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 int 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 [**?**, **?**, **?**]. 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!**. Most existing occlusion management techniques does not simultaneously meet the following requirements:

- Interactivity (R1): Give a fast unobstructed of the target,
- Flexibility (R2): The parameters of the tool can be modified by the user
- Focus+Context (R3): keep the global context,
- Robustness to densities similarity (R4): work for data-sets where the target and occluder are very similar density-wise.

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 [**?**]. Multiple viewports, a view paradigm using two or more views, can be used to see the data from different perspectives [**?**]. However, 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. Kruger et al. [**?**] 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

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

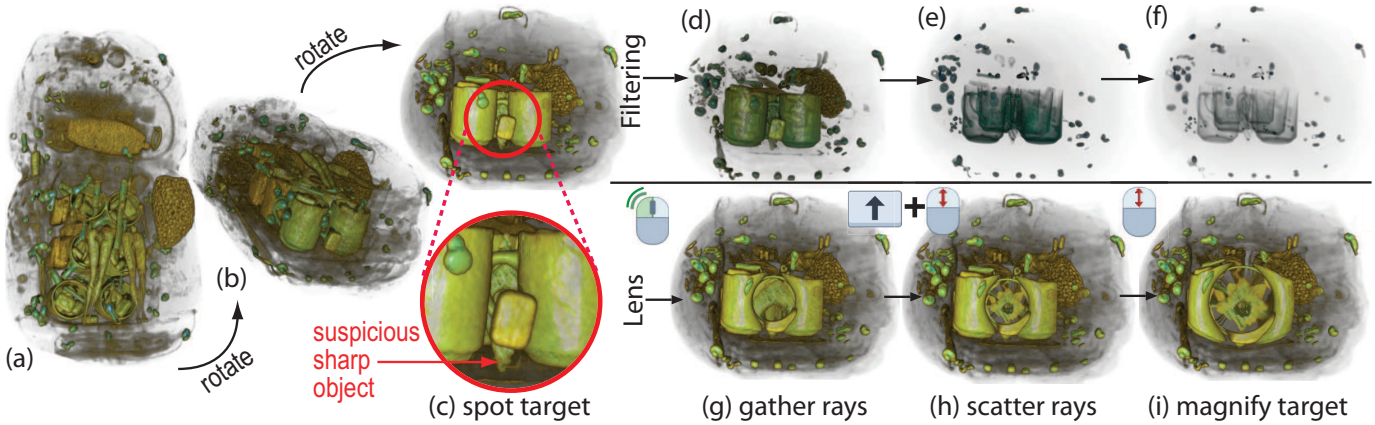

Fig. 1. (a-c) A baggage scan is viewed from different angles. In view (c), a suspicious sharp object is spotted between a set of mugs. (d-f) Filtering densities using a classical 1D opacity transfer function removes progressively more of the occluders (mugs), but also the target. (g) The user applies 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).

Ma [?] 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 (R4) [?]. 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 [?] 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 [?, ?]. 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. [?] 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 (R3), 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 [?]. However, 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 a localized visualization problems by temporarily altering a selected part of the visual representation of the data [?]. In other words, a lens is a parameterizable selection according to which a base visualization is altered (R2). 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 [?] or rectangular [?]. Our lens has also a circular shape in order to remind its magnifying property. Some lenses, such as the JellyLens [?] and the smart lenses [?], 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 [?] or guide the exploration path along interesting events [?]. 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 [?], which renders the occlusions with higher transparency and magnifies volumetric pre-computed features interactively or automatically in a pre-segmented dataset. However, this study fails to provide an interactive tool able to deal with similarities in density between the occluder(s) and the area of interest (R1,R4). 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 [?] 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 [?]. However, this technique requires significant computational effort to render a single image from features of interest at different scales. Bruckner and Groller [?] proposed exploded views for volume data by partitioning the volume into several segments. Correa et al. proposed a framework [?] allowing users to physically manipulate the geometry of a data object. McGuffin et al. [?] 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 [?]. Other techniques performed deformations using surgical metaphors [?, ?] 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. [?], 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 ray 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

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

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 α 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 visualizaton of the volume.

