Geological Quarterly, 2009, **53** (2): 241–248

BCFD — a *Visual Basic* program for calculation of the fractal dimension of digitized geological image data using a box-counting technique

Timotej VERBOVŠEK

Verbovšek T. (2009) — *BCFD* — a *Visual Basic* program for calculation of the fractal dimension of digitized geological image data using a box-counting technique. Geol. Quart., **53** (2): 241–248. Warszawa.

The *BCFD* program was developed for the analysis of digitized objects using a box-counting algorithm, which has the largest number of applications among the fractal methods in the geosciences. Counting is performed by scanning of image pixels in boxes of different sizes, and the number of boxes is determined automatically from the image resolution. The program calculates the fractal dimension D of the objects in the image, along with the coefficient of determination R^2. Input files are thus transferred to ubiquitous BMP images, in a 1-bit monochrome format. The program outputs the results on screen, into a text file and optionally also directly into *MS Excel*, where the data can be further used in charts or other calculations. It was tested with three fractal and three Euclidean objects with known theoretical values, plus three geological image data (a natural river network and two fracture networks), and gave results with very high or perfect theoretical accuracy. Application of data values obtained is presented with several examples. *BCFD* is written in *Visual Basic 6.0*. The source code is freely available, and is open for any modifications or integration with other software packages that are powered by *Visual Basic for Applications* (*VBA*) or its equivalent.

Timotej Verbovšek, University of Ljubljana, Faculty of Natural Sciences and Engineering, Department of Geology, Aškerčeva 12, SI-1000 Ljubljana, Slovenia; e-mail: timotej.verbovsek@ntf.uni-lj.si, timotej.verbovsek@guest.arnes.si *(received: December 10, 2007; accepted: October 17, 2008).*

Key words: box-counting, program, *Visual Basic*, image analysis, fractal dimension.

INTRODUCTION

Since its invention about 30 years ago, fractal analysis has proved to be very useful, and is above all applicable in the geosciences. Fractal methodologies are appropriate where classical geometry is not suitable for describing the irregular objects found in nature (Mandelbrot, 1983). Fractals are most easily defined as geometric objects with a self-similar property, which defines that they retain their shape under any magnification, i.e. they do not change shape with scale (Feder, 1988; Angeles *et al*., 2004). Another fundamental property is their fractal dimension (D), which yields important insights into the physical properties of geological materials (Turcotte, 1992; Dillon *et al*., 2001). It can occupy non-integer values, compared to the integer values characteristic of Euclidean objects, such as 3D cubes or 2D planar surfaces. As an example, a well-known fractal object, the Koch curve (Fig. 1), has a fractal dimension of about 1.26, and therefore exhibits properties of both 1D and 2D objects, as it fills more space than a line ($D = 1$) and less space than a surface ($D = 2$). Use of fractal analyses based on calculation of the fractal dimension has found an application in many fields, including geology and geophysics (Turcotte, 1992), speleology (Kusumayudha *et al*., 2000), geomorphology (Angeles *et al*., 2004), analysis of fracture networks (Bonnet *et al*., 2001), atmospheric research (Brewer and Di Girolamo, 2006), river networks analysis (Schuller *et al*., 2001), and also in other non-earth sciences, such as medicine, space sciences, physics, chemistry, economics, and others.

As most fractal analysis software is either specialized or commercial, it is often hard to find appropriate programs to perform analyses. The following paper describes the design, details of use and applications of the *BCFD* program, developed for determination of the fractal dimension of digitized objects using a box-counting algorithm. The program runs in the *MS Windows* environment and is written in *Visual Basic 6.0*. The source code is freely available, with the freedom for modification to meet the user's purposes or for onward integration with other software packages, especially using *Visual Basic for Applications (VBA)*.

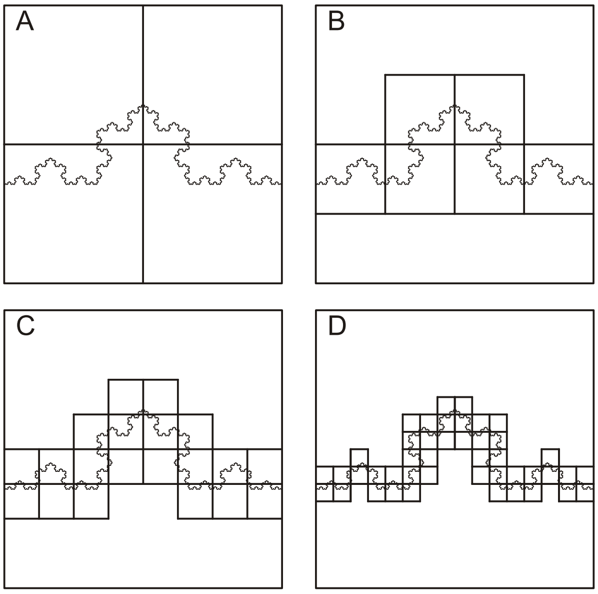

Fig. 1. Box-counting technique

Only the occupied boxes are shown covering the Koch curve (D = 1.2618); **A** — 4 boxes (step 2), **B** — 6 boxes (step 3), **C** — 20 boxes (step 4) and **D** — 44 boxes (step 5); the first step (one box) is not shown; see also Table 1

