2IMV20 (2018-GS2) Visualization

2018 Semester A Quartile 2

Volume rendering

Authors: (Assignment groups 100)

Sergio Jose Ruiz Sainz 1379550

Dagoberto Jose Herrera Murillo 1379291

Tutors:

Dr. Michel Westenberg

MSc Alberto Corvo

MSc Humberto Garcia Caballero

December 2018

Contents

1. Introduction

Volume visualization is related to volume data modeling, manipulation, and rendering. It enables the extraction of meaningful information from a massive amount of complex volumetric data through interactive graphics. In the present assignment, we are interested in volume rendering. It refers to a set of techniques taking a 3D volume and projecting it to a 2D representation. We explore the application of volume rendering methods to the display of volumetric data.

The following sections describe the implementation of Trilinear interpolation, Maximum Intensity Projection (MIP), Compositing, and 2-D transfer functions. Subsequently, the techniques are applied to simulation data of deep-water asteroids. Finally, we compare the strengths and weaknesses of each of the techniques.

2. Ray-casting

Based in the method slicer() in the class RaycastRenderer, we were asked to implement the following functionalities: 1) Tri-linear interpolation of samples along the viewing ray, 2) Maximum Intensity Projection (MIP), and 3) Compositing ray functions. Besides some considerations on 4) Responsiveness are included. These elements are explained in the following sections.

2.1 Trilinear interpolation

Interpolation is one of the components required for any ray-casting implementation. Trilinear interpolation estimates the value at a specific location considering the 8 voxels surrounding the interpolation point. In Figure 1, X represents the interpolation point; $X_{0...7}$ are the 8 voxels surrounding X; α, β, and γ are the proportions [0,1] of the sample position in *x*, *y*, and *z*.
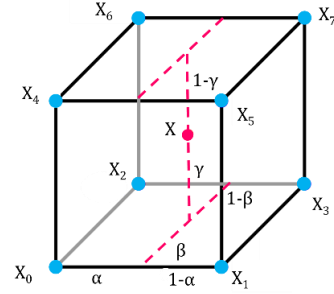

Figure 1. Notation for the trilinear interpolation.

The trilinear interpolation equation[1] is as follows:

$$S_x = (1-\alpha)(1-\beta)(1-\gamma)S_{x0} + \alpha(1-\beta)(1-\gamma)S_{x1}$$
$$+ (1-\alpha)\beta(1-\gamma)S_{x2} + \alpha\beta(1-\gamma)S_{x3}$$
$$+ (1-\alpha)(1-\beta)\gamma S_{x4} + \alpha(1-\beta)\gamma S_{x5}$$
$$+ (1-\alpha)\beta\gamma S_{x6} + \alpha\beta\gamma S_{x7}$$

As discussed by Ray et al. (1999), the fact that 8 voxel values and several multiplications are required to calculate each interpolation point implies that for a very large dataset, billions of multiplications are required for each projection. In consequence, interactive projections demand considerable computational power. By extension, higher order interpolation can improve image quality, but it has a high computational cost. Nearest-neighbor interpolation is a cheaper alternative, although its quality is worse.

The trilinear interpolation was implemented in the method RaycastRenderer.getVoxel().

2.2 Maximum intensity projection (MIP)

This method projects the voxel with maximum intensity falling in the path of the viewing ray. MIP was implemented by extending the RaycastRenderer.slicer() method to create a new

---

[1] This implementation was based in the lecture notes Spatial Data I (Slides 6-7)

method called RaycastRenderer.mip() which add a third loop that handles the depth of the vector (z-axis). The length of the ray in each direction can be represented by the expression $\frac{1}{2}\sqrt{V_x + V_y + V_z}$, where $V_x, V_y,$ and V_z are the lengths of the volume on the x, y, and z axes, respectively. This loop crosses the ray and captures the highest value. as the following equation[2] shows:

$$I(p) = \max_{t} s_t$$

To illustrate the result, Figure 2 shows the MIP implementation applied to a tomato.
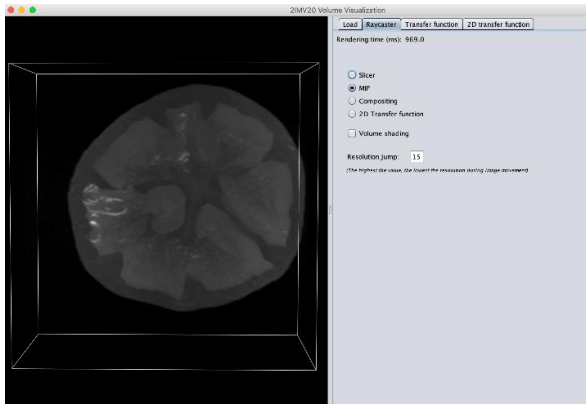
Figure 2. Result of Maximum intensity projection (MIP)

