Picturing randomness on a graphics supercomputer

by C. A. Pickover

This paper provides a light introduction to a simple graphics technique which can be used to represent random data on a graphics supercomputer. The representation, called a "noise-sphere," can be used to detect "bad" random-number generators with little training on the part of the observer. The system uses lighting and shading facilities of 3D extensions to the X-Windows or the PHIGS+ standard. To encourage reader involvement, computational recipes and suggestions for future experiments are included.

Introduction

> *"The generation of random numbers is too important to be left to chance."* — Robert Coveyou, mathematician, Oak Ridge National Laboratory

The idea that the human visual system can be used to detect trends in complicated data is not new—and neither is the application of that idea in computer graphics. What is new is the use of rich software tools and powerful new hardware to visualize random data. This paper provides a light introduction to a simple graphics technique which uses colored balls to visualize the output of random-number generators. The representation, called a "noise-sphere," can be used to detect "bad" random-number generators with little training on the part of the observer. In modern science, random-number generators have proven invaluable in simulating natural phenomena and in sampling data [1–6]. It is therefore useful to build easy-to-use graphic tools which, at a glance, help to determine whether a random-number generator being used is "bad" (i.e., nonuniform and/or with non-independence between various digits). Although a graphics supercomputer facilitates the particular representation described below, a simpler version would also be easy to implement on a personal computer; therefore, one of the objectives of this paper is to stimulate and encourage programmers, students, and teachers to explore this technique in a classroom setting.

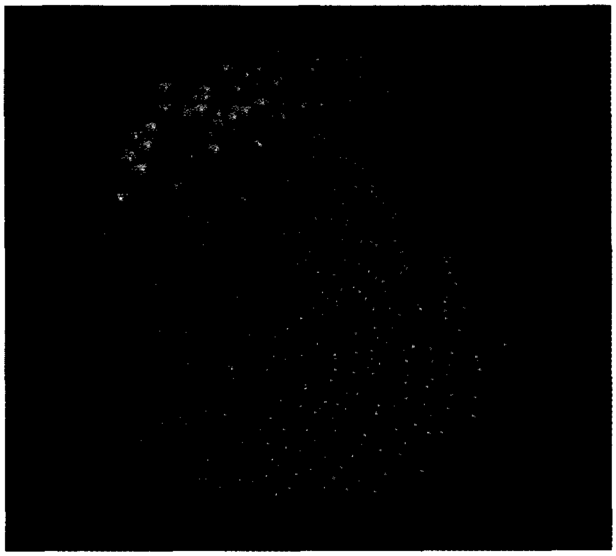

Figure 1

Noise-sphere for a bad pseudorandom generator used in a version of the BASIC language. As the ball is rotated, various tendrils appear.

Note also that the colorful representations can be considered solely as art objects, providing another example of the link between mathematics and art.[1]

Noise-sphere

Figures 1–3 are representations which I call *noise-spheres.* The various spiral projections in Figures 1 and 2 are indicative of a bad random-number generator. To produce the figures, simply map the output of a random-number generator to the position of spheres in a spherical coordinate system. This is easily achieved with a computer program using random numbers $\{X_i, i = 1, 2, 3, \cdots, N; 0 < X_i < 1\}$, where X_i, X_{i+1}, and X_{i+2} are converted to θ, ϕ, and r, respectively. For each triplet, an (r, θ, ϕ) is generated, and these coordinates position the individual spheres. The data points may be scaled to obtain the full range in spherical coordinates:

$$2\pi X_i \rightarrow \theta,$$

$$\pi X_{i+1} \rightarrow \phi,$$

$$\sqrt{X_{i+2}} \rightarrow r.$$

The square root for X_{i+2} serves to spread the ball density though the spherical space and to suppress the tight packing for small r.

The resultant pictures (Figures 1–3) can be used to represent different kinds of noise distributions or experimental data. In particular, by using this approach, "bad" random-number generators can be visually detected. Figure 3 represents a standard good random-number generator where no particular correlations in the ball positions are visually evident. I have, however, found surprising results when this approach is applied to the BASIC [7, 8] random-number generator. As the noise-sphere is rotated (Figures 1 and 2), the user can perceive various tendrils emanating from the cluster. There should be no such correlations if the distribution is truly random. The method is effective in showing that random-number generators prior to release 3.0 of BASIC have subtle problems.

