



# UNIVERSITY OF CALGARY

University of Calgary

**PRISM: University of Calgary's Digital Repository**

---

Science

Science Research & Publications

---

2018-10

## 3De Interactive Lenses for Visualization in Virtual Environments

Mota, Roberta Cabral Ramos; Rocha, Allan; Silva, Julio Daniel; Alim, Usman; Sharlin, Ehud

---

Mota, R.C.R., Rocha, A., Silva, J.D., Alim, U., & Sharlin, E. (2018, October). 3De Interactive Lenses for Visualization in Virtual Environments. Presentation at IEEE SciVis (SciVis Short Papers), Berlin, Germany.

<http://hdl.handle.net/1880/110554>

conference presentation

---

Unless otherwise indicated, this material is protected by copyright and has been made available with authorization from the copyright owner. You may use this material in any way that is permitted by the Copyright Act or through licensing that has been assigned to the document. For uses that are not allowable under copyright legislation or licensing, you are required to seek permission.

*Downloaded from PRISM: <https://prism.ucalgary.ca>*

# 3De Interactive Lenses for Visualization in Virtual Environments

Roberta C. R. Mota \*

Allan Rocha

Julio Daniel Silva

Usman Alim

Ehud Sharlin

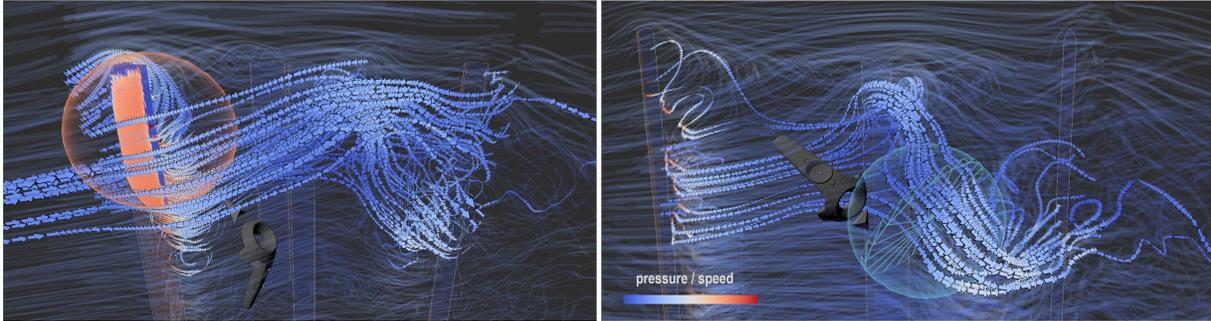


Figure 1: *3De lens* inspecting wind turbine aerodynamics. (left) The lens depicts strong-pressure gradient on the leftmost wind turbine blade; it also selects the air flowing around this blade and advancing to the rightmost one. (right) By setting its orientation, the lens selects a disorderly bundle of flow lines passing through the central blade.

## ABSTRACT

We present *3De lens*, a technique for focus+context visualization of multi-geometry data. It fuses two categories of lenses (3D and Decal) to become a versatile lens for seamlessly working on multiple geometric representations that commonly coexist in 3D visualizations. In addition, we incorporate our lens into virtual reality as it enables a natural style of direct spatial manipulation for exploratory 3D data analysis. To demonstrate its potential use, we discuss two domain examples in which our lens technique creates customized visualizations of both surfaces and streamlines.

**Index Terms:** Human-centered computing—Human computer interaction—Interaction paradigms—Virtual Reality; Human-centered computing—Visualization—Visualization Application Domains—Scientific Visualization

## 1 INTRODUCTION

In 3D visualizations, data is often represented as a combination of geometric representations (e.g., points, streamlines, surfaces), with associated attributes. To present data across multiple representations brings on the need for investigating the (inter-relations of) different data types to obtain an integrated understanding of the phenomenon. Yet, the complexity associated with multi-geometry data inevitably causes significant visual clutter and occlusion; this impedes users to properly analyze the data space.

To address the challenges of clutter and occlusion, the visualization community has embraced the idea of interactive lenses: focus+context tools that define focus areas of the visual data representation; focus is shown in details while remaining regions convey context. More precisely, a lens is an *interactively parametrizable spatial selection* according to which a base visualization is altered [28]. Key geometric parameters of lenses include shape and transformations (position, orientation, and scale). Shape is the most prominent property and commonly drives the spatial selection; position and scale attune the lens to different parts of the data; orientation is often an overlooked parameter but can be useful to align the lens in accordance with the underlying visualization, or even to fine-tune the

spatial selection based on a directional parameter. We here classify existing lenses for 3D data according to their geometric parameters as 2D, 2.5D, 3D, and Decal.

