

# Exploring Aerospace Design in Virtual Reality with Dimension Reduction

Slawomir K. Tadeja\*, Pranay Seshadri†, Per Ola Kristensson‡

*University of Cambridge, Cambridge, CB2 1PZ, U. K.*

*The Alan Turing Institute, London, NW1 2DB, U. K.*

One of the today’s most propitious immersive technologies is virtual reality (VR). This term is colloquially associated with head-sets that transport users to a bespoke, built-for-purpose immersive 3D virtual environment. It has given rise to the field of immersive visual analytics—a new field of research that aims to use immersive technologies for enhancing and empowering data analytics. In this paper we present a VR aerospace design environment with the objective of aiding the component aerodynamic design process by interactively visualizing performance and geometry. This virtual environment uses ideas from parameter-space dimension reduction to enhance the exploration and exploitation of the design space. We decompose the design of such an environment into function structures, present an implementation of the system, and verify the interface in terms of usability and expressiveness.

## I. Introduction

Virtual Reality (VR) is rapidly being hailed as the new paradigm for interactive visualization of data. Its ability to fuse visual, audio, and haptic sensory feedback in a computer-generated simulation environment is deemed to have tremendous potential. While the phrase *virtual reality* has been used for decades, in the context of computer aided visualization, today it is synonymous with head-mounted displays<sup>1</sup> (HMDs) or headsets.<sup>2,3</sup> Although still in a nascent stage, HMDs have demonstrated their usefulness in the computer gaming, education, fashion and real-estate industries, with countless more application areas currently being pursued, including information visualization in aerospace.<sup>4</sup> One potentially promising application is aerospace design—a complex, multi-disciplinary, multi-objective and multi-dimensional problem—where technologies that offer faster design cycle times, with potentially greater efficiency gains, can be real game-changers. However, as the aerospace community usually works on state-of-the-art computational tools and sophisticated computer-aided design packages, there are tremendous hurdles in getting the community to embrace VR. Furthermore, at this stage, it is not precisely clear what the benefits are in migrating to a VR-based design framework. Thus, what is required by the aerospace design community is an initial sketch of an immersive VR aerospace design environment—a computer-generated environment that leverages the full visual, audio, and haptic sensory frameworks afforded by VR technology.

Our focus in this paper is to explore how aerospace design workflows can benefit from VR. To aid our effort, we will be using ideas from parameter-space dimension reduction.<sup>5,6</sup> This topic has recently received considerable attention from both the applied mathematics and computational engineering communities, where the aim has been to reduce the cost of expensive computer parameter studies—that is, optimization, uncertainty quantification, and more generally *design of experiments*. In II, we present some of the key theoretical ideas. This is followed in section III with a presentation of the VR aerospace design environment including its function structures and interaction features. The next section IV describes a verification of the interface with respect to usability and expressiveness. Section V summarizes the contributions and outlines future work.

---

\*PhD Student, Department of Engineering, University of Cambridge, [skt40@cam.ac.uk](mailto:skt40@cam.ac.uk),

†Postdoctoral Fellow, Department of Engineering, University of Cambridge, and Group Leader (Aerospace), Data-Centric Engineering, The Alan Turing Institute [ps583@cam.ac.uk](mailto:ps583@cam.ac.uk), [pseshadri@turing.ac.uk](mailto:pseshadri@turing.ac.uk)

‡Reader, Department of Engineering, University of Cambridge, [pok21@cam.ac.uk](mailto:pok21@cam.ac.uk)

## II. Parameter-space dimension reduction

Consider a function  $f(\mathbf{x})$  where  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ . Here  $f$  represents our chosen quantity of interest (qoi); the desired output of a computational model. This qoi can be the lift coefficient of a wing or indeed the efficiency of a turbomachinery blade. Let  $\mathbf{x} \in \mathbb{R}^d$  be a vector of design parameters. Now when  $d \leq 2$ , visualizing the design space of  $f$  is trivial, one needs to simply run a design of experiment and view the results as a scatter plot. However, when  $d \geq 3$  visualizing the design space becomes difficult. One way forward is to approximate  $f$ , with

$$f(\mathbf{x}) \approx g(\mathbf{U}^T \mathbf{x}), \quad (1)$$

where  $g : \mathbb{R}^m \rightarrow \mathbb{R}$  and  $\mathbf{U} \in \mathbb{R}^{d \times m}$ , with  $m \leq d$ . We call the subspace associated with the span of  $\mathbf{U}$  its *ridge subspace* and  $g(\mathbf{U}^T \mathbf{x})$  its *ridge approximation*. Further, we assume that the columns of  $\mathbf{U}$  are orthonormal, i.e.,  $\mathbf{U}^T \mathbf{U} = \mathbf{I}$ . The above definitions imply that the gradient of  $f$  is nearly zero along directions that are orthogonal to the subspace of  $\mathbf{U}$ . In other words, if we replace  $\mathbf{x}$  with  $\mathbf{x} + \mathbf{h}$  where  $\mathbf{U}^T \mathbf{h} = 0$ , then  $f(\mathbf{x} + \mathbf{h}) = g(\mathbf{U}^T(\mathbf{x} + \mathbf{h})) = f(\mathbf{x})$ . Visualizing  $f_i$  along the coordinates of  $\mathbf{x}_i^T \mathbf{U}$  for all designs  $i$ , can provide extremely powerful inference; such scatter plots are called *sufficient summary plots*.

### A. Techniques for dimension reduction

Techniques for estimating  $\mathbf{U}$  build on ideas from sufficient dimension reduction<sup>6</sup> and more recent works such as active subspaces<sup>5</sup> and polynomial<sup>7,8</sup> and Gaussian<sup>9</sup> ridge approximations. While our work in this paper is invariant to the specific parameter-space dimension reduction technique utilized, we briefly detail a few ideas within ridge approximation. Our high-level objective is to solve the optimization problem

$$\underset{\mathbf{U} \in \mathbb{R}^{d \times m}, \alpha \in \mathbb{R}^p}{\text{minimize}} \quad \|f(\mathbf{x}) - g_\alpha(\mathbf{U}^T \mathbf{x})\|_2^2, \quad (2)$$

over the the space of matrix manifolds  $\mathbf{U}$  and the coefficients (or hyperparameters) associated with the parametric function  $g$ . This is a challenging optimization problem and it is not convex. In<sup>9</sup> the authors assume that  $g$  is the posterior mean of a Gaussian process (GP) and iteratively solve for the hyperparameters associated with the GP, whilst optimizing  $\mathbf{U}$  using a conjugate gradient optimizer on the Stiefel manifold (see<sup>10</sup>). In<sup>8</sup> the authors set  $g$  to be a polynomial and iteratively solve for its coefficients—using standard least squares regression—whilst optimizing over the Grassman manifold to estimate the subspace  $\mathbf{U}$ . It should be noted that these techniques are motivated by the need to break the *curse of dimensionality*. In other words, one would like to estimate both  $g$  and  $\mathbf{U}$  for a  $d$  dimensional, scalar-valued, function  $f$  without requiring a large number of computational simulations.

