Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

464

J. Dassow J. Kelemen (Eds.)

Aspects and Prospects of Theoretical Computer Science

6th International Meeting of Young Computer Scientists
Smolenice, Czechoslovakia, November 19–23, 1990
Proceedings

Springer-Verlag
Berlin Heidelberg New York London
Paris Tokyo Hong Kong Barcelona

Editors

Jürgen Dassow
Sektion Mathematik, Technische Universität Magdeburg
Postfach 124, O-3010 Magdeburg, FRG

Jozef Kelemen
Department of Artificial Intelligence, Comenius University
Mlynská dolina, 842 15 Bratislava, Czechoslovakia

CR Subject Classification (1987): D.2, F.1–4, I.2–4

ISBN 3-540-53414-8 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-53414-8 Springer-Verlag New York Berlin Heidelberg

Printed in Germany

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
2145/3140-543210 – Printed on acid-free paper

Foreword

This volume contains the text of the tutorial lecture, the texts of five invited lectures and the texts of twenty short communications contributed for presentation at the Sixth International Meeting of Young Computer Scientists, IMYCS'90, held at Smolenice Castle, Czechoslovakia, November 19-23, 1990.

The IMYCSs have been organized biennially since 1980 by the Association of the Slovak Mathematicians and Physicists in cooperation with Comenius University, Bratislava, and with other institutions. The aim of the meetings is threefold: (1) to inform on newest trends, results, and problems in theoretical computer science and related fields through a tutorial and invited lectures delivered by internationally distinguished speakers, (2) to provide a possibility for beginners in scientific work to present and discuss their results, and (3) to create an adequate opportunity for establishing first professional relations among the participants.

Short communications included in this proceedings were selected from 47 papers submitted in response to the call for papers. The selection was made on the basis of originality and relevance of presented results to theoretical computer science and related fields by the Programme Committee. The members of the Programme Committee were E. Csuhaj-Varjú (Budapest), J. Dassow (chairman, Magdeburg), S. K. Dulin (Moscow), K. P. Jantke (Leipzig), J. Karhumäki (Turku), A. Kelemenová (Bratislava), M. Křivánek (Prague), K. J. Lange (Munich), J. Sakarovitch (Paris), and M. Szijártó (Györ). The editors wish to thank all of them as well as to subreferees G. Asser, A. Brandstaedt, M. Broy, V. Diekert, C. Dimitrovici, P. Ďuriš, H. Giessmann, D. Hernandez, J. Hromkovič, J. U. Jahn, I. Kalaš, J. Kelemen, I. Korec, V. Koubek, M. Kráľová, M. Krause, F. Kröger, A. Kučera, M. Kunde, L. Kühnel, S. Lange, R. Letz, P. Mikulecký, M. Pawlowski, J. Procházka, H. Reichel, W. Reisig, G. Riedewald, E. Ružický, P. Ružička, S. Schönherr, K. Schultz, M. Tegze, E. Tiptig, J. Vyskoč, R. Walter, J. Wiedermann, R. Wiehagen, H. Wolter, T. Zeugmann, and maybe some others not mentioned here who assisted the members of the Programme Committee in evaluating the submissions.

On behalf of all the participants of IMYCS'90 we express our gratitude to the members of the organizational staff of the Meeting, especially to Peter Mikulecký for chairing the Organizing Committee.

The editors are highly indebted to all contributors for preparing their texts carefully and on time. We would like to acknowledge gratefully the support of the organizing institutions: Association of Slovak Mathematicians and Physicists, Institute of Computer Science and Department of Artificial Intelligence of the Comenius University, Bratislava, Department of Computers of the Slovak Institute of Technology, Bratislava, and the Mathematical Institute of the Slovak Academy of Sciences, Bratislava.

We highly appreciate the excellent cooperation with Springer-Verlag in the publication of this volume.