METHODS

ESTIMATION OF THE FRACTAL DIMENSION USING THE BOX-COUNTING TECHNIQUE

There are many definitions of dimension, and also many ways to attempt their determination. The most common dimensions are the self-similarity, compass and box-counting dimensions, and the latter has the most applications in science (Peitgen *et al*., 2004). One should note that many natural systems are self-affine rather than strictly self-similar and thus empirically derived box-counting dimensions for these objects are only estimates of their true fractal dimension. However, the box-counting dimension is still probably the most commonly used, as the principle of its use is rather simple. The digitized map of an object (for instance a river or fracture network) is covered by boxes of different side length "s", and then the number of occupied boxes $N(s)$ is counted for each box size (Feder, 1988; Bonnet *et al*., 2001). The process is repeated by reducing the box sizes by half their size (Fig. 1), with the largest box defined by the image resolution and the smallest box occupying one pixel of computer image. For fractal objects, the number of occupied boxes $N(s)$ follows the power-law relationship with the box size s: $N(s) \cdot s^{-D}$ and the fractal dimension D is therefore calculated as the slope of linear regression best-fit line of log-log data: $D = -\log N(s)/\log s$.

A typical log-log curve therefore represents a perfectly linear relationship of data points of the number of occupied boxes $N(s)$ and the box size s. Such a perfect relationship is valid only for ideal mathematical fractals such as the Koch curve described (Fig. 1). Mathematical fractals are by definition constructed from a set of rules, in contrast to natural fractals, and do not involve any random processes such as geological processes. Natural fractals can be classified as statistical fractals, which are not strictly self-similar and do not preserve their shape across all scales as do mathematical fractals. Statistical fractals have only their numerical or statistical measures preserved across all scales, and this measure is represented by the fractal dimension. The relationship of log $N(s)$–log s data plots on the graph as a perfect line for mathematical fractals.

Despite the apparent simplicity of the box-counting method, users should be aware of the potential pitfalls of the box-counting technique, especially for real geological data. If the correlation of log $N(s)$–log s data plots on the graph as a curve and not as a line (which is typical for natural fractals), only the valid range (the linear part of the curve) should be examined. In addition, the unmapped space outside the studied area should not be included in the analysis and the image examined must therefore be embraced completely within this area (Walsh and Watterson, 1993). Most common deviations from a line of log $N(s)$–log s data plots occur as a result of truncation and censoring effects. Truncation occurs as a shallowing of the line's slope at the lower end of the scale range, as for real data, and the number of smaller objects (e.g., fractures) below some threshold values can be under-sampled. On the other hand, censoring occurs if the objects analysed pass outside the observed region, causing steepening of the curve in plots at the upper end of the scale range (Bonnet *et al*., 2001). These effects are easily seen in the log-log plots, and are for the purpose of simplicity not implemented in the program calculations. To correctly determine the fractal dimensions of real geological objects, calculations must be carried out carefully, and the fractal structure should be not only calculated but also verified afterwards by examination of the log-log plots.

It is necessary to comment that the values of fractal dimensions of river and fracture networks used as geological examples in this paper are presented for both methods of box-counting: the complete one, using all data points in the log-log plots (as calculated by *BCFD*), and the linear one, using only the linear part of the log-log plots (later determined visually in *MS Excel*). It is not the main goal of this paper to discuss the meaning of the exact values of fractal dimensions for different geological data, so the user must examine the data carefully to analyse only the fractal part.

PROGRAM DESIGN

After opening the file, the program reads the header of the BMP image. It first checks the image for BMP format and looks for proper colour depth and resolution (width × height). If any of these parameters do not match appropriate values, the program warns the user of the error type and exits. After checking the file, the program consequently reads the image into dynamic memory array (pixel ()) by scanning each row starting from lower left to upper right corner (Fig. 2). Every byte is converted into eight bits of image object information. A bit value of zero represents white image background and a value of one represents the black pixel of the object. A hexadecimal value of 81h (10000001 binary) therefore represents the object occupying the first and the last bits (byte 34 in Fig. 2). The object is thus read into the virtual screen with the same resolution as the image. After the object has been read into the array, the

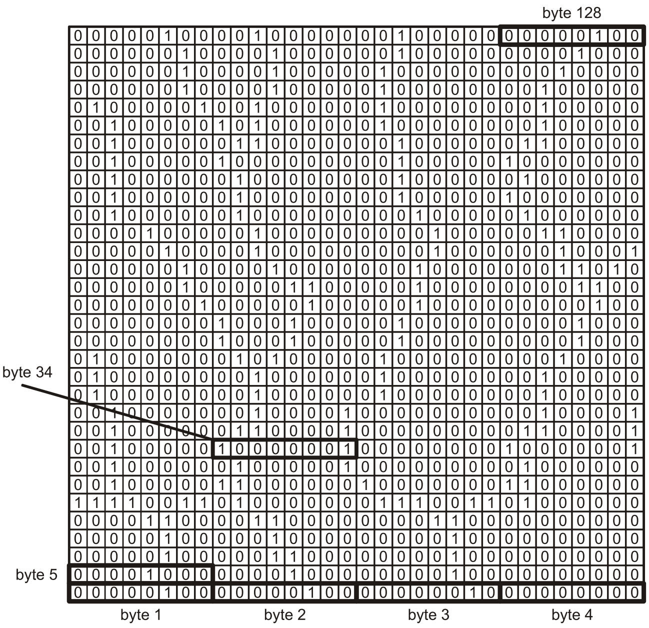

Fig. 2. Example image of 32 × 32 pixels, read by program into 128 bytes from lower left to upper right corner

