Vision, Modeling, and Visualization (2010)

Cost-effective Feature Enhancement for Volume Datasets

J. Díaz[1], J. Marco[2] and P. Vázquez[1]

[1]MOVING Group, Universitat Politècnica de Catalunya
[2]GESSI Group, Universitat Politècnica de Catalunya

Abstract

Volume models often show high complexity. Local details and overall shape may sometimes be difficult to perceive. Unsharp masking techniques improve the perception of those small features by increasing the local contrast. In this paper we present a simple and fast method for feature enhancement based on 3D mipmaps. In contrast to other approaches, in addition to increasing luminance on the feature details, we also darken the valleys of the volume thus increasing local contrast and making neighboring details more visible. Our approach is fast and simple, with small memory requirements thanks to the use of 3D mipmaps. We also propose a color selection strategy, based on harmonic colors, that further enhances the salient features without abrupt or uncomfortable color changes.

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

1. Introduction

Contrast enhancement is a popular 2D image processing tool that improves the image appearance. Concretely, it helps users to understand complex models by emphasizing small features. Several experiments have proven that users prefer enhanced scenes over the non-enhanced ones [LGK06, IRS*09]. Unsharp masking increases high frequency components of a signal by adding back a filtered version of the signal in which high-frequency components are enhanced. In computer graphics, unsharp masking is usually achieved through image post-processing. Unfortunately, this approach may lead to popping artifacts or incoherence in videos due to the visibility changes [RGS09].

Volume rendering has a wide range of applications in different fields. The rendering algorithm transforms the input data into an image by applying the so-called transfer function, that transforms density values to a color and opacity. The definition of the adequate transfer function is a tedious process. In many cases, the complexity of the models makes the images to show a large number of fine overlapping structures. Therefore, small details are difficult to distinguish. As a consequence, feature enhancement is useful to interpret complex structures and detect local details.

In this paper we present a 3D feature enhancement technique whose main advantages are:

- The method is simple and fast, thus maintaining high framerates for complex models.
- The memory consumption is limited: Auxiliary data structures consist of a set of 3D mipmaps of the volume dataset, leading to small memory requirements as compared with the original model.
- We increase the local contrast not only by increasing the luminance to the salient features, but also by darkening the concave regions (see Figure 1).

We show an example of the results we achieve in Figure 1. We use 3D mipmaps [LW90] of the original dataset in order to precompute average density values at different resolutions. As a result, the feature enhancement performs in realtime, and the memory required is low. Therefore, we may apply our algorithm to large volumetric models (such as up to 512^3 voxels wide). Furthermore, we propose an extension for improving the perception of detected features: we select a different color but preserving harmony throughout the image.

The rest of the paper is organized as follows. Next section reviews previous work. Section 3 presents our technique. The results are discussed in Section 4. We evaluate both qualitatively and quantitatively the results of our method through two different user studies in Section 5. We provide some possible extensions in Section 6, and finally, Section 7 wraps up our work and points some lines for future research.

DOI: 10.2312/PE/VMV/VMV10/187-194

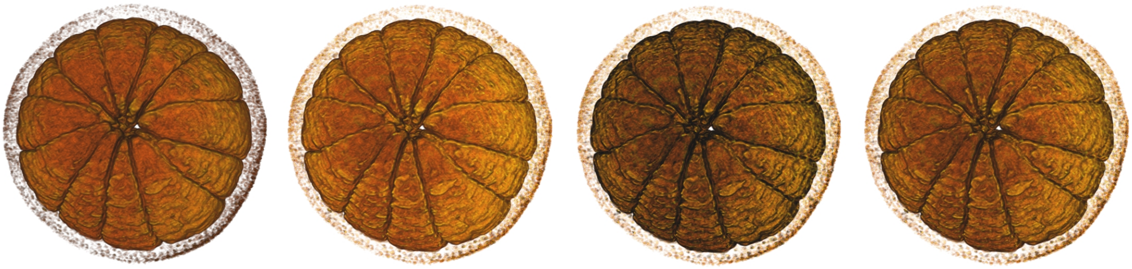

(a) Ray Casting
(b) Protrusion enhancement Mipmap 2, $\gamma = 1.0$
(c) Sinkage darkening Mipmap 2, $\gamma = 1.0$
(d) Enhanced view

Figure 1: *Feature enhancement is achieved by combining a luminance increase for the protruding zones (image* c*) and darkening for the sinking regions (image* d*). The final combination is shown in Figure* b.

