Differential properties of rough fractal surfaces

C Poull, C Gentil, C Roudet, L Druoton, M Roy

Journées Françaises de l'Informatique Graphique 2024 / N. Faraj and C. Schreck.
(Paper Chairs)

Differential properties of rough fractal surfaces

C. Poull C. Gentil C. Roudet L. Druoton and M. Roy

Laboratoire d'informatique de Bourgogne

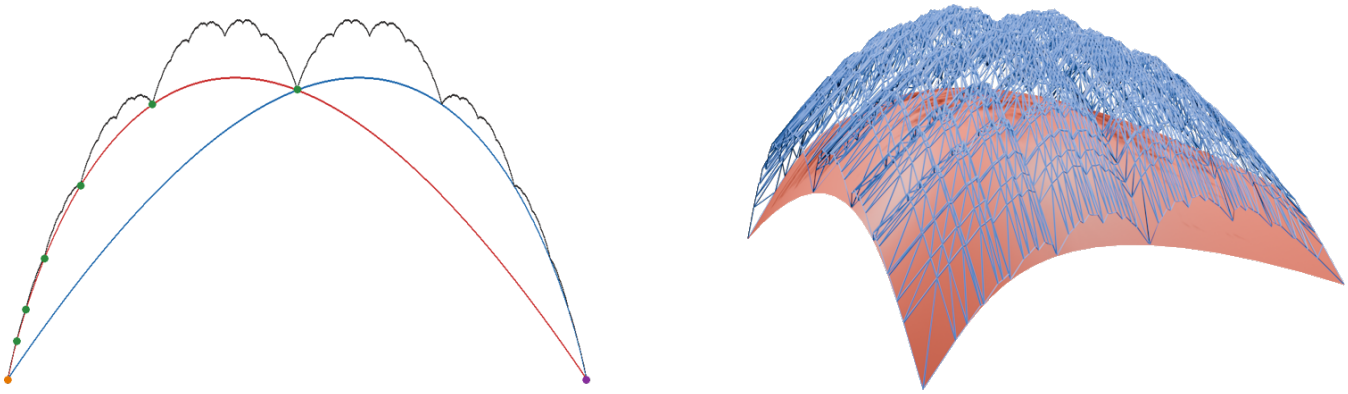

Figure 1: Left: Two Differential Characteristic Functions (DCF) in red and blue of a Takagi curve in black. This concept was introduced in [JGRP24] and is extended to tensor product surfaces in this paper.
Right: The tensor product of two Takagi curves is illustrated in blue wireframe, together with one of its SDCF in red.

Abstract

*Rough surfaces have many applications in industry or computer graphics, including quality control, CAD, texture generation and terrain synthesis. Analysing and controlling such surfaces can be tedious, making it difficult to obtain a desired roughness. We introduce the Surface Differential Characteristic Function (*SDCF*), an analytical form that helps characterising and analysing the differential properties of particular non differentiable fractal surfaces generated by tensor product of iterated function systems (*IFS*). The* IFS *allows the generation of self-similar multiscale objects, encompassing a large variety of possible roughness. The* SDCF *approach is an extension of the Differential Characteistic Function* DCF *model that was defined for analysing and characterising fractal curves. We use the* SDCF *tp compute the pseudo-curvatures. For smooth models, it corresponds to the curvatures obtained with classical approaches, but for fractal models, we get ranges of curvature due to the complex geometry. The self-similarity property of the surface results in these curvature to be transformed on all dyadic point in a deterministic way.*

CCS Concepts

•***Computing methodologies*** → ***Parametric curve and surface models;*** •***Mathematics of computing*** → *Differential calculus;*

1. Introduction

Roughness has many applications in both industrial and computer graphics contexts. Often shunned as a defect in CAD, it is also sought after for its various properties: complex tribology (friction, lubrication, ...), high thermal exchange, diffuse lighting... It is also considered an important parameter for quality control, be it at the end of a manufacturing chain to assess the quality of machined parts or to monitor the wear and tear of road surfaces. Computer graphics have been using roughness for a long time to model terrains [GGP*19], generate textures [DLY*20] or, more recently, to simulate the interaction between light and surface [ZZX*22]. Various methods exist for synthesising roughness [PJD*22]: procedural noises [LLC*10], simulations [STBB14], point processes [GAD*20]... Unfortunately, many methods lack geometric control, both global and local, or require user interaction [SPF*23], making it difficult to control the generated roughness precisely. For example, using the well known Perlin Noise to generate terrains, it is impossible to specify where we want a valley and where we want

a mountain. We can only chance upon the desired geometry by changing the seed of the random number generator it uses. In order to remedy this lack of geometric control, we propose to use a deterministic model that produces self-similar multiscale roughness, as it is an omnipresent property in roughness analysis contexts. We also propose to analyse the differential properties of this model in order to discriminate between orders of roughness. Our approach allows computing the pseudo-curvature of self-similar surfaces. For each fixed point of the transformations of an IFS, we compute a range of pseudo-curvatures on their SDCF.