Most techniques are so-called 2D lenses: polygonal shapes (e.g., circular) placed and manipulated in screen space. Their 2D nature constrains the lens transformations to the view plane, and leads to two major drawbacks: an inability to carry out 3D selection, and a lack of spatial correlation between the 2D position and orientation of the lens and the underlying 3D visualization. 2.5D lenses consist of a 2D lens embedded in 3D space. 2.5D lens transformations are extended to 3D; this alleviates the issue of spatial correlation. It introduces, however, increased interaction effort to place and align the lens in the 3D data set. In addition, the flat nature of the lens constrains the spatial selection to only a slice of the 3D data set.

An extension of the 2.5D lens is 3D lens, which consists of a 3D volume (e.g., sphere) onto which lens operators are applied. 3D lenses are usually used for spatial selection of volumetric data; for surface data, however, such lenses do not closely align with the underlying geometry due to their rigid shape. This can lead to perceptual issues such as the lens itself occluding parts of the surface (especially those of intricate geometry). Moreover, the manipulation of a 3D lens suffers from interaction effort like a 2.5D lens.

Another lens category, called Decal, supports selection of surface areas of interest. A recently proposed decal-lens [20] is built from the intersection of a sphere with a surface. The sphere itself does not provide the area where the lens effect takes place; rather, the lens refers to the part of the surface that lies inside the sphere. This leads to a lens-region that follows the surface geometry, like a decal. This lens allows for 3D manipulation due to its intrinsic volumetric shape; but the lens effect manifests solely on surfaces.

Given the abovementioned, we note two key remarks: 3D and decal-lenses may share common parameters (e.g., a spherical shape), and are somewhat complementary in terms of data types they best operate on (non-surface and surface, respectively). Beyond that, to date, lens techniques are commonly designed for a specific data type of the underlying visualization [28]. In this regard, multi-geometry 3D visualizations can use separate lenses for different data types; yet, this may increase user's effort to manipulate each lens individually and lead lenses in close proximity to occlude each other.

This work thus contributes *3De lens*, a technique that combines 3D and Decal lenses into one coherent entity to act on multi-geometry data. Lens manipulations and parameter adjustments become centralized, mutual occlusion no longer holds, and the lens seamlessly works on distinct geometric representations that com-

\*The authors are with the University of Calgary, Canada. E-mails: {roberta.cabralmota, acarocha, jd.silva, ualim, ehud}@ucalgary.ca

monly coexist in 3D visualizations. We illustrate the applicability of our lens through two domain examples to specifically support the visual exploration of surfaces and streamlines.

Moreover, in terms of interaction modality, most lens techniques are designed for classic desktop workstations with mouse and keyboard. In 3D visualizations, particularly, these systems face the fundamentally difficult problem of manipulating the lens in 3D within a 2D interaction space. For instance, one may need to place lenses to cut and reveal data values in the cavities of the aorta region of the heart; to seed particles and interrogate intricate blood flow patterns on an anatomical model of the left ventricle; etc. Nonetheless, to adjust the lens in relation to the underlying data using standard keyboard and mouse becomes demanding—in this regard, a recent survey by Tominski *et al.* [28] emphasizes the need for adopting lenses in novel visualization environments while citing first promising research. This work incorporates the 3De lens technique in consumer-grade virtual reality (VR) technology to enable direct lens interaction using an associative 3D interaction space; this may lead to more natural input mapping with less physical and mental strain.

Our main contributions are:

- The concept of *3De lens* that fuses 3D and Decal lenses in order to seamlessly operate on multi-geometry 3D visualizations.
- The visual and interaction design and implementation of our lens technique using VR technology. We discuss both the design process and technical challenges to demonstrate that, compared to widely-used desktop systems, VR presents a promising medium to ease interaction effort during lens-based 3D data exploration.

## 2 RELATED WORK

In this section, we review work that proposes lenses for 3D data according to the geometry and the interaction modality used as medium for interactive lens techniques. For a more extensive review of lenses, we refer the reader to the survey by Tominski *et al.* [28].

