Calculating Fractal Dimension from Vector Images

Kelly Ran

TJHSST Computer Systems Laboratory

2007-2008

Abstract

Fractal dimension can be used as an index of complexity in research applications. Currently, researchers calculate fractal dimension from images in raster graphics format. This project introduces an alternative method by calculating fractal dimension from vector graphics images. **Keywords:** fractal dimension, box dimension, vector graphics

1 Introduction

1.1 Fractals

The term ”fractal” was first coined by Benoit Mandelbrot. Used to describe geometric figures that exhibit self-similarity, fractals are now used in myriad applications. Every fractal has a numeric fractal dimension that can be used to indicate how complex the fractal is. Multiple methods for calculating fractal dimension have been formulated. The box dimension method is most commonly used in research applications.

Raster graphics are images that assign numeric values to every pixel. The values determine the color of each pixel. Vector graphics contain object types like paths and shapes to show images.

In research applications, box dimension is calculated from raster images. This paper describes how to calculate box dimension from vector images. It also describes how to display the step-by-step calculations so that users can understand every step of the process.

1.2 Fractal Dimension

To calculate fractal dimension using the box dimension method, a square grid of size s is superimposed over the image. *N(s)*, the number of grids that contain part of the image, is counted. Box dimension, D, can be calculated using the following formula:

$$D = \frac{\log N(s)}{\log \frac{1}{s}} \tag{1}$$

1.3 Previous Research

Fractal dimension is used in many research applications. Previous projects have focused on the correlation between fractals dimension and aspects of the natural world. For example, Corbitt and Garbary found a correlation between fractal dimension and brown algae development. They took photographs of brown algae in different stages of development, calculated the box dimension of the photographs. (Photographs are raster images.) They knew that as the algae developed and grew older, the algae became more complex in shape. Corbitt and Garbary concluded that as algae became more complex, the fractal dimension of the algae increased.

1.4 Raster and Vector Graphics

Raster graphics is way of representing images. In raster graphics, images are collections of pixels, and numeric color values are stored for each pixel. Bitmap (BMP) is a raster graphics format. Vector graphics is a way of representing images by using primitives like paths (lines and curves) and points. Scalable Vector Graphics (SVG) is a format for creating and displaying vector graphics.

Raster graphics have poor resolution when they are examined on small scales. Vector graphics, on the other hand, retain clarity. In this project, only path primitives were used. Paths consist of lines and curves. Path data is usually stored in a single attribute, and consists of commands and coordinates: **d='M 200,66 L 266,66 L 266,133 L 200,133 L 200,66 z '** Commands such as **M**, **L**, **C**, and **z** indicate what kind of line or curve should be drawn. Research applications of fractal dimension use raster graphics formats. Researchers take digital images (in raster format) and apply the box-counting method, using many grids of variable size on the same image. Their data are plotted on an x-y plane, with log($N(s)$) on the y-axis and log($1/s$) on the x-axis. Using a linear regression, they find the line of best fit for the data, and D is the slope of that line.

2 Methods

2.1 Tools

This project also uses the Processing Development Environment. Processing, a language based on Java, comes with libraries and methods that are useful for creating graphical displays. The Candy library allows users to import SVG files, embed them in graphical displays, and search through them to find information.

2.2 Creation of Vector Graphics Images

Using Inkscape, an open source vector graphics editor, images are created. These images include fractals and non-fractal shapes. In order to create fractals, an image is cloned a certain number of times until the desired level of iteration is achieved. For example, to make the Sierpinski Carpet fractal, a black square was cloned 9 times using the Tiled Clones option. The original square and the fifth cloned square were deleted. The remaining eight squares were grouped together and treated like the original square. Finally, all of the squares were transformed from object types to path types. SVG images were black-and-white, with black representing the object whose fractal dimension was to be calculated.

2.3 Calculation of Fractal Dimension

The Processing language and environment were used to calculate fractal dimension. The SVG data file was loaded and parsed into an array of strings (each line of the file was an element of the array). Using methods from the Candy library, the SVG image was displayed onscreen. The functions **split(String s)**, **trim()**, and **indexOf(String s)** were used to process the SVG data file. Information for each path in the file was stored as an array of Coordinates, paths.

A global integer was created to keep track of *s*, grid size. A nested for-loop created an array, grid, that stored the Coordinates of each grid boxs upper-left hand corner. Every element in grid was tested to see if it covered any part of the black object (in the SVG image). Two methods were used: **case1()** and **case2()**. The number of elements in grid that covered the object was *N(s)*. *S* and *N(s)* were stored in an array called **data**.

Whenever the button **step** was pressed, the grid size was decreased and the counting process was repeated. *D*, fractal dimension, was calculated using the 2 most recent additions to data:

$$D = \frac{\log N(s_2) - \log N(s_1)}{\log \frac{1}{s_2} - \log \frac{1}{s_1}} \tag{2}$$

3 Results

This project demonstrated that it is feasible to calculate fractal dimension from vector images. In the Sierpinski Carpet example, the calculated fractal dimension had 4.238% error, using all data points. A rectangle had a calculated dimension of 1.874, and a line had a calculated dimension of 0.939. The box-counting method of calculating fractal dimension was shown onscreen. Appendix A shows data.

