MORGAN & CLAYPOOL PUBLISHERS

# An Introduction to Verification of Visualization Techniques

Tiago Etiene
Robert M. Kirby
Cláudio T. Silva

SYNTHESIS LECTURES ON VISUAL COMPUTING
COMPUTER GRAPHICS, ANIMATION, COMPUTATIONAL PHOTOGRAPHY AND IMAGING

Brian A. Barsky, *Series Editor*

# An Introduction to Verification of Visualization Techniques

# Synthesis Lectures on Visual Computing

## Computer Graphics, Animation, Computational Photography, and Imaging

Editor
**Brian A. Barsky**, ***University of California, Berkeley***

This series presents lectures on research and development in visual computing for an audience of professional developers, researchers and advanced students. Topics of interest include computational photography, animation, visualization, special effects, game design, image techniques, computational geometry, modeling, rendering, and others of interest to the visual computing system developer or researcher.

An Introduction to Verification of Visualization Techniques
Tiago Etiene, Robert M. Kirby, and Cláudio T. Silva
2015

Finite Element Method Simulation of 3D Deformable Solids
Eftychios Sifakis and Jernej Barbič
2015

Virtual Crowds: Steps Toward Behavioral Realism
Mubbasir Kapadia, Nuria Pelechano, Jan Allbeck, and Norm Badler
2015

Efficient Quadrature Rules for Illumination Integrals: From Quasi Monte Carlo to Bayesian Monte Carlo
Ricardo Marques, Christian Bouville, Luís Paulo Santos, and Kadi Bouatouch
2015

Numerical Methods for Linear Complementarity Problems in Physics-Based Animation
Sarah Niebe and Kenny Erleben
2015

Mathematical Basics of Motion and Deformation in Computer Graphics
Ken Anjyo and Hiroyuki Ochiai
2014

Mathematical Tools for Shape Analysis and Description
Silvia Biasotti, Bianca Falcidieno, Daniela Giorgi, and Michela Spagnuolo
2014

Information Theory Tools for Image Processing
Miquel Feixas, Anton Bardera, Jaume Rigau, Qing Xu, and Mateu Sbert
2014

Gazing at Games: An Introduction to Eye Tracking Control
Veronica Sundstedt
2012

Rethinking Quaternions
Ron Goldman
2010

Information Theory Tools for Computer Graphics
Mateu Sbert, Miquel Feixas, Jaume Rigau, Miguel Chover, and Ivan Viola
2009

Introductory Tiling Theory for Computer Graphics
Craig S.Kaplan
2009

Practical Global Illumination with Irradiance Caching
Jaroslav Krivanek and Pascal Gautron
2009

Wang Tiles in Computer Graphics
Ares Lagae
2009

Virtual Crowds: Methods, Simulation, and Control
Nuria Pelechano, Jan M. Allbeck, and Norman I. Badler
2008

Interactive Shape Design
Marie-Paule Cani, Takeo Igarashi, and Geoff Wyvill
2008

Real-Time Massive Model Rendering
Sung-eui Yoon, Enrico Gobbetti, David Kasik, and Dinesh Manocha
2008

High Dynamic Range Video
Karol Myszkowski, Rafal Mantiuk, and Grzegorz Krawczyk
2008

GPU-Based Techniques for Global Illumination Effects
László Szirmay-Kalos, László Szécsi, and Mateu Sbert
2008

High Dynamic Range Image Reconstruction
Asla M. Sá, Paulo Cezar Carvalho, and Luiz Velho
2008

High Fidelity Haptic Rendering
Miguel A. Otaduy and Ming C. Lin
2006

A Blossoming Development of Splines
Stephen Mann
2006

An Introduction to Verification of Visualization Techniques
Tiago Etiene, Robert M. Kirby, and Cláudio T. Silva

www.morganclaypool.com

ISBN: 9781627058339 paperback
ISBN: 9781627058346 ebook

DOI 10.2200/S00679ED1V01Y201511CGR022

A Publication in the Morgan & Claypool Publishers series
*SYNTHESIS LECTURES ON VISUAL COMPUTING: COMPUTER GRAPHICS, ANIMATION, COMPUTATIONAL PHOTOGRAPHY, AND IMAGING*

Lecture #22
Series Editor: Brian A. Barsky, *University of California, Berkeley*
Series ISSN
Print 2469-4215 Electronic 2469-4223

# An Introduction to Verification of Visualization Techniques

Tiago Etiene
Modelo Inc.