We start by a reminder on the concepts required for understanding our approach, including the notion of differential characteristic function from Janbein et al [JGRP24], then we present how to extend the properties from curves to surface, and finally we highlight how to compute curvature on these surfaces.

2. Background

We introduce necessary notions for the study of fractal models and their curvature: first the Iterated Function System (IFS), a common model for generating fractals. Then, we present the Projected Iterated Function System (P-IFS) model, that allows free form deformations of the shapes generated by an IFS. Finally, we showcase the Differential Characteristic Function (DCF), a differential geometry approach to roughness analysis of IFS generated curves.

2.1. IFS

An Iterated Function System (IFS) [Hut81, Bar88], is a finite set of contractive operators $\mathbb{T} = \{T_i : \mathbb{X} \mapsto \mathbb{X}\}_{i=0}^{I-1}$ where $(\mathbb{X}, d)$ is a complete metric space, typically $\mathbb{X}$ is $\mathbb{R}^2$ or $\mathbb{R}^3$ and d is the euclidean distance. We call $\mathbb{X}$ the modeling space. Each operator can be represented as a transformation in the form of a matrix. The eigenvalues and eigenvectors of a transformation T are denoted by the pair $(\lambda_i, \boldsymbol{\nu}_i)$. The Hutchinson operator $\mathbb{T}(K)$ consists in applying all operators T_i to K, an arbitrary non-empty subset of compacts of $\mathbb{X}$: $\mathbb{T}(K) = \bigcup_{i=0}^{I-1} T_i K$. Banach fixed-point theorem [Ban22] states that there exists a unique non empty compact $\mathcal{A}$ of $\mathbb{X}$ such that it satisfies the self-similarity property: $\mathbb{T}(\mathcal{A}) = \mathcal{A}$. This fixed point $\mathcal{A}$ is called the attractor of $\mathbb{T}$, as it is the limit of iteratively applying the Hutchinson operator to K: $A = \lim_{n \mapsto \infty} \mathbb{T}^n(K)$. Note that the geometry of the attractor $\mathcal{A}$ does not depend on the choice of K, but only on the operators of $\mathbb{T}$. This approach allows the modeling of a large familly of self-similar objects, but not all are of interest to us, so we apply topological constraints on the operators T_i to ensures that the attractor is $C^{(0)}$ continuous, in other words, we are only interested in curves and surfaces. These adjacency constraints are similar to that used in Fractal Interpolation Functions (FIF) [Bar86] : $T_0 c_1 = T_1 c_0$ where c_i is the fixed point of T_i. Without these constraints, the attractor of an IFS can be analogous to a Cantor set.

2.2. Barycentric space and projected IFS

A barycentric coordinate system $\mathbb{B}^N$ is a coordinate system whose points are defined as weight vectors of dimension N. Any point ω of $\mathbb{B}^N$ can then be projected to another space using a set of N control points $P\omega = \sum_{i=0}^{N-1} \omega_i P_i$ where ω_i is the i^{th} element of ω. The points of $\mathbb{B}^N$ are constrained to normalised barycentric coordinates: $\sum_{i=0}^{N} \omega_i = 1$. A vector space is defined over of the barycentric space $\mathbb{B}^N$, and the vectors are constrained to homogeneous barycentric coordinates: for any vector $\boldsymbol{\omega}$ of the barycentric vector space, we have $\sum_{i=0}^{N} \boldsymbol{\omega}_i = 0$ where $\boldsymbol{\omega}_i$ is the i^{th} element of $\boldsymbol{\omega}$.