The dimension reduction strategy we pursue in this paper is based on active subspaces,<sup>5</sup> that is a computational heuristic tailored for identifying subspaces that can be used for the approximation in (2). Broadly speaking, active subspaces requires the approximation of a covariance matrix  $\mathbf{C} \in \mathbb{R}^{d \times d}$

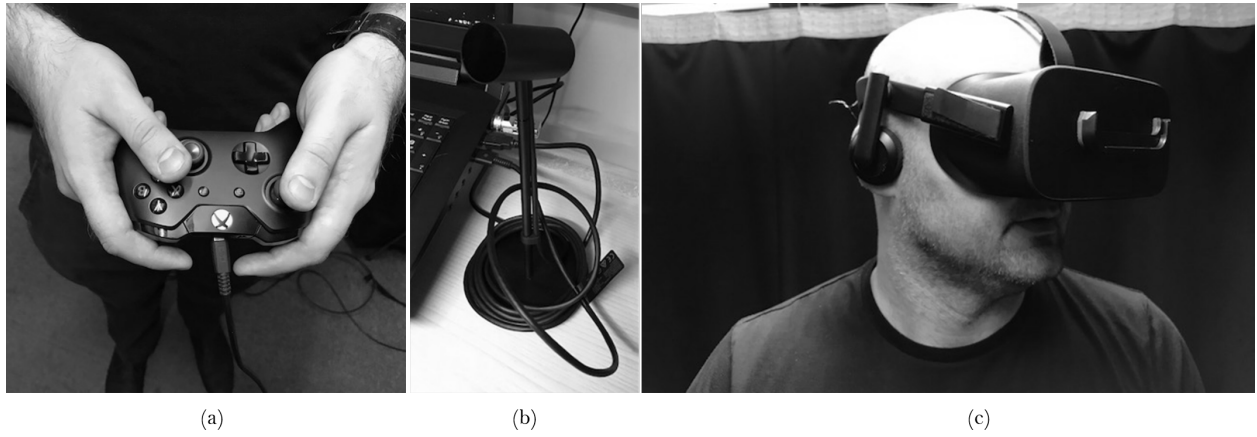
$$\mathbf{C} = \int_{\mathcal{X}} \nabla_x f(\mathbf{x}) \nabla_x f(\mathbf{x})^T \boldsymbol{\rho}(\mathbf{x}) d\mathbf{x}, \quad (3)$$

where  $\nabla f(\mathbf{x})$  represents the gradient of the function  $f$  and  $\boldsymbol{\rho}$  is the probability density function that characterizes the input parameter space  $\mathcal{X} \in \mathbb{R}^d$ . The matrix  $\mathbf{C}$  is symmetric positive semi-definite and as a result it admits the eigenvalue decomposition

$$\mathbf{C} = \mathbf{W} \boldsymbol{\Lambda} \mathbf{W}^T = \begin{pmatrix} \mathbf{W}_1 & \mathbf{W}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Lambda}_1 & \\ & \boldsymbol{\Lambda}_2 \end{pmatrix} \begin{pmatrix} \mathbf{W}_1 & \mathbf{W}_2 \end{pmatrix}^T, \quad (4)$$

where the first  $m$  eigenvectors  $\mathbf{W}_1 \in \mathbb{R}^{d \times m}$ , where  $m \ll d$ —selected based on the decay of the eigenvalues  $\boldsymbol{\Lambda}$ —are on average directions along which the function varies more, compared to the directions given by the remaining  $(d - m)$  eigenvectors  $\mathbf{W}_2$ . Readers will note that the notion of computing eigenvalues and eigenvectors of an assembled covariance matrix is analogous to principal components analysis (PCA). However, in (3) our covariance matrix is based on the average outer product of the gradient, while in PCA it is simply the average outer product of samples, i.e.,  $\mathbf{x}\mathbf{x}^T$ . Now, once the subspace  $\mathbf{W}_1$  has been identified, one can approximate  $f$  via

$$\begin{aligned} f(\mathbf{x}) &= f(\mathbf{W}\mathbf{W}^T \mathbf{x}) = f(\mathbf{W}_1 \mathbf{W}_1^T \mathbf{x} + \mathbf{W}_2 \mathbf{W}_2^T \mathbf{x}) \\ &\approx g(\mathbf{W}_1^T \mathbf{x}), \end{aligned} \quad (5)$$



**Figure 1. Hardware interfaces in VR:** (a) Video game controller (Xbox) with the user keeping his left thumb on the left-hand joystick and both of his point fingers at the triggers placed in front-bottom of the controller; (b) motion sensor (Oculus Rift); (c) VR headset (Oculus Rift).

in other words we project individual samples  $\mathbf{x}_i$  onto the subspace  $\mathbf{W}_1$ . Moreover, as the function (on average) is relatively flat along directions  $\mathbf{W}_2$ , we can approximate  $f$  using the directions encoded in  $\mathbf{W}_1$ .

But how do we compute (3), as for a given  $f$  we may not necessarily have access to its gradients? In,<sup>11</sup> the authors construct a global quadratic model to a 3D Reynolds Averaged Navier Stokes (RANS) simulation of a turbomachinery blade and then analytically estimate its gradients. We detail their strategy below as we adopt the same technique for facilitating parameter-space dimension reduction.

Assume we have  $N$  input-output pairs  $\{\mathbf{x}_i, f_i\}_{i=1}^N$  obtained by running a suitable design of experiment (see<sup>12</sup>) within our parameter space. We assume that the samples  $\mathbf{x}_i \in \mathbb{R}^d$  are independent and identically distributed and that they admit a joint distribution given by  $\mathbf{x}$ . Here we will assume that  $\rho\mathbf{x}$  is uniform over the hypercube  $\mathcal{X} \in [-1, 1]^d$ . We fit a global quadratic model to the data,

$$f(\mathbf{x}) \approx \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{c}^T \mathbf{x} + d, \quad (6)$$

using least squares. This yields us values for the coefficients  $\mathbf{A}, \mathbf{c}$  and the constant  $d$ . Then, we estimate the covariance matrix in (3) using

$$\hat{\mathbf{C}} = \int_{\mathcal{X}} (\mathbf{A}\mathbf{x} + \mathbf{c}) (\mathbf{A}\mathbf{x} + \mathbf{c})^T \rho(\mathbf{x}) d\mathbf{x}. \quad (7)$$

Following the computational of the eigenvectors of  $\hat{\mathbf{C}}$ , one can then generate sufficient summary plots that are useful for subsequent inference and approximation.

