GRay: Ray Casting for Visualization and Interactive Data Exploration of Gaussian Mixture Models

Kai Lawonn, Monique Meuschke, Pepe Eulzer, Matthias Mitterreiter, Joachim Giesen, Tobias Günther

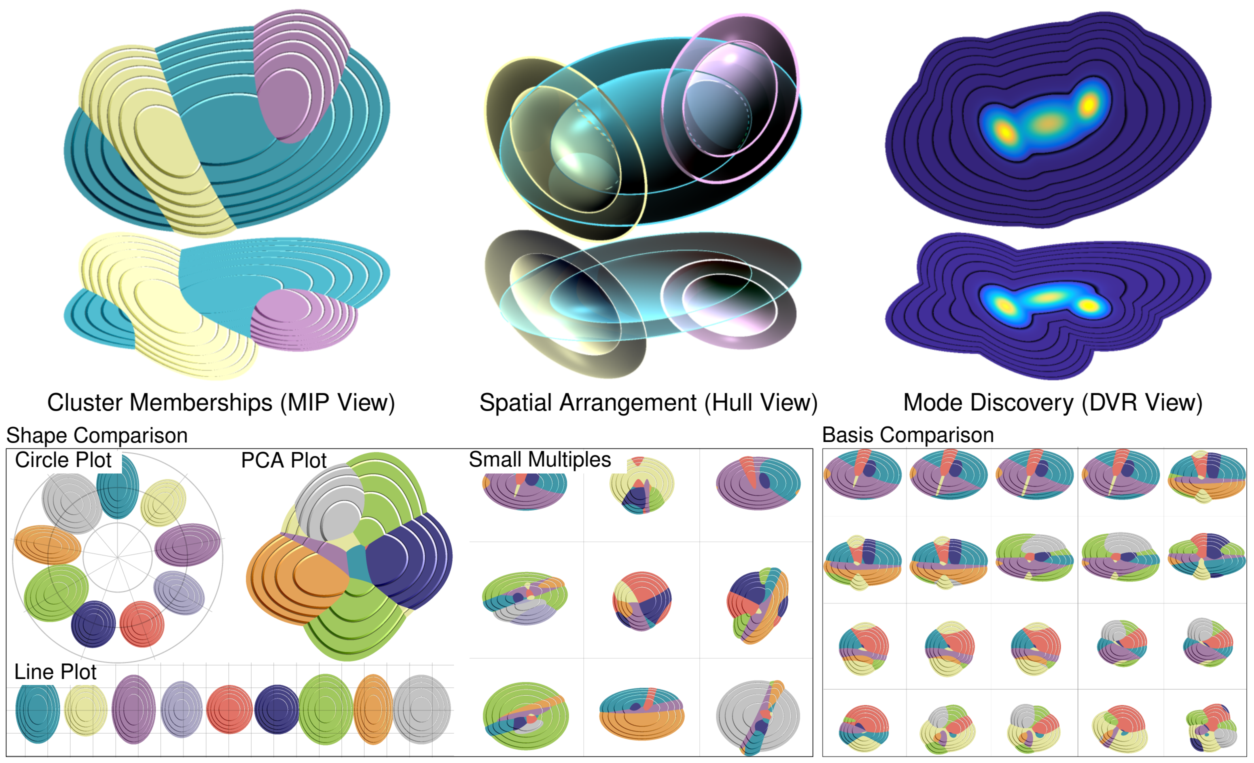

Cluster Memberships (MIP View)
Spatial Arrangement (Hull View)
Mode Discovery (DVR View)
Shape Comparison
Circle Plot
PCA Plot
Small Multiples
Line Plot
Basis Comparison

Fig. 1. We present an exploration framework for *Gaussian Mixture Models* that combines different visualization techniques. The top row shows raycasting-based visualizations that reveal cluster memberships (MIP view), spatial arrangement of Gaussians (hull view), and new modes (DVR view). To manage complexity, the bottom row contains overview visualizations that allow for the comparison of shapes (circle plot, line plot, PCA plot, and small multiples) and different choices of basis vectors (small multiples).

Abstract— The Gaussian mixture model (GMM) describes the distribution of random variables from several different populations. GMMs have widespread applications in probability theory, statistics, machine learning for unsupervised cluster analysis and topic modeling, as well as in deep learning pipelines. So far, few efforts have been made to explore the underlying point distribution in combination with the GMMs, in particular when the data becomes high-dimensional and when the GMMs are composed of many Gaussians. We present an analysis tool comprising various GPU-based visualization techniques to explore such complex GMMs. To facilitate the exploration of high-dimensional data, we provide a novel navigation system to analyze the underlying data. Instead of projecting the data to 2D, we utilize interactive 3D views to better support users in understanding the spatial arrangements of the Gaussian distributions. The interactive system is composed of two parts: (1) raycasting-based views that visualize cluster memberships, spatial arrangements, and support the discovery of new modes. (2) overview visualizations that enable the comparison of Gaussians with each other, as well as small multiples of different choices of basis vectors. Users are supported in their exploration with customization tools and smooth camera navigations. Our tool was developed and assessed by five domain experts, and its usefulness was evaluated with 23 participants. To demonstrate the effectiveness, we identify interesting features in several data sets.

Index Terms—Scientific visualization, Gaussian mixture models, ray casting, volume visualization

1 Introduction

High-dimensional data play an important role in many technical, biomedical, and sociological applications. To bring order, experts are interested in finding subgroups with similar characteristics. For this purpose, clustering techniques are frequently used to group the data entries in a meaningful way. However, a unique assignment of a data point to exactly one cluster does not always make sense. It may also be possible that it is in between two or even more clusters.

• *K. Lawonn, P. Eulzer, M. Mitterreiter, J. Giesen are with Friedrich Schiller University of Jena. E-mail: first name.last name@uni-jena.de.*
• *M. Meuschke is with Otto von Guericke University of Magdeburg. E-mail: meuschke@isg.cs.uni-magdeburg.de*
• *T. Günther is with Friedrich-Alexander-Universität Erlangen-Nürnberg. E-mail: tobias.guenther@fau.de*

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

For this reason, soft clustering techniques are of interest because they allow multiple clusters to be assigned to one data point. One of the early models, but still in the focus of current research, is the Gaussian mixture model (GMM), which is nowadays used in various fields, from acoustic signals anomaly detection [36, 56], emotion recognition [42], image segmentation [11, 52], image matching [25], background subtraction [55], visualization of volume data [24], to voice conversion [46], and more. As a result, GMMs continue to be an active research topic in machine learning, theory, and application communities.

Despite their popularity, few efforts have been made to explore and understand them visually. GMMs consist of multiple high-dimensional Gaussian distributions oriented in arbitrary directions. Although the data points and their orientation already reveal interesting structures, most representations show only the main axes by using simple visualizations, such as isolines in 2D. With growing number of dimensions and growing number of Gaussians, occlusions occur, which hinder an effective exploration of spatial arrangements. Therefore, showing the distribution of the data points with their Gaussians in a customizable subspace is necessary to reveal more complex relationships.

In this paper, we aim to fill this gap by presenting an understandable framework for visualizing GMMs. The framework consists of two parts: (1) Inspired by volume rendering, we apply three interactive 3D visualization techniques to GMMs, as the third dimension reduces the amount of overlaps and occlusions: maximum intensity projection, hull/isosurface rendering, and direct volume rendering. Each visualization technique is advantageous for a specific analysis tasks, such as locating the most-likely cluster membership, exploring the spatial arrangement, and discovering new modes. (2) In addition, overview visualizations are utilized to compare the shapes of Gaussians, and to compare different basis axis choices, which are used for projection to 3D. Furthermore, we present several interaction options to explore the high-dimensional data, including customizations of the basis vectors, as well as automated camera navigations to help users form a mental model of the spatial arrangements. The framework was developed and guided by experts in the field of machine learning, visualization, and human-computer interaction. The design was iteratively refined, and the final program was evaluated with 23 participants, who are familiar with these fields. In summary, we make the following contributions:

- We provide the, to the best of our knowledge, first 3D visualization system that supports users in the exploration of high-dimensional GMMs containing many Gaussians.
- We provide multiple helpful tools to assist users in the exploration, including overview visualizations of shape and basis choices, basis customizations, and smooth camera animations.
- We present an efficient GPU implementation that achieves real-time performance.

Section 2 gives an overview of recent work in the field of high-dimensional data visualization, followed by a formal introduction to Gaussian mixture models. Section 3 formally describes the problem by means of a data abstraction and a task abstraction, leading to a list of analysis tasks. Section 4 discusses the design rationale, provides an overview of the system and subsequently introduces all visualization components. Section 5 elaborates on the efficient GPU implementation. Section 6 contains an evaluation comprising feedback from five experts, questionnaire-based evaluations with 23 study participants, as well as performance measurements. Section 7 reports findings made with the tool in various data sets. Section 8 discusses the findings of the evaluation and the application. Section 9 concludes the paper and outlines potential avenues for future work.

2 Related Work

The visualization of high-dimensional data has been a core topic of visualization research for several decades. For an overview of visualizations that focus on high-dimensional data, we recommend the works by Chab [9], Liu et al. [22], He et al. [16], and Nobre et al. [30]. In addition, Leisch [20] presents an overview about commonly used 2D plot-based visualizations to explore hierarchical clustering results. According to Liu et al., the visualization of multidimensional data can be organized in three stages: *Data Transformation*, *Visual Mapping*, and *View Transformation*. First, data are pre-processed by, e.g., dimensionality reduction, or subspace clustering. Relevant for our work is the second stage, namely, Visual Mapping.

Classical visual mapping approaches are scatterplot matrices and parallel coordinates [5, 17], which are both coordinate axis based, i.e., they use orthogonal coordinate projections. For high-dimensional data, it can be especially challenging to identify interesting projections. This problem has been addressed by Seo and Shneiderman [41] who provide a rank-by-feature framework that suggests relevant dimensions to the user. The user can start from the suggestions and examine the data further using scatterplots. A related approach was presented by Sips et al. [43] who introduce the concept of class consistency. Tatu et al. [44] also employ an automatic analysis to identify structures in high-dimensional data from which suggestions for further explorations can be derived. Bertini et al. [6] compiled an overview of quality metrics for guiding high-dimensional data exploration.

