Morphological Characterisation of Microbial Aggregates by Image Analysis

António Luís Amaral, Maria Madalena Alves, Manuel Mota, Eugénio Campos Ferreira

Departamento de Engenharia Biológica - Universidade do Minho
Campus de Gualtar - 4710 BRAGA, PORTUGAL
e-mail: eb10336@ci.uminho.pt
e-mail: {madalena, mmota, eugenio}@deb.uminho.pt

Abstract: *A program, was accomplished in MATLAB, that allows the morphological characterisation of particles with special emphasis to microbial aggregates. After automatic identification and isolation of different particles in a real image, the program determines several fractal dimensions as the mass dimension, the surface dimension, the area versus perimeter dimension, and the area vs. Feret diameter. The values of many other parameters are also provided, such as: convexity, compactness, roundness, as well as the 0th, 1st, and 2nd moments, whether the 0th moment is the area or the volume of the particle. The program also displays the particle size distribution. The program is easy to operate and all the steps that need decisions from the user are displayed in menus and buttons. As an example of application, the morphological characterisation of different microbial aggregates present in anaerobic wastewater treatment systems has been performed.*

Keywords:, *Image analysis, microbial aggregates, fractal dimensions, flocs, granules, MATLAB.*

1. Introduction

Image analysis is commonly used in a wide range of applications within the biological sciences. Some examples are the study of algal cells, muscle fibres, DNA sequencing gel autoradiograph, and fungal hyphae [11]. It allows the enhancement of pictures as well as automatic identification and isolation of particles so that they can be properly identified and studied. It also provides an extremely fast mean of getting morphologic information of the microbial aggregates in the picture, and saving thus tremendous waste of time and effort. Image analysis can be subdivided into five stages: display, filters, segmentation, mathematical morphology, and measurements. Analysis of any particular image is likely to require several of these stages, in this order, but sometimes re-using techniques from previous stages. The way in which data was collected and the questions to be answered, are of crucial importance in determining how a particular image should be analysed.

There are several commercial software packages for image processing such as the one used in this work, which is the MATLAB (The Mathworks, USA). MATLAB was originally developed as a matrix laboratory, written to provide an easy access to matricial software. The basic data element in MATLAB is a matrix and the commands are written in a very similar way as the used by the mathematics and engineering. The MATLAB includes numeric computation and visualisation functions. It also includes specialised toolboxes as the *Signal Processing Toolbox*, the *System Identification Toolbox*, the *Image Processing Toolbox*, and the *Statistic Toolbox*. The *Image Processing Toolbox* offers a powerful and flexible environment for analysing and processing images. The MATLAB is ideal for processing images because it has a matrix oriented language and each image can be represented by a matrix with each element corresponding to a pixel of that image. A program called *Imago* was developed in MATLAB language in order to accomplish the objectives of this work. Among other parameters, the program calculates different fractal dimensions.

The fractal dimension, a parameter emerged from the fractal theory [6] is a generic term without a restricted definition that embraces a wide range of different yet interrelated dimensions [4]. It provides a mean to measure the complexity and irregularity of the objects. The values of the fractal dimension are highly affected by the resolution of the image and the pixel representation of the metric system. The last one is also responsible for the loss of information of the real object and the mathematical formulas that are applied in the metric system are not very precise in a pixel representation.

The fractal dimension concept has been applied to describe highly irregular and complex structures [1,5]. In aggregates formed in water and wastewater treatment systems, this concept explained some unexpected phenomena [6]. The fractal dimension reflects the hydrodynamic environment in which aggregates, both microbial and inorganic, are formed [7] and it is possible to use this parameter to study the process of aggregation. This possibility is particularly interesting in anaerobic wastewater treatment where the characteristics of the aggregates play a crucial role to the performance, and operational stability, specially in the Upflow Anaerobic Sludge Blanket (UASB) reactor. The process of granulation and the definition of granules characteristics is nowadays particularly important due to the economical interests involved. It has been proven that granular biomass developed in UASB reactors was morphologically different from flocculent biomass present in an anaerobic filter [2]. The fractal dimension was used to differentiate between this kind of microbial aggregates and, as stated by these authors, flocs have more irregular boundaries than granules. These morphological differences are evidenced in Fig. 1 for three floc particles and three granular particles.