An extension of the IFS model was presented by Zair et al. in [ZT96] as Projected Iterated Function System (P-IFS) in order to allow free-form deformations of the attractor, akin to Bezier curves and NURBS. If we use operators defined in $\mathbb{B}^N$, and a set of N control points $P = \{P_i\}_{i=0}^{N-1}$, we can have better control on the global geometry of the attractor. This is illustrated in Figures 2 and 3 who both have the same operators, but their attractors (in black) are projected in the modeling space using a different set of control points. Zair et al. [ZT96] actually showed that P-IFS can be used to model Bezier curves: a P-IFS with De Casteljau matrices as operators results in Bernstein polynomials as attractor, which can then be projected in $\mathbb{X}$.

2.3. DCF

Because an attractor is constructed by iteratively applying recursive operators, a sequences of points is constructed by iteratively applying a single operator T to a starting point q of $\mathbb{B}^N$. The DCF, defined by Janbein et al. in [JGRP24], is a parametric function that interpolates the path q takes as one iteratively apply T, thus it allows the capture of the differential properties of the sequence of points. It leverages the work by Bensoudane et al. [Ben09] and Podkorytov et al. [Pod13] that defines pseudo-tangents for IFS-generated fractals, allowing first-order continuity constraints on fractal curves and surfaces. For any operator T and starting point q, the DCF is defined as $\mathrm{DCF}(T, q, t) = \sum_{i=0}^{N-1} x_i \boldsymbol{\nu}_i t^{\alpha_i}$ with x_i the coordinates of q in the eigenbasis of T and $\alpha_i = \frac{\log(|\lambda_i|)}{\log(|\lambda_1|)}$. Note that this form is a reparametrization of the one in [JGRP24], which can be obtained with $\mathrm{DCF}(T, q, \frac{t}{x_1})$. This parametric representation is used as a way to compute the curvature of a fractal curve at an extremity (the fixed point of the first operator). Due to the fractal nature of the attractor, there is not only a single DCF, but a family of DCF when all the points of the attractor are considered as starting points. The key point of this approach is that there is a range of curvature, computed from the familly of DCF. There are three cases depending on the value of α_2:

- $\alpha_2 < 2$ there is a single DCF with infinite curvature
- $\alpha_2 = 2$ there are multiple DCF, resulting in a range of curvature
- $\alpha_2 > 2$ there is a single DCF with null curvature

For differentiable curves such as Bezier curves, there is a single DCF that is superimposed with the attractor, resulting in a single value for curvature.

3. Surface Differential Characteristic Function

We aim to show that the DCF of surfaces generated by tensor product of two IFS behaves in the same way as the DCF for curves, then we introduce the surface DCF, a bivariate function that is analogous to DCF but for tensor product IFS.

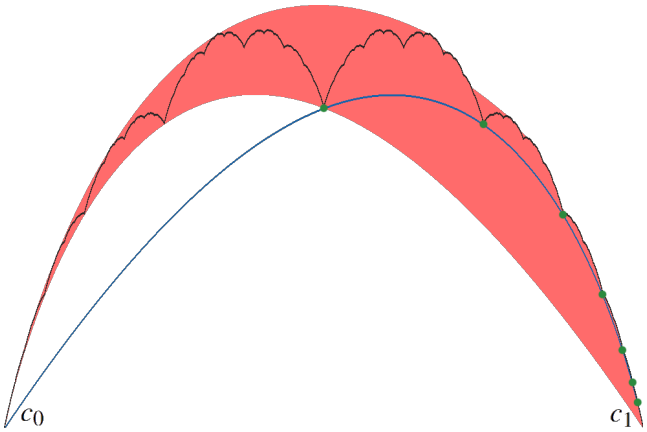

Figure 2: A P-IFS whose attractor (in black) is the Takagi curve. c_0 is the fixed point T_0, c_1 is the fixed point of T_1. The DCF of T_1 with starting point c_0 is shown in blue. The familly of DCF of T_0 we obtain when taking all the points of the attractor as starting points is shown in red. The green sequence of points are the points of the attractor obtained with $T_1^n c_0$. The set of control points for the projection is is $\{(0,0),(0,1),(1,1),(1,0)\}$.

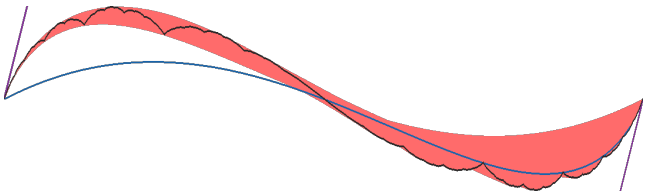

Figure 3: The same P-IFS as in Figure 2 with a different projection. The tangents of the transformations are shown in purple. The set of control points for the projection is is $\{(0,0),(0,1),(1,-1),(1,0)\}$.