Robert M. Kirby
University of Utah

Cláudio T. Silva
New York University

*SYNTHESIS LECTURES ON VISUAL COMPUTING: COMPUTER GRAPHICS, ANIMATION, COMPUTATIONAL PHOTOGRAPHY, AND IMAGING #22*

MORGAN & CLAYPOOL PUBLISHERS

# ABSTRACT

As we increase our reliance on computer-generated information, often using it as part of our decision-making process, we must devise tools to assess the correctness of that information. Consider, for example, software embedded on vehicles, used for simulating aircraft performance, or used in medical imaging. In those cases, software correctness is of paramount importance as there's little room for error. Software verification is one of the tools available to attain such goals. Verification is a well known and widely studied subfield of computer science and computational science and the goal is to help us increase confidence in the software implementation by verifying that the software does what it is supposed to do.

The goal of this book is to introduce the reader to software verification in the context of visualization. In the same way we became more dependent on commercial software, we have also increased our reliance on visualization software. The reason is simple: visualization is the lens through which users can understand complex data, and as such it must be verified. The explosion in our ability to amass data requires tools not only to store and analyze data, but also to visualize it.

This book is comprised of six chapters. After an introduction to the goals of the book, we present a brief description of both worlds of visualization (Chapter 2) and verification (Chapter 3). We then proceed to illustrate the main steps of the verification pipeline for visualization algorithms. We focus on two classic volume visualization techniques, namely, Isosurface Extraction (Chapter 4) and Direct Volume Rendering (Chapter 5). We explain how to verify implementations of those techniques and report the latest results in the field of verification of visualization techniques. The last chapter concludes the book and highlights new research topics for the future.

# KEYWORDS

visualization, verification, isosurfaces, volume rendering, geometry processing, verifiable visualization

# Contents

# Preface

The term *verification* has become ubiquitous in both the computer science and engineering communities as denoting a process that somehow convinces the user that verified tools, whether those be circuits, algorithms, implementations, etc. are more safe, accurate, or complete than other tools that have not been verified. Although the term verification has a common root usage within both communities, it has evolved to mean something specific to each subarea of computer science and of engineering. For instance, within computer science, the verification of a circuit denotes either the exhaustive testing or proof that under all possible inputs, the circuit will produce the correct (specified) outputs. Similarly, for software, verification relates how well an implementation represents the behavior of its specification under all possible inputs. Within the engineering world, verification takes on a different, more nuanced meaning. One assumes that there exists an "exact solution" or "exact representation" resulting from the solution of a mathematical system of equations. In all but the most trivial circumstances, this exact solution is not attainable, and approximate solutions must be formed. The process of quantifying how well a numerical scheme or representation approximates the exact solution is referred to as verification. Verification may involve looking at how well (or quickly) an approximate solution converges (in an appropriate norm) to the exact solution, or may involve identifying features or invariants of the solution that should be maintained regardless of the approximate representation. As visualization models, algorithms and implementations lie at the interface of these two branches, what does it mean to produce *verifiable visualizations*?

This question motivated the research work that has become the foundation of this book. To answer such a broad question, we started as most researchers would: by examining a concrete example in which our ideas could be refined. We started with isosurface extraction. Many tests and a few software bugs (which our process found) later, we realized that not only were our results worth communicating to the community, but that there was much work still to do. We moved to verifying different techniques used within the visualization community—in turn learning new things along the way. We began to appreciate that verification is a process, and that articulating the guiding principles of that process was itself a contribution to our community. The various papers we reference outline the specific contributions of our work. This book is meant to make that work accessible to the general reader in a pedagogical way. We hope the reader will take away not just a particular technique, but a way of approaching and testing visualization algorithms and their implementations. In the end, we hope that all successful visualization techniques will produce verifiable visualizations.