2.3 Compositing

Compositing is a ray casting technique which is based on the accumulation of colors weighted by opacity. For each voxel value, the color and opacity contributions are extracted from locations along a ray to produce a final color for display. The new composite color is computed by multiplying the voxel color and opacity and adding to the multiplication of the previous composite color and opacity. The corresponding volume rendering equation[3] is as follows:

$$I(p) = \sum_{i=0}^{n-1} c_i \prod_{j=i+1}^{n-1} (1 - \tau_j)$$

where c_i represent **color** and τ_j represents **opacit**y of the voxel

Composition builds the image, typically from back-to-front, voxel after voxel, starting with the farthest point from the viewer to define the appropriate color and opacity values. In the back-to-front formula shown below, it can be seen how the color value of a particular voxel (C_i) is determined relative to its predecessor (C_{i-1}) considering the value of the opacity (τ_j).

$$C_i = \tau_i c_i + (1 - \tau_i) C_{i-1}$$

Compositing was implemented by extending the RaycastRenderer.slicer() method to create a new method RaycastRenderer.compositing(). To illustrate the result, Figure 3 shows the compositing implementation applied to a tomato.
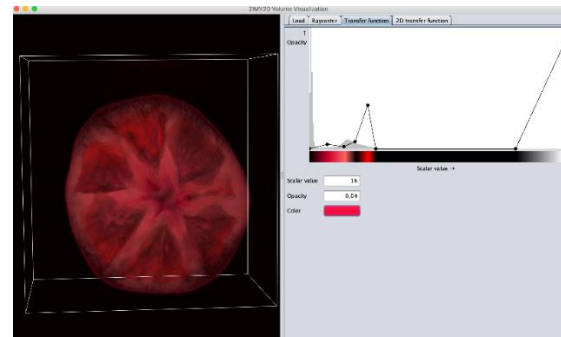

Figure 3. Result of Compositing

2.4 Responsiveness

Raycasting is a slow process. Therefore, we reduce the resolution in all the implemented methods (except slicer) while the user rotates the figure to increase the responsiveness of the application. The lower resolution is achieved by computing values for only some of the voxels

---

[2] This implementation was based in the lecture notes Spatial Data I (Slide 8)

[3] This implementation was based in the lecture notes Spatial Data I (Slide 31)

when the interactiveMode is on. As shown in Figure 4, we add to the interface a text box called "Resolution Jumps" where the user can manipulate the resolution by increasing or decreasing the frequency of resampling points. In the same figure, you can see a low-resolution image while the user rotates the object.
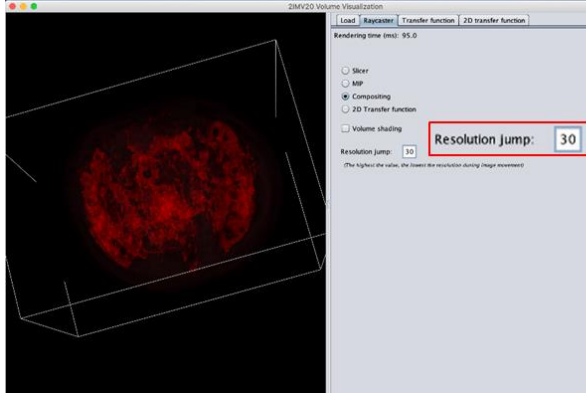

Figure 4. Resolution jumps to adjust the resolution

3. 2-D transfer functions

1D transfer functions do not allow to highlight all features of interest in volume data. According to Levoy (1988), 2D transfer functions offer greater flexibility in boundary visualization. In this section, we introduce methods that incorporate gradient-based opacity weighting and illumination in the ray-caster to gain additional visualization capabilities.

3.1 Gradient-based opacity

Compared to simple rendering, gradient-based opacity can identify boundaries quantifying the presence of a surface by the magnitude of the corresponding gradient. Levoy (1988) proposed the following function[4] to approximate the gradient vector for a voxel with coordinates $x, y,$ and z; this value eventually will be used by the gradient-intensity histogram.

$$\nabla s \approx \frac{1}{2} \begin{pmatrix} s(i+1,j,k) - s(i-1,i,j) \\ s(i,j+1,k) - s(i,j-1,k) \\ s(i,j,k+1) - s(i,j,k-1) \end{pmatrix}$$

The gradient equation was written in the GradientVolume.compute() method.