**Lens geometry.** 2D screen-space lenses have been useful for 3D data exploration [29,34]. For example, as the correlation between hemodynamic attributes is necessary to aneurysm analysis, Gasteiger *et al.* [12] proposed *FlowLens* for focus+context visualizations of pairs of attributes like wall shear stress and inflow jet. 2.5D lenses have been mostly used to navigate and to perform local (focus) operations on slices of volume data sets [8, 18]. Van Pelt *et al.* [31] proposed cross-section techniques to explore 4D PC-MRI blood-flow data. After the selection of probe cross-sections on vessel models, physicians can employ techniques to locally capture blood-flow dynamics such as exploded planar reformats, streamline or pathline seeding templates, and flow-rate arrow-trails. 3D lenses were first introduced by Viega *et al.* [32] to limit the lens effect to a finite subvolume of interest; this lens concept received a number of extensions for different data types. Fuhrman and Gröller [11], for instance, are among the first to propose 3D lenses for 3D flow visualization. They use a volumetric box to implicitly or explicitly define a spatial selection in the flow to depict higher-resolution streamlines. Rocha *et al.* [20] recently proposed decal-lenses to facilitate on-demand multivariate data exploration on surfaces; to the best of our knowledge, it is the only work fitting into Decal lens category. *3De lens* does not fit into the previous individual classification; rather, it merges lenses of distinct categories into a single coherent entity.

**Lens interaction.** Existing lens techniques are usually designed for classic desktop visualization settings with mouse and keyboard interaction. It is used in many of the previously examined research works, for instance in [12, 31, 34]. There has also been work towards interactive lenses for touch-enabled devices such as tabletops and high-resolution large displays [8, 23, 29]. Tangible objects were also coined as physical lenses, typically used jointly with tabletops to facilitate tracking strategies and use the space surrounding the

tangibles as context visualization [24,25]. One limitation that both touch- and mouse-based interaction face is the problem of 3D data manipulation within a 2D interaction space. Tangible interaction benefits from physicality and the use of human natural motor skills to ease coordination within the interaction space [9]. One limitation of tangible lenses, though, are their rather fixed size and shape. In terms of lens interaction, these limitations introduce challenges especially when dealing with convoluted volumetric data with cavities, holes, or folds. For instance, it may be difficult to precisely sit a lens into surface cavities or to orient it in areas of high curvature.

To afford more natural interactions, there is a growing interest in incorporating lenses in modern augmented [7, 16] and virtual reality [17, 30] systems. Niebling [18] recently proposed an interaction metaphor using 3D-tracked tablets, which closely resembles cutting planes, for continuous slicing of computational fluid dynamics (CFD) data sets. Advantages of these modern technologies for 3D data exploration include optical tracking (e.g., head and hands) for direct manipulation in 3D; stereoscopy and motion parallax; and proprioception/kinesthesia for body-relative spatial interactions, often in a physical 1:1 scale. Our work leverages the emerging generation of VR in order to minimize interaction effort for lens-based exploration, a major challenge faced by 2D interaction systems.

## 3 LENS CONCEPT

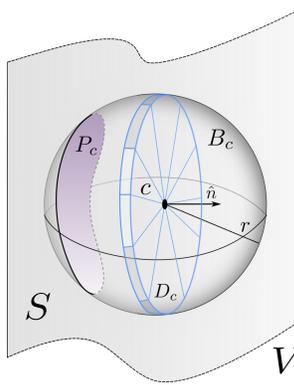


Figure 2: Lens illustration.

Based on the discussion on using different lens categories for 3D data, we propose *3De lens* as a combination of 3D and Decal lenses into a single one. This integration scheme enables the lens to act on different types of data, each of which benefiting from particular traits of its associated lens. We refer the reader to Fig. 2 to illustrate the concepts we now describe.

- Let us consider a three-dimensional Euclidean space  $\mathbb{R}^3$  containing a volume  $V$  and a surface  $S$  onto which the 3De lens  $L$  will be placed. We denote the lens region as  $B_c$ , a ball centered at an arbitrary point  $c$  with radius  $r$ . The lens  $L = \langle L_{3D}, L_{De} \rangle$  is a set of containment lenses:
- $L_{3D}$ .  $B_c$  represents the region that will contain the 3D lens; which allows for selecting a subvolume of  $V$  while displaying a non-surface focus attribute  $a_1$ . We also denote a disk within  $B_c$  as  $D_c$ , with normal vector  $\hat{n}$ , center  $c$ , and radius  $r$  so that the radial extension of the disk is restricted to the ball shape.  $D_c$  defines the orientation of the 3D lens; which enables an optional angular selection over  $V$  based on the disk's normal direction.
  - $L_{De}$ . The intersection of  $S$  and  $B_c$  defines a patch  $P_c = B_c \cap S$ .  $P_c$  can be understood as the area that will contain the decal lens: a 2D submanifold of  $S$ . This lens supports the task of selecting a region over  $S$  while encoding a surface focus attribute  $a_2$ .