2. Previous Work

Feature enhancement is important for the interpretation of complex structures and the detection of local details in volume visualization. Several methods for perception improvement of volumetric models have been presented in the last decade. Burns *et al.* [BHW*07] enhance important details by de-emphasizing non important materials. They achieve this by limiting their contribution of the output color so that materials with low importance are shaded less than those of high importance. Rusinkiewicz *et al.* [RBD06] introduced exaggerated shading. The method consists in changing lighting in a per-vertex basis. This can be used to reveal details at multiple scales.

Several approaches deal with the synthesis of ambient occlusion effects [RMSD*08, PM08, DYV08] in volumetric models. They improve depth perception, but do not necessarily enhance small details.

Nagy et al. [NSW02] combine line drawings and direct volume rendering techniques. Yuan and Chen [YC04] present a method for volume rendering images using silhouettes, ridge, and valley lines. Bruckner and Gröller [BG07] use volumetric halos for enhancing and highlighting structures of interest using a GPU-based direct volume rendering. Halos have also been used by Wenger et al. [WKZL04], Ebert and Rheingans [ER00]. Kindlman *et al.* [KWTM03] uses curvature information to define a transfer function that effectively communicates the curvature of a surface.

Other approaches use different ways of unsharp masking. Cignoni et al. [CST05] enhance the perception of discontinuous geometric features by applying unsharp masking on the normal field of a 3D surface. Luft *et al.* [LCD06] also apply unsharp masking for improving the depth perception of surface-based models. In order to do so, the information they use is the depth buffer, instead of the rendered image. Ritschel et al. [RGS09] perform 3D unsharp masking on the outgoing radiance field over the surfaces. This way the enhancement is coherent and popping artifacts are avoided if camera or object moves. Their approach has been proven useful with posterior studies [IRS*09]. All these approaches work with surface models. Volumetric unsharp masking has been previously addressed by Tao *et al.* [TLB*09]. They generate a 3D texture with the radiance values and iteratively smooth this texture. Then, unsharp masking is applied by comparing the values of the smoothed volume and the original radiance values. However, the use of auxiliary 3D textures limits the applicability to relatively small models. Our approach also deals with volumetric models, but, instead of smoothing the shading, we smooth the model by using a set of 3D textures containing various levels of 3D Mipmaps of the dataset. Moreover, we not only amplify local contrast by increasing the luminance on protuberance regions, but also darken the deeper regions. This increases the features enhanced by our algorithm.

3. Volumetric Feature Enhancement Using Mipmaps

3.1. Introduction

In computer graphics, a Mipmap (also MIP map) is a hierarchy of 2D textures that are built from a source texture intended to increase texturing speed. This hierarchy of textures is built by iteratively halving the initial texture size. Each mipmap level is half the width and height of the previous one, thus reducing storage by 4. OpenGL has hardware support for mipmapping, which means that the rendering process will use the most adequate mipmap level that improves speed but has little or no quality loss.

3D Mipmaps are the 3D counterpart of the classical mipmaps. They have been used previously to accelerate volume rendering [LW90]. In our case, we apply the mipmap reduction scheme to 3D textures, but, in contrast to the classical approach, we do not necessarily divide by 2 each dimension. Opposite to this, we reduce the original size by any integer value in every dimension. The construction process is shown in Figure 2, where the mipmaps 2 to 4 are built. Level 4 would be the second mipmap level in the classical case. However, we break this dependency because our application will load a set of mipmaps and use only the necessary

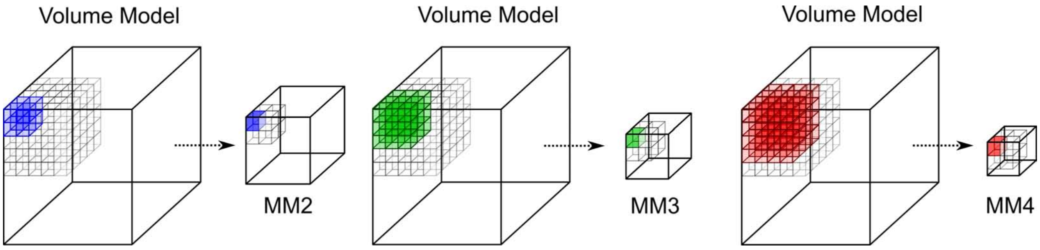


Figure 2: *Mipmap construction process.*

ones. As shown later, we could use only level 3 for darkening and level 4 or 5 (or even 16) for the feature emphasizing. Besides, we do not store the whole hierarchy, since a single level is usually enough.

