The Estimation of Hölderian Regularity using Genetic Programming

Leonardo Trujillo
Instituto Tecnológico de Tijuana
Tijuana, B.C., México
leonardo.trujillo.ttl@gmail.com

Pierrick Legrand
Université Victor Segalen Bordeaux 2
ALEA Team, INRIA Bordeaux Sud-Ouest
IMB, UMR CNRS 5251, France.
pierrick.legrand@u-bordeaux2.fr

Jacques Lévy-Véhel
REGULARITY Team, INRIA Saclay
Ile de France, France
jacques.levy-vehel@inria.fr

ABSTRACT

This paper presents a Genetic Programming (GP) approach to synthesize estimators for the pointwise Hölder exponent in 2D signals. It is known that irregularities and singularities are the most salient and informative parts of a signal. Hence, explicitly measuring these variations can be important in various domains of signal processing. The pointwise Hölder exponent provides a characterization of these types of features. However, current methods for estimation cannot be considered to be optimal in any sense. Therefore, the goal of this work is to automatically synthesize operators that provide an estimation for the Hölderian regularity in a 2D signal. This goal is posed as an optimization problem in which we attempt to minimize the error between a prescribed regularity and the estimated regularity given by an image operator. The search for optimal estimators is then carried out using a GP algorithm. Experiments confirm that the GP-operators produce a good estimation of the Hölder exponent in images of multifractional Brownian motions. In fact, the evolved estimators significantly outperform a traditional method by as much as one order of magnitude. These results provide further empirical evidence that GP can solve difficult problems of applied mathematics.

Categories and Subject Descriptors

I.2.2 [**Artificial Intelligence**]: Automatic Programming—*program synthesis*

General Terms

Algorithms, Experimentation, Performance

Keywords

Signal regularity, Hölder exponent, Genetic Programming

1. INTRODUCTION

The fields of signal processing and pattern recognition are primarily concerned with analyzing and understanding the information contained within complex signals or data patterns. This paper deals with the concept of Hölderian regularity which is used to analyze prominent signal variations, and which is therefore relevant in both fields. Hölderian regularity, also known as Lipschitz regularity, characterizes the singularities contained within non-differentiable signals using local or pointwise exponents [15, 21]. This measure can effectively describe the structure of a signal around each point, a property that has made it quite useful in various tasks, for instance see [11, 12, 13, 24]. However, one drawback of Hölderian analysis is that Hölder exponents can only be computed in a closed form for a limited number of signal types. In order to overcome this, several estimation methods have been developed and are widely used with multifractal analysis, each based on strong mathematical principles and derived using necessary assumptions regarding the underlying properties of the signal that is analyzed [6, 10]. These estimators have proven to be useful tools, and an open-source toolbox exists that can be used to test these methods [14]. However, it is also important to understand that in practice these algorithms depend on the correct setting of several parameters, and other ad-hoc decisions are required to obtain a desired performance. These methods tend to be complex and relatively slow, making their use difficult in domains that require fast processing.

In the present work, the goal is to automatically synthesize operators that can estimate the Hölderian regularity using Genetic Programming (GP). In particular, we focus on estimating the pointwise Hölder exponent in 2D digital signals (images), primarily for the following reasons. First, this measure of regularity has proven to be a powerful tool for basic problems of image analysis, such as noise removal [12], interpolation [11], and edge detection [13], to mention but a few. Moreover, it achieved highly competitive performance in the problem of local image description [24], one of the most widely used procedures in current computer vision literature [16]. However, despite these succesful applications, the limitations of current estimation methods, described above, limit its broader use.

Given our stated goal, let us describe the manner in which we attempt to achieve it. First, we generate several groups of synthetic images of multifractional Brownian motion. All images of the same groups share the same prescribed regularity; i.e, the images have the same regularity at each point but show different intensity patterns. The regularity is prescribed using a known function and the intensity images are then automatically generated using the methods developed in [3]. Then, by using these images as reference we are able to pose an optimization problem where the goal is to find image operators that can estimate the pointwise Hölder exponent with a minimum amount of error. This problem could be solved in different ways, but in this work we have chosen to use GP, probably the most advanced form of evolutionary computation. GP has proven to be well suited for problems where a specialized mathematical expression is required but the general structure of the expression is difficult to define a priori [9]. The experimental results show that the estimators evolved through GP provide a superior estimation when compared with a traditional method, in several cases the difference is one order of magnitude.

1.1 Related Work

The proposal developed in this paper is closely related to recent applications of GP in the areas of mathematics and image analysis. First, regarding the latter, we can say that our proposal is concerned with extracting a specific type of image feature that can be computed for each point, namely their Hölderian regularity. If we take this view, we can observe a close relation with the works in [22, 23, 25], for example, where GP was used to synthesize image operators that determine the saliency of each image pixel. Another example is the work of [17], where GP is given the task of finding operators that optimize the description of local image regions, or the work in [26] that uses GP to detect edge points. Regarding the application of GP to mathematics, several interesting proposals have also been developed. The traditional examples are symbolic regression problems, originally descibed by Koza and further extended by works such as [7, 5]. However, recent applications in solving differential equations [2] and in the study of finite algebras [19] have shown that GP need not be limited only to regression analysis. From these examples, we would like to restate the idea that a GP algorithm, when appropriately used, can indeed be characterized as a tool for *automatic scientific discovery* [8].

