

# Interactive obstruction-free lensing for volumetric data visualization

submission XYZ

**Abstract**—Occlusion is an issue in volumetric visualization as it prevents direct visualization of the region of interest. While many techniques such as transfer functions, volume segmentation or view distortion have been developed to address this, 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 investigate a new technique which maintains the general structure of the investigated volumetric dataset while addressing occlusion issues. With our technique, the user interactively defines an area of interest where an occluded region or object is partially visible. Then our lens starts pushing at its border occluding objects, thus revealing hidden 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 of the area under investigation within the lens. To provide real-time exploration, we implemented our lens using a GPU accelerated ray-casting framework to handle ray deformations, local lighting, and local viewpoint manipulation. We illustrate our technique with five application scenarios in baggage inspection, 3D fluid flow visualization, chest radiology, air traffic planning, and DTI fiber exploration.

**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 with applications in engineering, material sciences, and medical imaging sciences. However widely adopted, and able to handle large datasets at interactive rates, DVR inherently suffers from the problem of *occlusion*: Structures of interest located deep in the volume, called next *targets*, can be hard to spot and/or explore.

To aid with this, various techniques have been designed including transfer functions, segmentation, selection, and clipping. Yet, all such techniques have limitations. *Global* mechanisms, like transfer function editing, can remove both occluders and targets if these have similar densities. In certain applications, carefully designed transfer functions exist and should be used without (significant) modifications to facilitate understanding and user training [48]. *Local* mechanisms like segmentation, selection, or clipping are more effective in manipulating data confined to a given spatial region. Yet, many such mechanisms assume that one can easily and accurately select targets to remove them (occluders) or keep them (occluded). This is hard to do when *e.g.* one does not have direct access to the targets, or when significant 3D interaction is required to select occluder(s).

A different way to handle occlusion is to use *lenses*. These are flexible lightweight tools which enable local and temporary modifications of the DVR to reveal targets while keeping the global visualization context [6, 40, 41]. However, efficiently selecting the target and removing all in-between occluders is still challenging. In detail, most existing occlusion management techniques do not simultaneously meet all following requirements: Rapidly create an unobstructed view of the target (R1), allow an easy change of all parameters (R2), keep the global context (R3), and handle datasets where the target and occluders have very similar densities (R4).

In this paper, we increase the flexibility of lenses for DVR exploration to jointly cover all above requirements. 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, to provide a better perspective on targets. We specifically target the use-case of *partially occluded* objects, where the user has a glimpse of an interesting structure, buried deep in the data, but 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

target, by simple point, click, and scroll operations. Next, we provide several F+C modifications of the lighting parameters, transfer function, and geometry in the focus area to better understand the target. Our technique, implemented using a CUDA-based approach, can be easily incorporated in any generic DVR system.

The paper is structured as follows. Section 2 presents related work in occlusion management, lenses, and deformations for DVR visualization. Section 3 introduces the principle of our lens. Section 4 introduces implementation details. Section 5 presents five application scenarios for our lens in baggage inspection, 3D fluid flow visualization, chest radiology, air traffic planning, and DTI fiber exploration. Section 6 discusses our proposal. Finally, Section 7 concludes the paper.

## 2 RELATED WORK

Previous work on exploring volumetric data to address occlusion issues can be divided into occlusion management techniques and lenses-and-deformation techniques. We next discuss these and also point out existing limitations from the perspective of the four requirements (R1,...,R4) introduced in Sec. 1.

### 2.1 Occlusion management

Many approaches for occlusion management have been proposed [11]. Multiple viewports can be used to see the data from different perspectives [47]. This does not help when the target is strongly occluded from *all* possible viewpoints (R4). Virtual X-ray methods make targets visible by turning occluders invisible or half-transparent [5]. Kruger et al. [24] proposed ClearView, an interactive technique that enables focusing on particular areas in the data while preserving context information without visual clutter by modulating transparency. Correa and Ma [9] proposed visibility-driven transfer functions (TFs) to maximize the visibility of selected data-intervals. Designing good TFs is still challenging and time-consuming (R2) in general: For instance, in baggage inspection, a dissimulation strategy is to hide a threat among same-density objects, so TF editing cannot easily remove occluders but keep the target (R4) [44]. Similar situations occur when aiming to de-occlude a tumor from surrounding similar-density tissue in medical scans [32]. Rezk-Salama and Kolb [34] also considered the voxels’ occurrence on the cast ray, besides their densities and positions. Hurter et al. [19, 20] proposed several lens techniques to remove occluders by deforming (pushing them away) in a focus area, applicable to 2D images, multivariate volumes, and trail sets. Such techniques specify occluders based on data value-ranges, thus share limitations with ClearView and related techniques. Li et al. [27] proposed a system for luggage virtual unpacking where targets are cleared by interactively moving away occluders. This however severely alters the *context* where occluders occur (R3), by altering or removing potentially important

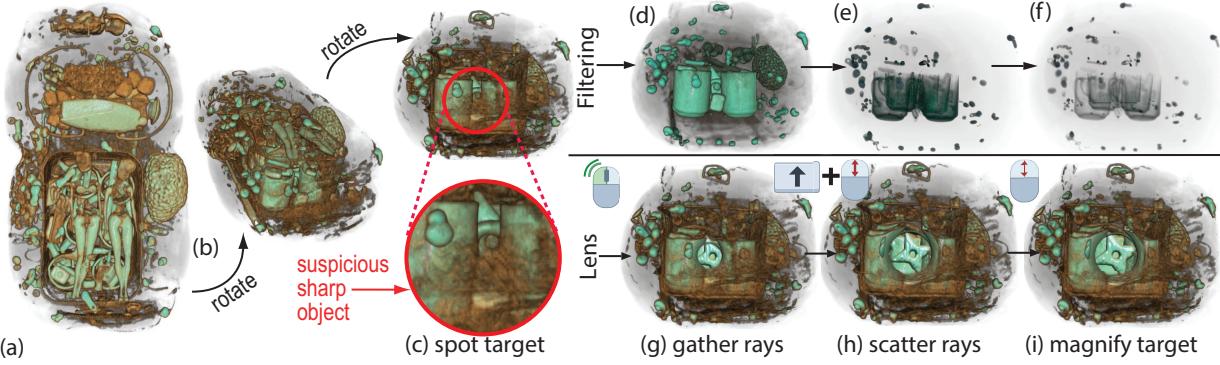


Fig. 1. (a-c) A baggage scan is viewed from different angles. In view (c), a suspicious sharp object is spotted between a set of mugs. (d-f) Filtering densities using a classical 1D opacity transfer function removes progressively more of the occluders (mugs), but also the target. (g) The user applies the lens on the target object (double-click). An animation starts opening the lens, rays are gathered to pass through occluders. Halfway the animation, the object is magnified, but only the area close to the lens is visible. (h) The fish-eye field of view at the end of the animation scatters rays to fully show the target. (i) The lens is increased to magnify the target (mouse scroll).

information, *e.g.*, relative position and connectivity – a clear limitation in this but also other, *e.g.* medical, contexts. Recently, an interactive visualization system was proposed for volumetric data exploration via direct voxel manipulation [44]. Extending such approaches in a DVR setting to more complex deformations or changes of the data in focus is computationally challenging (R1).