Graphics

Interestingly, only a single graphic primitive was used for each figure, namely a "polysphere" (i.e., n spherical surfaces at given centers with specified radii). The polysphere primitive is one of several 3D extensions to the X-Windows[2] and PHIGS+ standard provided by the Stellar GS1000 graphics supercomputer.[3] PHIGS+, itself, is an extension to the PHIGS[4] graphics standard (ISO 9592), and it addresses the lighting and shading of three-dimensional data [9]. Today, the noise-sphere representation is used on an IBM RISC System/6000[5] processor using the GL graphics language.

Lighting of the shapes in the figures was applied on a primitive-by-primitive basis; no interactions between objects such as shadows or reflections were defined. The reflectance calculation is conceptually applied at points on the spheres being lit and shaded, and produces color at these points. Input to the reflectance calculation includes the position on the primitive at which the reflectance equation is being applied, the reflectance normal, diffuse color at that position, the set of light-source representations, and the eye point (see for example [9]). For the figures, three lights were used (red, white, and green), and the lights were rotated in real time, using a mouse, to produce the desired final

[1] This paper is number 80 in an eighty-part "Mathematics and Beauty" series which emphasizes the aesthetic side of mathematics and scientific visualization. For others in the series, see for example C. Pickover, "Overrelaxation and Chaos," *Phys. Lett. A* **30,** No. 3, 125–128 (1988); C. Pickover, "Mathematics and Beauty: Time-Discrete Phase Planes Associated with the Cyclic System, $\{\dot{x}(t) = -f(y(t)), \dot{y}(t) = f(x(t))\}$," *Comput. & Graph.* **11,** No. 2, 217–226 (1986); C. Pickover, "A Note on Chaos and Halley's Method," *Commun. ACM* **31,** No. 11, 1326–1329 (1988); C. Pickover, "A Note on Rendering Chaotic 'Repeller Distance-Towers,'" *Computers in Physics* **2,** No. 3, 75–76 (May/June 1988); C. Pickover, "Blooming Integers," *Comput. Graph. World* **10,** No. 3, 54–57 (1987); C. Pickover, "Graphics, Bifurcation, Order and Chaos," *Comput. Graph. Forum* **6,** 26–33 (1987).

[2] X-Windows is a trademark of MIT.

[3] *Stellar* and *GS1000* are trademarks of Stardent Computer Inc., 85 Wells Ave., Newton, MA 02159. The Stellar machine contains four high-speed floating-point engines configured to work separately or in tandem. The rendering processor performs special computations such as per-pixel arithmetic for hidden-surface elimination, depth-cueing, and shading. The floating-point performance is 40 MFLOPS, single or double precision. The polygon rending rate is 150K Gouraud-shaded, Z-buffered triangles per second.

[4] PHIGS (the Programmer's Hierarchical Interactive Graphics System) goes beyond the older CORE and GKS standards to provide 3D primitives such as polyline and fill area, as well as hierarchical data structures.

[5] RISC System/6000 is a trademark of International Business Machines Corporation.

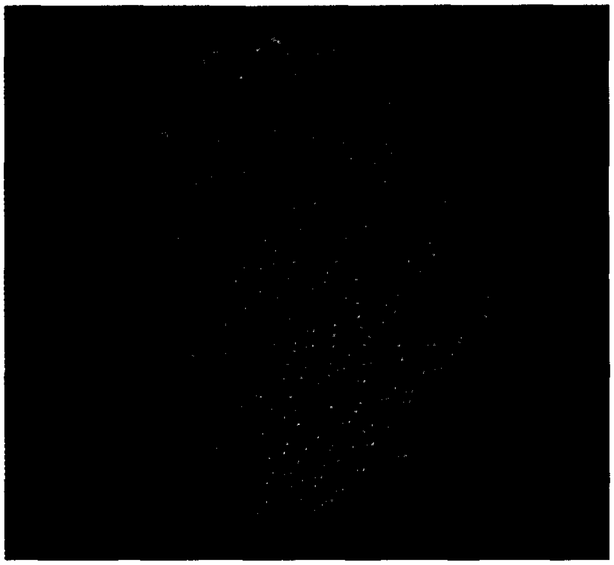

Figure 2

Same as Figure 1, viewed from another angle. By coloring the balls as a function of r, θ and ϕ, correlations become more obvious to the human analyst.