## 4 LENS TECHNIQUE

In this section, we define design goals (DGs) based on the identified limitations of using different interaction modalities and lens techniques for 3D data. Afterwards, we build upon our guidelines to describe the visual and interaction design of our lens technique. To aid our discussion, we use a public aneurysm data set [3] that contains pressure and mean curvature of the aneurysm vessel as surface attributes; and we trace streamlines to capture the flow profile inside the vessel, with velocity and vorticity as associated attributes.

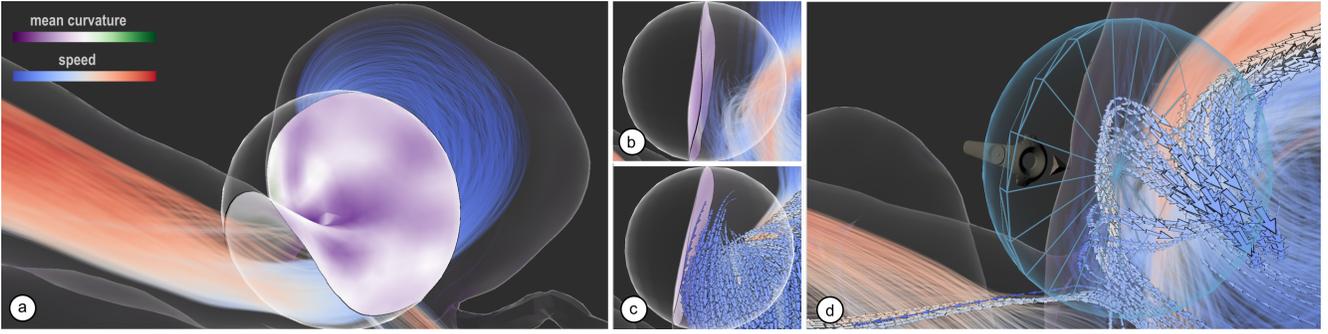


Figure 3:  $3D$  lens in the aneurysm scope. (a) Decal displaying mean curvature over an aneurysm vessel; the lens patch provides depth cues for perception of surface shape and spatial relations between vessel and underlying blood flow. (b)  $L_{De}$  depicts mean curvature on surface focus area. (c)  $L_{3D}$  selects near-wall flow described by the magnitude of velocity. (d)  $L_{3D}$  further identifies clusters of flow lines following a common direction and focus on a single flow pattern.

The aneurysm scenario contains two geometric representations: an isosurface and streamlines. It is important to highlight the general nature of our lens technique as it can be applied to other representations (e.g. point clouds and volumes) using the same paradigm.

- **DG<sub>Manip.</sub>** *Facilitate manipulation.* For 3D exploration, to support lens manipulation in a way that minimizes user effort is desirable. Our intent is to use direct 3D manipulation so that interaction operations are seamlessly embodied in the 3D visualization.
- **DG<sub>Fluid.</sub>** *Provide for fluid interaction.* Interaction is the catalyst for the interplay between data and user, and is an essential factor for lens-based data exploration. We strive for fluid interaction by *minimizing indirection in the interface* and *providing immediate visual feedback on interaction*, as proposed by Elmqvist *et al.* [10].
- **DG<sub>Occl.</sub>** *Minimize occlusion as necessary.* Clutter and occlusion are major challenges in 3D visualization. We thus focus on providing visualizations with minimal clutter in a way that does not compromise depth cues required to discriminate 3D saliency and relationships.

#### 4.1 Interaction Design

The potential for direct spatial manipulation led us to incorporate our lens technique in VR, using Unity [4] and an HTC Vive [2]. Its room-scale tracking allows users to walk around or inside the data, and seamlessly interact with it using tracked headset and controllers.

From an interface perspective, our lens technique captures input/output events from different lenses, keeps track of which lens produced which event, and places all events on a single visual entity. This prevents manipulating multiple independent widgets that compete for screen space, and decreases context switching as user’s attention can remain focused on the work area (**DG<sub>Fluid.</sub>**).

We now return our attention to the focus+context visualization of the aneurysm scenario in order to describe our lens interaction vocabulary in VR. First, the user can press and hold the trigger button on the controller to *grab* the lens  $L$  so that its center  $c$  mirrors the controller’s spatial position; and this translates to the constituent lenses since  $L$  employs an enclosing structure. The grabbing metaphor supports direct lens manipulation in 3D (**DG<sub>Manip.</sub>**), and is known to be very easily understood due to its natural “kinaesthetic correspondence”: an isomorphic correspondence between hand movement and the visually perceived motion [19, 35]. While grabbing  $L$ , the user can seek for surface areas of interest: whenever the lens intersects a surface geometry,  $L_{De}$  depicts the surface focus attribute. Fig. 3-a illustrates how a decal-lens adapts to the surface geometry even in areas of high curvature; and displays the mean curvature using a diverging purple-to-green colormap.