Given our brief introduction, we proceed to outline the remainder of this paper. Section 2 formally defines the concept of Hölderian regularity and describes a canonical method used to estimate the pointwise exponent. Then, in Section 3 we present a formal definition of our problem and give a detailed description of our GP proposal. The experimental setup is described in Section 4, along with a description of the results obtained. Finally, Section 5 contains our concluding remarks.

2. HÖLDERIAN REGULARITY

It is well understood that singular and irregular structures contain the most prominent, and most useful, information within a signal. For example, in images large discontinuities often correspond with salient image features that can be used for recognition tasks [16]. Hölderian regularity is a manner in which to characterize precisely these singular structures [15, 24]. The regularity of a signal at each point can be quantified by the pointwise Hölder exponent, which we define below.

Definition 1:Let $f : \mathbb{R} \to \mathbb{R}$, $s \in \mathbb{R}^{+*} \setminus \mathbb{N}$ and $x_0 \in \mathbb{R}$. $f \in C^s(x_0)$ if and only if $\exists \eta \in \mathbb{R}^{+*}$, and a polynomial P of degree $< s$ and a constant c such that

$$\forall x \in B(x_0, \eta), |f(x) - P(x - x_0)| \leq c|x - x_0|^s \ , \quad (1)$$

where $B(x_0, \eta)$ is the local neighborhood around x_0 with a radius η. The pointwise Hölder exponent of f at x_0 is $\alpha_p(x_0) = \sup_s \{f \in C^s(x_0)\}$.

The concept of Hölderian regularity is closely related to the Taylor series approximation of a function. However, the Hölder exponents refines this concept by also accounting for non-differentiable points [15]. Figure 1 shows a graphical illustration of the Hölder exponent, depicted as an envelope, for a non-differentiable signal.

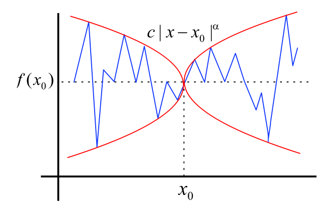


Figure 1: Hölderian envelope of signal f at point x_0.

The Hölder exponent, however, can only be computed analytically for a limited number of signal types. Therefore, in order to use Hölderian regularity the exponents must be estimated, and several numerical methods have been proposed for this purpose. The most direct is the oscillations method which is directly related to the definition given above [21].

2.1 Estimation through oscillations

The Hölder exponent of function $f(t)$ at point t is the $sup(\alpha_p) \in [0,1]$, for which a constant c exists such that $\forall\ t'$ in a neighborhood of t,

$$|f(t) - f(t')| \leq c|t - t'|^{\alpha_p} \ . \quad (2)$$

In terms of signal oscillations, a function $f(t)$ is Hölderian with exponent $\alpha_p \in [0,1]$ at t if $\exists c\ \forall \tau$ such that $osc_\tau(t) \leq c\tau^{\alpha_p}$, with

$$osc_\tau(t) = \sup_{t', t'' \in [t-\tau, t+\tau]} |f(t') - f(t'')| \ . \quad (3)$$

Now, if $t = x_0$ and $t' = x_0 + h$ in 2, we can also write that

$$\alpha_p(x_0) = \liminf_{h \to 0} \frac{\log|f(x_0 + h) - f(x_0)|}{\log|h|} \ . \quad (4)$$

Therefore, the problem is that of finding an α_p that satisfies 2 and 3, and in order to simplify this process we can set $\tau = \beta^r$. Then, we can write $osc_\tau \approx c\tau_p^\alpha = \beta^{(\alpha_p r + b)}$, which is equivalent to $log_\beta(osc_\tau) \approx \alpha_p r + b$.

Therefore, an estimation of the regularity can be built at each point by computing the slope of the regression between the logarithm of the oscillations osc_τ and the logarithm of the dimension of the neighborhood at which the oscillations τ are computed. Here, we use least squares regression to compute the slope, with $\beta = 2$ and $r = 1, 2, \ldots, 7$. Also, it is preferable not to use all sizes of neighborhoods between two values τ_{min} and τ_{max}. Hence, we calculate the oscillation at

(a) Original Image (b) Hölder Image

Figure 2: Estimation of the Hölder exponent using oscillations.

point x_0 only on intervals of the form $[x_0 - \tau_r : x_0 + \tau_r]$. For a 2D signal, x_0 defines a point in 2D space and τ_r a radius around x_0, such that $d(t', t) \leq \tau_r$ and $d(t'', t) \leq \tau_r$, where $d(a, b)$ is the Euclidean distance between a and b.

Figure 2 shows a visual example of the type of output this algorithm produces, it presents a sample image and the corresponding Hölder exponent for each pixel. This method has proven to be superior [10] in some cases to the wavelet leaders method [6], and useful in real-world applications [11, 12, 13, 24]. Therefore, we use the oscillations method for comparisons with our evolved estimators.