3.2. Algorithm Overview

In this section we present our algorithm for feature enhancement. Our algorithm has been designed for improving the rendering of volumetric models based on a ray-casting scheme [RGW*03].

First, we load the dataset into the CPU. Then, we build a series of 3D mipmaps of the model. All this information is passed to the GPU in 3D textures. Then, the ray-casting algorithm is slightly modified in order to test, for each sample, if the density of the sample point is different from the density of a cubic region around the point. If the values are different, we modify the shading in order to emphasize the detected feature. The overall rendering algorithm is depicted in Figure 3.

The final image is composed by combining two enhancements: protrusion enhancement, achieved through unsharp masking, and sinkage darkening, computed similarly. Currently, we use a linear blend of the two effects, though other approaches could be investigated. The GPU ray-casting algorithm is only slightly modified, as shown in Algorithm 1. We present next how to calculate these two effects.

3.3. Protrusion emphasizing

In order to detect salient features, we use a similar procedure from the literature: unsharp masking. The difference is how we calculate the smoothed version of the signal. Given a viewpoint, Tao *et al.* [TLB*09] generate a 3D texture containing the radiance values. Then, they iteratively smooth this 3D texture by using a ping-pong approach. Instead, what we do is to evaluate, for each sample of the ray, the actual difference between the Phong shaded color and the shading obtained by evaluating the Phong shading to the average density of a cubic region. This value is obtained from the Mipmap structure, that is accessed, like the model texture, using hardware trilinear interpolation.

Algorithm 1 Algorithm that performs feature emphasis.

```
...
vec3 position = calculateOriginRay(gl_FragCoord);
initializations, ...
while (! endrayCasting(...)) do
    float density = texture3D( Volume, position ).a;
    {Query the Mipmap textures}
    float densLight = texture3D( MMLight, position ).a;
    float densDark = texture3D( MMDark, position ).a;

    material = evaluateTF(density)
    col = calculatePhongShading(material, position)
    {Feature emphasis}
    col = emphasize(col, density, densLight, densDark)

    composite colors
    update position
end while
...
```

The color modification is performed in CIELab format. Once the sample point is evaluated and shaded, the resulting color is transformed to CIELab yielding a triplet $[L,a,b]$. We also sample the mipmap and shade the average density using the Phong shading as usual. The resulting color is another triplet, also in CIELab format: $[L_{MipMap}, a_{MipMap}, b_{MipMap}]$. The unsharp color in CIELab ($U(C)$) is built using the following formula:

$$U(C) = [L + \gamma(L - L_{MipMap}), ka, kb), \qquad (1)$$

where γ is the weight we give to the difference. If we use a high value, the image results in too overexposed, while a too low value does not emphasize the existing features. Like in [TLB*09], we add the k factor in order to guarantee that the saturation is preserved. The value k is computed as:

$$k = (L + \gamma(L - L_{MipMap}))/L. \qquad (2)$$

The resulting color takes as opacity value the one of the original ray-casting. A second advantage of this approach

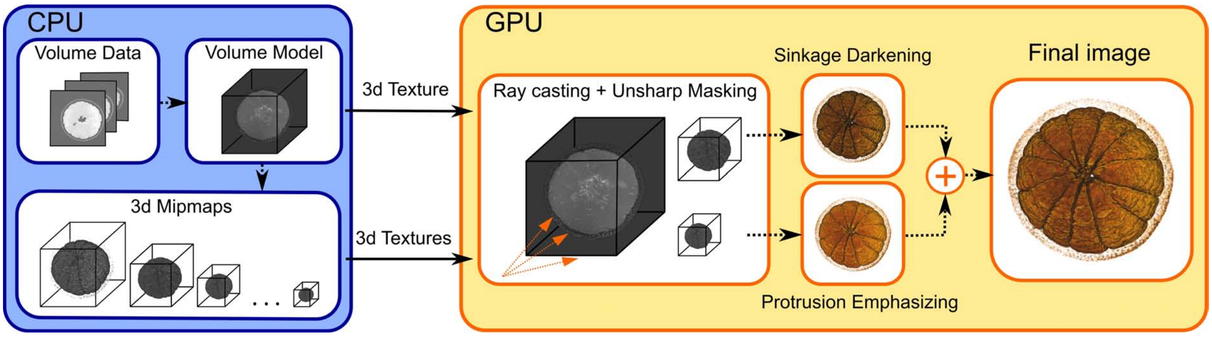
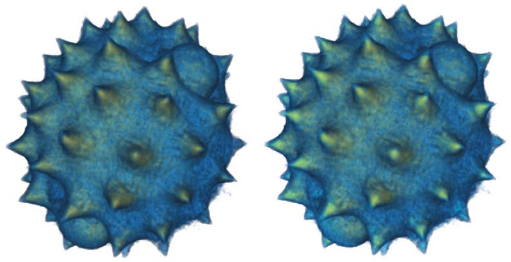