To avoid cluttered parallel coordinates plots of large or high-dimensional data, Artero et al. [3] filter the data first to provide a clearer view. Fanea et al. [13] introduced a different approach by developing parallel glyphs for high-dimensional data. Another approach yielding visualizations for parallel coordinates plots was presented by Yuan et al. [53]. Here, synthetic data points are integrated into a parallel coordinates plot that encodes additional derived information. Ferdosi and Roerdink [14] employed smart rules for the reordering of dimensions in parallel coordinates plots and scatterplot matrices.

For enabling further exploration of multidimensional data, Elmqvist et al. [12] introduced the scatterplot matrix together with navigation facilities. Im et al. [18] introduced an extension of scatterplot matrices, called generalized plot matrices, which consist of scatterplots, heatmaps, and bar charts. Another approach to explore multidimensional data, called Flexible Linked Axes, was introduced by Claessen and Wijk [10]. Here, various visualizations are linked together to facilitate exploration. A technique to represent multidimensional data by landscapes was introduced by Oesterling et al. [31–33]. Depending on the density of the data points, a terrain is constructed such that the height function corresponds to the local point density.

So far, we have discussed works that only make use of the natural coordinates and not non-linear functions thereof. However, the natural coordinates are often not the most effective representation of the data, especially, when the data is close to a (non-)linear lower-dimensional manifold. In this case, visual embeddings like multidimensional scaling [19], Isomap [45], or t-SNE [48] can be more effective in revealing interesting structures like clusters. Liu et al. [23] used subspaces and 2D linear projections. They implemented parallel coordinates plots enhanced by brushing and linking facilities that enable the exploration of clusters in the data. Besides identifying features for the user to investigate further, a few approaches try to cluster data directly in parallel coordinates plots [34, 35, 40, 51, 54]. Rheingans and DesJardins [38] used various projection techniques based on self-organizing maps for visualizing predictive model quality. Migut and Worring [28] employed a framework with mosaic plots and scatterplots for classifying models for risk assessment. Garg et al. [15] visualized model learning based on data segmentation and classification results, for which they used a graph structure with labeled clusters and edges connecting them.

Using popular languages like Python or Matlab, GMMs can be visualized, but these visualizations are limited to either density plots or isoline plots in the natural coordinates of the underlying data. Visualizing GMMs on certain sub-spaces, e.g., hyperplanes, requires additional effort. We present, to the best of our knowledge, the first framework for analyzing and visually exploring high-dimensional GMMs.

3 Problem Statement

With this paper, we introduce new visual encodings for the analysis of high-dimensional Gaussian mixture models (GMMs) used for cluster analysis. First, we briefly introduce GMMs and describe the resulting problems and visualization goals that our approach should fulfill.

Introduction to Gaussian Mixture Models. To set the notation, we briefly introduce Gaussians and Gaussian mixture models. The multivariate *Gaussian* distribution in $\mathbb{R}^k, k \in \mathbb{N}$ is defined as:

$$\mathcal{N}(\mathbf{x};\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^k \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right), \quad (1)$$

where $\Sigma \in \mathbb{R}^{k\times k}$ is a positive-definite covariance matrix and $\mathbf{x}, \mu \in \mathbb{R}^k$. The vector μ is called the mean vector. A GMM is parametrized by Gaussian distributions $\mathcal{N}(\mu_i,\Sigma_i)$ (later referred to as *Gaussians*) and corresponding positive weights $\phi_i \in \mathbb{R}_{>0}, i \in \{1,2,\ldots,N\}, N \in \mathbb{N}$:

$$\mathcal{G}(\mathbf{x};\phi_i,\mu_i,\Sigma_i) = \sum_{i=1}^{N} \phi_i \mathcal{N}_i(\mathbf{x};\mu_i,\Sigma_i). \quad (2)$$

Σ_i denotes the positive-definite covariance matrix for all i [7]. Note, that positive-definite matrices have eigenvalues greater than zero. We denote the greatest eigenvalue as λ_1, i.e., $\lambda_1 \geq \lambda_i, i \in \{1,2,\ldots,k\}$.

Projection in View Box. Visualizing a single Gaussian, cf. Eq. (1), or even a GMM, cf. Eq. (2), requires the projection of a high-dimensional space to two-dimensional screen space. This can be achieved by considering a three-dimensional subspace, which can then be projected on the screen space. Therefore, we need to define such a subspace by three basis vectors in $\mathbb{R}^k$. These basis vectors $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3 \in \mathbb{R}^k$ are mutually orthogonal and of unit-length. These vectors define the matrix $\mathbf{B} \in \mathbb{R}^{k\times 3}$:

$$\mathbf{B} = (\mathbf{b}_1\ \mathbf{b}_2\ \mathbf{b}_3) \quad (3)$$

which defines the axes of the view-box, which allows us to look into the high-dimensional space. The view-box defines a three-dimensional subspace such that each of its points $\mathbf{p} = (p_1,p_2,p_3)^T \in \mathbb{R}^3$ relates to a point in $\mathbb{R}^k$ by $\mathbf{Bp} \in \mathbb{R}^k$. With this, we can transform Eq. (1) and restrict it to the view-box:

$$\mathcal{N}(\mathbf{Bp};\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^k \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{Bp}-\mu)^T\Sigma^{-1}(\mathbf{Bp}-\mu)\right). \quad (4)$$

The resulting distribution is still a Gaussian, but it is not normalized.

Data Abstraction and Problem Statement. Following Munzner [29], we begin the visualization design with a data abstraction. The original high-dimensional point cloud is formed via k quantitative attributes that constitute the coordinates of a k-dimensional space. To approximate those points, a Gaussian mixture model with N components is given, containing the quantitative k-dimensional mean coordinates μ_i of the Gaussians, their $k \times k$ covariance matrices Σ_i, and their quantitative scalar-valued weight ϕ_i. An eigenanalysis of the covariance matrix reveals the most relevant basis vectors, ranked by the eigenvalues. In the following, we refer to the number of significant eigenvalues as m, which comprise 90% of the data: $m = \max\{m' \mid \sum_{i=1}^{m'} \lambda_i < 0.9\sum_{i=1}^{k} \lambda_i\}$ with eigenvalues λ_i being sorted in descending order. For a GMM, we later report m for each Gaussian. Thus, the three relevant parameters that characterize the scalability of a visualization are the domain dimensions k, the manifold dimensions m, and the number of Gaussians N. When those parameters are small, standard methods are available:

- k **is small**. When the number of attributes is small, a scatterplot matrix along with Gaussian level sets might suffice. Each scatterplot requires about 100×100 pixels [29], limiting k by the display resolution. Being inherently two-dimensional, complex relationships among multiple principle axes are not visible, and for multiple Gaussians overlaps are common, see Fig. 2 (left).
- m **is small**. In this case, the high-dimensional data can be reduced via PCA to a lower dimensional subspace, which can then be visualized with scatterplot matrices and Gaussian level sets. When the number of dimensions increases, occlusions will hinder the view in the PCA projection.

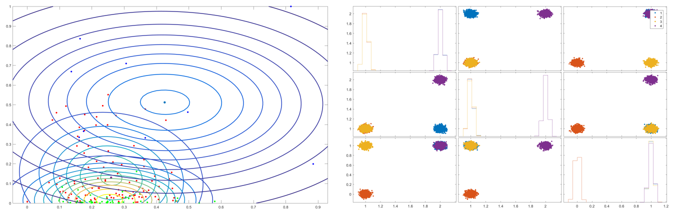


Fig. 2. Limitations of existing methods. Left: Scatterplot and Gaussian level sets are inherently two-dimensional and exhibit overlapping Gaussians. Right: Scatterplot matrix where no 2D projection along principal axes exists that is able to show all four clusters without occlusions.

- N **is small**. With only few Gaussians, the risk of occlusion in 2D visualizations is low. When the number of Gaussians increases, occlusion becomes a significant problem. To demonstrate this problem, Fig. 2 (right) shows a data set with four Gaussians, in which there is always one Gaussian occluded in each view.

The aforementioned approaches do not scale well for larger problems. Thus, in this paper, we present the first visualization approach that enables the exploration of Gaussian mixtures for high k, m, and N.

Task Abstraction and Goals. To define the tasks that the visualization system should address, we perform a task abstraction [29]. Based on contextual inquiries with three ML experts, who commonly utilize and analyze Gaussian mixture models, we identified the following challenges, which are confirmed as being relevant by the experts:

T1 Cluster Membership. Identify the most relevant cluster in the domain, since GMMs support the projection of further unseen data points for which the cluster assignment is important.

T2 Relative Spatial Arrangement. Judge proximity, overlap and spatial arrangement of multiple Gaussians since spatial similarities and groups are relevant for GMM parameter selection.

T3 Discover New Modes. Locate modes that are generated by a particular choice of principal axes, since those derived features allow experts to judge the utility of an axes combination.

T4 Manual Basis Definition. Explore not only permutations of principal axes, but also linear combinations of them, since principal axes alone are not creating an optimal separation.

T5 Point Distribution. Inspect how well Gaussians approximate the underlying data distribution and how closely data points are to the mean, since Gaussians need to fit points well enough to be a surrogate for further analysis, which needs manual inspection.

T6 Attribution. For a given data point, find out which attributes are responsible for a certain cluster membership, since this reveals why a data point belongs to a specific cluster.

T7 Compare Shapes. Compare the shapes of multiple Gaussians, since different sizes and elongations reveal structural information about the underlying point clouds.

T8 Compare Bases. Compare the separability and spatial arrangements of Gaussians for different choices of eigenvector bases, since more than three principal components carry significant information, and the different options need to be explored.

In the following chapter, we present the visualization system, designed to address the challenges.