3. OUTLINE OF OUR PROPOSAL

Returning to the main goal of our work, the automatic synthesis of operators that estimate the pointwise Hölder exponent for 2D signals, we state the following formal problem.

3.1 Problem statement

Let I represent a digital 2D signal, or more specifically an image, and suppose that H_I is a matrix that contains the value of the pointwise Hölder exponent for every pixel in I. Then, we can pose the problem of finding an optimal operator K^o as follows,

$$K^o = \arg\min_K \{Err[K(I), H_I]\} \ , \tag{5}$$

where $Err[,]$ represents an error measure, which in this work is given by the root-mean-square error (RMSE)

$$Err[K(I), H_I] = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(K(x_i) - H_{x_i})^2} \ , \tag{6}$$

where N is the number of pixels in an image I.

Here H_I is prescribed by a function p, and for each such function it is possible to build an infinite number of images that share the same regularity (see Section 4). Therefore, the aim is to find the symbolic expression of an operator that minimizes the estimation error. Note that the goal is to obtain the best possible estimation, without accounting for other possible objectives such as computation time. We assume that this is the appropriate choice given the novelty of the problem, and leave a possible multi-objective formulations for future research. Nevertheless, we do account for computation time in an indirect manner, by enforcing size constraints on the search process through bloat control. In what follows, we briefly present the paradigm of GP and then describe the algorithm proposed for the stated problem.

3.2 Genetic Programming

Evolutionary computation has developed a rich variety of search and optimization algorithms that base their core functionality on the basic principles of Neo-Darwinian evolution [4]. These techniques are population-based meta-heuristics, where candidate solutions are stochastically selected and modified in order to produce new, and possibly better, solutions for a particular problem. The selection process favors individuals that exhibit the best performance and the process is carried out iteratively until a termination criterion is reached. Of current algorithms, GP is one of the most advanced forms of evolutionary search [9]. In canonical GP each solution is represented using a tree structure, which can express a simple computer program, function, or operator. Individual trees are constructed using elements from two finite sets, internal nodes contain simple functions from a *Function* set F, and leaves contain the input variables from the *Terminal* set T. These sets define the search space for a GP algorithm, and when a depth or size limit is enforced, the space is normally very large but finite.

3.3 Proposed algorithm

The proposal of this work is to use standard Koza style GP to solve the optimization problem given in Eq. 5. In what follows, we define the fitness function and the search space for the GP algorithm.

3.3.1 Fitness evaluation

Fitness is defined for a maximization problem, where the fitness of an operator K is given by

$$f(K) = \frac{1}{\frac{1}{M}\sum_{j=1}^{M} Err[\widehat{K}(I_j), \widehat{H}_{I_j}] + \epsilon} \ , \tag{7}$$

where I_j is the jth image in the training set of M images, $\epsilon = 0.01$ avoids divisions by zero, and $\widehat{K}(I)$ and $\widehat{H_I}$ are normalized versions of $K(I)$ and H_I using the L2-norm. Here, fitness is assigned based on the mean RMSE computed for a set of training images. However, one constraint is added, when the standard deviation of the RMSEs is zero then the fitness is also set to zero. This is done because it would be naive to expect the same estimation for every image. The constraint removes the possibility of an operator estimating the same exponent for every pixel. For instance, if all exponents are set to zero the estimation is meaningless, but it might produce a better fitness than a random operator and could then take over the population in earlier generations. This scenario occurred often in preliminary runs of the GP-search that did not include this constraint.

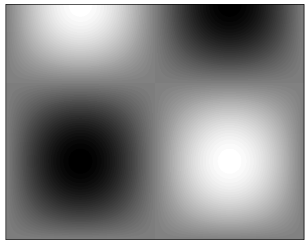
3.3.2 Search space

The search space for a GP is established by the sets of Terminals and Functions, in our work these are

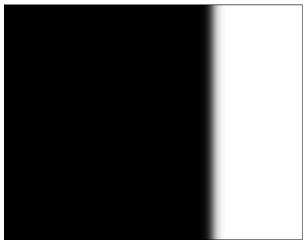
$$\begin{gathered} F = \{+, |+|, -, |-|, |I_o|, *, \div, I_o^2, \sqrt{I_o}, log_2(I_o), k \cdot I_o\} \\ \bigcup \{BF, G_\sigma, Avg_m, Med_m, Max_m, Min_m\} \ , \\ T = \{I\} \ , \end{gathered} \tag{8}$$

where I is the input image; I_o represents I or the output from any function in F; G_σ are Gaussian smoothing filters with $\sigma \in \{1, 3, 5\}$; $Avg_m, Med_m, Max_m, Min_m$ are average, median, max and min filters with a mask size of $m \times m$ and

(a) Polynomial p_1



(b) Sine p_2



(c) Exponential p_3

Figure 3: Prescribed regularity of our experimental data.

