Original Research Paper

Volume : 5 | Issue : 10 | October 2016 • ISSN No 2277 - 8179 | IF : 3.508 | IC Value : 69.48

Visual Study About the Fractals and New Means of Viewing

Engineering

KEYWORDS : fractals, complex mathematical methods, generation and visualization, parallel programming.

Bogdan Popa

University of Craiova, Department of Automation and Electronics, Craiova, Romania

ABSTRACT

— Starting from complex mathematical methods for generating fractals and visualization to their usual technological means, today through iterative functions for computational modeling can have improvement in various fields like biology, astronomy, medicine or economics. Fractal generation and visualization can be a very expensive process in time, it means that by the developing of the new technologies this process must be improved. This article highlights the results of several studies about Mandelbrot and Julia sets implementations of algorithms for image processing, treating new software deployment methods, parallel programming, or different construction depending on storage space and generation technics. Also it is proposed some analysis of methods for the generation systems, visualization, calculation and comparison of performance for the implementations presented over the fractal sets.

INTRODUCTION

Nowadays there are a lot of systems that offer simple generation on fractals domains. "Fractal" is a word invented in the 19th century by Benoit Mandelbrot to refer to a large class of objects that have a historical role in the evolution of pure mathematics. An empirical and remarkable type of ideas makes differences between the classical mathematics of the 19th century and the modern mathematics of the 20th. The mathematics has the scope has, in many cases, to meet the necessity of physical reality.

This study describes how two types of famous fractals were created and explains the most important fractal properties today, which make fractals useful for different domains of science. A simple tool for the fractal generation can offer beautiful images starting just from typical, mathematical formulas on the complex plane. Also this process offers a large number of iterations and complex calculus. Benoit Mandelbrot defines, some abstract definition, that a fractal such as a set for which the Hausdorff Besicovich dimension strictly exceeds the topological dimension. Some applications of the fractal theory today combines the mathematical background with image processing and a lot of different fields for applying those methods. For example, image processing offers today as an area based on great developments in mathematics, and also in general in different engineering domains, but also viewed as an area of artificial intelligence, dealing with new representation technology , reconstruction, recognition or image analysis are done today with powerful computers or clusters.

Also the fractal theory is presented in domains for studying the chaos teory, physical phenomena, disturbances and in all the fields that contains irregular forms [1]. Another study for the utility of fractal dimension proposes algorithms beyond uniform sampling for specific problems [2] or find long-living trajectories [3].

This paper presents an important study over the implementations of visual fractals based on Mandelbrot Set and Julia Set that is represented in some applications taking into account the costs for generate fractal images and the improvement based on the new technology used today, because the rendering systems for fractal are time expensive and nowadays the time and the power of calculus are very important related to the data transfer, the business and the information. The fractals methods for imaging are used today in medicine and special in the bone X-rays study. There are many applied examples for the images and video streams obtained from the fractal generations, an overview is provided by A. Fournier, D. Fussell, and L. Carpenter in their article "Computer Rendering of Stochastic Models" appearing on pages 371-384 of Communications of the ACM, Vol. 25, No. 6, June 1982. A less cumbersome way of generating fractals, which would allow their computation in shorter time is desirable because the time it generates images is slower than for video systems. Nowadays there are a lot of techniques for speeding-up this process and for reducing the redundant calculus. It is obvious that being an iterative process it can be implemented in parallel and also can be generated in parallel because each part of calculation, each pixel is a result of different complex number calculation.

ALGORITHMS BACKGROUND – THE MANDELBROT AND JULIA SET

The Mandelbrot set M is defined by a family of complex quadratic polynomials:

$$P_c: \mathbb{C} \to \mathbb{C} \qquad (1)$$

given by:

$$P_c: z \to z^2 + c \qquad (2)$$

where $\boldsymbol{C}$ is a complex parameter. For each $\boldsymbol{C}$, one considers the behavior of the sequence

$$\left(0, P_c(0), P_{c(P_c(0))}, P_{c\left(P_{c(P_c(0))}\right)}, \ldots\right) \qquad (3)$$

obtained by iterating starting at critical point , which either escapes to infinity or stays within a disk of some finite radius [4]. The Mandelbrot set is mentioned as the set of all points c so that the above sequence does not escape to infinity.

The Julia set of a function f is commonly denoted J(*f*), and the Fatou set is denoted F(*f*).These sets are named from the French mathematicians Gaston Julia and Pierre Fatou whose work began the study of complex dynamics, during the early 20th century.