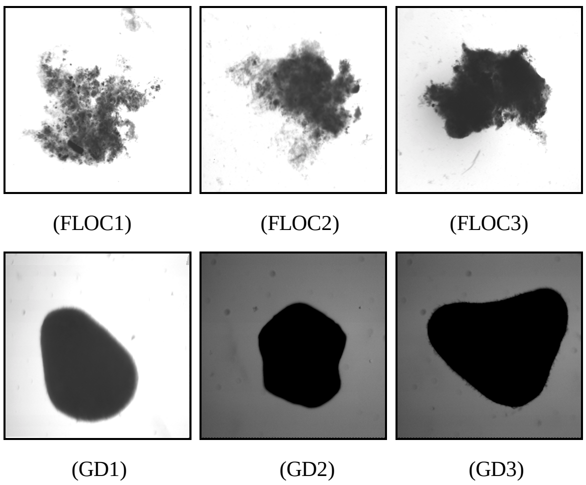

Figure 1. Digitised images of 3 flocs and 3 granules.

The irregular boundaries imposes lower values of fractal dimensions for the flocculent structures.

This work describes a MATLAB implementation of an image processing program. It is possible to the user to quantify how more complex an object is than other. Convexity, compactness, roundness and the moments are also calculated. As an example of application, the morphological characterisation of several microbial aggregates withdrawn from anaerobic wastewater treatment systems are presented.

2. Morphological parameters

2.1 Fractal Dimensions

For the different fractal dimensions be determined there is the need of calculating first several other dimensions, such as area, perimeter, and Feret diameter, in different box sizes. To determine the area and the perimeter of the objects for each box size it was used a box counting algorithm [1] which applies a grid over a binary image of the object (white pixels) and determination of that box category: interior box (contained entirely within the body of the object), border box (box which overlaps the boundary of the object) or empty box (which contains only black pixels). It is known that the number of S sized boxes needed to cover a fractal surface - N(S) - is proportional to S^{-D}, where D is the fractal dimension, which results:

$$N(S) = K \times S^{-D} \qquad (1)$$

Applying logarithms to both parts of the equation:

$$\log[N(S)] = \log[K] - D \times \log[S] \qquad (2)$$

The fractal dimension (D) is then obtained by representing -log (N(S)) *versus* S.

There are two mass fractal dimensions determined in the *Imago* program. The D_{Bm1} is calculated by the equation already described and the D_{Bm2} is calculated by the following plot:

$$-\log\left[N(s) - \frac{1}{2} \times N_{border}(s)\right] \textit{ versus } \log(S)$$

where N(S) is the area of the object in box size S and $N_{border}(S)$ is the perimeter of the object in box size S.

The surface fractal dimension (D_{Bs}) has values among 1 and 2, as all the other fractal dimensions determined in *Imago*. As the objects become more regular, the D_{Bs} value approaches 1. For circles, squares, and lines it should be equal to the unity. By the contrary, the mass fractal, the area *vs.* perimeter, and area *vs.* Feret diameter dimensions take greater values for objects with homogeneous and regular shapes. For the determination of the area *vs.* Feret diameter it is needed to know the value of this last one. The Feret diameter [2] can be defined as being the largest distance between two parallel tangents touching opposites sides of an object.

2.2 Convexity, Compactness, and Roundness

The convexity, compactness, and roundness are dimensions that seek to measure and describe the shape of the objects. These dimensions are independents from orientation, size, and location of the objects. Therefore two objects with the same shape that differ from each other only in orientation, size, and location should have similar values.

Probably, the most commonly used of these dimensions is the compactness, which is defined as the ratio between the area of the objects and the area of a circle with the same perimeter [4]. The circle is used as a comparison, because it is the most compact object known. The compactness, as well as the roundness, takes the value 1 for circular objects and lesser values for non-circular objects.

The convexity is defined as the ratio between the convex perimeter of an object and its perimeter [4]. The convex perimeter is the perimeter of the convex hull of an object that is defined as the smallest convex shape that contains the object. The convexity takes the value 1 for convex objects and lesser values for objects with irregular boundaries.

