

Interactive obstruction-free lensing for volumetric data visualization

submission XYZ

Abstract—Occlusion is an issue in volumetric datasets visualization as it prevents direct visualization of the region of interest. To address this problem, many techniques have been developed such as transfer functions, volume segmentation or view distortion. Even if these techniques have proven their efficiency, there is still room for improvement to better support the understanding of objects' vicinity. However, most existing Focus+Context fail to solve partial occlusion in datasets where the target and the occluder are very similar density-wise. For these reasons, we investigated a new technique which maintains the general structure of the investigated volumetric dataset while addressing occlusion issues. As such, we propose a focus+context technique. The user interactively defines an area of interest where an occluded region or object is partially visible. Then our lens starts to operate and pushes at its border occluding objects (i.e. local deformation), thus revealing hidden parts of the volumetric data. Next, the lens is modified with an extended field of view (fish-eye deformation) to better see the vicinity of the selected region. Finally, the user can freely explore the surroundings for the area under investigation within this lens. To develop this technique, we used a GPU accelerated ray-casting framework with a set of interactive tools to ease volumetric data exploration and real-time manipulation. We illustrated the efficiency of this technique thanks to three examples where the occlusion issue is addressed: 3D scanned luggage exploration, aircraft trajectories, and streamlines.

Index Terms—Interaction techniques, focus and context, volume visualization

1 INTRODUCTION

Thanks to various rendering techniques, volumetric data can be displayed in many different fields (engineering, material sciences, medical imaging, etc.). Direct volume rendering techniques or isosurfaces techniques render these volumes into graphical representations in order to allow their exploration. In volume rendering, occlusion management is a challenge. As such, in 3d representations of volumes, some areas or objects (subsets) can be partially or fully hidden by others because of their locations.

Global techniques such as transfer functions, segmentation, and selection/clipping are used to remove occlusion in the entire volume. Therefore, they are a good way to reduce occlusion and make visible interesting features. However, it is still difficult to create a good transfer function/segmentation especially when the data are heterogeneous. In fact, designing a good transfer function depends heavily on the type of dataset and on the user's purpose. For instance, in the field of baggage inspection, the variation of densities prevents to create a unique transfer function for each baggage. In contrast, it is easier to design a good transfer function for a system dedicated to visualizing the same type of datasets (brain CT scans, bone tissues, etc.).

Many studies propose different tools and interaction techniques to by-pass occlusion issues in 3D environments such as lenses, deformations [6], augmented reality, etc. Lenses, which are flexible lightweight tools that enable local and temporary modifications of the visualization, are suitable to deal with occlusion while keeping information about the global context. This is a good local solution for occlusion problems and an interesting way to keep the user aware of the global meaning of the dataset. Thus, while most lenses in volume rendering are used to magnify a volume subset [34], we propose a focus+context (F+C) lens that combines a distortion technique which pushes aside the occluding objects, and a fish-eye field of view in order to provide a better perspective on partially occluded items of interest in the volumes.

Furthermore, performances are still a challenge in volume rendering systems. In fact, depending on the size of the dataset and also the resolution of the resulting produced image: the rendering process can be very slow. Some optimization strategies such as empty space skipping [25], early ray termination [2], multiple and adaptive resolutions allow to speed up the rendering process by increasing the frame rate. With

the advent of CUDA as a higher-level GPU programming language, CUDA-based ray-casters were introduced [20]. So, in order to support our focus+context interactive lens, our volume visualization system relies on a CUDA-based ray-casting algorithm [29]. This framework enables volumetric datasets visualization and offers a set of interactive tools including our lens for the purpose of easing the exploration and the manipulation of the data.

The structure of this paper is as follows. Section 2 presents related work in the areas of ray-casting, occlusion management, lenses and deformations. Section 3 describes the principle of our lens. Section 4 presents a method to convert vector datasets into a volume. Section 5 illustrates our lens technique with 3 scenarios. Section 6 discusses the presented technique. Finally, section 7 concludes the paper.

2 RELATED WORK

Many previous works have explored how to visualize volumetric data with lenses and distortion features to address occlusion issues. One major challenge is to maintain the data structures when visualizing the distorted information. To ensure a good visual quality, ray casting techniques are used to produce high-quality images. The computation time of these deformations in volume rendering remains an issue, and the produced images can hardly be generated in an interactive way. Other techniques can support interaction thanks to geometrical deformation or specific application domain deformations. In our paper, we introduce a new technique which allies high-quality rendering and fast computation to support the interactive visualization of lens-based deformation of volumetric data.

According to the classification of view deformations by Carpendale et al. [6], our framework uses a nonlinear radial distortion through an interactive lens in order to remove occluding items and keep the global context while magnifying a partially occluded item. Within the large body of work and variations of lens techniques, we frame our contribution as follows:

- we introduced an interactive deforming lens based on GPU accelerated ray-casting. Our lens magnifies and pushes aside occluding objects located in front of the designated focal point with an interactive frame-rate,
- we allow flexible and real-time interaction with the modification of the local point of view, the customized bent ray and the lens deformation parameterization. The inner part of the lens can be independently modified to provide a different perspective on the designated focal point.

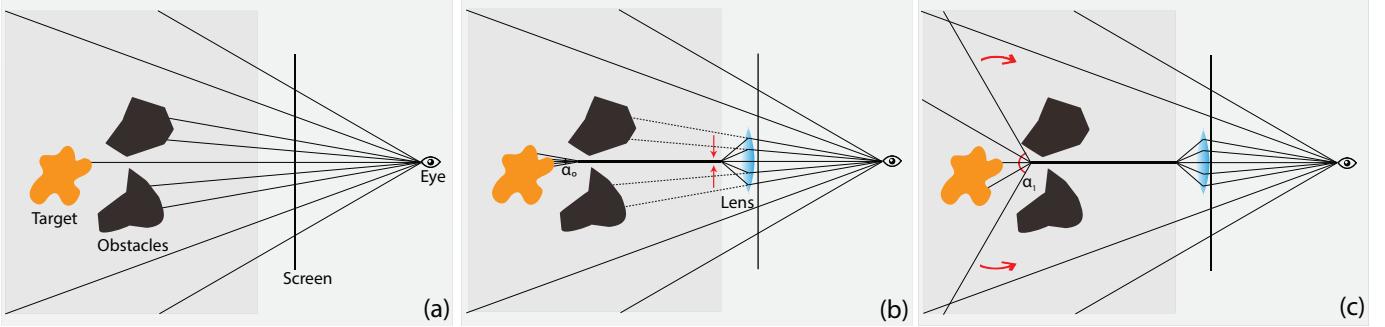


Fig. 1. The mechanism of the obstruction-free fish-eye lens. (a) The classic ray-casting where an interesting feature is partially hidden by other items in front of it. (b) The first main step: The lens makes converge the rays to avoid the obstacles. Once they are close to the target, the rays follow again their initial trajectory (with the initial angle) of view α_0 . Only a small part of the target is visible and magnified. (c) The target become visible by increasing the angle of view to $\alpha_1 \in [120, 180]$.