$m \in \{3, 5, 7\}$; BF is a bilateral filter [20]; and a scale factor $k = 0.05$. Set F contains functions that operate in a point by point manner, such as the addition of two matrix, and image filters that operate within a local neighborhood. The latter group allows the evolved estimators to exploit variations within a local image region, information which is essential for the oscillations method. Regarding the point to point arithmetic operations, we use the following conventions in order to avoid undefined operations: (1) we assume that $I \in \Re^+$; (2) the logarithm is protected by using $\log(0) = 0$, and $\log(-a) = \log(a)$; (3) division is protected by $\frac{a}{b} = a$ if $b = 0$; and (4) we define $\sqrt{-a} = \sqrt{a}$.

4. EXPERIMENTS AND RESULTS

This Section describes our GP algorithm, presents the experimental results, and provides comparisons with the oscillations method.

4.1 Implementation

4.1.1 Experimental setup

The GP algorithm was set-up with the parameters presented in Table 1, most are based on basic GP literature [9]. The only non-canonical aspect of our GP algorithm is the use of a bloat control method, a dynamic maximum tree depth [18]. The algorithm was programmed using the Matlab toolbox GP-Lab [1], and estimation of the Hölder exponent was carried out using code for the oscillations method that will be provided in the next release of FracLab [14].

4.1.2 Training and test data

In order to compute fitness and to perform further tests, we build a set of images with prescribed regularity using 2D multifractional Brownian motions. This is a generalization of the fractional Brownian motion where the constant exponent H is replaced with a Hölder continuous function [1]. This type of signal is a good model for real world data, and it can be directly generated with a prescribed regularity [3]. In this work, we generate three groups of images with Fraclab, using three different functions that take as input the point coordinates (x, y) of an image and provide as output the desired regularity; these functions are: (a) a *Polynomial* $p_1(x, y) = 0.1 + 0.8xy$; (b) a *Sine* $p_2(x, y) = 0.5 + 0.2(sin(2\pi x))(cos(\frac{3}{2}\pi y))$; and (c) an *Exponential* $p_3(x, y) = 0.3 + \frac{0.3}{1+e^{-100(x-0.7)}}$. These functions provide the prescribed regularity needed to build the synthetic images used for training and testing of our evolved operators, see Figure 3.

[1]GP-Lab by Sara Silva (gplab.sourceforge.net/index.html).

Table 1: GP parameters used in our experiments.

Parameter	Description
Population size	200 individuals
Iterations	100 generations
Initialization	*Ramped Half-and-Half*
Crossover probability	$p_c = 0.85$
Mutation probability	$p_\mu = 0.15$
Bloat control	Dynamic maximum tree depth
Initial max. depth	Six levels
Initial dynamic max. depth	Eleven levels
Max. tree depth	16 levels
Selection	Stochastic universal sampling
Survival	Elitism
Runs	Eleven

Here, we generate twelve images for each of the prescribed regularity functions p_1, p_2 and p_3; all images have the same size of 512×512 pixels. Figure 4 shows three examples from each of these groups. What is important to see is that images with the same prescribed regularity are nevertheless quite different, and in fact an infinite number of images exist that share the same regularity! Therein lies the intrinsic difficulty of obtaining an optimal estimator for the Hölder exponent. The images we generate are divided into two sets, one for training and one for testing. The training set contains six images with polynomial regularity, and all the rest, thirty in total, are used for testing.

4.2 Results and comparisons

This section describes the relevant experimental results of our work. First, Figure 5 shows the convergence plots for each of the eleven runs of our GP. Figure 5a plots the best individual fitness at each iteration, and Figure 5b shows the average fitness of the population. Each plot is tagged using the acronym HGP (*Hölderian regularity with GP*), and the corresponding run number. These plots show similar convergence patterns in all of the runs, with HGP-7 being the best in both cases, and HGP-5 and HGP-6 not far behind based on the fitness computed with the training set. The first observation is that the evolutionary search shows a steady convergence over the specified number of generations, from which we can say that the proposed algorithm appears to be well suited for the stated problem. On the other hand, it could be said that the small number of runs can only provide a weak statistical inference, and that the termination criteria does not eliminate the possibility that further improvements could have been achieved. However, there is a practical restraint that accounts for these shortcomings, namely that the computation time for a single run is very long. In some experiments it required several days. Therefore, we believe that in such applications the somewhat small number of runs is justified.