The user releases the button to *ungrab*  $L$ . This action triggers events via  $L_{3D}$ : the selection of all streamlines passing through its lens region and the display of non-surface focus attribute. Fig. 3

shows flow velocity magnitude using a diverging cool-to-warm colormap; we use animated arrow glyphs to communicate the direction of the selected flow near the vessel wall.

Whenever the user places the controller inside  $L$ , a disk-like widget appears that orients  $L_{3D}$ . By pressing and holding the trigger button, the user *grabs* the disk so that its normal direction mirrors the controller’s upward orientation (**DG<sub>Manip.</sub>**). This in turn produces an angular-based selection carried out via  $L_{3D}$ : only streamlines that are approximately aligned with the normal of the disk will be selected (see Fig. 3-d). The angular tolerance allowed is 15 degrees.

Lastly, one can *scale* the lens  $L$  in order to alter its radius  $r$ ; this transformation transfers to all enclosed lenses. This is also useful for keeping costs at a level that facilitates interactivity and comprehensibility at all times. This concerns computational costs (**DG<sub>Fluid.</sub>**) but also cognitive costs, i.e., the effort required to make sense of the lens effect (**DG<sub>Occl.</sub>**). In conjunction with the disk-like widget, a small ball shape appears at the lens’ center  $c$  whenever the user places the controller inside  $L$ . By grabbing the ball widget and moving their arms to right/left, the user scales up/down  $L$ .

#### 4.2 Visual Design and Implementation

The HTC Vive uses a display resolution of 2160x1200 (1080x1200 per eye) at a 90 Hz refresh rate. The combination of high-resolution and frame-rate requirements for VR presents a significant challenge compared to traditional desktop visualization, where low frame rates and intermittent pauses for computations or data loading are more tolerable. We, therefore, rely on GPU features such as rasterization functionality, fragment and stencil tests, and blending operations (**DG<sub>Fluid.</sub>**). We obtain interactive frame rates (45 FPS) using a desktop with an Intel®Core™i3 processor with a GeForce GTX Titan 6G GPU. Our GPU-based implementation is explained via the aneurysm data; this contains 15K surface triangles and 2K blood flow streamlines. We refer the reader to Fig. 4 to illustrate the discussion that follows.

**Context.** In our visualization, context consists of one or more semi-transparent regions to lessen occlusion while revealing inner, opaque focus regions (**DG<sub>Occl.</sub>**). Concomitantly, important characteristic parts of the context (e.g., contours) are visualized to accommodate better understanding of spatial relationships between structures in the data.

In our implementation, we first read all data (attribute and polygonal geometry) from a volume data set and semantically classify them either as *surface* (S) or *non-surface* (NS) data (Fig. 4-left). These steps happen in a pre-processing stage before the actual rendering process. In the aneurysm scenario, (S) is the vessel surface and (NS) refers to the blood flow streamlines.

We employ multi-pass techniques to render surface information (Fig. 4-middle). The surface outline is first drawn using a two-pass silhouette rendering algorithm similar to Rustagi [21]. The first

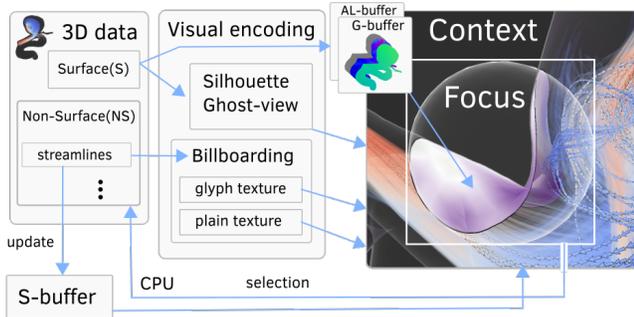


Figure 4: Visualization pipeline

pass enables the stencil buffer and renders the surface; this creates a binary mask of the surface and the background. In the second pass, we extrude the surface vertices along the normal direction (in camera space) in the geometry shader. The stencil buffer visibility test is used to mask out surface fragments, and the remaining fragments compose the surface silhouette. We then draw the surface using a Fresnel-reflection model that maps reflection to opacity values, similarly to Gasteiger *et al.* [13]. In this approach, the Fresnel opacity  $F_o$  assigned to the surface is given by  $F_o = 1 - |\hat{v} \cdot \hat{n}|^r$ , where  $\hat{n}$  is the surface normal,  $\hat{v}$  is the view vector, and  $r \geq 0$  is the edge fall-off parameter; we obtained good results using  $r$  values around 0.5. The combination of silhouette and Fresnel shading conceive an illustrative representation of the surface (Fig. 3, 4); these techniques together minimize occlusion by achieving more opacity in regions facing away from the viewer and more visibility in regions facing towards the viewer (**DG<sub>Occl.</sub>**).