Figure 3: *Overview of our rendering algorithm. Once the dataset has been loaded, we build a series of 3D Mipmaps from it. Both the model and the mipmaps are passed to the GPU, that uses them to generate the enhanced image. Both the darkening and the emphasizing is achieved by unsharp masking the accumulated color of the Ray-Casting.*



(a) Ray-Casting (b) Unsharped MM8, $\gamma = 0.4$

Figure 4: *Feature enhancement of a volumetric model.*

is that it does not change the hue value, and therefore, although emphasizing the features, the overall look does not change. The emphasis we apply depends on two parameters, the value γ, that we may change through a slider, and the level of the mipmap, selected with another slider, thus making transparent to the user the fact that several mipmaps are available. Larger mipmaps do erode the dataset to a larger extent, and therefore, the difference $L - L_{MipMap}$ is higher. Thus, the emphasis is more important. However, for thin structures this might lead to irregularly placed emphasis, often shown as light and dark stripes.

We show the original Ray-casting of the pollen model side to side with the enhanced version using the mipmap of level 8 in Figure 4. For the protrusion enhancement, we usually use mipmaps of level 2 (each sample value is built using 8 neighbors (2^3) in the original volume), 3 (3^3) to 5 (5^3 neighbors). If the features we want to emphasize are large (such as in the pollen grain), larger mipmaps can be used to better enhance the salient geometry. The models we have tested are relatively large, starting with roughly 256^3, up to 512^3. For small models, such as 64^3, probably small mipmaps are better, but also the amount of detectable details.

3.4. Sinkage darkening

In order to further improve the overall appearance, we also darken the low regions. The algorithm we use is equivalent to the unsharp masking presented above, but we change the reference color by using the color of the Transfer Function (L_{TF}, a_{TF}, b_{TF}), without the illumination calculation. Thus, the darkening color $(D(C_{TF}))$ is computed as:

$$D(C_{TF}) = [L + \gamma(L - L_{TF_{MipMap}}), ka, kb). \qquad (3)$$

This produces the interesting effect of darkening the sinkage regions. Final color is composited by averaging both effects: $finalCol = 0.5 * (U(C) + D(C_{TF}))$.

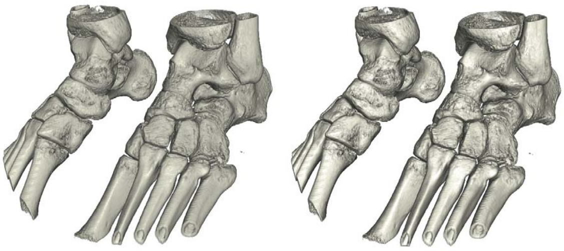
The opacity of the computed color is the one of the Phong shading of the sample point. We may see the result of applying this method in Figure 1, where we show the final result (top right), but also the two effects (protuberance emphasizing and sinkage darkening) as they behave separately.

For the darkening, mipmaps of level 2 or 3 are used as larger mipmaps produce extra, non pleasant shadows.

4. Results

We have applied our method to several models. The usual configuration we have by default is mipmap 2 or 3 for the darkening, and mipmap 2 to 5 for the lightening. However, there are cases where the most suitable mipmap required may be quite large, such as in Figure 5. For the correct enhancement of the model, in this case we use a mipmap of level 16 for the light part, and level 2 for the darkening.

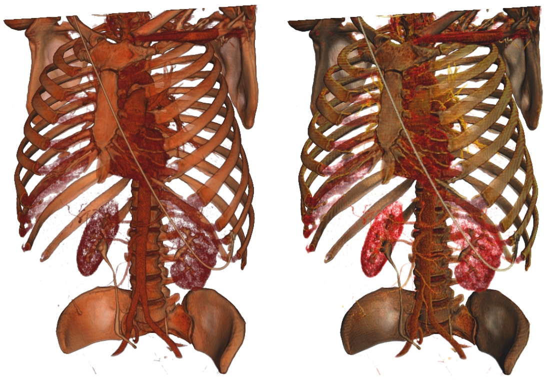
Our approach also works for semi-transparent models. We show how fine details are improved in the presence of opaque and translucent materials in Figure 6. This image shows how small details are emphasized due to the luminance change and concave regions are better perceived due to the darkening.