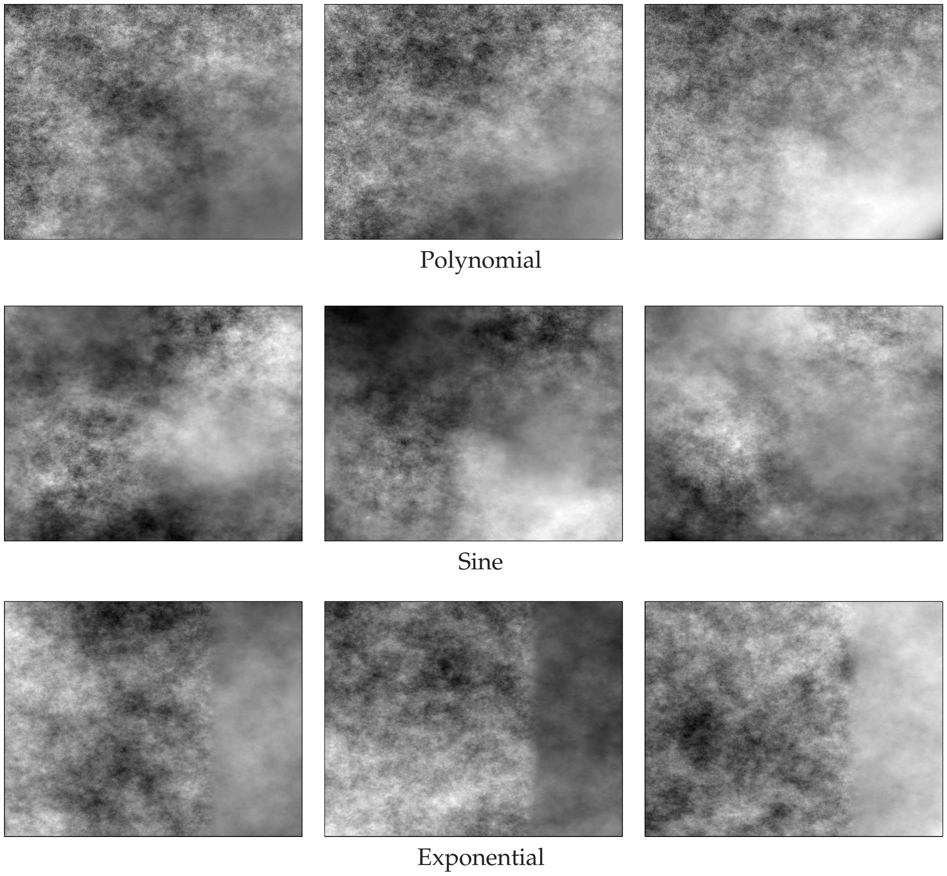


Figure 4: These images have a prescribed regularity given by functions p_1 (Poly.), p_2 (Sine) and p_3 (Exp).

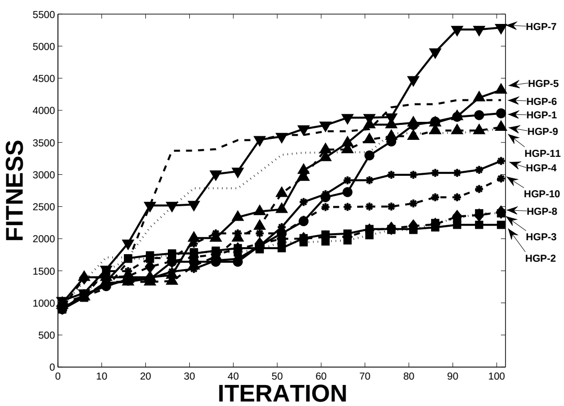
Moving on, in Table 2 we show the result of testing our evolved operators on the additional test images, thirty in total. Here, we show the average RMSE computed for each of the three groups (polynomial, sine and exponential prescribed regularity functions), the standard deviation, and the corresponding fitness value. In these tests we use the best individual found at the end of each run. For comparisons we show the same statistics computed for the estimation obtained with the oscillations method. Among the evolved operators, HGP-7 also achieves the highest marks on the test images with polynomial regularity. It is also the second best when tested on the images with sine function regularity, while HGP-11 is the best in this case (the RMSE of both is of the same order). However, for the exponential function, HGP-7 is only average amongst this group, but still better that the oscillations method, the best operator for these images is HGP-1. A few remarks are relevant for these results. First, we can affirm that the evolved operators are indeed superior to oscillation-based estimation, in some cases the difference is one order of magnitude. Second, using the proposed training set we obtain an operator that achieves high fitness on the test images with polynomial and sine regularity. The former result was expected given the training set, and the latter is a good indication that GP does not over-fit the solutions that it generates. In the case of the exponential function, HGP-7 was not among the best but still reaches a better performance than the oscillation-based method. Third, we can see that some of the operators that achieve only average performance on the training set are able to achieve much better scores on other types of regularity functions. This further suggests that our algorithm does not over-fit the solutions to the specific problem given by the training set, it is a well-posed search for regularity estimator in a more general sense. Finally, in order to give a sense of what GP generates in work, the program tree of the best individual found is shown in Figure 6.

A qualitative comparison between our evolved operators and the oscillations method is shown in Figure 7, where the Hölderian regularity is estimated for three of the test images, one for each type of prescribed regularity. The first column of Figure 7 is the estimation for a test image with a prescribed regularity given by the polynomial function and the next column shows the difference between the estimated regularity and the prescribed regularity of Figure 3. The following two pairs of columns are similar for the sine and exponential test cases. Note that four operators are not included (HGP-2,3,8 & 10) to reduce the length of the paper. Moreover, the excluded ones achieve the worst performance among the HGP estimators but are still better than the oscillations method. This comparison graphically confirms the numerical results of Table 2, from which we can reaffirm the conclusion that the HGP operators are superior estimators of the pointwise Hölder exponent.