It is also provided a measure of roundness, which is defined as the ratio between the area of the object and the area of a circle with the same convex perimeter [4]. As well as the compactness this dimension takes the value 1 for circular objects and lesser values for non-circular objects.

2.3 Moments

The program also includes the determination of the 0th, 1st, and 2nd moments (meaning the area or the volume of the particle). These parameters are particularly useful because they can be used to specify the location and spatial distribution of an object. The 1st order moments reflect the position of the mass centre of the object, and the 2nd order moments reflect the dispersion of the object in space [10].

3. Hardware Specifications

The images were acquired with a video camera *SONY AVC-D5CE CCD* (Japan) adapted to an *OLYMPUS SZ40* (Tokyo, Japan) binocular lens. The analogic images were then digitised into a 512×512 pixel matrix with 256 colours by the *DATA TRANSLATION DT-2851* (Marlboro MA, USA) frame grabber with the help of a program that store them in a byte array file. The images were, at the same time, displayed in a *SONY PVM 1440QM* monitor. The images are saved in a BMP file.

The *Imago* program takes approximately 5 minutes to run on images with 512×512 pixel matrix with a 15 objects average, in a PC with Pentium 133 MHz CPU and 32 Mbytes of RAM memory. This time will depend on the chosen dimensions, and it becomes larger under the option “Volume”, with a 6 minutes average time.

4. Program

Imago is composed by 45 ‘script’ routines programmed in MATLAB for Windows (version 4.2). *Imago* can be divided in 4 major parts: the first one is the image filtering, noise reduction and cut off boundary objects elimination; in the second the objects are identified and isolated; in the third phase all the dimensions are determined; in the fourth, the normal probability distribution of the areas of the objects is determined.

The results are saved in a text file, chosen by the user at the end of the program. This file contains the values for all the objects of the fractal dimensions determined, convexity, compactness, roundness, the 0th, 1st, and 2nd moments as well as the mean values of these dimensions. The mean areas of the peaks and their standard deviations determined in normal probability distribution of the size of the object and the co-ordinates of the graphic are also saved.

4.1 Image treatment

The *Imago* program only supports BMP (Windows bitmap) and TIFF (Tagged Image File Format) image files with 256 colours. When the program starts a file manager box is displayed and the user can then choose the image to be treated. The program, trough the different options provided to the user, displays menus and push buttons, which allows an immediate and intuitive understanding of the program.

The first stage is to scale the values from 1-256 (read from the image file) to 0-1, which allows the image to be visualised in MATLAB. Figure 2 represents a picture from the particle set #1 as it is displayed in MATLAB.

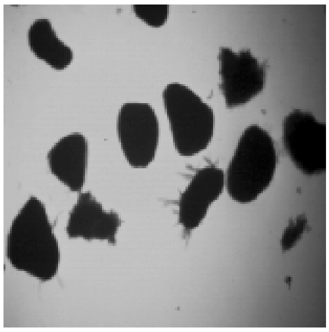

Figure 2. A picture from the particle set #1.

The Otsu’s filter [9] is used to calculate the best threshold in order to filter the image with a high recognition efficiency of the objects. The images acquired contained some light differences, in the background, within themselves. Part of the right side of the images was darken than the rest of the image. Due to those light differences, the image is first divided in columns and then each one is filtered alone. With that purpose there are used MATLAB’s own files that provided the image in 4 columns, large enough so that every column has at least one object (in real images), but small enough to avoid such light differences.

After that, in order to obtain a binary image, the value of the objects is set to 1 and the background is set to 0. Another matrix is also made with the real value of the objects and the background set to 0. This matrix is necessary to obtain the volume (in pixels) of the objects.

The elimination of boundary objects and the noise reduction is then accomplished. The last one removes isolated particles (noise) and separates objects that are in contact with each other by a few pixels. For this purpose it is used a MATLAB’s own function that can only be used in binary images. Figure 3 represents the final binary image obtained after treatment of the picture from Fig 2.

4.2 Identification and isolation of the objects.

The method used in *Imago* is a contour’s following method. The first stage consists in obtaining an image of the contour of the objects. To accomplish that it is used a MATLAB's own function, which originates a image with the contour of the object. However the contour of the objects provided is a 2 pixel wide contour and not a 1 pixel contour as expected. On cause of that some (few) objects with more irregular shapes are not identified.