(a) Ray-casting

(b) Enhanced view
Light: MM16, $\gamma = 0.1$
Dark: MM2, $\gamma = 1.1$

Figure 5: *Feature enhancement of the feet model. Left image shows the original ray-casted model. The enhanced view uses a level 16 mipmap for the protrusion enhancement* ($\gamma = 0.1$) *and a level 2 mipmap with* $\gamma = 1.1$ *for darkening.*



(a) Ray-casting

(b) light: MM3, $\gamma = 3.0$
dark: MM2, $\gamma = 3.7$

Figure 6: *Enhancement of a complex model with fine details and semi-transparent structures. Note how small details such as vessels are more visible in right image. Ribs are also more distinguishable thanks to the luminance change. The darkening feature makes bones curvature more visible.*

We have tested with models of up to 512^3, and the mipmaps we generated go from level 2 (each cell is computed by averaging 2^3 cubes into one) to level 16 ($16^3 \rightarrow 1$). As the memory requirements are relatively low, we may afford to store several mipmap levels. The time required for building the mipmaps is also relatively low, when compared with the loading time of the 3D model. We show the construction times in Table 1. Compared to the time required to load a model, usually in the range of several seconds for large models, the time needed for the mipmap generation is negligible. The results have been calculated in a Intel Core2 DUO CPU running at 3.0 GHz equipped with a nVidia GeForce GTX 280 GPU with 1GB of RAM memory.

Model	**Resolution 3D model**	**Loading time**	**Mip-map building**
Pollen	$192 \times 180 \times 168$	483	192
Orange	$256 \times 256 \times 64$	260	142
Engine	$256 \times 256 \times 256$	623	562
Abdomen	$512 \times 512 \times 171$	12376	1530
Feet	$512 \times 512 \times 282$	14246	2513
Body	$512 \times 512 \times 512$	20694	4559

Table 1: *Construction times of the mipmaps (levels 2, 3, 4, 8, 16, and 32) expressed in milliseconds. Note how the mipmap generation is roughly one order of magnitude faster than the loading of the original model.*

The framerates obtained with the models we tested are shown in Table 2. If we use mipmaps of the same level for both the protrusion enhancement and sinkage darkening, the framerates only decrease up to a 20%. When different mipmaps are used, texture accesses have less coherency, and therefore the framerate decreases more. In this case up to a 50%, while in Tao *et al.* [TLB*09] fps decay is up to 70%. In most cases we will prefer the use of two different mipmaps, because the quality is better with little user tuning of the γ parameter and mipmap level. Pollen grain framerates are higher, but result in 60 fps due to the vertical sync.

5. User Study

In order to evaluate our technique, we carried two different studies, which evaluate qualitatively and quantitatively our results. Therefore, users had to solve 2 different tasks.

For the first task, we selected a set of six models, and rendered three different images using direct volume rendering, the method by Tao *et al.* [TLB*09], and our method (see Figure 8). The three images were randomly ordered for each model and users had to answer which image was better to perceive the features of the model. This task was performed by a set of 42 people and their answers are shown in Figure 7. For the second task, we built a set of synthetic datasets where we randomly placed a number of cylinders over a plane, and in a semi-transparent cube. Users had to count the number of cylinders, and we measured both the correctness and the time devoted to each counting. Images where rendered using DVR and our feature enhancement technique (see Figure 9). A set of 20 images were displayed randomly. A group of 16 people took part in this task. In enhanced images, users counted correctly the number of cylinders, whereas when using DVR, they wrongly answered in 9.3% of the cases. Since the approach by Tao *et al.* [TLB*09] obtain qualitatively similar results, we used here only our enhanced images. We analyzed the results and computed, for the first task, the 95% Confidence Intervals (CI) for the sample proportion of answers DVR (10%), Tao *et al.* (18%), and our technique (72%) of a survey where $n = 252$. The 95% CI for the mean

Model	**Ray Casting**	**Enhanced 1 MM same Mipmap**	**Impact 1 MM same Mipmap**	**Enhanced 2 MM different Mipmaps**	**Impact 2 MM different Mipmaps**
Daisy Pollen Grain	59.5	59.50	0.08%	42.5	28.6%
Orange	59.47	48.23	18.9%	27.88	53.1%
Engine	58.98	52.43	11.1%	30.83	47.7%
Abdomen	28.38	25.14	11.4%	15.23	46.3%
Feet	22.52	18.01	20.0%	11.84	47.4%
Body	12.87	11.23	12.7%	8.00	37.8%

Table 2: *Framerates yielded with different models. Timings are compared with the original Ray Casting and the feature improvement algorithm. The GPU is a nVidia GeForce GTX 280 with 1GB of memory. Note the difference in timings if we use one single mipmap (1MM) for both feature emphasis and sinkage darkening or two different maps (2MM), one for each effect.*

of the times have been also computed for the second task, where $n = 160$. The results obtained, using the usual estimation formulae, namely $\hat{p} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$ are shown in tables 3 and 4.