We apply this quadratic recipe and show the sufficient summary plots for a 3D turbomachinery blade in section III, both in a standard desktop environment and in virtual reality. This comparison—the central objective of this paper—is motivated by the need to explore the gains VR technologies can afford in aerospace design. That said, prior to delving into our chosen case study, an overview of existing immersive visual technologies and their associated frameworks is in order.

### III. Supporting aerospace design in virtual reality

Visual analytics (VA), a phrase first coined by J.J. Thomas et al.<sup>13</sup> has two ingredients: interactive visual interfaces<sup>13</sup> and analytical reasoning.<sup>13</sup> Recent advances and breakthroughs in the development of VR (Virtual Reality), MR (Mixed Reality) and AR (Augmented Reality) have spun off another branch of research known as Immersive Visual Analytics<sup>14</sup> (IVA). IVA seeks to understand how the latest wave of immersive technologies can be leveraged to create more compelling, more intuitive and more effective visual analytics frameworks.

There are still numerous hurdles to overcome for VR-based data analytics tools to be widespread. Issues associated with any type of a 3D interface, for example, potential occlusion effects,<sup>15</sup> the requirement of vast

computing resources and specialized, (often costly) additional hardware, for example, the headset and necessary additional sensors have to be resolved or at least minimized. Moreover, interaction techniques have to facilitate a user's understanding of the visualization and, at the same time not serve as a distraction. Finally, certain guidelines have to be incorporated to mitigate the risk of the simulation sickness<sup>16,17</sup> symptoms that can manifest during or immediately after the use of a VR headset.

Many interaction techniques and devices have been conceived that can be used separately, or in combination, for interaction and control of the VR environment. In this paper we use the standard Xbox<sup>18</sup> controller shown in Fig. 1(a)) that comes prepacked with the Oculus Rift<sup>2</sup> bundle (see Fig. 1(b) and Fig. 1(c)). This controller-style has achieved very high adoption in gaming and the design is ergonomic and easy to learn. As an example of wider option, the US Navy recently adopted the use of a Xbox controller to operate the periscope on nuclear-powered submarine<sup>a</sup>.

We here present a VR aerospace design environment with a focus on dimension reduction. VR technology is starting to gain ground in aerospace design as pointed out by Garca-Hernandez et al.,<sup>4</sup> which list a range of aerospace research topics in which VR already is, or can be, successfully applied. This includes, among others, *spacecraft design optimization* (e.g. Stump et al.<sup>19</sup> used IVA to aid a satellite design process) and *aerodynamic design*, in which 3D scatter plots are already in-use (see Jeong et al.<sup>20</sup>).

Garca-Hernandez et al.<sup>4</sup> suggests that three elements are especially promising for a VR-based approach: 1) integration of multiple 2D graphs for 3D data; 2) 3D parallel coordinates; and 3) visualization of complex graphs. In this paper we explore the aerodynamic design space of a turbomachinery blade.

## A. Applications in design

Our dimension reduction results and case study is based on the work undertaken in.<sup>11</sup> Here the authors study the 25D design space of a fan blade using the quadratic active subspaces recipe detailed in II. Towards this end, we carried out the design of an experiment with  $N = 548$  3D RANS computations with different designs; the design space used in this study included five degrees of freedom specified at five spanwise locations. These degrees of freedom comprised of an axial displacement, a tangential displacement, a rotation about the blade's centroidal axis, leading edge recambering and trailing edge recambering, specified at 0, 25, 50, 75 and 100% span. Thus, we obtained values of the efficiency and pressure ratio for each design vector  $\mathbf{x}_i$ . These are two important output quantities of interest in the design of a blade. Then, by studying the eigenvalues and eigenvectors of the covariance matrix for these two objectives the authors were able to discover a 1D ridge approximation for the pressure ratio of a fan and a 2D ridge approximation for the efficiency. These sufficient summary plots are shown in Fig. 2. There are a few important remarks to make regarding these plots.

For the pressure ratio sufficient summary plot, shown in Fig. 2(a), the horizontal axis is the first eigenvector of the covariance matrix associated with the pressure ratio,  $\mathbf{u}_1$ . For the efficiency sufficient summary plot, shown in Fig. 2(b), the two horizontal axes are the first two eigenvectors of the covariance matrix associated with the efficiency  $[\mathbf{u}_1, \mathbf{u}_2]$ . It is important to note that the subspaces associated with efficiency and pressure ratio are distinct.

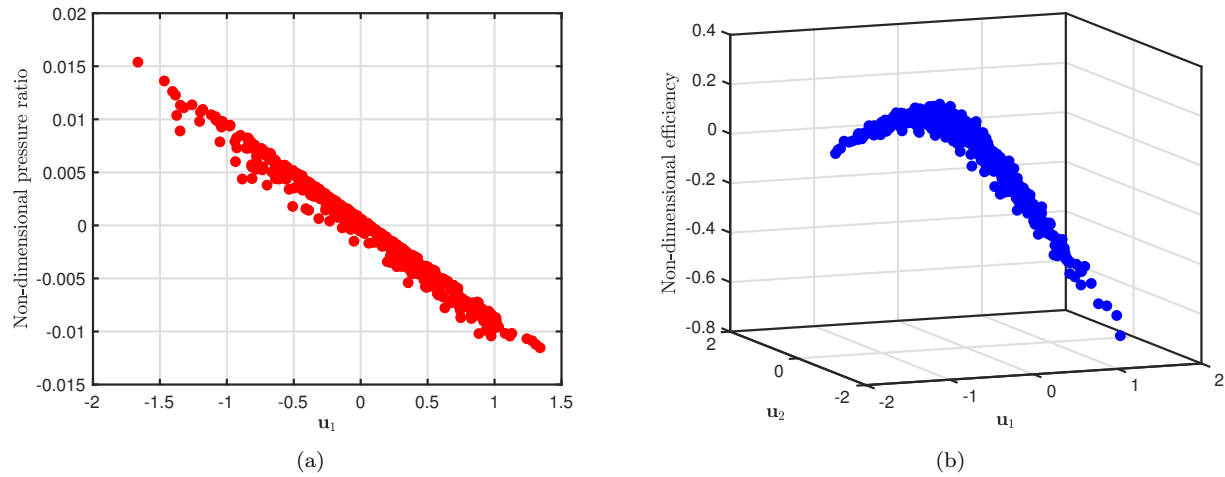
The sufficient summary plots above permit us to identify and visualize low-dimensional structure in the high-dimensional data. More specifically, these plots can be useful in the design process as they permit engineers to make the following enquiries:

- What linear combination of design variables is the most important for increasing / decreasing the pressure ratio?
- How do we increase the efficiency?
- What are the characteristics of designs that satisfy a certain pressure ratio?
- What are the characteristics of designs that satisfy the same efficiency?