Figure 3. Binary image of the picture from Fig 2.

The objects to be identified and isolated must not be separated by less than 3 pixels and with a minimum distance of 4 pixels from the border of the image. There is an option that provides manual identification of the objects, i.e. the user chooses the objects to be analysed. The user can this way, using the mouse, choose how many and which objects are to be analysed.

After an object is identified it is extracted to another matrix and all the dimensions are then determined.

4.3 Determination of the different dimensions

There are three possible options regarding the determination of the fractal dimensions: the determination of the area *vs.* Feret diameter fractal dimension, the determination of the area *vs.* perimeter fractal dimension or both. According to the users option the other fractal dimensions available are also determined. With the D_{AvP}, the area, and perimeter of the objects for each box size is calculated, and the dimension is then obtained by placing on graphic the log (area) values *vs.* the log (perimeter) values. It is required to know the area and Feret diameter values for each box size, when calculating the D_{AvDf}. The Feret diameter is the biggest Feret diameter calculated for box size 1 in 8 directions (22.5º, 45º, 77.5º, 90º, 112.5º, 135º, 167.5º, and 180º). The Feret diameters for each box size are then calculated in a straightforward way.

The box sizes (S) are automatically chosen by the program to minimise the deviations of the fractal dimension values with the size of the objects. The minimum size of an object, to be identified as one, is 25×25 pixels. For objects within 25×25 to 36×36 the S values are 1, 2, 4, and 8, for objects within 36×36 to 68×68 the S values are 2, 4, 8, and 16. If the objects are larger than 68×68 the values are 4, 8, 16, and 32. These values, when in logarithm scales, provide an equal spacing of the calculated plots.

The perimeter is calculated by a MATLAB's own function, and it can assume an 4 connected neighbourhood or an 8 connected neighbourhood. Neither of these two values obtained corresponds entirely to the real value of the perimeter; therefore the perimeter was determined by an equation using these two values. There are two options for determining the perimeter of the objects: one that uses the Euclidean geometry [4], or an empirical equation that has proven to obtain very accurate results.

There are two options for the determination of the 0th, 1st, and 2nd moments. In the option "Volume" the 0th order moment calculated, is the volume of the object; in the option "Area" the 0th order moment is the area of the object.

4.4 Normal probability distribution of the areas

After determining the dimensions of all the objects in the image, the program calculates the normal probability distribution of the areas of the objects and displays it in a graphic. In this process the program determines if the size distribution function of that image is unimodal or plurimodal, *i.e.*, if there is one or more peaks. A MATLAB's own function is used at this stage that runs the Student's t test for separating two sets of values. The mean areas of the peaks and its standard deviations are also provided.

5. Results

The *Imago* program was first tested in images with single objects to verify the agreement of the obtained values with the expected (theoretical) values. Those images (see Fig. 1) were GD1, GD2, and GD3 representing granules (real images), FLOC1, FLOC2, and FLOC3 representing flocs (real images) and binary images representing a circle, a square, and a hexagon.

The program was not able to identify automatically the FLOC1, FLOC2, and FLOC3 structures due to its size (extremely large) and highly irregular boundaries. The identification of these 3 images was made by the manual identification option.

The program was then applied to microbial aggregates with floculent structures taken from an anaerobic digestor (anaerobic filter). Four samples of particles concernig for different operating conditions were characterised. For each sample a set of images was processed (particle set #1 to particle set #4). A total of 94 images were analized.

5.1 Fractal Dimensions

Table I depicts the different fractal dimensions values for the images with single objects. As expected the higher values were achieved by the granular particles (GD1, GD2, and GD3) whereas the flocs presented lower values of D_{AvP}, D_{AvDf}, D_{Bm1}, and D_{Bm2}.

Table I - Fractal dimension values (single objects).