In this section, we introduce previous work categorized by occlusion management and lens deformations techniques in volume rendering.

2.1 Occlusion management

As mentioned by Elmqvist and Tsigas [11], many studies about occlusion management had been investigated. Multiple viewports, characterized by a view paradigm based on two or more separate views, can be used to see the scene under different perspectives [39]. Virtual X-ray methods make targets visible by turning occluding surfaces invisible [5] or semitransparent. Kruger et al. [22] introduced an interactive technique called ClearView, which enables the user to focus on particular areas in the data while preserving context information without visual clutter by using transparency. Correa and Ma [9] proposed visibility-driven transfer functions that maximize the visibility of the intervals of interest and provide high-quality images of volume data. However, it is still challenging to design a good transfer function, especially when the system has to deal with heterogeneous data. For instance, in baggage inspection, a dissimulation strategy is to hide a threat among objects with the same density. For direct volume rendering, in addition to removing voxels according to their density values and position, Rezk-Salama and Kolb [28] also took into account the voxels' occurrence on the casted ray. Later, Hurter et al. [17] selected voxels in feature spaces, such as color or intensity distribution. Li et al. [24] proposed a system for luggage visualization where any object is clearly distinguishable from its neighbors. They performed virtual unpacking for luggage visualization by visually moving any object away from its original location. This is a tool to manage occlusion, but destroying the context and the potential removal of important information inside the volume (positioning, connectivity). In our previous work [37], we have proposed an interactive visualization system that can support volumetric data exploration with direct manipulation of voxels. Little work has been done with a ray-casting approach for volume manipulation due to the computational costs involved. In order to better preserve the global context while addressing local obstruction problems, We extend our previous system with a lens that pushes aside occluding objects thanks to rays deformation.

2.2 Lenses and deformations

An interactive lens is a lightweight tool to solve a localized visualization problems by temporarily altering a selected part of the visual representation of the data [34]. Using this lens approach, we propose an interactive volume deformation based on GPU accelerated ray-casting to free a designated target from local occlusion while keeping the global context.

A lens is a parameterizable selection according to which a base visualization is altered. Typically, a lens is added to a visualization to interactively solve a specific localized problem. This property is very interesting with the aim of providing a focus+context solution to occlusion in volume rendering. Lenses can have different geometric properties not only defining their appearance, but also determining the

selection, which is the subset of the data where these lenses take effect. The major geometric properties of a lens are shape, position, and size as well as the orientation. The shape of a lens is usually chosen to fulfill the requirements of the application and is strongly linked to the lens function. Most virtual lenses are circular [33] or are rectangular [21] as the real-world ones (magnifying glass, windows). Our lens has also a circular shape in order to remind its magnifying property. Some lenses, such as the JellyLens [27] and the smart lenses [31] can adapt their shape automatically according to the data. The position and the size parametrization can increase the flexibility of an interactive lens. Modifying this position or size will set its focus on a different part of the data according to the user's interest. It is possible to update automatically these parameters in order to guide the user toward interesting events in the data [32], or adjust the lens position according to predefined paths as the Routelens [1] does. With this mind, our lens updates automatically its properties once a target has been selected. This allows a smooth transition towards an unobstructed and magnified area of interest.

Lenses for volume visualization face challenges mainly related to spatial selection and occlusion. Wang et al. addressed these issues by proposing the Magic Lens [38]. This Magic Lens renders the obstructions with higher transparency and magnifies volumetric pre-computed features interactively or automatically in a pre-segmented dataset. In addition to interactively magnifying areas and objects of interest, our lens frees them from obstruction and allows local modification of the camera to see the target under other perspectives. Tong et al. proposed the GlyphLens [36] that removes the occluding glyphs by pulling the glyphs aside through the animation, but this tool is only well suited for systems where 3D volumetric dataset are visualized using glyphs. Lenses can create discontinuities between their inner part and the rest of the volume. Deformation can be a solution to this discontinuity issue.

Hsu et al. developed a framework that can generate non-linear sampling rays that smoothly project objects in a scene at multiple levels of detail onto a single image [13]. Such a technique requires a lot of computational time to render a single image from features of interest at different scales. Bruckner and Groller [4] proposed exploded view for volume data by partitioning the volume into several segments, while Correa et al. proposed a framework [8] allowing the users to physically or actively manipulate the geometry of a data object. McGuffin et al. [26] performed deformations using peeling to see hidden parts of the data. However, these techniques have the disadvantage of removing potentially important surrounding contextual information while trying to solve the local occlusion.

Deformations can reveal predefined features in the dataset by taking into account the precomputed segmentation. Tong et al. proposed a deforming Lens which moves streamlines to observe the inner part of streamline bundles [35]. Some studies performed deformations using surgical metaphors [7, 19] to see hidden parts of the volume, but they do not offer tools for local manipulation of the point of view which allows perceiving a target under a different perspective while keeping the global context.