Values of 1 represent objects, and 0 background

box-counting is performed by subdividing the image into smaller boxes, and in each of them the pixels of the array are scanned from the lower left to upper right corner of the box. If at least one object part (marked by a bit value of 1) is found, the box is regarded as occupied. As the scanning of each of smaller boxes reaches the end of the array, the size of boxes is reduced by half and scanning is repeated. The process is complete when all the boxes of one pixel size are scanned. The pixels on the box edges are scanned only once.

Finally, the fractal dimension (D) and squared value of Pearson's correlation coefficient (R^2), which illustrates the goodness of fit (Borradaile, 2003), are calculated. D is calculated as a negative value of the slope of the best linear-fit regression line $D = -\sum\left[\left(x-\bar{x}\right)\left(y-\bar{y}\right)\right] / \sum\left[x-\bar{s}\right]^2$. In *MS Excel*, both values are calculated by internal functions, D by the *SLOPE* function and R^2 by the *RSQ* function.

Due to the nature of the program, only one-bit (two-colour black and white) uncompressed bitmap files (*.BMP) are supported as input files. Supported image resolutions are 256 × 256, 512 × 512, 1.024 × 1.024, 2.048 × 2.048 and 4.096 × 4.096 pixels. The minimum value is chosen as such because images of lower resolution cannot faithfully represent fractal and natural objects, and images greater than the maximum value are slow to process and are seldom used. The supported resolutions can easily be added or changed in a single line (Case 4096, 2048, 1024, 512, 256) of code in the subroutine CheckResolution().

Output results are given as a table of box sizes s and corresponding number of occupied boxes N (s) plus the D and R^2 values. These results are written on screen and into the text (*.txt) file. If the checkbox "Write to Excel file" below the Open button is turned on, output is also sent directly into an *MS Excel* spreadsheet, with direct calculation of D and R^2. Both files are written in the same folder as the analysed image, and the file names are presented in the *BCFD* output screen (Fig. 3). Output to *Excel* is supported and preferred, as it is possible to further represent the data on charts, include them into other calculations or to check the slope of fitted data for truncation or censoring effects (Bonnet *el al.*, 2001).

