Supporting Iterative Virtual Reality Analytics Design and Evaluation by Systematic Generation of Surrogate Clustered Datasets

Sławomir K. Tadeja* University of Cambridge

Patrick Langdon† University of Cambridge, Edinburgh Napier University

Per Ola Kristensson‡ University of Cambridge

ABSTRACT

Virtual Reality (VR) is a promising technology platform for immersive visual analytics. However, the design space of VR analytics interface design is vast and difficult to explore using traditional A/B comparisons in formal or informal controlled experiments—a fundamental part of an iterative design process. A key factor that complicates such comparisons is the dataset. Exposing participants to the same dataset in all conditions introduces an unavoidable learning effect. On the other hand, using different datasets for all experimental conditions introduces the dataset itself as an uncontrolled variable, which reduces internal validity to an unacceptable degree. In this paper, we propose to rectify this problem by introducing a generative process for synthesizing clustered datasets for VR analytics experiments. This process generates datasets that are distinct while simultaneously allowing systematic comparisons in experiments. A key advantage is that these datasets can then be used in iterative design processes. In a two-part experiment, we show the validity of the generative process and demonstrate how new insights in VR-based visual analytics can be gained using synthetic datasets.

Index Terms: Virtual Reality, Immersive Visual Analytics, Evaluation

1 INTRODUCTION

The ability of Virtual Reality (VR) to immerse the user within a 3D world unlocks the potential of VR amplifying users' ability to understand and gain insights about complex spatial datasets. A key factor to drive such development is the ability to carry out iterative design and evaluation of new interaction techniques and novel visualization representations with users.

However, a current obstacle for iterative design and evaluation of immersive analytics is the fact the datasets involved form an uncontrolled variable in any empirical evaluation. This is because the assessment of how well users can manipulate, understand and gain insights about datasets in a VR environment is intrinsically linked to users' exposure and learning of the datasets. At the same time, it is not realistic to expect a large quantity of natural datasets to be available—and even if they are—to be statistically equivalent in terms of complexity, data density and other factors that may influence user performance.

The central contribution in this paper is a solution to this problem in the form of a generative process for synthesizing datasets that allow designers and researchers to carry out iterative design and evaluation using successive A/B testing with the same set of participants. This is otherwise problematic as the participants are going to learn the intricacies the dataset and thus the (unavoidable) learning of the dataset becomes an uncontrolled variable.

*e-mail: skt40cam.ac.uk

†e-mail: p.langdon@napier.ac.uk

‡e-mail: pok21@cam.ac.uk

There are currently two methods to mitigate this problem. The first method is to use different datasets. However, if these datasets are natural datasets then there is a high risk the characteristics of the individual datasets will inadvertently become a variable that risks explaining a large proportion of the variability between different conditions. The second method is to change to a between-subjects design and recruit new participants for each successive iteration, which is expensive, reduces statistical power (as the variability of each individual is no longer controlled across all conditions and/or iterations), and prevents studying longer-term learning effects of the interaction techniques or VR tools themselves.

In this paper we present a solution, which synthesizes 3D datasets using a robust underlying process. This paper demonstrates that this approach leads to datasets that still allow researchers to detect significant differences for different tasks over a variety of metrics, while at the same time minimizing the influences of the datasets themselves on the results. The process therefore allows repeated measures designs in which, for example, the parameters of an interaction technique can be manipulated, and the effects of such manipulation quantified, without having to be overly concerned of the influence of either participants learning a dataset or the intrinsic differences of individual datasets dominating as explanatory variables.

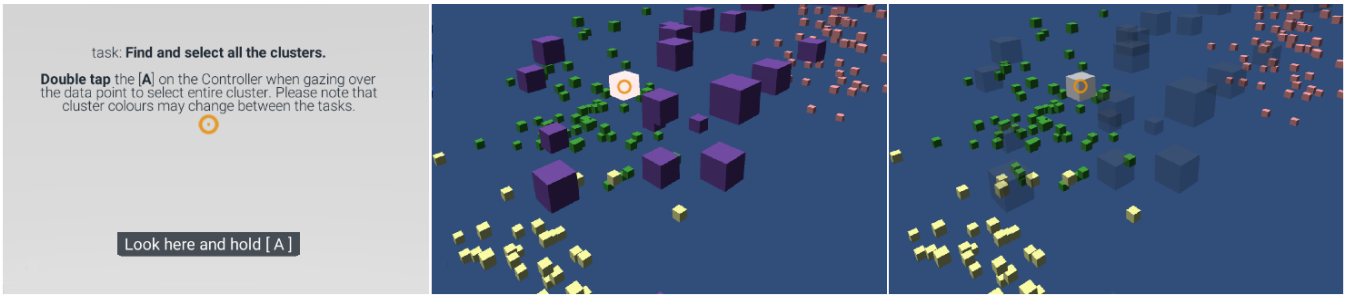
Prior work [2, 22, 56] has introduced a form of synthetic data generation. However, we note that none of these papers actually check whether the dataset is itself an explanatory variable. This introduces a risk of a systematic methodological error in the literature as failing to control for the dataset reduces internal validity. To ensure rigorous findings from controlled experiments it is vital to validate the instrument, in this case, the dataset. This requires developing a specific replicable and transparent mechanism for generating synthetic datasets *and* carrying out specific controlled experiments that ensure the dataset is indeed not an explanatory variable.

We address this gap in the literature by presenting a generative method that allows the designer or researcher to quickly generate a sample of color-coded clustered 3D data points of any desired size (number of clusters and number of data points in clusters). Further, the process allows parameter modifications to facilitate particular needs for an individual study. While this generative process specifically addresses clustering concerns, Bach et al. [2] reflect on the fact that 3D point-clouds can represent a number of different 3D visualizations, including, but not limited to, "*3D-scatterplots, specific spacetime cubes, as well as biomedical images*" [2] and thus exhibit rich applicability. Importantly we validate that the dataset is not an explanatory variable in two experiments specifically designed to check this important property.

In summary, this paper makes the following contributions: (1) We present a generative process for synthesizing surrogate clustered datasets for use in virtual reality analytics design and evaluation. (2) We show the validity of the approach in a two-part evaluation that also demonstrates how new insights in VR analytics can be gained using surrogate clustered datasets.

2 RELATED WORK

Evaluating visualizations in general is acknowledged to be challenging [51]. A systematic overview of these issues can be found in



(a) (b) (c)

Figure 1: Interaction occurs through gaze-tracking and double tapping the [A] button on the Xbox controller. (a) shows a screenshot of the dialogue scene with instructions of *Task 1* as they are presented to the user in VR. (b) shows a gaze-operated orange cross-hair hovering over a data point in a cluster (violet), which automatically highlights the gaze-acquired object, thus informing the user that it can be interacted with. (c) shows the same cluster selected (semitransparent) once the user has double tapped the [A] button on the controller while gazing over an interactive object.

Isenberg et al. [17]. In general, the 3D world data category identified by Shneiderman et al. [41] is a visualization category which *"is still controversial"* [41] and brings additional challenges, echoed by many others [5, 10, 24, 40].

Comparing different three-dimensional information visualization designs is a difficult problem, as pointed out by for example Wiss et al. [54]. Certain designs are not suitable for some datasets and not every interface can support all the fundamental user tasks as listed by Shneiderman [39, 54].

Using artificial data as part of the experimental design is a practice that has been used before (e.g. [2, 13, 22, 56]). However, no prior study has considered the dataset itself as an explanatory variable. In contrast, the objective of this work is to present an easy-to-understand well-motivated non-data driven mechanism that allows generating different clustered datasets while avoiding introducing the dataset as an uncontrolled variable.

Holten et al. [13] used Gaussian distributions to generate point clouds and background noise to evaluate different design variants of parallel coordinate plots shown on a computer monitor. Holten et al. [13] remark that to further generalize their findings they would have to evaluate their visualizations on larger datasets and that the selection of the data type is a potentially limiting factor [13]. The dataset itself was not an explanatory variable.

