# MASTER THESIS

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering at the University of Applied Sciences Technikum Wien - Degree Program Game Engineering and Simulation Technology

# Scalable Interactive Visualization of Large Curve Data

By: Eduard Wolfgang Pranz, BSc

Student Number: 1810585017

Supervisors: Dipl.-Ing. Dr. Gerd Hesina
Dr. Johanna Schmidt

Wien, September 6, 2020

FH University of Applied Sciences
TECHNIKUM WIEN

# Kurzfassung

Industrielle Prozesse erzeugen immer mehr Daten. Manche Sensoren melden ihren Status mehrmals pro Sekunde, wodurch die resultierenden Datensätze schnell große Mengen von Samples anhäufen. Um diese Daten zu analysieren wird häufig Computersoftware eingesetzt, welche die aufgezeichneten Werte in einem für Menschen lesbaren Format visualisiert. Eine der Visualisierungsmethoden ist hierbei die Darstellung in Form von 1D Kurven. Werden alle Kurven eines großen Datensatzes gleichzeitig angezeigt, so treten oft Probleme mit Renderzeiten auf; zusätzlich kann die Masse an Daten auch zu visueller Übersättigung führen, die es Menschen erschwert, Muster oder Anomalien in dem erzeugten Bild zu erkennen. Im Laufe dieser Masterarbeit werden Probleme mit visueller Übersättigung, Renderperformanz und Clusterbildung analysiert und moderne Lösungen vorgeschlagen. Danach wird k-means Clustering zusammen mit funktionalen Boxplots, einer Methode zur Reduktion visueller Übersättigung, in eine Softwareplattform für visuelle Datenanalyse namens Visplore integriert. Beispielcodes, erzeugte Bilder und Zeitmessungen werden erklärt, so wie auch Algorithmen und deren Verhalten in diversen Situationen. Ergebnisse zeigen, dass die Reduktion visueller Übersättigung schnelleres Rendern ermöglicht, aber hohe Berechnungszeiten verursacht. Durch die Kombination von Übersättigungsreduktion mit Clustering können die Berechnungszeiten wieder reduziert werden.

**Schlagworte:** visuelle Datenanalyse, Reduktion visueller Übersättigung, Renderperformanz, Clustering

# Abstract

Industrial processes generate ever growing amounts of information. With some sensors reporting their state in sub-second intervals, data sets quickly reach large numbers. Computer software is commonly used to analyze and visualize this information in a human-readable format. One specific problem area is the efficient representation of data in the form of 1D curves. Drawing all curves on screen at once is not always optimal, because too much information not only impacts render performance, but also makes it hard for users to find anomalies or trends within a visually overloaded image. In the course of this thesis, problems and solutions regarding visual clutter are explored and possibilities to form clusters of curves are presented. After an analysis of novel methods, k-means clustering and functional boxplots for visual clutter reduction are combined and integrated into the visual analytics software platform Visplore. Code snippets, example figures and performance measurements are used to compare program behavior in diverse situations. Results show that visual clutter reduction speeds up render times, at the cost of significantly increased computation times. However, if clustering is done in combination with clutter reduction, the heightened computation cost can again be mitigated.

**Keywords:** visual analytics, visual clutter reduction, render performance, clustering

# Contents

# 1 Introduction

## 1.1 Motivation

Visual clutter is an effect that occurs, when a large amount of information is placed into a single image. According to (Rosenholtz et al., 2007), an image is cluttered, when the amount of information, or the presentation thereof leads to reduced human performance when interacting with it.

As shown in an experiment by Phillips & Noyes (1982), where certain tasks had to be performed on a topographic map with different features superimposed, the best results came from maps that were less cluttered.

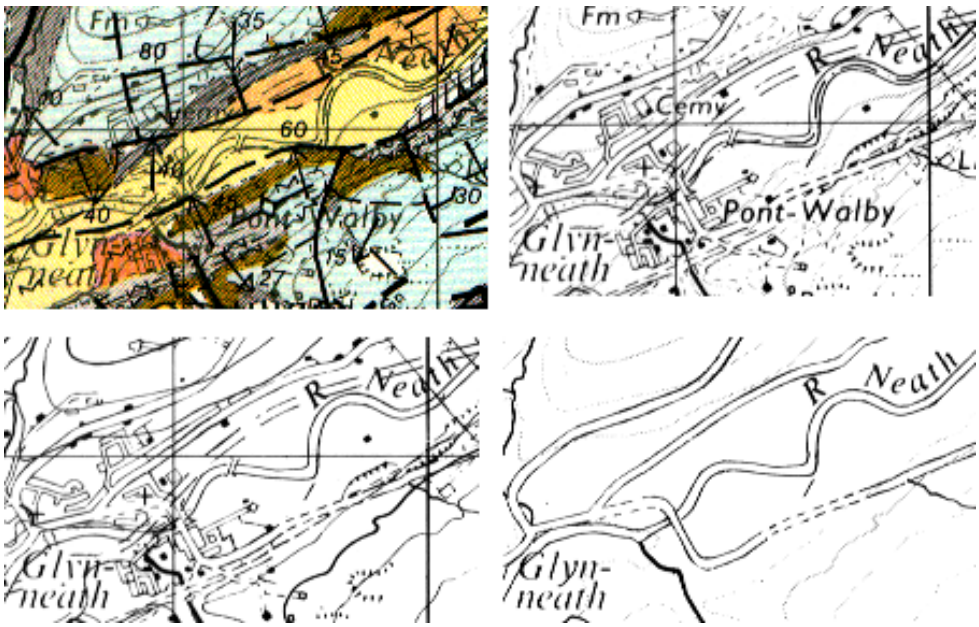


Figure 1: Excerpt from a map with different amounts of superimposed information (Phillips & Noyes, 1982).