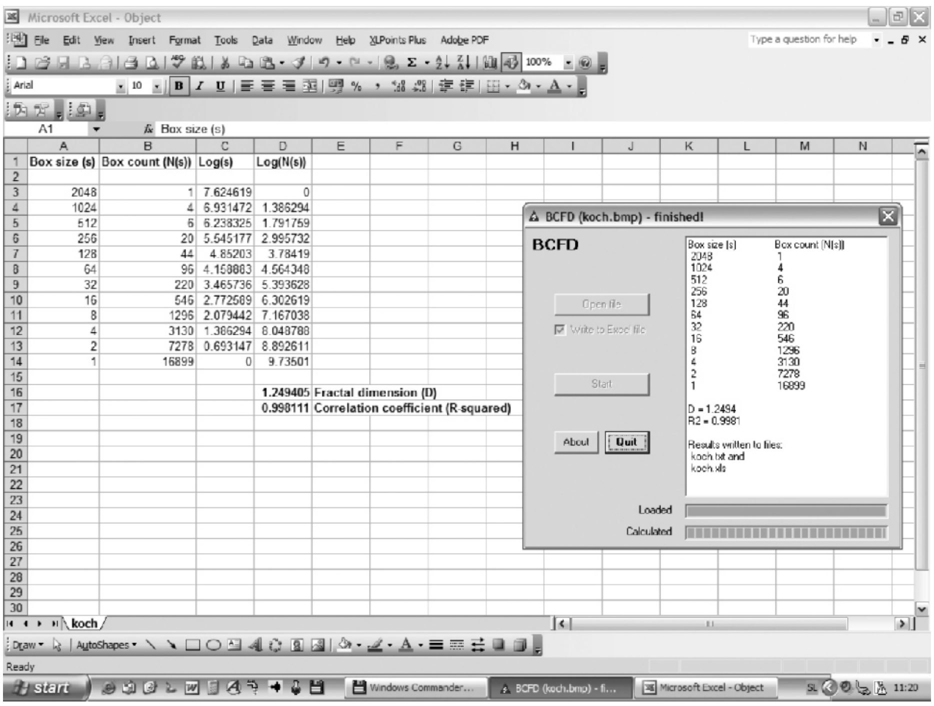

Fig. 3. Screenshot of the program *BCFD* and *MS Excel* with finished results

The program is written in *Visual Basic* (*VB*), *version 6.0*. This language has become one of the world's most widely used programming languages due to its simplicity and ease of use. Even if *BCFD* is intended to run as a stand-alone program, it can straightforwardly be integrated into various applications by *Visual Basic for Applications* (*VBA*). This represents a version of the VB language integrated into applications, which does not permit development of stand-alone executable files. It is fully compatible with *Visual Basic* and is intended for the automatization and customization of applications in other programs (Hart-Davis, 1999). *VBA* or compatible versions of *Visual Basic* power, for example, some of the most known popular software, such as *Microsoft Office*, *AutoCAD* (*AutoDesk*, *Inc*.), *Statistica* (*StatSoft*, *Inc*.), *Adobe Photoshop* (*Adobe*, *Inc*.), *Surfer* (*Golden Software*, *Inc*.), *ArcGIS* (*ESRI*, *Inc*.), and many others. The code can be also be transferred to other versions of *Visual Basic*, such as the 2005 version or the NET platform (Patrick *et al*., 2006), without modification or with only minor modifications.

RESULTS AND DISCUSSION

ANALYSIS OF MATHEMATICAL FRACTAL TEST DATA AND EUCLIDEAN OBJECTS

The program has been tested with several images (Fig. 4): three fractals and three Euclidean objects with known fractal

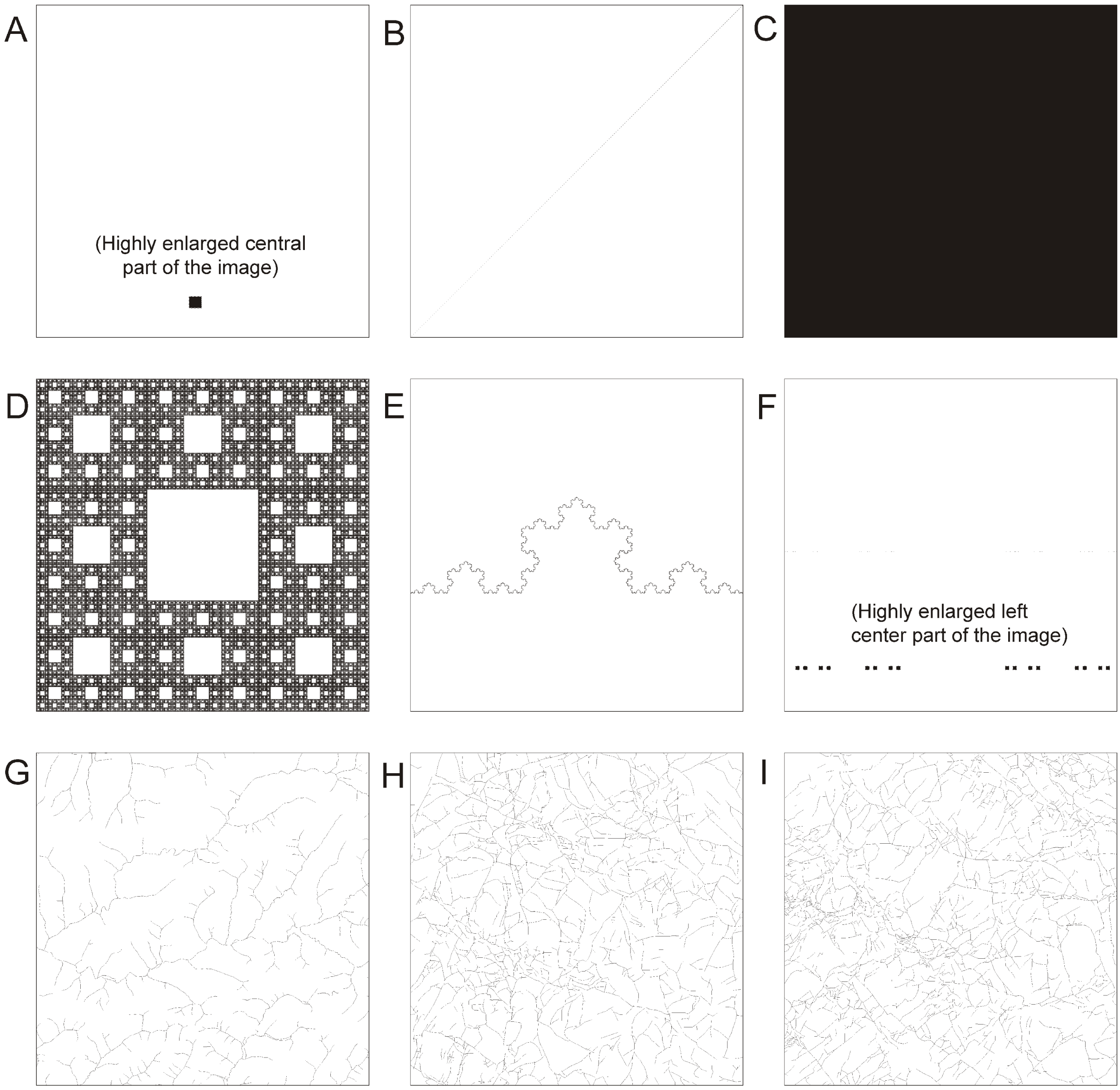

Fig. 4. Analysed images (example files in the folder "test_data")

