**Visualization**
Prof. Bernhard Schmitzer, Uni Göttingen, summer term 2025

# Problem sheet 6

- *Submission by* ***2025-06-25*** *18:00 via StudIP* ***as a single jupyter Notebook/ZIP****. Please combine all results into one notebook or archive.*
- *Use Python 3 for the programming tasks as shown in the lecture. If you cannot install Python on your system, the GWDG jupyter server at* https://jupyter-cloud.gwdg.de/ *might help. Your submission should contain the final images as well as the code that was used to generate them.*
- *Work in groups of up to three. Clearly indicate names and enrollment numbers of all group members at the beginning of the submission.*

**Exercise 6.1: PCA for histograms.**
The file `points.npz` contains 100 point clouds in 1 dimension with 1000 points per cloud. Each cloud was sampled from a normal distribution with different mean and standard deviation.

1. Import the point clouds and for each compute the mean, standard deviation, and the histogram (normalized as density) over 50 bins on the range $[-5, 5]$.
2. Apply PCA on the histogram vectors (100 vectors in 50 dimensions).
3. Plot the spectrum and cumulative variance as in the lecture. Comment on the intrinsic dimension of the dataset.
4. Show the two-dimensional PCA embedding and how the embedded position of a histogram relates to its mean and standard deviation.
5. Adopt the interactive example from the lecture to this problem: When clicking in the 2d embedding plot, show in a separate plot the 'projected histogram' corresponding to these coordinates.
   *Hint:* This is interactive and will not work in a PDF. Therefore, for this problem sheet please submit a jupyter notebook.

**Exercise 6.2: nonlinear manifold learning with UMAP.**
The directory `imgs/` contains a large number of small PNG images of a sandal.[1] Each filename is of the form '`sandal_e{x}_a{y}.png`' where `x` and `y` are integers that specify the *elevation* and *azimuth* angles that were used in the rendering of the mesh.[2] The size of the images is 100 by 100 pixels. The images were generated by varying two underlying parameters. This means, we expect that the images roughly form a two-dimensional sub-manifold of $\mathbb{R}^{100\times 100}$. In the following, we will test how well UMAP can extract this.

[1] source for the original 3d data: https://raw.githubusercontent.com/plotly/datasets/master/ply/sandal-ply.csv
[2] see https://matplotlib.org/stable/api/toolkits/mplot3d/axes3d.html for documentation on view angles

1. Import all images into Python as 2d numpy arrays and extract the corresponding values for *elevation* and *azimuth* from the filenames. What are the value ranges for *elevation* and *azimuth* in the data?
   *Hint:* The functions `glob.glob`, `imageio.imread`, and the module `re` could be useful for this.

2. Analogous to the lecture, use UMAP to obtain a three-dimensional embedding of the images. For this, treat all images as flattened vectors of length 10,000. Plot the embedding coordinates in a scatter plot and additionally encode the *elevation* and *azimuth* parameters of the images as additional visual attributes.
   *Hint:* An interactive 3d plot will not work in the submitted static PDF. Either simply include a few static images from well-chosen angles or an interactive plot in the jupyter notebook.

3. Briefly comment on which manifold structure you would expect, given how the data was generated, and how well UMAP can recover this.