In general, several prior studies have relied on generative approaches using Gaussian distributions [2, 22, 32, 38]. Sedlmair et al. [38] investigated separation factors for clustered data using real and synthetically generated datasets. The specific synthetically generated datasets used in the research were selected based on the authors' prior experience. Bach et al. [2] and Prouzeau et al. [32] prepared a number of clustered datasets for evaluating immersive interfaces using Gaussian distributions. Bach et al. [2] studied the impact of an augmented reality head-mounted display, a tablet and a desktop computer on users' understanding of 3D visualizations while Prouzeau et al. [32] explored scatterplots in VR. None of these studies controlled for the dataset itself.

Kraus et al. [22] studied the impact of immersion on cluster identification tasks in scatterplot visualizations. Kraus et al. [22] generated datasets by manually creating data points, which were subsequently transformed into clusters using a clustering algorithm. Additional noise was inserted by randomly inserting data points. Kraus et al. [22] did not explicitly control for the impact of the dataset itself.

Further examples of synthesizing datasets with various characteristics can be found in Theodoridis et al. [44], Matejka et al. [28] and Manniono et al. [27]. These datasets were not considered in immersive environments and only Manniono et al. [27] carried out a user study, which did not control for the impact of the dataset itself.

Clusters represent an important class of features [2, 13, 22, 38]. However, they are not the only ones encountered in real-world data [27, 28]. For example, the *scaganostics* approach that describes various methods of interpreting appearance of data on the scatterplots or graphs [45] catalogues and reasons about data traits, such as trends, outliers, smears or other possible anomalies [45, 52, 53].

Some of these features can be easily incorporated to aid a generative process. For example, outliers can be randomly added to a particular cluster or to the dataset as a whole by randomly generating data points outside of clusters or their bounding volumes.

Another approach to data generation was proposed by Yang et al. [56], which used the MINST dataset of handwritten digits [26] as well as the t-SNE dimensionality reduction technique [47] to synthesize point clouds. Filho et al. [9,48] relied in their experiments on a dataset constructed out of real-world voting data. Furthermore, Bach et al. [3] proposed a method of generating random graph data.

3 Approach

The difficulty in evaluating visualization techniques is exacerbated in VR. All problems identified by, for example, Isenberg et al. [17] still apply. These concerns, among others, include validation, verification, and concerns about reproducibility, as well as approaches for rigorous evaluation of effectiveness, efficiency and aesthetics of a given visualization tool.

In addition, VR *in itself* introduces several additional design variables, such as the navigation strategy, means of supporting spatial awareness, and effective direct and indirect manipulation techniques. Other classes of well-known problems include overplotting and occlusion [33]. Yet, to successfully navigate this design space, it will be necessary to empirically compare different interaction solutions. However, this raises a methodological problem in that the dataset itself is an important explanatory variable of user behavior.

This paper proposes a solution to this problem, which is relying on generating datasets based on a model. The generated datasets share similar fundamental characteristics and virtually any number of them can be generated in a matter of seconds. Depending on their needs, designers and researchers can change design parameters, such as the number of clusters or their densities. To decide upon the clusters' placements in 3D space, the process adopts a *random walk* (Brownian motion [8, 42] or Wiener process [50]). This model is widely used in physical chemistry, computational physics, stock market models and crystallography [29, 49]. A *random walk* is by its nature self-similar [6, 12], which is in itself a beneficial characteristic as many datasets also have some sort of fractal structure.

4 Generating Synthetic Clusters

The generative process can be split into four steps: (1) determine the position of the central point of each cluster in 3D space; (2)

Require:
Number of clusters N
Set of K distinguished colors
Ensure:
$N \geq 0$
$K > 0$

```
clusters ← GETEMPTYLIST(void)
D_{n,n} ← GETEMPTYMATRIX(void)
i ← 1
while i ≤ N do
    cluster ← GENERATECLUSTER(N, clusters)
    APPENDCLUSTER(cluster, clusters)
    i ← i + 1
ASSIGNCOLORS(clusters, K)

procedure GENERATECLUSTER(N, clusters)
    brownian ← DRAWBROWNIANTRAIL(2N)
    while True do
        s ← DRAWFROMPOISSON(void)
        if s ≤ N then
                    ▷ Append current trail with s − N samples.
            brownian ← APPENDBROWNIANTRAIL(N + s)
            N = N + s
        P ← brownian[s]
        if P was not drawn before then
            size ← DRAWSIZEFROMPOWER(void)
            cluster ← GENERATE3DPOINTCLOUD(P, size)
            S ← SETBOUNDINGSPHERE(cluster, r, O)
            if S overlap acceptable with others then
                UPDATEDMATRIX(D_{n,n})
                return cluster

procedure GENERATE3DPOINTCLOUD(P, n)
                ▷ Generate cluster of size n using N(σ², μ).
    s ← SETSIGMA(void)
    points ← GETEMPTYLIST(void)
    while i ≤ n do
        p_{x,y,z} ← GETCOORDNORMDIST(σ² = s, μ = P_{x,y,z})
        APPENDPOINT(p, points)
        i ← i + 1
    return points

procedure SETBOUNDINGSPHERE(C, r, O)
            ▷ Set bounding sphere surrounding entire cluster C.
    S ← SETSPHERE(C, r, O)
    return S
```

Algorithm 1: An algorithm describing in pseudocode the surrogate dataset generation method. For clarity, the color-coding function is presented in detail in separate Algorithm 2.

determine the size of each individual cluster; (3) generate samples of data around each clusters' central points; and (4) color-code all individual clusters. Pseudocode for both the generation process and color-coding are shown in Algorithm 1 and Algorithm 2 respectively. The cluster's size and, indirectly, its spatial spread of individual data points influence if a newly generated cluster will be included within the dataset or regenerated again if it overlaps with other clusters above a certain threshold. As the last, non-mandatory step, all points are translated so the bounding sphere spanning along the entire dataset has its center at the axes-origin, which is also where our testing framework described later in this paper initially places the user. We will now describe steps 1–4 in detail.

4.1 Cluster Placement in the 3D Space

We first generate a 3D Brownian motion trail, which results in a set of $2N$ candidate points for possible cluster placements (see `DrawBrownianTrail()` in Algorithm 1). We then draw cluster placements $P_1, P_2, \ldots, P_N$ from this set of candidate points by successively sampling from a Poisson distribution with the rate parameter $\lambda = N$ (see `DrawFromPoisson()` in Algorithm 1). If the candidate P_i was drawn previously, a new cluster is generated about this point. This cluster is either added to the dataset or it is disregarded due to it exceeding the overlapping threshold with pre-existing clusters. In the case when the newly generated cluster is discarded, the candidate point necessitating this new cluster is disregarded as an optimization step as it may provoke repeated overlap. Instead, once the candidate is disregarded, a new candidate P_i is generated in its place.

4.2 Cluster Size

To determine the cluster's size in terms of its total number of data points we sample from a power-law probability distribution $f(x,a) = ax^{a-1}$, with $a = 1^2/_3$, $x \in [0,1]$ (see `DrawSizeFromPower()` in Algorithm 1). The choice of a power-law is motivated by its scale-invariance and by how frequently many physical, biological and artificial systems generate this relationship.

Once a cluster size is determined, we successively generate the desired number of points by sampling from a multivariate normal distribution $\mathcal{N}(\mu = P_i, \sigma^2)$ (see `GetCoordFromNormDist()` in Algorithm 1)

4.3 Maximally Acceptable Overlap

There are many ways to define an overlap. For example, it can be determined by the total number of individual volumetric markers that occupy the same space. Another possibility is to consider a cluster as a chunk of space that is occupied by all its elements. Such a slice can be obtained by encapsulating it within a bounding body. Although not very precise, this method is not only conceptually and computationally simple but also commonly used in computer graphics [25] and it is the approach taken here. This approach does not generate clusters within the spheres but is using the bounding volumes calculated on top of the pre-generated set of 3D volumetric points. Such clusters can take on any shape due to their pseudo-random generation process. Using a bounding body may also be convenient when managing dynamic datasets where the overlap may have to be quickly recalculated during program execution.

