UNIVERSITÄT OSNABRÜCK

Faculty of
Cognitive Science

Bachelor Thesis

Romeo: Scripting Environment with interactive Visualizations

Author: Simon Danisch
sdanisch@email.de

Supervisor: Prof. Dr.-Ing. Elke Pulvermüller

Co-Reader: Apl. Prof. Dr. Kai-Christoph Hamborg

Filing Date: 01.02.2014

I Abstract

This bachelor thesis is about writing a simple scripting environment for scientific computing, with focus on visualizations and interaction. Focus on visualization means, that every variable can be inspected and visualized at runtime, ranging from a textual representation to complex 3D scenes. Interaction is achieved by offering simple GUI elements for all parts of the program and the visualizations. All libraries are implemented in Julia and modern OpenGL, to offer top notch performance, opening the world to scientists which work which large datasets. Julia is a novel high-level programming language for scientific computing, promising to match C speed.

II Table of Contents

III List of Figures

IV List of Tables

V Listing-Verzeichnis

VI List of Abbreviations

1 Introduction

2 Introduction

In this Bachelor thesis I will first talk about this, than that aftewards a little bit about other stuff. And such... Pretty intense!

2.1 Motivation and Problem Description

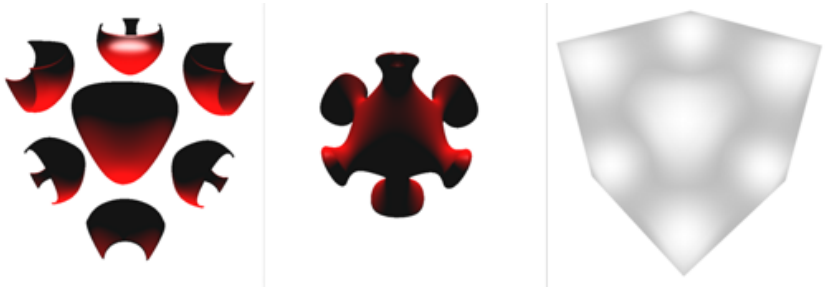

Figure 1: different visualizations of f(x,y,z)=sin(x/15)+sin(y/15)+sin(z/15), visualized with Romeo. From left to right: Isosurface, isovalue=0.76, Isosurface, isovalue=0.37, maximum value projection

When one tries to visualize a volume described by f(x,y,z)=sin(x/15f0)+sin(y/15f0)+sin(z/15f0), the limitations of the human mind becomes clear. While a computer doesn't have the creativity of humans yet, the task of visualizing this volume is a no-brainer for a computer (see Figure 1). In science, there are many comparable problems, which are unsolvable without the aid of a computer. For example making sense of one petabyte of data, predicting the folding of a novel protein, simulating black holes and many more. This naturally leads to computers being an essential part of state of the art research. Researchers have unique demands for their tools, which makes writing software for scientific computing an interesting research field. Consider, that researchers mostly don't have a background in computer science, some problems need a lot of computational power and due to the novelty of some problems hand tailored solutions are needed. In addition, high interactivity is a crucial feature, since a lot of research is a creative process which involves exploring many alternative solutions. This bachelor thesis is about writing a scripting environment with advanced visualization capabilities, while keeping the previous mentioned demands of scientific computing in mind. The resulting demands for the scripting environment are as follows.

No background in computer science puts a strong constraint on the used programming language. It should be high level and easy to learn as researchers should not be forced to waste their time with fixing memory corruptions or obscure language constructs. The entire suite should be open source and written in one language, to allow the researchers to

modify every part easily, enabling them to create a solutions which is hand tailored to their particular problem. The two constraints of writing even the core libraries in one language and having also research areas with high demand for computational power means, that in addition to being high-level the chosen language must also be fast. As an example, the high level language Python can be substantially slower than a high-performance language like Fortran. In extreme cases, Python is 1155 times slower than Fortran [**?**], which means if a computation needs 1 day with Fortran, it could take 3 years within Python, rendering the problem intractable within Python.

Finally, the scripting environment should offer simple GUI elements like sliders, to enable the researcher a simple way to interact with the parameters in his script.

These considerations lead to the 3 main design choices: Julia, a novel scientific computing language, is chosen as the main language, Modern OpenGL for high performance graphic rendering and writing generic,Reusable open source code for a high amount of customizability. The feature set consists of parametrized scientific visualizations, simple GUIs for the parameters and a simple script editor.

3 Background

4 Design

5 Results and Analysis

6 Conclusion

6.1 Lessons Learned

6.2 Related Work

6.3 Future Work

Anhang

A GUI

Ein toller Anhang.

Screenshot

Unterkategorie, die nicht im Inhaltsverzeichnis auftaucht.

Erklärung

Hiermit versichere ich, dass ich meine Abschlussarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Datum: .....

(Unterschrift)