Any work of this size and scope has benefitted by many people both indirectly and directly. We wish to thank our collaborators that helped to shape this work, in particular Luis

Gustavo Nonato, Carlos Eduardo Scheidegger, Julien Tierny, Thomas J. Peters, Valerio Pascucci, Daniel Jönsson, Timo Ropinski, João Luiz Dihl Comba, Anders Ynnerman, Lis Custódio, and Sinésio Pesco. We also thank the various faculty and students at the SCI Institute (University of Utah) with whom we sharpened our ideas. In addition, we would like to thank the various Federal Funding Agencies that have supported our research efforts over the years. The papers we reference which are co-authored by us detail those acknowledgements. Lastly, we would like to thank our spouses, without whose patience and encouragement we would probably not have made it this far.

Tiago Etiene, Robert M. Kirby, and Cláudio T. Silva
December 2015

## CHAPTER 1

# Introduction

The scientific method, as introduced by Aristotle, was formulated around the idea of postulating a model (or ideal in the Platonic sense) of natural phenomenon, making observations to validate one's model, and correcting the model based upon discrepancies between the phenomena and nature. Sir Francis Bacon is attributed with extending this process to include the idea of the controlled experiment. No longer were the scientists limited to passively observing the world around them to deduce the correctness of the model. This gave rise to the idea of devising controlled experiments designed to evaluate the correctness of the hypothesis in a systematic manner. This systematic process allowed the model to evolve based upon the lessons learned through the experiment. The late Microsoft researcher Dr. Jim Gray argued in [51] that since Bacon, there have been four paradigms of scientific discovery: experimental science, theoretical science, computational science, and data science. The first two of these paradigms reigned from the time of Bacon through the early part of the 20th century. Since the advent of computing, the latter two paradigms have risen to prominence.

With the advent of modern computing, the first of the two paradigms, called simulation science, has emerged. In this paradigm, the *experiment* now employed within the scientific method consists of the computational solution of the model. The scientific method underlying simulation science is composed of the following stages.

- **Scientific Problem of Interest (*Problem Identification*).** Statement of the scientific or engineering problem of interest. Questions should be developed in such a way that quantifiable metrics for determining the level of success of the simulation science endeavor can be evaluated.

- **Modeling.** The development of a model that abstracts the salient features of the problem of interest in such a way that exploration and evaluation of the model allows an answer to the questions specified concerning the problem of interest. Modeling techniques include, but are not limited to, deterministic or probabilistic, discrete or continuous mathematical models. Means of validating the model (determining the error introduced due to the model abstraction of the real phenomenon) should be established.

- **Computation.** The generation of algorithms and implementations that accurately and efficiently evaluate the model over the range of data needed to answer the questions of interest. This simulation of the physical phenomenon by computational expression of the model provides the experiment upon which the simulation scientific method hinges.

- **Evaluation.** The distillation and evaluation of the data produced through computational simulation to answer the questions of interest and to provide quantifiable determination of the success of the experiment. Methods such as scientific visualization provide a means of tying the simulation results to the problem of interest.

The use of simulation science as a means of scientific inquiry is increasing at a tremendous rate. It is now used in a diversity of fields such as aircraft and automobile design, climate modeling, and drug design. The process of mathematically modeling physical phenomena, experimentally estimating important key modeling parameters, numerically approximating the solution of the mathematical model, and computationally solving the resulting algorithm has inundated the scientific and engineering worlds, allowing for rapid advances in our modern understanding and utilization of the world around us. But at the end of the day, how do we know that our computational results are "right?" That is to say, how do you know that they should be trusted, or to what level should they be trusted?