4 Visualization of GMMs

When aiming for scalability, a key observation is that common visualization techniques that map into a 2D view, suffer from occlusions, as demonstrated in Fig. 2. When staying in 2D, one approach would be the utilization of distortion-oriented techniques [47]. Drawbacks of distortion-oriented approaches are the inability to preserve distances, the limited support of object constancy (i.e., the inability to form a mental model of spatial relations during the non-linear motion of data points when moving the focus region), as well as the potential unawareness of distortions taking place [29]. Instead, we decided for an interactive 3D visualization, which allows the user to rotate the view, effectively exploring three principal axes at a time, instead of only two. Despite the increased mental load of navigating 3D space, a 3D system is able to communicate spatial relations, i.e., we can better understand the relative positioning of multiple Gaussians in 3D. When designing a 3D system, the following design questions arise:

- *How do we decide which principal axis to map spatially?*
  Not only is it possible to perform a PCA on the entire point cloud or on the Gaussians mean coordinates, which provides a *global* view. Each individual Gaussian also has principal axes, from which three axes can be chosen to form a *local* view concentrating on one particular Gaussian. Our system provides an overview of possible basis choices using *small multiples*, users may choose a desired basis combination for inspection, and a custom basis can be formed interactively via linear combinations.

- *How do we support spatial understanding?*
  Choosing a three-dimensional view raises visualization challenges. To support the spatial scene understanding, shadow projections are added, as well as shading effects. Interactive camera navigations are vital to resolve occlusions and to form a mental model of the spatial relations via motion parallax, see the video for examples. When transitioning between different basis vector choices, a camera animation is provided to retain object constancy, i.e., points can be visually tracked to the new view.

- *How do we aggregate information from 3D into a 2D image?*
  We introduce three aggregation operators to fulfill the different tasks. The methods are inspired from well-established raycasting-based volume visualization techniques, including maximum intensity projection (MIP), surface visualization of transparent Gaussian ellipsoids utilizing edge enhancement, and cumulative direct volume rendering to reveal new modes.

In the following, we present an overview of the visualization system. Afterwards, the components are explained individually in more detail.

4.1 Overview

An overview of the system is shown in Fig. 1. We divide the individual views into two groups:

Raycasting Views The raycasting-based views visualize Gaussians and derived attributes in 3D, including cluster memberships using maximum intensity projection (task **T1**), spatial arrangement of Gaussians using hull surfaces (task **T2**), as well as new modes discovered through cumulative direct volume rendering (task **T3**). Each view requires the specification of a basis, which can be interactively composed (task **T4**), underlying point distributions can be shown to judge how well the Gaussians fit the data (task **T5**), and for each point, the reason for a cluster assignment can be explored (task **T6**).

Shape and Basis Overviews To better compare the shape of Gaussians, overviews and alternative spatial arrangements are introduced that provide positional encodings with aligned axes (task **T7**). To further support the exploration of different basis choices (task **T8**), overviews of different options are provided using small multiples, and a smooth camera animation is implemented to transition smoothly from one basis choice to another.

4.2 Raycasting Views

We propose three raycasting-based visualizations that address different tasks. All three views map the given data attributes onto the position channels, since those are most accurate in terms of visual perception [29]. Derived attributes will be mapped to color, as explained in the following. For this section, the camera position $\mathbf{p} \in \mathbb{R}^3$ is given and the ray direction $\mathbf{r} \in \mathbb{R}^3$ is defined by the orthonormal basis of the view-box $\mathbf{B}$, cf. Eq. (3). A ray is cast through each pixel in the image.

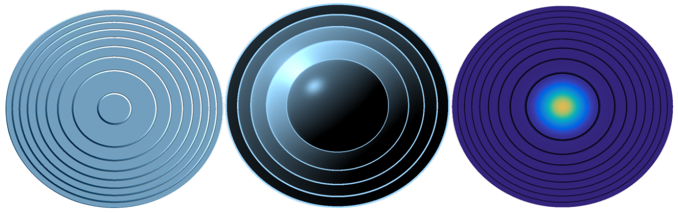


Fig. 3. Raycasting-based visualizations: MIP, (five) Hulls, and DVR.

4.2.1 Maximum Intensity Projection (MIP)

Task **T1** requires the user to identify for any point in the domain, which cluster it belongs to, i.e., which Gaussian has the highest weighted value according to Eq. (1). One option would be to employ 3D slicing, which introduces a user parameter: the slice position, which is crucial for the success of finding interesting structures. Instead, we utilize a maximum intensity projection to locate the highest Gaussian value along a view ray, making sure that maxima cannot be missed. For a given Gaussian $\mathcal{N}_i$ with weight ϕ_i, the location of the maximum value along the ray at $\tau_i \in \mathbb{R}$ fulfills the necessary condition $\frac{\partial}{\partial \tau}\phi_i \mathcal{N}_i(\mathbf{B}(\mathbf{p}+\tau_i\mathbf{r});\mu_i,\Sigma_i) = 0$ which has a closed-form solution:

$$\frac{\partial}{\partial \tau}\mathcal{N}_i = -(\mathbf{r}^T\mathbf{B}^T\Sigma_i^{-1}\mathbf{B}\mathbf{p} + \tau_i \mathbf{r}^T\mathbf{B}^T\Sigma_i^{-1}\mathbf{B}\mathbf{r} - \mathbf{r}^T\mathbf{B}^T\Sigma_i^{-1}\mu_i)\phi_i\mathcal{N}_i$$
$$\Rightarrow \tau_i = \frac{-\mathbf{r}^T\mathbf{B}^T\Sigma_i^{-1}\mathbf{B}\mathbf{p} + \mathbf{r}^T\mathbf{B}^T\Sigma_i^{-1}\mu_i}{\mathbf{r}^T\mathbf{B}^T\Sigma_i^{-1}\mathbf{B}\mathbf{r}} = \frac{\mathbf{r}^T\mathbf{B}^T\Sigma_i^{-1}(\mu_i - \mathbf{B}\mathbf{p})}{\mathbf{r}^T\mathbf{B}^T\Sigma_i^{-1}\mathbf{B}\mathbf{r}} \quad (5)$$

The sufficient condition $\frac{\partial^2}{\partial \tau^2}\mathcal{N} < 0$ holds, confirming a maximum. We compare the maximum values of each Gaussian with its corresponding weight, cf. Eq. (2), along this ray using the calculated τ_i value, to obtain the Gaussian with the largest value: $\hat{\imath} = \arg\max_i\ \sigma_i \mathcal{N}_i(\mathbf{B}(\mathbf{p}+\tau_i\mathbf{r}))$. Cluster membership $\hat{\imath}$ is a categorical attribute, which is mapped to the identity channel hue. This yields a Voronoi-like visualization of the GMM that subdivides space into one region for each Gaussian such that the Gaussian has the highest probability in this region among all components. Within each region, we indicate the direction of the corresponding mean vector by the visual cue of stairs. The values of the Gaussians are discretized, and stairs are drawn for the discrete level values. The number of stairs is set by the user. We shaded the stairs to provide a clue for the position of the mean vector, see Fig. 3 (left).

4.2.2 Hull Intersection

Task **T2** requires the user to obtain a spatial understanding of the relative positioning among Gaussians. To this end, we visualize the Gaussians in 3D by displaying enclosing hull isosurfaces with a silhouette-enhancing transparency mapping. For a given Gaussian $\mathcal{N}$, we compute the closest intersection of the view ray with the hull isosurface for isovalue $h \in \mathbb{R}$, resulting in the intersection distance $\tau \in \mathbb{R}$ (if there is an intersection), i.e., $\phi\mathcal{N}(\mathbf{B}(\mathbf{p}+\tau\mathbf{r}),\mu,\Sigma) = h$, which gives:

$$(\mathbf{B}(\mathbf{p}+\tau\mathbf{r})-\mu)^T\Sigma^{-1}(\mathbf{B}(\mathbf{p}+\tau\mathbf{r})-\mu) = R \quad (6)$$

with $R = -2\ln\left(h\sqrt{(2\pi)^k\det(\Sigma)}/\phi\right)$. Solving Eq. (6) for τ leads to a quadratic root finding problem $\tau^2 a + \tau b + c = 0$ with

$$a = \mathbf{r}^T\mathbf{B}^T\Sigma^{-1}\mathbf{B}\mathbf{r}, \quad b = 2(\mathbf{p}^T\mathbf{B}^T\Sigma^{-1}\mathbf{B}\mathbf{r} - \mu^T\Sigma^{-1}\mathbf{B}\mathbf{r}),$$
$$c = R - \mathbf{p}^T\mathbf{B}^T\Sigma^{-1}\mathbf{B}\mathbf{p} - \mu^T\Sigma^{-1}\mu + 2\mu^T\Sigma^{-1}\mathbf{B}\mathbf{p}$$

and the closed-form roots $\tau_{1,2} = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$. For each Gaussian, we identify the smallest positive solution for τ. The intersections are computed for all Gaussians and are sorted from front-to-back for rendering. To achieve an expressive compositing, we first alter the color of the Gaussian by changing the v-value of the HSV color space. The outermost hull has a v-value of 0 and the innermost has the original v-value, between, we interpolate. In addition, we add a silhouette enhancement $(1-\mathbf{n}^T\mathbf{v})^4$ using view normal $\mathbf{n}$ and view direction $\mathbf{v}$. Each transparent

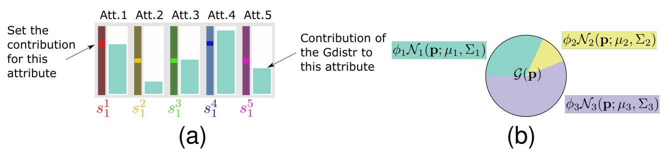

Set the contribution for this attribute
Att.1
Att.2
Att.3
Att.4
Att.5
Contribution of the Gdistr to this attribute
s_1^1
s_1^2
s_1^3
s_1^4
s_1^5
(a)
$\phi_1\mathcal{N}_1(\mathbf{p};\mu_1,\Sigma_1)$
$\phi_2\mathcal{N}_2(\mathbf{p};\mu_2,\Sigma_2)$
$\mathcal{G}(\mathbf{p})$
$\phi_3\mathcal{N}_3(\mathbf{p};\mu_3,\Sigma_3)$
(b)

Fig. 4. (a): Defining the x-axis by setting $s_1^1 \dots s_1^5$ with a slider. Here, the first and fourth attribute have equal weight as these attributes are strongly represented by this Gaussian (see bar charts). (b): For every data point, we the contribution of every Gaussian is color-coded.

layer is weighted using $1 - \text{current}_{\text{depth}}/\text{max}_{\text{depth}}$, where $\text{current}_{\text{depth}}$ means the current number of intersections. Therefore, with an increasing number of intersections, the term vanishes gradually. The final color is multiplied with $1 - \text{current}_{\text{depth}}/\text{max}_{\text{depth}}$, see Fig. 3 (middle).