Jürgen Dassow
Jozef Kelemen

Contents

Part I
Tutorial

Methods for Generating Deterministic Fractals and Image Compression *

Karel Culik II
Dept. of Computer Science
University of South Carolina
Columbia, SC 29208, USA

Simant Dube
Dept. of Computer Science
University of South Carolina
Columbia, SC 29208, USA

Abstract

We survey recently developed methods for generating deterministic fractals that have the potential for compression of arbitrary (practical) images. They are the Iterative Function Systems developed by Barnsley, the probabilistic finite generators, and probabilistic mutually recursive systems that generalize both former methods. We briefly introduce the formal notion of an image both as a compact set (of black points) and as a measure on Borel sets (specifying greyness or colors). We describe the above mentioned systems for image generation, some mathematical properties and discuss the problem of image encoding.

1 Introduction

Recently, the fractal geometry introduced by B. Mandelbrot [19] is getting increased attention in relation to the study of deterministic chaos (complex systems) [16]. The relation of fractal geometry to classical geometry is similar to the relation of classical physics, which handles primarily phenomena described by linear differential equations, to the new "chaos" physics. "Chaos" physics studies complex phenomena, mathematically described by nonlinear differential equations, like the flow of gases. Classical geometry handles well "man-made" objects like polygons, circles, etc. The new "fractal" geometry should handle well all the classical objects as well as those of fractal (recurrent) type. Examples are H-trees, Sierpinski triangles and also all natural objects like plants, trees, clouds, mountains, etc. The study of fractal geometry was pioneered by B. Mandelbrot [19] and the study of practical "computational fractal geometry" by M. Barnsley [1]. He introduced the Iterative Function Systems (IFS) that are used to define an object (an image) as the limit (attractor) of a "chaotic process." He has used IFS to generate exclusively Deterministic fractals. Voss et al. have considered techniques to generate Random fractals [4]. Barnsley's hyperbolic IFS [1] is specified by several affine transformations, and the attractor is the limit of the sequence generated from an arbitrary starting point by randomly choosing and applying these affine transformations.

*This research was supported by the National Sciences Foundation under Grant No. CCR-8702752.

A number of various methods for generating both deterministic and random fractals have been developed, here we will concentrate on those that have potential to be used for efficient encoding of a wide variety of images. Besides Barnsley's IFS we will discuss rational expressions (probabilistic finite generators), affine expressions, (probabilistic) affine automata, and (probabilistic) mutually recursive IFS.

Barnsley's collage theorem [1] gives the mathematical basis for infering concise IFS-description of any given image, which includes texture or color. We will discuss how this can be done also for finite generators. Collage theorem can be extended also to affine automata and mutually recursive IFS, even if presently we do not have any efficient methods of encoding arbitrary images by these methods.

Encoding of images by IFS and other methods have tremendous potential for practical applications because they allow drastic compressing of data and their efficient processing. For example, from an IFS description of an image it is possible to regenerate effectively not only the original image but also its various modifications, e.g. a view from a different angle. Applications of this are studied not only to computer graphics [3] but also to compression of videos, to medical imaging, to high-resolution TV, etc.

We first discuss the formal notions of an image in Section 2. A black and white image is formalized as a compact set and texture (color) image is formalized as a normalized measure (greyness density). Then we introduce Barnsley's IFS method to generate fractals.

In [9], automata-theoretic techniques have been developed for image representation, manipulation and generation, which we describe in Section 3. A similar approach to represent patterns by finite automata has been independently taken in [5]. Our approach in [9] is more general as we have defined images in terms of languages of infinite words, rather than finite words, and we have also considered textures. Images here are sets of points in n-dimensional space (or sometimes functions on this space specifying the level of grey or color). Points are represented by coordinates, i.e. n-tuples of rational numbers. In turn, rational numbers are represented by strings of bits usually in binary notation (e.g. string 011 represents number 0.011 in binary notation). Hence an n-tuple of strings can be interpreted as a point in the n-dimensional space $< 0,1 >^n$, and a relation $\rho \subseteq \underbrace{\Sigma^\star \times \ldots \times \Sigma^\star}_{n-times}$ as a set of points, i.e. an object (image). Similarly, an ω-string is interpreted as a real number in the interval $< 0,1 >$.