Let R(z) be a rational function

R(z)=(P(z)) / (Q(z)) (4)

where z € C*, C* is the Riemann sphere C union {infty}, and P and Q are polynomials without common divisors. The „filled-in" Julia set J is the set of points z which do

not approach infinity after R(z) is repeatedly applied (corresponding to a strange attractor). The true Julia set J is the boundary of the filled-in set (the set of „exceptional points"). There are two types of Julia sets: connected sets (Fatou set) and Cantor sets (Fatou dust). Quadratic Julia sets are generated by the quadratic mapping:

$$Z_{n+1} = Z_n^2 + c \qquad (5)$$

for fixed c. For almost every c, this transformation generates a fractal.

It can be said that the Mandelbrot set is a particular case for the Julia set because the second set offers a fraction with two different polynomials. It is obvious that the generated images are strongly influenced by the nature of the polynomials used in the description of the function form. The two sets have the same implementations for the construction of the pictures, results to the iterative process.

The region of the complex plane that is computed is subdivided into some number of pixels, related to the dimension of the picture that is proposed to be built. To color any such pixel, let c be the midpoint of that pixel. The process iterate the critical point 0 under , checking at each step whether the orbit point has modulus larger than 2.

When this is the possibility , we know that c does is not included into the Mandelbrot set or Julia set with the proposed type, and there are colored the pixel according to the number of iterations used to find out. In this moment it is known that at the time the number of iterations processed is used, when the algorithm escapes, and this number is used from a palette of colors to choose the color of the pixel, respectively defined by the starting points. If it is not possible, it is more and more iterating up to a fixed number of steps, after which it is decided that our parameter is „probably" in the Mandelbrot set or Julia proposed form set, or finally very close to it, and color this pixel in black.

VIEWING MANDELBROT AND JULIA SETS EFFICIENCY AND THE PROCESS TIME

Sequential Algorithm:

The usual sequential algorithm in generating Mandelbrot set and Julia set images shows the entire set of pixels in a rectangular area (related to the complex plane), while checking every pixel from plane (related to the values of constant c) if it is included to the Mandelbrot Set or not.

A simple example can be compressed to:

- for x=0 to width for y=0 to height count=0

- z=c do z =z^2+ c (Julia Form Set-(P(z)) / (Q(z)))

- count=count+1

- while (|z|<2.0) and (count<max)

*- if |z| ≤ 2.0 then display pixel /*Pixel Is Mandelbrot set */*

This process can be done in parallel for every iteration type, because there are similar calculus for different starting points, directly related with the pixels positions. The concept is simple, the main loop, the landline, is done in parallel for a normal execution. Mandelbrot number of equations to be solved is fixed and it is given by the image size and initial parameters, it allows parallel execution of the loop. Internal time calculation is established between the starting time of calculation for first pixel until completing the entire image generation. The same concept is used for the Julia generator because just the form of the functions is modified.

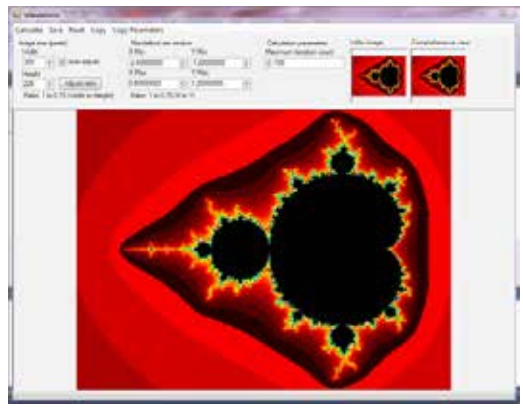

Figure 1: Sample resulting image 1 with the generating tool.

Also this tool contains two types of figures, one generated rapidly with little dimensions and another with bigger resolution. Parallel computing is a new form of computation methods where many calculations steps are done simultaneously, [4] operating on the principle that large problems can often be divided into smaller ones, such as the pixel generation method, which are then solved concurrently ("in parallel"). There are several different forms of parallel computing: bit-level, instruction level, data, and task parallelism. Parallelism has been employed for many years, mainly in high-performance computing, but interest in it has grown lately due to the physical constraints preventing frequency scaling [5]. As power consumption (and consequently heat generation) by computers has become a concern in recent years, [6] parallel computing has become the dominant paradigm in computer architecture, mainly in the form of multicore processors [7]. The tool implemented and presented in the Figure 1 generates images based on different functions related to the Julia and Mandelbrot Sets, also gives the time of the process because the time consumption is the main important factor that is compared in this article.