3.1. Tensor Product

Given two P-IFS $\mathbb{T} = \{T_i : \mathbb{B}^N \mapsto \mathbb{B}^N\}_{i=0}^{I-1}$ and $\mathbb{T}' = \{T'_j : \mathbb{B}^M \mapsto \mathbb{B}^M\}_{j=0}^{J-1}$, we can construct a new P-IFS $\mathbb{T}^{\otimes}$ using the tensor product: $\mathbb{T}^{\otimes} = \mathbb{T} \otimes \mathbb{T}' = \{\mathcal{T}_{ij} : \mathbb{B}^{NM} \mapsto \mathbb{B}^{NM}\}_{i=0,j=0}^{I-1,J-1}$ where $\mathcal{T}_{ij} = T_i \otimes T'_j$. The attractor $\mathcal{A}$ of this new P-IFS is the tensor product of the attractors of $\mathbb{T}$ and $\mathbb{T}'$ [Zai98]. This attractor requires a grid of $N \times M$ control points to be projected to the modeling space $\mathbb{X}$. An example of attractor obtained from this process is shown in red in Figure 4.

We can use the definition of the DCF directly on tensor products of P-IFS, as the tensor product of two P-IFS is also an P-IFS. We aim to compute the DCF of the operator $\mathcal{T} = T \otimes T'$ at starting point $\mathcal{Q} = q \otimes q'$. We have Λ_k the eigenvalues of $\mathcal{T}$ and their associated eigenvectors $\boldsymbol{V}_k$. Note that each pair (λ_i, λ_j) corresponds to a unique Λ_k such that $\Lambda_k = \lambda_i \lambda_j$, and similarly, each pair $(\boldsymbol{v}_i, \boldsymbol{v}'_j)$ corresponds to a unique $\boldsymbol{V}_k$ such that $\boldsymbol{V}_k = \boldsymbol{v}_i \otimes \boldsymbol{v}'_i$. We order the eigenvalues such that they are of decreasing modulus. Similarly, we have $X_k = x_i x'_j$ the coordinates of $\mathcal{Q}$ in the eigenbasis of $\mathcal{T}$. As it was done for curves, we use $A_i = \frac{\log(|\Lambda_i|)}{\log(|\Lambda_1|)}$ and $A'_j = \frac{\log(|\Lambda'_j|)}{\log(|\Lambda'_1|)}$.

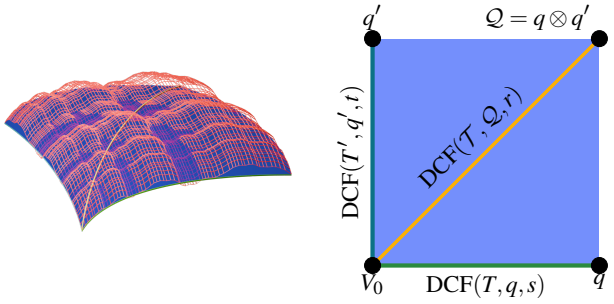

Figure 4: The attractor of a projected IFS that is the tensor product of two P-IFS $\mathbb{T}$ and $\mathbb{T}'$ is represented in wireframe (red). The DCF of T_0 and T'_0 are in green and cyan respectively. The DCF of $\mathcal{T}_{00} = T_0 \otimes T'_0$ is represented in yellow. Finally, the SDCF of $\mathcal{T}_{00}$ is represented in blue. Note that all 3 DCF are included in the SDCF.

$$\mathrm{DCF}(\mathcal{T}, \mathcal{Q}, t) = \sum_{k=0}^{NM-1} X_k \boldsymbol{V}_k t^{A_k} = \sum_{i=0}^{N} \sum_{i=0}^{N} x_i x'_j \boldsymbol{v}_i \otimes \boldsymbol{v}'_j t^{\alpha_i \alpha'_j}.$$

A DCF computed on a tensor product of P-IFS is illustrated in yellow in Figure 4.