Another component of Levoy's method is the calculation of opacity, which is represented in the following formula[5]:

$$\alpha(\mathbf{x_i}) = \alpha_v \begin{cases} 1 & \text{if } |\nabla f(\mathbf{x_i})| = 0 \text{ and } f(\mathbf{x_i}) = f_v \\ 1 - \frac{1}{r} \left| \frac{f_v - f(\mathbf{x_i})}{|\nabla f(\mathbf{x_i})|} \right| & \text{if } |\nabla f(\mathbf{x_i})| > 0 \text{ and} \\ & f(\mathbf{x_i}) - r\,|\nabla f(\mathbf{x_i})| \leq f_v \leq f(\mathbf{x_i}) + r\,|\nabla f(\mathbf{x_i})| \\ 0 & \text{otherwise} \end{cases}$$

Where r represents the thickness in the voxels of the transition area. This formula can be found in the RaycastRenderer.twodimension() method.

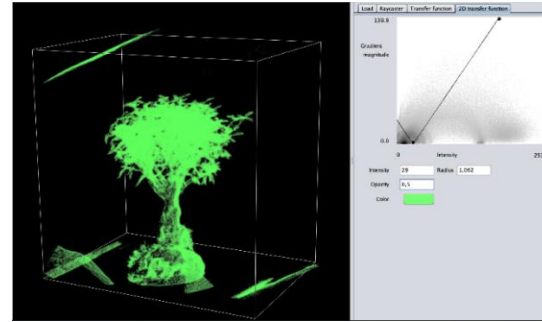To illustrate the result, Figure 5 shows the gradient-based opacity implementation applied to a bonsai.

Figure 5. Result of Gradient-Based Opacity

3.2 Illumination model

The shading model proposed by Phong (1975) improves the quality and realism of the images by differentially illuminating the various regions of the object according to their position with respect to the

---

[4] This implementation was based in the lecture notes Spatial Data I (Slide 24)

[5] This implementation was based in the lecture notes Spatial Data II (Slide 15)

observer. The Phong model uses the following formula[6] to calculate the shading coefficient (a value between 0 and 1):

$$I = I_a k_{ambient} + I_l k_{diff}(\mathbf{L} \cdot \mathbf{N}) + I_l k_{spec}(\mathbf{V} \cdot \mathbf{R})^\alpha$$

Where L is the normalized vector in direction of light source, N is the surface normal, R is the normalized vector in direction of specular light reflection, V is the normalized vector in direction of observer, $K_{ambient}$ ambient reflection coefficient, K_{diff} is the diffuse reflection coefficient, K_{spec} is the specular reflection coefficient, and α is an exponent used to estimate the highlight. Because we start from the assumption that light comes from the observer, then H is equal to L and V. The formula can be found within an if-else condition within the RaycastRenderer.twodimension() method which considers the Boolean parameter called illuminated.

To illustrate the result, Figure 6 shows the illumination implementation applied to a piggy bank.
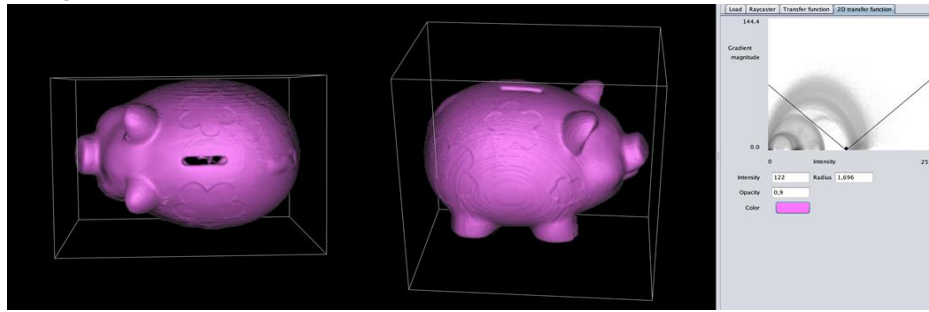
Figure 6. Result of the Illumination Model

4. Data exploration

This section shows the application of the techniques previously explained to the context of the Scientific Visualization Challenge 2018, which features simulation data of deep-water asteroids. The dataset contains several time points of the impact and each point has the following scalar fields associated: PRS (pressure in microbars), TEV (temperature in electron volt), V02 (volume fraction water), and V03 (volume fraction of asteroid).

The exploration was guided by the following questions:

1. What is the performance of the techniques studied to represent each of the available attributes?

3. What is the history behind a collision based on the information provided by the visualization of the four attributes?
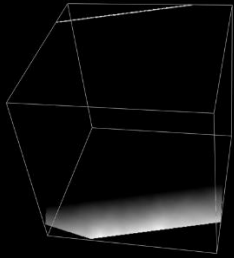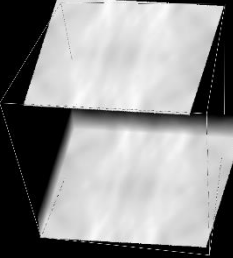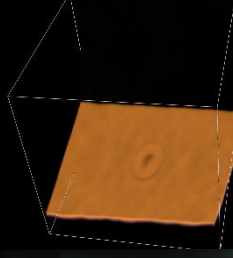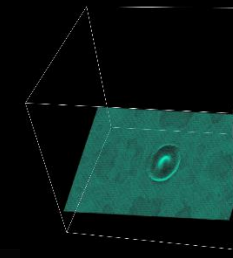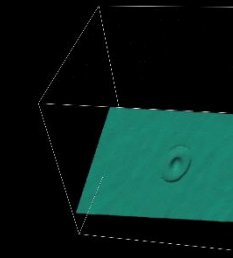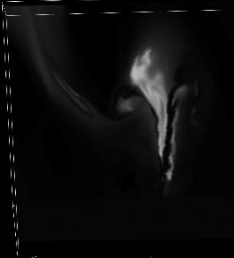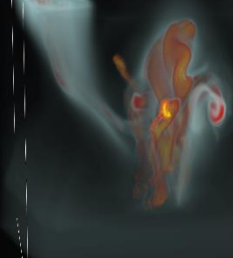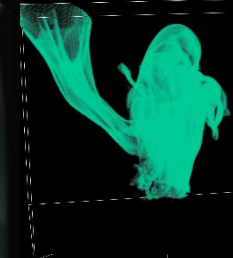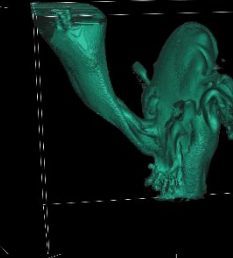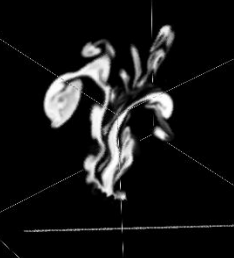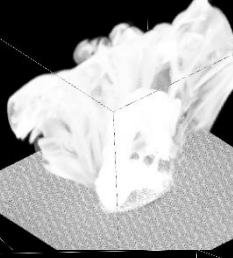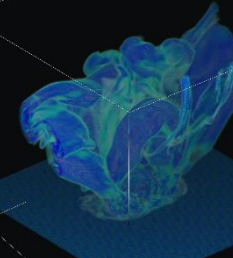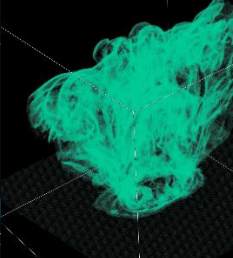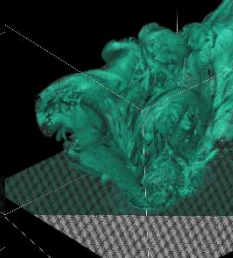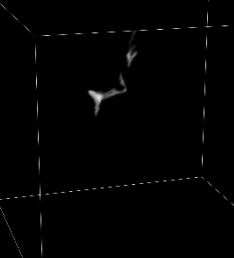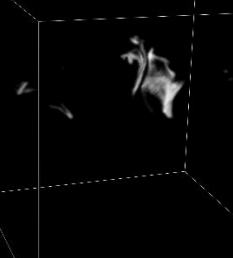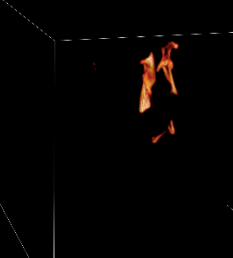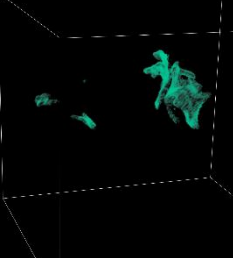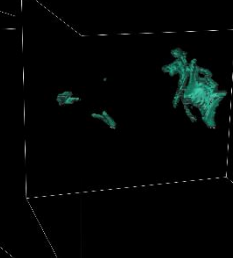
4. What are the main characteristics of the temporal evolution of an asteroid collision?

5. What are the most notorious differences between the collision of the asteroid (yA31) and the asteroid (yB31)?

To answer the first question, we decided to explore the four scalars for the first time point (39476) of the asteroid (yA31) using each of the available techniques. The results are contained in Table 1. In this table, each column represents one of the techniques (Slicer, MIP, Compositing, Gradient-based opacity, Illumination). While the rows contain the corresponding attributes (temperature, water, asteroid, pressure). Our interpretation of the images is based on very rudimentary knowledge about astronomy. A finer and more rigorous analysis could only be carried out by an expert in the field.

---

[6] This implementation was based in the lecture notes Spatial Data I (Slide 22)

2IMV20 (2018-GS2) Visualization

Table 1. Comparison of volume rendering techniques to visualize the attributes of the collision of an asteroid

	Slicer	MIP	Compositing	Gradient based opacity	Illumination
PRS					
TEV					
V02					
V03					

All visualizations correspond to time point 39476 of the asteroid "yA31". PRS (pressure in microbars), TEV (temperature in electron volt), V02 (volume fraction water), and V03 (volume fraction of asteroid).

