

# 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

Direct volume rendering (DVR) is a pervasive visualization technique for displaying 3D scalar fields in many application fields such as engineering, material sciences, and medical imaging sciences. Recent DVR methods are able to display large such scalar fields at interactive rates and allow exploration of structures of interest. However widely adopted, and able to accommodate large volumes of data, DVR inherently suffers from the problem of *occlusion*: Structures of interest located deep in the volume can be hard to spot and/or explore.

To aid with this, various mechanisms have been designed including transfer functions, segmentation, selection, and clipping. However, all such mechanisms have limitations. *Global* mechanisms, such as transfer function editing, can remove both occluders and objects of interest when these have similar densities. Moreover, in certain applications, carefully designed transfer functions exist and should be used without (significant) modifications to facilitate understanding and user training [?]. *Local* mechanisms such as segmentation, selection, or clipping are more effective in manipulating data confined to a given spatial region. However, many such mechanisms assume that one can easily and accurately select objects of interest to remove them (occluders) or keep them (occluded). This is hard to do when *e.g.* one does not have direct access to the occluded objects, or when significant 3D interaction is required to select the occluder(s).

A different approach to handling occlusion is to use *lenses*. Generically, these are flexible lightweight tools which enable local and temporary modifications of the DVR so as to reveal occluded objects while keeping the global visualization context [?, 5, 33]. However, efficiently selecting the occluded object of interest and removing all in-between occluders in such contexts is still challenging **ALEX: Must explain FAR more clearly which are the limitations we address with our lens!**

In this paper, we propose to increase the flexibility of lenses for DVR exploration in several directions. We propose a focus-and-context (F+C) lens that combines a distortion technique, which pushes aside the occluding objects, with a fish-eye field of view in order to provide a better perspective on partially occluded items of interest in the volumes. We specifically target the use-case of *partially occluded* objects, where the user has a glimpse of an interesting structure, buried deep

within the data, and only slightly visible from a given viewpoint and transfer-function setting. We allow the user to ‘open up’ the volume without changing these settings, and reveal the structure of interest, by a simple point, click, and scroll operation. Next, we provide several F+C modifications of the lighting parameters, transfer function, and geometry within the focus area so as to better understand the structure of interest. Our technique, implemented using a CUDA-based approach, can be easily incorporated in any generic DVR system.

The structure of this paper is as follows. Section 2 presents related work in occlusion management, lenses, and deformations for DVR visualization. Section 3 introduces our 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

Previous work has explored how to visualize volumetric data with lenses and distortion techniques to address occlusion issues. In this context, one major challenge is to maintain a comprehensible embedding of the deformed focus area within its context. Another challenge is to have techniques that operate at truly interactive rates (tens of frames per second), which can be difficult when the deformations are applied to volumetric data rendered by DVR. Related work can be further divided into occlusion management techniques and lenses-and-deformation techniques, as follows.

### 2.1 Occlusion management

Many approaches for occlusion management have been proposed [10]. Multiple viewports, a view paradigm using two or more views, can be used to see the data from different perspectives [38]. However, this does not help when the target is strongly occluded from *all* possible viewpoints. Virtual X-ray methods make targets visible by turning occluders invisible [4] or half-transparent. Kruger et al. [22] proposed ClearView, an interactive technique that enables users to focus on particular areas in the data while preserving context information without visual clutter by modulating the transparency. Correa and Ma [8] proposed visibility-driven transfer functions (TFs) to maximize the visibility of data intervals of interest. Yet, designing good TFs is still challenging in general: For instance, in baggage inspection, a dissimulation strategy is to hide a threat among objects with the same density, case in which one cannot easily remove occluders but keep the target by TF editing [?]. Similar situations occur when aiming to de-occlude a tumor from surrounding similar-density tissue in medical scans [?]. For DVR, in addition to removing occluding voxels based on density values and position, Rezk-Salama and Kolb [27] also considered the

voxels' occurrence on the casted ray. Later, Hurter et al. proposed several lens techniques that remove occluders by deforming (pushing them away) in a focus area, applicable to 2D images, multivariate volumes, and trail sets [17, 18]. However, such techniques rely on the specification of occluders based on data value-ranges, thus share limitations with ClearView and related techniques. Li et al. [24] proposed a system for baggage visualization where occluded objects are clarified by moving away occluders via an interactive process called virtual unpacking. However, this technique severely alters the *context* in which the occluder occurs, by altering or removing potentially important information, *e.g.*, relative position and connectivity – a clearly unwanted proposition in *e.g.* medical contexts. Recently, an interactive visualization system was proposed for volumetric data exploration with direct manipulation of voxels [36]. However, extending such approaches in a DVR setting to more complex deformations or changes of the data in focus is computationally challenging.