This book was motivated by the use of visualization as a means of evaluation in the third paradigm, but it is relevant to both the simulation science paradigm and the data science paradigm. Visualization is often employed as part of the simulation science pipeline. It is the lens through which scientists often examine their data for deriving new science, and the lens used to view modeling and discretization interactions within their simulations. In [24], we proposed that visualization itself must be explicitly considered with similar scrutiny as other parts of the pipeline if it is to be used as part of this scientific process.

Examining the pipeline itself and understanding what procedures need to be in place to help ensure that its results are meaningful and reliable falls under the general heading of "V&V" – Validation and Verification—within the simulation science community. We define validation and verification as follows (definitions are taken directly from [2]).

- ***Validation.*** The process of determining if a mathematical model of a physical event represents the actual physical event with sufficient accuracy.

- ***Verification.*** The process of determining if a computational model obtained by discretizing a mathematical model of a physical event and the code implementing the computational model can be used to represent the mathematical model of the event with sufficient accuracy.

Based upon these two definitions, it is easy to see that fundamental to simulation science is the idea of the "error budget"—those assumptions and approximations that introduce error (or approximations) into the simulation process and their corresponding impact (or cost) on the scientific pipeline. Quantification, and ideally elimination, of modeling errors (those errors introduced through the choice of a mathematical model to describe observable data), approximation error (those errors introduced in the numerical computation of solutions of the model), and uncertainty errors (those errors due to variation in model parameters) are critical components of the scientific process. They allow scientists to judiciously evaluate which component of the process described above (e.g., modeling, numerical approximations) requires refinement in comparison with the real phenomenon of interest. Over the last 40 years, tremendous effort has been exerted in the pursuit of numerical methods that are both *flexible* and *accurate*, hence providing sufficient fidelity to be employed in the numerical solution of a large number of models and sufficient quantification of accuracy to allow researchers to focus their attention on model refinement and uncertainty quantification. It is in light of this that the verification process currently used in simulation science has been solidified; it is a means of "proving the mettle" of the computational and mathematical model [2].

The verification process is commonly partitioned into two areas recognizable to most visualization researchers: solution verification and code verification. In solution verification, effort is directed toward assuring the accuracy of the input data, estimating the numerical approximation error due to discretization, and assuring the accuracy of the resulting simulation output data. In code verification, effort is directed toward finding and removing source-code mistakes and finding and removing (numerical) algorithmic errors. When these two forms of "debugging" are accomplished, they allow researchers not only to correct and refine their scientific tools, but also to build a confidence in the design and handling of the scientific tool and the corresponding results it produces.

When these results are then to be used in the scientific setting, differences between computational and experimental results can be examined in light of the assumptions that were employed in the model generation and simulation. If visualization is the lens through which simulation scientists view their data, is that lens free of flaws? Is it possible that visual discrepancies between simulation and experimental results could be due to assumptions and approximations built into the visualization method? Are the visualization techniques designed based upon (and, in particular, to respect) properties of the model and the simulation used to generate the data being visualized? To place visualization firmly within the scientific process, it must undergo the same level of rigorous analysis.

## VERIFIABLE VISUALIZATION

Data visualization has become an indispensable means of presenting data due to its ability to succinctly summarize and support ideas and concepts that are being examined or presented. A basic premise of visualization is that visual information can be processed at a much higher rate than raw numbers and text. As the cliché goes, "A picture is worth a thousand words" [23]. Visualization techniques and systems [22, 43, 44, 48, 55] have thus emerged as a key enabling technology in this endeavor: helping people explore and explain data by allowing the creation of both static and interactive visual representations.