## 2.2 Lenses and deformations

An interactive lens is a lightweight tool to solve localized visualization problems by temporarily altering a selected part of the visual data representation [40]. In other words, a lens is a parameterizable selection that alters a base visualization, and as such can effectively provide focus-and-context (F+C) solutions to volumetric data occlusion (R2). Parametrizable lens properties include position, shape, appearance, size, orientation, and selection of included data (focus). The lens *shape* is usually chosen to fulfill the requirements of the application and is strongly linked to the lens function. Most lenses are circular [39] or rectangular [23], a design that we also choose. The JellyLens [31] and smart lenses [37], 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 based on the user’s interest. Position and size can be changed automatically to guide the user toward interesting events in the data [38] or guide the exploration path along interesting events [1]. 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 DVR face spatial selection and occlusion challenges. The Magic Lens [46] addresses these by rendering occlusions with lower opacity and magnifying volumetric pre-computed features interactively or automatically in a pre-segmented dataset. This however does not provide an (interactive) way to deal with similar-density occluders and targets (R1,R4). Besides interactively magnifying targets, our lens frees them from occlusion and allows local modification of the viewpoint, lighting, and TF, to offer many exploration perspectives. The GlyphLens [43] removes occluding glyphs by pulling them aside through animation. While effective, this technique covers 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.

Hsu et al. [15] increase deformation flexibility by using non-linearly sampled rays to smoothly project objects at multiple levels of detail onto a single image. This requires significant computational effort to render a single image from features of interest at different scales (R2). Bruckner and Gröller [4] proposed exploded views for volume data by partitioning the volume into several segments. Correa et al. [8] allow users to physically manipulate the geometry of a data object. McGuffin et al. [28] performed deformations using peeling to see hidden parts of the data. In general, these techniques have the issue of removing

potentially important contextual information surrounding the target when trying to solve the local occlusion (R3).

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 [42]. Other techniques performed deformations using surgical metaphors [7,22] 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 (R3). 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 Detailed contributions

Summarizing the above discussion on the requirements and related work on occlusion management, we propose a new technique which combines high-quality DVR with a fast, versatile, and easy to use, lens to support the interactive exploration of occluded data in volumes. In the classification of view deformations by Carpendale *et al.* [6], 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. Related to 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 which meets the four requirements in Sec. 1;
- the combination of flexible and real-time interactive changing of the focal point, custom bent rays used for DVR, lens deformation, and shading and transfer function in the focus area allow us to provide *on the fly* a range of perspectives of the targets, without having to change the viewpoint or manipulate complex parameters in multiple linked views.

## 3 PRINCIPLE

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 that passes through  $V$  and ends at  $\mathbf{x}$ . In classical DVR (Fig. 2-a), such rays are defined by the eye position  $\mathbf{e}$  and a ray direction unit vector  $\mathbf{d} = (\mathbf{x} - \mathbf{e}) / \|\mathbf{x} - \mathbf{e}\|$ . Consider now a focus point  $\mathbf{f} \in I$  (the lens center) and a lens radius  $R > 0$ . We modify all rays passing through the lens (focus/0 area  $D = \{\mathbf{x} \in I | \|\mathbf{x} - \mathbf{f}\| \leq R\}$ ) in order to de-occlude, magnify, and emphasize a target object. Our ray behavior is divided into three steps: (1) Provide a clear view of the target by moving closer to it and by pushing occluders aside. (2) Set a wide field-of-view (fisheye) to better see the target. (3) Modify the parameters of the lens, lighting, and opacity TF in real time to better explore the target. These steps are detailed next.

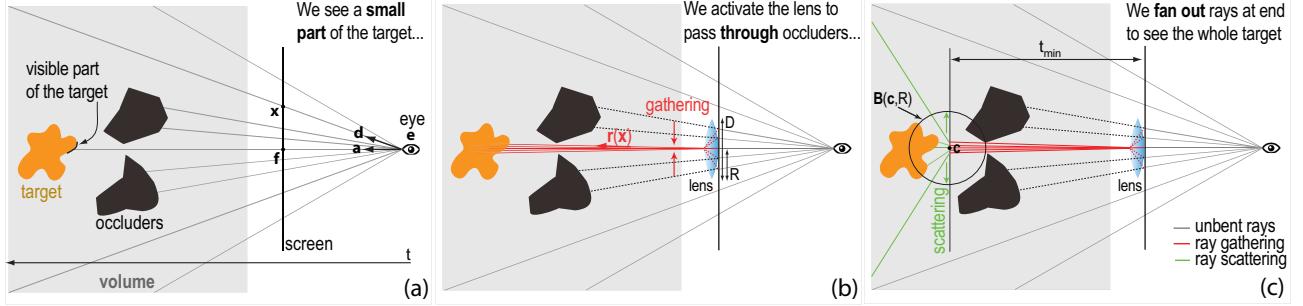


Fig. 2. Principle of the obstruction-free lens. An interesting target object is mostly hidden by occluders in front of it. (a) Classic raycasting result shows a small part of the target. (b) Our lens gathers the rays to avoid occluders. Once close to the target, rays follow again their initial paths.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

### 3.1 Creating an unobstructed view

The scenario we address 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), all rays follow the lens axis  $\mathbf{a}$ , thus, can best pass through obstacles. When  $\alpha = 1$ , rays follow a straight path. Changing  $\alpha$  with the mouse wheel smoothly navigates between the lens effect, *i.e.* opening up a ‘hole’ in the volume to see the target, and classical DVR.

### 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,  $\beta \geq 0$ , adjusted via the mouse scroll wheel while pressing the Shift key (Fig. 2-c), controls the ray scattering: Small values magnify a small volume area close to the ray  $\mathbf{r}(\mathbf{x})$ ; larger values sample more of the volume behind the lens. Intuitively, this is 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, our lens squeezes rays to pass between occluders and next fans them out to explore the target.

### 3.3 Interactive exploration of the target

To achieve a more effective exploration, we can interactively modify several parameters of the DVR and the lens, as follows.

**Lens radius:** The radius  $R$ , telling how big is the ‘hole’ to open up in the volume to see the target, is set via the mouse wheel. The parameters  $\alpha$  and  $\beta$  (affecting the gathering and scattering of rays respectively) are set by the mouse wheel and modifier keys. The value  $t_{min}$  (depth from which scattering starts) is set 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  $\mathbf{a}$  effectively samples the target from many viewpoints, allowing the user to look ‘around’

it to see parts which are not visible from the current viewpoint, but *without* actually changing the viewpoint. This is of high added value, since changing the viewpoint can bring us to a view where the target is fully invisible, so we do not know where precisely to activate the lens any more. Figure 5 shows three such local rotations for the baggage dataset introduced in Fig. 1. From these, we see that the star-shaped target is relatively thick.

**Lighting:** We modify the volumetric Phong lighting parameters to better explore the target, as follows. Let  $\mathbf{c} = \mathbf{e} + t_{min}\mathbf{a}$  be a point at depth  $t_{min}$  along the lens axis, and let  $B(\mathbf{c}, R)$  be a sphere of radius  $R$  around this point (Fig. 2b). We call voxels in this sphere ‘in focus’, and all other voxels in  $V$  ‘out of focus’. Let  $\phi$  be the specular term coefficient, set to a high value (default: one). First, for all voxels  $\mathbf{x} \in B(\mathbf{c}, R)$ , we use a specular coefficient  $\phi(\mathbf{x}) = \phi(1 - d)$ , where  $d = \|\mathbf{x} - \mathbf{c}\|/R$ . For all voxels outside  $B(\mathbf{c}, R)$ , we use  $\phi(\mathbf{x}) = 0$ . Hence, voxels close to the focus point  $\mathbf{c}$  appear highly specular; further away from  $\mathbf{c}$ , voxels get less specular, and voxels out of focus are purely diffuse. Secondly, we allow the user to locally rotate the light vector using the same trackball mechanism as for the lens axis rotation. Let  $\mathbf{l}^{lens}$  be this vector, and let  $\mathbf{l}^{global}$  be the global light vector used by standard DVR. For all voxels in focus, we use a light vector  $\mathbf{l}(\mathbf{x}) = (1 - d)\mathbf{l}^{lens} + d\mathbf{l}^{global}$ . As the user rotates  $\mathbf{l}^{lens}$ , the light direction will visibly change in the middle of the lens, stay constant outside it, and smoothly change in between.