The first conclusion that we can extract is that in this case the slicer alone does not generate representations that are sufficiently clear to obtain significant insights from data that is not intuitive.

MIP allows us to see where the densest parts are located. For the pressure, it generates a uniform plane which is not very informative. In the case of the temperature, MIP paints with a more intense white those areas that we intuit correspond to the hottest areas detected by the sensor. Following the same logic, we can also appreciate that the tail of the asteroid keeps a slight trace of heat after the impact. For the case of the volume fraction of water, we can observe that the column of water that is generated after the water is distributed quite uniformly around the impact zone. Finally, the image of the volume fraction of the asteroid suggests that some rock fragments bounce back into the atmosphere after the impact.

The compositing technique allows us to distinguish several depths with different colors, as long as the histogram is adjusted to do it (for some cases, it is more than difficult to select different depths by colors). Something similar can be seen when we analyze the volume fraction of the asteroid. For the volume fraction

of water and the temperature, different color with different opacities are used, to put in contrast the different depths of the image. In addition, with this technique and a good color configuration, we can already represent an object in three dimensions.

Gradient based opacity & Illumination: With this technique, the boundaries of an object can be easily identified. When we incorporate illumination, it is possible to clearly distinguish the surface of the object (as you can see, for example, in temperature and volume fraction of water). The main difference when not using illumination is that we can play with the parameters to have a semitransparent object. Assuming that the variable pressure can provide clues as to where the collision occurred on the surface of the water, then Gradient Based Opacity provides us with the clearest visualization of that point.