We first calculate bounding spheres using Ritter's algorithm. [34] (see `SetBoundingSphere()` in Algorithm 1). We then define an overlap level $o_{i,j}$ of two clusters x_i and x_j as a function of the Euclidean distance between the centers of their respective bounding spheres $w_{i,j} = d(x_i, x_j)$ and the sum of their respective radii:

$$o_{i,j} = 2[1 - \frac{w_{i,j}}{(r_i + r_j)}]. \quad (1)$$

If $w_{i,j} = r_i + r_j$ (that is, $o_{i,j} = 0$), the clusters overlap in a single point, that is. their respective bounding spheres touch each other. If $o_{i,j} < 0$, the spheres are disjoint and if $o_{i,j} > 0$, the clusters overlap.

The values of $o_{i,j}$ are compared with thresholds $t_{i,j}$ to decide if the overlap is acceptable or not. Each $t_{i,j}$ is sampled from a power-law probability distribution with density $a = 1^2/_3$, $x \in [0,1]$ (see `DrawSizeFromPower()` in Algorithm 1). If any of the thresholds are exceeded, the cluster is disregarded. Optionally, the algorithm's decision can be assessed independently either by inspection or by automatically using any of the well known clustering algorithms [35, 55]. Since the data points are generated in clusters this information can be overlaid with clusters determined separately by a clustering algorithm to automatically estimate the percentage of points overlapping with the other clusters.

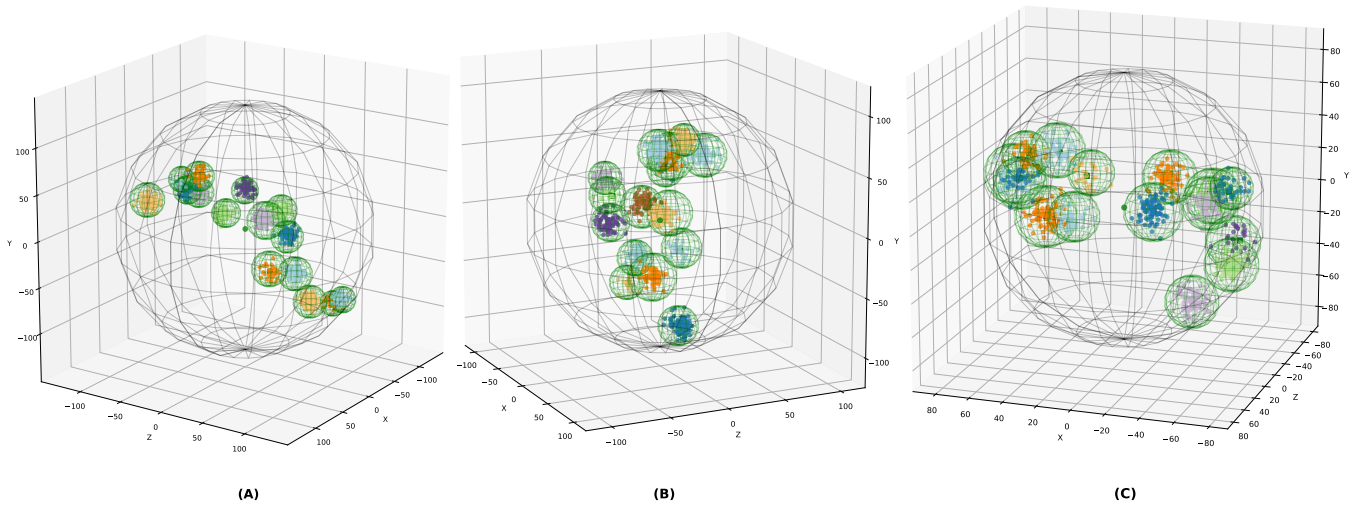


Figure 2: The figure shows the synthetic datasets A, B and C respectively, each containing $N = 15$ clusters. The datasets are presented from different angles, selected to emphasize the spatial distribution of each individual cluster. The figure also shows the bounding spheres of all individual clusters (in green) and the overall bounding sphere (in black) with its center (green dot).

4.4 Cluster Coloring

Coloring clusters is a non-trivial task as factors, such as distance between clusters, their spatial distribution and the limited number of perceptually distinct colors, have to be considered. In addition, a further design constraint is the desire to reduce the risk of overlapping clusters with the same color. We color code the clusters according to a heuristic method that is independent from the generation method, and incorporates all of the above factors (see `AssignColors()` in Algorithm 2). The algorithm starts by assigning a color Ω_0 to the first cluster located at x_0. Next, in each step, by using a greedy optimization procedure, we choose the color for one cluster located at x_i. First, for all colors $c_1, c_2, \ldots, c_K$ we calculate quality coefficients $A(c_k)$ (see `GetCoeff()` in Algorithm 2) defined as

$$A(c_k) = \sum_{j=1}^{i-1} w_{i,j}^{-2}[K - g(\Omega_j, c_k)], \tag{2}$$

where the weights $w_{n,i}$ are inverse and squared to ensure that the further away the clusters are, the smaller impact this distance will have on their color assignment, K is the number of available colors and g is a *minimal color distance* between the clusters defined as:

$$g(c_i, c_j) = \min(|c_i - c_j|, M - |c_i - c_j|). \tag{3}$$

In our case, c is an integer index that can be easily mapped to the respective color from a set of $K = 12$ different colors downloaded from the *ColorBrewer2.0* [7]. The coefficient M is either equal to $M = K$ when K is even, or $M = K + 1$ otherwise. Subsequently, we pick $\text{argmin}(A(c_i))$ as the desired color. The whole operation has $O(N^2K)$ time complexity. For a reasonably small number of clusters, the weights w and color distances g are calculated when needed. However, the method can be optimized by pre-computing all of $w_{i,j}$ and $g(c_i, c_j)$. The symmetric matrix containing all pre-computed distances $w_{i,j}$ between all pairs of clusters would look as follows:

$$\begin{pmatrix} 0 & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & 0 & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & 0 \end{pmatrix}. \tag{4}$$

5 Immersive Analytics Environment

We built a VR analytics environment to visualize the generated clusters and to be able to interact with them using the Unity game engine and additional freely available assets [1, 46].

We used the Oculus Rift [1] supported by a high-end laptop computer. Hardware included the GeForce GTX 1080, Intel Core i7 6700K 4GHz and 16GB DDR4 RAM working under Windows 10 Pro 64bit.

An example of a visualization of a scatter plot can be seen in Fig. 1, which features four distinctive clusters, with the violet clusters presented before (b) and after selection (c). Each cluster is constructed from a set of points visualized as volumetric objects.

Clusters are visualized by grouping points together in terms of their respective attributes, such as colors and placements in 3D space. However, the final assessment on whether a point belongs to a particular cluster or not is left to the user, and only once the point is selected the whole cluster instantly becomes translucent.

Movement in any direction in 3D space is achieved using an Xbox controller. Movement occurs with respect to the user's gaze and current position within the VR environment. To move within the horizontal plane the user tilts the joystick. To move along the vertical plane the user presses one of the controller triggers (right to go up $\Uparrow$ and left to go down $\Downarrow$). These actions can occur simultaneously, thus allowing any movement trajectory.

When the user hovers over a data point (see Fig. 1) with the cross-hair, the system automatically highlights the data point by instantly changing its color to white. Once a user is gazing over any of the data points in a cluster, the cluster can be selected by double tapping the [A] button. This makes all the elements belonging to the selected cluster translucent, as suggested by Rekimoto et al. [20] (see Fig. 1). Choosing this selection method helps to decrease occlusion, as pointed out by Shneiderman [40].

