Computer Graphics & Visualisation

2019 – 2020 Project

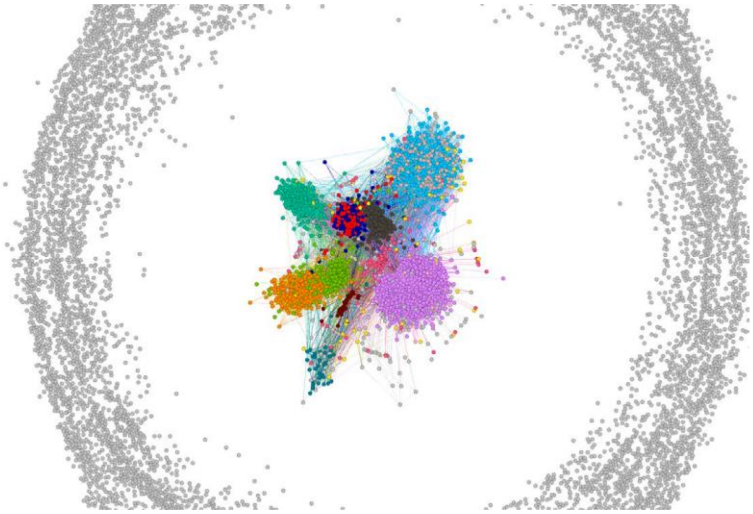


Figure 1 - An example of using graph and OpenGL to visualise a big dataset on customer reviews from Yelp

1. Introduction

In this project, we will create an information visualisation system using OpenGL. We will use a dataset collected in the domain of bibliometrics, especially scientometrics, to measure, analyse and visualise the scientific outcomes results through the analysis of books, articles, and other publications. In order to succeed in this project, you will learn to read and pre-process various data types of the dataset; to design method(s) to represent and communicate the data; to implement an interactive OpenGL software to visualise the data; and to follow a research question and find answers to it from the data.

For more information on the scientific publications in the field of bibliometrics, you can familiarise yourself with few concepts of scientific publishing, academic peer review and scientific literature at https://en.wikipedia.org/wiki/Scientific_literature.

There are many existing works have been published in visualising and analysing bibliometric dataset. Few examples can be found in the Zip file of the dataset, which could inspire you to design and develop your own system.

2. Project Dataset

For this project, we will use a small subset of the DBLP computer science bibliography (https://dblp.uni-trier.de/) which provides a comprehensive list of research papers in major computer science journals and proceedings. As of today (02/2020), there are near 5 million publications and 2.5 million authors listed in the full DBLP dataset. They are extracted from 5850 conference proceedings and 1669 journals. The sheer amount of data collected in this full dataset begs for visualisation and analytics methods that go beyond simple platforms and conventional visualisation approaches to embrace a systematic understanding of large sets of publications in connection to co-authorship network and evolution of research domains overtime amongst others.

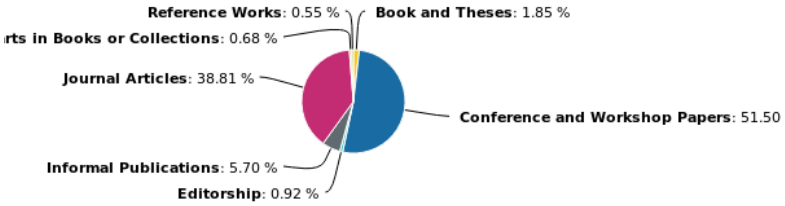


Figure 2 - Distribution of publication type of the DBLP dataset

The subset of the DBLP provided in this project has been pre-processed to facilitate your focus on the design and implementation of a visualisation module using OpenGL. In the dblp.v11.csv file, each line represents a paper whose schema is:

Field Name	Field Type	Description	Example
id	string	paperID	10087658
title	string	paper title	keynote lecture fur and hair practical modeling and rendering techniques
authors	string	author names in the format of name:id, separated by semicolons	anton krupkin:2462621991;sergey zhukov:2488761331
year	int	published year	2015
n_citation	int	citation number	5
fos	string	paper fields of study in the format of name:weight (string:float), separated by semicolons	data mining:0.444738626;data visualization:0.5979673

The text in the dblp.v11.csv file has been "standardised" using different methods of reformatting each word by removing punctuations, escape sequences and converting the characters into lower case. If it happens that there is no id of the author, a random number will be created to replace it. Two main domains of 'computer graphics' and 'data visualization' have been picked to be used as the keywords in searching the fields of study of the whole dataset. As a result, we have 63501 papers listed in the csv file which contain either 'computer graphics' or 'data visualization' or both in their fos field. In addition, a metadata fos.csv file is also included, which lists the name and number of occurrences of each field of study from the dblp.v11.csv field.