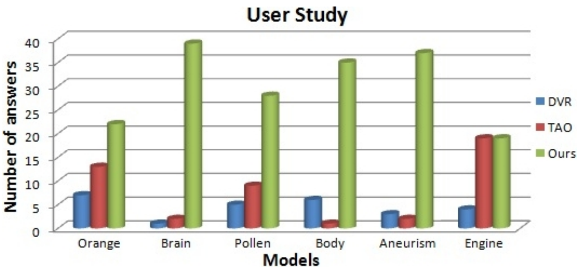
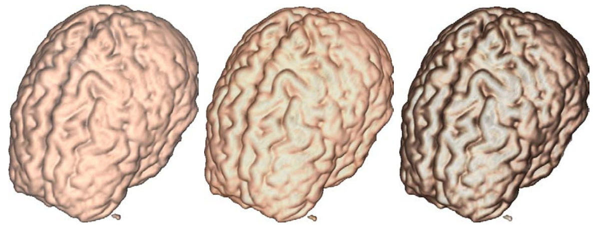


Figure 7: *Number of answers to our user study for the six models analyzed.*



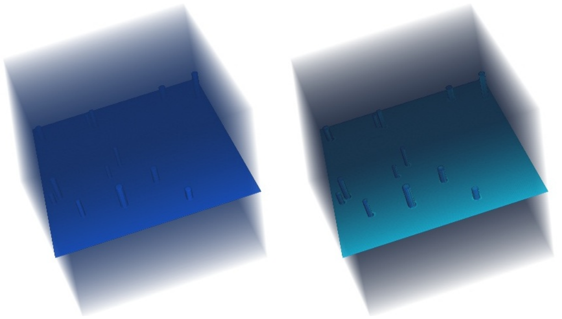
(a) Ray Casting (b) Tao's method (c) Our technique

Figure 8: *Example of images shown in task 1.*

95%	**DVR**	**Tao's method**	**Our technique**
Lower bound	6.29	13.25	66.45
Upper bound	13.70	22.74	77.54

Table 3: 95% *confidence intervals for answers in task 1.*

In conclusion, we can state that our rendering method clearly enhances the main features of the data with respect to DVR and provides results at least as good as the method by Tao *et al.* We used $\gamma = 1.0$ for their method and MipMap levels 2 and 4 for lightening and darkening, respectively, both with a $\gamma = 1.0$. Surely, more fine parameter tweaking would improve the results obtained by Tao *et al.*'s method. However, our algorithm also darkens low regions, an effect that is difficult to achieve with their approach. Moreover, it was also proven that our approach helps the user to detect features more accurately than Phong shading.



(a) DVR (b) Feature-enhanced image

Figure 9: *Example of synthetic dataset used in task 2. The users had to count the number of cylinders in the volume.*

95%	**DVR**	**Our technique**
Lower bound	7.07	6.69
Upper bound	7.96	7.62

Table 4: 95% *confidence intervals for the mean of times (sec) of task 2.*

6. Extensions and Discussion

In this Section we discuss other different approaches we tested throughout the development of our method. Furthermore, we also propose an extension for further emphasizing local details, harmonized color change, introduced next.

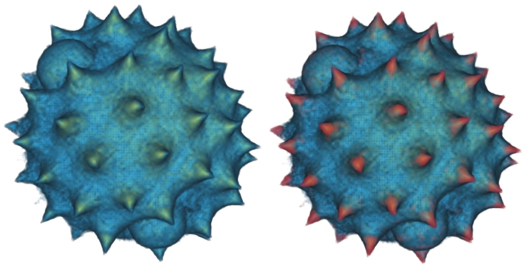


Figure 10: *Harmonic color-based feature emphasis for the pollen grain model.*

6.1. Harmonized color change

So far, in order to emphasize local features we have only modified the color by changing luminance and maintaining saturation. Illustrators often use different colors to stress important details. We propose here the use of harmonic colors in order to substitute the color of the region of interest. The use of harmonic colors [COSG*06] ensures that the overall impression will not change in a sudden or unpleasant way. In order to apply a harmonic color, we first select a color from the original rendering and determine its harmonic color. This can be done in different ways. In our case, we select as the harmonic color the complementary one in the Hue color circle. Then, this color is transformed to CIELab. For each ray sampling, if the density detected at this point is significantly larger than the one of the mipmap, the original color is changed with the harmonic color and emphasized using equation 3. We show how harmonic color feature emphasis behaves for the pollen grain model in Figure 10. Note how details are enhanced with a different though pleasant color change. In this case, we use gamma values of 0.1 or 0.2, otherwise, the color may tend to the original one.