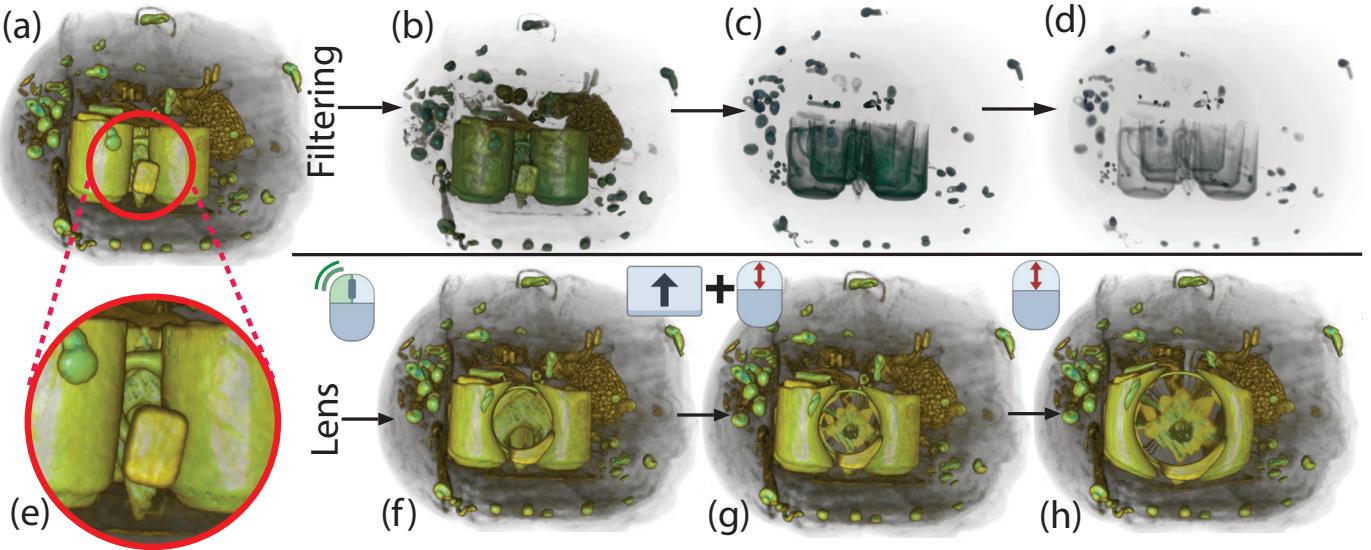


Fig. 2. (a) A suspect object is spotted between a set of mugs. Almost all the densities are still displayed. (b)-(d) Filtering the densities using a classical 1D transfer function. (c) The materials with low densities are hidden. The object of interest is still visible. (d) The target disappears as we try to see through the mugs. (e) The user applies the lens tool on the targeted object (double-click). The animation starts with the opening of the lens. After half of the animation, the blunt object is magnified but only the area close to the location where the lens has been applied is visible. (f) The fish-eye field of view at the end of the animation allows perceiving all the front part of the shuriken. (g) The size of the lens is increased to magnify the shuriken (scroll).

3 PRINCIPLE

Our flexible fish-eye based lens consists in modifying the rays (origin and direction) which cross the subset of the volume located under this lens. Our volume rendering framework relies on a GPU accelerated ray-casting algorithm. The basic idea of GPU-based ray-casting is to store the dataset into a 3D texture, and run a compute kernel that casts rays through this volume. Each pixel of the output image corresponds to the compositing of all sampled data along a single ray of sight. In classic ray-casting algorithms (Fig. 1-a), each ray is a line defined by the eye position e or the camera which is the same for all rays, and a normalized direction vector $\vec{d}(x,y)$ where x and y are the screen space coordinates of the pixel. The position of each sample along a single ray is then $\vec{p}(x,y,t) = \vec{e} + t \times \vec{d}(x,y)$ with $t \in [t_{near}, t_{far}]$, where t represents the depth or basically the distance between the eye and the sample which varies from the distance where the ray meets the volume t_{near} to the distance where the ray leaves the volume t_{far} . Our circular lens modifies the behavior of all the rays that travel through it in order to magnify a target. The new ray behavior can be divided into 3 steps. The first step is to provide an unobstructed view. It consists of moving closer to the target while avoiding the obstacles and pushing them aside. The second step is to set a wider field of view (fish-eye) in order to see most of the target. The last step which is optional offers a set of interaction to modify some visual parameters in real time such as the rays orientation and the field of view.

3.1 Setting an unobstructed view

While exploring a volumetric dataset, people want to have more details and information on items which are partially visible because of the occlusion created by other items in front of them. The aim of this step is to provide an unobstructed view by moving closer to the target while avoiding the obstacles and pushing them aside. Getting closer to the target can be carried out automatically or manually. When the user selects an occluded target, the rays traveling through the lens will share the same constant direction \vec{d}_{target} inside the volume until they reach same the depth t_{target} near the selected target. By default, this direction \vec{d}_{target} goes straight towards the selected item (the focal point \vec{p}_{target}). A sampled value inside the dataset located before the target has a position \vec{p} defined by $\vec{p}(x,y,t) = \vec{p}_{near}(x,y) + t \times \vec{d}_{target}$ where

$\vec{p}_{near}(x,y)$ is the point where the ray meets the volume while following its initial direction $\vec{d}(x,y)$. In addition to getting closer to the target, this new rays' direction also allows magnifying the subset of the volume located below the lens. In fact, inside the volume, these rays become parallel and converge toward the axis of the lens. This behavior creates a magnification below the lens (Fig. 1-b).

Moreover, we offer the possibility to describe manually the ray trajectory inside the volume. In this case the rays' trajectories are parallel curves going through the volume towards a selected location. The direction $\vec{d}_{target}(t)$ associates a normalized vector indicating a location to a depth t according to the user inputs. The position of a sampled value inside the volume is then: $\vec{p}(x,y,t) = \vec{p}_{near}(x,y) + t \times \vec{d}_{target}(t)$.

After the previous step, the rays are close to the target which can still be occluded. To address this issue, we introduced a radial attraction force for each ray which goes from the initial ray position after the previous step to the center of the lens. The new position of a sample point can be defined as $\vec{p}(x,y,t) = \vec{p}_{near}(x,y) + t \times \vec{d}_{target} + a \times \vec{f}_{attraction}$ with $a \in [0, 1]$ which represents the attraction factor. When a is close to 0 the rays follow their initial parallel directions, and conversely when a is close to 1 the rays follow the same trajectory which is the axis of the lens (when the trajectory was set automatically) or the curve designed by the user. During this step, attracting all the ray inside the lens towards a single trajectory allows pushing aside the obstacles around this trajectory. However, these new trajectories create discontinuities at the lens border. We propose a solution in section 3.4 .

3.2 The field of view

Once the rays of sight are close to the target, we are free from the obstacle but only see a small part of this target because of the zero distance between the rays. To address this issue, we propose to increase the field of view after the previous step. In fact, we diverge the rays which were previously sharing the same trajectory. Now, each ray is heading towards the outside of the lens according to the screen space coordinates of its corresponding pixel, as well as the angle of view. This wide angle of view (fish-eye) is by default set to $\alpha = 120^\circ$ instead of 180° in order to keep the focus on the initial target because the rays can still be surrounded by other occluding items. The field of view can be adjusted by the user to have a better visibility of the target by

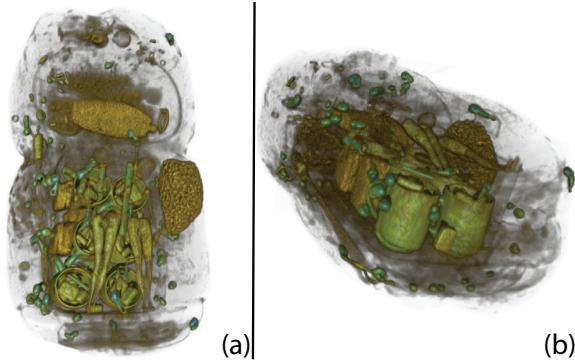


Fig. 3. A baggage presented under two different perspectives. (a) The baggage is composed by various type of items. the camera is located at the top front of the baggage. (b) The camera is located at the baggage side.

using the scroll wheel while pressing the Shift key according to his/her desired action (Fig. 1-c).