To answer the second question that has to do with the description of an impact, we decided to explore the four variables of the first time point (14533) of the asteroid (yB31) using the technique that in our opinion produced the most representative visualization according to each variable. The results can be seen in Figure 7.
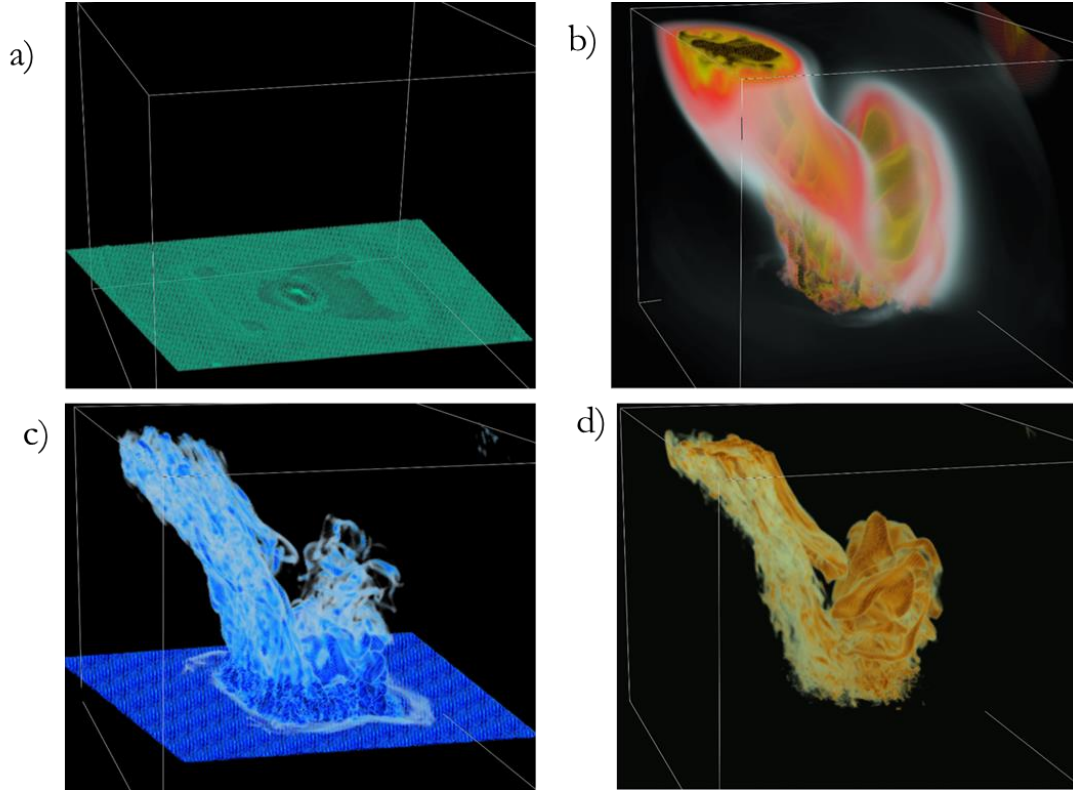
Figure 7. Impact of the asteroid (yB31) in 14533. a) Gradient based opacity applied to pressure, b) Compositing applied at temperature, c)) Compositing applied to the volume fraction of water, d) Compositing applied at temperature applied to the volume fraction of the asteroid.
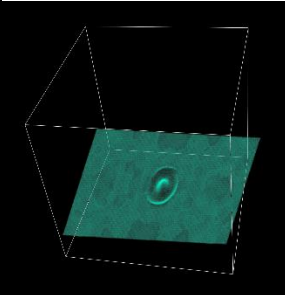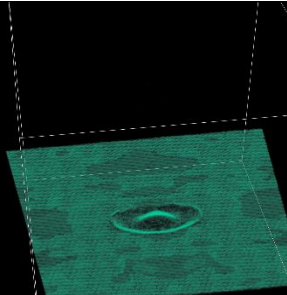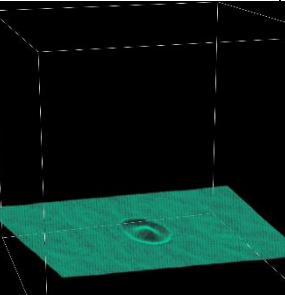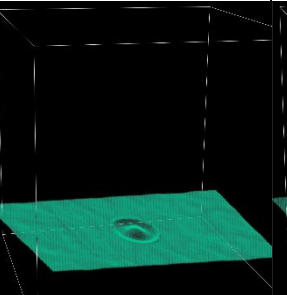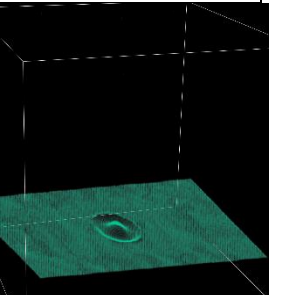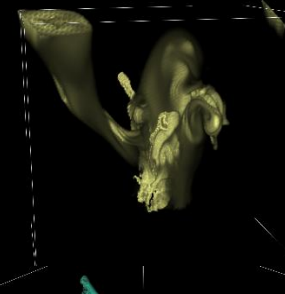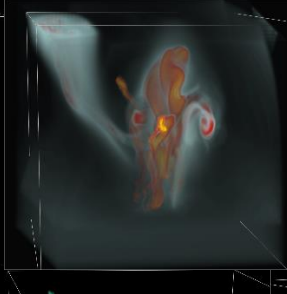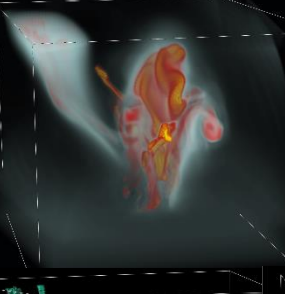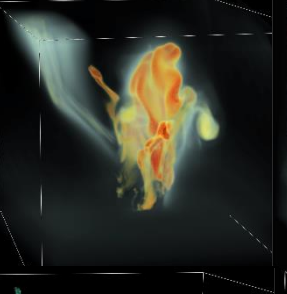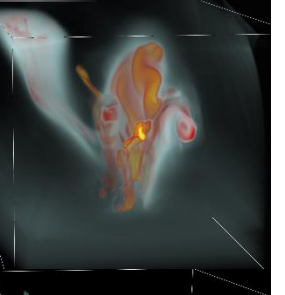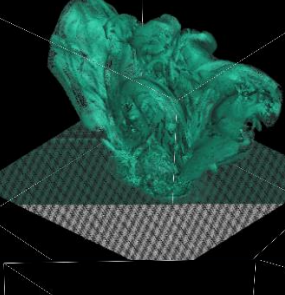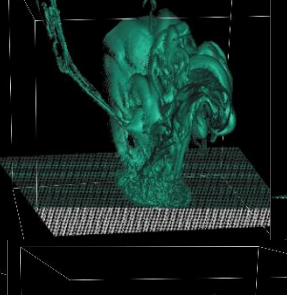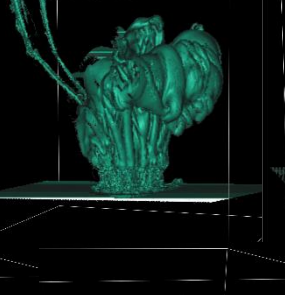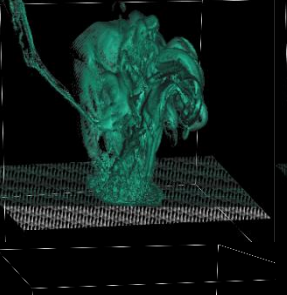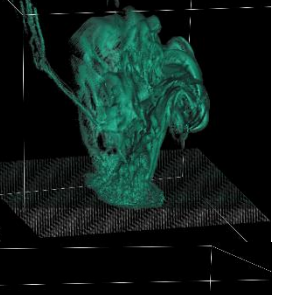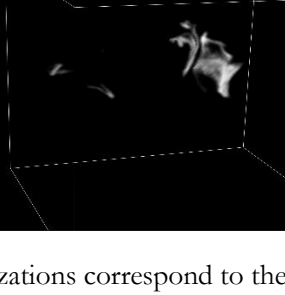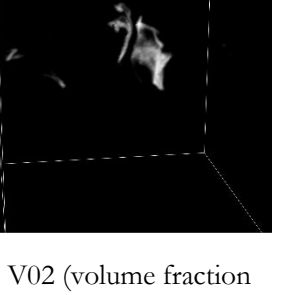
The point that is highlighted in the relief of the pressure map in (a) could be the specific place where the impact occurred and the shape that it had. The tail of the temperature figure in (b) could indicate the trajectory that the asteroid has followed from the atmosphere to the point of collision and the bulb around the point of impact could be the heat wave generated after the collision. In (c), when the impact occurs a column of water rises into the atmosphere that can be clearly distinguished and certain propagation waves are seen around the impact zone. Finally, the volume fraction of the asteroid in d) has a surface comparable to the amount of water displaced and we believe it corresponds to the particles or fragments of the asteroid that are released after the explosion.