4.2.3 Direct Volume Rendering

Task **T3** requires the user to identify modes, i.e., local maxima of the mixture distribution. Note that the modes are not necessarily aligned with the mixture components [2], i.e., with the principal axes chosen for **B**. Thus, the user needs to inspect combinations of principal axes manually, which is done by interactively placing the camera, since the ray direction is formed from a linear combination of the view principal axes. Modes are discovered by visualizing accumulated values along a ray, cf. Eq. (2). Similar to Rapp et al. [37], we integate a Gaussian along the ray, here by integrating its projection:

$$\int_{-\infty}^{\infty} \phi\mathcal{N}(\mathbf{B}(\mathbf{p}+\tau\mathbf{r}))\,\mathrm{d}\tau = \int_{-\infty}^{\infty} C\cdot\exp(-\frac{1}{2}(\tau^2 a+\tau b))\,\mathrm{d}\tau$$

$$= C\sqrt{\frac{\pi}{2a}}\exp\left(\frac{b^2}{8a}\right)\operatorname{erf}\left(\frac{2a\tau+b}{\sqrt{8a}}\right)\Bigg|_{-\infty}^{\infty} = 2C\sqrt{\frac{\pi}{2a}}\exp\left(\frac{b^2}{8a}\right) \quad (7)$$

with the scalars $a = \mathbf{r}^T\mathbf{B}^T\Sigma^{-1}\mathbf{Br}$, $b = 2(\mathbf{p}^T\mathbf{B}^T\Sigma^{-1}\mathbf{Br} - \mu^T\Sigma^{-1}\mathbf{Br})$ and $C = \frac{\phi}{\sqrt{(2\pi)^k\det(\Sigma)}}\exp(-\frac{1}{2}(\mathbf{p}^T\mathbf{B}^T\Sigma^{-1}\mathbf{Bp} + \mu^T\Sigma^{-1}\mu - 2\mu^T\Sigma^{-1}\mathbf{Bp}))$. The error function $\operatorname{erf}(x)$ converges to ± 1 for $x \to \pm\infty$, respectively. We accumulate this integral for every Gaussian to obtain the final value, which is a quantitative derived attribute that needs to be mapped to a magnitude channel. We visualize the scalar using the Viridis colormap, which is a continuous multi-hue color map to enhance the visual contrast, equipped with a perceptually-linearized luminance encoding. For exploration purposes, we allow users to define custom transfer functions. Furthermore, we add isolines, which are similar to the stairs metaphor from the MIP paragraph. This enables identifying the direction of increasing values, see Fig. 3 (right).

4.2.4 Basis Exploration

The three raycasting views depend on the choice of basis **B**. Task **T4** requires the user to freely explore different combinations of basis vectors. Thus, we offer the possibility to assemble an orthonormal basis via linear combination of the natural basis vectors, i.e., the attributes. For each attribute, a slider allows setting a scaling factor used in the linear combination, see Fig. 4(a). Next to the slider, a bar chart shows how strongly this attribute contributes to each of the Gaussians, which is measured by projecting its eigenvectors (scaled by their eigenvalues) into the natural basis. The bar charts serve as scented widgets [50] to support the user in the placement of meaningful slider positions. We show three rows of bar charts and sliders, one for each custom basis vector of the view box. Thus, the user can set scaling factors $s_1^j, s_2^j, s_3^j \in \mathbb{R}$ with $j \in \{1,\dots,k\}$, leading to the basis $\widetilde{\mathbf{b}}_1 = \sum_{j=1}^k s_1^j\mathbf{e}^j$, $\widetilde{\mathbf{b}}_2 = \sum_{j=1}^k s_2^j\mathbf{e}^j$, and $\widetilde{\mathbf{b}}_3 = \sum_{j=1}^k s_3^j\mathbf{e}^j$, where $\mathbf{e}^j \in \mathbb{R}^k$ are the eigenvectors of the selected Gaussian's covariance matrix inverse Σ^{-1}. To ensure orthonormality of $\widetilde{\mathbf{b}}_1$, $\widetilde{\mathbf{b}}_2$, $\widetilde{\mathbf{b}}_3$, we apply the Gram-Schmidt process [21, Eq.1] to obtain the orthonormal basis $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$. Instead of setting the coefficients of the eigenvectors, the user can also set the coefficients of the standard Euclidean basis.

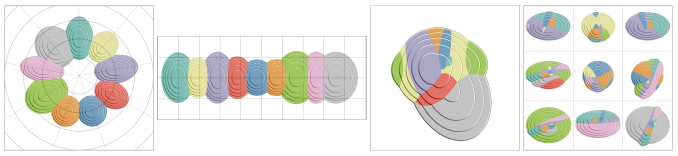


Fig. 5. Different overview visualizations using the MIP-based rendering: circle plot, line plot, principal component plot, and small multiples.

4.2.5 Point Visualization and Analysis

The GMM approximates an underlying point distribution. Task **T5** requires the visual inspection of those points. For this, we render the points as spheres with the same color as the associated Gaussian. Visualization of the mixed probabilities is also possible and activated by default. Thus, we determine the likelihood to be assigned to the individual Gaussian in $[0,1]$ and map this to the angle $[0,2\pi]$, see Fig. 4(b) and later Fig. 10 for an example. In the case of MIP (Section 4.2.1) and cumulative direct volume rendering (Section 4.2.3), we draw spheres as an overlay on top of the rendering. If the hull rendering (Section 4.2.2) is chosen, we place the points at the correct 3D position. Similar to the color compositing for the hull rendering, we multiply the v-value of the HSV color model with $0.5 - \text{current}_{\text{depth}}/\text{max}_{\text{depth}}$ to improve spatial perception. This means, if the sphere lies in the innermost hull it appears darker, because of the superposition of the hulls. It appears brightest in the outermost hull. For further analysis, the user can click on individual points to highlight the assigned Gaussian. To highlight the Gaussian, the stairs or the hulls are drawn in red, depending on which rendering technique is used, see later Fig. 10. An information toolbox shows the coordinates of the selected data point as well as the assigned Gaussian. Also, a histogram is shown that indicates how many data points are assigned to the different Gaussians. A drop-down menu allows the user to select a Gaussian, for which a table shows the coordinates of assigned data points. Task **T6** requires closer inspection of the reasons why a data point is assigned to a certain cluster. This can be revealed by an attribution experiment. In turn, each data point component is replaced by the cluster average, revealing how much the cluster assignment changes when varying the respective component. The result is communicated in the information toolbox.

4.3 Shape and Basis Comparison

The previous chapter introduced the raycasting-based visualizations that aggregate 3D information into 2D images. In the following, we provide methods to support the comparison of Gaussians as well as overview methods helping in the selection of a useful basis.

4.3.1 Shape Comparison

The raycasting-based views used the projected coordinates for the spatial arrangement, which is needed to convey an impression of spatial relationships (**T2**). However, when Gaussians are too close to each other, overlap occurs, which makes it difficult to compare the shape of Gaussians directly, which is required for task **T7**. For that reason, we consider two alternative spatial arrangements, which we later evaluate with users. The arrangements are free from occlusion to support the direct comparison of Gaussian shapes, see Fig. 5 (left). For the *circle plot*, we define a circle and uniformly place Gaussians such that their mean vector coincides with the position on the circle. Afterward, we define the view-box such that the eigenvector with the highest eigenvalue aligns with the normal, and the eigenvector with the second largest eigenvalue is oriented along the tangent of the circle. Again, the eigenvector with the third-largest eigenvalue corresponds to the last basis vector of the view-box. This yields Gaussians placed on the circle with the eigenvectors aligned with the normal and tangent. Similarly, the *line plot* places Gaussians on a line and orients the eigenvectors with the largest and second-largest eigenvalue orthogonal and tangential to the line. While the line plot requires a wider aspect ratio compared to the circle plot, it aligns all Gaussians along a common horizontal line, which benefits size comparisons [29].

In addition, we generate overview arrangements that derive the spatial layout from the data, see Fig. 5 (right). For the *principal component* *plot*, we first apply a principal component analysis (PCA) of the mean vectors. The three eigenvectors with the highest eigenvalues yield an orthonormal basis for the view-box, which provides a *global view*. Lastly, every Gaussian can become the center of the view. Using *small multiples* we provide an overview of possible *local views* onto the spatial arrangements relative to a selected Gaussian.

4.3.2 Basis Comparison

With the visualizations presented so far, we can easily compare the Gaussians and analyze them. For each raycasting-based visualization, however, a number of basis vector permutations are available to choose the principal axis of the plots. Further, the choice of the eigenvectors that are used for the alignment can be varied by the user. While fine control is desirable for detailed inspections, an overview visualization is required such that users do not have to probe each principal axis permutation individually, which is task **T8**. Thus, we provide a *small multiples* visualization that arranges multiple basis vector choices in a matrix layout. This overview serves as preview for different principal axis permutations that the user can choose from to further refine. The views are sorted in descending order by the sum of the eigenvalues of the three chosen principal axes, showing the most relevant views first. To analyze and compare the eigenvalues directly, we provide an overview showing the eigenvalues of every Gaussian with a histogram. Single Gaussians can be disabled in the ray casting views.

4.3.3 Camera Animation

The previous section introduced methods to interactively pick principal axis permutations for the definition of the view-box. When exploring those different options, following for example task **T4**, a smooth transition from one view-box to another is desirable in order to retain object constancy, i.e., to see how individual Gaussians and groups of Gaussians transform and change their spatial relationship. For this reason, we construct a smooth camera animation when transitioning from one view-box to another. This requires the construction of an orthonormal, time-dependent view-box $\mathbf{B}(t) = (\mathbf{b}_1(t)\ \mathbf{b}_2(t)\ \mathbf{b}_3(t)) \in \mathbb{R}^{k\times 3}$ that defines a view-box for every $t \in [0,1]$ during the animation.