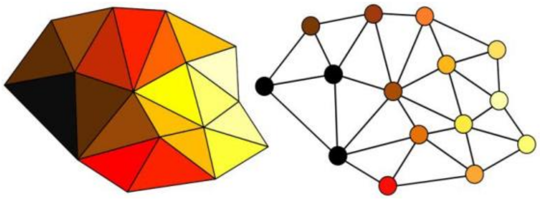
3. Project Design and Implementation

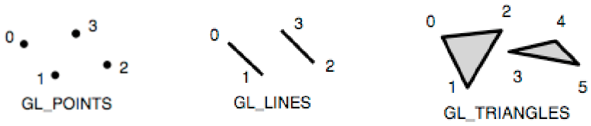
Your task is to build a system that would help to summarise the whole structure of the dataset. A graph-based visualisation (ref. Figure 1) is recommended to visualise the dataset but other creative visualisation techniques and metaphors are also encouraged. This project can be tackled as follows:

Data extraction and preparation: analyse the dataset and find a way to extract and represent data in a certain format. For instance, if the data is to be represented as a graph (directed or undirected), what do nodes, edges represent and how to calculate a local clustering coefficient (as a measure of the degrees to which nodes tend to cluster together). Few examples in preparing graphs can be found in the following list:

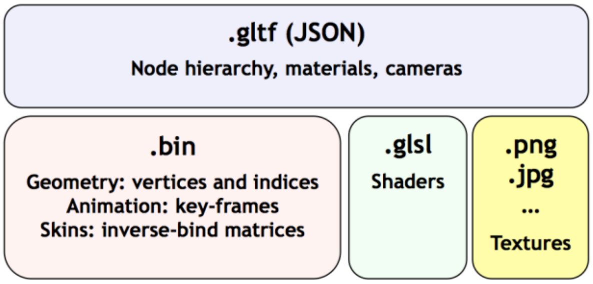
- METIS: Heuristic but works really well in practice
  http://glaros.dtc.umn.edu/gkhome/views/metis
- Louvain: Based on Modularity optimization
  http://perso.uclouvain.be/vincent.blondel/research/louvain.html
- Spectral: Based on eigenvectors of Laplacian matrix
  https://people.eecs.berkeley.edu/~demmel/cs267/lecture20/lecture20.html
- Graclus: Based on kernel k-means
  http://www.cs.utexas.edu/users/dml/Software/graclus.html
- Clique percolation method: For finding overlapping clusters
  https://cran.r-project.org/web/packages/CliquePercolation/vignettes/CliquePercolation.html

Data processing and storage: data from the original dataset needs to be converted into a new format (e.g., graph-based format of nodes and edges). This new format then needs to be transformed into a set of meshes with vertices and indices in OpenGL. The data in the new format can be stored as text, binary files or in a database.





For the small dataset, text format can be used to simplified the process. Otherwise, for big data files, glTF (GL Transmission Format) can be used to store 3D scenes and models using the JSON standard. A structure of using glTF for graph storage can be found as follows:


.gltf (JSON)
Node hierarchy, materials, cameras
.bin
Geometry: vertices and indices
Animation: key-frames
Skins: inverse-bind matrices
.glsl
Shaders
.png
.jpg
...
Textures

Visualisation and rendering: the visualisation module has to allow users to observe the dataset from different viewpoints. It is necessary then to implement some basic interactions via mouse and/or keyboard in order to change, for instance, modes of visualisation, different data layouts, camera position. A datapoint selection possibility via mouse and/or keyboard to visualise the details of datapoints will be bonus in this project. Lighting and textures are also an important part of this project in order to create visual effects to render the data visualisation more interestingly.

4. Project Evaluation and Submission

Design (30%)

A short report of maximum 2 pages will be submitted by Saturday 8th February before midnight at 23:59, which details the selected data pre-processing and data representation layouts. You can use hand drawings to illustrate your design if needs be.

Project (70%)

Presentation (25%)

The demonstration and presentation will take place during the last session on the 14th February 2020 starting from 8:30 am. You will have maximum 10 minutes for the presentation and demonstration and 2 minutes for questions. Be positive about your project but do not exaggerate your results.

It is recommended in your presentation and demonstration to include:

- 30 seconds to pitch your project: state your objective and what and/or who is your tool for
- a short explanation of what the tool provides
- a short demo of the possible interactions in the tool
- a short demo of some interesting points you found in the data and/or answers to a research question.

Report (25%)

You will need to submit a report, which is due two weeks after the presentation on the 28th February 2020 before midnight at 23:59. You can use French or English as your written language. It is possible to improve your program during this time but the new version has to be consistent with the presented one during the presentation session (it means you cannot change the whole program 100%).

An example of the report's structure can be as follows:

1. Introduction
2. Objectives
3. Design and Implementation
4. Results
5. Evaluation and Analyse
6. Conclusion and future work of how to improve the program
7. Annexe: Instructions of how to install and launch your source code (if necessary)

The prototyping and implementation criterion (25%) as well as the adaptation (25%) will be extracted from the report of how the project has been evolved from the prototyping to presentation and reporting. An assessment criteria rubric will be communicated later.

Submission

1. In a .zip file, include the OpenGL program with all libraries if necessary to compile the project, and several screenshots of your program.
2. Your report in .pdf format.

These files will be sent to the email: thi-thuong-huyen.nguyen@u-psud.fr with the subject: G&V-Project. If the submission files are too big to be transferred over the email, a downloadable link is to provide in the email. If you have any questions, please send me an email or post them on Slack and I will try to answer them as soon as possible.