The above two mechanisms combined yield the effect of a moving flashlight turning around a shiny target embedded in a constantly-lit diffuse scene. Figure 3 shows these mechanisms for a chest CT dataset containing a deeply buried tumor (the dataset and use-case are described in detail in Sec. 5.3). We see how turning the light highlights small-scale details on the target surface (tumor) without changing the viewpoint or lens location. Moreover, the high specularity in the lens attracts the user’s attention to this area; the diffuse lighting outside the lens put less emphasis on the context area.

**Opacity:** We modify the opacity transfer function along a similar idea as for lighting (Fig. 4). Let  $TF_o^{global} : \mathbf{R} \rightarrow [0, 1]$  be the user-chosen opacity function used globally for the volume. Let  $\Gamma$  be a Gaussian pulse of unit height centered at the average density value  $\bar{\rho}$  in  $B(\mathbf{c}, R)$  and with standard deviation  $\sigma$ . We estimate  $\bar{\rho}$  and  $\sigma$  by considering the density  $\rho$  at 150 points randomly sampled inside  $B(\mathbf{c}, R)$ . Then, for voxels in  $B(\mathbf{c}, R)$ , we use an effective opacity transfer function  $TF_o = TF_o^{global} + (1 - d)\Gamma$ . For voxels outside  $B$ , we use  $TF_o^{global}$  (standard DVR). The effect is that voxels in  $B$  become more opaque, thus more visible. Voxels with the same densities but outside  $B$  use the default transfer function, which can make them transparent. This makes voxels with similar densities either opaque (if close to the target, thus of interest) or transparent (if *e.g.* in front of the target, thus occluding). Figure 3 has been generated this way.

### 3.4 Smooth transitions

If we bend rays passing through the lens pixels  $D$  (Eqns. 1 and 2) and trace rays starting at pixels in  $I \setminus D$  as straight lines, discontinuities appear at the lens borders. We solve this as follows. Let  $\mathbf{p}(\mathbf{x}, t)$  be

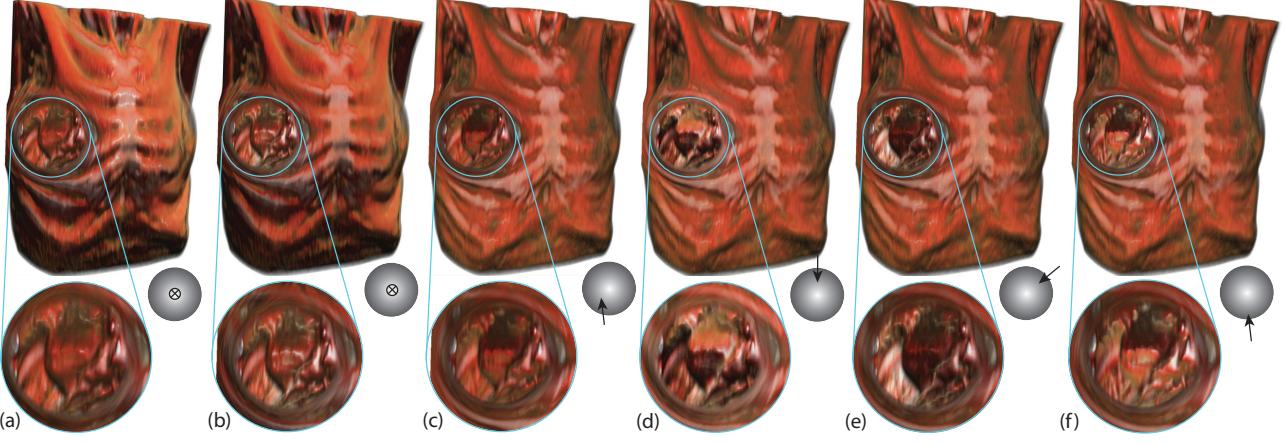


Fig. 3. Changing lighting parameters in the lens. (a) Constant global specular coefficient. (b) Specular coefficient high in the lens and low outside. (c-f) Changing the in-lens light vector yields the effect of a flashlight rotating around the target. The ball icons illustrate the local light vector direction.

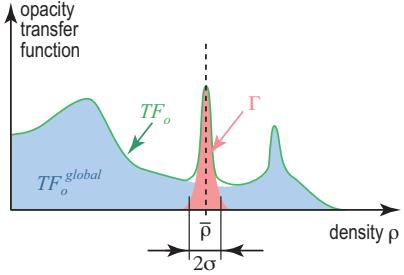


Fig. 4. Construction of local transfer function  $TF_o$ . See Sec. 3.3.

the voxels along a lens ray starting at screen pixel  $\mathbf{x}$ , as computed by Eqn. 2. Let  $\mathbf{p}^{line}(\mathbf{x}, t)$  be the voxels along a straight-line ray starting at  $\mathbf{x}$ , i.e., computed using  $\alpha = 1$  and  $\beta = 0$  in Eqns. 1 and 2 respectively. For every value  $t$  along every such ray, we compute the interpolated ray  $\tilde{\mathbf{p}}(\mathbf{x}, t) = (1 - f(d))\mathbf{p}(\mathbf{x}, t) + f(d)\mathbf{p}^{line}(\mathbf{x}, t)$ , where  $d$  is the distance of  $\mathbf{x}$  to the lens axis (normalized to unit by dividing it by  $R$ ) and  $f : [0, 1] \rightarrow [0, 1]$  is an interpolation function. Next, we use the rays  $\tilde{\mathbf{p}}(\mathbf{x}, t)$  to compute the DVR by standard composition. This way, rays effectively vary smoothly from their bent versions (close to the lens axis) to straight lines (outside the lens). Setting  $f(d) = d^2$  keeps the interpolation transitions close to the lens border, so most of the lens is dedicated to show the desired fisheye effect.

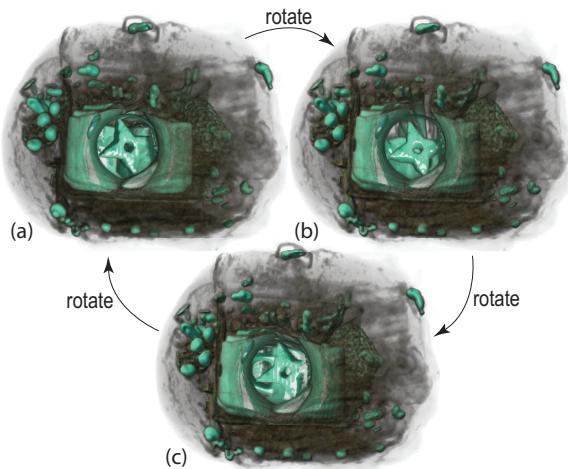


Fig. 5. Performing local rotations in the lens allows better seeing the shape and thickness of the partially occluded target object (ninja star).

Separately, we use a slow-in/slow-out animation [10] to introduce

the lens effect. When activating the lens, we vary  $\alpha$  and  $\beta$  from their defaults ( $\alpha = 1, \beta = 0$ , i.e. straight-line classical DVR) to their actual user-set values, compute the volume rendering on-the-fly, and display the resulting images. The effect resembles gradually opening a hole in the volume – see the associated video. The speed increase at the start of the animation helps one to quickly see what is revealed in the lens; the decreasing speed at the end helps seeing where the pushed-away occluders actually go. This also gives some semantic to the moving shapes, allowing one to interpret the motion as a magnification of a target, and to keep the focus on visual entities during this transition. When deactivating the lens, we play back the animation in the opposite sense, which suggests closing the opened hole in the volume.