Camera Position To construct a smooth path, we define a cubic Bézier curve $\mathbf{c}_{ij}(t) : \mathbb{R} \to \mathbb{R}^k$ connecting two Gaussians $\mathcal{N}_i, \mathcal{N}_j$:

$$\mathbf{c}_{ij}(t) = (1-t)^3\mathbf{p}_0 + 3(1-t)^2 t\mathbf{p}_1 + 3(1-t)t^2\mathbf{p}_2 + t^3\mathbf{p}_3, \qquad (8)$$

which is defined by four control points $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3 \in \mathbb{R}^k$

$$\mathbf{p}_0 = \mu_i, \qquad \mathbf{p}_3 = \mu_j, \qquad (9)$$

$$\mathbf{p}_1 = \mathbf{p}_0 + c\phi_i\sqrt{\lambda_1^i}\mathbf{e}_1^i, \qquad \mathbf{p}_2 = \mathbf{p}_3 - c\phi_j\sqrt{\lambda_1^j}\mathbf{e}_1^j, \qquad (10)$$

where λ_1^i is the largest eigenvalue of Σ_i^{-1} and $\mathbf{e}_1^i$ its corresponding eigenvector (the sign is discussed later). Here, $c = 1/\max_i \lambda_1^i$ is a scaling factor preventing the camera path to overshoot the scene. With the choice of $\mathbf{p}_0$ and $\mathbf{p}_3$ we ensure the start and endpoint of a curve to begin and end with a Gaussian. The points $\mathbf{p}_1, \mathbf{p}_2$ are chosen such that we go from μ_i (μ_j) in direction of $\mathbf{e}_1^i$ ($\mathbf{e}_1^j$), thus, the camera passes through the largest extent of the Gaussian such that we see the two largest extents on the x-axis and y-axis of the screen (and the third largest extent as the z-axis). One problem remains, as the eigenvector is uniquely defined except for the sign. Therefore, when connecting two Gaussians, four possible paths exist from which we select the shortest path [49].

Orthonormal View-Box During the camera position change, the view-box $\mathbf{B}(t)$, that reveals the data in the three-dimensional subspace, needs to be changed, too. We define the starting view-box as $\mathbf{B}_{ij}(0) := (s_i\mathbf{e}_1^i\ \mathbf{e}_2^i\ \mathbf{e}_3^i)$, where $\mathbf{e}_1^i, \mathbf{e}_2^i, \mathbf{e}_3^i$ are the eigenvectors of Σ_i^{-1} with the corresponding three largest eigenvalues $\lambda_1^i \geq \lambda_2^i \geq \lambda_3^i$. The sign $s_i \in \{-1,+1\}$ is the correct sign of the eigenvector according to the transition $\mathbf{c}_{ij}$. Similarly, we define the end view-box as $\mathbf{B}_{ij}(1) := (s_j\mathbf{e}_1^j\ \mathbf{e}_2^j\ \mathbf{e}_3^j)$. The task is now to find a smooth transition $\mathbf{B}_{ij}(t) = (\mathbf{b}_1(t)\ \mathbf{b}_2(t)\ \mathbf{b}_3(t))$

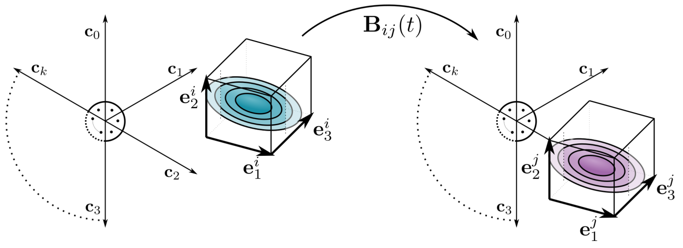


Fig. 6. The left view-box is transformed to the view-box on the right by applying $\mathbf{B}_{ij}(t)$. The transformation ensures that it remains orthonormal. The vectors $\mathbf{c}_0, \ldots, \mathbf{c}_k$ represent the natural basis vectors (attributes).

between $\mathbf{B}_{ij}(0)$ and $\mathbf{B}_{ij}(1)$ such that $\mathbf{b}_k(t)\cdot\mathbf{b}_l(t) = \delta_k^l$ where δ_k^l is the Kronecker delta for $k,l \in \{1,2,3\}, t \in [0,1]$. This means we need a time-dependent view-box that remains on an orthonormal subspace basis. First, we linearize the view-box: $\mathbf{B}_{ij}(t) = \mathbf{B}_{ij}(0)\cdot(1-t) + \mathbf{B}_{ij}(1)\cdot t$, which ensures a transition. To ensure orthonormality, we apply the Gram-Schmidt process [21, Eq.1]. Note, that while $\mathbf{e}_2^i, \mathbf{e}_3^i$ (at $t=0$) are set, the choice of $\mathbf{e}_2^j, \mathbf{e}_3^j$ (at $t=1$) is not uniquely defined regarding the sign. We choose the option with smallest enclosing angle, thus if $\mathbf{e}_2^i\cdot\mathbf{e}_2^j < 0$, we flip the orientation of $\mathbf{e}_2^j$. Similarly, $\mathbf{e}_3^j$ is oriented to have smallest enclosing angle with $\mathbf{e}_3^i$. Optionally, but used by default, the mean vector is replaced by $\mu \to \mathbf{B}\mathbf{B}^T(\mu - \mathbf{c}(t))$ to project it onto the 3D subspace given by the view-box. If we would omit this projection, the visualization would only reveal a sliced view, where the projected data points may not be visually located around the mean of the Gaussian. This may be correct in terms of the projection but may also be confusing.

5 Implementation

In the following, we elaborate on the implementation details, where we use OpenGL with C++ to achieve real-time performance, see Fig. 7.

Pre-processing In Eqs. (5), (6) and (7) several terms can be pre-computed, including the following 4×4 matrix and the 4×1 vector:

$$\begin{pmatrix} \mathbf{B}^T\Sigma_i^{-1}\mathbf{B} & \mathbf{0} \\ \mathbf{0}^T & \frac{1}{\sqrt{(2\pi)^k\det(\Sigma_i)}} \end{pmatrix}_{4\times 4}, \quad \begin{pmatrix} 2\mathbf{B}^T\Sigma_i^{-1}(\mathbf{c}-\mu_i) \\ (\mathbf{c}-\mu_i)^T\Sigma_i^{-1}(\mathbf{c}-\mu_i) \end{pmatrix}_{4\times 1}$$

which are precomputed for each Gaussian and sent to uniform buffers on the GPU whenever view-box $\mathbf{B}$ or the camera position $\mathbf{c}$ change. We further use shader storage buffers objects (SSBOs) to store all data points on the GPU. In addition, we use four SSBOs to transfer the orthonormal basis vectors of the view-box $\mathbf{B}$ and the current camera position $\mathbf{c}$ to the GPU. To ensure real-time rendering in the case of many data points, we perform calculations with compute shaders.

GPU Implementation To render the visualization, we use several compute shaders. We start with a shader to determine the projected positions of the data points. For every data point $\mathbf{q}$, the shader is called and we determine the $3D$ coordinates $\mathbf{B}(\mathbf{q}-\mathbf{c})$ by iterating over the entries of the SSBOs for view-box $\mathbf{B}$ and the camera position $\mathbf{c}$. We project the $3D$ coordinates from world space to normalized device coordinates, where the projected result is stored in an image-space texture. We use the red channel to store depth information, the green channel to store the assigned Gaussian, and the blue channel to store the index of the data point. Afterward, we call a compute shader twice that generates spheres from the projected points. So far, the information of the data point is only stored in one single pixel. Therefore, we use another compute shader that expands the pixel in x- and y-direction to cover a sphere. We compare the depth information in case spheres overlap and store the sphere closest to the camera. Additionally, we use the alpha channel of the texture to store the distance to the center of the data point, which is useful for later shading.

Independent of the underlying rendering, we do all calculations described in Sec. 4.2 in a computer shader. To generate a transfer function for DVR, we use atomic counters to count the resulting value of a pixel from the GMM, cf. Eq. (2). These values are stored in a

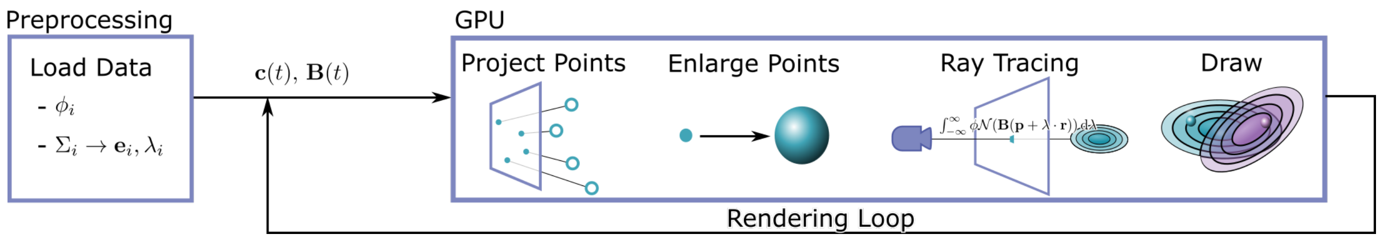

Preprocessing
Load Data
- ϕ_i
- $\Sigma_i \rightarrow \mathbf{e}_i, \lambda_i$
$\mathbf{c}(t)$, $\mathbf{B}(t)$
GPU
Project Points
Enlarge Points
Ray Tracing
$\int_{-\infty}^{\infty} \phi \mathcal{N}(\mathbf{B}(\mathbf{p} + \lambda \cdot \mathbf{r}))d\lambda$
Draw
Rendering Loop

Fig. 7. Pipeline of our tool. The pre-processing comprises data loading and the computation of eigenvector/eigenvalues. Then, for every point in time the camera position $\mathbf{c}(t)$ and the view-box $\mathbf{B}(t)$ is determined to project the data points that are enlarged and visualized as spheres. Then, ray tracing is performed and the result is shown. This is repeated for a different camera position and view-box.

texture to display the values by a histogram. To highlight Gaussians, we store the index of the data point that the mouse position hovers over. The compositing of the rendering and the data points is performed in the subsequent compute shader.