## 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 [33]. In other words, a lens is a parameterizable selection according to which a base visualization is altered. These property are very effective for providing focus-and-context (F+C) solutions to occlusion in volumetric data. Parametrizable properties of a lens include the position, shape, appearance, size, orientation, and selection of the included data (focus). The *shape* of a lens is usually chosen to fulfill the requirements of the application and is strongly linked to the lens function. Most lenses are circular [32] or rectangular [21]. Our lens has also a circular shape in order to remind its magnifying property. Some lenses, such as the JellyLens [26] and the smart lenses [30], can adapt their shape automatically to the focus data. Modifying the lens *position* and/or *size* sets its focus on a different part of the data according to the user's interest. One can automatically update position and size to guide the user toward interesting events in the data [31] or guide the exploration path along interesting events [1]. In this sense, 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. address these issues by proposing the Magic Lens [37], which renders the occlusions with higher transparency and magnifies volumetric pre-computed features interactively or automatically in a pre-segmented dataset. In addition to interactively magnifying areas of interest, our lens frees them from obstruction and allows local modification of the viewpoint, lighting, and TF, to offer many exploration perspectives. Tong et al. proposed the GlyphLens [35] that removes occluding glyphs by pulling them aside through animation. While effective, this technique addressed only 3D glyph-based volumetric visualizations. Lenses can create discontinuities between their inner part and the rest of the volume. Deformation can be a solution to this discontinuity issue.

To increase the flexibility of deformation, Hsu et al. developed a framework that uses non-linearly sampled rays to smoothly project objects in a scene at multiple levels of detail onto a single image [13]. However, this technique requires significant computational effort to render a single image from features of interest at different scales. Bruckner and Groller [3] proposed exploded views for volume data by partitioning the volume into several segments. Correa et al. proposed a framework [7] allowing users to physically manipulate the geometry of a data object. McGuffin et al. [25] performed deformations using peeling to see hidden parts of the data. In general, these techniques have the disadvantage of removing potentially important contextual information surrounding the target when trying to solve the local occlusion.

Deformations can reveal predefined features in the data by taking into account a precomputed segmentation. Tong et al. proposed a deforming lens which moves streamlines to observe the inner part of streamline bundles [34]. Other techniques performed deformations using surgical metaphors [6, 20] to show hidden parts of a volume. Such techniques do not offer tools for local manipulation of the viewpoint that allows seeing a target under multiple perspectives while keeping

the global context. To this end, our lens proposes an interactive volume deformation based on GPU accelerated ray-casting to free a designated target from local occlusion while keeping the global context.

## 2.3 Contributions

Summarizing the above, we propose a new technique which combines high-quality DVR with a fast, versatile, and easy to use, lens to support the interactive visualization of occluded data in volumes. According to the classification of view deformations by Carpendale et al. [5], we use a nonlinear radial distortion through an interactive lens to remove occluding items and keep the global context while magnifying a partially occluded item. Within the body of work on volumetric lens techniques, we frame our contribution as follows:

- we propose an interactive deforming lens that magnifies and pushes aside occluding objects located in front of a designated focal point at an interactive frame-rate. For this, we propose a GPU-accelerated raycasting scheme able to compute ray deformations;
- we allow flexible and real-time interactive modification of the focal point, custom bent rays used for DVR, lens deformation, and shading and transfer function in the focus area. Combined, these allow us to provide *on the fly* a range of perspectives of the objects in focus, that is, without having to change the viewpoint or manipulate complex parameters in linked views.