#### 4 IMPLEMENTATION

We implemented our occlusion-free lens by modifying a standard DVR ray caster, publicly available in NVIDIA CUDA’s SDK [30]. We modified this ray caster to incorporate the new ray definition (Eqns. 1 and 2), the lens effect, and the local per-voxel Phong lighting parameters, all controlled via keyboard and mouse. On a PC with 16 GB RAM and a GeForce GTX TITAN X card, we achieve 15 frames per second for volumes up to  $512^3$  voxels at a  $1900 \times 1200$  pixels screen resolution. Currently, we use a compositing ray function. Any other functions can be directly used with no restrictions. All in all, adding our lens to an existing ray caster should pose no significant implementation problems.

#### 5 APPLICATION SCENARIOS

We next demonstrate our obstruction-free lens via five use-cases considering scalar density volumes from baggage inspection, 3D flow simulation, radiology, air traffic planning, and diffusion tensor imaging.

##### 5.1 Baggage inspection: An unusual blunt object

In airports, security agents deal with volumetric data exploration during baggage inspections. While automatic systems can detect densities of harmful substances such as C-4, TNT, and nitroglycerin, and even some prohibited articles like classical firearms and knives, unusual threats (prohibited objects) are hard to find. Four main concealment strategies exist [44]:

**Superposition:** A threat may be sheltered among dense materials. While possible to see through such a ‘shield’ using high penetration (enhanced X-ray power) or image processing (contrast improvement) techniques, such techniques are not universally available and also require fine-tuning many parameters, which slows down inspection.

**Location:** Objects located in the corners, edges, or in the luggage’s frame are very hard to spot.

**Dissociation:** One can conceal a threat by spreading its parts in the luggage, e.g., by disassembling a weapon and scattering its parts.

**Lure:** A minor threat (lure) like a small scissors will be clearly visible and catch the security agent’s attention who can miss the real threat.

Baggage labeled as suspicious by human inspection or automated scan heuristics need to be checked by human agents. Besides time-consuming physical unpacking, one can use ‘virtual unpacking’ tools that segment the 3D scan by a density-based confidence measure and next move the segmented objects away by animation to reduce occlusion [27]. Such systems have been patented and used in production [26]. However, when the automatic segmentation is not optimal, the user must manually change its parameters, repeat the segmentation and animation, which goes back to being time consuming.

Consider the baggage scan in Fig. 1 ( $283 \times 189 \times 344$  voxels, dataset obtained from an actual airport scan). Automatic baggage inspection systems will not detect anything suspect here. However, while visually exploring this baggage from different angles (Fig. 1a-c), we see an object hidden between a set of mugs. To reduce occlusion, a common solution in baggage inspection is to filter materials by density in order to show or hide subsets of the volume. However, for our dataset, the suspect target has almost the same density as the surrounding mugs, so removing the latter also removes the target (Fig. 1d-f). Using the obstruction-free fish-eye lens helps here: Clicking on the sharp detail visible in Fig. 1c first gathers rays so they pass through the low-density zone between the mugs (Fig. 1f). The animation that opens the lens (Fig. 1e-g) reveals an unobstructed view of the target. However, this shows only a small part of the target. Scattering rays next fully reveals the target (Fig. 1h). Adjusting the lens size shows a more detailed view of the target (Fig. 1i). Next, locally turning the viewpoint around the target (Fig. 5) allows the agent to decide that the target is a shuriken (Japanese ninja star weapon). Since the object is very thick and blunt (see Fig. 5), it is not an actual weapon, thus not a threat.

We got feedback on our obstruction-free lens from three baggage security operators. According to them, our tool can provide them a better perception of the items inside the baggage as compared to the classical 2D single-viewpoint X-ray machinery they routinely use. However, our tool should not be used for the typical carry-on baggage inspection which has an very small allowed inspection time of about 15 to 20 seconds in order to keep a smooth traffic flow. Our tool is much more interesting for inspecting checked-in baggage, where inspection time-windows are up to 3 minutes. The added value for this use-case is also higher: Opening up checked-in baggage for manual inspection is much more complicated and time-consuming than for carry-on baggage. Moreover, the only system for inspecting checked-in baggage currently in use is a scanner that aims to automatically detect suspicious objects in X-ray imagery; this system suffers from false positives, so having a manual examination tool like ours can quickly eliminate such false positives, and thus the unnecessary delays caused by opening up checked-in baggage.

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

Flow visualization using streamlines has a long history in scientific visualization [3, 29]. 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 [14], 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) [35]. Similar techniques have been used to compute density maps of 2D trail-sets [16, 18, 45]. To increase computational speed, we compute the KDE in the frequency space and using GPU acceleration, following [25]. 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 target and turn on the lens. This effectively pushes away the occluding stream bundles, and lets us see that our item is nearly perfectly spherical (Fig. 6b). 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 [13, 14, 25, 36]. To make sure our target is spherical, we view it in the lens from different directions, by interactively changing the ray directions in the lens (Fig. 6c). Finally, we can close the lens but keep the target magnified (Fig. 6d).

## 5.3 Chest scan: A hard to see tumor

In our third use-case, we consider a contrast chest CT scan ( $512 \times 512 \times 110$  voxels) of an elderly patient having a sizeable lung tumor. The tumor was detected at a CT scan ordered by the pulmonologist in charge of the patient, after the patient reported acute chest pain. Typical examination of these scans by the pulmonologist and radiologist in charge involves slice-based views. Figures 7a-c and 7d-f show two such slice sets (axial, coronal, and sagittal views), produced using typical lung, respectively mediastinal, contrast presets. Although the tumor is visible in all these views, its exact shape, morphology, and connection to the lung walls are not easy to assess. Finding such details on the tumor is essential, explained both doctors in charge, for determining the TNM score and also planning treatment. Using standard DVR makes the tumor and its 3D position partially visible (Fig. 7f). However, occlusion from the rib cage and other tissues is still present. Using both TF presets and manually changing the TFs in the 3D Slicer tool [33] used to create the DVR could not help de-occluding the tumor without making (parts of) it transparent. This is also visible in the slice images in Fig. 7a-f, where the gray values for the tumor and surrounding skin-and-muscle tissue on the rib cage are very similar. This density similarity is due to the fact that the tumor had grown rapidly in a short time span, explained the pulmonologist. As such, the tumor started necrotizing, which filled it with fluids, making its density very similar to that of the obstructing (skin and muscle) tissue. In conclusion, one cannot remove such occluding tissue in a classical DVR setting by opacity TF manipulation without also removing the tumor. This situation makes examining this specific tumor harder than for regular cases.

We next used our lens to examine the tumor. Sample snapshots obtained in this process are shown in Fig. 3. Comparing these with standard DVR (Fig. 7d), several points can be made. First, the tumor is significantly more visible when using the lens, both in terms of removing the occluding tissue and in terms of the tumor’s opacity – compare the inset in Fig. 7d with the images in Fig. 3. Secondly, relighting the tumor from various directions allows one to see small-scale morphological details such as the tumor’s surface shape and its connection via protuberances and veins with the lung walls.

