complexity maximum intensity projection. *ACM Trans. Graph. 24*, 4 (2005), 1392–1416.

[MHG00] MROZ L., HAUSER H., GRÖLLER E.: Interactive high-quality maximum intensity projection. *Comput. Graph. Forum 19*, 3 (2000).

[Pfa02] PFAUTZ J. D.: *Depth perception in computer graphics*. Tech. Rep. UCAM-CL-TR-546, University of Cambridge, Computer Laboratory, Sept. 2002.

[PM08] PENNER E., MITCHELL R.: Isosurface ambient occlusion and soft shadows with filterable occlusion maps. In *Volume and Point-Based Graphics* (Aug. 2008), pp. 57–64.

[RKH08] ROPINSKI T., KASTEN J., HINRICHS K. H.: Efficient shadows for GPU-based volume raycasting. In *Proceedings of the 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2008)* (2008), pp. 17–24.

[RSH06] ROPINSKI T., STEINICKE F., HINRICHS K. H.: Visually supporting depth perception in angiography imaging. In *Proceedings of the 6th International Symposium on Smart Graphics (SG06)* (2006), vol. 4073 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 93–104.

[SE03] SVAKHINE N. A., EBERT D. S.: Interactive volume illustration and feature halos. In *Pacific Conference on Computer Graphics and Applications* (2003), pp. 347–354.

[SPN*98] SATO Y., PHD Y. S., NAKAJIMA S., SHIRAGA N., PHD S. T., MD R. K.: Local maximum intensity projection (lmip): A new rendering method for vascular visualization. *Journal of Computer Assisted Tomography 22*, 6 (1998), 912–917.

[WFG92] WANGER L. C., FERWERDA J. A., GREENBERG D. P.: Perceiving spatial relationships in computer-generated images. *IEEE Comput. Graph. Appl. 12*, 3 (1992), 44–51, 54–58.