Available online at www.sciencedirect.com
ScienceDirect
Journal of Visual Languages and Computing
18 (2007) 613–616

Journal of
Visual Languages
& Computing

www.elsevier.com/locate/jvlc

Book review

P. Fishwick (Ed.) Aesthetic Computing, MIT Press, Cambridge, 2006, ISBN 0-262-06250-X.

This book is part of the MIT Press Leonardo book series, whose mission is "to publish texts by artists, scientists, researchers, and scholars that present innovative discourse on the convergence of art, science, and technology".

Fishwick in the book's Preface speaks of the authors in the book seeking the possibilities of art contributing to computing, as a groundbreaking attempt. He defines Aesthetic Computing as the *application of aesthetics to computing*, and thus the book emphasizes how aesthetics affects computing. The book was contributed by:

- Researchers in computer science, particularly in human–computer interaction, James Alty, Olav Bertelsen, Stephan Diehl, Paul Fishwick, Carsten Görg, Frederic Fol Leymarie, Frieder Nake, Ray Paton, Aaron Quigley, and Paul Vickers, Information systems researchers, Noam Tractinsky, and Dror Zmiri;
- Media art and design researchers, most of whom are artists themselves: Jay David Bolter, Donna Cox, Monika Fleischmann, Ben Fry, Susanne Grabowski, Diane Gromala, John Lee, Jonas Löwgren, Lautent Mignonneau, Jane Prophet, Christa Sommerer, and Wolfgang Strauss;
- Agent technology researcher, Mark d'Inverno; mathematics researchers Michele Emmer, Michael Leyton who is also a renowned fine artist; independent fine artists Kenneth Huff; astronomer and space scientist Roger Malina.

The book is divided into four sections: philosophy and representation, art and design, mathematics and computing, and interface and interaction. I will, however, categorize the chapters based on my own understanding of the topics they cover, into *What is aesthetic computing*, *visualization*, and *arts, design and interaction*. I will focus more on the chapters that offer fundamental explanations and arguments than on those that introduce specific systems and techniques.

What is aesthetic computing?

As the most important chapter of the book, Chapter 1 is Fishwick's introduction to aesthetic computing, the exciting new area of study, closely related to art, design, mathematics, and computing. He attempts to define the terms of aesthetics, art, and computing. In the subsequent text, the terms "aesthetics" and "art" are used interchangeably (but distinct according to Nake and Grabowski in Chapter 4), and

doi:10.1016/j.jvlc.2007.01.004

"computing" and "computer science" imply the same field of study. Therefore, aesthetic computing is defined as *the application of the theory and practice of art to the field of computing*. It involves two types of such applications: *analysis* that evaluates certain computing artifacts based on the classic aesthetic qualities such as symmetry; and *synthesis* that represents computing artifacts beautifully. I can easily relate analysis here to the objectives of symmetry and minimal crossing in graph drawing [1], and metaphoric representations of objects of interest in information visualization, such as the botanic view of tree structures [2]. Fishwick's work at the University of Florida is considered the synthetic approach in applying aesthetics to mathematical modeling.

Fishwick has noted that aesthetics may be applied to computing in three aspects: *modality, culture*, and *quality*. My own interpretation of these aspects is that quality has been playing an implicit but noticeable role as below, and modality and culture are yet to play significant roles in computing as technologies become more advanced. Cognitive scientists suggest that mental visualization has been a process in which a mathematician formulates a problem, specifies theories and derives proofs, and a programmer envisaging dancing symbols. Emmer articulates such kind of mathematic beauty in Chapter 13. Knuth is a strong and perhaps the first advocate of programming being an art. This is elaborated by Lee in Chapter 2 by interpreting Goodman's aesthetics [3] and confining the contribution of the artist to the writing of programs ("execution" in Goodman's term). These aesthetic activities may be considered to be measured by the aesthetic *quality*.

Fishwick lists three example activities:

- Representing programs and data structures with customized, culturally specific notations;
- Incorporating artistic methods in typically computing-intensive activities, such as scientific visualization;
- Improving the emotional and cultural level of interaction with the computer.

The process architecture illustrated by Fishwick explains well how aesthetics affects computing. Raw computing artifacts can be modulated by a subject/medium filter, and then applied by art theory and practice. The resulting computing artifacts through the filer and application are generally more usable, or have high usability, and considered in the domain of aesthetic computing. The usability here is achieved not only by beautiful and intuitive representations, but also by efficient data structure and smooth interaction support.

It is clear that the activities in reverse direction, such as computer creativity and digital art, are not considered part of aesthetic computing. They have the converse goal to that of aesthetic computing.

A more fundamental viewpoint on aesthetics is expressed by Leyton in Chapter 15, who argues that aesthetics is based on the principles of maximization of transfer and maximization of recoverability. Leyton defines Gestalt theory as the transferring of stimuli to form cohesive wholes. He further demonstrates that both arts and sciences are driven by the single goal of maximizing memory storage that forms the basis of transfer and recoverability. This is evidenced by the principle of object-oriented programming that aims at maximizing reusability of objects. This chapter serves a nice summary of Leyton's several stimulating books around these topics.