To assess the added-value of our lens, we asked the two medical specialists (pulmonologist and radiologist) involved in treating the patient that this dataset came from to study the lens’ features and state its potential advantages and/or limitations as compared to standard slicing and DVR techniques they use in their practice. Both specialists have over 10 years of medical experience in treating lung cancer, and routinely use several slicing and DVR software tools. They work in a private hospital in Belgium and are not actively associated to medical imaging research. Moreover, our (authors’) identities were hidden from them, by using a third person in the communication. The provided input can be summarized as follows: The occlusion-free lens is definitely easier and faster to use than classical DVR and/or slicing techniques. It is especially more effective than these to get a quick, first impression of a deep buried anatomical detail. Changing the lens’ parameters by direct interaction is as simple as changing window/level functions in a

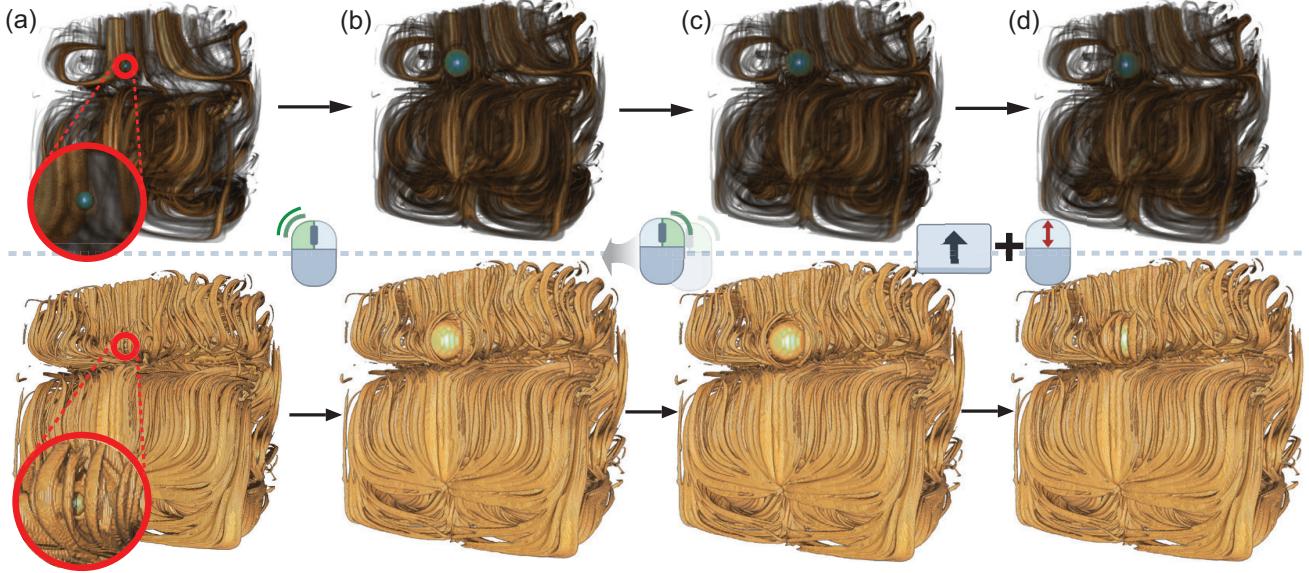


Fig. 6. Flow volume exploration with two different opacity transfer functions (top and bottom rows). In viewpoint (a), we notice a small high-density spherical item. (b) We apply the lens at that location (double click). (c) The directions of rays in the lens are changed to see the whole target in the lens (right click + mouse drag change direction). (d) The lens is gradually closed while keeping the focus area magnified (shift + scroll).

typical slice-based visualization, and is definitely simpler than tuning typical DVR parameters to obtain similar results. This ‘enticies’ the user to explore, which is a good aspect. The fact that the lens minimizes viewpoint change (volume rotation), *i.e.*, after a suitable viewpoint was found from which a (small) part of the target is visible, one doesn’t need to change this viewpoint, is a strong feature, as 3D viewpoint changes are disruptive and cost time. This is important in a cost-aware environment where specialists have very limited time (10–15 minutes) to assess a CT scan. However, the lens cannot and should not replace classical slice-based investigation, which shows small-scale details better. This is especially important when examining small-size lesions, tumors, or other similar anatomical features, that the lens will arguably not be able to help with, as these are too small in the first place to attract the attention of the examiner looking at a standard DVR rendering. In the context of the current dataset (Fig. 7), the lens was useful to both confirm the TNM score (T3 grade tumor, 6.5 cm in size) found via the 2D slices, but much more so for understanding how and where the tumor is connected to surrounding tissue, which is very hard to do using only 2D slices.

#### 5.4 Aircraft trajectories: Outliers in the French sky

We next consider a task from the air traffic planning field – detecting and studying outliers in large-scale datasets containing tens of thousands of 3D (latitude, longitude, height) trails of aircraft over a given spatio-temporal region [17]. Typically, such datasets are displayed using 2D (latitude, longitude) plots where opacity encodes the spatial density of flights. Fig. 8-a shows one day of recorded aircraft trajectories over the French air space using this 2D technique. Fig. 8(b) shows a detail zoom-in of this dataset, where we can see an abnormal – that is, far from straight or slightly curved – aircraft trajectory: A tanker aircraft performed an eight-shaped loop as it was waiting to refuel other aircraft. Revealing such patterns using 2D techniques, *e.g.* [21], is very hard. In particular, it is hard to de-occlude these patterns from the overall context of criss-crossing aircraft trails, even when one knows their 2D spatial location.

Our lens can help for this task, as follows. We first convert the set of 3D trails to a  $500^3$  density volume, using KDE as described for the streamline use-case (Sec. 5.2). Examining this volume using standard DVR allows us to see that there is an outlier (that is, not straight) trail at some point in space, see curved patterns in Fig. 9a. By activating the lens on this area and interactively tuning the target depth  $t_{min}$  (since we don’t know the height of this trail), we can quickly

obtain a view where the outlier trajectory is in focus and the occluding ones are pushed away (Fig. 9a). Finally, just as in the other examples presented so far, the user can quickly change the magnification factor and view direction to better study this trajectory in context (Fig. 9b-d). From these images, one directly and easily sees that the outlier trajectory has an eight shape. Revealing this outlier trail using standard 2D visualization techniques [21] costs several minutes. Doing the same using our lens approach costs under one minute, for the same users. Additionally, if we compare Figs. 8b and 9b-d, we argue that the eight-shape of the outlier trajectory is much more prominent, and thus recognizable, in the latter images (using our lens) than in the former ones. Last but definitely not least: The 3D volume rendering approach that our lens is based on explicitly encodes the flight height information, so our lens can use it by interactively tuning the depth value  $t_{min}$  where the lens is focused. This is not possible with 2D techniques which ignore this depth dimension.

We validated our findings with an air traffic data scientist with more than 10 year experience in air traffic control and air traffic planing. She confirmed that this specific eight-shaped trail in Fig. 8(b) is an actual aircraft which performed waiting loops and acted as a fuel supplier for military aircraft. Other comments included the following: Compared to standard 2D visualization techniques, our tool makes detecting outliers easy since there is no need for complex manipulation to reveal such outlier trails. Also, the user does not have to deal with color and alpha mapping parameter-tuning to make specific outliers emerge. Separately, trail visualization easily creates many occlusions leading to either fully opaque areas or too much local overlap, both which hinder seeing and examining specific trails. Our lens does help such cases by distorting the space to locally remove such occlusions. All in all, in the studied dataset (Fig. 8), the lens was specifically useful since, for high transparency, one would not detect the outlier trail, while for low transparency, one would get a hint of the outlier’s existence, but not see it in detail due to too much occlusion; the lens allows using low transparency, but removes the clutter caused by it to reveal the outlier.