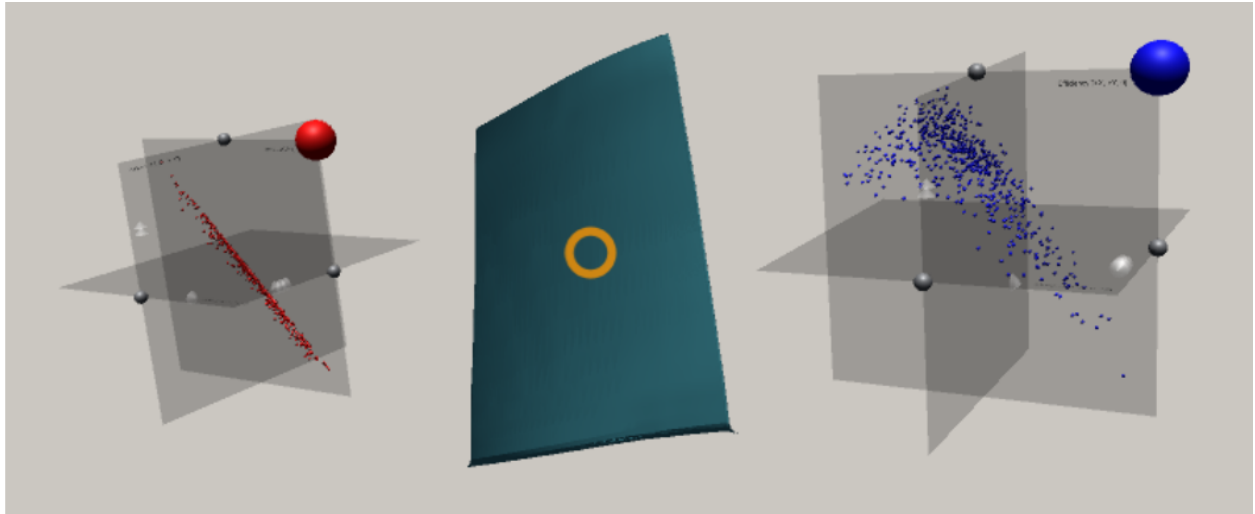
We use these sufficient summary plots in a bespoke VR environment. Our high-level objective is to ascertain whether it is possible to leverage tools in VR in conjunction with parameter-space dimension reduction to facilitate better design decision-making and inference. To achieve this goal, we seamlessly integrate the aforementioned sufficient summary plots with the 3D geometric design of the blade, i.e.,

$$\text{pressure ratio} \rightleftharpoons \text{geometry visualization} \rightleftharpoons \text{efficiency}.$$

<sup>a</sup><https://www.digitaltrends.com/cool-tech/navy-xbox-controllers-attack-submarines/>



**Figure 2.** Sufficient summary plots of (a) Pressure ratios; (b) Efficiency, for a range of different computational designs for turbomachinery blade, obtained from a design of experiment study. Based on work in.<sup>11</sup>



**Figure 3.** The user's field-of-view: the right-hand plot can, for example, show the lift coefficients whereas the left-hand side plot can contain the drag coefficient values. The nominal blade geometry is visualized in the middle between the two plots.

In other words, as the user selects a different design—by selecting a suitable level of performance from the sufficient summary plots—they visualize the geometry of the blade that yields that performance. Moreover, they should be able to compare this geometry with that of the nominal design. We clarify and make precise these notions in the forthcoming subsections.

## B. Function structures

We model the function structures of the system using FAST (Function Analysis Systems Technique).<sup>21</sup> Fig. 4 shows the function structures of the VR visualization environment for an aerospace design workflow with dimension reduction. The FAST-diagram in Fig. 4 models the level of abstraction on the horizontal axis and function sequence on the vertical axis.

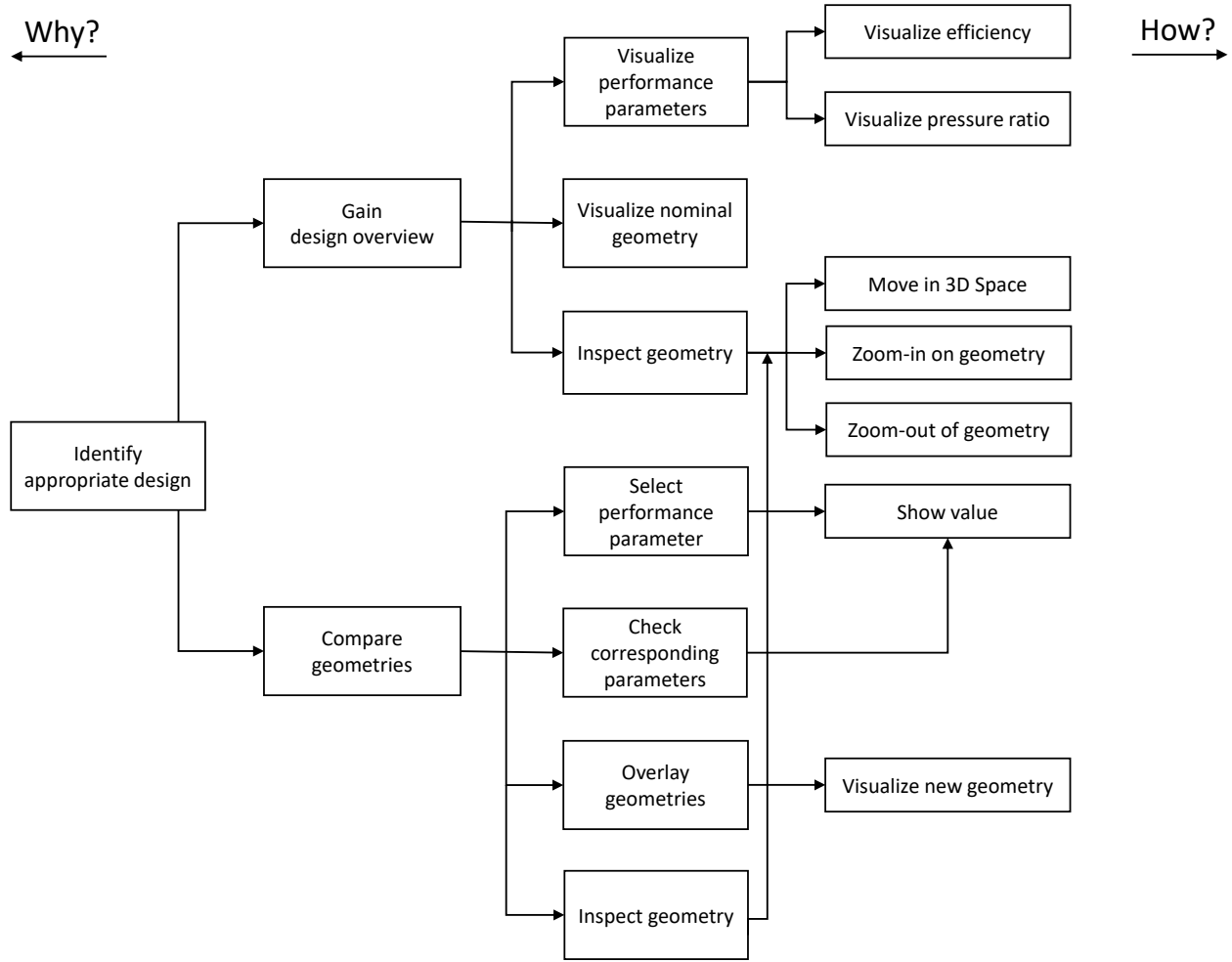


Figure 4. Function structures<sup>21</sup> allow capture and analysis of system requirements by providing a simple and efficient way of identifying the functionality that has to be incorporated by the system in order for users to accomplish their goals.