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

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

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 lens. Intuitively, this works as if we moved a maginfying 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 β 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 α and β 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 many viewpoints, thereby allowing the user to look 'around' it so as to see its parts which are not visible from the current viewpoint, but *without* having to actually change the viewpoint. This is of high added value, since changing the viewpoint may bring us to a view in which the target is completely invisible, so we do not know where precisely to activate the lens any more. Figure 4 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 ϕ 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 become less specular, and voxels out of focus appear 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. Secondly, 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, surrounded by a constantly-lit diffuse scene. This highlights small-scale details on the target surface, again, without having to change the viewpoint or lens location. Additionally, the high specularity in the lens attracts the user's attention to this focus area; the diffuse lighting outside the lens put less emphasis on this context area.

Opacity: Finally, we modify the opacity transfer function along a similar idea as for lighting. Let $TF_o^{global} : \mathbf{R} \rightarrow [0,1]$ be the user-chosen opacity function used globally for the volume. Let Γ be a Gaussian pulse of unit height centered at the average density value $\bar{\rho}$ in $B(\mathbf{c},R)$ and with standard deviation σ. We estimate $\bar{\rho}$ and σ by considering the density ρ at xxx 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}, as in standard DVR. The effect is that voxels in B become ore opaque, thus more visible. Tet, voxels having the same densities but outside B will use the default transfer function, which can make them transparent. This allows to have voxels with similar densities either opaque (if close to the target, thus of interest) or transparent (if they are *e.g.* in front of

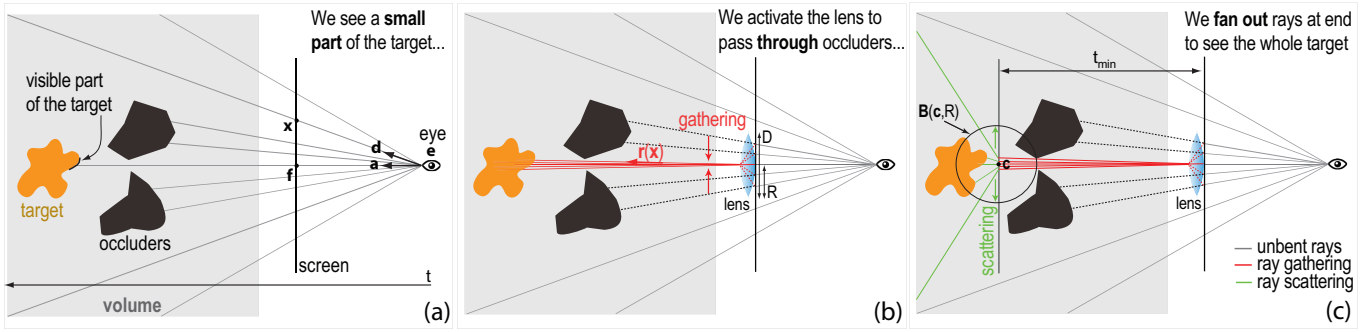

Fig. 2. Principle of the obstruction-free lens. An interesting target object is mostly hidden by occluders in front of it. (a) Classic raycasting result shows a small part of the target. (b) Our lens gathers the rays to avoid occluders. Once close to the target, rays follow again their initial paths.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

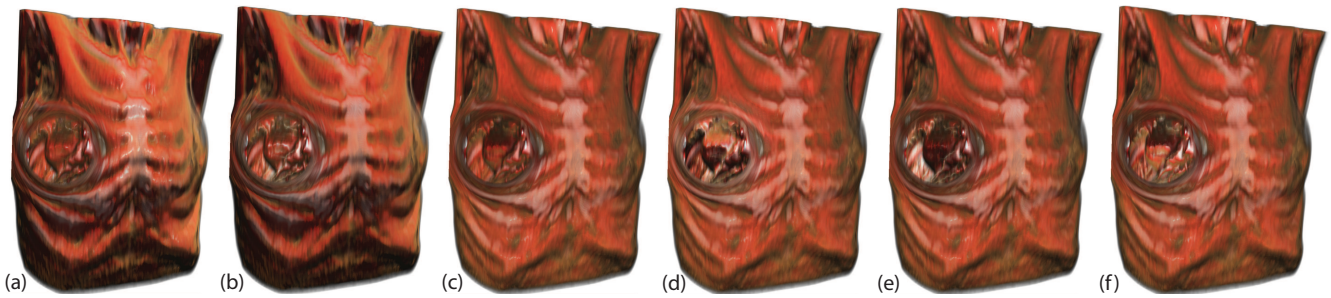

Fig. 3. Changing lighting parameters in the lens. (a) Constant specular coefficient. (b) High specular coefficient in the lens. (c-f) Changing the in-lens light vector yields the effect of a flashlight rotating around the target.