3.3 Interactive modification of the lens

Our lens can be interactively modified in order to offer more flexibility thanks to a set of different parameters involved in both previous steps. The first editable parameter is the size of the lens. We opted for a circular shaped lens so its size is then decided by the radius that can be modified interactively by the user with the mouse wheel. The purpose is to provide an interactive focus+context tool. The bigger the radius is, the more the target inside the lens will be magnified.

In addition, the trajectory which is automatically proposed after a target is selected can be modified by changing the location of the focal point \vec{p}_{target} via the arrow keys. The final direction of the rays \vec{d}_{target} can also be edited. This action can be carried out by a local rotation with the right button of the mouse. This allows adjusting the position of the rays near the target in a way to reduce the occlusion. Changing the directions of the rays can help to look around the target and get more information about its local context such the surrounding items and their relative positions.

Furthermore, the factor applied to the radial attraction force $\vec{f}_{attraction}$ helps to push aside or reposition the occluding items located before the target by using the mouse wheel while holding the Shift key. This allows the user to restore the initial context at will, which can be very helpful to understand the actual configuration. By default, we set this attraction factor a to the maximum $a = 1$. In other terms, the rays follow the same trajectory as the axis of the lens. The new position of a sample point is defined as $\vec{p}(x, y, t) = \vec{p}_{near}(x, y) + t \times \vec{d}_{target} + \vec{f}_{attraction}$ in order to reduce the occlusion and provide an unobstructed view on the selected target.

3.4 Continuity and transitions

All the rays traveling through the lens have trajectories and behaviors very different from those which never cross this lens. Without any additional post-treatment, the previous steps create discontinuities at the boundary of the lens. To address this issue we use an interpolation function between the final ray trajectory and the one before the previous steps. In fact, the closer a ray is to the lens border, the closer is new trajectory will be to the previous one. This interpolation offers a smoother transition between the lens viewport and the rest of the volume. This linear interpolation $p(k)$ between the new position p^1 of a sample along the ray and the one if it was not modified by the lens p^0 can described as: $p(k) = p^0 + f(k)(p^1 - p^0)$ where $k \in [0, 1]$ is a function that modifies the speed of the interpolation. For instance, we used $f(k) = k^2$ in order to reduce the interpolation near the center of the lens. The algorithm 1 shows a pseudo code of the behavior of our deforming lens

```

Input :  $\vec{e}$ : the eye position,
 $\vec{d}(x, y)$ : the ray direction according to the screen space coordinates of the resulting pixel ( $x$  and  $y$ ),
 $step$ : the sampling distance along the ray.
Result: the pixel color.
Parameter:  $a$ : the attraction factor,
 $\alpha$ : the angle of view.

 $k \leftarrow$  the normalized distance to the axis of the lens
 $\vec{p}_{near} \leftarrow \vec{e} + t_{near} \times \vec{d}(x, y)$  //The initial position ;
 $\vec{p}^0 \leftarrow \vec{p}_{near}$  ;
while  $t \leq t_{far}$  And  $Opacity \leq OpacityThreshold$  do
    if the current ray is inside the lens then
        if  $t < t_{target}$  then
             $\vec{p}^1 \leftarrow \vec{p}_{near}(x, y) + t \times \vec{d}_{target} + a \times \vec{f}_{attraction}$  ;
            else  $\vec{p}^1 \leftarrow \vec{p}_{target} + (t - t_{target}) \times \vec{d}_{fishEye}(\alpha)$  ;
        end
         $\vec{p} \leftarrow \vec{p}^0 + f(k) \times (\vec{p}^1 - \vec{p}^0)$  ;
        else  $\vec{p} \leftarrow \vec{p}^0$  ;
    end
    Sampling at the position  $\vec{p}$  ;
    Shading the sampled value ;
    Compositing the shaded sampling point with the previous values ;
     $t \leftarrow t + step$  ;
     $p^0 \leftarrow p^0 + step \times \vec{d}(x, y)$  ;
end
 $color_{final} \leftarrow$  composed colors ;
return  $color$ 
Algorithm 1: Pseudo code of our lens deformation algorithm

```

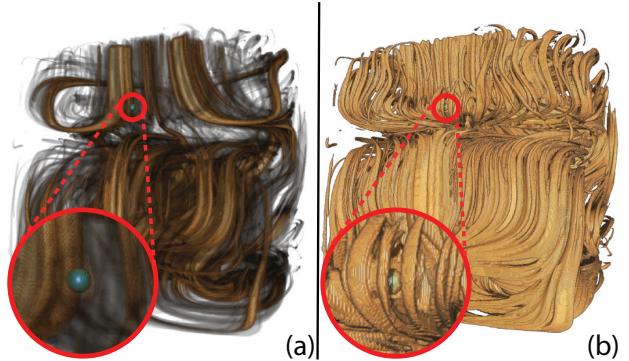


Fig. 5. This figure shows 2 visualizations of a streamlines dataset. (a) The transfer function used allows to spot a spherical dense item inside the volume thanks to transparency. (b) This spherical object become occluded when the opacity of the surrounding whirlpool is increased in order to analyze its shape and behavior.