Aesthetic computing is reviewed by Malina in Chapter 3 in the context of the 40-year history of the Leonard journal. The activities in aesthetic computing started in July 2002 when a week-long workshop was held in Dagstuhl, Germany, participated by over 30 representatives from the art, design, computer science, and mathematics disciplines. The last day of the workshop resulted in a manifesto of aesthetic computing [4]. Apart from this workshop, there has been no other activity on this subject. As noted by Fishwick, whether a new field or simply rehashed old material, aesthetic computing is yet to be tested by time. I will be excited to see more discussions on this topic in the years to come.

Visualization

Paton in Chapter 5 proposes various metaphors and their associations with different diagrammatic concepts. Cox in Chapter 6 presents the metaphor theory, uses the term "visaphor" for digital visual metaphors, and demonstrates several examples of visaphor uses in scientific visualization. Chapter 10 by Prophet and d'Inverno discusses a collaborative project on adult stem cell research through modeling and visualization. I unfortunately failed to identify the main point of this chapter, perhaps partly due to my lack of knowledge in the nature of the project.

Reas and Fry in Chapter 11 share their ideas of teaching programming languages, such as Java and C++, through visual experience. They emphasize on the drawing and visual design aspects of programming by making programs produce various artistic images. The example programming systems introduced in the "Future Directions" section are in fact what we call visual programming languages and program visualization systems. Software visualization and its beauty are further demonstrated by Diehl and Grög in Chapter 12. Similarly, Quigley in Chapter 16 shows the beauty of information visualization through graph drawing that follows several aesthetic principles such as a minimal number of edge crossings and maximal symmetry. Program auralization using carefully mapped music motifs is described by Vickers and Alty in Chapter 17.

Arts, design, and interaction

Nake and Grabowski in Chapter 4 focus on the role of aesthetics in human–computer interfaces and owe their tie to semiotics, the theory of signs. Stimulating questions are raised such as "should we play with the interface, rather than the interface function for us?"

Chapter 7 by Fleischmann and Strauss describes the authors' work on an online archive media art called netzspannung.org, and on a mixed reality. The architectural design and navigational structure in netzspannung.org are interesting and artistic. My initial experience with browsing this site is that the response is way too slow due to the heavy use of imagery data. Huff in Chapter 8 shows some fascinating photorealistic images of artificial objects that were produced by visually encoding and rendering prime numbers, which reminds me of fractal images based on the same principle. In Chapter 9, Mignonneau and Sommerer attempt to interpret programming for multi-model user interactions as a creative process by demonstrating the dynamic and adaptive behavior of complex spaces.

Leymarie in Chapter 14 addresses how to represent 3D shapes beautifully based on the Gestalt principles. Bertelsen associates human-computer interaction design with aesthetic

computing through tertiary artifacts in Chapter 18. Bolter and Gromala in Chapter 19 advocate the role of digital art in interface design. They argue that two aesthetics, transparency and reflectivity, corresponding to the code view and interface view in computing, provide two complementary approaches to interface design. Löwgren focuses in Chapter 20 on the usability of digital designs and its potential influence on human–computer interaction. Tractinsky and Zmiri in the final chapter analyze the effect of physical skins of digital appliances on human emotion and report their findings in a usability study.

Summary

The questions raised in the Preface hopefully provoke a deep thinking of aesthetic computing as possibly a new discipline:

> to what extent can the traditional definitions of aesthetics in computing and art be interrelated and connected, with each informing the other? What roles can quality, subjectivity, and emotion play in mathematics and computing as ways to achieve a balance between form and function? What are effective social frameworks in which artists, designers, mathematicians, and computer scientists can collaborate in teams or in distributed networks?

The viewpoints expressed and many examples demonstrated in this book, some of which bear deep root in arts and sciences, are stimulating and of great interest to anyone who is interested in both computing (or mathematics) and art. I was slightly disappointed in not seeing any discussion on color and form, two essential dimensions of arts and design, for their roles in aesthetic computing, particularly in visualization and interface design. Aesthetic computing being a field of study may be premature. We have however come to a stage when we can "afford" thinking of this idea. This is because the modern computer graphics and interface technologies are so advanced that guidelines for effective yet enjoyable human–computer interactions are greatly needed.

References

[1] G. Di Battista, P. Eades, R. Tamassia, I.G. Tollis, Graph Drawing: Algorithms for the Visualization of Graphs, Prentice-Hall, Englewood Cliffs, NJ, 1999.
[2] E. Leiberg, H. van de Wetering, J.J. van Wijk, Botanical visualization of huge hierarchies, Proceedings of the IEEE Symposium on Information Visualization (InfoVis'01) (2001) 87–94.
[3] N. Goodman, Languages of Art, Bobbs-Merrill, Indianapolis, 1968.
[4] P. Fishwick, Aesthetic computing manifesto, Leonardo, vol. 36(4), MIT Press, Cambridge, 2003, pp. 255–256

Kang Zhang
Department of Computer Science,
The University of Texas at Dallas, USA
E-mail address: kzhang@utdallas.edu