6 Evaluation

The evaluation is divided into three parts. First, we collected informal feedback on our GMM exploration tool, which was used to extend our framework to the current state. Second, we performed a qualitative evaluation to assess the usefulness of our tool, and third, we measured the performance of the algorithms on different data sets.

Informal Feedback The tool was developed in close collaboration with three experts in machine learning (ML), M1, M2, M3, with a focus on GMMs (10, 8, 3 years of experience) and two visualization experts, V1, V2 (7, 3 years of work experience). For the informal feedback, we used various revisions of the tool and discussed its current state. We showed the tool to experts and introduced them to new features. The experts then used the tool on their own to analyze and explore models and data, while we recorded their comments and suggestions that we used later to improve the tool.

The first prototype showed the MIP rendering without the stairs. V1 and M1 commented that it is hard to perceive where the mean vector is. Therefore, we implemented the stairs. M1 asked for a feature to identify different modes, i.e., regions with high probability in the GMM. For this purpose, we integrated the cumulative DVR that accumulates the values of the GMM. M2 also asked for a stairs metaphor on this rendering style, which we added. V1 and V2 commented that it is hard to distinguish between the individual Gaussians. Hence, we added the histogram showing the eigenvalues as well as the circle and the line plot. They also asked for a better visualization of the 3D subspace and suggested using hulls, as they are similar to the stairs metaphor. During visualization design, we experimented with various ray-tracing techniques, e.g., refraction, which were considered visually pleasing (M1), but distorted the data. Therefore, we kept the hull visualization. M2 asked for means to navigate in high-dimensional space. Thus, we integrated different view-boxes that are determined by an orthonormal basis which can be set by the user. Furthermore, M3 asked for a way to detect data points that lie in-between clusters. We, therefore, integrated controls to filter data points. M3 asked for a method to view the image data associated with a selected data point as well as its cluster assignment distribution, for which we added an information box.

In a final session, we showed the tool again to experts. V1 and V2 were satisfied by the options to vary the visualizations. M1, M2, and M3 commented on the alternatives and stated that this tool allows analyzing the data in a way that is tailored to their needs. M1 stated, “With this framework, I can clearly explore and analyze high dimensional data points and their GMMs.” M3 commented, “This tool is very extensive!” He also asked us to provide him with the executable such that he can use it for his research. Also, the smooth camera animation allows getting better insights into the data. They consider this tool to be a clear and novel contribution to the visual analysis of GMMs and would like to have the source code publicly available.

Quantitative Evaluation To assess the usefulness of our tool, we utilized an anonymous online questionnaire, see the additional material. First, information about the participants’ background was gathered and

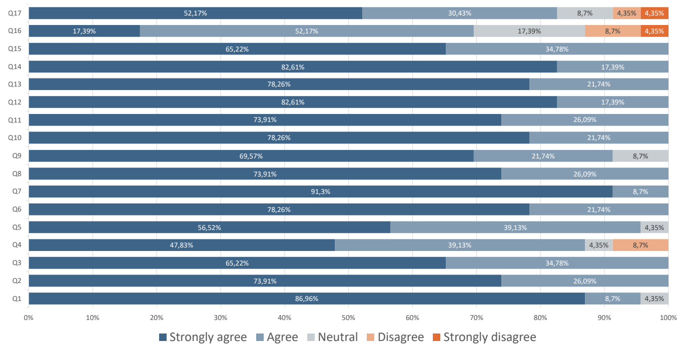


Fig. 8. The results of the evaluation are summarized. Here Q refers to the individual questions, which could be answered with a five point Likert scale from 1-strongly disagree to 5-strongly agree.

the tool was explained. In this study, 23 people participated, including 10 females and 13 males, ranging from 23 to 53 years. Further, we asked if they are familiar with machine learning (ML), visualization (Vis), human-computer interaction (HCI), or other fields. Multiple selections were possible, where 17 stated they are familiar with ML, 19 with Vis, and 16 with HCI. Twelve participants have 1–10 years of experience, seven between 10–19, and four between 19–28, with average of 9.5 and a median of 8 years of experience. Afterwards, we asked several questions about the tool and provided a five-point Likert scale from 1-strongly disagree to 5-strongly agree, see Fig. 8 for a summary. In the following, we analyze the study results.

First, it was confirmed that the stairs of the MIP visualization, cf. Sec. 4.2.1, help to localize the mean (Q1). Second, the participants stated that the visualization helps to understand where the probability of a Gaussian is the highest (Q2) (fulfill task **T1**). The same questions resulted in more mixed responses for the hull rendering (Q3, Q4), cf. Sec. 4.2.2. One participant commented that during the exploration it is easier to identify the regions of high probability, but from a screenshot alone it is more difficult. In addition, the dashed lines of the hull rendering, indicating the intersection of the innermost hulls, support the spatial impression (Q5) (fulfill task **T2**).

Next, we evaluated the DVR rendering, cf. Sec. 4.2.3. The participants confirmed that it helps to identify new modes (Q6). Moreover, we can conclude that the visualization supports the identification where the probability is the highest (Q7) and that the isolines yield an indication in which direction the probability increases (Q8). Finally, the majority confirmed that changing the transfer function allows identifying new emergent modes (Q9) (fulfill task **T3**).

Next, we evaluated the basis exploration and data point visualization, cf. Sections 4.2.4 and 4.2.5. Setting the portions of the orthonormal basis gains insights into the data (Q10) and this allows exploring the data easily (Q11) (fulfill task **T4**). The participants had no problem identifying data points with high probability that could belong to another Gaussian (Q12). Moreover, the information box supports understanding and helps analyzing why a data point has a higher probability of being part of another cluster (Q13) (fulfill task **T5**, **T6**).

Finally, we determined which shape comparison layout suits the exploration best, cf. Sec. 4.3.1. The circle layout was slightly preferred over the line layout, where both provide an overview to compare the

Table 1. Minimal, average, and maximal computation time in seconds during camera navigation visiting each of $k+1$ Gaussians in dimension k, number of points, for a screen resolution of 2048×2048 pixels.

k	#Points	Rendering	min t(s)	∅ t(s)	max t(s)
3	1,000	MIP/DVR	0.0014	0.0034	0.0084
		Hull	0.0016	0.0037	0.0086
3	10,000	MIP/DVR	0.0013	0.0032	0.0071
		Hull	0.0022	0.0036	0.0087
8	1,000	MIP/DVR	0.0019	0.0033	0.0076
		Hull	0.0022	0.0051	0.0118
8	10,000	MIP/DVR	0.0015	0.0032	0.0074
		Hull	0.0023	0.0052	0.0114

extents of the Gaussians (Q14, Q15) (fulfill task **T7**). More mixed responses were given for the Principal Component Plot (PCP) (Q16). For basis comparison, cf. Sec. 4.3.1, the small multiples visualization (Q17) gives an overview of the GMM (fulfill task **T8**). Between the principal component plot and the small multiples visualization, we asked which layout the participants would prefer, where four preferred the principal component plot, ten the small multiples, and nine both.

Performance To measure the performance, we used artificial data sets. We track the computation time of every frame during a camera animation. The computationally expensive part is the projection of the data points onto the 3D subspace and the following upload to the GPU. For the data, we generated a standard 3-simplex and an 8-simplex with unit edge lengths. Then, we placed a Gaussian at every vertex, i.e., the mean vector equals the vertex, and fixed the covariance matrix $\Sigma = \mathbf{I}$ as the identity matrix. Finally, we combined the Gaussians into a GMM using uniform weights and sampled random data points from the GMM. Thus, we have $N = 4$ or $N = 9$ Gaussians, $k = 3$ or $k = 8$ dimensions, and the data lives on a $m = k - 1$ dimensional subspace. Tab. 1 lists performance results, for scenarios with $1,000$ and $10,000$ points and a resolution of 2048^2 pixels. The experiments were conducted on an Intel Core i9 @3.60GHz, 32 GB RAM, and an NVIDIA GeForce GTX 2080. Changing the positions of the mean vectors or the variances has no significant impact on the performance. Even with this comprehensive data set, we achieve real-time performance.

7 Findings

In this section, we describe insights that we obtained on two different data sets. A third example can be found in the additional material.

Country Data Kaggle's country data set [39] provides information about 167 countries like the mortality of children under five, exports and imports in terms of GDP, or health and income. We trained a GMM on this data set using the EM algorithm [27] and the Akaike information criterion (AIC) [1] to obtain the best number of components. The data points were then clustered into three groups by assigning them to the respective component with the highest conditional probability. Thus, we have $N = 3$ Gaussians, $k = 9$ dimensions, and the data lives on a $m \approx 6$ dimensional subspace, as the three Gaussians capture 90% of the information for 5, 6, and 7 eigenvalues, respectively. We first explored the data using the small multiple overview to familiarize ourselves with the data and the model, see Fig. 9 (left). One view shows a clear cluster separation, i.e., colored points are within their respective colored GMM region and not closer to another cluster. The view is selected and shown in more detail in Fig. 9 (middle). The thereby chosen basis vectors are a linear combination of the natural basis, i.e., the attributes of the data. It is therefore interesting to learn which attributes of the underlying data are combined to form the basis vectors. Looking at the basis contribution sliders in Fig. 9 (right), reveals that the first row (x-axis) is formed from export and import. Note that a medium slider position corresponds to zero contribution. Similarly, the second row (y-axis) is formed from child mortality and total fertility. The bar charts also reveal why those basis vectors are separating the clusters very well. The first cluster is strongly influenced by export/import (orange box) and less influenced by child mortality and total fertility, which is reversed for the third cluster (blue box). Inspection of the data points inside the clusters revealed that the first cluster contains Qatar, Luxembourg, Singapore, Ireland, and Brunei, which are among the richest countries in the world. The poorest countries in the world, namely, Burundi, Central African Republic, and the Democratic Republic of the Congo, are all part of the third cluster.