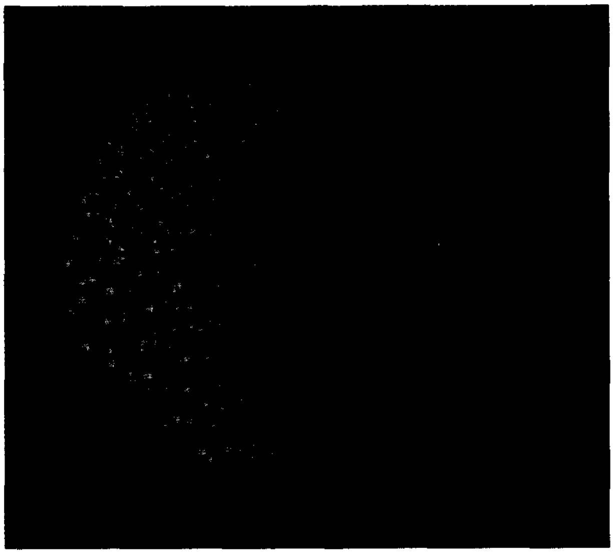

Figure 3

Noise-sphere for a good pseudorandom generator. No particular correlations are seen.

graphical effects. Specular reflections produced the highlights on the shiny surfaces. As is traditional, the intensity of specular reflections, unlike diffuse reflections, is highly dependent on the viewing angle of the observer. The specular exponent [10] used to control the shininess was about 100.

Care must be taken in the coloration of the individual component spheres within the noise-spheres to help emphasize the various spiral striations. I have found that simply mapping the random-number triplet values to red, green, and blue intensities helps the eye see correlated structures in 3D.

It is fascinating to note that the Stellar GS1000 and IBM RISC System/6000 can be used to rotate the spheres with only a few seconds' pause between images. This is useful, since correlations not visible from one viewpoint may become visually apparent when the sphere is viewed from another angle. The required computations for a similar animated sequence of spheres, with hidden-surface removal, could take many minutes on traditional high-powered mainframe computers. However, interested users without access to graphics supercomputers can render their data simply by plotting a scattering of dots and projecting them on a plane. The visual effect may be less striking without the real-time rotation, but with just a few test rotations of the noise-sphere, the user can detect a problem generator with relative computational ease. Since the noise-ball approach is sensitive to small deviations from randomness, it may have value in helping researchers find patterns in complicated data such as genetic sequences and acoustical waveforms. For example, James Ramsey (New York University) and I have used this approach to examine stock market data. Interestingly, the spherical coordinate transformation used for the noise-ball allows the user to see trends more easily, and with many fewer trial rotations, than an analogous representation which maps the random-number triplets to three orthogonal coordinates.

Reader experimentation

The reader is urged to test the approach outlined here on a variety of random-number generators. For example, consider the linear congruential generator: $s(x) = (137x + 187) \bmod 256$, which is given by Knuth as an example of a bad random-number generator [1]. Generate several thousand points, map them to the noise-sphere, and view the results.

In another experiment, consider introducing a slight Markov dependence of the data points to determine how sensitive the noise-sphere is to this kind of deviation from randomness. In a computer program, generate random numbers X_{new}, save the previous random number in variable X_{old}, and execute the following code:

GenRandom(X_{new})
if $X_{old} < X_{new}$ then $X_{new} = \max(0, X_{new} - \delta)$
else $X_{new} = \min(1, X_{new} + \delta)$
$X_{old} = X_{new}$

The parameter δ controls the dependence of the value of each new random number on the value of the preceding number. The standard, uncorrelated numbers are created when $\delta = 0$.

As a last experiment, consider a uniform distribution of random numbers between 0 and 1. Take these random numbers and round or truncate them to finite accuracy so that each is an integer multiple of $1/\nu$ for some given number ν. If this is done, the scattering of points in the noise-sphere will show a regular pattern. The noise-sphere acts as a kind of microscope, revealing the "grain" of the random numbers. The resulting figures for these and other experiments are left as a puzzle and surprise to the reader. Enjoy!