A — point (enlarged), **B** — line, **C** — filled square, **D** — Sierpinski carpet, **E** — Koch curve, **F** — Cantor's dust (enlarged), **G** — natural river network example, **H** — natural fracture network example no. 1 (see Fig. 5A), **I** — natural fracture network example no. 2 (see Fig. 5B) dimensions plus three natural objects, described in the next section. The first comprise familiar self-similar fractal objects: the Sierpinski carpet (Fig. 4D), the Koch curve (Fig. 4E), and Cantor's dust (Fig. 4F). These fractals have non-integral dimensions and are constructed by an infinite series of iterations (Feder, 1988; Peitgen *et al*., 2004), so they belong to the group of strictly self-similar mathematical objects. All tested images have a resolution of 2.048 × 2.048 pixels. This resolution is sufficient to allow faithful representation of fractal objects by seven iteration steps, so the smallest irregularity on the fractals is equal to or smaller than one pixel of the image. This requirement is important, as the number of iterations affects the value of D, and too small a number of iterations yields inferior results (Dillon *et al*., 2001). Euclidean objects with integral dimensions, used as examples, include a point (Fig. 4A), a line (Fig. 4B) and a filled square (Fig. 4C).

Results (Table 1) show that the program calculates the fractal dimension (D_c) with perfect or very high accuracy. Deviations from ideal dimension values (D_t) can be attributed to two facts. First, all tested fractals are, by definition, self-similar objects constructed by an infinite number of iterations, and in the example images only the first seven iterations are shown. As the pixel size achieved on monitor is not infinitely small, to allow representation of all iterations, some minor error is always present. Secondly, several fractals such as Koch curve are triangular objects and are therefore impossible to represent faithfully on the square grid of the computer screen or array.

ANALYSIS OF NATURAL GEOLOGICAL DATA

Natural data representing a river channel network and two fracture networks were analysed in order to present some geological examples of the program application. The first example is a natural river network (Fig. 4G), digitized from topographical maps[1]. The area of the image covers about 8.8 × 8.8 km in south-west Slovenia. The river network is developed in flysch rocks of Tertiary age, composed mostly of low-permeability rocks (marls, mudstones and sandstones). The theoretical limits of fractal dimensions for river networks are 1 (single straight channels) and 2, which would imply a fully braided river, filling the complete terrain. However, the expected dimensions are lower than 2, because of the geological, topological and hydrological restraints that reduce the ability of the stream network to develop fully (Schuller *et al*., 2001). Skeletal images of rivers were used instead of complete digitized images, as they are likely to provide better material to estimate the fractal dimension then the original images (Foroutan-pour *et al*., 1999).

The obtained value of D (Table 1) for natural river networks ($D = 1.36$) is in agreement with dimensions for river networks, which can vary widely (1.28–1.71 according to the method used; Schuller *et al*., 2001). Variations occur because of the self-affine properties of river networks. The fractal dimension and self-similar or self-affine properties obtained from image analysis can be further applied to an understanding of the behaviour of river network development and the fluvial erosion topography which influences the networks (Veneziano and Niemann, 2000). If the log-log plots are afterwards analysed visually, the fractal dimensions can be higher for only the linear part of the curve. However, the discussion on exact values of different fractal properties of rivers or other data and the visual inspection are not the focus of this paper, and the verification of the fractal structure must be carried out by the user, based on the data itself. The user must for instance decide which data points of the curve to use, and this decision can be very subjective and cannot be directly implemented in the program code. The same note applies to the following calculations of fractal dimensions of fracture networks. Program *BCFD* should be therefore viewed as an open-source starting point code with further possible modifications and upgrades.

Two geological examples of analysed natural fracture networks are also given. They are obtained from the "Main" dolomite (analogous to German Hauptdolomit or Italian Dolomia

Table 1

Example objects (Fig. 4) with their theoretical (D_t) and calculated (D_c) dimensions, coefficient of determination (R^2) and difference (dif.) between D_t and D_c

Object type	Object	D_t	D_c	R^2	dif. [%]
Fractal	Cantor's dust	0.6309 (= log2/log3)	0.6367	0.9784	0.92
Fractal	Koch curve	1.2618 (= log4/log3)	1.2494	0.9981	–0.99
Fractal	Sierpinski carpet	1.8928 (= log8/log3)	1.8994	0.9997	0.35
Euclidean	point	0	0.0000	–	0.00
Euclidean	line	1	1.0000	1.0000	0.00
Euclidean	filled square	2	2.0000	1.0000	0.00
Natural	river network	(between 1 and 2)	1.3625	0.9699	–
Natural	fracture network 1	(between 1 and 2)	1.5111	0.9744	–
Natural	fracture network 2	(between 1 and 2)	1.5254	0.9750	–

[1] EUROWATERNET http://nfp-si.eionet.eu.int/ewnsi/ (accessed on 21.05.2004) and http://eionet-si.arso.gov.si/Dokumenti/GIS/voda/index_eng.htm (accessed on 18.09.2007)