## 3 PRINCIPLE

Our proposed lens combines the modification of several parameters of a typical DVR rendering of volume data, as follows. Consider the typical DVR algorithm: Given a scalar volume  $V \subset \mathbf{R}^3 \rightarrow \mathbf{R}$ , each pixel  $\mathbf{x} \in I$  in the DVR image  $I \subset \mathbf{R}^2$  thereof corresponds to the compositing of sampled data along a ray starting passing through  $V$  and ending at  $\mathbf{x}$ . In classical DVR (Fig. 2-a), such rays are defined by the eye position  $\mathbf{e}$  and a viewing-direction unit vector  $\mathbf{d} = (\mathbf{x} - \mathbf{e}) / \|\mathbf{x} - \mathbf{e}\|$  pointing from  $\mathbf{e}$  to  $\mathbf{x}$ . Consider next a focus point  $\mathbf{f} \in I$  (the *lens center*) and a lens radius  $R > 0$ . In our proposal, we modify all rays traveling through the disk  $D = \{\mathbf{x} \in I | \|\mathbf{x} - \mathbf{f}\| \leq R\}$ , or *focus area*, in order to de-occlude, magnify, and emphasize a target object. Our new ray behavior can be divided into three steps: (1) Provide an unobstructed view of the occluded object. This moves closer to the target while avoiding the obstacles by pushing them aside. (2) Set a wide field-of-view (fisheye) to better see the target. (3) Interactively modify various parameters of the lens, lighting, and opacity TF in real time to better explore the target. These steps are detailed next.

### 3.1 Creating an unobstructed view

The scenario our lens addresses is as follows: Given a volume  $V$ , users produce a DVR thereof, using whatever suitable TFs and other parameters are applicable. When examining  $V$  from various viewpoints, (at least) one viewpoint  $(\mathbf{e}, \mathbf{d})$  is found from which some intriguing structure is *partially* visible in  $I$ . We call this structure the *target*. Users next want to quickly and easily unravel the target. For this, we proceed as follows: We first *gather* all rays passing through the lens pixels (focus area  $D$ ) to follow the lens' axis vector  $\mathbf{a} = (\mathbf{f} - \mathbf{e}) / \|\mathbf{f} - \mathbf{e}\|$ . As explained above, at the location  $\mathbf{f}$  of the lens center, we do see an interesting partially occluded target. Hence, by definition, the gathered rays pass *through* occluders to hit this target, otherwise we would not see it. We control gathering by setting the ray direction passing through  $\mathbf{x} \in D$  to

$$\mathbf{r}(\mathbf{x}) = (1 - \alpha)\mathbf{a} + \alpha\mathbf{d}, \quad (1)$$

with  $\alpha \in [0, 1]$ . When  $\alpha = 0$  (default value), all rays follow the lens axis  $\mathbf{a}$ , thus, can best pass through obstacles. When  $\alpha = 1$ , rays follow their original classical DVR path. Changing  $\alpha$  with the mouse wheel allows one to smoothly navigate between the lens effect, *i.e.* opening up a 'hole' in the volume to see the target, and a classical DVR visualization of the volume.

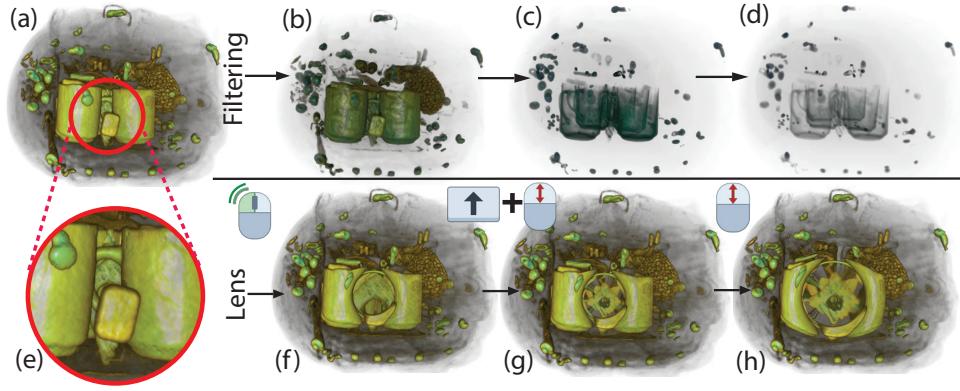


Fig. 1. (a) A suspicious object is spotted between a set of mugs. (b)-(d) Filtering densities using a classical 1D transfer function removes progressively more of the occluders (mugs), but also removes the target. (e) The user applies the lens on the target object (double-click). An animation starts opening the lens. Halfway the animation, the object is magnified, but only the area close to the lens is visible. (f) The fish-eye field of view at the end of the animation fully shows the target. (g) The lens is increased to magnify the target (mouse scroll).