Why graphics?

Some readers may wonder why we should consider the noise-sphere approach over traditional brute-force statistical calculations. One reason is that this graphic description requires little training on the part of users, and the software is simple enough to allow users to quickly implement the basic idea (without the sophisticated lighting and shading) on a personal computer. The graphics description provides a qualitative awareness of complex data trends, and may subsequently guide the researcher in applying more traditional statistical calculations (e.g., see [1, 3]). Also, fairly detailed comparisons between sets of "random" data are useful and can be achieved by a variety of brute-force computations, but sometimes at a cost of the loss of an intuitive feeling for the structures. When one is just looking at a page of numbers, differences among the statistics of the data may obscure the similarities. The approach described here provides a way of simply summarizing comparisons between random data sets and capitalizes on the feature integration abilities of the human visual system to see trends.

Summary

A paper such as this can only be viewed as introductory; however, it is hoped that the techniques, equations, and systems will provide useful tools and stimulate future studies in the graphic characterization of morphologically rich shapes produced by simple random-number generators. For the use of tiling patterns to represent random numbers, see the American Mathematical Society paper in [4].

Acknowledgment

I thank Terry Richards for providing the BASIC random numbers and for helpful discussions.

References and note

1. D. Knuth, *The Art of Computer Programming*, Vol. 2, 2nd ed., Addison-Wesley Publishing Co., Reading, MA, 1981.
2. K. Mckean, "The Orderly Pursuit of Disorder," *Discover*, pp. 72–81 (January 1987).
3. G. Gordon, *System Simulation*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1978.
4. H. Jeffrey, "Chaos Game Representation of Gene Structure," *Nucleic Acids Res.* **18**, No. 8, 2163–2170 (1990); C. Pickover, "Mathematics and Beauty: Several Short Classroom Experiments," *Amer. Math. Soc. Notices* **38**, No. 3, 192–196 (1991); C. Pickover, *Computers, Pattern, Chaos, and Beauty*, St. Martin's Press, New York, 1990; C. Pickover, *Computers and the Imagination*, St. Martin's Press, New York, 1991.
5. S. Park and K. Miller, "Random Numbers Generators: Good Ones Are Hard to Find," *Commun. ACM* **31**, No. 10, 1192–1201 (1988).
6. J. Voelcker, "Picturing Randomness," *IEEE Spectrum* **26**, No. 8, 13 (1988).
7. *IBM Personal Computer BASIC, Version A2.10.* This is an interpreted BASIC.
8. T. Richards, "Graphical Representation of Pseudorandom Sequences," *Comput. & Graph.* **13**, No. 2, 261–262 (1989).
9. A. van Dam, PHIGS+ functional description, revision 3.0; *Computer Graph.* **22**, No. 3 (ACM SIGGRAPH) (entire volume) (July 1988).
10. J. Foley and A. van Dam, *Fundamentals of Interactive Computer Graphics*, Addison-Wesley Publishing Co., Reading, MA, 1984.

Received November 8, 1989; accepted for publication May 11, 1990

Clifford A. Pickover *IBM Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598.* Dr. Pickover received his Ph.D. in 1982 from Yale University's Department of Molecular Physics and Biochemistry. He joined IBM in October 1982 at the Thomas J. Watson Research Center as a member of a speech synthesis group. Currently he is a member of the scientific visualization group, where he has developed GALAXY, a general-purpose visualization program for the IBM RISC System/6000 processor. Dr. Pickover's primary interest is scientific visualization; he is the author of the books *Computers, Pattern, Chaos, and Beauty* (St. Martin's Press, New York, 1990) and *Computers and the Imagination* (St. Martin's Press, 1991), as well as more than 200 journal and periodical articles on topics in science, art, and mathematics. He is an associate editor for the international journal *Computers and Graphics*, a member of the editorial boards of *Computers in Physics* and *Speculations in Science and Technology*, and the editor of a number of books and topical journal issues on subjects in his field. Dr. Pickover received the first-place prize in a computer graphics competition sponsored by the American Institute of Physics, and his book *Computers, Pattern, Chaos, and Beauty* was honored as "one of the best science books of 1990" by *Library Journal.* He holds eight IBM Invention Achievement Awards and an IBM Research Division Award.