### C. Visualization framework

The visualization framework is built using Unity3D—one of the most widely used game engines with built-in VR development support. Both of the two mainstream HMD headsets provide supporting packages developed natively for Unity3D, which substantially speeds up the development process. The software is built on top of the *Unity VR Samples pack*<sup>22</sup> and uses the *Oculus Utilities for Unity*<sup>23</sup> package. In addition, we use the asset store available for the Unity3D game engine, which contains many VR-ready tools and supporting packages.

As remark, the survey by Wagner et al.<sup>24</sup> points out that game engines *do not support any data exploration* techniques. In other words, these features have to be coded and built from scratch. To allow user interaction with data we use an Xbox Controller connected with the laptop via USB cable (see Fig. 1(a)) in combination with gaze tracking through a cross-hair which moves with the user’s head and is placed straight from the camera (visualized as an orange cross-hair, see Fig. 3, Fig. 6 and Fig. 7).

### D. Interaction and movement

The interaction is designed around head tracking in combination with the standard buttons on the Xbox<sup>18</sup> controller (see Fig. 5). Supported interactions are mapped as follows:

- Left-hand joystick and [T] buttons (see Fig. 5): Triggers movement along the  $X$ - $Z$  plane and movement along the vertical axis respectively, right-hand [T] is assigned to “up” and left-hand [T] is “down”. The





**Figure 5.** The Xbox controller: (a) shows the top view with the left-hand joystick  $[J]$  used to control the 2D movement on the  $X$ - $Z$  plane whereas (b) shows the front view with the two triggers  $[T]$  responsible for vertical movement along the  $Y$  axis. The other action buttons indicated in (a) have the following meanings:  $[R]$  for reload of the visualization;  $[L]$  for loading next dataset;  $[A]$  for selection of an interactive item; and  $[X]$  for moving or rotating the scatter plots.

movement in the  $X$ - $Z$  plane is always with respect to the user's gaze. This manoeuvring combination permits the user to move in any direction and in any position in 3D space. The user moves with constant velocity and with fluid movement to ensure the user is receiving continuous closed-loop visual feedback on their changing position in relation to the surroundings.

- Action button  $[A]$ : Selection of an interactive element, such as a scatter plot rotation and movement selector, or a data point (see Fig. 7). Objects highlight themselves when the user's gaze, as indicated by a cross-hair, is on them. Double-tapping on the  $[A]$  button selects the highlighted object.
- Button  $[X]$ : When tapped after the selection of a scatter plot point, it will re-position the point to a certain distance towards the user's present gaze direction. Furthermore, if the rotation selector is active, selecting this button will initiate the scatter plot's rotation over  $90^\circ$  based on the current direction of the user's gaze.
- Button  $[R]$ : Resets the visualization and all its associated elements to their original state.
- Button  $[L]$ : This button loads the next dataset.

### E. Blade visualization

As alluded to previously, the central artifact in our VR environment is the geometry of the designs. Our virtual environment contains as many geometries as there are data points, resulting in a total of 548 stereo lithography (STL) files. Hence, whenever a data point is selected on one of the plots, the accompanying shape is instantly visualized. To provide the user with a quick and an effective way of comparing the new perturbed design, the nominal geometry is still kept visible and rendered as a translucent object, as can be seen in Fig. 6. This solution, combined with unlimited movement dexterity, allows the user to visually inspect and observe any differences between the two geometries. Furthermore, by simply changing their position, or by tilting their head (thereby changing the rotational angle), the user can zoom-in and zoom-out on any of the blades parts for a close inspection as presented in Fig. 6(f).

### F. Sufficient summary plots

The key element in the framework is the sufficient summary plots; they are visualized as three fixed-size, axis-aligned translucent orthogonal rectangles. The data points are scaled so the values of their respective coordinates are within the range of the translucent surfaces. When any of the spheres denoting a data point

is selected, the marker lights up and switches to a selection color (light green). Moreover, as we have a 1:1 mapping between the plots, the corresponding design on the other plot is selected. Furthermore, a number of semitransparent cones<sup>b</sup> was embedded into these sufficient summary plots to denote their axes: one for the  $X$ -axis, two for the  $Y$ -axis and three for the  $Z$ -axis. Selection of a shape with its pointing tip has an additional advantage—the orientation informs the user of the positive side of a given axis. Each selection can be reverted by double-tapping the  $[A]$  button while gazing over it.

The 3D spheres in the plots were used to denote both the data points and various selectors' markers. Using shape perception has a long-standing application history for VR-based visualizations. For instance, Ribarsky et al.<sup>25</sup> used simple 3D shapes such as cones, spheres and cuboids in their system. Ribarsky et al.<sup>25</sup> also highlight that glyphs with their intrinsic characteristics, such as “*position, shape, color, orientation and so forth*”<sup>25</sup> are very useful when visualizing complex datasets.

### 1. Initial placement and re-positioning

The two scatter plots—one for pressure ratio and another for efficiency—are automatically positioned on both sides of the blade, which in turn is positioned in front of the initial user's field of view; see Fig. 3. The plots are placed at the same, pre-configured distance from the user, at a roughly 45° angle from the  $X$ -axis.

To ensure that the user does not feel constrained in a nearby region and to make better use of virtually infinite 3D space provided by the VR-environment, users are provided with the possibility of moving the scatter plots. The interaction occurs via head-tracking and the *select & move* metaphor. Every scatter plot has a color-coded interactive sphere, that is, a selector (see Fig. 7(b)), attached to it in the right-hand top corner. When the user's gaze hovers over it, the selector automatically highlights it and, if selected by double-tapping the  $[B]$  button, changes its color to orange (see Fig. 7(b)). If the user presses the  $[X]$  button while a selection is active, the scatter plot is re-positioned at a certain distance towards the point determined by the user's current gaze. If the button is held the plot will follow the cross-hair's movement.