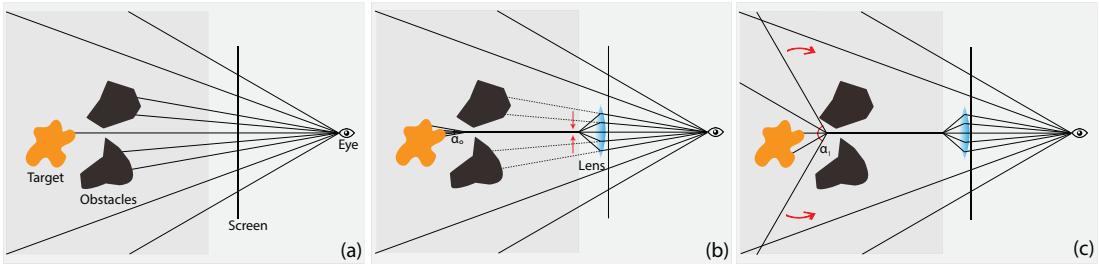


Fig. 2. Principle of the obstruction-free lens. An interesting object is partially hidden by occluders in front of it. (a) Classic raycasting result. (b) Our lens gathers the rays to avoid occluders. Once close to the target, rays follow again their initial paths. However, only a small part of the target is visible. (c) Scattering the rays makes the full target visible.

### 3.2 Setting a wide field of view

Once the rays pass obstacles (Sec. 3.1), we want to *scatter* them so as to best sample the target. Consider that this target is at some depth  $t_{target} > 0$  within  $V$ . After the rays pass the occluders, but before they hit the target, i.e., travel past a distance  $t_{min} < t_{target}$  through  $V$ , we deflect (scatter) them so as to best sample the target. For this, we set the parametric position of a ray point to

$$\mathbf{p}(\mathbf{x}, t) = \mathbf{r}(\mathbf{x})t + \beta(\mathbf{x} - \mathbf{f})(t - t_{min}) \quad (2)$$

for any pixel  $\mathbf{x} \in D$  and any  $t \geq t_{min}$ . Here,  $\beta \geq 0$  controls the ray scattering: Small values magnify a small volume area located close to the ray  $\mathbf{r}(\mathbf{x})$ ; larger values sample more of the volume area behind the ray. Intuitively, this works as if we moved a magnifying lens to a depth  $t_{min}$  inside  $V$ . Summarizing, after the user finds an interesting but partially occluded target using *standard* DVR, the lens squeezes rays to pass between occluders and next fans them out to reveal the target in full detail. The parameter  $\beta$  can be adjusted by the user via the mouse scroll wheel while pressing the Shift key (Fig. 2-c).

### 3.3 Interactive exploration of the target

We allow users to interactively modify several parameters of the DVR and the lens to achieve a more effective exploration, as follows.

**Lens radius:** The lens radius  $R$  can be controlled via the mouse wheel, thereby specifying how big is the ‘hole’ to open up in the volume to see the target. The parameters  $\alpha$  and  $\beta$  controlling respectively the gathering and scattering of rays are controlled by the mouse wheel and modified keys. The value  $t_{min}$  controlling the depth from which scattering starts is controlled using the arrow keys.

**Lens axis:** Users can rotate the lens axis  $\mathbf{a}$  using a virtual trackball activated by the right mouse button. Changing this direction effectively samples the target from all possible viewpoints, thereby allowing the user to look ‘around’ it so as to see its parts which are not visible

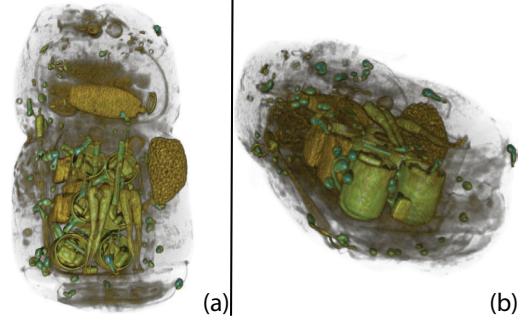


Fig. 3. A baggage presented under two different perspectives. (a) The baggage is composed by various type of items. the camera is located at the top front of the baggage. (b) The camera is located at the baggage side.

from the current viewpoint, but *without* having to actually change the viewpoint. This is very important, since changing the viewpoint may bring us to a situation from where the target is completely invisible, so we do not know where precisely to activate the lens any more.