#	D_{Bm1}	D_{Bs}	D_{Bm2}	D_{AvP}	D_{AvDf}
GD1	1.90	1.03	2.01	1.85	1.95
GD2	1.88	0.99	2.00	1.90	1.97
GD3	1.91	1.01	2.01	1.89	1.96
FLOC1	1.72	1.30	1.90	1.31	1.73
FLOC2	1.81	1.15	1.99	1.57	1.87
FLOC3	1.84	1.12	1.97	1.64	1.84

Among the regular structures the D_{AvP} and D_{AvDf} should have the value 2. However, the fractal dimensions obtained were for the circle (D_{AvP} = 1.89 and D_{AvDf} = 1.96), for the square (D_{AvP} = 1.89 and D_{AvDf} = 1.98), and for the hexagon (D_{AvP} = 1.85 and D_{AvDf} = 1.92). The deviation from the value 2 can be explained by some loss of information of those structures, when represented in pixels and the inaccurate behaviour of the metric system formulas when applied to pixel representation. The square did not suffer any loss of information when in pixel representation and, in result of that, obtained the highest values.

Within the granular structures the highest values (using D_{AvDf}) were obtained by GD2 followed by GD3 and by GD1. The flocculent structures obtained the lower values for the fractal dimensions, and the most irregular of the three (FLOC1) obtained, as expected, the lower results.

The values obtained by the fractal dimensions D_{Bm1}, D_{Bs}, and D_{Bm2} were not as sensitive as the obtained by D_{AvP} and D_{AvDf}, although they clearly separate the regular and granular structures from the flocculent structures.

The application to four different samples of flocullent aggregates taken from two anaerobic filters, with different configurations, was investigated. Digester I was a single stage anaerobic filter and digester II was a three stage anaerobic filter. The particles were developed in different operating conditions. Samples #2 and #4 were withdrawn from the top section of each digester during a low load period (organic loading rate =4.5 Kg COD/m^3.day) and samples #3 and #1 were taken from the same points, but during a high load period (organic loading rate =13.5 Kg COD/m^3.day). Table II assigns the different samples, where the number in brackets means the total number of particles considered.

Table II - Sample identification

	Digester I (top section)	Digester II (top section)
High load	set #1 (190)	set #3 (149)
Low load	set #4 (158)	set #2 (119)

It was observed (Table III and IV) that particles developped in high load conditions were bigger and more regular than those developped in low load conditions.

Table III - Fractal dimensions
(± 99% Confidence interval)

#	D_{Bm1}	D_{Bs}	D_{Bm2}	D_{AvP}	D_{AvDf}
1	1.79±0.02	1.04±0.01	1.98±0.02	1.72±0.02	1.85±0.02
2	1.73±0.03	1.06±0.02	1.94±0.03	1.63±0.04	1.79±0.03
3	1.80±0.02	1.05±0.01	1.99±0.01	1.72±0.03	1.86±0.02
4	1.74±0.02	1.04±0.01	1.95±0.02	1.67±0.03	1.79±0.02

These bigger particles were morphologically similar to the granular structures described before. All the fractal dimension calculated showed the same trend for the different samples.

Table IV - Average areas
(± 99% Confidence interval)

#	Area (mm^2)
1	12.1±1.7
2	5.4±1.7
3	12.8±2.1
4	3.6±0.6

A statistical analysis (t-test) revealed that differences between sets #1 and #4 and between sets #3 and #2 were highly significant (p value <0.0001). However, no significant differences were obtained between sets #1 and #3 and sets #2 and #4. That means that loading conditions imposed a strong effect on aggregate morphology, but, for the same applied organic loading rate, digester configuration had no effect on this parameter.

5.2 Convexity, Compactness, and Roundness

The values of convexity and compactness, for the "Euclidean Geometry" option were, in one or two structures slightly higher than 1. This is explained by some imprecision on the determination of both the perimeter and the convex perimeter of the objects. However, the results proved to be very accurate for the convexity, the compactness, and the roundness in the "Euclidean Geometry" option and even better in the "More precise results" option.

The regular structures obtained convexity values of approximately 1, as expected, and the granular structures values nearby the unity, reflecting almost convex structures that is indeed the case. The flocculent structures obtained lower values of convexity, due to their irregular boundaries. The values obtained for the 'Euclidian' option are presented in table V.

Table V - Geometric dimension values (Euclidian).