Another example of clutter would be in user interface design, where toolbars are too extensive for finding the desired functionality in a timely fashion. A lack of organization or hierarchies within UI are also detrimental for meaningful interactions.

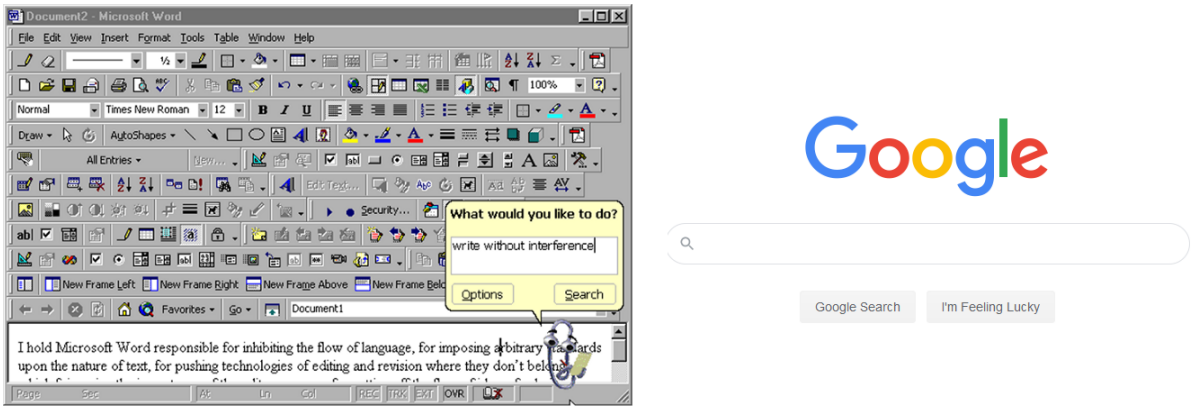


Figure 2: Left: Overloaded UI in Microsoft Word (Babich, 2017). Right: A positive example - Google Search

In the above examples, clutter can be controlled by reducing the amount of information on maps, or by removing unneeded toolbars and menu items in text editors and browsers.

During industrial production, multiple sensors collect great amounts of data to monitor variations to the expected behavior. Process engineers, who are trying to find anomalies or deviations within the sampled values, are tasked to correlate the found differences in search of potential issues that may have caused a problem during production.

A common way to represent sampled data is to connect each measurement with lines to form curves. With multiple curves rendered to the same image, overplotting occurs. At a certain point the amount of overplotting can reach a level, where the image appears cluttered, as seen in Figure 3.

Plotting too many curves into the same output image can become detrimental to analysis, since individual observations are difficult to differentiate. This makes visual clutter a problem that needs to be addressed for efficient visual data analysis.

Furthermore, with higher amounts of curves being rendered, delays become more and more noticeable. While modern GPUs can handle hundreds of curves in the fraction of a second, many computers in the industry do not have the luxury of hardware acceleration, which leads to significant waiting times.

With a large amount of curves rendered to the screen, a further problem arises. After all, data analysts want to look at the curves in a way to spot trends, curve densities, anomalies and outliers, to name only a few. Visual analysis gets harder with a larger data set, since features will be occluded, and outliers will be hidden within hundreds of graphs.

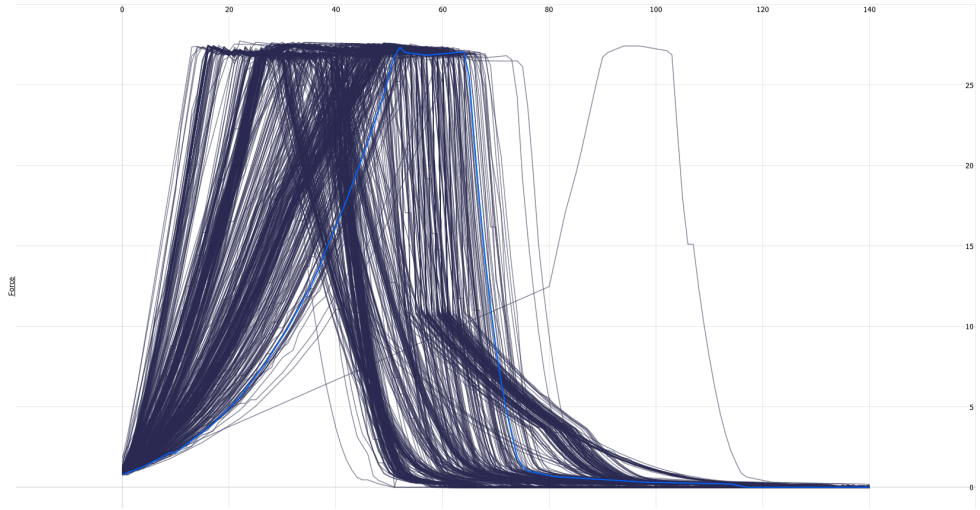


Figure 3: Sample data set ‘Anonymous Production’ with approx. 500 curves, plotting force (y) against time (x) rendered in Visplore prior to visual clutter reduction algorithms.