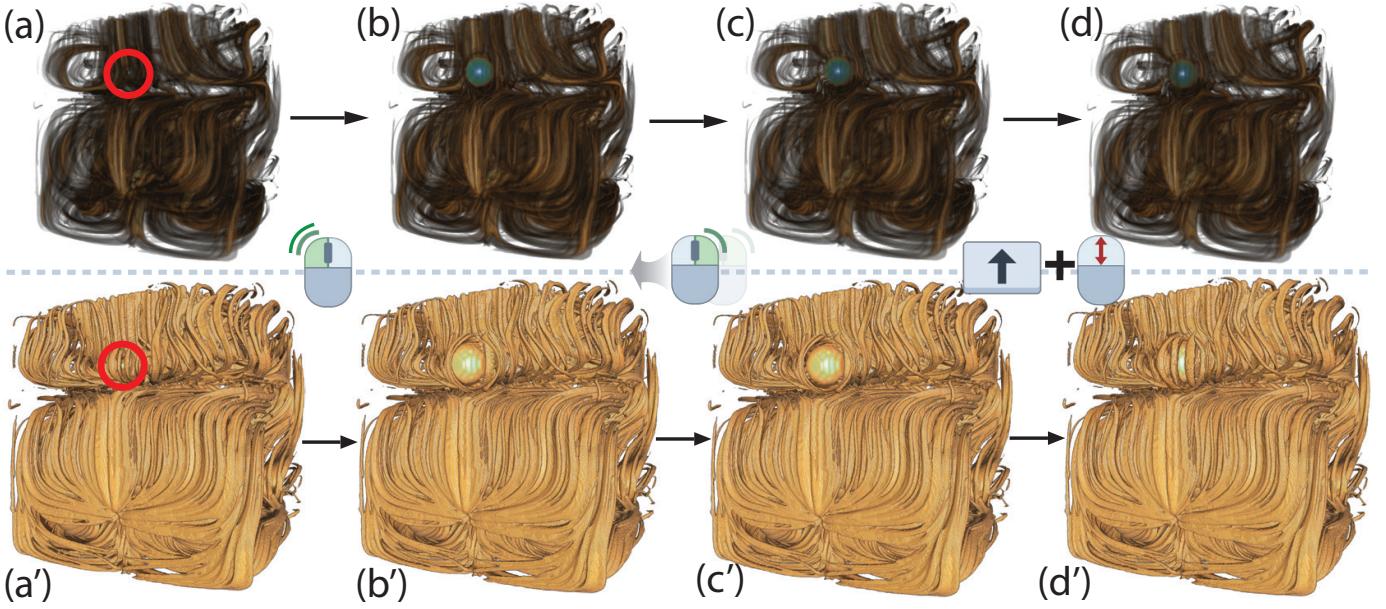


Fig. 4. A spherical item and its surrounding area are inspected using to our lens with 2 different transfer functions. The first part is displayed with transparency in opposition to the part below. The opacity helps to see the shape of the surrounding whirlpool. (a-a') The streamline dataset displayed using our framework. A dense object is hidden inside a whirlpool. (b-b') The lens tool is applied to the partially hidden object (double-click). The area is magnified and the occluding part of the whirlpool located in front of the spherical item is pushed aside. (c-c') The directions of the rays inside the lens are modified to see the whole sphere through the lens (right click+ mouse drag toward the desired direction). (d-d') The occluding part of the whirlpool can be restored/pushed gradually while keeping the area magnified (Shift + scroll).

After the selection of a target, the lens offers a smooth transition between the previous view and the new one using a slow-in/slow-out technique [10]. With this animation, the increase of the gradual speed at the beginning of the animation helps the user to start tracking the moving obstacles, and the decreasing speed at the end allows the user to predict where these occluding items stop moving. This also gives some semantic to the moving shapes, allowing the human mind to interpret the motion as a magnification of a target, and to keep the focus on visual entities during this transition.

4 FROM VECTOR TO VOLUMETRIC DATASET

In this section, we present our rational and method to transform existing vector datasets into a volume. Moving objects are entities that change locations in space over time. This is the case with aircraft trajectories where an aircraft takes off at a given airport at the beginning of its strip and lands onto another one at the end of its journey. Classical methods use projected lines on a 2D visualization to explore dataset composed of such moving objects [15]. Even if such methods have shown valuable results, additional insight and data retrieval can be envisaged with other rendering techniques like volume rendering. While volume rendering techniques mainly focus on medial data visualization, the visualization of moving objects opens new opportunities taking advantage of the volumetric data rendering process. Using such technique with moving object dataset is not trivial, one must transform the original data (vector data) into a compatible volumetric dataset. This process is also called rasterization which turns 3D floating points into a 3D raster grid. In the following, we explain such process and then we detail our investigated datasets.

Dataset rasterization: While Bresenham's line algorithm is a standard algorithm to turn a line from the vector space to a raster 2D or 3D space, it will not fulfill volume rendering compatibility. Raster line must have a given thickness to maximize the number of intercepting rays and thus ensuring their visualization thanks to volume rendering methods. Other techniques can be envisaged with the computation of meshes (i.e. pipes that encompasses the 3D line) but such computation can create geometric artifacts and might be long to process. Therefore, we used an extended version of the kernel density estimation algorithm [30] into a 3D space. Such method has already shown interesting

results with visual simplification and trajectory aggregation [16] and is part of so-called pixel-based visualization [14]. Such process works as follows: we first define a 3D Gaussian kernel of a radius R ; the center of this kernel has the highest value, while its border the lowest. Second, we re-sample every trajectory with a minimum distance between two consecutive points inferior to the half of the kernel radius. Third, we compute the convolution of this 3D kernel with the re-sampled trajectories.

Convolution can take time with a large dataset to process, therefore, we turned this computation into the frequent space. Such technique has already been applied with trajectory dataset and can even be accelerated thanks to GPU techniques [23].

5 SCENARIOS

The obstruction-free fish-eye lens can be used with different types of volumetric datasets. First, we present a scenario based on baggage inspection (an heterogeneous dataset). Second, we apply our lens technique to a volume of streamlines where the transparency doesn't allow seeing a dense spherical item within its context. Finally, we observe a special aircraft trajectory within its context inside a dataset representing one day of traffic.

5.1 Baggage inspection: an unusual blunt object

In most airports, security agents deal with volumetric data exploration during baggage inspections. While automatic systems are now able to detect harmful densities (such as C-4, TNT, Nitroglycerin, etc.) and even some prohibited articles (such as classical firearms and knives), it remains difficult to identify unusual threats. In addition, baggage inspection faces 4 main concealment strategies:

Superposition: A threat (e.g. prohibited object like a knife, a cutter) may be sheltered behind dense materials. Sometimes, it's possible to see through these blind shield using some functionalities such as high penetration (enhanced X-ray power) or image processing (contrast improvement).

Location: Depending on its location inside the luggage, a threat can be difficult to detect. Objects located in the corners, in the edges or inside the luggages frame are very difficult to identify.

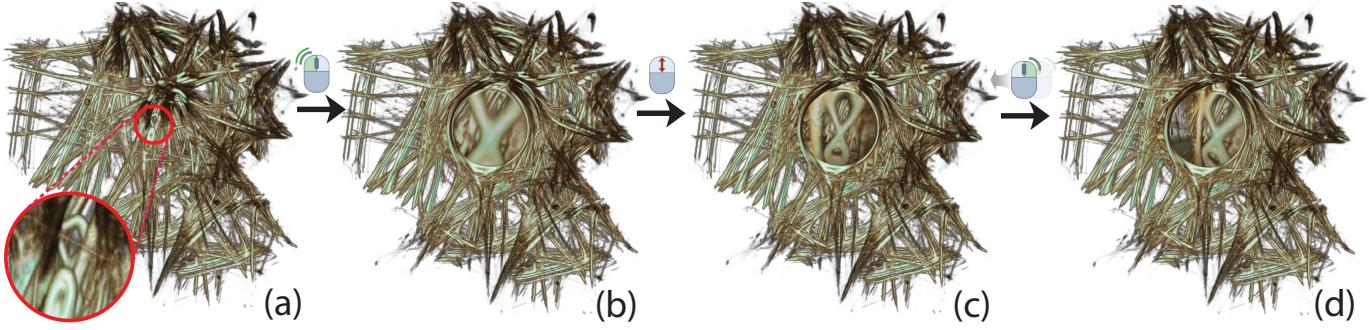


Fig. 6. Inspecting an abnormal aircraft trajectory. (a) The initial view on the trajectories where the abnormal trajectory was spotted. (b) The transition towards a magnified trajectory starts just after the lens tool has been used (double click). (c) The trajectories is magnified. (d) Local Rotations allow to look around the target (right click+ mouse drag toward the desired direction).