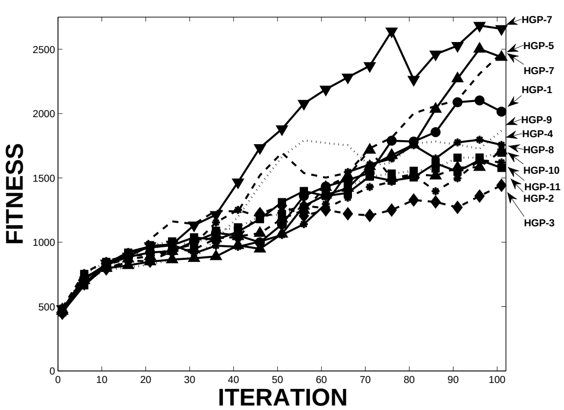
Finally, Table 3 provides a informal comparison of computation time between the HGP estimators, and the estimation obtained using Fraclab on a single test image, showing the average over five executions. These results were obtained

Table 2: Comparison of the best solution from each run with the oscillations method using the thirty test images from the three different functions for the prescribed regularity. The mean and std are scaled to 10^{-3}, and bold marks the best results.

	HGP-1	HGP-2	HGP-3	HGP-4	HGP-5	HGP-6	HGP-7	HGP-8	HGP-9	HGP-10	HGP-11	Osc.
Polynomial (six images)												
Mean	0.149	0.638	0.317	0.207	0.127	0.125	**0.087**	0.328	0.154	0.233	0.159	0.473
Std.	0.003	0.048	0.044	0.009	0.003	0.005	0.004	0.037	**0.002**	0.008	0.004	0.046
Fit.	4015	1355	2396	3254	4414	4452	**5343**	2337	3932	3001	3861	1744
Sine (twelve images)												
Mean	0.047	0.408	0.194	0.082	0.064	0.064	0.051	0.243	0.072	0.115	**0.028**	0.323
Std.	0.004	0.147	0.043	0.02	0.002	0.004	0.002	0.025	0.002	0.008	**0.003**	0.046
Fit.	6822	1969	3396	5484	6096	6105	6617	2915	5809	4645	**7799**	2364
Exponential (twelve images)												
Mean	**0.055**	0.395	0.165	0.102	0.103	0.103	0.166	0.282	0.078	0.141	0.06	0.329
Std.	0.004	0.184	0.013	0.03	**0.001**	0.003	0.001	0.049	0.003	0.007	0.002	0.029
Fit.	**6438**	2022	3770	4939	4923	4923	3766	2617	5615	4148	6256	2331



(a) Best fitness



(b) Population average

Figure 5: Convergence plots for each of the eleven runs.

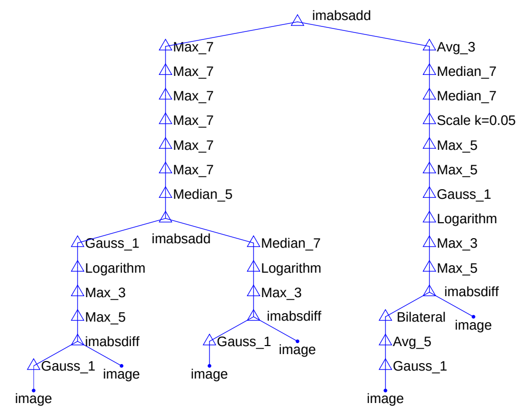


Figure 6: The program tree for the HGP-7 estimator.

using a standard Laptop-PC with Intel Dual Core 64-bit 2.39 GHz processor, 4 GM of shared RAM, and running 32-bit Win-XP SP3 and Matlab 2009a. Before evaluating these results, several observations are relevant. First, our estimators were not simplified, the complete GP trees are used in all tests. Second, time of computation was not included within our fitness criteria. And finally, neither our code nor the Fraclab functions for the oscillations method was optimized in any way. Nevertheless, the results presented in Table 3 do provide a rough estimate of which estimation method is more efficient, and under this comparison we can again conclude that GP produces better methods for estimation.

5. CONCLUDING REMARKS

This paper presents an approach that automatically synthesizes image operators that estimate the pointwise Hölder exponent for 2D signals. It employs a GP to search for operators that minimize the estimation error given a prescribed Hölderian regularity. Training is carried out using a small set of images, all of which have the same regularity given by a polynomial function. Then, the evolved estimators are tested on a new set of images, most of which have a different Hölderian regularity given by a sine and an exponential function. The experimental results show that the evolved HGP estimators produce a good estimation of the pointwise Hölder exponent, from both a quantitative and qualitative perspective. In fact, the HGP estimators consistently and significantly outperform the oscillations method by as much as one order of magnitude. Moreover, the GP algorithm is able to produce estimators that generalize quite well given the limited set of training examples, and the evolutionary process shows a steady and progressively improving convergence. These results suggest the following main conclusions. First,

Table 3: Run-time comparisons of the HGP estimators and the oscillations method from Fraclab; all values are provided in seconds and represent the average over five executions.