Today there are many studies that convert in fabulous time execution for clusters systems for the iterative generation process, also there are some similar images obtained with the natural images. The presented tool can set the maximum number of iteration to stop the process, the limit point when also the process can be stopped, the dimensions of the generated picture that it is proposed to be generated. Starting to the parallel calculus and another techniques, it is imperative to minimize the time for the generation. To improve having to do huge numbers of iterations for other points in the set, that is time redundant, one can do "periodicity checking"; which means finding if a point reached in iterating, a pixel has been reached before. If so, the pixel cannot diverge, and must be in the set. This is most relevant for fixed-point calculations, where there is a relatively high chance of such periodicity—a full floating-point (or higher-accuracy) implementation would rarely go into such a period. An interesting method for speeding-up this process is represented by Julia programming that is a high-level, high-performance dynamic programming language for technical computing, with syntax that is familiar to users of other technical computing environments [8]. It provides a sophisticated compiler, distributed parallel execution, numerical accuracy, and an extensive mathematical function library.

Volume : 5 | Issue : 10 | October 2016 • ISSN No 2277 - 8179 | IF : 3.508 | IC Value : 69.48

ANALYSIS OF RESULTS

The results are presented in Table 1. These results are strongly influenced by the processor model used, the operating capacity of the computer, the load of the processor, the image type, so there were used more complex images, generated from fractals processing. Tables above represent the processing time for compression and decompression resulted from previous images, as a simple example. This time, it is strongly influenced by factors such as image size, their quality, but there is, in conclusion, the result of a calculation based on the set of selected filters. Since the actual implementation is in C ++ it can be said that research can take into new implementations based on another technology and also based on parallel programming.

TABLE – 1
RESULTS OF THE EVALUATION IN DIFFERENT CASSES IN MILLISECONDS

Number of pixels	Mandelbrot function set time sequential	Mandelbrot function set time parallel	Julia set sequential	Julia set Parallel
25000	1528	1129	1842	1322
40000	3210	2558	7432	5621
60000	4277	3214	10821	8912
100000	8237	6324	13652	11182
120000	11870	8821	16721	13997
150000	15665	13535	20650	18271
180000	18921	15197	24023	21011
200000	21021	18821	26119	24732
240000	26531	22113	32463	30884
300000	29311	26122	35223	32611
360000	34424	30992	41432	37321
400000	43921	38713	52921	43221
500000	58904	42314	62491	50032

It must be mentioned that the Julia model set is based on the

$$(1\text{-}z^3/6)/(z - z^2/2)^2 + c \qquad (6)$$

form [9] and it consumes much time than the basic Mandelbrot model, also the best implementations results are offered by the parallel proposed algorithms in every cases. Another important factor is the number of iteration when the algorithm is stopped, in this example the number is 100, and was tested with a C++ implementation, on the Intel I7 CPU Q720 at 1.60GHz type, the 64-bit system used from Microsoft. A better approach can build a network for many processes involved in the calculus process the same time, it can be the MPI (message passing interface) method used for this kind of tests.

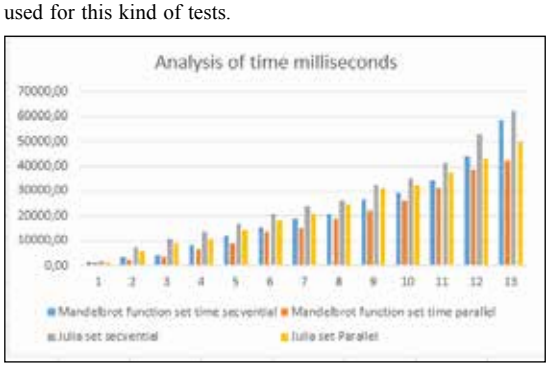

Figure 2: Analysis of time in the four cases problems

Also the number of cores, the capacity of the processor or the clustering system is important and for this case it can be proposed like a future work to test the algorithm in big computing cluster system. This time, it is strongly influenced by factors such as image size, their quality, but there is, in conclusion, the result of a calculation based on the set and types of iterative functions, the pictures can't be compared because at present, there are totally different images and palettes. It can be seen that as the number of iterations is larger, sequential computation becomes heavy and requires other solutions.