6.2. Full sampling vs sampling at isosurface level

Being the details we want to enhance placed near the surfaces the transfer function reveals, one may think if it would be enough to unsharp mask only at the surface level. As shown in Figure 11, this strategy does not yield very good results. In this image, both enhancements use the same mipmap and gamma values. Although we might probably obtain similar quality with surface level unsharp masking, this would require appropriate fine tuning. Note how the quality of the features extracted is higher for the case of the full sampling (center). Moreover, the gainings obtained by sampling the mipmap only at surface level are roughly 2% to 10% in speed when the same mipmap is used for protrusion and sinkage enhancement. When using different mipmaps for each effect, the impact is roughly the same than with the previous method. Another possible strategy would be subsampling the mipmap, that is, accessing the mipmap only one out of two sampling points. This yields even poorer results, as the surface of opaque structures is not detected properly and the unsharp masking produces irregular patterns.

6.3. Automatic mip map selection

Although highly subjective, we have started working on the automatic selection of the correct mip map level. Our approach deals with an image analysis on the amount of information that is provided with the different mip map levels when the transfer function changes. We believe that an image-based analysis could be suitable. However, the preliminary results only work for some models.

7. Conclusions and Future Work

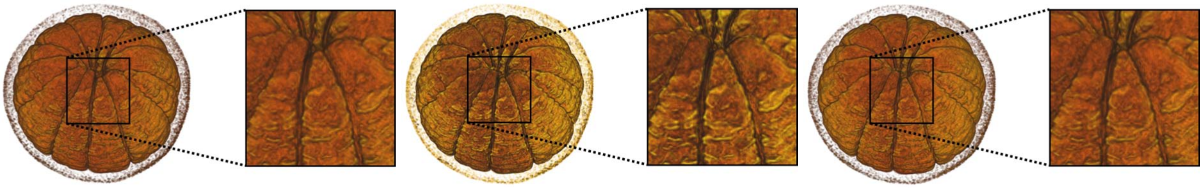
In this paper we have proposed a new approach for fast feature emphasis based on 3D mipmaps. The main advantages of our method are low memory requirements and relative low impact on the rendering times. Small details can be emphasized with low-level mipmaps, but large images may require large ones. However, the overall memory required for a set of mipmaps is not large. The largest one (level 2) requires $1/8$th of the original data. Thus, multiple mipmaps can be loaded at the same time. The lower resolution of our mipmap approach is compensated with the darkening feature, which also adds details on the low parts of the model. We have also proposed the use of harmonic colors in order to further emphasize the local details, as shown in Figure 10, although for some models (such as medical models) it might be not adequate, we believe it may result in nice-looking illustrative renderings. Tao *et al.* [TLB*09] also perform feature emphasis on a volumetric basis. In contrast to their approach, we do not smooth the resulting shading of the volume model, but we smooth the dataset by building a set of mipmaps that are later used for shading. This results in a single pass rendering algorithm. Moreover, our method does change the shading by increasing the luminance of the salient features, and darkens the sinkage regions at the same time, thus, augmenting the local contrast, as compared with regular unsharp masking. We performed both a quality and quantitative test and the results assess the utility of our method. In future we want to fully automate the mip map level selection based on an image analysis of the detected features.

8. Acknowledgments

Thanks to A. Chica and T. Ropinski for their help and discussion. Thanks to the reviewers for their valuable comments. Supported by TIN2007-67982-C02-01 of the Spanish Government. J. Díaz is also supported by FPI BES-2008-002404.