**Lighting:** We modify the volumetric Phong lighting parameters to better explore the target, as follows. Let  $d(\mathbf{x})$  be the distance from a voxel  $\mathbf{x}$  to the target center  $\mathbf{e} + \mathbf{at}_{min}$ . Let  $\phi$  be the specular term coefficient, set by default to a high value. First, for all voxels  $\mathbf{x} \in V$  that project in the lens ( $d(\mathbf{x}) \leq R$ ), we use a specular coefficient  $\phi(\mathbf{x}) = \phi(1 - d)/R$ ; all voxels projecting outside the lens lens ( $d(\mathbf{x}) > R$ ) use a value  $\phi(\mathbf{x}) = 0$ . Hence, voxels close to the lens center appear specular, while voxels outside the lens appear diffuse. Secondly, we allow the user to rotate the light vector using the same trackball mechanism as for the lens axis rotation. These two mechanisms combined realize

the effect of a moving flashlight turning around a shiny target. This highlights small-scale details on the target surface, again, without having to change the viewpoint or lens location.

**Opacity:** Finally, we modify the opacity transfer function as follows. Let  $TF_o : \mathbf{R} \rightarrow [0, 1]$  be the user-chosen opacity function. Let  $TF_o^{lens}$  be an opacity function computed by adding to  $TF_o$  a Gaussian pulse centered at the average density value  $\rho$  in the target. Then, for each voxel  $\mathbf{x}$  projecting in the lens, we use an effective transfer function  $TF(\mathbf{x}) = TF_o d / R + TF_o^{lens} (1 - d) / R$ ; for voxels projecting outside the lens, we use  $TF_o$ . The effect is that voxels in a ball of radius  $R$  around the target will become less transparent, thus more visible. However, voxels having the same densities but outside this ball will use the default transfer function (which could make them transparent). This allows to have voxels with similar densities either opaque (if they are close to the target, thus interesting) or transparent (if they are e.g. in front of the target, thus occluding).

### 3.4 Smooth transitions

If we apply Eqns. 1 and 2 to all rays passing through the lens pixels  $D$ , and trace all other rays starting at pixels in  $I \setminus D$  as usual in DVR, discontinuities will appear at the lens borders. We solve this as follows. Let  $\mathbf{p}(\mathbf{x}, t)$  be the voxels along a lens ray starting at pixel  $\mathbf{x}$ , as computed by Eqn. 2. Let  $\mathbf{p}'(\mathbf{x}, t)$  be the voxels computed along an straight ray starting at the same pixel, i.e., using  $\alpha = 1$  and  $\beta = 0$  in Eqns. 1 and 2 respectively. For every value  $t$  along every such ray, we then compute the interpolated ray  $\bar{\mathbf{p}}(\mathbf{x}, t) = (1 - f(d))\mathbf{p}(\mathbf{x}, t) + f(d)\mathbf{p}'(\mathbf{x}, t)$ , where  $d$  is the unit-normalized distance of  $\mathbf{x}$  to the lens axis, and  $f : [0, 1] \rightarrow [0, 1]$  is an interpolation function. Next, we use the rays  $\bar{\mathbf{p}}(\mathbf{x}, t)$  to compute the DVR by standard composition. This way, rays effectively vary smoothly from their bent versions (close to the lens center) to straight lines (outside the lens). Setting  $f(d) = d^2$  keeps the interpolation area close to the lens border, so most of the lens displays the desired fisheye effect.

Separately, we use a slow-in/slow-out animation [9] to introduce the lens effect. When the lens is activated, we vary the values of  $\alpha$  and  $\beta$  from the defaults ( $\alpha = 1, \beta = 0$ , i.e. straight-line DVR) to their actual user-set values, compute the volume rendering on-the-fly, and display the resulting images. The overall effect resembles opening a hole in the volume. The speed increase at the start of the animation helps one to start tracking the pushed-away occluders; the decreasing speed at the end helps seeing where these occluders actually go. 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. When the lens is deactivated, we play back the animation in the opposite sense, which suggests closing the opened hole in the volume.

## 4 IMPLEMENTATION

Algorithm 1 shows a pseudo code of the behavior of our deforming lens

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.