To answer the third question that has to do with the temporal evolution of the variables associated with the impact, we created Table 2. In this table, each column represents the sequence of available time point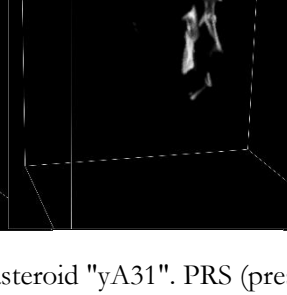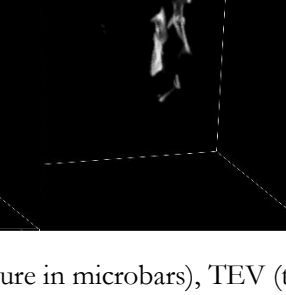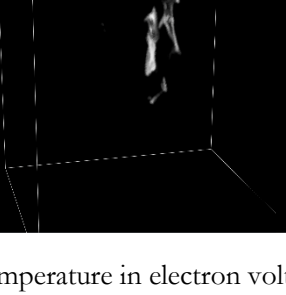s for the asteroid "yA31". While the rows contain the corresponding attributes (temperature, water, asteroid, pressure) and the technique used for their visualization.

Our initial impression is that in the covered interval there are no abrupt differences to report. Probably to observe more dramatic changes in the sequence would require a greater separation between successive time points.

2IMV20 (2018-GS2) Visualization

Table 2. Comparison of the attributes of the collision of an asteroid in different time points

	39476	39558	39640	39722	39803
PRS (Gradient Based Opacity)					
TEV (Compositing)					
V02 (Illumination)					
V03 (MIP)					

All visualizations correspond to the asteroid "yA31". PRS (pressure in microbars), TEV (temperature in electron volt), V02 (volume fraction water), and V03 (volume fraction of asteroid).

To cover the last question where the idea is to compare the impact of both asteroids, we select the initial time point of each asteroid and put side by side each of the variables to see if there is any identifiable difference. Each component is visualized according to the method that in our opinion generated the most representative visualization. For example, MIP was used to represent the volume fraction of the asteroid because we believe that being a rock can better identify where to find the bulk of the material. Each column of the Table 3 corresponds to an asteroid (with the respective time point). While the rows contain the corresponding attributes (temperature, water, asteroid, pressure).

Table 3. Comparison of the impact of the asteroids "yA31" and "yB31"

PRS (pressure in microbars), TEV (temperature in electron volt), V02 (volume fraction water), and V03 (volume fraction of asteroid).

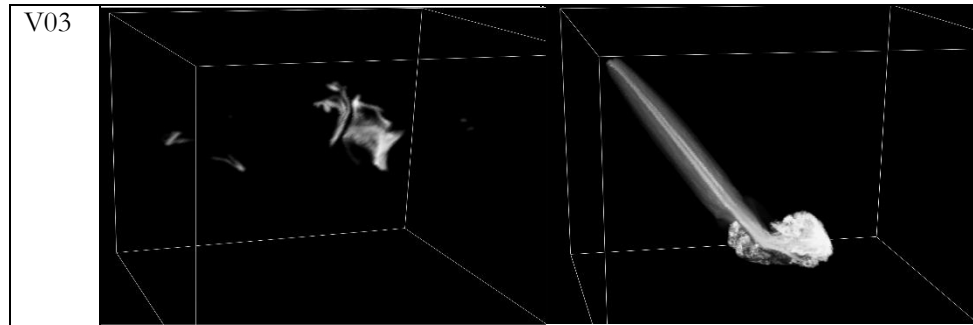In general terms, we can see that there are perceptible differences in the four variables. The pressure trace of yB31 seems more irregular. The zone in which the temperature is distributed in "yA31" seems larger. The way in which the volume of displaced water is distributed is different for each case. And finally, the area of the volume fraction of the asteroid of yB31 covers a surface considerably greater than yA31.