Dissociation: Another way to dissimulate a threat is to separate and to spread parts of it in the luggage (weapon or explosive are composed of many separated items like the trigger, the cannon...). This dissociation can be combined with other dissimulation techniques.

Lure: An ill-intentioned individual may use a lure to hide the real threat. For instance, a minor threat like a small scissors may be clearly visible and catch security agents attention while a more important threat remains hidden.

As an example, a baggage containing different types of objects as in Fig. 3 (with a volume size of 283x189x344) can be declared non-suspect by automatic inspection systems. However, while exploring this baggage with different angles and perspectives, it appears that an object is hidden between a set of mugs. A common solution to this type of issue in baggage inspection is to filter the materials by density in order to show or hide subsets of the volume and reduce the occlusion. But in this case, trying to discriminate the materials on the basis of their densities is not enough to fully reveal the hidden object (Fig. 2b-d). In fact, this suspect item shares almost the same density as the surrounding ceramic mugs.

Using the obstruction-free fish-eye lens can help in this kind of situation. The user has just to use this tool on the partially hidden target. Then, a transition inside the lens will start and smoothly provide the finale unobstructed view of the blunt object which is, in this case, a ceramic shuriken (Fig. 2e-g).

5.2 Streamlines visualization

Streamlines visualization has a long history in the scientific visualization community [3]. One of the major challenges is to produce an efficient visualization to display flow directions and their structure in a dense and tangled set of streamlines. In this scenario, we do not claim to propose a new visualization technique, but we rather show how our interactive fish-eye lens can leverage user ability to inspect interesting phenomena within its context (a dense and cluttered streamline dataset). Fig. 5 shows 2 visualizations of our streamline dataset. We use the dataset extracted from [12] which corresponds to a water flow in a basin computed on a grid of 128x85x42 cells, a set of 4595 stream lines with 183k sampled points. We use an accumulative blending process to distinguish dense areas where many streamlines are aggregated and generate a volume with a size of 500x500x500 voxels. In the first picture Fig. 5-a the streamlines are displayed with transparency, which allows noticing a dense spherical item inside the dataset. But when we increase the opacity of the surrounding whirlpool in order to better see its shape and behavior, the spherical item become occluded. Thanks to our interactive lens, the user can distort the view and push on the border of the lens the occluding stream ,and then, inspect this spherical object as well as its context in more details by defining different points of view (Fig. 4). Interesting enough, this object corresponds to a streamline with a tiny velocity which makes this line very short. Since every stream in our dataset contains the same number of sampled points, this object has a high density (we used a data convolution technique to turn our vectorial streamline dataset into a volumetric one). In conclusion,

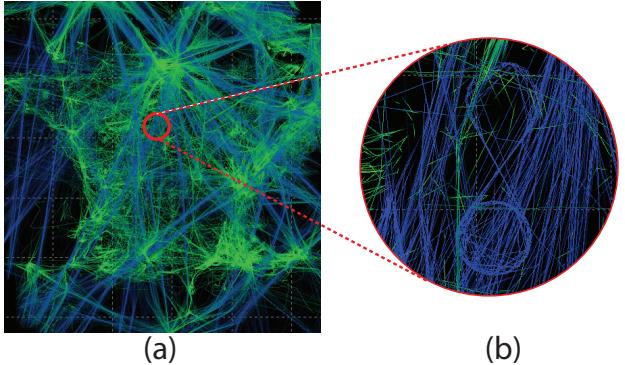


Fig. 7. Standard visualization of one day of recorded aircraft trajectories over France. [18]. (a) An unusual trajectory is spotted but needs some manipulation to be unobstructed. (b) Zoom and color filtering technique to visualize abnormal trajectory of one aircraft performing a loop with an eight shape trajectory. This aircraft corresponds to a tanker waiting for refueling other aircraft.

our technique eases the detection and the investigation of dataset subset. Other techniques can be applied, but our method is fast and flexible enough to leverage user ability with this streamline dataset exploration.

5.3 Unusual aircraft trajectories in French sky

One of the major issue when visualizing large dataset of moving object is to address the occlusion issue where too many lines spoil their investigation. Many investigations have already been done, especially regarding aircraft movement exploration [15]. Figure Fig. 7-a shows one day of recorded aircraft trajectories and Fig. 7(b) shows a subset of such dataset where one can visualize an abnormal aircraft trajectory. A tanker aircraft performed an eight shape loop while waiting to refuel other aircraft. Even if the visualization of such specific trajectory is possible with existing tools, it remains a difficult task which requires time and complex settings. With our technique (with a volume size of 500x500x500), the user can easily spot the corresponding aircraft and even if this trajectory remains barely visible, our lens tools will remove the occluding trajectory thus providing a suitable point of view to fully investigate this trajectory. Furthermore, this obstruction free lens allows to look around the targeted trajectory in order to look for neighboring trajectories.

6 DISCUSSION

In this paper, we presented three different scenarios where we showed how our lens is a fast and flexible way to overcome occlusion issues: the exploration of a heterogeneous dataset (baggage inspection), the analysis of a dense object within its context, and the exploration of a special aircraft trajectory.