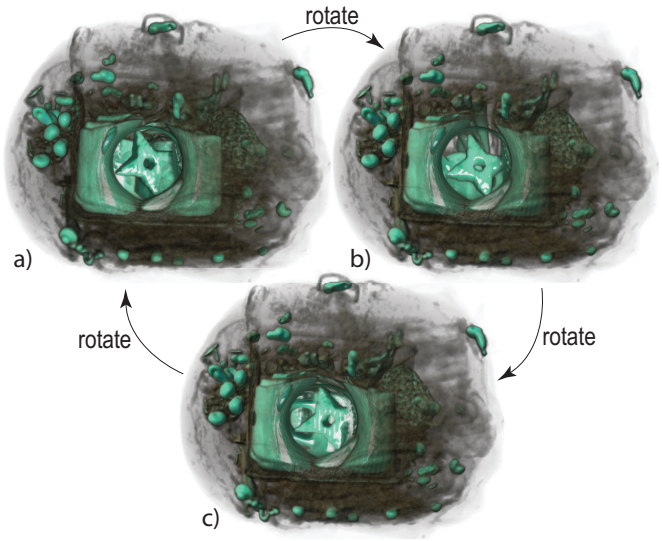

Fig. 4. Performing local rotations in the lens allows better seeing the shape and thickness of the target object.

the target, thus occluding).

3.4 Smooth transitions

If we apply Eqns. 1 and 2 to bend rays passing through the lens pixels D, and trace all other 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 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 computed along an straight-line 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 compute the interpolated ray $\bar{\mathbf{p}}(\mathbf{x},t) = (1 - f(d))\mathbf{p}(\mathbf{x},t) + f(d)\mathbf{p}^{line}(\mathbf{x},t)$, 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 $\bar{\mathbf{p}}(\mathbf{x},t)$ to compute the DVR by standard composition. This way, rays effectively vary smoothly from their bent versions (close to the lens 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.

Separately, we use a slow-in/slow-out animation [**?**] to introduce the lens effect. When the lens is activated, we vary the values of α and β 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 overall 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 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

We implemented our occlusion-free lens by modifying a standard DVR ray caster implemented in turn using NVIDIA's CUDA platform. Such

a ray caster is publicly available in CUDA's SDK [**?**]. The key changes we applied the definition of the ray (Eqns. 1 and 2), and controlling the lens and local per-voxel Phong lighting parameters via mouse and keyboard events. On a PC with xxx GB RAM and a GeForce xxx card, we achieve an interactiv framerate (xxx fps) for volumes up to xxx^3 voxels. Currently, we use a compositing ray function. However, 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 problems.

5 Application Scenarios

Our obstruction-free fish-eye lens can be used with different types of volumetric datasets. To illustrate this, we next present four such use-cases considering scalar density volumes coming from baggage inspection, 3D flow simulation, radiology, and air traffic management.

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 densities of harmful substances such as C-4, TNT, and nitroglycerin, and event some prohibited articles such as classical firearms and knives, it remains difficult to identify unusual threats. In addition, baggage inspection faces four main concealment strategies [**?**]:

Superposition: A threat (prohibited object) may be sheltered among dense materials. It is sometimes possible to see through such a 'shield' using high penetration (enhanced X-ray power) or image processing (contrast improvement) techniques. However, such techniques are not universally available and also require fine-tuning various parameters, which slows down the inspection process.

Location: Depending on its location inside the luggage, a threat can be hard to detect. Objects located in the corners, edges, or in the luggages 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 lure can be used to to hide the real threat. For instance, a minor threat like a small scissors c be clearly visible and catch the security agent's attention while a more important threat remains hidden.