In the aneurysm scenario, non-surface information consists of streamlines. Similarly to Schirski *et al.* [22], we employ a piecewise view-aligned billboarding technique to increase rendering speed and allow for interactive visualization of a large number of lines (**DG<sub>Fluid.</sub>**). The billboards are quadrilaterals computed on the GPU. For each point  $p$  along a streamline, two vertices are sent to the graphics system. The upper and lower vertices,  $p_0$  and  $p_1$ , related to the original position  $p$  are determined as  $p_i = p + r(-1)^i(\hat{v} \times \hat{d})$ , ( $i \in \{0, 1\}$ ); where  $r$  is the line thickness,  $\hat{v}$  is the vector to the viewer, and  $\hat{d}$  is the directional vector from the current to the next point on the streamline. Billboard textures are then set based on a selection discrimination: a semi-transparent plain texture for unselected data; and an opaque, animated arrow glyph to convey movement of particles through the flow domain for selected data (Fig. 4-right). A coloration is also applied to encode the non-surface focus attribute. In Fig. 3, the flow magnitude is represented using a cool-to-warm colormap.

Furthermore, the selection discriminator resides in a uniform buffer object (S-buffer); whose access identifiers are sent to the graphics system as vertex attribute. The number of elements in the buffer depends on the 3D data of input. In the aneurysm scenario, we employ a binary selection in which each streamline is either entirely selected or not; the buffer size thus corresponds to the number of lines in the blood flow field. Each point along a streamline stores the unique id of its seed, which is used as a per-vertex attribute on the GPU to access the S-buffer.

**Focus.** The lens is designed to convey shape perception while minimizing visual interference (**DG<sub>Occl.</sub>**). The lens is drawn in two steps: we first draw the lens surface using Fresnel shading (with  $r = 3$ ); afterwards, we generate a decal-lens on the surface (Fig. 4-right). The second step is explained in more detail below.

Similarly to Rocha *et al.* [20], we build two view-dependent buffers consisting of the *surface* data: a geometry buffer (G-buffer) and an attribute-layer buffer (AL-buffer). The G-buffer is a framebuffer object with three attachments: vertices, normals, and depth. The AL-buffer consists of a 2D texture array; the number of 2D tex-

tures is equal to the number of surface attribute layers. By applying a multi-pass process (in the draw call), we render each attribute and its visual representation to the AL-buffer. After this off-screen step, all surface attributes are available in screen space as 2D textures.

Afterwards, we can create a decal-lens using both the G-buffer and the AL-buffer. For each pixel within the projected lens' bounding box, we access the corresponding surface position  $p$  from the G-buffer and determine whether  $p$  lies inside the lens centered at  $c$  with radius  $r$ . The batch of all valid  $p$  inside the lens corresponds to the decal-lens region in screen space. We then map the visual representation of the surface attribute using the AL-buffer. In Fig. 3, the mean surface curvature is displayed using a large decal-lens.

Finally, we render the lens' widgets as wireframes following the approach proposed by Baeretzen *et al.* [6]. This technique renders both the model and its wireframe lines in a single pass inside the rasterization pipeline.

## 5 PRELIMINARY CASE ANALYSIS

In this section, we illustrate the use of our *3De lens* to inspect wind turbine aerodynamics using a publicly available data set [33]; we refer the reader to the supplementary video for complete frames. One's main interest here is to answer whether kinetic energy transported by the wind is efficiently transformed into rotational kinetic energy for each set of blades, e.g., in view of improving the economic viability of renewable energy generation.

The data at our disposal consist of wind turbines (total of 362K surface triangles) and air flow (5K streamlines); pressure is defined on the blades (surface attribute) and streamlines are computed from fluid speed (volumetric attribute). By using a 3De lens, we first investigate whether the pressure gradients in the blades are strong—a requirement for efficient energy conversion. In Fig. 1-left, the leftmost blade has a clear, strong pressure gradient (represented with a cool-to-warm colormap), which amounts to satisfactory energy transfer to this blade. Alas, one can see from the same picture that the wind flow transitions from laminar to possibly turbulent near the middle of the turbine (centre of the figure); this may reduce the turbine efficiency as disorderly flow is unlikely to create the required strong pressure gradients. Further inspection with the 3De lens confirms small pressure gradients on the other blades, thus suggesting an unsatisfactory energy conversion. By inspecting how the fluid flow behaves past the first blade with the lens (by filtering streamlines passing through the lens and retaining only those with the same direction as the lens orientation), we observe that such disorderly behaviour is due to the left and centre blades interacting with the flow (Fig. 1-right).