Visualizations libraries such as Kitware's VTK contain a very large number of highly complex visualization algorithms with thousand of lines of code implementing them. The most powerful of these algorithms are often based on complex mathematical concepts, e.g., Morse-Smale complex [9], spectral analysis [45], and partial differential equations (PDEs) [3]. Robust implementations of these techniques require the use of nontrivial techniques (e.g., simulation of simplicity [10], linear systems solvers [6], and Delaunay meshing [50]). The overall complexity and size of these datasets leave no room for inefficient code, thus making their implementation even more complex. On top of all this, hardware keeps changing quickly and many platforms need to be supported. In particular, the use of GPUs just adds to the overall complexity. *Given all this complexity, an important question that must be asked is whether the derived visualizations are correct—both mathematically and perceptually.*

As we become more reliant on computational algorithms and systems in our day-to-day lives, there is an increasing need to develop metrics by which we can attest to the "quality" of the hardware and software components that we employ. Good design specifications are not enough as many stages of development exist between the conceptual design phase and the finished product. Furthermore, system complexity has been increasing rapidly, making it easy for "bugs" to creep inside even the most carefully designed and implemented codes.

The issue of guaranteeing correctness of complex systems has been studied in different contexts and it continues to be an active area of research [17, 18, 26]. In computer science, such considerations have proved to be important in areas such as circuit and software design. In the context of engineering, such considerations are important in the modeling and simulation of physical phenomena. Although the specific processes used in these two areas can vary significantly, they have at their core a common root paradigm, that of *validation* and *verification*.

Despite the fact that visualizations are widely used, the problem of verifying visualization algorithms and techniques has been largely overlooked [16, 20, 24, 54, 60]. Although there are ad hoc solutions for testing implementations, no technique provides a commonly accepted framework for verifying the (mathematical) accuracy, reliability, and robustness of visualization tools.

As mentioned earlier, this is distinct from, but intimately related to, questions of perception and visual representation efficacy or correctness. In fact, there has been substantial anecdotal evidence of visualization techniques whose flaws caused the misinterpretation of the underlying phenomena. Some researchers have even argued that the problem is so acute that users should avoid third-party visualization tools due to their concern about potentially incorrect results [24].

But what does it mean to produce *verifiable* visualizations? This book presents our efforts in trying to formally define a process. We start the book with a brief introduction to visualization in Chapter 2. This is followed by a brief description of validation and verification in simulation science in Chapter 3. Through a simple example, we illustrate the main steps necessary to implement a V&V pipeline. The next two chapters form the technical core of the book where we undertake a formal verification study of the correctness of isosurfacing and volume rendering techniques. In Chapter 4, we introduce the tools necessary for the verification of geometrical properties of isosurface extraction algorithms. In Chapter 5, we introduce the principles of verification of volume rendering algorithms. The final chapter, Chapter 6, provides concluding remarks and highlights new research topics for the future.

# CHAPTER 2

# Visualization in the Real World

In this chapter, we briefly introduce the field of "visualization" and explain the goal of this book—to introduce techniques capable of answering the question "How do I know that the visualization I see is correct?" We will focus on the importance of visualization and why it should be verified. We start by introducing the many flavors of visualization, followed by a brief history and applications. Then, we explain the typical pipeline used in scientific visualization, covering the process of data acquisition, filtering, and mapping. Next, we explain how different errors can affect that pipeline. We list some of the many error sources that hinder visualizations and present an historical account of the pursuit of the correctness of a well-known visualization *technique*. Last, we introduce some of the current practices within the visualization community and demonstrate the need for more tools for verifying visualizations.