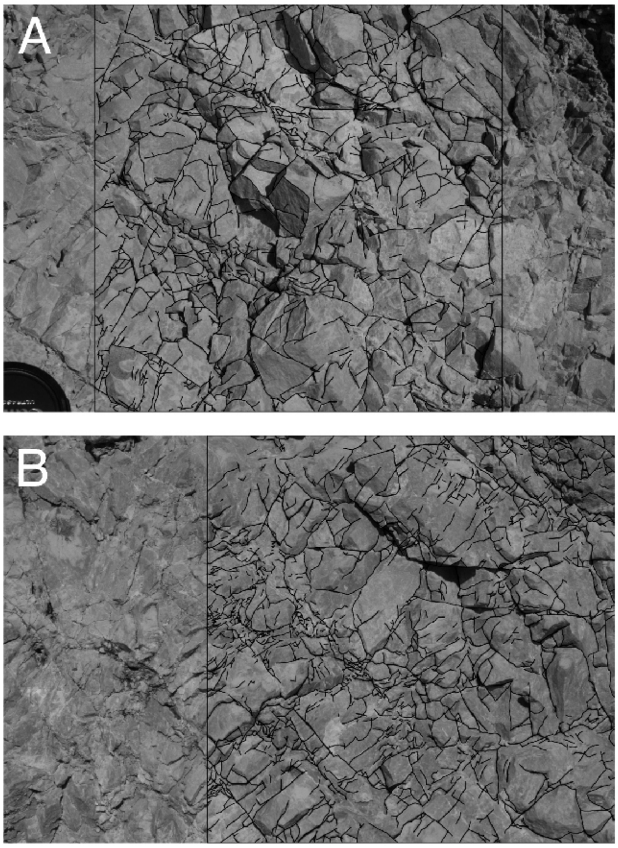

Fig. 5. Photographs of dolomite exposures with superimposed digitized fracture network traces

Principale) of Upper Triassic age (Verbovšek, 2008) in Southern Slovenia. Fracture networks were obtained by photographing the dolomite outcrops, where fractures were well exposed (Fig. 5A, B), and the traces of the fractures were further digitized into vector format. The width of the superimposed lines in the photographs is 40 times larger than the one actually used in digitalization, to make the fractures visible on the photographs. The width of both photographs is approximately 32 cm. Fractures are digitized inside a square instead of inside a complete photograph owing to box-counting requirements. Vector format images were further converted into 1-bit BMP images, used for processing in the *BCFD* program.

The values of D for natural fracture networks (D_1 = 1.51 and D_2 = 1.53) are almost identical (average value D_{av} = 1.52) and are in agreement with the values of fracture networks given in a review paper by Bonnet *et al*. (2001). In some tectonic environments, several generations of fracturing can affect the rocks, and thus each generation of fractures is younger than the previous ones. Analysis of separate fracture characteristics from one generation to the next reveals the fracture networks' development. Observations by Barton (1995) show that the first-generation fractures are long and subparallel and network connectivity is poor. Second-generation fractures are shorter and form polygonal blocks with first-generation fractures. Younger fractures are generally shorter, variously oriented, and form small polygonal blocks. Addition of younger fractures therefore contributes to an increase in the fractal dimension of the complete network, as first-generation subparallel fractures exhibit low fractal dimensions, which increase with the addition of many smaller ones of later generations. The influence of fracture network evolution can possibly be tested in the laboratory or by physical experiments rather than in the field on exposures of dolomite rocks. When outcrops are, for example, inadequately exposed or fractures are affected by mineral infillings, this approach becomes inoperable (Barton, 1995). Additionally, the tectonic stresses for each fracturing episode (from the oldest to the youngest) should be known, and these data are mostly unavailable. The values of fractal dimensions of fractures networks in dolomites are therefore presented as a simple example of complete networks, as it was not possible to divide the fractures into separate generations.

The values of fractal dimensions of natural fracture networks can be further used as a parameter or analysed in several fields of geology. The first examples can be found in the field of engineering geology, where they are used to analyse the roughness of rock or soil particles and rock surfaces, to analyse the distribution of rock fragments resulting from blasting, or to describe the statistical homogeneity of jointed rock masses (Vallejo, 1997). A second application can be found in the study of underground water flow and transport in fractured rocks based on the interconnectivity and distribution of fractures and the influence of these two factors on permeability. It has been recognized that the flow dimension obtained from the well-test pressure curve is a function of the geometrical fractal dimension and these geometrical fractal dimensions are always equal to or greater than the flow dimension (Polek *et al*., 1990; Doughty and Karasaki, 2002). It is the flow dimension that should be used to characterize a network's behaviour during well tests. Knowledge of the geometrical fractal dimension, which can be acquired by the box-counting method, is therefore of great importance in understanding fluid flow and transport in fractured rocks. To illustrate the applicability of the geometrical dimension, we can extrapolate the obtained two-dimensional average value of the geometrical fractal dimension of fracture networks D = 1.52 to three dimensions. Extrapolation can be carried out using the formula $D_{3D} = D_{2D} + 1$ to obtain a value of D_{3D} = 2.52, as the intersection of a 3D fractal with a plane results in a fractal with D_{2D} equal to $D_{3D} - 1$ (Barton, 1995; Bonnet *et al*., 2001). The extrapolation is valid for non-mathematical and isotropic fractals, and fractures in intensely fractured dolomites are indeed close to this idealization. From these results we can therefore conclude that the flow dimension describing the geometry of water flow towards the water well (Barker, 1988) can reach a maximum value of D_{3D} = 2.52 in the dolomites analysed due to channelling effects (Polek *et al*., 1990; Doughty and Karasaki, 2002). For a more detailed analysis of these applications, more fracture networks should certainly be studied and checked for truncation and censoring effects, as only two examples are used in this paper to illustrate the applicability.