It is also possible to move both plots at once if more selectors are simultaneously active. In such a scenario, to keep the current relative position of these scatter plots, a barycentre  $B = (x_B, y_B, z_B)$  of all these objects is calculated using the conventional geometric formula:

$$B = \frac{1}{N}(x_B = \sum_{i=0}^N x_i, y_B = \sum_{i=0}^N y_i, z_B = \sum_{i=0}^N z_i). \quad (8)$$

This point is moved along the forward vector from the camera in the same manner as in the case of a single scatter plot. Selected objects are then grouped together and displaced with respect to the new position of the barycentre whilst simultaneously keeping their internal (current position with respect to the local axes) and external (axis-alignment of surfaces) rotations.

### 2. Scatter plot rotation

The scatter plot can be rotated in 90° steps about one of the three main axes. This is achieved by double-tapping the  $[A]$  button while gazing over one of the three rotation selectors (see Fig. 7 (b-c)). A further press of the  $[X]$  button will result in a rotation of all the plot's components, including data points, rotation and movement selectors, and axis cone markers. If users are situated in such a way that their gaze is located exactly in front of the active axis selector, the rotation will occur towards the direction provided by the camera's forward vector. Similar to the movement, the plot's orientation in the global coordinate system will not be affected.

### 3. Relationships between the visualization elements

The data points on one of the plots are correlated in a 1:1:1 (one-to-one-to-one) manner onto the other plot and vice-versa. Moreover, each of the data points is mapped onto a one-and-only-one unique blade design. Therefore, whenever a marker is selected on one of the plots the system will automatically highlight the corresponding data point on the other plot and switch the visualized blade onto the new, corresponding shape. Furthermore, the nominal shape will be kept as a translucent point of reference (see Fig. 6) that

<sup>b</sup>W. Kresse, used under CC BY-SA 3.0; <http://wiki.unity3d.com/>



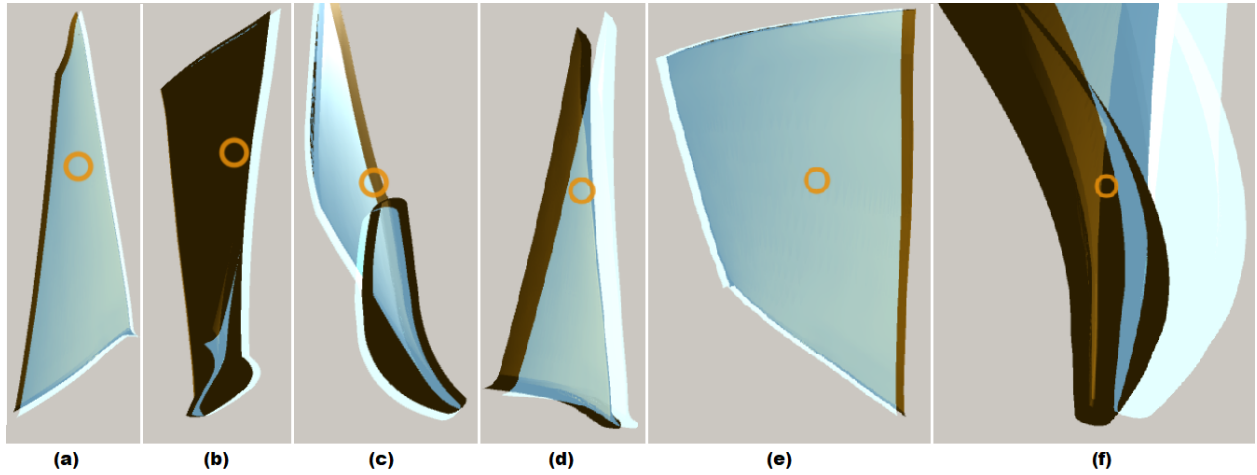


Figure 6. By selecting any data point on any plot, the user can immediately observe the blade’s geometry associated with this particular design. Moreover, the user can observe and compare the differences between the nominal and perturbed geometries as the former is kept rendered as a semi-transparent shape overlaying the latter. As users can freely maneuver in 3D space they can visually inspect the entire blade from any direction and zoom in on any of its parts, as shown in (a-f).

overlays the new design to show the user how, where, and to what degree, the new shape differs from the nominal one.

#### 4. Labeling

Both the data points and the axes are automatically labeled. In case of the latter, the strings embedded into the edges of the semitransparent rectangles denoting the axes are read directly from the input text file (see Fig. 7 (a)). The labeling of the data points is only visible once the user is hovering with his or her gaze over the marker (for example, a sphere) and disappears once the user looks at another point, or other parts of the visualization (see Fig. 7 (a)). Furthermore, the small box with the values associated with the point (in this case its coordinates) is always rotated towards the user and follows their gaze. It is rendered on top of any other visualization elements as seen in Fig. 7 (a).

## IV. Interface Verification

The VR aerospace design environment in this paper is still at an early stage and the objective of this paper is not to present a complete solution but to demonstrate potential benefits of VR for aerospace design.

As a result we here verify the fundamental usability of the system using two formative evaluation methods. First, we assess the usability of the system using Nielsen’s<sup>26,27</sup> guidelines. Second, we use the cognitive dimensions of notations framework<sup>28,29</sup> to reason about the expressiveness of the system.

### A. Usability

- *Visibility of system status:* The system provides immediate feedback to users in response to their actions. For instance, whenever a user’s gaze hovers over an interactive object (such as, for example, movement and rotation selectors or a data point) it is instantly highlighted. In addition, once an object is selected, the object also changes its color in response. In addition, following the selection of a data point its corresponding data points on the other plot are also simultaneously selected and the accompanying geometry is loaded automatically. This ensures the user remains synchronized with the system’s current state. Finally, whenever the user’s gaze is hovering over an object, a gaze-locked text is also displayed to the user, which reveals values associated with this particular data point.
- *Match between the system and the real-world:* First, the system uses the cross-hair concept which well known in the real-world to focus and help guiding users’ gaze and hence their attention on the objects