We define the surface differential characteristic function (SDCF) of an operator $\mathcal{T} = T \otimes T'$ as the tensor product of the DCF of the operators T and T'.

$$\begin{aligned}\mathrm{SDCF}(\mathcal{T}, \mathcal{Q}, s, t) &= \mathrm{DCF}(T, q, s) \otimes \mathrm{DCF}(T', q', t) \\ &= \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} x_i x'_j \boldsymbol{v}_i \otimes \boldsymbol{v}'_j s^{\alpha_i} t^{\alpha'_j}\end{aligned}$$

A SDCF obtained with this formula for one of the 4 operators of the P-IFS is illustrated in Figure 4 as the blue surface. It converges to the fixed point of $\mathcal{T}_{I,J} = T_0 \otimes T'_0$ (bottom left corner) and emerges from the fixed point of $\mathcal{T}_{1,1}$ (top right corner).

4. Range of Curvatures

We first focus on the curvature at the fixed points of the transformations. Since the attractor is built as an iterative process of transformations, if we know a property at the fixed point, we can compute it on any dyadic point of the attractor. For surfaces, there exists multiple definition of curvature. We compute the Gaussian curvature of the SDCF.

4.1. Curvature of SDCF

For brevity, we introduce the following notation: $\boldsymbol{D}_{i,j} = x_i x'_j \cdot \boldsymbol{v}_i \otimes \boldsymbol{v}'_j$, corresponding to the part of the *SDCF* formula that is independent of the variables. Using the common definition of curvature for surfaces, we can compute its limit for the second derivative of the SDCF as our parameters s and t approach to 0, in other word at the fixed point of the transformation. We denote $\boldsymbol{n}_{s,t}$ the normal of the surface at (s,t). Note that this normal is computed in the modeling space. We remind the formulas for the first and second fundamental forms, used to compute curvatures on surfaces, and the formulas for

the gaussian curvature $\mathcal{K}$, the mean curvature $\mathcal{H}$ and the principal curvatures K_1 and K_2 (denoted $K_\square$).

$$E = \frac{\partial \mathrm{SDCF}(s,t)}{\partial s}^2,\ F = \frac{\partial \mathrm{SDCF}(s,t)}{\partial s} \cdot \frac{\partial \mathrm{SDCF}(s,t)}{\partial t},\ G = \frac{\partial \mathrm{SDCF}(s,t)}{\partial t}^2$$

$$L = \frac{\partial^2 \mathrm{SDCF}(s,t)}{\partial s^2} \cdot \boldsymbol{n},\ M = \frac{\partial^2 \mathrm{SDCF}(s,t)}{\partial s \partial t} \cdot \boldsymbol{n},\ N = \frac{\partial^2 \mathrm{SDCF}(s,t)}{\partial t^2} \cdot \boldsymbol{n}$$

$$\mathcal{K} = \frac{LN - M^2}{EG - F^2}$$

$$\mathcal{H} = \frac{EN + GL - 2FM}{2(EG - F^2)}$$

$$K_\square = \frac{L + N \pm \sqrt{L^2 + 4M^2 + N^2 - 2LN}}{2}$$

As the SDCF is an approximation of the fractal surface at $s = 0$ and $t = 0$, we compute the limit of the normal and derivatives:

$$lim_{(s,t)\to(0,0)} \boldsymbol{n}_{s,t} = \boldsymbol{n}_{0,0} = \frac{P\boldsymbol{D}_{1,0} \times P\boldsymbol{D}_{0,1}}{||P\boldsymbol{D}_{1,0} \times P\boldsymbol{D}_{0,1}||}$$

$$\lim_{(s,t)\to(0,0)} \frac{\partial \mathrm{SDCF}(\mathcal{T},\mathcal{Q},s,t)}{\partial s} = \boldsymbol{D}_{1,0}$$

$$\lim_{(s,t)\to(0,0)} \frac{\partial \mathrm{SDCF}(\mathcal{T},\mathcal{Q},s,t)}{\partial t} = \boldsymbol{D}_{0,1}$$

$$\lim_{(s,t)\to(0,0)} \frac{\partial \mathrm{SDCF}(\mathcal{T},\mathcal{Q},s,t)}{\partial s \partial t} = \boldsymbol{D}_{1,1}$$

$$\lim_{(s,t)\to(0,0)} \frac{\partial^2 \mathrm{SDCF}(\mathcal{T},\mathcal{Q},s,t)}{\partial s^2} = \lim_{s\to 0} \alpha_2(\alpha_2 - 1)\boldsymbol{D}_{2,0} s^{\alpha_2 - 2}$$

$$\lim_{(s,t)\to(0,0)} \frac{\partial^2 \mathrm{SDCF}(\mathcal{T},\mathcal{Q},s,t)}{\partial t^2} = \lim_{t\to 0} \alpha_2'(\alpha_2' - 1)\boldsymbol{D}_{0,2} t^{\alpha_2' - 2}$$