INFLUENCE OF IMAGE RESOLUTION ON THE RESULTS

Image resolution does not greatly influence the calculated values of D_c (Table 2), as shown for the example of the

Table 2

Influence of image resolution on calculated dimension

Resolution	D_c	R^2	dif. [%]
256 × 256	1.8656	0.9985	–1.44
512 × 512	1.8881	0.9991	–0.25
1.024 × 1.024	1.8946	0.9995	0.10
2.048 × 2.048	1.8994	0.9997	0.35
4.096 × 4.096	1.8967	0.9997	0.21

Example for Sierpinski carpet (D_t = 1.8928, Fig. 4D); for notation see Table 1

Sierpinski carpet. An original image with a resolution of 6.561 × 6.561 pixels was scaled down in five steps for the analysis. The coefficient of determination is very high in all cases and the difference dif. (= 100*(D_c–D_t)/D_t) does not show any increasing or decreasing trend, although the deviation is highest for the lowest resolution. As R^2 is very high for all resolutions, it can be concluded that the fractal dimension is more or less the same and hence independent of the image resolution. This is in agreement with observations (e.g., Dillon *et al*., 2001), which state that objects with the same form but of different size should retain a constant fractal dimension. Yet, when using digitized maps of natural objects, it is better to use high-resolution images, as these capture more details. Calculated values of D_c for all Euclidean objects are equal to theoretical ones.

COMPARISON WITH OTHER AVAILABLE BOX-COUNTING PROGRAMS

Very few box-counting programs are freely available on the internet. Many authors of the works discussed in the following paragraphs do not mention either the method or the program used for calculating the box-dimension. Therefore a comparison of *BCFD* with other programs cannot be made faithfully, or perhaps cannot be made at all. A short comparison with other programs is given in the following paragraphs.

The program *Fractal Dimension Calculation Software* (Foroutan-Pour *et al*., 1999) is available for Apple Macintosh computers. Therefore for reasons of incompatibility it cannot be tested in the *MS Windows* environment.

Angeles *et al*. (2004) have developed a specialized function in *MATLAB*[2] software for automatic box-counting. Their code is not freely available. In the code they have developed specific criteria to avoid the double counting of points that fall on box sides. In *BCFD* code, the code is designed in such a way that double counting is not possible.

Tang and Marangoni (2006) used the 2D and 3D algorithms that are now part of the commercial *TruSoft*[3] software (programs 3D-FD and Benoit). In addition, they also used the mass dimension method, which is not comparable to the box-counting method.

Fractscript (Dillon *et al*., 2001) is a macro developed for fractal analysis of multiple objects as part of the free *ImageTool*[4] package. It is written in a dialect language of *Pascal* and requires *ImageTool* to be installed. When compared with *BCFD*, the latter can be implemented in a much broader range of software, as *Visual Basic* language is integrated as VBA in a much broader spectrum of programs, as described in *Section 2.2*, than *ImageTool* and *Pascal*.

Although these programs are not directly comparable with *BCFD* due to the different methods or different computer codes used, it is most likely that *BCFD* code is easier to integrate into other programs as it uses *Visual Basic* and gives output directly into *MS Excel*.

Therefore, the only strictly comparable program is *VSBC* (*Visual Screen Box-Counting*; Gonzato, 1998). It is written in C language and uses PCX image format, which is now virtually obsolete. The program is small, fast and easy to use; however, in some cases it gives erroneous results. An image of the Koch curve (Fig. 4E) is used as an example. If the image is rotated by 90 degrees, the results of box-counting should be the same for all four rotations, as the image has the same width and height. *BCFD* calculates the number of occupied boxes for all rotations equally (N = 16.899), but *VSBC* gives different values (N = 23.043 for 0° and 180° angles and N = 23.380 for 90° and 270° angles). *BCFD* therefore fulfills the quality requirement that the object's final value must be independent of rotation and reflection (Dillon *et al*., 2001). The calculated fractal dimension value produced by *BCFD* (D = 1.25) is also much closer to the theoretical value (D = 1.26) than is that produced by *VSBC* (D = 1.33 for 0° and 180° angles or D = 1.31 for 90° and 270° angles). Similar conclusions can be made for R^2, although values produced by both programs are very high. Image resolution can be changed in *VSBC* only in the C source code, whereas *BCFD* determines box sizes from the image itself. *VSBC* also has problems with calculations for single point (Fig. 4A), as it outputs the number of occupied boxes as zero, one or two for different box sizes, and the true values, calculated by *BCFD*, are one (single point occupied) for all cases.

