A visualization of biodiversity through time

Master Thesis

December 29, 2021

by Wuxing Dong, 18-946-038

Supervisors:
Prof. Dr. Renato Pajarola
Dr. Gaudenz Halter

Visualization and MultiMedia Lab
Department of Informatics
University of Zürich

University of Zurich[UZH]

VISUALIZATIONANDMULTIMEDIALAB

Abstract

Abstract heading is similar to main section without numbering. Abstract text is similar to main body text format. Avoid referencing bibliography in abstract text.

Lore feum quisciliqui ting eugait ullandignim zzrit iriustrud doluptat volum il il iustin utet, sum dolore tat volobor autpat alisim quipis nit iure vendrerit eugait ing et ad magnim amconse min ulla corper in heniat accum am dipit lutatuero od minismo orercipis ad tie veraesectet ipit lum incil dolor augiam dolorpe ciliquis at. Aute duisim zzriusto elit illut nismodo uptat, quis am veliquisi. Lor sequis augait lam vel del ullan velis nulputet utat dit nonsed tionsequat, sum amet ulputpat. Ut ad magna facilis amet iuscipsum velenissis amet prat.

Contents

1 Introduction

In this scientific, technical report skeleton we outline the structure and formatting rules and guidelines to follow for a written project, Bachelor or Master thesis report. In Chapter 2 we describe the basic structure and sections of such a report, and in Chapters 3 and 4 we give guidelines for a consistent typesetting and formatting of the text.

2 Report Structure

A clear organization of a report typically encloses the inner main technical part(s) by an introduction and related work section at the beginning, as well as a discussion and/or conclusion section at the end. The inner parts must include the precise problem statement, a detailed description of the technical solution (e.g. mathematical models, data structures and algorithms), implementation details if applicable, and experimental results. A typical outline of a scientific report can look like this:

- Introduction
  - Visualization of biodiversity data for general public
  - Current gap in biodiversity visualization
  - Aim of the thesis
  - Thesis structure
- Current biodiversity visualization and problems
- Problem statement
  - Need tree data
  - Need fossil data
  - Need map data and transfer the location
- Technical solution
  - Data processing stage
    - building of tree from PBDB data
    - building of tree from Wiki data
    - reconstructing fossil point positions
  - Web development stage
    - Data queries for tree, and for fossil points and locations
    - Visualization of tree, computation of arc angles
- Implementation
  - Data processing stage stacks
  - Web development stage stacks
- Results
  - Data analysis
    - PBDB tree data
    - PBDB fossil data
    - Wiki species tree data
  - Data visualization
    - Time selector component
      - Stand-alone functionality
      - Interactive functionality with map and tree
    - Map component
      - Stand-alone functionality
      - Interactive functionality with tree

    - Tree component
        - Stand-alone functionality
        - Interactive functionality with map
- Conclusion and discussion

Introduction The introduction is a crucial part of your report and sets the stage as well as motivates the work you have completed. It briefly introduces the problem, emphasizes its importance and shows to the reader what you did in your work. That is, you show the reader that your problem is fundamental and that it needs to be solved as it fills a gap in the known literature and prior work.

The introduction starts by addressing the problem in general and points the reader in the direction how you want to solve your problem / research question. As a metaphor, John Swales developed a three-stage model for research introductions:

Move 1: Establish a territory (claim centrality of topic)

Move 2: Establish a niche (indicate gap)

move 3: Occupy the niche (outline purpose and indicate research structure/methods).

You may follow that model while writing the introduction. In the first part of the introduction or in a separate section, you are expected to mention and organize supporting literature where you outline the *state of the art* of your research question.

At the end of the introduction you may include a short summary of the structure of the rest of the paper. The introduction together with the conclusion should give a complete short version of your work and what you have achieved.

Related work After the introduction there follows the related work section. Try to classify the related methods in the literature with respect to the problems and solutions in your work. Establish the position of your work with respect to those competing on the same turf. Summarize the core contributions of the most important related prior solutions within the context of your work: if different from yours, clearly indicate the limitations that your solution improves upon; if similar to yours, state its advantage over other methods and thus indirectly justify your method. At the end, it should be clear why a new solution is required to solve the same problem and how your method is different from other related work.

You should know your literature well in order to find the missing link, or the most important unsolved problem, understand your contribution to the field, place your contributions in the midst of other works, and to sell your work.

Select references carefully and organize them into related groups. Within one topic you can organize references from old (first seminal) paper to newest achievements, and if too many related solutions exist, restrict to the last few most important papers if necessary.