It has been known for more than twenty years that, for example, the Cantor Set (as subset of $< 0,1 >$) can be represented in ternary notation by regular expression $\{0+2\}^+$ [15,17]. We can show how most "regular" 2-dimensional geometrical objects (both classical and fractal) can be represented by simple rational expressions. Unlike Barnsley's IFS the rational expressions allow to build complex images from simple ones by set and other operations. We can convert the rational expressions into probabilistic finite generators that are used to generate images much like the Barnsley's Chaos Game algorithm. We also have a method to automatically infer the probabilistic generator from an arbitrary given image. This is based on the quad tree representation of images that has already been used in Computer Graphics for compressing data [18] and computing the Hausdorff dimension of images [24]. Hence, much like Barnsley, we can concisely represent objects (both fractal and classical) and regenerate them in the original or modified version.

In addition to considering images as compact sets, we can define "texture" images, which are formalized in terms of probabilistic measures. A finite approximation of such a texture image on a computer screen will be a matrix of pixels which are assigned grey tones (or colors).

Considering the practical descriptive power, our rational expressions (probabilistic generators) are incomparable with Barnsley's IFS [9].

In Section 4, we briefly discuss a generalization of Barnsley's IFS called Affine Expressions which define a bigger class of images of more complex geometries [8]. Intuitively, an affine expression generates an image based on a finite set of affine transformations that are applied in an order controlled by a regular set.

We consider two more generalizations of IFS in Sections 5 and 6. The first one is probabilistic affine automaton (PAA), which is informally probabilistic finite generator whose input symbols are affine transformations. PAA are equivalent to recurrent IFS, as introduced in [2], under certain assumptions [8]. Fractal dimension of recurrent IFS is computed in [14]. The other one is called mutually recursive IFS (MRIFS) and is given by a number of "variables" which are defined in terms of each others as unions under affine transformations. We consider both deterministic and probabilistic variations of MRIFS. Barnsley's Collage Theorem can be generalized to PAA and MRIFS, and provides for the basis of the algorithm to automatically synthesize a PAA (or a MRIFS) for a given image.

Rational expressions are special case of affine expressions. This allows one to efficiently implement rational expressions (probabilistic finite generators) by PAA and PMRIFS. This implementation does not use the bit-by-bit approach and hence yields algorithms that using standard (numerically oriented) software and hardware are almost as fast as Barnsley's.

All these generalizations of IFS—affine expressions, PAA and MRIFS, are equivalent in their power to generate images (as compact sets).

In Section 7, we consider another approach to generate interesting images, which is based on L-systems (string rewriting systems). We give two examples illustrating that certain L-systems can be simulated by MRIFS.

Finally, we conclude the paper by briefly discussing the problem of an image encoding.

2 Preliminaries

2.1 Two Notions of an Image

Following [1] we introduce two different formalizations of an image:

(1) Given a complete metric space (X, d), an image is a compact subset of X. The quality of an approximation for such images is measured by the Hausdorff metric $h(d)$ on the complete metric space $\mathcal{K}(X)$ of the nonempty compact subsets of X. This is a formalization of such an image as consisting of black and white regions. A finite approximation of such an image on the computer screen is an assignment of 0 (white) or 1 (black) to each pixel of a matrix of pixels.

(2) Given a complete metric space (X, d), an image is a normalized invariant measure on X, that is an additive function f defined on the Borel subsets of X such that $f(X) = 1$ (see [1] for more details). The quality of an approximation for such images is measured by the Hutchinson metric d_H on the complete metric space of all the normalized measures on X. This is a formalization of an image as a texture, either of various tones of grey or