#### 5.5 Brain fibers: Uncluttering the bridge

Finally, we consider a last example related to the exploration of fiber tracts visualized as streamlines of the major eigenvector of a diffusion tensor imaging (DTI) field. Such datasets shows a spatially complex structure which makes them hard to explore [2]. In particular, fiber tracts are spread volumetrically over the entire extent of the brain, and create tangled patterns which make it very hard to see deep within the

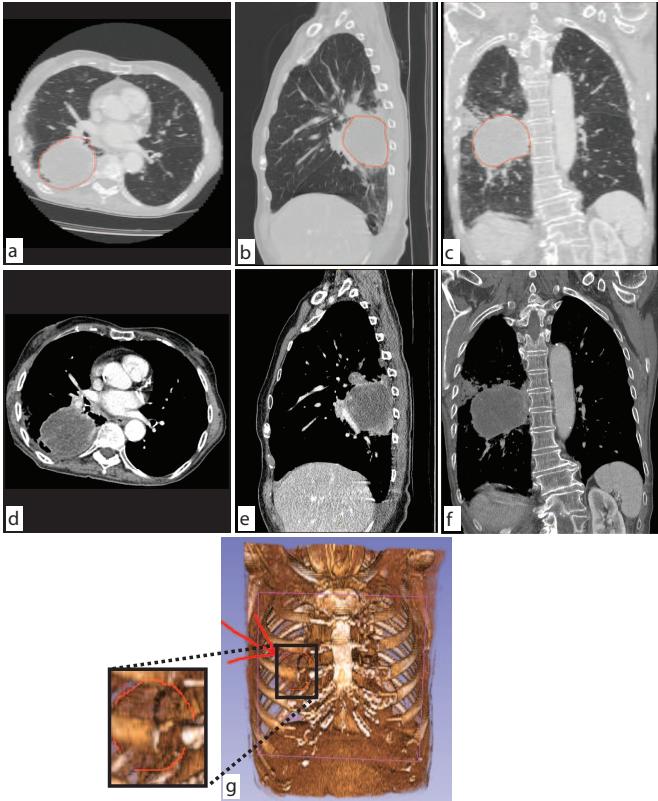


Fig. 7. Lung tumor visualization using slices (a-c) and standard DVR (d). Annotations are manually added by the examiner to delineate the tumor location. Images constructed using the 3D Slicer tool [33].

emerging structures. DVR techniques are often used to render such fiber tracts, one of the advantages being that close fibers get visually ‘merged’ to reveal spatially coherent structures, an effect which is not possible when fibers are rendered using polyline (streamline-like) representations. However, DVR methods also create more occlusion, thus difficulties in seeing structures deep within the volume.

For this use-case, we use an  $128 \times 128 \times 51$  DTI volume (same dataset as in [12]). From this volume, we traced 150352 fibers seeded in, and going over, regions of high fractional anisotropy. Next, we filtered out fibers shorter than 2mm, yielding a total of 120593 fibers to display (6.4M sample points). Next, we converted this fiber-set to a  $512^3$  density volume, using kernel density estimation with a 3D isotropic kernel of radius 15 pixels. The conversion process is identical to the one used for the streamlines in Sec. 5.2. Figure 10a shows the result, rendered with DVR, with a simple opacity function mapping the fiber density. While terminal fibers are well visible, it is impossible to discern anything inside the volume. Activating the lens in the middle of the volume opens a hole through which a small part of the *corpus callosum*, the fiber bundle wrapping the bridge that connects the two hemispheres, becomes visible. To obtain an even better view hereof, we slightly decrease the opacity (Fig. 10c). The *corpus callosum* is now clearly visible, appearing as a compact structure, due to the KDE blending of neighbor fibers. Note that obtaining such a view on the *corpus callosum* using only DVR would be very hard, since transfer functions would either render separated (non-merged) fibers, or else make the fibers surrounding the structure of interest too thick and occluding.

There is one notable difference to this scenario as compared to all previous ones. In all earlier cases, the standard DVR of the data (that is, without the activated lens) showed us a partial small cue of the structure of interest within the volume, and we used the screen-space location of this structure as the center point where to activate the lens. In this last scenario, there is no clear point within the original DVR image

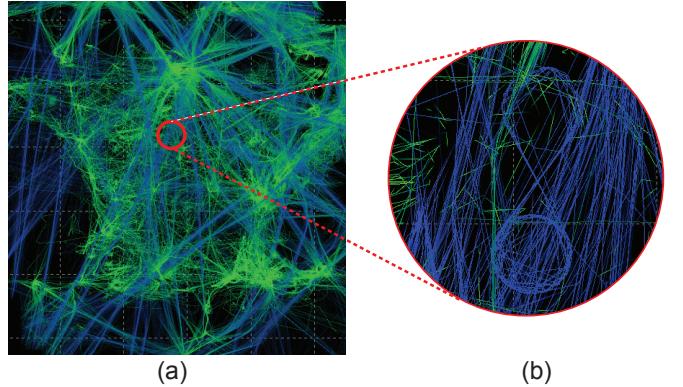


Fig. 8. Visualizing one day of aircraft trajectories over France [21]. (a) Overview of all trails. (b) Zoom, filtering, and color mapping techniques are used to highlight an outlier trajectory of an aircraft performing an eight-shaped loop. Revealing this outlier costs significant user effort.

(Fig. 10a) from which the *corpus callosum* is even partially visible, due to the high opacity given by the transfer function used. As such, the user activates the lens here at *any* desired point to peek inside, and towards the center of, the volume. However, given the nature of the data, the structure of interest is quite easily visible from most such viewpoints (see lens inset in Fig. 10b). Once its presence is revealed, the user can next adjust the viewpoint and/or the opacity transfer function to get an optimal view on the target, such as the one shown in Fig. 10c.

## 6 DISCUSSION

Several points of our lens proposal are worth discussing, as follows.

**Lens activation:** For our lens to be effective, the user needs to perceive at least a *part* of the target in a normal DVR image. Obviously, if the target is fully occluded, either due to transfer function or viewport settings, this is not possible. However, we argue that in such cases, no other visualization tool can help the user find and explore such a target, since its presence is completely absent in the rendered image. Conversely, when at least a fragment of the target is visible, no matter how small, the user can point the lens at that fragment, after which the lens’ mechanism performs the de-occlusion. Hence, the usage pipeline we support works as follows: (1) The user changes the viewpoint and/or transfer function until she sees some fragment of the target of interest; (2) The user points the lens at that fragment, without needing to change the viewpoint or the transfer function, and fully reveals it; (3) Optionally, the user adjusts the lens parameters to better examine the target.

**Lens shape:** Occluders are pushed away, and deformed, isotropically (Secs. 3.2, 3.4). The advantage of this simple lens model is that it requires a single parameter, the lens radius  $R$ , which makes its usage easy. The deformations evolve smoothly from the lens center (maximal) to outside the lens (no deformation), see Sec. 3.4, which effectively blends the local (in lens) focus with the global (out of lens) contexts ( $R_3$ ). However, a side-effect is that this mechanism compresses the deformed occluders strongly close to the lens border, making them hardly visible when the lens is fully active. A possible refinement would be to reduce the deformation of the pushed-away occluders while still pushing them away, thereby improving the focus+context effect ( $R_3$ ). However, this would occlude the areas outside the lens with these undeformed, but pushed-away, occluders, so it will basically shift occlusion from *inside* the lens to *outside* and close to it. Finding an optimal balance between minimal deformation (so one can recognize the pushed-away occluders) and minimal clutter (so these occluders do not destroy the lens context) is an interesting topic for future work.