Problem statement The technical sections introduce and motivate the proposed solution in the light of the problem, which requires a precise problem statement together with any assumptions and requirements. Where algorithmic or mathematical descriptions are not appropriate, other technical and implementation problems can be stated that are to be solved in this work.

Technical solution One or more sections should be directed towards the detailed description of the proposed solution, including technical details about the used data structures, algorithms and mathematical methods. The technical description should allow the resourceful and interested reader to reproduce and verify your work, together with the implementation information given in a later section.

This is the most important part of the report that should answer every little technical question that arises in the reader's mind. Your algorithm might be a puzzle with many pieces which are described in a linear order in the report – and many such orders may be possible. A good order to describe the different components of your solution is one that allows to clearly explain one component after another based exclusively on what the reader has already seen in any previous sections, thus minimizing any forward references.

This core part of the report should be organized into coherent subsections, giving an overview and introducing formalism first. Following an overview of the necessary steps of the entire method, each step can then be described elaborately in each subsection.

Note that implementation details should be avoided as much as possible, and the focus should be on the formal and algorithmic solution; the implementation section is specifically targeted to explain programming details.

Implementation The implementation section focuses on the programming problems and details such as the organization of the source code, the dependencies of the different modules etc.

Experimental results The evaluation discusses the proposed and competing solutions in the light of the initially stated problem requirements and limitations. This typically involves some sort of experimental evaluation which leads to some type of qualitative or quantitative results.

Quantitative results include observed numbers indicating performance timings (speed) or accuracy measures of the given implementation and test datasets. If possible, statistical tests and analysis should be given, or where applicable formal proofs. Meaningful and informative numerical results must be complete and unambiguous. Explain in detail how the evaluation has been designed, as well as the experimental setup and test cases. This includes accurate description of the test data (type, properties, size etc.) as well as the test setup (e.g. view settings, screen resolution etc.) and the measured variables (frame rate, throughput, accuracy etc.).

Qualitative results may be reported if clear quantitative measures are not feasible or applicable. Qualitative results clearly show the features and functionality of the completed work, indicating if and how they are novel or different from prior work. Qualitative results are especially suitable if something *new* has been achieved that no-one has done before in the same way.

Essentially, the goal of the experimental results is to convince the reader by numbers, tests and images (and maybe user studies), giving some sort of proof why the proposed solution is good, different and/or better than other solutions.

Conclusion and discussion Puts the results in perspective, discussing it in relationship to other related work. Indicate possible (side-)effects and eventual limitations due to the evaluation. State the *take home message* of the paper that the reader should remember and provide an outlook on possible future work that extends the given solution or fixes specific limitations. Close with a brief description (that is different from the Abstract) of the proposed solution.

Summarize your main findings in one or maximum two pages. Try to keep yourself short and clear. Give a short discussion about your results where you focus on what your findings mean. E.g., show how your results and interpretations agree with the original question and with other published work or if there are any possible practical applications for your work. At the end, give hints on further improvements or development directions / areas.

3 Formatting and Typsetting

The main text area should have a margin of 2cm on both sides, and the main text should have a top and bottom margin of about 3cm each.

The main body text is typeset in 11pt Times font (serif font family). Text body is one-column, justified and single-spaced. First line of a paragraph is generally indented with about 0.8cm, however, first paragraph after any section heading may also be non-indented.

4 Chapter Title

4.1 Section Title

4.1.1 Subsection Title

4.2 Formatting

Section headings have nested numbering and are typeset in sans-serif bold font type, e.g. Helvetica or Arial. Sizes range from 18pt for main sections down to at least one point larger than the main body text size. Place some vertical space before and after section headings.

List elements are indented and formated similar to main body text. Leave up to one line of space before and after any numbered or bulleted list.

1. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper
2. Nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit
3. Dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent dolore eu feugiat nulla facilisis at vero eros et accumsan

4.2.1 Figures and Cross-References

Figures are centered and *placed in the text flow always after* they are referenced by the text and move with the text. If formatting constraints are difficult, figures are placed at the top of the next possible new page. Figures are continually numbered.

Cross-references to numbered items such as sections and figures are capitalized as in the following examples. In Section 4.2 we provide a brief summary of text formatting instruction and Figure 4.1 shows our group logo. A Figure can be divided into subfigures as illustrated in Figure 4.2.

Figure 4.1: Example figure with caption text.

(a) vmml logo

(b) uzh logo

Figure 4.2: Example subfigure with caption referencing (a) the VMML and (b) the UZH logos.

4.2.2 Mathematical Expressions

Mathematical expressions should follow a consistent set of rules and formatting as indicated in our `math.tex` style file. A few guidelines that are useful are given below, use them if possible (but adjust as necessary to make formulas clear).