**Dataset rasterization:** While Bresenham's line algorithm is a standard algorithm to turn a line form 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 [28] into a 3D space. Such method has already shown interesting

```

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
 $color_{final} \leftarrow$  composited colors ;
return  $color$ 

```

**Algorithm 1:** Pseudo code of our lens deformation algorithm

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 frequential 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, its 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

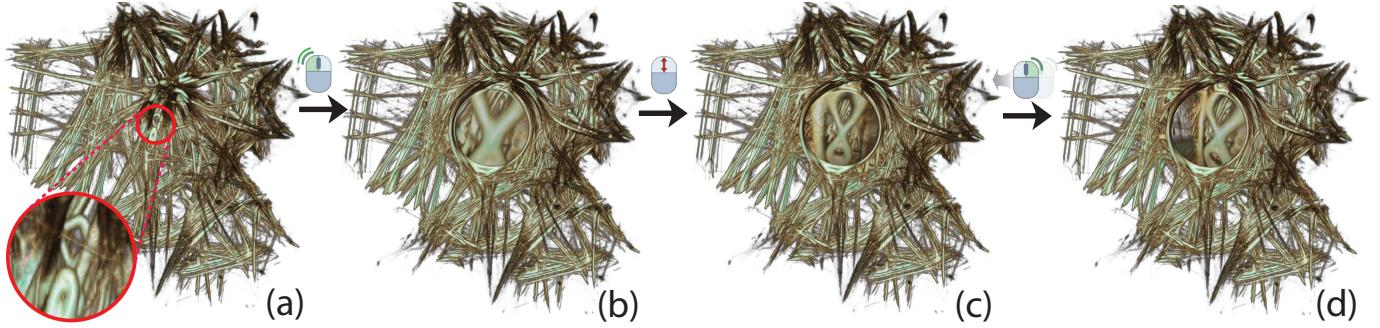


Fig. 4. Inspecting an abnormal aircraft trajectory. (a) The initial view on the trajectories where the abnormal trajectory was spotted. (b) The transition towards a magnified trajectory starts just after the lens tool has been used (double click). (c) The trajectories is magnified. (d) Local Rotations allow to look around the target (right click+ mouse drag toward the desired direction).

inside the luggages frame are very difficult to identify.

**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  $283 \times 189 \times 344$ ) 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. 1b-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. 1e-g).

## 5.2 Fluid flow: A deep-buried spherical vortex

Flow visualization using streamlines has a long history in scientific visualization [?, 2]. When applied to 3D datasets, a key challenge is to balance the streamline density. Low values allow seeing inner regions in the data but can subsample (miss) important patterns. High values show more data but create too much occlusion. We next show how our lens can be used to discover interesting patterns in the second case, *i.e.*, a 3D volume densely filled with streamlines. The dataset, introduced in [12], captures the simulation of water flow in a basin computed on a grid of  $128 \times 85 \times 42$  cells. A set of 4595 streamlines with 183K sample points is next traced by pseudo-random seeding over this vector field. We convert this set of 3D curves (polylines) to a scalar volume by using kernel density estimation (KDE) [28]. Similar techniques have been used to compute density maps of 2D trail-sets [?, 14, 16]. To increase computational speed, we compute the KDE in the frequency space and using GPU acceleration, following [23]. The resulting volumes have a resolution of  $500^3$  voxels and can be directly displayed using DVR (Fig. 6). Note that, given the smoothing effect of KDE, streamlines appear now as finite-thickness tubes rather than pixel-thin curves.

For a first overview, we display the volume using standard DVR. After turning the viewpoint a bit, we notice a dense spherical item inside the dataset (Fig. 6a). To see its shape better, we increase the opacity; however, this immediately increases occlusion so the item becomes invisible. Conversely, decreasing opacity to reduce occlusion makes the item almost transparent. Our lens solves the problem: In the

initial view (Fig. 6a), we point at the object and turn on the lens. This effectively pushes away the occluding streamline bundles, and lets us see that our item is nearly perfectly spherical (Fig. 5). This is something we could not have assessed from *any* viewpoint and with likely any opacity modulation using standard DVR. Our object is a set of densely-packed, low-speed, tightly-turning streamlines that create a ball-like vortex. Interestingly, this spherical vortex has not been discovered by any of the visualization techniques that we are aware of that used this same dataset [?, 11, 12, 23, 29].