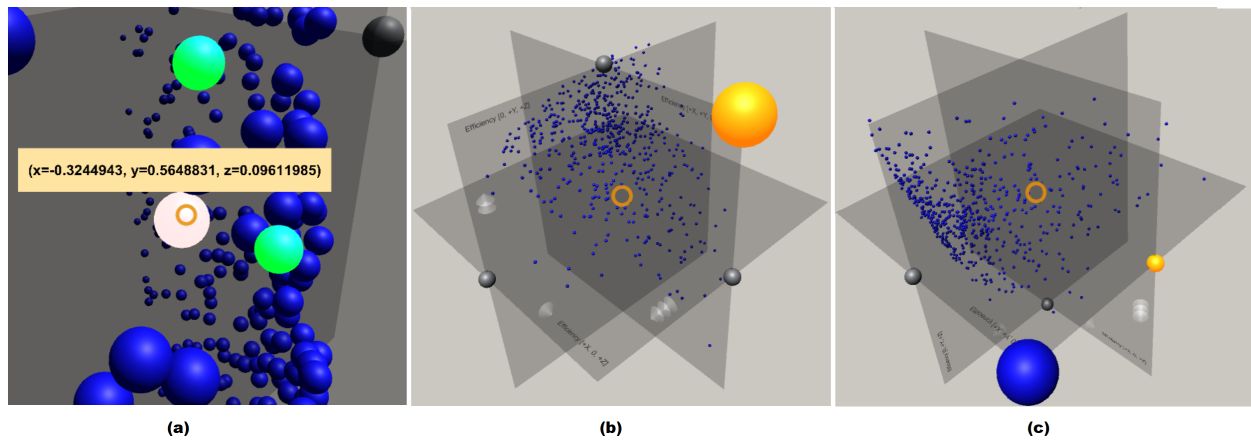


Figure 7. Screen shots of the scatter plot as it is seen by the user. (a) shows a user's gaze (orange cross-hair) hovering over a data point which instantly displays the associated values (e.g. its coordinates) in. Visible, formerly selected points (in green) are reflected on the other scatter plot and the associated geometries were also shown. (b) shows the same plot from a distance. Moreover, the highlighted spheres (in orange) which are the movement selectors - if users look at any point at space and taps the [X] button on the controller the plots will be translated towards that point at space. Whereas (c) presents it rotation by a  $90^\circ$  towards the user along the  $X$ -axis with the axis rotation selector highlighted (in orange). Only a single rotation selector can be active at once across all the scatter plots.

placed directly behind or near it. Moreover, the 3D scatter plots were designed to immediately resemble their two- or three-dimensional desktop-based counterparts. In addition, initially a user observes all the main elements of the visualization in the field-of-view placed at a roughly the same height and direction as how they would be perceived in the real-world if they were visualized on a standard computer displays.

- *User control and freedom:* The user can either load the new set of data or reload the entire visualization with a click of a button. No direct “undo” and “redo” actions were directly implemented, however, users can always deselect any object or redo the last executed operation. Moreover, using a combination of the gaze- and head-tracking and controller-based interaction, users can locate themselves at any position in 3D space.
- *Consistency and standards* and *Flexibility and efficiency of use:* As the VR in its current, almost fully immersive form is a fairly recent development, the technology itself, not to mention its main applicability areas or interaction design principles, are not yet fully understood. However, the system design is as consistent as possible, for instance, all interactive objects can be (de)selected using exactly the same method.
- *Error prevention* and *Help users recognize, diagnose, and recover from errors:* Measures to prevent the user from errors, such as missing or broken input data (for example, geometries and data points), are directly built into the system. As there is a 1:1:1 mapping between the system elements (data points) on the two plots and the geometries, missing any of the elements would lead to omitting this particular entry from the visualization and detailed information of such event being written into the log file. Hence, the main error-prone conditions are eliminated. In addition, the system incorporates certain constraints, such as a user cannot have more than a single axis-rotation selected at the time—the new selection will automatically deselect any previously activated selector. This prevents an error caused by the system being unable to recognize about which axis the scatter plot should be rotated.
- *Recognition rather than recall* and *Aesthetic and minimalist design:* The blade's geometry is visualized “as it is” in the real-world. Moreover, both plots use volumetric glyphs to denote the points which reassembles the 2D scatter plots versions (see Fig. 7). Previously selected data points are also highlighted. Furthermore, the system was designed to be minimalist—only the efficiency and pressure ratio plots together with the geometry visualization are included. Hence, for instance, the values of the data points are not initially visible, however, the scatter plots offer a possibility of gaining a high-level

understanding of the data at a first glance. More detail of each individual point is available on an on-demand basis when the gaze cursor hovers over a data marker.

- *Help and documentation*: A single page documentation sheet describing the interaction techniques is provided to the users.

## B. Expressiveness

The expressiveness of the system can be described using the cognitive dimensions of notation framework.<sup>28</sup> This framework provides a vocabulary for analyzing the possibilities and limitations of an interactive system. Below we articulate how the keywords in this vocabulary maps onto the expressiveness of the system.

- *Closeness of mapping*: The visualized geometry is a detail mapping of how would the blade would look like in the real-world.
- *Consistency*: All interactions are consistently designed and all interactive objects have consistent interaction qualities, such selectors.
- *Diffuseness / terseness* The number of used symbols is minimized by only using sphere-like markers with their characteristics (such as color, size and relative placement) to denote all the interactive elements of the visualization.
- *Error-proneness*: Errors are prevented using prevention mechanisms against error states, such as selection of multiple rotation-selectors.
- *Hard mental operations*: Cognitive load and mental demand is kept to a minimum with straight-forward interaction methods and use of comparative visualizations.
- *Hidden dependencies*: As there is a 1:1:1 mapping between the visualization elements all the interdependencies are easily observed by the user since selection of a data point on one of the plots leads to a simultaneous selection of a corresponding data point on the other plot as well and the visualization of the associated geometry.
- *Juxtaposability and Visibility*: All three main parts of the visualization, that is, the efficiency and pressure ratio plots together with the blade's geometry are initially placed next to each other. Furthermore, the plots can be freely rearranged in the space as a group or individually. Moreover, the selection of any data point in a plot is automatically mapped on to the other plot as well and the corresponding geometry is immediately visible. In addition, selected data points are clearly visible through change in color and luminosity.
- *Premature commitment*: The user's workflow with the system is flexible and a user is free to initially inspect the nominal geometry, any or both plots, or to immediately select a data point
- *Progressive evaluation*: Since all the user's actions result in immediate visual feedback (closed-loop interaction) the progress of the visual analytics task can be evaluated by the user at any time.
- *Role-expressiveness*: The individual roles of the three components, that is, the efficiency and pressure-ratio plots and the blade's geometry, are clear from the beginning, especially if the system is used by a domain expert.

## V. Conclusions and future work

The goal in this paper has been to introduce a novel VR aerospace design environment with a particular emphasis on dimension reduction. We have identified the main structures of the design environment and implemented a fully working system for commodity VR headsets. We have also verified the interface from two perspectives: usability and expressiveness.

The two identified main tasks were (i) gaining the overview over the design and (ii) compare the nominal geometry with the one associated with a specific performance parameters. The former was achieved through a mixture or visualization (e.g. blade geometry and scatter plots) and lower-level tasks (e.g. movement and

interaction). The latter was achieved through visualization of the overlaying geometries: semi-transparent nominal blade superimposing over solid shape of a new design.

Moving forwards, our goal is to undertake the complete 3D design of a turbomachinery component in VR. In addition to the sufficient summary plots, our goal is to incorporate characteristics of blade performance at multiple operating points and have reduced order models to estimate the performance characteristics of new designs.