#	Convexity	Compactness	Roundness
GD1	1.034	0.961	0.899
GD2	1.041	0.982	0.906
GD3	1.023	0.871	0.831
FLOC1	0.271	0.037	0.505
FLOC2	0.523	0.173	0.633
FLOC3	0.674	0.302	0.664

Table VI present the values obtained with the 'More precise' option.

Table VI- Geometric dimension values (More precise).

#	Convexity	Compactness	Roundness
GD1	0.987	0.876	0.900
GD2	0.997	0.900	0.906
GD3	0.978	0.796	0.831
FLOC1	0.258	0.034	0.506
FLOC2	0.513	0.168	0.639
FLOC3	0.643	0.275	0.664

The highest values of roundness and compactness were obtained for the circle, once more. The granular and regular structures obtained the higher values and the flocculent structures the lowers reflecting faithfully their nature. Within the regular structures, the circle obtained the highest value, followed by the hexagon and by the square, as expected.

5.3 Moments

The 0th moment in the "Area" option, represents the area of the object, and their results were 99.8% coincident with the results of a MATLAB's own function to calculate the areas. All the other moments calculated were found to be correct in both "Area" and "Volume" options.

6. Conclusions

The values of fractal dimensions obtained in the program are slightly influenced by the size of the object, which results in a slight underestimation for considerably small objects. This occurs because the smaller the object is, more imprecise is the measurement of its dimensions. That can be explained by the poorer pixel representation of the real object and essentially by the high ratio between the bigger box size used and the size of the object itself.

The identification and isolation method were found to be effective when applied in real images as well as the noise reduction and the elimination of boundary objects. The program identified 540 objects in a total of 605 objects in images obtained from an anaerobic digestor, which results in a 90 recognition percentage. Considering that every image with more than two non recognised objects should be treated manually, that would correspond to 5% of the total. This means that 95% of the images are successfully treated in the automatic identification option.

The program was successfully applied to the morphological characterisation of four sets of particles withdrawn from two different anaerobic digesters. It was possible to identify morphological differences between samples taken at low and high loading rate.

7. References

[1] M.A. Aon, S. Cortassa, On Fractal Nature of Cytoplasm. *FEBS Letters*, 3344, 1-4, 1994.

[2] M. Bellouti, M.M. Alves, J.M. Novais, M.Mota, Flocs versus Granules: Differentiation by Fractal Dimension, *Wat. Res. (in press).*

[3] D.P. Donnelly, M.F. Wilkins & L. Boddy, An Integrated Image Analysis Approach for Determining Biomass, Radial Extent and Box-Count Fractal Dimension of Macroscopic Mycelial Systems, *Binary*, 7, 19-27, 1995.

[4] C.A. Glasbey and G.W. Horgan, Image Analysis for the Biological Sciences, John Willey and Sons, Chichester, 1994.

[5] C.L. Jones, G.T Lonergan, D.E. Mainwaring, Mycelial Fragment Size Distribution: an Analysis Based on Fractal Geometry. *App. Microbiol. Biotechnol.* 39, 242-249, 1993.

[6] D.H. Li, J.J. Ganczarczyk, Fractal Geometry of Particle Aggregates Generated in Water and Wastewater Treatment Processes. *Environ. Sci. Technol.* 23, 1385-1389, 1989.

[7] B.E. Logan, J.R. Kilps, Fractal Dimensions of Aggregates Formed in Different Fluid Mechanical Environments. *Wat. Res.*, 29:2, 443-453, 1995.

[8] B.B. Mandelbrot, *The Fractal Geometry of Nature*. W.H. Freeman and Company, New York, 1982.

[9] N. Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Trans. in Systems, Man, and Cybernetics, SMC-9, 62-66, 1979.

[10] M.N. Pons, H. Mona, J.F. Drouin, H. Vivier, Texture Characterisation of Colonies on Solid Substrate, in Preprints of the 6th Int. Conference on Computer Applications in Biotechnology, Garmisch-Partenkirchen, 189-194, 1995.

[11] S.E. Vecht-Lifshitz, A.P. Ison, Biotechnological Applications of Image Analysis: Present and Future Prospects, *J. of Biotechnology*, 23, 1-18, 1992.