	HGP-1	HGP-2	HGP-3	HGP-4	HGP-5	HGP-6	HGP-7	HGP-8	HGP-9	HGP-10	HGP-11	Osc.
Time	1.20	9.43	10.32	6.26	1.40	4.79	9.34	5.08	**0.77**	12.61	5.81	365.2

new estimators for the pointwise Hölder exponent can be developed using a GP-based search and optimization process. This gives further empirical evidence that confirms the applicability of GP to difficult mathematical problems in applied domains. Second, the GP-based approach could synthesize estimators that simplify the practical use of regularity-based analysis for a wider variety of application domains. Finally, we speculate that the estimators produced by GP might lead us towards new analytical approaches for regularity analysis, although this will be left as a topic for future research.

Acknowledgments

We sincerely thank Dr. Patricia Melin from the Instituto Tecnológico de Tijuana for providing the necessary infrastructure to carry out this research, Dr. Olivier Barriere for providing invaluable code used during the development of our work and Dr. Gustavo Olague who offered very useful comments and observations.

References

[1] A. Ayache and J. Lévy-Véhel. The generalized multifractional brownian motion. *Statistical Inference for Stochastic Processes*, 3:7–8, 2000.

[2] P. Balasubramaniam and A. Vincent Antony Kumar. Solution of matrix riccati differential equation for nonlinear singular system using genetic programming. *Genetic Programming and Evolvable Machines*, 10(1):71–89, 2009.

[3] O. Barriere. *Synthése et estimation de mouvements browniens multifractionnaires et autres processus á régularité prescrite. Définition du processus auto-régulé multifractionnaire et applications*. PhD thesis, Ecole centrale de Nantes et Universite de Nantes, France, 2007.

[4] K. De Jong. *Evolutionary Computation: A Unified Approach*. The MIT Press, 2001.

[5] S. Gustafson, E. K. Burke, and N. Krasnogor. On improving genetic programming for symbolic regression. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, 2-4 September 2005, Edinburgh, UK*, pages 912–919. IEEE, 2005.

[6] S. Jaffard. Wavelet techniques in multifractal analysis. In *Fractal Geometry and Applications: A Jubilee of Benoit Mandelbrot, Proceedings of Symposia in Pure Mathematics*, volume 72, pages 91–151, 2004.

[7] M. Keijzer. Scaled symbolic regression. *Genetic Programming and Evolvable Machines*, 5(3):259–269, 2004.

[8] M. Keijzer and V. Babovic. Declarative and preferential bias in gp-based scientific discovery. *Genetic Programming and Evolvable Machines*, 3(1):41–79, 2002.

[9] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[10] P. Legrand. *Debruitage et interpolation par analyse de la regularite Hölderienne. Application a la modelisation du frottement pneumatique-chaussee*. PhD thesis, Université de Nantes, France, 2004.

[11] P. Legrand and J. Lévy-Véhel. Local regularity-based interpolation. In *WAVELET X, Part of SPIE's Symposium on Optical Science and Technology*, volume 5207, 2003.

[12] P. Legrand and J. L. Vehel. Local regularity - based image denoising. *ICIP03, Spain, IEEE International Conference on Image Processing*, pages 377–380, September 14-17, 2003.

[13] J. Lévy-Véhel. *Fractal Image Encoding and Analysis*, chapter Introduction to the Multifractal Analysis of Images, pages 299–341. 1998.

[14] J. Lévy-Véhel and P. Legrand. *Thinking in Patterns*, chapter Signal and Image Processing with FRACLAB, pages 321–322. 2004. http://fraclab.saclay.inria.fr/homepage.html.

[15] S. Mallat. *A wavelet tour of signal processing*. Elsevier, San Diego, CA, 2nd edition, 1999.

[16] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[17] C. B. Perez and G. Olague. Evolutionary learning of local descriptor operators for object recognition. In *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 1051–1058, New York, NY, USA, 2009. ACM.

[18] S. Silva and J. Almeida. Dynamic maximum tree depth. In E. C.-P. et al., editor, *Proceedings of the Genetic and Evolutionary Computation - GECCO 2003, Genetic and Evolutionary Computation Conference, Chicago, IL, USA, July 12-16, 2003, Part II*, Lecture Notes in Computer Science, pages 1776–1787. Springer-Verlag, 2003.

[19] L. Spector, D. M. Clark, I. Lindsay, B. Barr, and J. Klein. Genetic programming for finite algebras. In *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 1291–1298, New York, NY, USA, 2008. ACM.

[20] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, page 839, Washington, DC, USA, 1998. IEEE Computer Society.

[21] C. Tricot. *Curves and Fractal Dimension*. Springer-Verlag, 1995.

[22] L. Trujillo and G. Olague. Synthesis of interest point detectors through genetic programming. In M. Cattolico, editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Seattle, Washington, July 8-12*, volume 1, pages 887–894. ACM, 2006.

[23] L. Trujillo and G. Olague. Automated design of image operators that detect interest points. *Evolutionary Computation*, 16(4):483–507, 2008.