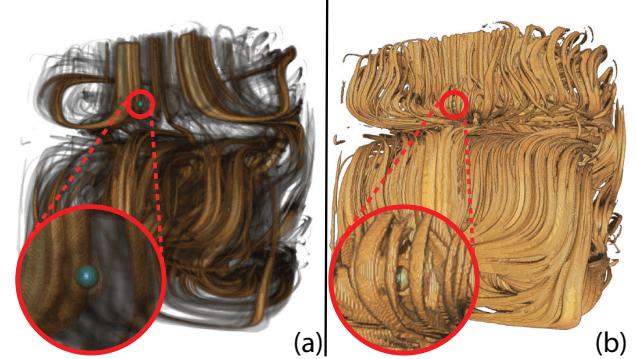


Fig. 6. Two views of a streamlines dataset. (a) The transfer function used allows to spot a spherical dense item inside the volume thanks to transparency. (b) This spherical object become occluded when the opacity of the surrounding whirlpool is increased in order to analyze its shape and behavior. **ALEX: Could and should be merged with Fig. 5.**

## 5.3 Aircraft trajectories: Outliers in the 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  $500 \times 500 \times 500$ ), 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.

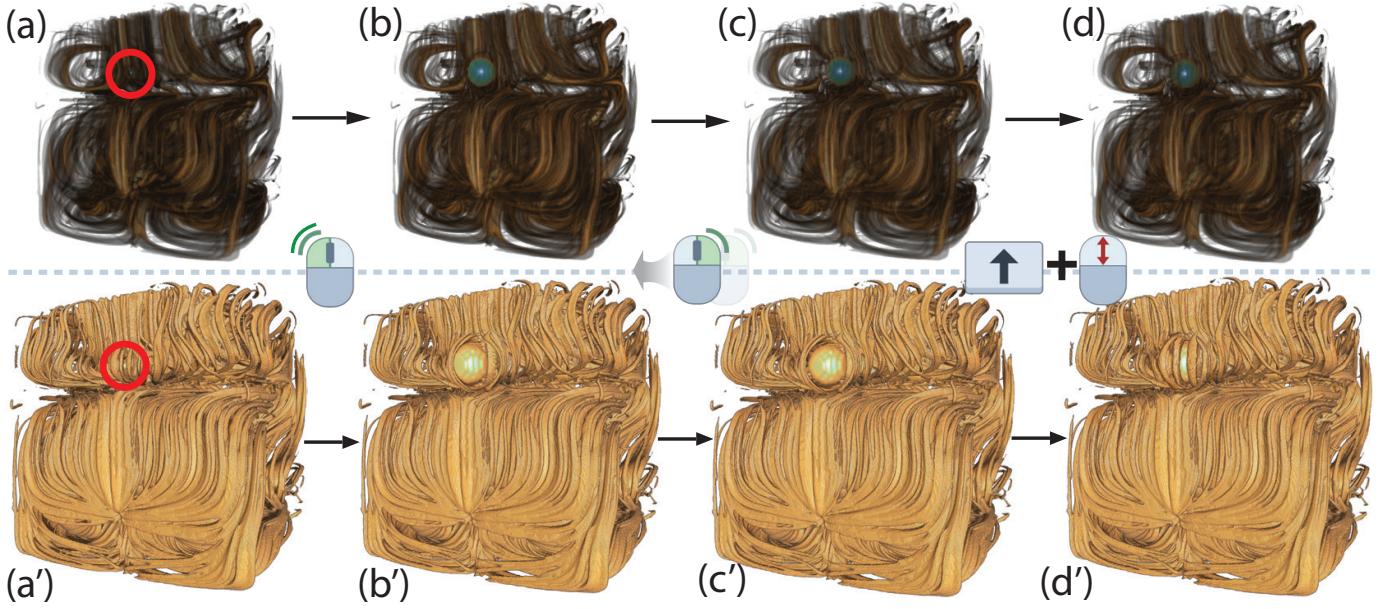


Fig. 5. A spherical item and its surrounding area are inspected using our lens with two different transfer functions. The first part is displayed with transparency in opposition to the part below. The opacity helps to see the shape of the surrounding whirlpool. (a-a') The streamline dataset displayed 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).

## 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.

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,

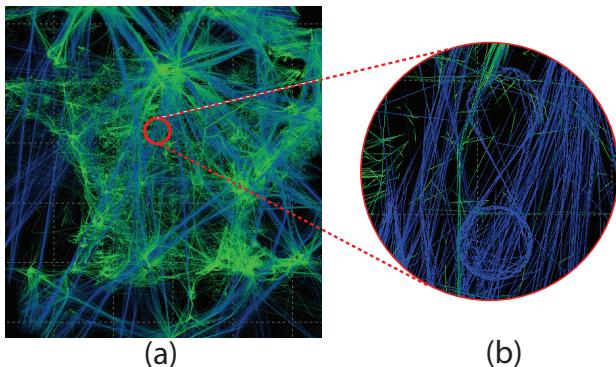


Fig. 7. Standard visualization of one day of recorded aircraft trajectories over France. [19]. (a) An unusual trajectory is spotted but needs some manipulation to be unobstructed. (b) Zoom and color filtering technique to visualize abnormal trajectory of one aircraft performing a loop with an eight shape trajectory. This aircraft corresponds to a tanker waiting for refueling other aircraft.