The environment partially supports six of Shneiderman's et al. [41] seven basic information visualization tasks (*details-on-demand task* is not implemented). Among the remaining six, the *overview task* and the *zoom task* are supported by the user's movement capabilities in 3D space. The *relate task* is part of the visualization itself through a mixture of cluster elements' color-coding and their placement in space. To a limited extent, the filter and history tasks are supported by keeping previously selected clusters translucent.

Table 1: The 2nd and 4th columns list indexes of selected colors and overlapping clusters respectively, whereas the 3rd one denotes their sizes.

Dataset	Selected colors	Sizes *(the largest and smallest are in bold)*	Overlapping
A	{0,6,0,7,7,1,8,2,9,2,8,1,7,0,6}	{66,**99**,69,32,30,62,94,94,64,92,88,42,44,**14**,98}	{0,3,12,13}
B	{0,6,0,7,1,8,2,9,6,0,7,0,1,6,11}	{87,78,85,85,18,62,**15**,73,**99**,73,62,32,95,20,74}	{3,4,9,10,13}
C	{0,6,7,1,0,7,0,7,1,8,1,2,8,9,2}	{66,30,50,33,75,95,45,71,**97**,75,39,77,86,**27**,80}	{10,9,2,3}
D	{0,6,7,0,7,6,0,11,1,0,5,1,8,2,9}	{37,92,46,22,58,19,22,73,**96**,71,94,94,65,84,**18**}	{9,13,4,12}
E	{0,6,0,7,1,8,6,2,11,4,9,10,11,4,11}	{73,43,21,55,10,86,65,15,**4**,92,76,62,74,**97**,91}	{3,4,9,11,12,13,14}
F	{0,6,11,4,10,4,3,10,9,3,10,4,11,10,4}	{86,72,88,84,54,92,55,83,63,58,30,66,54,**96**,**24**}	{9,10,4,6}

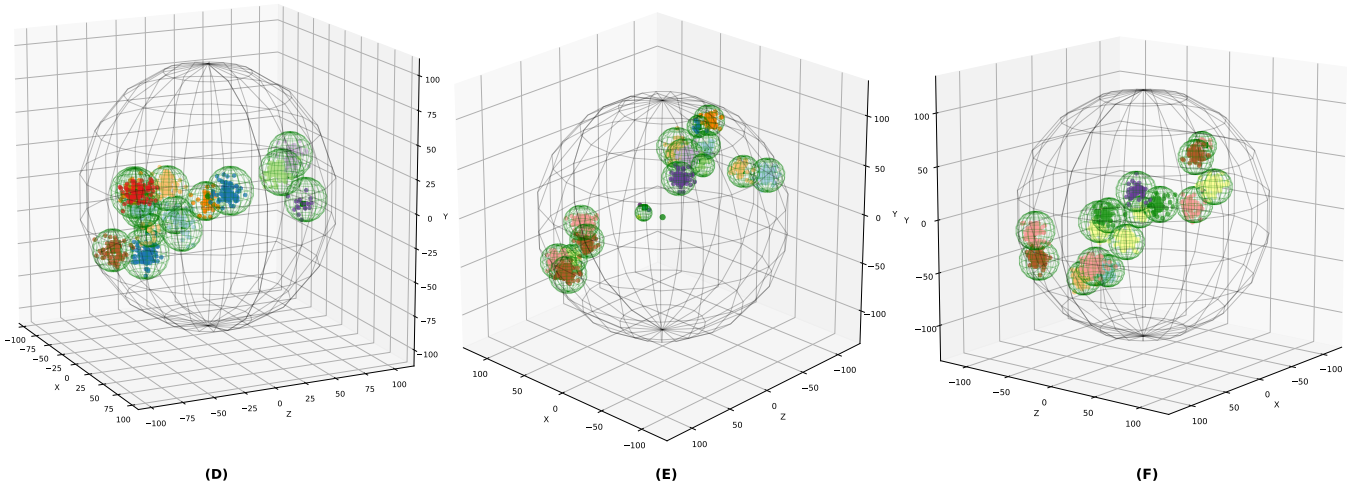


Figure 3: Surrogate datasets *D*, *E* and *F*. The main characteristics of the datasets are the same as for *A*, *B* and *C*.

6 Validation

We validated the generative process in two experiments with identical designs. The validation had two objectives. First, to investigate whether the generative process resulted in suitable clusters for analytics design and evaluation. Second, to assess how much of the variability that can be explained by treating the dataset as an independent variable in a typical analytics experimental task.

Ideally the dataset variable can be treated as a controlled variable across several A/B studies, which then allows iterative A/B testing and fine-tuning of VR analytics interaction techniques without worrying about 1) the learning effect of using a specific dataset in every single condition; or 2) an undue noise effect arising from using datasets with widely different characteristics in every condition.

We find that, overall, the dataset does not result in significant differences at a weak significance level of $\alpha = 0.05$, which suggests designers and researchers can use the generative process described in this paper to keep generating new datasets for every within-subject condition and thereby eliminate the learning effect of using a specific dataset in every condition.

As we wanted to validate the generation method, we investigated the strength of the null hypothesis that there will be no noticeable significant performance differences between users for both the task type and dataset. While accepting the null hypothesis does not form conclusive evidence that it is true, failing to reject the null hypothesis at a significance level of $\alpha = 0.05$ (which is a weak significance level and therefore conservative in this case), indicates that for practical purposes of standard A/B testing, the dataset itself is unlikely to be the dominant explanatory variable of any experimental results. In addition, we also report effect sizes which help quantify how much of the variability is explained by treating the dataset as an independent variable.

To examine the sensitivity of the participant groups and datasets themselves, we split the study into two independent experiments carried out as within-subjects designs that were later analyzed together.

6.1 Participants

For the first experiment we recruited 18 participants using opportunity-sampling. All of them were pre-screened with a short version of the Ishihara's [18] color deficiency test before commencing the experiment. Participants were within the age of 22–47, with the majority being under 30 years of age. Four of them were female and 14 were male.

For the second experiment we recruited 21 participants using opportunity-sampling. Half were female and half were male. The youngest participant was 22 years old and only three participants were above the age of 30.

6.2 Procedure

We used the generative process presented in this paper to generate six datasets for two identically designed experiments. Fig. 2 and Fig. 3 show *Matplotlib* [15] visualizations of the datasets *A*, *B*, *C* and *D*, *E*, *F* used in both experiments, respectively. The main characteristics of these datasets are listed in Tab. 1.

The two experiments were carried out in an identical fashion. Both were split into three series of three tasks and each series had a balanced order of the three tasks and used its own generated dataset (see Tab. 2).