References

[BG07] BRUCKNER S., GRÖLLER E.: Enhancing depth-perception with flexible volumetric halos. *IEEE Transac-*



(a) Ray Casting (b) Sampling at each point (c) Sampling at isosurface level

Figure 11: *Comparison of different sampling strategies. (a) Regular Ray Casting. (b) The presented method, mipmap is queried each time the ray intersects the volume, we query the mipmap. (c) Mipmap is only queried when reaching the surface of the model. This is implemented by setting a threshold opacity value, once this value is reached, we start sampling the mipmap.*

tions Visualization and Computer Graphics 13, 6 (2007), 1344–1351. 2

[BHW*07] Burns M., Haidacher M., Wein W., Viola I., Gröller E.: Feature emphasis and contextual cutaways for multimodal medical visualization. In *EG/IEEE VGTC Symposium on Visualization* (May 2007), IEEE, pp. 275–282. 2

[COSG*06] Cohen-Or D., Sorkine O., Gal R., Leyvand T., Xu Y.-Q.: Color harmonization. *ACM Trans. Graph. (Proc. SIGGRAPH) 25*, 3 (2006), 624–630. 7

[CST05] Cignoni P., Scopigno R., Tarini M.: A simple normal enhancement technique for interactive nonphotorealistic renderings. *Computer & Graphics 29*, 1 (feb 2005), 125–133. 2

[DYV08] Díaz J., Yela H., Vázquez P.-P.: Vicinity occlusion maps: Enhanced depth perception of volumetric models. In *Computer Graphics International* (2008), pp. 56–63. 2

[ER00] Ebert D., Rheingans P.: Volume illustration: Non-photorealistic rendering of volume models. In *VIS'00: Proceedings of IEEE Visualization 2000* (2000), Ertl T., Hamann B., Varshney A., (Eds.), pp. 195–202. 2

[IRS*09] Ihrke M., Ritschel T., Smith K., Grosch T., Myszkowski K., Seidel H.-P.: A perceptual evaluation of 3d unsharp masking. In *Proceedings Human Vision and Electronic Imaging XIII* (2009). 1, 2

[KWTM03] Kindlmann G., Whitaker R., Tasdizen T., Möller T.: Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of IEEE Visualization* (2003), pp. 513–520. 2

[LCD06] Luft T., Colditz C., Deussen O.: Image enhancement by unsharp masking the depth buffer. *ACM Trans. Graph. 25*, 3 (2006), 1206–1213. 2

[LGK06] Lin W., Gai Y., Kassim A.: Perceptual impact of edge sharpness in images. *Vision, Image and Signal Processing, IEE Proceedings - 153*, 2 (2006), 215–223. 1

[LW90] Levoy M., Whitaker R.: Gaze-directed volume rendering. In *SI3D '90: Proceedings of the 1990 symposium on Interactive 3D graphics* (New York, NY, USA, 1990), ACM, pp. 217–223. 1, 2

[NSW02] Nagy Z., Schneider J., Westermann R.: Interactive volume illustration. In *Vision, Modeling and Visualization 2002* (Nov. 2002), pp. 497–504. 2

[PM08] Penner E., Mitchell R.: Isosurface ambient occlusion and soft shadows with filterable occlusion maps. In *Volume and Point-Based Graphics* (2008), pp. 57–64. 2

[RBD06] Rusinkiewicz S., Burns M., DeCarlo D.: Exaggerated shading for depicting shape and detail. *ACM Trans. Graph. (Proc. SIGGRAPH) 25*, 3 (July 2006). 2

[RGS09] Ritschel T., Grosch T., Seidel H.-P.: Approximating dynamic global illumination in image space. In *I3D '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games* (2009), ACM, pp. 75–82. 1, 2

[RGW*03] Roettger S., Guthe S., Weiskopf D., Ertl T., Strasser W.: Smart hardware-accelerated volume rendering. In *Proceedings of EG/IEEE TCVG Symposium on Visualization* (2003), pp. 231–238. 3

[RMSD*08] Ropinski T., Meyer-Spradow J., Diepenbrock S., Mensmann J., Hinrichs K. H.: Interactive volume rendering with dynamic ambient occlusion and color bleeding. *Computer Graphics Forum (Eurographics 2008) 27*, 2 (2008), 567–576. 2

[TLB*09] Tao Y., Lin H., Bao H., Dong F., Clapworthy G.: Feature enhancement by volumetric unsharp masking. *Vis. Comput. 25*, 5-7 (2009), 581–588. 2, 3, 5, 7

[WKZL04] Wenger A., Keefe D. F., Zhang S., Laidlaw D. H.: Interactive volume rendering of thin thread structures within multivalued scientific data sets. *IEEE Transactions on Visualization and Computer Graphics 10* (2004), 664–672. 2

[YC04] Yuan X., Chen B.: Illustrating Surfaces in Volume. In *Proceedings of Joint IEEE/EG Symposium on Visualization* (2004), pp. 9–16, 337. 2