The document "Deep Water Impact Ensemble Data Set" that explains the fundamentals of the

IEEE Vis 2018 contains Figure 8, which represents the injection of water generated by impacts with and without airburst. When comparing these images with the results of Table 3, we believe that "yA31" probably corresponds to an event without airburst while "yB31" is an event with airburst. And this explains the differences observed. The same source[7] mentions that the direct impacts are those that offer the greatest risk of tsunami due to the higher height of the propagation waves.
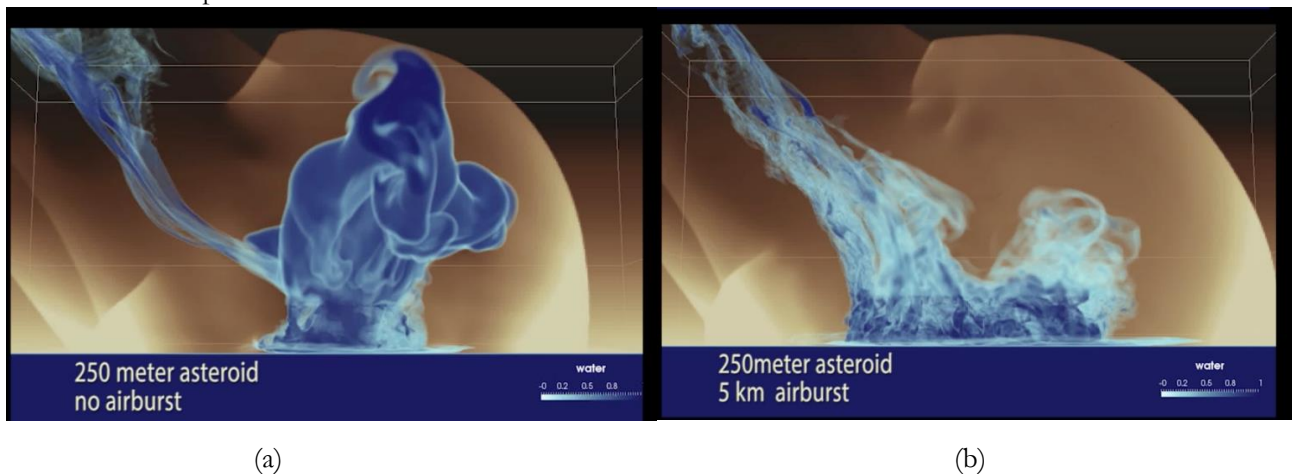
(a)   (b)

Figure 8. Comparison of an impact with and without airburst (Source: Patchett & Gisler, 2017)

---

[7] See the video "Visualization and Analysis of Threats from Asteroid Ocean Impacts", https://www.youtube.com/watch?v=95z0qRNFFxs

5. Conclusion

In general terms, the implementation of these techniques and data exploration allow formulating the following conclusions. The slicer, despite its high responsiveness and simplicity, was not particularly useful to obtain findings during the exploration.

 MIP is particularly effective in identifying the densest material or objects. However, since this technique does not take into account factors such as light reflection, then the 3D representation it provides is not realistic.

Compositing provides a more realistic 3D representation incorporating elements such as color and opacity, but at the expense of a lower interactive performance and involves the user defining various parameters.

Finally, 2D transfer functions provide a better result to visualize the boundaries of the object at the expense of a greater demand for computational resources and a tedious process of trial and error to identify the most appropriate parameters. The Phong shading model improves the realistic perception of the image.

Finally, about the decision of which method to use. We consider that the choice depends on the specific questions that the user seeks to answer, the familiarity and the nature of the object that is intended to be analyzed. We also believe that the combination of methods can be a valid strategy to maximize the number of relevant findings that can be derived from the analysis.

6. References

Levoy, M. (1988). Display of surfaces from volume data. IEEE Computer Graphics and Applications, 8(3):29–37.

Patchett, J. M., & Gisler, G. R. (2017). Deep Water Impact Ensemble Data Set. Los Alamos National Laboratory.

Phong, B. T. (1975). Illumination for computer generated pictures. Communications of the ACM, 18(6), 311–317.

Ray, Harvey, Hanspeter Pfister, Deborah Silver, and Todd A. Cook. 1999. Ray casting architectures for volume visualization. IEEE Transactions on Visualization and Computer Graphics 5(3): 210-223.