4 Conclusion

In the future, this vector graphics method of calculating fractal dimension may be shown to be more efficient than the raster graphics method. In that case, researchers may be able to save time by converting their raster images to vector images and using the vector graphics method.

The properties of vector graphics allow minute image details to be shown. Researchers may want to find the fractal dimension of objects that have such minute detail, and using the vector graphics method may yield the best results in terms of accuracy. Researchers who use computer models of fractals may choose to use the vector graphics representation of their models.

5 Recommendations

A useful extension of this project would be to create a function that uploads multiple SVG files at once and writes the corresponding box dimensions to an external file. This would be useful for researchers who need to find fractal dimension en masse.

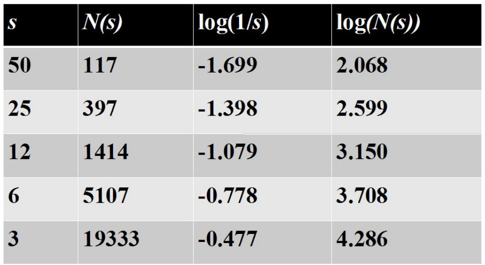
Another extension would be to create a function that calculates box dimension from raster images. The Processing program would time how long the raster image calculations would take, and then compare that with how long the SVG calculations would take.

An imcomplete extension calculates a least-squares linear regression from collected data and graphs the data, along with a regression line, on the screen.

6 Acknowledgements

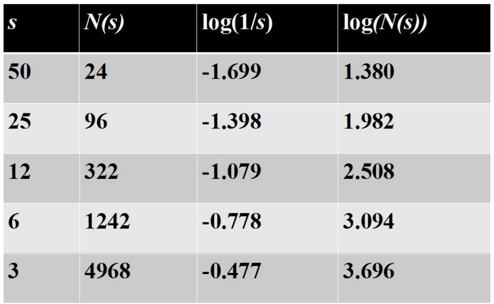
Thanks to Mr. Latimer for being my mentor and to my parents for being my parents.

7 Appendix A: Data



s	*N(s)*	log(1/*s*)	log(*N(s)*)
50	117	-1.699	2.068
25	397	-1.398	2.599
12	1414	-1.079	3.150
6	5107	-0.778	3.708
3	19333	-0.477	4.286

Figure 1: TABLE 1. Grid size s and number of grid boxes N(s), calculated from a Sierpinski Carpet.

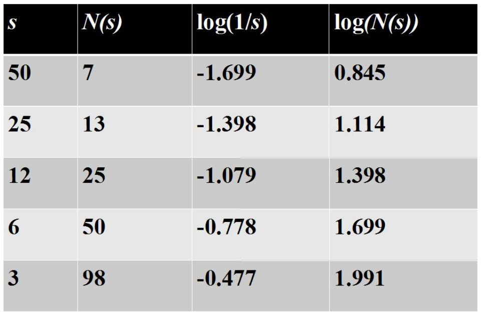


s	*N(s)*	log(1/*s*)	log(*N(s)*)
50	24	-1.699	1.380
25	96	-1.398	1.982
12	322	-1.079	2.508
6	1242	-0.778	3.094
3	4968	-0.477	3.696

Figure 2: TABLE 2. Data calculated from a rectangle

References

[1] Bourke, P. (1993). Fractal Dimension Calculator Manual. Retrieved 18:09:02 10, 2008, from `http://local.wasp.uwa.edu.au/~pbourke/fractals/fracdim/fdc_orig/`



s	*N(s)*	log(1/*s*)	log(*N(s)*)
50	7	-1.699	0.845
25	13	-1.398	1.114
12	25	-1.079	1.398
6	50	-0.778	1.699
3	98	-0.477	1.991

Figure 3: TABLE 3. Data calculated from a line

[2] Kraft, R. (1995). 3.2 Box Dimension. Retrieved 18:12:04 10, 2008, from http://www.weihenstephan.de/ane/dimensions/subsection3_4_2.html

[3] Lilley, C., & Schepers, D. (2007). Scalable Vector Graphics (SVG). Retrieved 18:07:31 10, 2008, from http://www.w3.org/Graphics/SVG/

[4] Peterson, I. (1993). From surface scum to fractal swirls. , 53. Retrieved 18:09:40 10, 2008, from http://www.jstor.org/view/00368423/ap070941/07a00070/0?currentResult=00368423%2bap070941%2b07a00070%2b0%2c03&searchUrl=http%3A%2F%2Fwww.jstor.org%2Fsearch%2FBasicResults%3Fhp%3D25%26si%3D1%26gw%3Djtx%26jtxsi%3D1%26jcpsi%3D1%26artsi%3D1%26Query%3Dfractal%2Bdimension%26wc%3Don

[5] Sutherland, S. (Tran.). (2002). Fractal Dimension (S. Sutherland, Tran.). Retrieved 18:12:37 10, 2008, from http://www.math.sunysb.edu/~scott/Book331/Fractal_Dimension.html