After each cycle we asked participants to fill out a set of questionnaires, the NASA Task Load Index (NASA TLX) [11], the Simulator Sickness Questionnaire (SSQ) by Bouchard et al. [4] and originally developed by Kennedy et al. [21], and an English version of the Igroup Presence Questionnaire (IPQ) [16] administered through a web-based interface. We decided not to ask participants to fill out the forms after each individual task as this would significantly extend the time required to finish the experiment, which in turn would have an effect on a participant's levels of fatigue and overall performance. An analysis of these forms revealed slight increase in nausea and/or oculo-motor effects in the majority of the participants (13 in the first and 10 in the second study, respectively). However, in no case did

Require:
List of *clusters*
Set of K indexed *colors*
Ensure:
$K > 0$

procedure ASSIGNCOLORS(*clusters*, K)
  $N = \text{LEN}(clusters)$
  if $N \leq K$ **then** ▷ If we have less clusters than colors.
    COLORCLUSTERS(K, *clusters*)
    EXIT(*SUCCESS*)
  $picked \leftarrow \text{GETEMPTYLIST}(N)$
  $picked[0] = colors[0]$ ▷ Set color of first cluster.
  $i \leftarrow 0$
  while $i < N$ **do**
    $A \leftarrow \text{GETEMPTYLIST}(K)$
    $j \leftarrow 0$
    while $j < i$ **do**
      if $w_{i,j} \neq 0$ **then**
        $A[i] = \text{GETCOEFF}(colors, i, j, K, A)$
      $j \leftarrow j + 1$
    $picked[i] = colors[\text{MAPTOINDEX}(A[i])]$
    $i \leftarrow i + 1$
  COLORCLUSTERS(*picked*, *clusters*)

procedure GETCOEFF(*colors*, i, j, K, A)
  $k \leftarrow 0$
  while $k < K$ **do**
    $d = |colors[i] - colors[j]|$
    if K is even **then**
      $A[k] = w_{i,j}^{-2}(K - \text{MIN}(d, K - d))$
    else
      $A[k] = w_{i,j}^{-2}(K - \text{MIN}(d, K + 1 - d))$
    $k \leftarrow k + 1$
  return ARGMIN(A)

Algorithm 2: The color coding procedure. The pseudocode assumes that all vectors starts with an index 0.

a participant decide to stop the experiment or to directly report any moderate or severe symptoms.

Participants were orally briefed before the experiment and task-specific instructions and a written repetition of the oral brief were provided through the HMD before a participant begun a new task, as shown in Fig. 1. Participants were instructed to carry out the tasks as quickly and as accurately as possible.

6.3 Tasks

Participants were instructed to perform three tasks for all three datasets in each experiment. *T1*: Find and select all the clusters; *T2*: Find the smallest cluster and the largest cluster in terms of the total number of data points. Pick them in either order. There may be more clusters of the same size; *T3*: Find and select all the overlapping clusters. The tasks required the users to visually inspect the individual clusters and understand their spatial (*T1* and *T3*) or quantitative (*T2*) relation to other clusters, rather than any detailed knowledge of their specific parameters, such as the exact number of data points in a cluster.

There are many tasks that can be considered here (e.g. [37,48]). However, as remarked by Wijk [51], such low-level tasks as *T1–T3* are often a subject of consideration in the information visualization community. We chose tasks that are widely used and representative at large. Further work can study other tasks, if necessary, as method development is an organic process that requires an active research community willing to investigate different effects, variants and tweaks.

Table 2: The order of the tasks ($T1 - T3$) and accompanying datasets ($A - F$) was repeated for each consecutive group of six participants. We chose this ordering as balancing of the datasets was deemed more important as we were primarily interested in the differences between the datasets rather than the particular tasks.

Tasks execution order			Datasets order
A(T1,T2,T3)	B(T2,T3,T1)	C(T3,T1,T2)	ABC
A(T2,T3,T1)	C(T3,T1,T2)	B(T1,T2,T3)	ACB
B(T3,T1,T2)	A(T1,T2,T3)	C(T2,T3,T1)	BAC
C(T1,T2,T3)	A(T2,T3,T1)	B(T3,T1,T2)	CAB
B(T2,T3,T1)	C(T3,T1,T2)	A(T1,T2,T3)	BCA
C(T3,T1,T2)	B(T1,T2,T3)	A(T2,T3,T1)	CBA

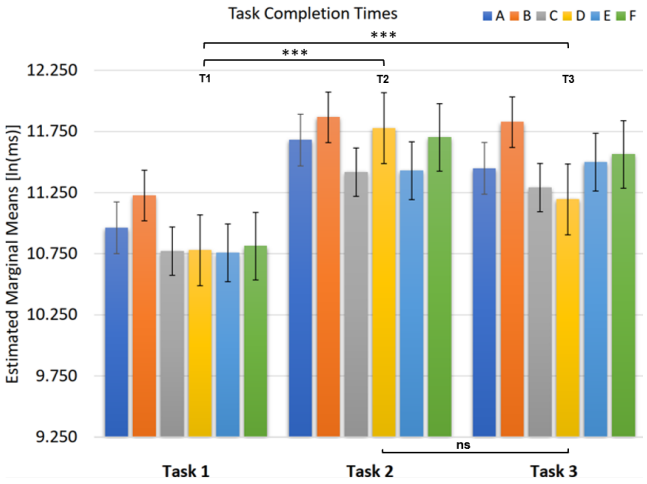


Figure 4: Task completion times with standard error. As expected, task *T1* took the least amount of time to complete across all the datasets. The colors indicate the datasets (see Table 2). The statistically significant differences ($p \leq 0.001$) between the tasks are marked with three asterisks. "ns" denotes no difference ($p \geq 0.05$).

The system automatically marked the tasks as completed and displayed a completion message to the participant as soon as all the clusters were selected. Participant were not given information of any errors in their responses.

6.4 Results

A General Linear Model three-way mixed repeated measures analysis of variance (ANOVA) was used to analyze task completion times, rotation and travelled distance measurements. Time durations were log-transformed prior to analysis. Error counts were analyzed using a Generalized Linear Model using a log-Poisson kernel. All statistical analyses were carried out at an initial significance level of $\alpha = 0.05$, which was adjusted for multiple comparisons with Holm-Bonferroni correction, where applicable.

The NASA TLX, IPQ and SSQ scores were analyzed using Friedman's test. However, no significant differences were detected so for brevity we omit these results in the analysis.

6.4.1 Task Completion Times

The first timestamp of each task was taken the moment the task scene was loaded. Each participant's selection, including subsequent repeated selections, were also timestamped and recorded in sequence. The difference between the first timestamp and the last timestamp that fulfilled the task's requirements was calculated as task completion time. Fig. 4 summarizes task completion times.

No statistically significant effects were detected between the two experiments ($F_{1,34} = .487, \eta_p^2 = .012, p = .527$) with respect to task completion times. All the possible interaction combinations

Table 3: Total counts of the two kinds of errors: "repeated selection" and the "wrong identification" for each task (*T1*, *T2* and *T3*) and dataset (*A*, *B*, *C* and *D*, *E*, *F*) separated by backslash (\).

	A	*B*	*C*	*D*	*E*	*F*
T1	76\76	72\72	55\55	33\33	50\50	31\31
T2	40\85	22\95	13\78	19\66	17\69	7\73
T3	20\53	31\93	35\85	5\73	18\27	64\71

were also insignificant. However, there were statistically significant differences between tasks ($F_{2,68} = 36.267, \eta_p^2 = .516, p \leq .001$). Pairwise comparisons of the tasks revealed, as expected, that there were also significant differences ($p \leq .001$) between *T1* and both *T2* and *T3*. The difference between *T2* and *T3* was not significant ($p = .205$).

In other words, the differences between the datasets were insignificant, but the tasks did indeed result in significant differences. This demonstrates that the process of synthesizing datasets does result in comparable datasets that can still be used to detect significant differences across tasks.

6.4.2 Errors

For all tasks, an error will occur if the participant repetitively selects any of the previously selected clusters (see Fig. 3). In the case of task *T2*, an additional type of error happens if the participant selects neither the smallest, nor the largest, cluster. The same happens if a non-overlapping cluster is selected in task *T3*. To reduce the risk that the analysis model is overdispersed, we discarded outliers three standard deviations away from the mean.

An analysis using a general linearized model with a log-Poisson kernel revealed that only tasks ($\chi^2 = 9.454, df = 2, p = .009$) are significant predictors of errors whereas the datasets are not. The interaction was insignificant. Linearly independent pairwise comparisons of estimated marginal means revealed a statistically significant difference between *T1* and *T2* ($df = 1, p = .016$). This outcome was to be expected, as tasks *T1* and *T2* required selection of the largest ($N = 15$) and the smallest number ($N = 2$) of clusters respectively (see Tab. 1).