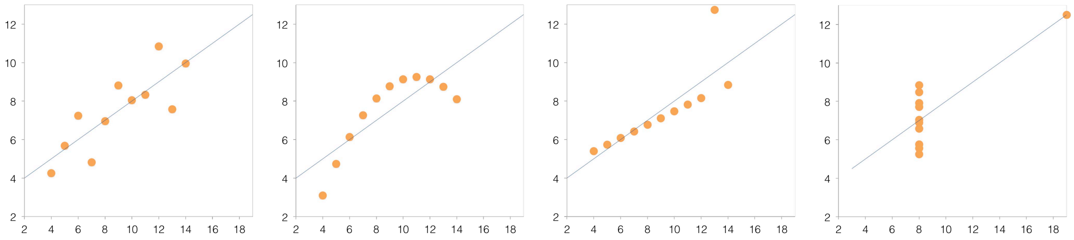
## 2.1 VISUALIZING DATA

We live in the age of data, an age defined by the use of data to augment our capacity to understand and solve real-world challenges. In medicine, for instance, data helps medical diagnosis. In business, customer data is a rich source of information about customer tendencies and needs. For the individual, data can provide insights into one's health via sensors that measure weight and blood pressure. These are only a small fraction of the applications that benefit from understanding data. Nevertheless, data serves no purpose if it cannot be analyzed. As our capacity to amass and store data grows, so does the need to analyze, explore, extract meaning, and present that data to empower its users. The multitude of sources—the census, weather, medicine, satellites, numerical simulations, wearables, to name a few—adds to the problem. As the goal is to learn as much information from the data as possible, a combination of statistics, computer software, and data visualization is essential to allow us to gain insights from data.

Even though this combination of tools is taken for granted nowadays, it is not straightforward to realize that "data visualization" should be part of this tool set. In fact, there was a time when statistical graphics did not enjoy the prestige that it does today, as it was thought of as just a means "for showing the obvious to the ignorant" [52], or not as rigorous as numerical calculations. By visualizing data, the user is exposed to features that may be hard to understand otherwise. The Anscombe's quartet is a classic example advocating the need to visualize data [1]. The quartet is composed of four distinct sets of $(x, y)$ pairs, whose mean, variance, linear regression, and other metrics are nearly identical (see Table 2.1). In other words, by these measures, the datasets are also nearly identical. By using a simple plot, however, one can clearly see that the opposite is true

**Table 2.1:** The Anscombe's quartet. Each dataset possesses nearly identical mean, variance, linear regression, and other metrics. By these metrics, the four datasets are also nearly identical.

| Dataset I | | Dataset II | | Dataset III | | Dataset IV | |
|---|---|---|---|---|---|---|---|
| *x* | *y* | *x* | *y* | *x* | *y* | *x* | *y* |
| 10 | 8.04 | 10 | 9.14 | 10 | 7.46 | 8 | 6.58 |
| 8 | 6.95 | 8 | 8.14 | 8 | 6.77 | 8 | 5.76 |
| 13 | 7.58 | 13 | 8.74 | 13 | 12.74 | 8 | 7.71 |
| 9 | 8.81 | 9 | 8.77 | 9 | 7.11 | 8 | 8.84 |
| 11 | 8.33 | 11 | 9.26 | 11 | 7.81 | 8 | 8.47 |
| 14 | 9.96 | 14 | 8.1 | 14 | 8.84 | 8 | 7.04 |
| 6 | 7.24 | 6 | 6.13 | 6 | 6.08 | 8 | 5.25 |
| 4 | 4.26 | 4 | 3.1 | 4 | 5.39 | 19 | 12.5 |
| 12 | 10.84 | 12 | 9.13 | 12 | 8.15 | 8 | 5.56 |
| 7 | 4.82 | 7 | 7.26 | 7 | 6.42 | 8 | 7.91 |
| 5 | 5.68 | 5 | 4.74 | 5 | 5.73 | 8 | 6.89 |



(a) Dataset I (b) Dataset II (c) Dataset III (d) Dataset IV

**Figure 2.1:** Scatterplot of the Anscombe's quartet shown in Table 2.1. The line in the figure is the linear regression of each dataset.

(see Figure 2.1). By visualizing the data, the user is able to make better decisions regarding, for instance, the best mathematical model for adjusting the data (Figure 2.1 is not well represented by a linear model) to detect outliers, clusters, and potentially other information not easily accessible from the data table and other statistical summaries. As the cliché goes, "a picture is worth a thousand words," or "a thousand numbers."

### 2.1.1 PRECURSORS OF MODERN VISUALIZATIONS

Of course, Anscombe's quartet is not the first successful display of quantitative data. Although the field of data visualization is relatively new, the use of images to depict information is much older.