## Acknowledgements

This work was supported by studentships from the Engineering and Physical Sciences Research Council, the Cambridge Commonwealth, European and International Trust and Trinity Hall. The authors are grateful for all the generous support. The authors would also like to thank Timoleon Kipouros for his suggestions.

## References

- <sup>1</sup>Sutherland, I. E., "A Head-mounted Three Dimensional Display," *AFIPS '68 (Fall, part I) Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, 1968, pp. 757–764, DOI:10.1145/1476589.1476686.
- <sup>2</sup>Oculus VR, "Oculus Rift, Oculus Quest, Oculus Go," <https://www.oculus.com>, Last accessed: November 2018.
- <sup>3</sup>Nokuo, T. and Sumii, T., "Head mounted display," Dec. 23 2014, US Patent App. 29/502,182.
- <sup>4</sup>García-Hernández, R. J., Anthes, C., Wiedemann, M., and Kranzlmüller, D., "Perspectives for using virtual reality to extend visual data mining in information visualization," *2016 IEEE Aerospace Conference*, March 2016, pp. 1–11.
- <sup>5</sup>Constantine, P. G., *Active subspaces: Emerging ideas for dimension reduction in parameter studies*, Vol. 2, SIAM, 2015.
- <sup>6</sup>Cook, R. D., *Regression graphics: Ideas for studying regressions through graphics*, Vol. 482, John Wiley & Sons, 2009.
- <sup>7</sup>Hokanson, J. M. and Constantine, P. G., "Data-driven polynomial ridge approximation using variable projection," *SIAM Journal on Scientific Computing*, Vol. 40, No. 3, 2018, pp. A1566–A1589.
- <sup>8</sup>Constantine, P. G., Eftekhari, A., Hokanson, J., and Ward, R. A., "A near-stationary subspace for ridge approximation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 326, 2017, pp. 402–421.
- <sup>9</sup>Seshadri, P., Yuchi, S., and Parks, G. T., "Dimension Reduction via Gaussian Ridge Functions," *arXiv preprint arXiv:1802.00515*, 2018.
- <sup>10</sup>Absil, P.-A., Mahony, R., and Sepulchre, R., *Optimization algorithms on matrix manifolds*, Princeton University Press, 2009.
- <sup>11</sup>Seshadri, P., Shahpar, S., Constantine, P., Parks, G., and Adams, M., "Turbomachinery Active Subspace Performance Maps," *Journal of Turbomachinery*, Vol. 140, No. 4, 2018, pp. DOI: 10.1115/GT2017-64528.
- <sup>12</sup>Pukelsheim, F., *Optimal design of experiments*, Vol. 50, siam, 1993.
- <sup>13</sup>Thomas, J. J. and Cook, K. A., "A Visual Analytics Agenda," *IEEE Comput. Graph. Appl.*, Vol. 26, No. 1, Jan. 2006, pp. 10–13.
- <sup>14</sup>Chandler, T., Cordeil, M., Czauderna, T., Dwyer, T., Glowacki, J., Goncu, C., Klapperstueck, M., Klein, K., Marriott, K., Schreiber, F., and Wilson, E., "Immersive analytics," *2015 Big Data Visual Analytics (BDVA)*, edited by U. Engelke, J. Heinrich, T. Bednarz, K. Klein, and Q. Nguyen, IEEE, Institute of Electrical and Electronics Engineers, United States, 2015.
- <sup>15</sup>Shneiderman, B., "Why Not Make Interfaces Better Than 3D Reality?" *IEEE Comput. Graph. Appl.*, Vol. 23, No. 6, Nov. 2003, pp. 12–15, DOI:10.1109/MCG.2003.1242376.
- <sup>16</sup>Kennedy, R. S., Lane, N. E., Berbaum, K. S., and G.Lilienthal, M., "Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness," *The International Journal of Aviation Psychology*, Vol. 3, No. 3, 1993, pp. 203–220, DOI:10.1207/s15327108ijap0303.3.
- <sup>17</sup>Ruddle, R. A., "The effect of environment characteristics and user interaction on levels of virtual environment sickness," *IEEE Virtual Reality 2004*, 2004, pp. 141–285, DOI:10.1109/VR.2004.1310067.
- <sup>18</sup>Microsoft, "Xbox gaming platform," <https://www.microsoft.com/en-gb/store/b/xbox>, Last accessed: November 2018.
- <sup>19</sup>Stump, G. M., Yukish, M., Simpson, T. W., and O'Hara, J. J., "Trade space exploration of satellite datasets using a design by shopping paradigm," *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, Vol. 6, March 2004, pp. 3885–3895 Vol.6.
- <sup>20</sup>Jeong, S., Chiba, K., and Obayashi, S., "Data Mining for Aerodynamic Design Space," *23rd AIAA Applied Aerodynamics Conference*, American Institute of Aeronautics and Astronautics.
- <sup>21</sup>Shelfelbine, S., Clarkson, J., Farmer, R., and Eason, S., *Good Design Practice for Medical Devices and Equipment - Requirements Capture*, University of Cambridge Engineering Design Centre and University of Cambridge Institute for Manufacturing, 2002.
- <sup>22</sup>Unity3D Game Engine, "Unity VR Samples pack," <https://assetstore.unity.com/packages/essentials/tutorial-projects/vr-samples-51519>, Last accessed: November 2018.
- <sup>23</sup>Oculus VR, "Oculus Utilities for Unity 5," <https://developer.oculus.com>, Last accessed: November 2018.
- <sup>24</sup>Wagner, M., Blumenstein, K., Rind, A., Seidl, M., Schmiedl, G., Lammarsch, T., and Aigner, W., "Native Cross-Platform Visualization: A Proof of Concept Based on the Unity3D Game Engine," *20th International Conference Information Visualisation*, 2016, pp. 39–44, DOI:10.1109/IV.2016.35.
- <sup>25</sup>Ribarsky, W., Bolter, J., Op Den Bosch, A., and Van Teylingen, R., "Visualization and Analysis Using Virtual Reality," *Computer Graphics and Applications, IEEE*, Vol. 14, Feb. 1994, pp. 10–12.

- <sup>26</sup>Nielsen, J., "Usability Inspection Methods," *Conference Companion on Human Factors in Computing Systems*, CHI '94, ACM, New York, NY, USA, 1994, pp. 413–414.
- <sup>27</sup>Nielsen, J., "Enhancing the Explanatory Power of Usability Heuristics," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, ACM, New York, NY, USA, 1994, pp. 152–158.
- <sup>28</sup>Green, T. R. G., "Cognitive dimensions of notations," *People and Computers V*, University Press, 1989, pp. 443–460.
- <sup>29</sup>Green, T. R. G., "The Cognitive Dimension of Viscosity: A Sticky Problem for HCI," *Proceedings of the IFIP TC13 Third International Conference on Human-Computer Interaction*, INTERACT '90, North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 1990, pp. 79–86.