We introduce the following notations:

$$\mathcal{K}_s(s,t) = P\alpha_2(\alpha_2 - 1)\boldsymbol{D}_{2,0} s^{\alpha_2 - 2}$$

$$\mathcal{K}_t(s,t) = P\alpha_2'(\alpha_2' - 1)\boldsymbol{D}_{0,2} t^{\alpha_2' - 2}$$

The computation of the gaussian curvature of the SDCF at the point $(0,0)$ is expressed as follows:

$$\lim_{(s,t)\to(0,0)} \frac{\mathcal{K}_s(s,t) \cdot \boldsymbol{n}_{0,0} \cdot \mathcal{K}_t(s,t) \cdot \boldsymbol{n}_{0,0} - (P\boldsymbol{D}_{1,1} . \boldsymbol{n}_{0,0})^2}{(P\boldsymbol{D}_{1,0})^2 \cdot (P\boldsymbol{D}_{0,1})^2 - (P\boldsymbol{D}_{1,0} \cdot P\boldsymbol{D}_{0,1})^2}$$

We have 3 cases for the Gaussian Curvature of each operator:

- $\alpha_2 < 2$: the first term $\alpha_2(\alpha_2 - 1)\boldsymbol{D}_{2,0} s^{\alpha_2 - 2}$ is infinite and all other terms don't matter ($\lim_{s\to 0} s^{\alpha_i - 2} = \infty$)
- $\alpha_2 = 2$: the first term $\alpha_2(\alpha_2 - 1)\boldsymbol{D}_{2,0} s^{\alpha_2 - 2}$ is a constant and all other terms are either constant or null.
- $\alpha_2 > 2$ all terms are zero ($\alpha_i - 2$ will always be positive so $\lim_{s\to 0} s^{\alpha_i - 2} = 0$)

Assuming we don't have the degenerate case with linearly dependent vectors (pinched corner), we have 9 possibilities for the gaussian curvature:

$\lim_{(s,t)\to(0,0)} \mathcal{K}(s,t)$	$\alpha_2 < 2$	$\alpha_2 = 2$	$\alpha_2 > 2$
$\alpha_2' < 2$	$\pm\infty$	$\pm\infty$	indefinite
$\alpha_2' = 2$	$\pm\infty$	$\mathcal{C}$	$\mathcal{C}$
$\alpha_2' > 2$	indefinite	$\mathcal{C}$	$\mathcal{C}$

where $\mathcal{C}$ is a finite value of $\mathbb{R}$.

We can compute the mean curvature and the principal curvatures with the same reasoning.

4.2. Curvatures of an attractor

Just as it was for curves, the fractal nature of our attractors entails a familly of SDCF that leads to a range of curvature for each different type of curvature we can compute. For curves, the familly of DCF gave an area that bounded the attractor. For surfaces, we have a familly of *SDCF* that gives a hull that bound the attractor. The lower and upper bound can be computed numerically as the tensor product of the lower bound DCF and the upper bound DCF of T and T'. Thus, we have a range of gaussian pseudo-curvature, mean pseudo-curvature and principal pseudo-curvatures for all operators.

5. Conclusion

In this paper, we have extended the definition of pseudo-curvature from fractal curves to tensor product fractal surfaces. This was done through the definition of the surface differential characteristic function as the tensor product of the two DCF of the curves from which the attractor was formed. This allows to compute curvatures on a fractal attractor generated by a tensor product of two P-IFS. Unlike for smooth surfaces that have a single value of curvatures per point, fractal curves have ranges of pseudo-curvature. For Bezier surfaces, there is a single SDCF that is superimposed to the Bezier attractor.

6. Perspectives

These ranges of pseudo-curvature will allow the characterisation of the nature of a fractal surface at any dyadic point: concave/convex ellipsoid, cylindrical, hyperboloid... It can also be used to specify constraints to enforce second order roughness. We are also interested in studying the DCF, resp SDCF, of P-IFS with more than two, resp four, transformations. A similar approach could also be considered for non tensor-product surfaces.