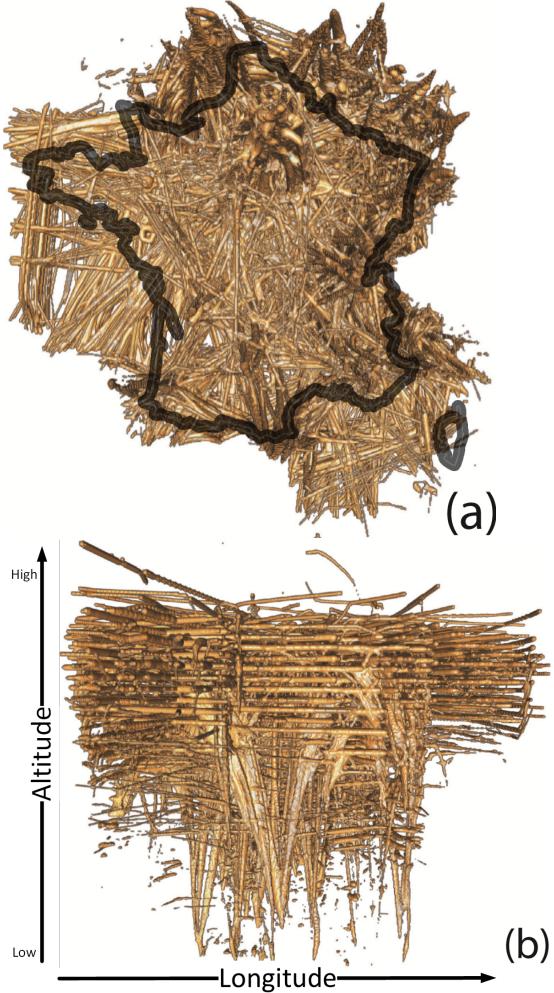


Fig. 8. Visualization of one day of recorded aircraft trajectories over France using our volume rendering framework. (a) The trajectories almost draw the map of France. (b) The trajectories are displayed according to their altitudes.

Nevertheless, some aspects of the lens we presented throughout this document suffers from some limitations which we detail in the following.

First, keeping the continuity between the inner part of the lens and the rest of the volume is very important to preserve a good understanding of object deformations. To ensure this continuity, we used a linear interpolation function between the final ray trajectory and the one before the previous steps. In fact, the closer a ray is to the lens border, the closer is new trajectory will be to the previous one. We used linear interpolation, whose parameter was modified with a function $f(k) = k^2$ in the purpose of reducing the interpolation near the center of the lens. Different functions and mathematical tools could have been used but the result obtained with the current function was satisfying.

Second, we used a circular lens throughout this study. The circular shape seemed more natural for us but, we could have proposed different lens shapes. At this stage, only the radius can be increased or reduced in order to customize this shape. Furthermore, during the deformation of the rays, the shape of the items considered as obstacles and pushed aside, are modified. It would be interesting to keep their original shapes to improve this focus + context lens. A solution is to adapt our rays' trajectories modification to physical based deformation inside the lens. However keeping the original shapes of all the objects that have been pushed aside can change the global context. In fact, they will either create more occlusion outside the lens or change the items' locations outside the lens by moving them away.

Third, the angle of view in the second step of our algorithm which allows having a good sight of the targeted item and its local context can be improved. In fact, using a segmentation algorithm that computes the bounding box of the target will help to define precisely and automatically the most suitable angle of view. However, we do think it's still important to offer the possibility to modify this angle at will. In fact, this flexibility allows further exploration of the local context.

In addition during the first main step of our lens algorithm, we try to get closer to the targeted item or area while pushing the encountered obstacles aside. Automatically finding the right distance to the target, that offers the optimum balance between a good perspective on the full target and obstacle avoidance can be sometimes difficult according to the structure of the dataset. In our future works, we will instigate automatic curved rays in deforming lens in the purposes of providing in most cases a very good perspective on the target while avoiding occlusion.

Finally, our ray distortion is flexible enough to support any deformation but it suffers from a lack of suitable interaction paradigm. Thanks to our technique, the casted ray can be freely deformed and thus twirled around the whole volume. We tested this feature which creates complex deformations. To define this twisted path, we let the user freely explore the volume with a free navigation paradigm; the user moves the point of view camera and we record its path. When the user stops his/her navigation, we go back to the initial camera location and we use the recorded path as the main deformation ray. While this interaction paradigm is suitable to show the flexibility of our lens deformation, it remains not satisfactory in terms of interaction efficiency. The user free navigation takes times and others path defining technique must be designed. This part remains a difficult but promising future work.

7 CONCLUSION

In this paper, we detail a new deforming lens as a solution to occlusion in volume rendering. The mechanism of this interactive lens consists in firstly pushing aside occluding items in order to provide an unobstructed view, and secondly magnifying the targeted object and its local context thanks to a fish-eye field of view. The flexibility of this interactive lens allows modifying its parameters such as the rays' directions, the size of the lens, the angle of view for the purposes of adjusting an automatically provided perspective on a target, and exploring its local context.

This lens is well suited to magnify a partially hidden object in datasets where the transfer function alone cannot resolve occlusion issues while preserving the global object structure. Three concrete scenarios using different types of volumetric datasets (baggage, stream-

lines, and aircraft trajectories) illustrate how to take advantage of this tool.

Some improvements can be carried out in order to automatically provide a better perspective on a selected target, such as taking into account its bounding box or automatically propose curved rays.