Consider the baggage scan in Fig. 1 with a volume size of 283x189x344 voxels. Automatic baggage inspection systems will not detect anything suspect on this scan. However, while visually exploring this baggage from different angles (Fig. 1a-c), 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 materials by density in order to show or hide subsets of the volume and reduce the occlusion. However, in this case, this solution does not work, as the suspect target has almost the same density as the surrounding mugs. Hence, removing the occludes will also remove the target (Fig. 1d-f). Using the obstruction-free fish-eye lens helps in this kind of situation. 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 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). However, this shows only a small part of the target. Scattering rays next fully reveals the target (Fig. 1h). The user can adjust the lens size to get a more detailed view of the target (Fig. 1i). Next, the user can locally turn the viewpoint around the target, as already shown in Fig. 4. From these views, the controller decides that the target is a copy of a shuriken (Japanese ninja star weapon). However, since the object is very thick and blunt (see Fig. 4), it is clearly not a threat.

5.2 Fluid flow: A deep-buried spherical vortex

Flow visualization using streamlines has a long history in scientific visualization [**?**, **?**]. 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 [**?**], captures the simulation of water flow in a basin computed on a grid of 128x85x42 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) [**?**]. Similar techniques have been used to compute density maps of 2D trail-sets [**?**, **?**, **?**]. To increase computational speed, we compute the KDE in the frequency space and using GPU acceleration, following [**?**]. The resulting volumes have a resolution of 500^3 voxels and can be directly displayed using DVR (Fig. 5). 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. 5a). 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. 5a), 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. 5b). 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 [**?**, **?**, **?**, **?**, **?**]. 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. 5c). Finally, we can close the lens but keep the target magnified (Fig. 5d).

5.3 Chest scan: A hard to see tumor

In our third use-case, we consider a contrast chest CT scan (512x512x110 voxels) of an elderly patient having a sizeable (roughly 8 cm diameter) lung tumor. Typical examination of such data by the pulmonologist and radiologist in charge involves using slice-based views. In some of these views, the tumor is clearly visible (Fig. 6a,c), though not in all of them (Fig. 6b). Moreover, the exact tumor shape, morphology, and connection to the lung walls is not easy to assess. Using standard DVR makes the tumor partially visible (Fig. 6d). However, occlusion from the rib cage and other tissues is still present. Using both TF presets and manually changing the TFs of the 3D Slicer tool [**?**] used to construct 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. 6a-c, where the grayvalues for the tumor and surrounding skin-and-muscle tissue on the rib cage are very similar. Hence, we cannot remove such occluding tissue by opacity TF manipulation without also removing the tumor.

We next use our lens for this task. Sample snapshots obtained in this process are shown in Fig. 3. Comparing these with standard DVR (Fig. 6d), 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. 6d 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 techniques they use in their practice. Both specialists have a 10-plus year 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 neutral proxy communicator in the interaction. The provided input