## 6 CONCLUSION AND FUTURE WORK

We presented *3De lens*, a focus+context technique for multi-geometry 3D visualizations. It combines 3D and Decal lenses, and allows flexible use of either one or both lenses on data types they best operate on. We demonstrated the potential use of our lens on two domain visualizations, with surfaces and streamlines. For future work, we would like to tackle other domain scenarios and data geometries. We also plan to support the use of simultaneous multiple lenses, investigating ways of easily applying and composing 3De lenses. In this sense, we are particularly inspired by Rocha *et al.*'s composite operations for decal-lenses [20]—e.g., one could brush or lasso over the surface to conceive lens-regions of arbitrary shapes.

Moreover, we discussed the major problem faced by 2D interaction systems of manipulating lenses in 3D, and how it can be aided by VR technology. We plan to expand upon 3D interactions that allow users to work through or parametrize the lens. For instance, to enable one to set a local parametrization on the decal patch as to condition an angular-based selection over a surface; or even to adjust the angular tolerance to select wider/narrower bundles of curves. [1, 5, 14, 15, 26, 27]

## REFERENCES

- [1] Norwegian Petroleum Technology: A Success Story. Technical report, Norwegian Academy of Technological Science (NTVA), Offshore Media Group, INTSOK, 2005. ISBN: 82-7719-051-4.
- [2] HTC Vive. <http://www.vive.com>, Accessed: 2018-05-28.
- [3] VisItusers.org - Blood Flow Aneurysm Tutorial Dataset Exploration. <http://www.tinyurl.com/jcdm1v>, Accessed: 2018-05-28.
- [4] Unity 3D. <https://unity3d.com>, Accessed: 2018-06-10.
- [5] S. Allen. The Canadian Economy at a Glance. <http://www.investorsfriend.com/canadian-gdp-canadian-imports-and-exports/>, July 2018. Accessed: 2018-09-07.
- [6] A. Bærentzen, S. L. Nielsen, M. Gjøll, B. D. Larsen, and N. J. Christensen. Single-pass wireframe rendering. In *ACM SIGGRAPH*, 2006. doi: 10.1145/1179849.1180035
- [7] L. D. Brown and H. Hua. Magic lenses for augmented virtual environments. *IEEE Computer Graphics and Applications*, 26(4):64–73, 2006. doi: 10.1109/MCG.2006.84
- [8] D. Coffey, N. Malbraaten, T. Le, I. Borazjani, F. Sotiropoulos, A. G. Erdman, and D. F. Keefe. Interactive slice wim: Navigating and interrogating volume data sets using a multisurface, multitouch vr interface. *IEEE Transactions on Visualization and Computer Graphics*, 18(10):1614–1626, 2012. doi: 10.1109/TVCG.2011.283
- [9] M. Cordeil, B. Bach, Y. Li, E. Wilson, and T. Dwyer. Design space for spatio-data coordination: Tangible interaction devices for immersive information visualisation. In *IEEE Pacific Visualization Symposium*, pp. 46–50, 2017. doi: 10.1109/PACIFICVIS.2017.8031578
- [10] N. Elmqvist, A. Vande Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T. J. Jankun-Kelly. Fluid interaction for information visualization. *Information Visualization*, 10(4):327–340, 2011. doi: 10.1177/1473871611413180
- [11] A. Fuhrmann and E. Gröller. Real-time techniques for 3d flow visualization. In *Proceedings of the Conference on Visualization*, pp. 305–312, 1998.
- [12] R. Gasteiger, M. Neugebauer, O. Beuing, and B. Preim. The flowlens: A focus-and-context visualization approach for exploration of blood flow in cerebral aneurysms. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2183–2192, 2011. doi: 10.1109/TVCG.2011.243
- [13] R. Gasteiger, M. Neugebauer, C. Kubisch, and B. Preim. Adapted surface visualization of cerebral aneurysms with embedded blood flow information. In *Eurographics Conference on Visual Computing for Biology and Medicine*, pp. 25–32, 2010. doi: 10.2312/VCBM/VCBM10/025-032
- [14] J. Jerald. *The VR Book: Human-Centered Design for Virtual Reality*. ACM Books (Book 8). Morgan Claypool Publishers, 2015. ISBN: 19-7000-112-7.
- [15] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. *Visual Analytics: Definition, Process, and Challenges*, vol. 4950. Springer, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-70956-5\_7