Further, we could identify edge cases, where countries have a probability associated with multiple clusters. We observed that the United Arab Emirates (UAE) belong to the second cluster with a probability of ≈25% (see Fig. 10). The tool helped us to immediately and interactively explore the underlying reasons. We found that the UAE have a lower income and gdpp value than the average of this cluster. Setting these values in an attribution experiment to the cluster mean would significantly decrease the probability from 25% to 10% for a change in income and to 9% for a change in gpdd.

Covid-19 Data The next data set contains several attributes for 101 countries, including information on the Covid-19 pandemic [4]. We trained a GMM by the EM algorithm and the AIC and then used the GMM to cluster the data points. Here, we have $N = 3$ Gaussians, $k = 11$ dimensions, and the data lives on a $m \approx 7$ dimensional subspace, as the three Gaussians capture 90% of the information for 6, 7, and 7 eigenvalues, respectively. We started our data exploration by looking at the distributions of deaths due to Covid-19 per capita and the daily tests, respectively. It revealed, for instance, that at the time of record Belgium had the highest number of deaths per capita, see Fig. 11(a).

Using our tool, we could answer why the countries Denmark and Luxembourg share a single cluster separate from other countries (see Fig. 11(b)). When exploring the portion of eigenvectors w.r.t. the canonical basis, we found that both countries have a low average temperature and the most daily tests. This observation allowed us to identify an error in the publicly available data set: the average temperature of Denmark according to the data is $-14^\circ C$. Actually, it should be around $9^\circ C$. We contacted the data author who corrected the error.

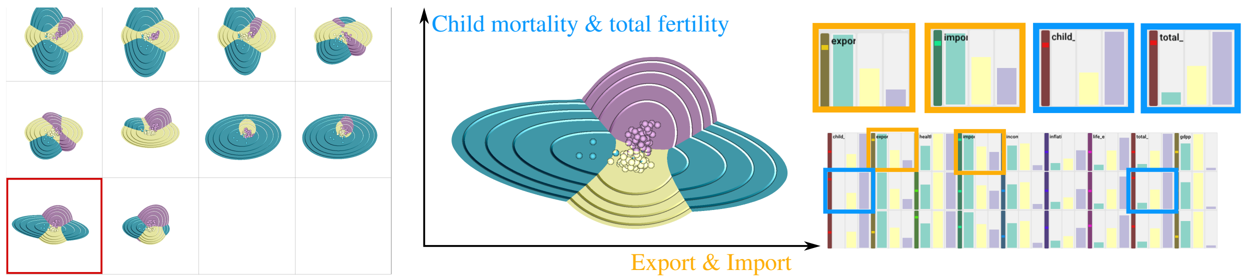
From a mathematical point of view, a further interesting question is the emergence of modes, i.e., the number of modes can be larger than the number of mixture components. Emergent modes might not be very pronounced, i.e., non-persistent in terms of persistent homology, and are thus difficult to detect. During our analyses of the two data sets, we found the tool helpful for this task. Fig. 12 shows a setup of three Gaussian placed at the points of an equilateral triangle in three dimensions. In the barycenter of the triangle, a new mode emerges that we can identify by adjusting the transfer function for the GMM.

CIFAR-10 Data We refer to the additional material for another example on a data set with $N = 14$ Gaussians, $k = 246$ dimensions, and data living on a $m \approx 212$ dimensional subspace.

8 Discussion

Based on the problem statement in Sec. 3, we developed a GMM exploration framework to address the stated problems.

To visualize the GMMs, we devised three raycasting-based visualization techniques, cf. Sec. 4.2. The MIP visualization provides a clear and descriptive metaphor for understanding the regions associated with the Gaussians (task **T1**). A downside is the necessary exploration process. Rotating the camera in 3D subspace can be difficult for beginners, as some regions from a so-far unseen Gaussian may suddenly appear. This can happen when the camera observes a Gaussian that locally contributes a high probability to the mixture. The ray of the camera hits fragments of high probability. In the next frame, this area can be rotated, and another Gaussian appears, which has a larger probability along this ray. To avoid this, the experts switched between MIP and Hull rendering. Using Hull rendering the exploration in 3D is easy to follow (task **T2**), but the higher likelihood of the Gaussians is not visually encoded. The DVR method helps identify regions of high and small probabilities (task **T3**). Adapting the transfer function allows detecting the emergence of modes that do not directly correspond to components of the mixture. This requires some exploration. However, finding these modes analytically is a hard problem [8]. The informal feedback reflects that the DVR is a valuable addition for better understanding the GMM. Defining an arbitrary axis combination for the view-box is possible for all three raycasting-based views to explore



Small multiples of the first Gaussian ($m = 5$)   For one multiple, we highlight attributes contributing to Gaussians with orange/cyan boxes.

Fig. 9. The small multiple view (left) on the country data set reveals basis vectors for which a cluster separation between the richest and the poorest countries in the world is apparent (middle). The basis vector weights (right) show which data attributes are represented by the basis vectors.

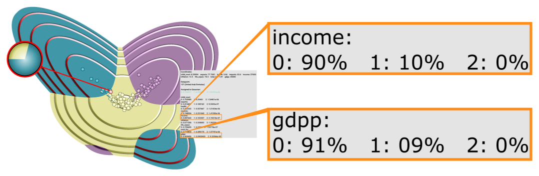


Fig. 10. One data point, the United Arab Emirates (UAE), has a probability over 25% that it belongs to the second cluster (1) due to low income and low gdpp. The attribution experiment shows that the probability drops to $\approx 10\%$ if income and gdpp were replaced by cluster means.

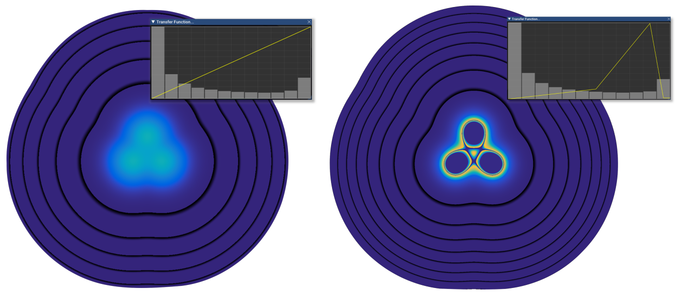


Fig. 12. Initial position (left) and a fourth mode (new maximum) has emerged at the center after the transfer function was changed (right).

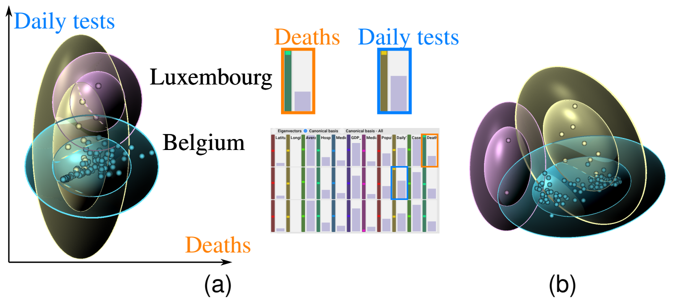


Fig. 11. (a): Regarding the deaths due to or with Covid-19 divided by the population and the daily tests, respectively, Belgium has the highest number of deaths and Luxembourg has the highest number of daily tests. (b): A closer look reveals that Denmark and Luxembourg belong to one cluster due to an error in the data.

the data more freely, cf. Sec. 4.2.4 (task **T4**). The DVR visualization mostly aids the exploration of the GMM itself and is less important for a data points-driven exploration. Instead, exploring the data points is better assisted by the MIP and the Hull rendering. Both techniques are well-suited to understand the GMM by simultaneously identifying the individual Gaussians and their border regions. We display the data points to convey how well the GMMs approximate the data, cf. Sec. 4.2.5 (task **T5**). The impact of an attribute on the assignment to a cluster is revealed by an attribution experiment (task **T6**).

To further understand the Gaussians and to obtain an overview, layouting techniques were presented, cf. Sec. 4.3. These techniques allow for a shape comparison (task **T7**). The circle plot is more space-saving than the line plot, while the line plot may exhaust the screen space, requiring either panning or zooming out. In contrast to the circle plot, the line plot allows comparing the eigenvalues. With its alignment, it is easier to visually detect if the eigenvalues of a Gaussian are larger or smaller than the eigenvalues of another Gaussian. We observed that the experts preferred the circle plot, mostly due to the more succinct representation. Nonetheless, they would not omit the line plot. The PCA plot was barely used and not considered as particularly useful. Mostly the Gaussians were projected very close to each other such that no interesting features or outliers could be identified. To provide an overview of different basis combinations, we provided a small multiple visualization that provided useful insights of the data points (task **T8**). Since it depends strongly on the scientific question of what is interesting at all or what the experts want to find, we cannot guarantee to find all useful arrangements. Nevertheless, the small multiples greatly improve the exploration process and provide a better overview than the standard display of selected camera positions.

A common problem in clustering that GMMs inherit is that the units need careful scaling and normalization, e.g., by z-score normalization.

Our rendering implementations currently scale linearly in the number of Gaussians N, which could be accelerated with parallel prefix sums. Currently, we use colors from the *iWantHue* website [26]. If a large number of Gaussians are rendered and their colors become perceptually indistinguishable, the circle and line plots still allow examining individual Gaussians. However, for the small multiples, user interaction is needed to explore individual Gaussians, e.g., deactivating and highlighting. The domain dimensionality k can be arbitrarily high. What matters is the dimensionality of the data manifold m, for which it is difficult to specify a strict scalability limit since this is data-dependent. The larger m, the less complete is a 3D projection. Our approach orders subspaces by the magnitude of the eigenvalues, making sure that the most informative subspaces are shown first.

9 Conclusion and Future Work

We presented the first visualization tool to analyze and explore high-dimensional GMMs. The system was developed iteratively in close collaboration with ML and visualization experts. With our tool, users gained new insights into GMMs and the underlying data, and were able to generate new hypotheses, which was not possible before to this extent. In the future, we will investigate other dimensionality reduction and spatial layouting techniques. The challenge is to align the Gaussians such that they fit the projected data points and mean vectors. If the mean vectors μ_i and the data points $\mathbf{q}_j$ are projected in a 3D subspace, i.e., $\bar{\mu}_i, \bar{\mathbf{q}}_j$, we have to find a 3×3 covariance matrix $\bar{\Sigma}$ such that $\sum_{k=1}^{n}(\|(\bar{\mathbf{q}}_k - \bar{\mu}_i)^T \bar{\Sigma}^{-1}(\bar{\mathbf{q}}_k - \bar{\mu}_i)\| - \|(\mathbf{q}_k - \mu_i)^T \Sigma^{-1}(\mathbf{q}_k - \mu_i)\|)^2 + \sum_{k=1}^{N}(\|(\bar{\mu}_k - \bar{\mu}_i)^T \Sigma^{-1}(\bar{\mu}_k - \bar{\mu}_i)\| - \|(\mu_k - \mu_i)^T \Sigma^{-1}(\mu_k - \mu_i)\|)^2$ is minimized. We consider this challenging task as future work.