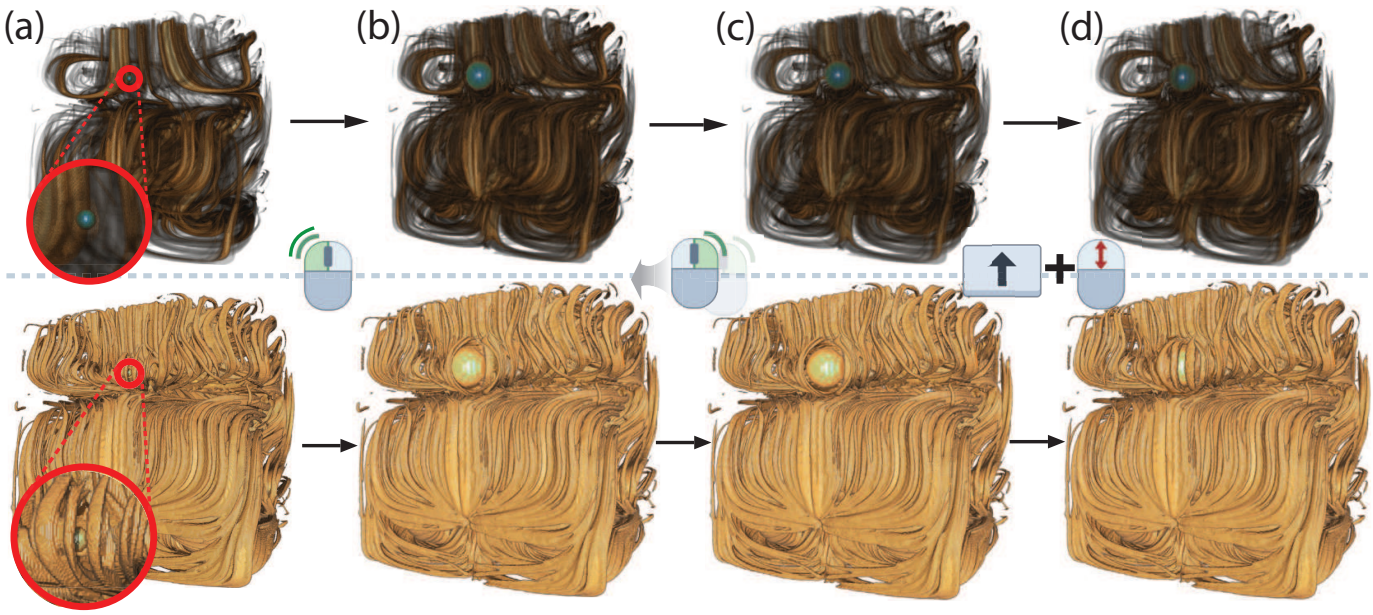

Fig. 5. Flow volume exploration using two different opacity transfer functions (top and bottom rows). From the viewpoint (a), a small high-density spherical item appears between the streamlines. (b) The lens is applied at that location (double click). (c) The directions of rays inside the lens are changed to see the whole spherical target in the lens (right click + mouse drag change direction). (d) The lens is gradually closed while keeping the focus area magnified (shift + scroll).

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. While it has several parameters, changing these by direct interaction is considerably easier to do than tuning the typical parameters of DVR to obtain similar results. This 'tempts' the user to exploration, 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 very strong feature, as viewpoint changes are disruptive and cost time. This is especially important in a cost-aware environment where specialists have very limited time (10 minutes) to fully assess a patient examination. However, the lens cannot and should not replace classical slice-based investigation, which shows small-scale details better. This is especially important for assessing 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.

5.4 Aircraft trajectories: Outliers in the French sky

Our final use-case considers a task from the air traffic planning field – detecting and studying outliners in large-scale datasets containing tens of thousands of 3D (latitude, longitude, height) trails of aircraft over a given spatio-temporal region [**?**]. Typically, such datasets are displayed using 2D (latitude, longitude) plots where opacity encodes the spatial density of flights. Fig. 7-a shows one day of recorded aircraft trajectories over the French air space using this 2D technique. Fig. 7(b) shows a detail zoom-in of this dataset, where we can see an abnormal – that is, not relativlyu straight – 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.* [**?**], 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 outliner (that is, not straight) trail at some point in space, see curved patterns in Fig. 8a. 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. 8a). 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. 8b-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 [**?**] costs several minutes. Doing the same using our lens approach costs under one minute, for the same users. Additionally, if we compare Figs. 7b and 8b-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.

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.