The same analysis was repeated for task-specific errors labeled as "incorrect identification" (see Tab. 3). Specifically, this included repeated selections in task *T1*, failing to select the largest or the smallest cluster in task *T2*, or selecting a non-overlapping cluster in task *T3*. The differences were not statistically significant for the datasets. However, both tasks ($\chi^2 = 11.678, df = 2, p = .003$) and the interaction ($\chi^2 = 9.588, df = 4, p = .048$) of the factors were statistically significant. Further analysis of linearly independent pairwise comparisons of estimated marginal means revealed statistically significant differences between task *T1* and both task *T2* ($df = 1, p = .002$) and *T3* ($df = 1, p = .018$) respectively. There were also statistically significant differences in the interaction observed between $B(T1)$ & $B(T3)$ ($df = 1, p = .019$), $B(T3)$ & $C(T1)$ ($df = 1, p \leq .001$) and $C(T1)$ & $C(T3)$ ($df = 1, p = .036$).

For the second study, the tasks were significant predictors of errors ($\chi^2 = 14.269, df = 2, p \leq .001$) whereas the datasets were not. An interaction analysis also revealed no statistically significant results. Pairwise comparisons of estimated marginal means revealed statistically significant differences between task *T1* and task *T2* ($df = 1, p = .011$) and *T3* ($df = 1, p = .004$). In addition, pairwise comparisons also revealed statistically significant results between the interaction of dataset/task $D(T1)$ & $D(T3)$ ($df = 1, p = .009$) and $D(T1)$ & $F(T2)$ ($df = 1, p = .037$). As before, only the tasks ($\chi^2 = 8.836, df = 2, p = .012$) were significant predictors of errors, whereas the interaction of tasks and the datasets were not. As in the first study, pairwise comparisons of estimated marginal means revealed a statistically significant difference between task *T1* and task *T2* ($df = 1, p = .008$). Pairwise comparisons of interactions between dataset/task revealed that $D(T3)$ & $E(T3)$ were statistically significant ($df = 1, p = .038$).

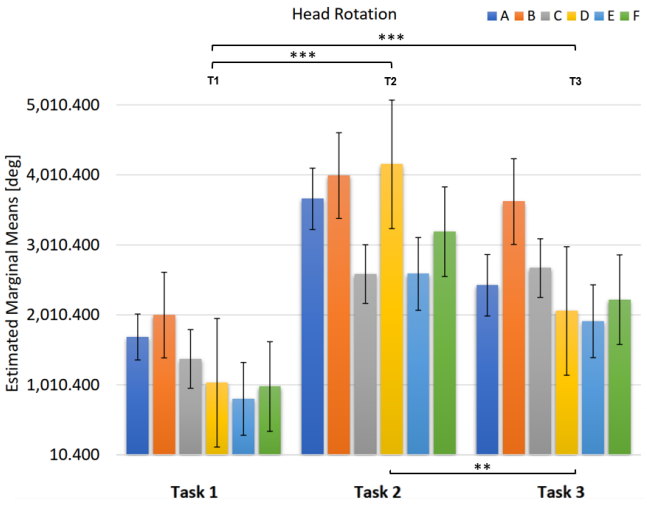


Figure 5: Cumulative head rotation data in degrees with the T-bars denoting the standard error. The statistically significant differences between the tasks are marked with asterisks (two and three asterisks for $p \leq 0.01$ and $p \leq 0.001$ respectively).

6.4.3 Head Rotation and Traveled Distance

To calculate an approximation of the user's head rotation we captured the forward vector extending from the middle of camera's frustum together with changes in the camera position. The angles between successive records of the forward vectors were summed to provide an estimation of total change in the user's head rotation. The analysis of this measurement yielded statistically significant differences between tasks ($F_{2,68} = 34.312, \eta_p^2 = .502, p \leq .001$) and of the interaction between the experiments and the datasets ($F_{2,68} = 3.714, \eta_p^2 = .098, p = .029$), which appears to be driven by datasets *B* and *C* (see Fig. 5). No statistically significant effects were observed between the two experiments with respect to head rotation. As expected, pairwise comparisons of the estimated marginal means revealed significant differences between all three tasks, *T1* and *T2* ($p \leq .001$), *T1* and *T3* ($p \leq .001$) and *T2* and *T3* ($p = .004$).

The Euclidean distance between consecutive points were used to compute the length of the user's trajectory in the VR environment; the results are shown in Fig. 6. Statistically significant differences were found between tasks ($F_{2,68} = 19.447, \eta_p^2 = .364, p \leq .001$) and between datasets ($F_{2,68} = 4.857, \eta_p^2 = .125, p = .011$). However, there was no significant difference between the experiments. As expected, a pairwise comparison revealed statistically significant differences between *T1* and *T2* ($p \leq .001$) and *T1* and *T3* ($p \leq .001$). Again, the difference between the datasets appear to be driven by datasets *B* and *C* (see Fig. 6).

We conjecture the differences between the datasets is due to the variation in vertical organization of the clusters in relation to the natural gaze patterns of the users. In other words, users are more likely to scan the scene laterally than looking up and down. This effect is likely a good indicator of how to place data in the VR environment and to study solutions for mitigating the problem when this is not possible, such as navigation or spatial awareness aids that can assist the user in fully exploiting the VR visualization.

7 Discussion

This paper has presented a generative process for synthesizing surrogate clustered datasets that allow iterative design and evaluation of interaction techniques and visualization representations in VR

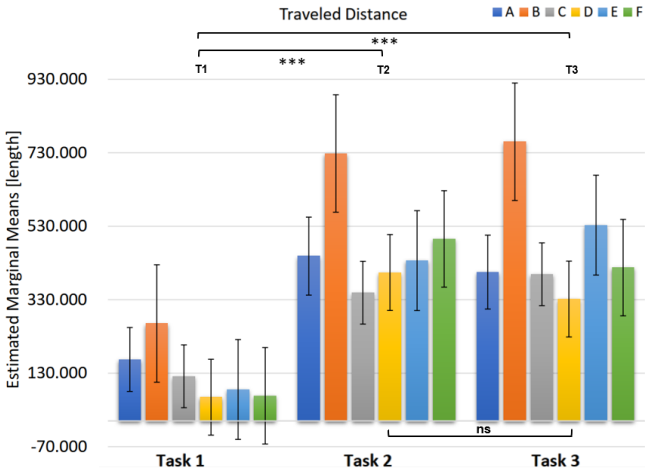


Figure 6: Total traveled distance in Unity3D's native units with the T-bars denoting the standard error. As expected, the least amount of movement was observed in task *T1* across all the datasets. The statistically significant differences ($p \leq 0.001$) between the tasks are marked with three asterisks. "ns" denotes no difference ($p \geq 0.05$).

that avoids having to account for the datasets themselves forming an explanatory variable.

The validation provides evidence that the generative process for synthesizing clustered datasets did indeed result in datasets that can be used in experiments without the datasets themselves being the dominant contributor of the variability between conditions. Overall, the tasks revealed significant differences, which was to be expected, while the influences of the datasets as an independent variable in the experimental design did not result in significant differences—even though we used statistical significance tests with as high statistical power as possible and a weak significance level of $\alpha = 0.05$ to err on the conservative side. Further, effect sizes have been reported throughout, which help quantify the proportion of the explained variability. In combination, this supports the hypothesis that the generative process described in this paper is suitable for iterative A/B testing-based design and experimentation.

The generative process can greatly simplify such testing by eliminating the learning effect of using a single dataset by instead introducing participants to series of generated synthetic datasets that are unlikely to be dominant causal variables explaining differences in performance across conditions. The differences observed between the tasks were to be expected, specifically as tasks *T2* and *T3* required a deeper understanding of the relation between the individual clusters, especially compared to task *T1*. Further, all of the tasks required participants to gain some understanding of the spatial distribution of the clusters, which, as the participant is fully immersed in the data and retains only a first-person perspective, can only become apparent to the user over time.