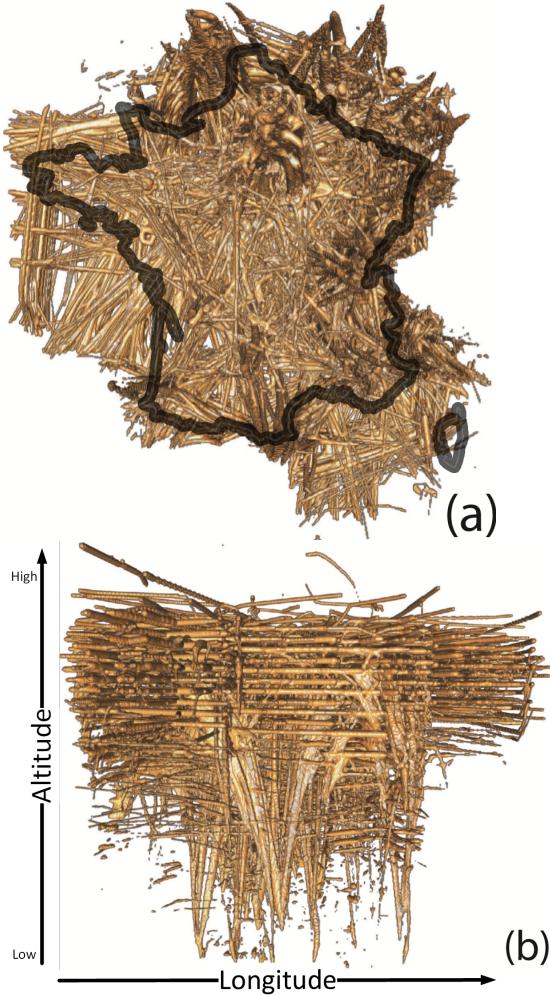


Fig. 8. Visualization of one day of recorded aircraft trajectories over France using our volume rendering framework. (a) The trajectories almost draw the map of France. (b) The trajectories are displayed according to their altitudes.

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 us 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, streamlines, 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] 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.
- [3] 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
- [4] 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
- [5] 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
- [6] 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
- [7] 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

- [8] 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
- [9] 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
- [10] 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
- [11] M. H. Everts, H. Bekker, J. Roerdink, and T. Isenberg. Depth-dependent halos: Illustrative rendering of dense line data. *IEEE TVCG*, 15(6):1299–1306, 2009.
- [12] M. Griebel, T. Preusser, M. Rumpf, M. A. Schweitzer, and A. Telea. Flow field clustering via algebraic multigrid. In *Proc. IEEE Visualization*, pp. 35–42, 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, A. Telea, and O. Ersoy. MoleView: An attribute and structure-based semantic lens for large element-based plots. *IEEE TVCG*, 17(12):2600–2609, 2011.
- [19] 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.
- [20] 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
- [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. FFTEB: Edge bundling of huge graphs by the Fast Fourier Transform. In *Proc. IEEE PacificVis*, 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] 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
- [26] 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
- [27] 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
- [28] B. W. Silverman. Density estimation for statistics and data analysis, 1986.
- [29] A. Telea and J. J. van Wijk. Simplified representation of vector fields. In *Proc. IEEE Visualization*, pp. 263–270, 1999.
- [30] 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.
- [31] C. Tominski. Event-based concepts for user-driven visualization. *Information Visualization*, 10(1):65–81, Jan. 2011. doi: 10.1057/ivs.2009.
- [32] 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
- [33] C. Tominski, S. Gladisch, U. Kister, R. Dachsel, and H. Schumann. Interactive lenses for visualization: An extended survey. *Computer Graphics Forum*, pp. n/a–n/a, 2016. doi: 10.1111/cgf.12871
- [34] 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
- [35] 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
- [36] 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
- [37] 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
- [38] 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