7. Acknowledgements

This work benefited from the support of the project FRACLETTES ANR-20-CE46-0003 of the French National Research Agency (ANR).

References

[Ban22] BANACH S.: Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae 3* (1922), 133–181. doi:10.4064/fm-3-1-133-181. 2

[Bar86] BARNSLEY M. F.: Fractal functions and interpolation. *Constructive Approximation 2*, 1 (Dec 1986), 303–329. URL: https://doi.org/10.1007/BF01893434, doi:10.1007/BF01893434. 2

[Bar88] BARNSLEY M. F.: *Fractals Everywhere*. Dover Publications, Inc., 1988. doi:10.1016/C2013-0-10335-2. 2

[Ben09] BENSOUDANE H.: *Etude différentielle des formes fractales*. PhD thesis, Université de Bourgogne, 2009. 2

[DLY*20] DONG J., LIU J., YAO K., CHANTLER M., QI L., YU H., JIAN M.: Survey of procedural methods for two-dimensional texture generation. *Sensors 20*, 4 (2020). URL: https://www.mdpi.com/1424-8220/20/4/1135, doi:10.3390/s20041135. 1

[GAD*20] GUEHL P., ALLÈGRE R., DISCHLER J.-M., BENES B., GALIN E.: Semi-Procedural Textures Using Point Process Texture Basis Functions. *Computer Graphics Forum* (2020). doi:10.1111/cgf.14061. 1

[GGP*19] GALIN E., GUÉRIN E., PEYTAVIE A., CORDONNIER G., CANI M.-P., BENES B., GAIN J.: A review of digital terrain modeling. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 553–577. 1

[Hut81] HUTCHINSON J. E.: Fractals and self similarity. *Indiana University Mathematics Journal 30* (1981), 713–747. doi:10.1512/iumj.1981.30.30055. 2

[JGRP24] JANBEIN M., GENTIL C., ROUDET C., POULL C.: Pseudo-curvature of fractal curves for geometric control of roughness. In *19th International Conference on Computer Graphics Theory and Applications* (Rome, Italy, Feb. 2024), vol. 1 of *GRAPP, HUCAPP and IVAPP*, pp. 177–188. URL: https://hal.science/hal-04474390. 1, 2

[LLC*10] LAGAE A., LEFEBVRE S., COOK R., DEROSE T., DRETTAKIS G., EBERT D. S., LEWIS J. P., PERLIN K., ZWICKER M.: A survey of procedural noise functions. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 2579–2600. 1

[PJD*22] POULL C., JANBEIN M., DRUOTON L., ROUDET C., LANQUETIN S., ROY M., GENTIL C.: La rugosité des surfaces et ses applications. In *Journées Françaises d'Informatique Graphique (JFIG 2022)* (Bordeaux, France, Nov. 2022). URL: https://hal.science/hal-03892076. 1

[Pod13] PODKORYTOV S.: *Espaces tangents pour les formes autosimilaires*. PhD thesis, Université de Bourgogne, 2013. 2

[SPF*23] SCHOTT H., PARIS A., FOURNIER L., GUÉRIN E., GALIN E.: Large-scale terrain authoring through interactive erosion simulation. *ACM Transactions on Graphics 42*, 5 (2023), 15. URL: https://hal.science/hal-04049125, doi:10.1145/3592787. 1

[STBB14] SMELIK R. M., TUTENEL T., BIDARRA R., BENES B.: A survey on procedural modelling for virtual worlds. *Comput. Graph. Forum 33*, 6 (Sept. 2014), 31–50. URL: https://doi.org/10.1111/cgf.12276, doi:10.1111/cgf.12276. 1

[Zai98] ZAIR C. E.: *Formes fractales à pôles basées sur une généralisation des IFS*. PhD thesis, 1998. 3

[ZT96] ZAIR C. E., TOSAN E.: Fractal modeling using free form techniques. In *Computer Graphics Forum* (1996), vol. 15, Wiley Online Library, pp. 269–278. doi:10.1111/1467-8659.1530269. 2

[ZZX*22] ZHU J., ZHAO S., XU Y., MENG X., WANG L., YAN L.-Q.: Recent advances in glinty appearance rendering. *Computational Visual Media 8*, 4 (2022), 535–552. URL: https://www.sciopen.com/article/10.1007/s41095-022-0280-x, doi:10.1007/s41095-022-0280-x. 1