[24] L. Trujillo, G. Olague, P. Legrand, and E. Lutton. Regularity based descriptor computed from local image oscillations. *Optics Express*, 15:6140–6145, 2007.

[25] L. Trujillo, G. Olague, E. Lutton, and F. Fernández de Vega. Multiobjective design of operators that detect points of interest in images. In M. Cattolico, editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Atlanta, GA, July 12-16*, pages 1299–1306, New York, NY, USA, 2008. ACM.

[26] Y. Zhang and P. I. Rockett. Evolving optimal feature extraction using multi-objective genetic programming: a methodology and preliminary study on edge detection. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 795–802, New York, NY, USA, 2005. ACM.

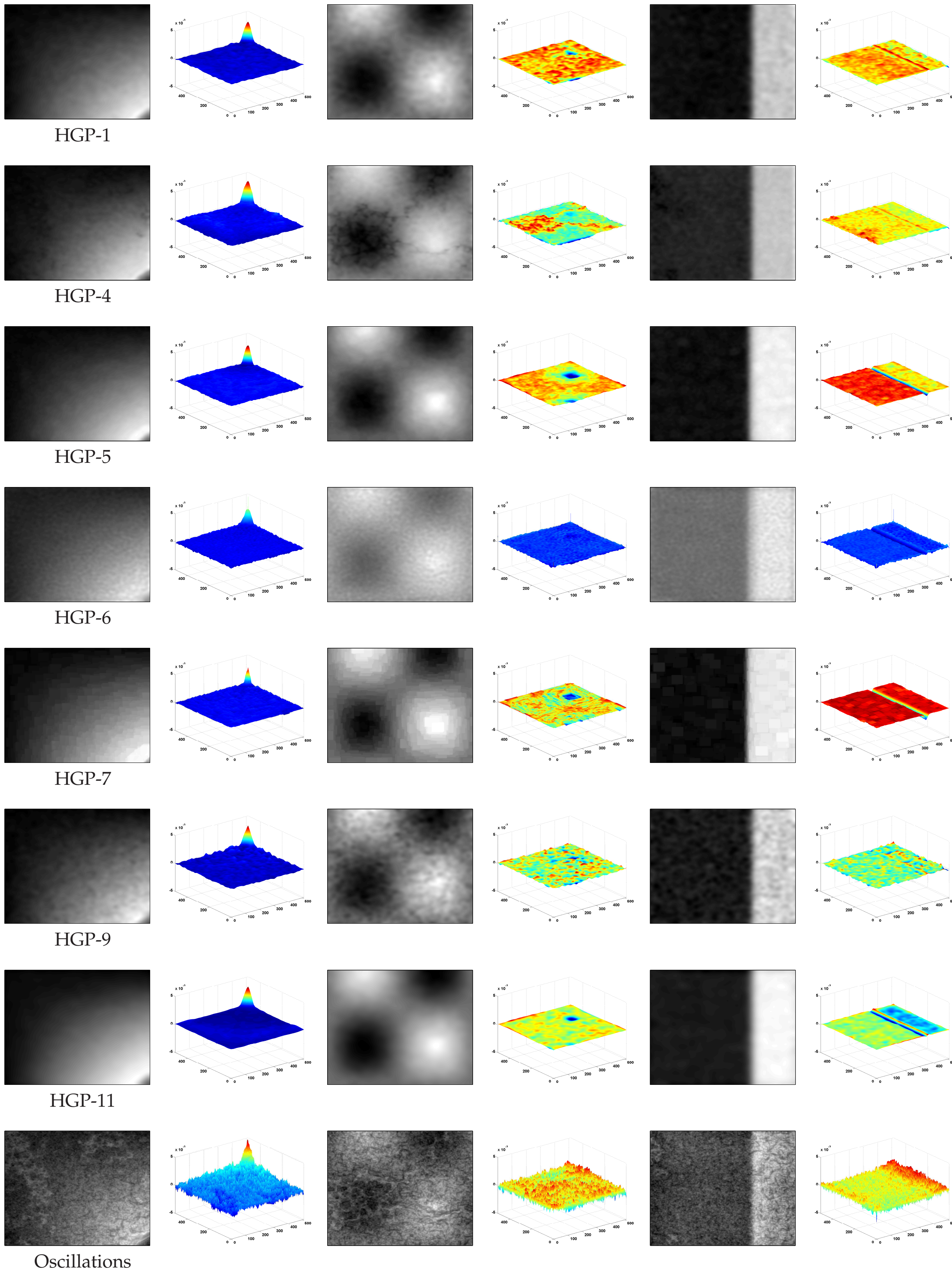


Figure 7: Qualitative comparison between the evolved HGP estimators and the oscillations method using one test case from each prescribed regularity function. The first column shows the estimated regularity for a test image with a prescribed regularity given by the polynomial function, and the next column shows the difference between the estimated regularity and the prescribed regularity, see Figure 3. The following two pairs of columns are similar, the second pair is for the sine function, and the final pair for the exponential function. All of the difference plots share the same z-scale, $[-0.005, 0.005]$.