- use (lower-case) italic letters for normal (scalar) variables such as x or t

- prefer i and j as index variables and m and n to denote number of elements or iterations
- use for example other letters such as a, b and c for constants
- use lower-case bold italic to denote vectors such as $\boldsymbol{u}$ or $\boldsymbol{v}$
- use upper-case bold italic letters such as $\mathbf{M}$ or $\mathbf{N}$ for sets and matrices
- use upper- or lower-case italic letters such as $f()$ or $G()$ to denote functions
- use regular plain font and decimal point to denote explicit constants such as 2, 100.12

Mathematical equations that are important or are referenced should be laid out as a separate special-equation paragraph with consecutive numbering to the right as below:

$$E = m \cdot c^2 \tag{4.1}$$

4.2.3 Pseudo Code

```
nlevels = log2(I) + 1
for l = nlevels : -1 : 1
  loop over all 2x2x2 blocks (i,j,k)
    A_l(i,j,k) = average(block)
  end
end
```

4.2.4 FIXME

You can use the command `\FIXME{}` in order to mark sections in the text, which need to be edited and fixed. E.g. **FIX**: check reference XY.

5 Pages and Bibliography

5.1 Cover and Abstract

Title of thesis, author, affiliation, date and any other administrative information should be placed on a separate cover page (see first page).

Main title of work should be typeset in large sans-serif bold font type. Title is to be followed by author and affiliation. At the bottom, separated group affiliation is set.

Abstract follows on separate page after cover page and before the table-of-content page(s).

5.2 TOC

The table-of-content (TOC) contains all Section and (Sub-)Sub Section headings and their corresponding pages and is listed on separate pages before the first section. It has its own heading typeset as a section heading without numbering.

5.3 Bibliography References

The bibliography with list of references is placed on a new page at the end of the document, titled "References" or "Bibliography" and typeset similar to the main section headings but without numbering. Bibliography entries should follow standard IEEE or ACM proceedings or transaction journal formatting styles, and be referenced by last name abbreviation and year such as [Paj07] and [GSSP10], or by number as for [1].

5.3.1 Formatting References

Bibliography entries should be clean, accurate and consistent. Thus for the entries in the bibliography follow the following fundamental rules:

- Have complete entries, including full author names, full venue name (conference or journal), pages and year, and with volume, issue and month for journal articles.
- Use consistent venue description (name the same conference/journal in the same way).
- Use compact *inproceedings* booktitle for conferences without year, e.g. "Proceedings IEEE Visualization" instead of "8th Int. Conference on Visualization (VIS'08), IEEE 2008".
- Do not use separate organization field if clear from the venue title, and usually no publisher, month and address for conferences.
- Keep entries concise, avoid unnecessary or duplicate information such as conference location or duplicated association/organization data etc.
- Use capitalized titles in the (BibTeX) references database.

Example bad BibTeX entry:

```
@ inproceedings{as78439729asf,
  Title = {Tile-based LOD for the Parallel Age},
  Author = {Niski, Krzysztof and Cohen, J.˜D.},
  Booktitle = {Proceedings IEEE Visualization Conference (IEEE Vis'10)},
  Volume = {13},
  Pages = {1352},
  Organization = {IEEE},
  Series = {IEEE TVCG journal},
  Publisher = {Computer Society Press},
}
```

Resulting in a not very nice reference:

[NC] Krzysztof Niski and J.. Cohen. Tile-based lod for the parallel age. In *Proceedings IEEE Visualization Conference (IEEE Vis1710)*, volume1713 of *IEEE TVCG journal*, page 1352. IEEE, Computer Society Press.

Corresponding clean and consistent BibTeX entry:

```
@article{NC:07,
  Title = {Tile-based {LOD} for the Parallel Age},
  Author = {Niski, Krzysztof and Cohen, Jonathan~D.},
  Journal = {IEEE Transactions on Visualization and Computer Graphics},
  Month = {November/December},
  Volume = {13},
  Number = {6},
  Pages = {1352--1359},
  Year = {2007}
}
```

Resulting in a nice reference:

[NC07] Krzysztof Niski and Jonathan. Cohen. Tile-based LOD for the parallel age. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1352171359, November/December 2007.

Bibliography

[GSSP10] Prashant Goswami, Philipp Schlegel, Barbara Solenthaler, and Renato Pajarola. Interactive SPH simulation and rendering on the GPU. In *Proceedings ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 55–64, 2010.

[Paj07] Renato Pajarola. Efficient data structures. In Markus H. Gross and Hanspeter Pfister, editors, *Point-Based Graphics*, Series in Computer Graphics, pages 148–165. Morgan Kaufmann Publishers, 2007.