- [16] J. Looser, R. Grasset, and M. Billinghurst. A 3d flexible and tangible magic lens in augmented reality. In *International Symposium on Mixed and Augmented Reality*, pp. 1–4, 2007. doi: 10.1109/ISMAR.2007.4538825
- [17] R. C. R. Mota, S. Cartwright, E. Sharlin, H. Hamdi, M. Costa Sousa, and Z. Chen. Exploring immersive interfaces for well placement optimization in reservoir models. In *Symposium on Spatial User Interaction*, pp. 121–130, 2016. doi: 10.1145/2983310.2985762
- [18] F. Niebling. Spatial Interaction for the Post-Processing of 3D CFD Datasets. In *EuroVis - Short Papers*, 2017. doi: 10.2312/eurovisshort.20171136
- [19] I. Popyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst. Egocentric object manipulation in virtual environments: Empirical evaluation of interaction techniques. *Computer Graphics Forum*, 17(3):41–52, 2001. doi: 10.1111/1467-8659.00252
- [20] A. Rocha, J. D. Silva, U. Alim, S. Carpendale, and M. C. Sousa. Decalenses: Interactive lenses on surfaces for multivariate visualization. *IEEE Transactions on Visualization and Computer Graphics*, in press, 2018. doi: 10.1109/TVCG.2018.2850781
- [21] P. Rustagi. Silhouette line display from shaded models. *IRIS Universe*, 1989(9):42–44, 1989.
- [22] M. Schirski, T. Kuhlen, M. Hopp, P. Adomeit, S. Pischinger, and C. Bischof. Efficient visualization of large amounts of particle trajectories in virtual environments using virtual tubelets. In *ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry*, pp. 141–147, 2004. doi: 10.1145/1044588.1044615
- [23] S. Somanath, S. Carpendale, E. Sharlin, and M. C. Sousa. Information visualization techniques for exploring oil well trajectories in reservoir models. In *Graphics Interface*, pp. 145–150, 2014.
- [24] M. Spindler, S. Stellmach, and R. Dachselt. Paperlens: Advanced magic lens interaction above the tabletop. In *ACM International Conference on Interactive Tabletops and Surfaces*, pp. 69–76, 2009. doi: 10.1145/1731903.1731920
- [25] J. Steimle, A. Jordt, and P. Maes. Flexpad: Highly flexible bending interactions for projected handheld displays. In *SIGCHI Conference on Human Factors in Computing Systems*, pp. 237–246, 2013. doi: 10.1145/2470654.2470688
- [26] M. Sumrow. Norsk hydro optimizes well paths with virtual reality technology. May 2002. [Online; posted 13-May-2002].
- [27] M. Sumrow. Norsk hydro optimizes well paths with virtual reality technology. *Oil and Gas Journal*, 100(19):53–54, 2002.
- [28] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. Interactive lenses for visualization: An extended survey. *Comput. Graph. Forum*, 36(6):173–200, 2017. doi: 10.1111/cgf.12871
- [29] 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, 2016. doi: 10.1109/TVCG.2015.2502583
- [30] X. Tong, C. Li, and H. W. Shen. Glyphlens: View-dependent occlusion management in the interactive glyph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):891–900, 2017. doi: 10.1109/TVCG.2016.2599049
- [31] R. Van Pelt, J. Olivan Bescos, M. Breeuwer, R. E. Clough, M. E. Gröller, B. ter Haar Romenij, and A. Vilanova. Exploration of 4d mri blood flow using stylistic visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1339–1347, 2010. doi: 10.1109/TVCG.2010.153
- [32] J. Viega, M. J. Conway, G. Williams, and R. Pausch. 3d magic lenses. In *ACM Symposium on User Interface Software and Technology*, pp. 51–58, 1996. doi: 10.1145/237091.237098
- [33] *Visualize This* Contest. Compute Canada and West Grid. <https://tinyurl.com/y9rmb3et>, Accessed: 12-June-2018.
- [34] L. Wang, Y. Zhao, K. Mueller, and A. Kaufman. The magic volume lens: An interactive focus+context technique for volume rendering. In *IEEE Visualization*, pp. 367–374, 2005. doi: 10.1109/VISUAL.2005.1532818
- [35] C. Ware and D. R. Jessome. Using the bat: a six-dimensional mouse for object placement. *IEEE Computer Graphics and Applications*, 8(6):65–70, 1988. doi: 10.1109/38.20319