**Parameter setting:** Our lens depends on several parameters: the 2D lens center  $\mathbf{f}$ , lens radius  $R$ , lens axis direction  $\mathbf{a}$ , local light direction  $\mathbf{l}^{lens}$ , scattering start-distance  $t_{min}$ , and gathering and scattering parameters  $\alpha$  and  $\beta$ . As explained in Sec. 3, all these parameters can be easily

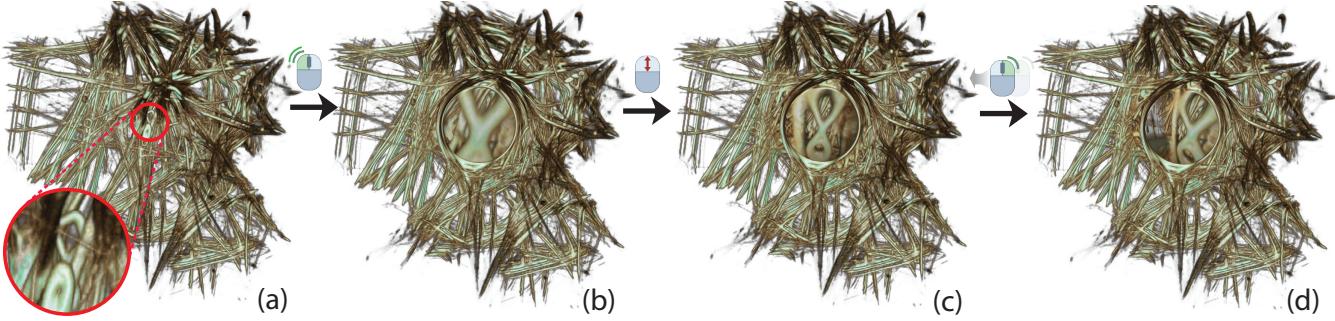


Fig. 9. Inspecting an abnormal aircraft trail. (a) The abnormal trail is spotted in an all-trails view as it is highly curved while all other trails are relatively straight. Activating the lens at the outlier location (b) and changing the magnification factor (c) reveals the trail's eight-shape. (d) Rotating the viewpoint provides spatial insight on the embedding of the outlier in the surrounding trails.

controlled via a mouse-driven virtual trackball, key modifiers, and the arrow keys. While this seems complex at first sight, performing such operations is in fact quite simple, since the lens works at interactive frame rates (15 frames per second), so the user can quickly tune the parameters and see their effect. Moreover, all parameters start with typically good preset values (see Sec. 3 for details). A possible refinement would be to pre-segment the target of interest, based on user-given values for  $f$ ,  $R$ , and  $t_{min}$ , thereby determining  $\beta$  automatically. This would help the first requirement (R1) which consists in giving a fast unobstructed view of the target. However, even if this were present, we believe that manual control of the scattering  $\beta$  is important to allow users to choose their most suitable field-of-view angle. In fact, this flexibility allows a better exploration of the local context (R2).

**Implementation:** The entire lens is implemented by modifying the ray trajectories constructed in the inner loop (per-pixel raycasting) of a typical DVR raycaster [30]. Apart from this modification, we also change the per-voxel lighting function and transfer function based on the voxel location with respect to the lens and the parameters supplied by user interaction (Sec. 3.3). As mentioned, such changes are limited and should be easily applicable to any raycaster. Since these changes work in a per-voxel or per-ray fashion, they are directly applicable to raycasters which parallelize computations for different ray sets.

**Limitations:** Several limitations of our approach must be mentioned too. First and foremost, as explained, we cannot de-occlude a target that the user is not aware of (does not partially see) from any viewpoint. However, we argue that such a use-case is not in scope for a lens. At a higher level, many lens mechanisms exist in the literature, as discussed in Sec. 2. While we have argued that, to our knowledge, none of them simultaneously supports requirements R1,...,R4, comparing such mechanisms with our lens for specific use-cases and datasets is still an important test for the end-to-end effectiveness of our proposal. We have not covered this point as obtaining several implementations of such lenses in the same interactive framework or tool, for comparison fairness, is very challenging. However, this remains an important open point for future work – both for our proposal but also for all other volumetric lens proposals in the literature. Related to this, we have performed three user evaluations involving specialists in airport baggage security (Sec. 5.1), pulmonology (Sec. 5.3), and air traffic control (Sec. 5.4). In all cases, the users were not involved in this work, nor with other work of the authors. However, the set-up of these evaluations stays at the level of formative user experiments. To confirm and refine the obtained (positive) findings, more formal user studies are needed, which we plan to cover next.

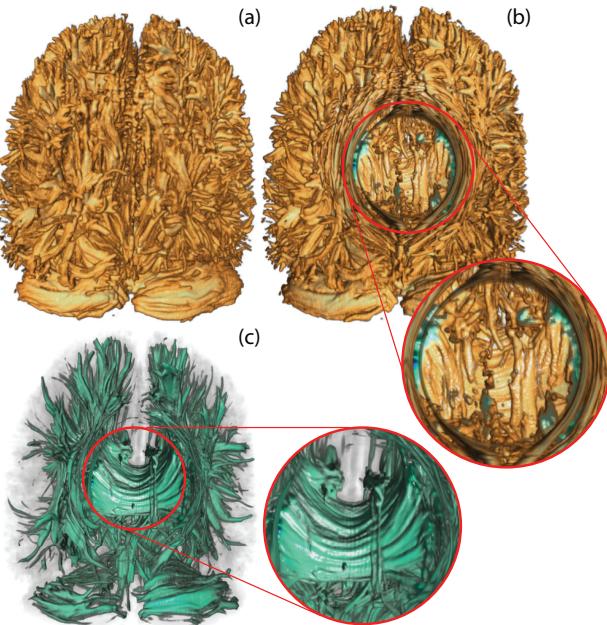


Fig. 10. Revealing the *corpus callosum* in a DVR of a set of DTI tracts.

## 7 CONCLUSIONS

In this paper, we presented a new fish-eye-like context-and-focus lens that addresses the occlusion problems inherent in scalar volume rendering. The principle of our lens consists in first gathering (squeezing) rays so that they easily pass through occluding densities (given a user-specified opacity transfer function) and next scattering (fanning out)

rays to best sample the target of interest. Our lens can be directly applied to any DVR raycaster and scalar volume dataset. Its main constraint is that the user should be able to find a viewpoint from which the target of interest, deep buried in the data, is at least slightly visible. We also present several modifications of the local rendering parameters within the lens (view direction, lighting parameters, opacity transfer function) that aim to both better separate the focus (lens) from the context (volume) and also allow more detailed examining of the target. Our lens is easy to use – all its parameters are controlled via direct mouse-and-keyboard interaction – and can be efficiently implemented atop of a standard GPU ray caster. Our lens is especially useful for highlighting structures of interest which are both deeply embedded in volumetric data and cannot be revealed by standard transfer function manipulations due to similar densities in the occluders and target. We demonstrate these points using five use-cases involving datasets from baggage detection, fluid visualization, air traffic control, and chest radiology, and DTI fiber tracts.

Several improvements to our proposal are possible, as follows. First and foremost, heuristics can be sought to link all our free parameters (lens size, focus depth, interpolation between focus and context) directly to the volume data, so the user interaction is minimized and therefore exploration efficiency is increased. Secondly, our lens could be extended to different types of volumetric datasets, such as multivariate (vector, tensor) fields. Last but not least, a formal wider-scale evaluation of how the lens addresses more specific tasks, and how it compares to existing tools for these tasks, such as other lens types, is a goal we aim to pursue next.

## ACKNOWLEDGMENTS

To be inserted in the final version of the paper.