CONCLUSIONS

The *BCFD* program was shown to perform box-counting analysis with high accuracy. It has been tested extensively with several objects (fractal, Euclidean, and three natural geological examples — river channels and two fracture networks). Application of data values obtained has been presented with a few examples. The program gives results very close or equal to theoretically expected values. It is user-friendly and utilizes BMP format, available to most graphic programs. As the source code is freely available, it can easily be modified or integrated into any

[2] The MathWorks Inc., *http://www.mathworks.com*.
[3] TruSoft International Inc., *http://www.trusoft.netmegs.com*.
[4] UTHSCSA ImageTool, *http://ddsdx.uthscsa.edu/dig/itdesc.html*.

image analysis software that uses *Visual Basic for Applications* (*VBA*) or a compatible language as a background engine. As *Visual Basic* powers the majority of today's software applications, the program can be used broadly, either as a stand-alone or integrated version. It will therefore hopefully be of use to all geoscientists dealing with fractal analyses of images.

The program (executable version, test files, source code, and other files) is available on http://www.geo.ntf.uni-lj.si/tverbovsek/ programi.html

Acknowledgments. The author thanks G. Gonzato for providing the executable version of the *VSBC* program and D. J. Lowe and others for smoothing the English version of the text.

REFERENCES

ANGELES G. R., PERILLO M. E., PICCOLO M. C. and PIERINI J. O. (2004) — Fractal analysis of tidal channels in the Bahía Blanca Estuary (Argentina). Geomorphology, **57**: 263–274.

BARKER J. A. (1988) — Generalized radial flow model for hydraulic tests in fractured Rock. Water Resour. Res., **24** (10): 1796–1804.

BARTON C. C. (1995) — Fractal analysis and spatial clustering of fractures. In: Fractals in the Earth Sciences (eds. C. C. Barton and P. R. La Pointe): 141–178. Plenum Press, New York.

BONNET E., BOUR O., ODLING N. E., DAVY P., MAIN I., COWIE P. and BERKOWITZ B. (2001) — Scaling of fracture systems in geological media. Rev. Geophys., **39** (3): 347–383.

BORRADAILE G. (2003) — Statistics of Earth Science Data. Springer, Berlin.

BREWER J. and DI GIROLAMO L. (2006) — Limitations of fractal dimension estimation algorithms with implications for cloud studies. Atmos. Res., **82**: 433–454.

DILLON C. G., CAREY P. F. and WORDEN R. H. (2001) — Fractscript: a macro for calculating the fractal dimension of object perimeters in images of multiple objects. Comput. Geosc., **27**: 787–794.

DOUGHTY C. and KARASAKI K. (2002) — Flow and transport in hierarchically fractured rock. J. Hydrol., **263**: 1–22.

FEDER J. (1988) — Fractals. Plenum Press, New York.

FOROUTAN-POUR K., DUTILLEUL P. and SMITH D. L. (1999) — Advances in the implementation of the box-counting method of fractal dimension estimation. Appl. Math. Comput., **105**: 195–210.

GONZATO G. (1998) — A practical implementation of the box counting algorithm. Comput. Geosc., **24** (1): 95–100.

HART-DAVIS G. (1999) — Mastering VBA 6. Sybex, San Francisco, CA.

KUSUMAYUDHA S. B., ZEN M. T., NOTOSISWOYO S. and SAYOGA GAUTAMA R. (2000) — Fractal analysis of the Oyo River, cave systems, and topography of the Gunungsewu karst area, central Java, Indonesia. Hydrogeol. J., **8**: 271–278.

MANDELBROT B. (1983) — The Fractal Geometry of Nature. W. H. Freeman & Co., New York.

PATRICK T., ROMAN S., PETRUSHA R. and LOMAX P. (2006) — Visual Basic 2005 in a nutshell. O'Reilly Media, Sebastopol, CA.

PEITGEN H-O., JÜRGENS H. and SAUPE D. (2004) — Chaos and fractals. New Frontiers of Science. Springer-Verlag, New York.

POLEK J., KARASAKI K., LONG J. C. S. and BARKER J. A. (1990) — Flow to Wells in Fractured Rock with Fractal Structure. Department of Material Science and Mineral Engineering. Univ. California, Berkeley. Rep. LBL-27900, Lawrence Berkeley Laboratory, Berkeley, California: 74–76.

SCHULLER D. J., RAO A. R. and JEONG G. D. (2001) — Fractal characteristics of dense stream networks. J. Hydrol., **243**: 1–16.

TANG D. and MARANGONI A. G. (2006) — 3D fractal dimension of fat crystal networks. Chem. Phys. Lett., **433**: 248–252.

TURCOTTE D. L. (1992) — Fractals and Chaos in Geology and Geophysics. Cambridge Univ. Press, Cambridge.

VALLEJO L. E. (1997) — Fractals in engineering geology. Preface. Eng. Geol., **48**: 159–160.

VENEZIANO D. and NIEMANN J. D. (2000) — Self-similarity and multifractality of fluvial erosion topography. Mathematical conditions and physical origin. 1. Water Resour. Res., **36** (7): 1923–1936.

VERBOVŠEK T. (2008) —– Diagenetic effects on the well yield of dolomite aquifers in Slovenia. Environ. Geol., **53** (6): 1173–1182.

WALSH J. J. and WATTERSON J. (1993) — Fractal analysis of fracture patterns using the standard box-counting technique. Valid and invalid methodologies. J. Struct. Geol., **15** (12): 1509–1512.