Another good example of use of the parallel processing over the Mandelbrot is represented by the study "Parallel Implementation and Analysis of Mandelbrot Set Construction" by Isaac K. Gäng, David Dobson, Jean Gourd and Dia Ali [10] which provides insight into the parallel implementation set based on the MPI technology. The proposed solution is one that takes into account the degree of calculation on several processors on MPI also a solution of high performance computing. They also utilize the Portable Batch System (PBS), a job submission facility, to submit our work on two MCSR parallel systems: Mimosa and Sweetgum. For comparison purposes, they use two different systems to develop and test the implementation. One of these systems, Mimosa, is a distributed memory system. [11]

Taking into account the case of the last example test, and the other case presented, type and capacity of each processor computing is very important and can massively influence the results.

CONCLUSIONS

The results of this study can be mentioned such as a testing and comparison over some fractal applications Sets, The Julia Set and the Mandelbrot set, with an analysis over the obtained results and to reiterate better than the parallel process in this cases. Nowadays time costs money, because it represents information transfer, working time, data transfer, databases management for figures and a lot of another fields of study.

Today, big companies like IBM and Mitsubishi invest colossal amounts of money in this study as it is seen as a future domain with a potentially major impact for the next technological world. The problems we are facing today in fractal research, are technological ones because we need space, superior processing and time to understand and asses an infinite iterative set. The capacity to generate and keep certain images and process others is important today and fractals have a large domain that can be helpful, starting to the compressing methods, natural image processing or medical. Other applications, already acknowledged by the fractal analysis include the mechanics of the aeronautical and aero spatial structures (field in which the results of some original research from the starting 90s were recalled), the dynamics of slim civil construction structures under the wind and earthquakes, the population dynamics, epidemiology, the internal dynamics and ecosystems' interaction, the analysis of the cosmic radiation, the dynamics of the stock markets, of the investments and generally the economic actions, artificial intelligence and many more. This article aims to present an interesting way to render different sets of fractals taking into account different method, starting from the parallel programing and with another speed-up methods. In conclusion the study is benefic for the understanding of the fractals' view systems, for better implementations and for the next step to the current programing and visual systems, because the mathematical form can have different representations in our lives and for the future development for growing technics domains.

In conclusion this is an interesting study about the time and memory needed for generate fractals images and an interesting analyze for this domain.

REFERENCES:

1. R.P. Taylor & J.C. Sprott (2008). "Biophilic Fractals and the Visual Journey of Organic Screen-savers" . Nonlinear Dynamics, Psychology, and Life Sciences, Vol. 12, No. 1. Society for Chaos Theory in Psychology & Life Sciences. Retrieved 1 January 2009.
2. A. P. S. de Moura and C. Grebogi (2001), Phys. Rev. Lett. 86, 2778 .
3. H. E. Nusse and J. A. Yorke V (1989), Physica D 36, 137.
4. Adrien Douady şi John H. Hubbard (1984 / 1985), "Etude dynamique des polynômes complexes", Prépublications mathémathiques d'Orsay 2/4 .
5. Gottlieb, Allan; Almasi, George S. (1989). Highly parallel computing. Redwood City, Calif.: Benjamin/Cummings.
6. S.V. Adve et al. (November 2008). "Parallel Computing Research at Illinois: The UPCRC Agenda" (PDF). Parallel@Illinois, University of Illinois at Urbana-Champaign. "The main techniques for these performance benefits – increased clock frequency and smarter but increasingly complex architectures – are now hitting the so-called power wall. The computer industry has accepted that future performance increases must largely come from increasing the number of processors (or cores) on a die, rather than making a single core go faster.".
7. Asanovic, Krste et al. (2006), "The Landscape of Parallel Computing Research: A View from Berkeley" (PDF). University of California, Berkeley. Technical Report No. UCB/EECS-2006-183. "Old [conventional wisdom]: Increasing clock frequency is the primary method of improving processor performance. New [conventional wisdom]: Increasing parallelism is the primary method of improving processor performance ... Even representatives from Intel, a company generally associated with the 'higher clock-speed is better' position, warned that traditional approaches to maximizing performance through maximizing clock speed have been pushed to their limit.", December 18.
8. http://julialang.org/, "The Julia language", Free and Open Source library
9. Gert Buschmann (2010), Field lines for an iteration of the form, example for simple Julia set exploration.
10. Isaac K. Gäng, David Dobson, Jean Gourd and Dia Ali. "Parallel Implementation and Analysis of Mandelbrot Set Construction" , School of Computing University of Southern Mississippi.
11. Ignatios Vakalis. „Mandelbrot Set: A Parallel Programming Approach", NSF Grant No. 9952806