Acknowledgments

This work was supported by the Carl Zeiss Foundation within the project Interactive Inference.

References

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[2] C. Améndola, A. Engström, and C. Haase. Maximum number of modes of Gaussian mixtures. *Information and Inference: A Journal of the IMA*, 9(3):587–600, Jun 2019.

[3] A. O. Artero, M. C. F. de Oliveira, and H. Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. In *IEEE Symposium on Information Visualization*, pp. 81–88, 2004.

[4] S. Belkacem. COVID-19 data analysis and forecasting: Algeria and the world. *arXiv preprint arXiv:2007.09755*, 2020.

[5] M. Ben-Menachem. Parallel coordinates: Visual multidimensional geometry and its applications, is written by alfred inselberg, and published by springer; © 2009; isbn 978-0-387-21507-5; pp. 580. *SIGSOFT Softw. Eng. Notes*, 35(3):39, May 2010.

[6] E. Bertini, A. Tatu, and D. Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2203–2212, 2011.

[7] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[8] M. Á. Carreira-Perpiñán and C. K. I. Williams. On the number of modes of a Gaussian mixture. In L. D. Griffin and M. Lillholm, eds., *Scale Space Methods in Computer Vision*, pp. 625–640. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[9] W. W. Chan. A survey on multivariate data visualization. In *Dep. Comput. Sci. Eng. Hong Kong Univ. Sci. Tech.*, vol. 8, pp. 1–29, 2006.

[10] J. H. T. Claessen and J. J. van Wijk. Flexible linked axes for multivariate data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2310–2316, 2011.

[11] A. Das, U. R. Acharya, S. S. Panda, and S. Sabut. Deep learning based liver cancer detection using watershed transform and Gaussian mixture model techniques. *Cognitive Systems Research*, 54:165 – 175, 2019.

[12] N. Elmqvist, P. Dragicevic, and J. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1148, 2008.

[13] E. Fanea, S. Carpendale, and T. Isenberg. An interactive 3D integration of parallel coordinates and star glyphs. In J. Stasko and M. Ward, eds., *Proceedings of the IEEE Symposium on Information Visualization (InfoVis, October 23–25, Minneapolis, Minnesota, USA)*, pp. 149–156. IEEE Computer Society, Los Alamitos, CA, 2005.

[14] B. J. Ferdosi and J. B. Roerdink. Visualizing high-dimensional structures by dimension ordering and filtering using subspace analysis. *Computer Graphics Forum*, 30(3):1121–1130, 2011.

[15] S. Garg, I. V. Ramakrishnan, and K. Mueller. A visual analytics approach to model learning. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pp. 67–74, 2010.

[16] X. He, Y. Tao, Q. Wang, and H. Lin. Multivariate spatial data visualization: a survey. *Journal of Visualization*, 22(5):897–912, Aug 2019.

[17] J. Heinrich and D. Weiskopf. State of the Art of Parallel Coordinates. In M. Sbert and L. Szirmay-Kalos, eds., *Eurographics 2013 - State of the Art Reports*. The Eurographics Association, 2013.

[18] J.-F. Im, M. McGuffin, and R. Leung. Gplom: The generalized plot matrix for visualizing multidimensional multivariate data. *IEEE Transactions on Visualization and Computer Graphics*, 19:2606–14, 12 2013.

[19] J. Kruskal and M. Wish. *Multidimensional Scaling*. Sage Publications, 1978.

[20] F. Leisch. Visualizing cluster analysis and finite mixture models. In *Handbook of data visualization*, pp. 561–587. Springer, 2008.

[21] S. J. Leon, Å. Björck, and W. Gander. Gram-schmidt orthogonalization: 100 years and more. *Numerical Linear Algebra with Applications*, 20(3):492–532, 2013.

[22] S. Liu, D. Maljovec, B. Wang, P. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1249–1268, 2017.

[23] S. Liu, B. Wang, J. J. Thiagarajan, P.-T. Bremer, and V. Pascucci. Visual exploration of high-dimensional data through subspace analysis and dynamic projections. *Comput. Graph. Forum*, 34(3):271–280, June 2015.

[24] J. Ma, J. Chen, L. Chen, X. Zhou, X. Qin, Y. Tang, G. Sun, and J. Chen. Gaussian mixture model-based target feature extraction and visualization. *Journal of Visualization*, 24(3):545–563, 2021.

[25] J. Ma, X. Jiang, J. Jiang, and Y. Gao. Feature-guided Gaussian mixture model for image matching. *Pattern Recognition*, 92:231 – 245, 2019.

[26] Mathieu Jacomy. iwanthue. Website, 2014. Called Mar. 30, 2021 from: `http://http://medialab.github.io/iwanthue/`.

[27] G. J. McLachlan and D. Peel. *Finite mixture models*, vol. 299 of *Probability and Statistics – Applied Probability and Statistics Section*. Wiley, New York, 2000.

[28] M. Migut and M. Worring. Visual exploration of classification models for risk assessment. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pp. 11–18, 2010.

[29] T. Munzner. *Visualization analysis and design*. CRC press, 2014.

[30] C. Nobre, M. Streit, M. Meyer, and A. Lex. The state of the art in visualizing multivariate networks. *Computer Graphics Forum (EuroVis)*, 38:807–832, 2019.

[31] P. Oesterling, C. Heine, H. Jänicke, and G. Scheuermann. Visual analysis of high dimensional point clouds using topological landscapes. In S. North, H.-W. Shen, and J. van Wijk, eds., *IEEE Pacific Visualization Symposium PacificVis 2010*, pp. 113–120. Los Alamitos, CA, USA, 2010.

[32] P. Oesterling, C. Heine, H. Jänicke, G. Scheuermann, and G. Heyer. Visualization of high-dimensional point clouds using their density distribution's topology. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1547–1559, 2011.

[33] P. Oesterling, C. Heine, G. H. Weber, and G. Scheuermann. Visualizing nD point clouds as topological landscape profiles to guide local data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):514–526, 2013.

[34] G. Palmas, M. Bachynskyi, A. Oulasvirta, H.-P. Seidel, and T. Weinkauf. An edge-bundling layout for interactive parallel coordinates. In *Proc. IEEE PacificVis*. Yokohama, Japan, March 2014.

[35] G. Palmas and T. Weinkauf. Space Bundling for Continuous Parallel Coordinates. In E. Bertini, N. Elmqvist, and T. Wischgoll, eds., *EuroVis 2016 - Short Papers*. The Eurographics Association, 2016.

[36] H. Purohit, R. Tanabe, T. Endo, K. Suefusa, Y. Nikaido, and Y. Kawaguchi. Deep autoencoding GMM-based unsupervised anomaly detection in acoustic signals and its hyper-parameter optimization. In *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pp. 175–179, 2020.

[37] T. Rapp, C. Peters, and C. Dachsbacher. Visual analysis of large multivariate scattered data using clustering and probabilistic summaries. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1580–1590, 2020.

[38] P. Rheingans and M. DesJardins. Visualizing high-dimensional predictive model quality. In *Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145)*, pp. 493–496, 2000.

[39] ROHAN KOKKULA. Unsupervised learning on country data. Website, 2014. Called Mar. 30, 2021 from: `www.kaggle.com/rohan0301/unsupervised-learning-on-country-data`.

[40] R. Rosenbaum, J. Zhi, and B. Hamann. Progressive parallel coordinates. *IEEE Pacific Visualization Symposium 2012, PacificVis 2012 - Proceedings*, pp. 25–32, 02 2012.

[41] J. Seo and B. Shneiderman. A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. In *in Proceedings of IEEE Symposium on Information Visualization*, pp. 65–72, 2004.

[42] I. Shahin, A. B. Nassif, and S. Hamsa. Emotion recognition using hybrid Gaussian mixture model and deep neural network. *IEEE Access*, 7:26777–26787, 2019.

[43] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum*, 28(3):831–838, 2009.

[44] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnor, and D. Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Proc. IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pp. 59–66, Oct 2009. Won the SPP Collaboration Award in the DFG priority program on Scalable Visual Analytics (SPP 1335).

[45] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319, 2000.

[46] T. Toda, Y. Ohtani, and K. Shikano. Eigenvoice conversion based on Gaussian mixture model. In *International Conference on Spoken Language Processing (INTERSPEECH)*, pp. 2446–2449, 11 2006.

[47] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. Interactive lenses for visualization: An extended survey. *Computer Graphics Forum*, 36(6):173–200, 2017.

[48] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *J Mach Learn Res*, 9(86):2579–2605, 2008.

[49] S. Vincent and D. Forsey. Fast and accurate parametric curve length computation. *Journal of graphics tools*, 6(4):29–39, 2001.

[50] W. Willett, J. Heer, and M. Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, 2007.

[51] Y. Xiang, D. Fuhry, R. Jin, Y. Zhao, and K. Huang. Visualizing clusters in parallel coordinates for visual knowledge discovery. In P.-N. Tan, S. Chawla, C. K. Ho, and J. Bailey, eds., *Advances in Knowledge Discovery and Data Mining*, pp. 505–516. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[52] S. Yin, Y. Zhang, and S. Karim. Large scale remote sensing image segmentation based on fuzzy region competition and Gaussian mixture model. *IEEE Access*, 6:26069–26080, 2018.

[53] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu. Scattering points in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1001–1008, 2009.

[54] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual clustering in parallel coordinates. *Computer Graphics Forum*, 27(3):1047–1054, 2008.

[55] Z. Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2, pp. 28–31 Vol.2, 2004.

[56] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.