Finally, another factor that may impact the overall differences in task completion times is the user's selection strategy. The most obvious tactic, which would yield the fastest execution times while simultaneously causing, on average, the most errors, would be to select all clusters as quickly as possible under any and all experimental conditions. However, the recorded data indicates that no participant used this approach.

7.1 Limitations and Future Work

This paper used the frequentist statistical analysis paradigm [30] when analyzing the results. However, the generative method addresses a general experimental design problem and the solution is therefore readily applicable to other types of statistical inference, such as Bayesian inference [19].

We have here focused on generating clustered datasets, as cluster analysis is a fundamental visual analytics task, which we feel is a particularly promising research application for VR analytics. However, a potential fruitful avenue of future work is to explore systematic generation processes for different types of datasets suitable for VR analytics, such as spatiotemporal traces [23] or graphs [3]. Such approaches must be carefully motivated and evaluated to ensure relevance and validity and are therefore out-of-scope for this paper.

In line with prior work, this paper studied clustered data in a clean virtual environment. It is an open research question how well the approach advocated in this paper would apply to more complex virtual environments, which may encompass non-abstract data (e.g. terrain, urban, or nature) synthesized using a variety of procedural generation methods [14, 36].

Another compelling research question is whether it is possible to generate datasets equivalent on higher-level metrics such as, for example, presence, task load, cyber-sickness or aesthetics. Such investigations can be carried out using both clustered datasets, as in this paper, and for other types of data, such as spatiotemporal data or graphs. Related, it would be worthwhile to explore more complex tasks, such as the users having to learn the intricacies of the virtual environment, finding objects within it or evading targets. We hope this paper will stimulate fruitful research in these directions.

Finally, the generative method has the potential to be generalized to synthesize multidimensional data. This involves replacing the 3D trail with an n-dimensional Brownian motion trail to determine the clusters' placements within the n-dimensional space. As a consequence, the color-coding scheme is then no longer viable, as the data is now unsuitable for visualization as a 3D point cloud. One alternative is parallel coordinates [13, 43]. Further, a high-dimensional clustering algorithm [55] needs to be applied to determine the range of overlaps between the individual clusters. We leave further exploration of such an extension as future work.

8 Conclusions

A barrier towards increased virtual reality analytics design and evaluation is the difficulty to iteratively evaluate new interaction techniques and visualization representations as the datasets themselves become an explanatory variable.

This paper has presented a solution that simplifies iterative A/B testing of cluster-based VR analytics tasks. By robustly generating synthetic clustered datasets that give rise to similar user behavior but are still perceived as different by participants, it is possible to A/B test successive interactions of interaction techniques and VR analytics tools without being overly concerned of the learning effect of using a single natural dataset, or the heterogeneity induced by using several natural datasets, whose underpinning properties are uncontrollable. The process can be further enhanced if the algorithms are tailored to facilitate individual needs.

This paper has demonstrated that artificially generating datasets is a viable method and a two-part evaluation has shown the validity of the approach. However, we urge caution when implementing this method as part of an empirical study. Similar to how repeated measures designs should always be checked for asymmetrical skill-transfer effects [31], as a precaution we recommend analyzing the dataset as an independent variable to reaffirm that the dataset was not a significant contributor in a particular study.

Currently, controlling for the dataset has been a major stumbling block for widespread in-depth empirical evaluation in VR analytics. We hope this work will stimulate increased activity in this area.

Open Science

The Python source code used to generate the datasets and datasets *A*–*F* are available as supplemental materials.

REFERENCES

[1] Oculus Rift. https://oculus.com, [Online]: Nov. 2018.

[2] B. Bach, R. Sicat, J. Beyer, M. Cordeil, and H. Pfister. The hologram in my hand: How effective is interactive exploration of 3d visualizations in immersive tangible augmented reality? *IEEE Trans. Vis. Comput. Graphics*, 24:457–467, 2018.

[3] B. Bach, A. Spritzer, E. Lutton, and J.-D. Fekete. Interactive Random Graph Generation with Evolutionary Algorithms. In W. Didimo and M. Patrignani, eds., *Graph Drawing 2012*, vol. 7704 of *Lecture Notes in Computer Science*, pp. 541–552. Springer, Berlin, Germany, Sept. 2012. doi: 10.1007/978-3-642-36763-2_48

[4] S. Bouchard, G. Robillard, and P. Renaud. Revising the factor structure of the Simulator Sickness Questionnaire. *Annual Review of CyberTherapy and Telemedicine (ARCTT)*, 5:117–122, 2007.

[5] R. Brath. 3D InfoVis is here to stay: Deal with it. In *2014 IEEE VIS International Workshop on 3DVis (3DVis)*, pp. 25–31, Nov. 2014. doi: 10.1109/3DVis.2014.7160096

[6] R. D. Cohen. Self similarity in Brownian motion and other ergodic phenomena. *J. Chem. Educ.*, 63(11):933, Nov. 1986. doi: 10.1021/ed063p933

[7] B. Cynthia, M. Harrower, B. Sheesley, A. Woodruff, and D. Heyman. Colorbrewer2.0. http://colorbrewer2.org/, [Online]: Nov. 2018.

[8] A. Einstein. Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Annalen der Physik*, 322(8):549–560, 1905. doi: 10.1002/andp.19053220806

[9] J. A. W. Filho, M. F. Rey, C. S. Freitas, and L. Nedel. Immersive visualization of abstract information: An evaluation on dimensionally-reduced data scatterplots. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 483–490. IEEE Computer Society, Los Alamitos, CA, USA, mar 2018. doi: 10.1109/VR.2018.8447558

[10] R. J. García-Hernández, C. Anthes, M. Wiedemann, and D. Kranzlmüller. Perspectives for using virtual reality to extend visual data mining in information visualization. In *2016 IEEE Aerosp. Conference*, Mar. 2016. doi: 10.1109/AERO.2016.7500608

[11] S. G. Hart and L. E. Staveland. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Advances in Psychology*, vol. 52, pp. 139–183. Elsevier, 1988. doi: 10.1016/S0166-4115(08)62386-9

[12] S. Havlin and D. Ben-Avraham. Diffusion in disordered media. *Advances in Physics*, 36(6):695–798, Jan. 1987. doi: 10.1080/00018738700101072

[13] D. Holten and J. J. V. Wijk. Evaluation of Cluster Identification Performance for Different PCP Variants. *Computer Graphics Forum*, 29(3):793–802, 2010. doi: 10.1111/j.1467-8659.2009.01666.x

[14] R. Huijser, J. Dobbe, W. F. Bronsvoort, and R. Bidarra. Procedural natural systems for game level design. In *2010 Brazilian Symposium on Games and Digital Entertainment*, pp. 189–198, 2010. doi: 10.1109/SBGAMES.2010.31

[15] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55

[16] Igroup. Survey on experiences in virtual worlds. http://www.igroup.org, [Online]: Nov. 2018.

[17] T. Isenberg, P. Isenberg, J. Chen, M. Sedlmair, and T. Möller. A Systematic Review on the Practice of Evaluating Visualization. *IEEE Trans. Vis. Comput. Graphics*, 19(12), Dec. 2013. doi: 10.1109/TVCG.2013.126

[18] S. Ishihara. *Ishihara's Tests for Colour Deficiency*. Kanehara Trading Inc, Tokyo, Japan, 38 plates edition ed., 2017.

[19] H. Jeffreys. *The Theory of Probability*. Oxford University Press, 1939.

[20] M. G. Jun Rekimoto. The Information Cube: Using Transparency in 3D Information VisualizationInformation Cube. In *Proceedings of the WITS'93*, pp. 125–132, 1993.

[21] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness. *The International Journal of Aviation Psychology*, 3(3):203–220, July 1993. doi: 10.1207/s15327108ijap0303_3

[22] M. Kraus, N. Weiler, D. Oelke, J. Kehrer, D. Keim, and J. Fuchs. The impact of immersion on cluster identification tasks. *IEEE Transactions on Visualization and Computer Graphics*, 26:525–535, 2020.