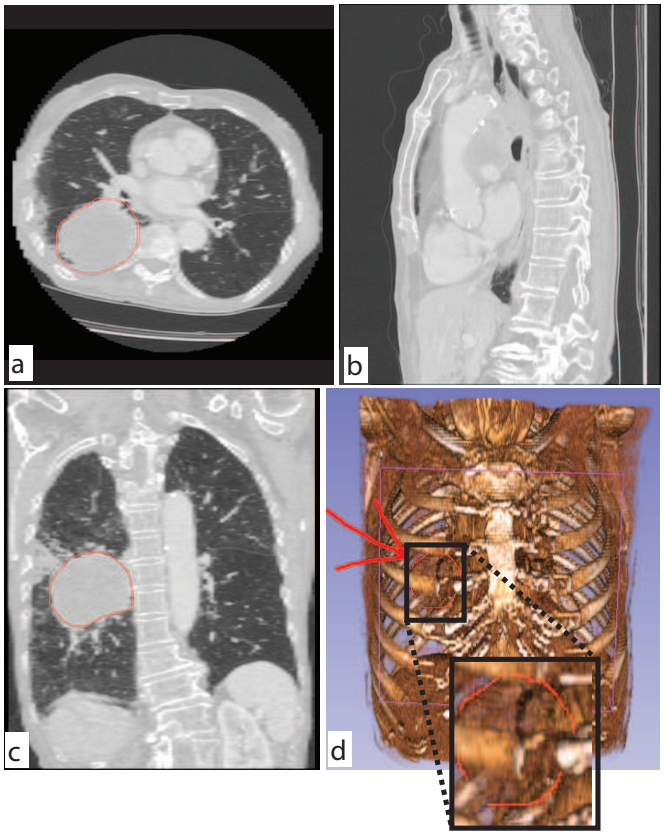

Fig. 6. Lung tumor visualization using slices (a-c) and standard DVR (d). Annotations are manually added by the examiner to delineate the tumor location. Images constructed using the 3D Slicer tool [**?**].

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

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

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

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

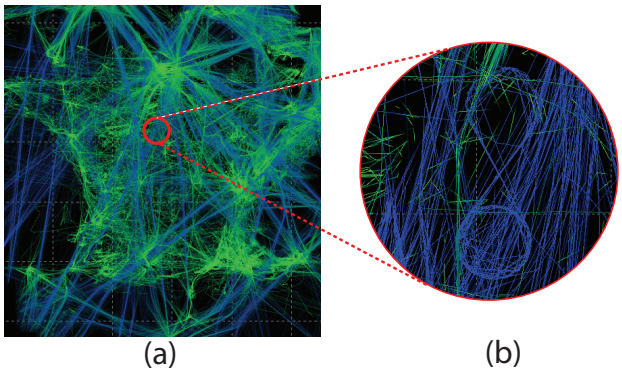

Fig. 7. 2D visualization of one day of recorded aircraft trajectories over France [**?**]. (a) Overview of all trails. (b) Zoom, filtering, and color mapping techniques are used to highlight an outlier trajectory of an aircraft performing an eight-shaped loop. Revealing this outlier costs significant user effort.

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 only constraint is that the user can find a viewpoint from which the target of interest, deep buried in the data, is at least slightly visible. We next presented several modifications of the local rendering parameters within the lens (view direction, lighting parameters, opacity transfer function) that both stronger 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 any 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. We demonstrate these points using four use-cases involving datasets from baggage detection, fluid visualization, air traffic control, and medical imaging.

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, is a goal we aim to pursue next.

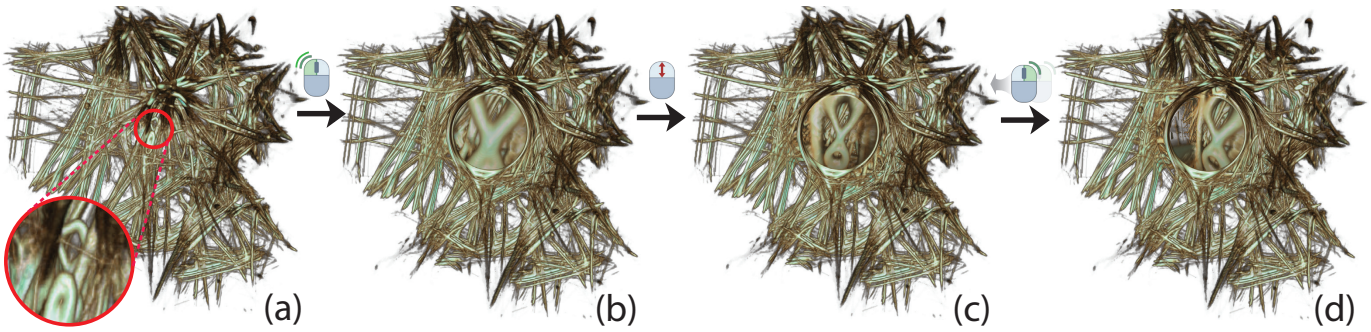

Fig. 8. Inspecting an abnormal aircraft trajectory. (a) The initial view on the trajectories where the abnormal trajectory is spotted, as it is highly curved while all other trajectories are relatively straight. Activating the lens at the location of the spotted outlier (b) and changing the magnification factor (c) reveals the trajectory's eight-shape. (d) Rotating the viewpoint provides more spatial insight on the embedding of the outlier in the surrounding trajectories.