## REFERENCES

- [1] J. Alvina, C. Appert, O. Chapuis, and E. Pietriga. RouteLens: Easy route following for map applications. In *Proc. Intl. Working Conference on Advanced Visual Interfaces (AVI)*, pp. 125–128. ACM, 2014. doi: 10.1145/2598153.2598200
- [2] Y. Assaf and O. Pasternak. Diffusion tensor imaging (DTI)-based white matter mapping in brain research: A review. *Journal of Molecular Neuroscience*, 34(1):51–61, 2008.
- [3] A. Brambilla, R. Carneky, R. Peikert, I. Viola, and H. Hauser. Illustrative flow visualization: State of the art, trends and challenges. In *Proc. EuroVis – State of the Art Reports*. Eurographics, 2012. doi: 10.2312/conf/EG2012/stars/075-094
- [4] S. Bruckner and M. E. Gröller. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1077–1084, Sept 2006. doi: 10.1109/Transactions on Visualization and Computer Graphics.2006.140
- [5] M. Burns and A. Finkelstein. Adaptive cutaways for comprehensible rendering of polygonal scenes. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 27(5):154:1–154:7, 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, 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/Transactions on Visualization and Computer Graphics.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/Transactions on Visualization and Computer Graphics.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, 2011. doi: 10.1109/Transactions on Visualization and Computer Graphics.2010.35
- [10] P. Dragicevic, A. Bezerianos, W. Javed, N. Elmquist, 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. Elmquist and P. Tsigas. A taxonomy of 3D occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1095–1109, 2008. doi: 10.1109/Transactions on Visualization and Computer Graphics.2008.59
- [12] M. H. Everts, E. Begue, H. Bekker, J. B. T. M. Roerdink, and T. Isenberg. Exploration of the brain’s white matter structure through visual abstraction and multi-scale local fiber tract contraction. *IEEE Transactions on Visualization and Computer Graphics*, 21(7):808–821, 2015.
- [13] M. H. Everts, H. Bekker, J. Roerdink, and T. Isenberg. Depth-dependent halos: Illustrative rendering of dense line data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1299–1306, 2009.
- [14] 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.
- [15] 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
- [16] C. Hurter. Image-based visualization: Interactive multidimensional data exploration. *Synthesis Lectures on Visualization*, 3(2):1–127, 2015.
- [17] 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.
- [18] C. Hurter, O. Ersoy, and A. Telea. Graph bundling by kernel density estimation. *Computer Graphics Forum*, 31(3pt1):865–874, 2012.
- [19] C. Hurter, R. Taylor, S. Carpendale, and A. Telea. Color tunneling: Interactive exploration and selection in volumetric datasets. In *Proc. IEEE PacificVis*, pp. 225–232, March 2014. doi: 10.1109/PacificVis.2014.61
- [20] C. Hurter, A. Telea, and O. Ersoy. MoleView: An attribute and structure-based semantic lens for large element-based plots. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2600–2609, 2011.
- [21] C. Hurter, B. Tissoires, and S. Conversy. FromDaDy: Spreading data across views to support iterative exploration of aircraft trajectories. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1017–1024,

- 2009.
- [22] 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/Transactions on Visualization and Computer Graphics.2007.48
- [23] R. Kincaid. SignalLens: Focus+context applied to electronic time series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):900–907, 2010. doi: 10.1109/Transactions on Visualization and Computer Graphics.2010.193
- [24] 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, 2006.
- [25] A. Lhuillier, C. Hurter, and A. Telea. FFTEB: Edge bundling of huge graphs by the Fast Fourier Transform. In *Proc. IEEE PacificVis*, 2017.
- [26] W. Li and G. Paladini. Luggage visualization and virtual unpacking, 2013. WO patent WO2013142072A3, Siemens Corp.
- [27] W. Li, G. Paladini, L. Grady, T. Kohlberger, V. K. Singh, and C. Bahlmann. Luggage visualization and virtual unpacking. In *Proc. WASA SIGGRAPH Asia Workshop*, pp. 161–168. ACM, 2012. doi: 10.1145/2425296.2425325
- [28] M. J. McGuffin, L. Tancau, and R. Balakrishnan. Using deformations for browsing volumetric data. In *Proc. IEEE Visualization*, pp. 401–408, 2003. doi: 10.1109/VISUAL.2003.1250400
- [29] W. Merzkirch. *Flow visualization*. Elsevier, 2012.
- [30] NVIDIA, Inc. CUDA SDK code samples, 2018. [http://www.nvidia.com/object/cuda\\_get\\_samples\\_3.html](http://www.nvidia.com/object/cuda_get_samples_3.html).
- [31] C. Pindat, E. Pietriga, O. Chapuis, and C. Puech. JellyLens: Content-aware adaptive lenses. In *Proc. ACM Symp. on User Interface Software and Technology (UIST)*, pp. 261–270, 2012. doi: 10.1145/2380116.2380150
- [32] B. Preim, A. Baer, D. Cunningham, T. Isenberg, and T. Ropinski. A survey of perceptually motivated 3d visualization of medical image data. *Computer Graphics Forum*, 35(3):501–525, 2016. doi: 10.1111/cgf.12927
- [33] R. K. R., S. D. Pieper, and K. Vosburgh. 3D Slicer: A platform for subject-specific image analysis, visualization, and clinical support. *Intraoperative Imaging Image-Guided Therapy*, 3(19):277–289, 2014.
- [34] 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
- [35] B. W. Silverman. Density estimation for statistics and data analysis, 1986.
- [36] A. Telea and J. J. van Wijk. Simplified representation of vector fields. In *Proc. IEEE Visualization*, pp. 263–270, 1999.
- [37] C. Thiede, G. Fuchs, and H. Schumann. Smart lenses. In *Proc. Intl. Symp. on Smart Graphics*, pp. 178–189. Springer, 2008.
- [38] C. Tominski. Event-based concepts for user-driven visualization. *Information Visualization*, 10(1):65–81, 2011. doi: 10.1057/ivs.2009.32
- [39] C. Tominski, J. Abello, F. van Ham, and H. Schumann. Fisheye tree views and lenses for graph visualization. In *Proc. Intl. Conf. on Information Visualisation (IV)*, pp. 17–24, 2006. doi: 10.1109/IV.2006.54
- [40] C. Tominski, S. Gladisch, U. Kister, R. Dachsel, and H. Schumann. Interactive lenses for visualization: An extended survey. *Computer Graphics Forum*, 36(3):173–200, 2017. doi: 10.1111/cgf.12871
- [41] C. Tominski, H. Schumann, G. Andrienko, and N. Andrienko. Stacking-based visualization of trajectory attribute data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2565–2574, 2012. doi: 10.1109/Transactions on Visualization and Computer Graphics.2012.265
- [42] 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/Transactions on Visualization and Computer Graphics.2015.2502583
- [43] X. Tong, C. Li, and H. W. Shen. Glyplens: View-dependent occlusion management in the interactive glyph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):891–900, 2017. doi: 10.1109/Transactions on Visualization and Computer Graphics.2016.2599049
- [44] 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
- [45] M. van der Zwan, V. Codreanu, and A. Telea. CUBu: Universal real-time bundling for large graphs. *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2550–2563, 2016.
- [46] L. Wang, Y. Zhao, K. Mueller, and A. Kaufman. The magic volume lens: an interactive focus+context technique for volume rendering. In *Proc. IEEE Visualization*, pp. 367–374, 2005. doi: 10.1109/VISUAL.2005.1532818
- [47] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proc. Working Conf. on Advanced Visual Interfaces (AVI)*, pp. 110–119. ACM, 2000. doi: 10.1145/345513.345271
- [48] Y. Wu and H. Qu. Interactive transfer function design based on editing direct volume rendered images. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1027–1040, 2007. doi: 10.1109/TVCG.2007.1051