[23] P. O. Kristensson, N. Dahläck, D. Anundi, M. Björnstad, H. Gillberg, J. Haraldsson, I. Mårtensson, M. Nordvall, and J. Ståhl. An evaluation of space time cube representation of spatiotemporal patterns. *IEEE Transactions on Visualization and Computer Graphics*, 15(4):696–702, 2009. doi: 10.1109/TVCG.2008.194

[24] M. Kyritsis, S. R. Gulliver, S. Morar, and R. Stevens. Issues and Benefits of Using 3d Interfaces: Visual and Verbal Tasks. In *Proceedings of the MEDES '13*, pp. 241–245. ACM, NY, USA, 2013. doi: 10.1145/2536146.2536166

[25] T. Larsson. Fast and tight fitting bounding spheres. In *SIGRAD 2008. The Annual SIGRAD Conference Special Theme: Interaction; November 27-28; 2008 Stockholm; Sweden*, pp. 27–30. Linköping University Electronic Press, 2008.

[26] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010.

[27] M. Mannino and A. Abouzied. Is this real? generating synthetic data that looks real. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, p. 549–561. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3332165.3347866

[28] J. Matejka and G. Fitzmaurice. Same stats, different graphs: Generating datasets with varied appearance and identical statistics through simulated annealing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, p. 1290–1294. Association for Computing Machinery, New York, NY, USA, 2017. doi: 10.1145/3025453.3025912

[29] J. P. Meyburg and D. Diesing. Teaching the Growth, Ripening, and Agglomeration of Nanostructures in Computer Experiments. *J. Chem. Educ.*, 94(9):1225–1231, Sept. 2017. doi: 10.1021/acs.jchemed.6b01008

[30] J. Neyman. Outline of a theory of statistical estimation based on the classical theory of probability. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 236(767):333–380, 1937.

[31] E. Poulton and P. Freeman. Unwanted asymmetrical transfer effects with balanced experimental designs. *Psych. Bulletin*, 66(1):1, 1966.

[32] A. Prouzeau, M. Cordeil, C. Robin, B. Ens, B. Thomas, and T. Dwyer. Scaptics and highlight-planes: immersive interaction techniques for finding occluded features in 3d scatterplots. In A. Cox and V. Kostakos, eds., *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery (ACM), United States of America, 2019. International Conference on Human Factors in Computing Systems 2019, CHI 2019 ; Conference date: 04-05-2019 Through 09-05-2019. doi: 10.1145/3290605.3300555

[33] A. Prouzeau, M. Cordeil, C. Robin, B. Ens, B. H. Thomas, and T. Dwyer. Scaptics and Highlight-Planes: Immersive Interaction Techniques for Finding Occluded Features in 3D Scatterplots. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1—12. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300555

[34] J. Ritter. An Efficient Bounding Sphere. In *Graphics Gems*, pp. 301–303. Elsevier, 1990. doi: 10.1016/B978-0-08-050753-8.50063-2

[35] L. Rokach and O. Maimon. Clustering Methods. In O. Maimon and L. Rokach, eds., *Data Mining and Knowledge Discovery Handbook*, pp. 321–352. Springer US, Boston, MA, 2005. doi: 10.1007/0-387-25465-X_15

[36] T. J. Rose and A. G. Bakaoukas. Algorithms and approaches for procedural terrain generation - a brief review of current techniques. In *2016 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, pp. 1–2, 2016. doi: 10.1109/VS-GAMES.2016.7590336

[37] A. Sarikaya and M. Gleicher. Scatterplots: Tasks, data, and designs. *IEEE transactions on visualization and computer graphics*, 24(1):402—412, January 2018. doi: 10.1109/tvcg.2017.2744184

[38] M. Sedlmair, A. Tatu, T. Munzner, and M. Tory. A taxonomy of visual cluster separation factors. *Computer Graphics Forum*, 31(3pt4):1335–1344, 2012. doi: 10.1111/j.1467-8659.2012.03125.x

[39] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on*

Visual Languages, pp. 336–343, Sept. 1996. doi: 10.1109/VL.1996.545307

[40] B. Shneiderman. Why not make interfaces better than 3D reality? *IEEE Computer Graphics and Applications*, 23(6):12–15, 2003. doi: 10.1109/MCG.2003.1242376

[41] B. Shneiderman and C. Plaisant. *Designing the User Interface: Strategies for Effective HCI*. Addison-Wesley, USA, 5 ed., 2009.

[42] M. v. Smoluchowski. Zur kinetischen Theorie der Brownschen Molekularbewegung und der Suspensionen. *Annalen der Physik*, 326(14):756–780, 1906. doi: 10.1002/andp.19063261405

[43] S. K. Tadeja, T. Kipouros, and P. O. Kristensson. IPCP: Immersive Parallel Coordinates Plots with Engineering Design Processes. In *Proceedings of AIAA SciTech Forum and Exposition*, Jan. 2020. doi: 10.2514/6.2020-0324

[44] Y. Theodoridis, J. R. O. Silva, and M. A. Nascimento. On the generation of spatiotemporal datasets. In *Proceedings of the 6th International Symposium on Advances in Spatial Databases*, SSD '99, p. 147–164. Springer-Verlag, Berlin, Heidelberg, 1999.

[45] J. W. Tukey and P. A. Tukey. Computer Graphics and Exploratory Data Analysis: An Introduction. In *Proc. the Sixth Annual Conference and Exposition: Computer Graphics '85, Vol. III, Technical Sessions*, pp. 773–785. Nat. Computer Graphics Association, 1985.

[46] Unity. Unity VR Samples pack. `https://unity.com/`, [Online]: Nov. 2018.

[47] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[48] J. A. Wagner Filho, C. Freitas, and L. Nedel. Virtualdesk: A comfortable and efficient immersive information visualization approach. *Computer Graphics Forum*, 37(3):415–426, 2018. doi: 10.1111/cgf.13430

[49] G. H. Weiss. *Aspects and Applications of the Random Walk*. Elsevier Science Ltd, Amsterdam The Netherlands ; New York, Mar. 1994.

[50] N. Wiener. Differential-Space. *Journal of Mathematics and Physics*, 2(1-4):131–174, 1923. doi: 10.1002/sapm192321131

[51] J. J. v. Wijk. Evaluation: A Challenge for Visual Analytics. *IEEE Computer Society*, 46(7):56–60, July 2013. doi: 10.1109/MC.2013.151

[52] L. Wilkinson, A. Anand, and R. Grossman. Graph-Theoretic Scagnostics. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05. IEEE Computer Society, Washington, DC, USA, 2005. doi: 10.1109/INFOVIS.2005.14

[53] L. Wilkinson, A. Anand, and R. Grossman. High-Dimensional Visual Analytics: Interactive Exploration Guided by Pairwise Views of Point Distributions. *IEEE Trans. Vis. Comput. Graphics*, 12(6):1363–1372, Nov. 2006. doi: 10.1109/TVCG.2006.94

[54] U. Wiss, D. Carr, and H. Jonsson. Evaluating three-dimensional information visualization designs: a case study of three designs. In *Proceedings. 1998 IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics (Cat. No.98TB100246)*, pp. 137–144, July 1998. doi: 10.1109/IV.1998.694211

[55] D. Xu and Y. Tian. A Comprehensive Survey of Clustering Algorithms. *Ann. Dat. Sci.*, 2(2):165–193, June 2015. doi: 10.1007/s40745-015-0040-1

[56] Y. Yang, M. Cordeil, J. Beyer, T. Dwyer, K. Marriott, and H. Pfister. Embodied navigation in immersive abstract data visualization: Is overview+detail or zooming better for 3d scatterplots?, 2020. doi: 10.1109/tvcg.2020.3030427