Similar to the way clutter is reduced in maps or applications, - by only keeping the most relevant information - a number of solutions for functional data and curves are available. Functional boxplots (Mirzargar et al., 2014), just like bagplots & HDR plots (Hyndman & Shang, 2010) provide information about a median or modal curve, the central region of similar curves, and outliers. Out of these options, functional boxplots use a powerful technique called ‘functional depth’ to determine data centrality and percentile information. However, all of these methods always consider the entire data set as a whole, but are unable to deal with multiple subgroups that might be present in the data.

Through the application of k-means clustering, subgroups of curves can be quickly detected, which leads to an overall very effective visual clutter reduction method. With multiple functional boxplots on screen, clutter can be removed, while extracting the most relevant pieces of information out of the original data set.

## 1.2 Outline

After the introduction, the problem area is outlined in Chapter 2. Of interest are state-of-the-art methods that are dealing with visual clutter reduction and clustering. After discussing potential use-cases in the industry, a short explanation of functional data follows. Towards that chapter’s end, the work environment section gets more specific, since the broad concepts need to be applied to Visplore, a software platform for visual analytics.

In Chapter 3, implementations of the selected algorithms are performed. Visual clutter reduction with functional boxplots and k-means clustering are implemented and integrated into Visplore. Specific adaptations and speed-ups are explained and linked to a few example code snippets.

Results are then presented in Chapter 4, where the effects of different parameters and inputs are analyzed. While visual quality is quite subjective, measured execution times give an overview of the algorithms' scalability and performance in various conditions.

Discussion takes place in Chapter 5, where the found results are evaluated and problems and their solutions that came up in the course of development are described, closing off with the final conclusion and an outlook into future work.

## 1.3 Task Description

The aim of this thesis is to improve the visual analytics software platform Visplore through implementation of a modern visual clutter reduction method suitable for functional data.

Visplore has a curve view at its disposal that operates well with smaller data sets, but has increased render times for larger amounts of curves. The resulting plots end up to be quite cluttered, as seen in Figure 3.

Functional boxplots are a battle-tested method to reduce visual clutter, but we have created an expansion of the algorithm by first clustering data into subgroups, which appears to be an untested method so far. While functional boxplots are able to distill a large data set down to its most relevant features, the granularity is sometimes insufficient for multi-modal data. Therefore, overplotting a number of functional boxplots aid in discovery and individual assessment of each mode.

New insights are generated, stemming from experiments that combine clustering with visual clutter reduction. Impact on performance and visual output is analyzed, and visual clutter reduction (VCR) is coupled with k-means clustering, so that both algorithms work hand-in-hand within Visplore. The resulting software must scale well, since industrial data sets consisting of hundreds of curves with hundreds of sample points each need to be supported. This is why many of the algorithm's variables are parameterized, so users may adjust them based on their needs.

Due to the importance of the application's interactivity, and its resulting time-critical nature, visual clutter reduction and clustering algorithms are implemented in C++. The most performance-hungry sections of the resulting plugin need to be specifically addressed, and some trade-offs between performance and accuracy are unavoidable.

Additionally, an experiment regarding progressive visual analytics (PVA) was conducted, with the goal to enhance interactivity by supplying a number of previews that meant to reduce waiting

times. The PVA experiment only showed partial success, which will be expanded on in the course of this thesis.

# 2 Problem Area

## 2.1 State of Research

### 2.1.1 Visual Clutter Reduction

**Definitions of visual clutter**

Rosenholtz et al. (2007) used a real-life analogy to describe clutter:

*When trying to gauge the level of visual clutter, one could compare it to a set of circumstances, where a co-worker should be notified, by leaving behind a note on their desk. The note should be placed as such that it will stand out enough for the colleague to take action upon it. In case the co-worker's desk is uncluttered, the task of finding a spot for the note to stand out can be performed with ease. It is highly probable that the co-worker will see the note. However, if the desk is cluttered, it is unlikely that the note can be placed in a way that it would call to action, therefore, one might decide that the better option would be to place the note onto the uncluttered chair next to the desk, instead of leaving it directly on the desk.*

Figure 4: How to (not) place a note visibly.

They also describe clutter as a condition, where the way in which items are presented or sorted, or a surplus thereof, leads to downgraded performance of humans dealing with these items.

Doyon-Poulin et al. (2012) describe a number of main factors contributing to clutter in a similar way. The amount of information present increases clutter just as the lack of relevancy within information or bad organization thereof.