REFERENCES

- [1] J. Alvina, C. Appert, O. Chapuis, and E. Pietriga. Routelens: Easy route following for map applications. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, AVI '14, pp. 125–128. ACM, New York, NY, USA, 2014. doi: 10.1145/2598153.2598200
- [2] J. Beyer, M. Hadwiger, and H. Pfister. State-of-the-art in gpu-based large-scale volume visualization. *Computer Graphics Forum*, 34(8):13–37, 2015. doi: 10.1111/cgf.12605
- [3] A. Brambilla, R. Carneky, R. Peikert, I. Viola, and H. Hauser. Illustrative flow visualization: State of the art, trends and challenges. *Visibility-oriented Visualization Design for Flow Illustration*, 2012.
- [4] S. Bruckner and M. E. Groller. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1077–1084, Sept 2006. doi: 10.1109/TVCG.2006.140
- [5] M. Burns and A. Finkelstein. Adaptive cutaways for comprehensible rendering of polygonal scenes. In *ACM SIGGRAPH Asia 2008 Papers*, SIGGRAPH Asia '08, pp. 154:1–154:7. ACM, New York, NY, USA, 2008. doi: 10.1145/1457515.1409107
- [6] M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Extending distortion viewing from 2d to 3d. *IEEE Computer Graphics and Applications*, 17(4):42–51, Jul 1997. doi: 10.1109/38.595268
- [7] C. Correa, D. Silver, and M. Chen. Feature aligned volume manipulation for illustration and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1069–1076, Sept. 2006. doi: 10.1109/TVCG.2006.144
- [8] C. Correa, D. Silver, and M. Chen. Illustrative deformation for data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1320–1327, Nov. 2007. doi: 10.1109/TVCG.2007.70565
- [9] C. D. Correa and K. L. Ma. Visibility histograms and visibility-driven transfer functions. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):192–204, Feb 2011. doi: 10.1109/TVCG.2010.35
- [10] P. Dragicevic, A. Bezerianos, W. Javed, N. Elmqvist, and J.-D. Fekete. Temporal distortion for animated transitions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 2009–2018. ACM, New York, NY, USA, 2011. doi: 10.1145/1978942.1979233
- [11] N. Elmqvist and P. Tsigas. A taxonomy of 3d occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1095–1109, Sept 2008. doi: 10.1109/TVCG.2008.59
- [12] M. Griebel, T. Preusser, M. Rumpf, M. A. Schweitzer, and A. Telea. Flow field clustering via algebraic multigrid. In *Visualization, 2004. IEEE*, pp. 35–42. IEEE, 2004.
- [13] W.-H. Hsu, K.-L. Ma, and C. Correa. A rendering framework for multi-scale views of 3d models. *ACM Trans. Graph.*, 30(6):131:1–131:10, Dec 2011. doi: 10.1145/2070781.2024165
- [14] C. Hurter. Image-based visualization: Interactive multidimensional data exploration. *Synthesis Lectures on Visualization*, 3(2):1–127, 2015.
- [15] C. Hurter, S. Conversy, D. Gianazza, and A. Telea. Interactive image-based information visualization for aircraft trajectory analysis. *Transportation Research Part C: Emerging Technologies*, 47:207–227, 2014.
- [16] C. Hurter, O. Ersoy, and A. Telea. Graph bundling by kernel density estimation. *Computer Graphics Forum*, 31(3pt1):865–874, 2012.
- [17] C. Hurter, R. Taylor, S. Carpendale, and A. Telea. Color tunneling: Interactive exploration and selection in volumetric datasets. In *2014 IEEE Pacific Visualization Symposium*, pp. 225–232, March 2014. doi: 10.1109/PacificVis.2014.61
- [18] C. Hurter, B. Tissoires, and S. Conversy. Fromdady: Spreading data across views to support iterative exploration of aircraft trajectories. *IEEE TVCG*, 15(6):1017–1024, 2009.
- [19] S. Islam, D. Silver, and M. Chen. Volume splitting and its applications. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):193–203, March 2007. doi: 10.1109/TVCG.2007.48
- [20] B. Kainz, M. Grabner, A. Bornik, S. Hauswiesner, J. Muehl, and D. Schmalstieg. Ray casting of multiple volumetric datasets with polyhedral boundaries on manycore gpus. In *ACM SIGGRAPH Asia 2009 Papers*, SIGGRAPH Asia '09, pp. 152:1–152:9. ACM, New York, NY, USA, 2009. doi: 10.1145/1661412.1618498
- [21] R. Kincaid. Signallens: Focus+context applied to electronic time series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):900–907, Nov. 2010. doi: 10.1109/TVCG.2010.193
- [22] J. Kruger, J. Schneider, and R. Westermann. Clearview: An interactive context preserving hotspot visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):941–948, Sept 2006. doi: 10.1109/TVCG.2006.124
- [23] A. Lhuillier, C. Hurter, and A. Telea. Ffbeb: Edge bundling of huge graphs by the fast fourier transform. In *PacificVis 2017, 10th IEEE Pacific Visualization Symposium*. IEEE, 2017.
- [24] W. Li, G. Paladini, L. Grady, T. Kohlberger, V. K. Singh, and C. Bahlmann. Luggage visualization and virtual unpacking. In *Proceedings of the Workshop at SIGGRAPH Asia*, WASA '12, pp. 161–168. ACM, New York, NY, USA, 2012. doi: 10.1145/2425296.2425325
- [25] B. Liu, G. J. Clapworthy, and F. Dong. Accelerating volume raycasting using proxy spheres. In *Proceedings of the 11th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis'09, pp. 839–846. The Eurographics Association & John Wiley & Sons, Ltd., Chichester, UK, 2009. doi: 10.1111/j.1467-8659.2009.01466.x
- [26] M. J. McGuffin, L. Tancau, and R. Balakrishnan. Using deformations for browsing volumetric data. In *IEEE Visualization, 2003. VIS 2003.*, pp. 401–408, Oct 2003. doi: 10.1109/VISUAL.2003.1250400
- [27] C. Pindat, E. Pietriga, O. Chapuis, and C. Puech. Jellylens: Content-aware adaptive lenses. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pp. 261–270. ACM, New York, NY, USA, 2012. doi: 10.1145/2380116.2380150
- [28] C. Rezk-Salama and A. Kolb. Opacity peeling for direct volume rendering. *Computer Graphics Forum*, 25(3):597–606, 2006. doi: 10.1111/j.1467-8659.2006.00979.x
- [29] S. Roettger, S. Guthe, D. Weiskopf, T. Ertl, and W. Strasser. Smart hardware-accelerated volume rendering. In *Proceedings of the Symposium on Data Visualisation 2003*, VISSYM '03, pp. 231–238. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2003.
- [30] B. W. Silverman. Density estimation for statistics and data analysis, 1986.
- [31] C. Thiede, G. Fuchs, and H. Schumann. Smart lenses. In A. Butz, B. Fisher, A. Krüger, P. Olivier, and M. Christie, eds., *Smart Graphics: 9th International Symposium, SG 2008, Rennes, France, August 27–29, 2008. Proceedings*, pp. 178–189. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [32] C. Tominski. Event-based concepts for user-driven visualization. *Information Visualization*, 10(1):65–81, Jan. 2011. doi: 10.1057/ivs.2009.32
- [33] C. Tominski, J. Abello, F. van Ham, and H. Schumann. Fisheye tree views and lenses for graph visualization. In *Tenth International Conference on Information Visualisation (IV'06)*, pp. 17–24, July 2006. doi: 10.1109/IV.2006.54
- [34] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. Interactive lenses for visualization: An extended survey. *Computer Graphics Forum*, pp. n/a–n/a, 2016. doi: 10.1111/cgf.12871
- [35] X. Tong, J. Edwards, C. M. Chen, H. W. Shen, C. R. Johnson, and P. C. Wong. View-dependent streamline deformation and exploration. *IEEE Transactions on Visualization and Computer Graphics*, 22(7):1788–1801, July 2016. doi: 10.1109/TVCG.2015.2502583
- [36] X. Tong, C. Li, and H. W. Shen. Glyphlens: View-dependent occlusion management in the interactive glyph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):891–900, Jan 2017. doi: 10.1109/TVCG.2016.2599049
- [37] M. Traoré and C. Hurter. Interactive exploration of 3d scanned baggage. *IEEE Computer Graphics and Applications*, 37(1):27–33, Jan 2017. doi: 10.1109/MCG.2017.10
- [38] L. Wang, Y. Zhao, K. Mueller, and A. Kaufman. The magic volume lens: an interactive focus+context technique for volume rendering. In *VIS 05. IEEE Visualization, 2005.*, pp. 367–374, Oct 2005. doi: 10.1109/VISUAL.2005.1532818
- [39] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '00, pp. 110–119. ACM, New York, NY, USA, 2000. doi: 10.1145/345513.345271