Evolving estimators of the pointwise Hölder exponent with Genetic Programming

Leonardo Trujillo [a,*], Pierrick Legrand [b,c], Gustavo Olague [d], Jacques Lévy-Véhel [e]

[a] Doctorado en Ciencias de la Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica, Instituto Tecnológico de Tijuana, Blvd. Industrial y Av. ITR Tijuana S/N, Mesa Otay C.P. 22500, Tijuana BC, México
[b] Université Victor Segalen Bordeaux 2 and the IMB, Institut de Mathématiques de Bordeaux, UMR CNRS 5251, France
[c] ALEA Team, INRIA Bordeaux Sud-Ouest, France
[d] EvoVision Project, Computer Science Department, Centro de Investigación Científica y de Educación Superior de Ensenada, Km. 107 Carretera Tijuana-Ensenada, 22860 Ensenada, BC, México
[e] REGULARITY Team, INRIA Saclay, Ile de France, France

ARTICLE INFO

ABSTRACT

The regularity of a signal can be numerically expressed using Hölder exponents, which characterize the singular structures a signal contains. In particular, within the domains of image processing and image understanding, regularity-based analysis can be used to describe local image shape and appearance. However, estimating the Hölder exponent is not a trivial task, and current methods tend to be computationally slow and complex. This work presents an approach to automatically synthesize estimators of the pointwise Hölder exponent for digital images. This task is formulated as an optimization problem and Genetic Programming (GP) is used to search for operators that can approximate a traditional estimator, the oscillations method. Experimental results show that GP can generate estimators that achieve a low error and a high correlation with the ground truth estimation. Furthermore, most of the GP estimators are faster than traditional approaches, in some cases their runtime is orders of magnitude smaller. This result allowed us to implement a real-time estimation of the Hölder exponent on a live video signal, the first such implementation in current literature. Moreover, the evolved estimators are used to generate local descriptors of salient image regions, a task for which a stable and robust matching is achieved, comparable with state-of-the-art methods. In conclusion, the evolved estimators produced by GP could help expand the application domain of Hölder regularity within the fields of image analysis and signal processing.

1. Introduction

Image analysis and understanding entails the detection and extraction of meaningful descriptive features to carry out higher level tasks such as object recognition [47] or image indexing [61], two common examples. For most application, the most prominent and informative parts of an image correspond with regions that exhibit an irregular structure with a high amount of local variation. Therefore, many works have addressed the problem of detecting and describing salient image regions [38,60,57,58]. One approach towards describing local shape and appearance within an image is through the concept of signal regularity, an approach that can characterize the singularities contained within non-differentiable signals [11,35]. Regularity-based analysis has been used to describe local image patches [29,58] and to detect salient regions [31].

---

* Corresponding author.
  E-mail addresses: leonardo.trujillo.ttl@gmail.com (L. Trujillo), pierrick.legrand@u-bordeaux2.fr (P. Legrand), olague@cicese.mx (G. Olague), Jacques.levy-vehel@inria.fr (J. Lévy-Véhel).

In this paper, we focus on Hölder regularity, quantifies the amount of regularity any given point on a signal, using what is known as the pointwise Hölder exponent [11,35] (see Section 3.1 for a formal definition). The Hölder exponent has proven to be a useful tool for image analysis [29,58,31], but computing the exponent is not a trivial task. In fact, closed form solutions only exist for a narrow class of functions, while for real-world signals the exponent must be estimated. Therefore, several estimation methods have been proposed, derived from a formal analysis of the Hölder exponent, using techniques from fractal theory and signal processing [53,16,31,2]. However, some of these estimators are based on necessary assumptions regarding the underlying structure of the signal. Moreover, to use and develop practical implementations of current methods a system designer must make several parametric choices and ad hoc decisions. Furthermore, these estimators usually require a large run-time, which limits their use for systems that operate in real-time. Therefore, we pose the following research question: *Can the pointwise Hölder exponent be estimated using an image operator that achieves an accurate estimation using a simple and fast algorithm?* We believe that if such an operator exists, it can open new application domains for regularity-based image analysis.

Therefore, in this work the goal is to find operators that can provide a positive answer to the above question. To achieve this, we pose a search/optimization problem and solve it using Genetic Programming (GP). Over the past two decades, GP has proven to be a powerful paradigm for the development of computer algorithms that automatically synthesize solutions for complex tasks. Unlike black-box methods, when a search is properly framed, GP can produce solutions that are amenable to further analysis [57,41,18]. Moreover, GP is also quite flexible, with successful applications in various domains [21,22,19], including image analysis [13,57,39,44], computer vision [25,7,40,34] and applied mathematics [5,15,51]. The power and flexibility of GP comes from the fact that it solves two tasks simultaneously: (1) it searches for the desired functionality and (2) determines the structure of the solution [20,26].

For these reasons, we use GP to search for specialized image operators that estimate the pointwise Hölder exponent for digital images. Experimental results show that GP is capable of evolving highly competitive estimators, that approximate the estimation produced by a traditional approach, the oscillations method, with a small error and a high correlation. Moreover, the GP estimators are efficient, in terms of computation time they achieve a 50% improvement with respect to one estimation method, and several orders of magnitude with respect to other approaches. In fact, a noteworthy result is that GP found a novel computational operator that extracts a measure of image regularity that can be implemented in real-time. The quality of the evolved estimators is verified using a common problem of modern computer vision, the description and matching of local image features [38]. In this task, the evolved estimators can be used to construct meaningful and discriminative local descriptors that compare favorably with the original Hölder-based descriptor [58,55].

The remainder of this paper proceeds as follows. Section 2 contains a brief overview of related works and applications of GP. Then, Section 3 gives a brief introduction to Hölder regularity and GP. Section 4 poses the task of estimating the Hölder regularity of an image as an optimization problem, and presents a GP approach to solve it. Experimental results are detailed in Section 5, and qualitative and quantitative comparisons are made between the evolved estimators and traditional methods. Moreover, in Section 6 the evolved estimators are used to build local image descriptors and are compared with the local Hölder descriptor of [58]. Finally, a summary and concluding remarks are outlined in Section 7.

2. Related work

Image analysis encompasses a diverse group of complex problems, where the relationships between the input signal and desired output are poorly understood, closed-form analytical solutions normally do not exist, and the structure of the desired solution cannot de defined a priori. Conversely, a large amount of experimental data is usually available or easy to obtain. These characteristics makes problems in this field appropriate candidates for GP-based solutions [25,7]. For instance, GP has been used for image classification [23,52], object recognition [14,10,13], feature synthesis [24,44], image segmentation [42,50], depth estimation [34] and local image description [40,41]. In particular, the proposal made in this paper is related to other works that extract a descriptive value for each image pixel. For example, the problem of interest point detection has been posed as a single [56,57] and a multi-objective problem [59,39] and solved with GP. Another example is [62], where GP is used to detect edge points, a fundamental problem in computer vision. In those works, the operators produced by GP compute a measure of *edgeness* or *saliency*, which is a similar proposition to the goal we pursue here. On the other hand, our proposal is based on the formal mathematical concept of image regularity while the above cited works focus on ambiguous semantic concepts that could be interpreted in a variety of ways. In a previous paper [54], we focused on the same high-level goal of evolving estimators of Hölder regularity. However, in [54] the ground truth for the learning process utilized images of multifractal Brownian motion. Conversely, the work presented here uses images of real-world scenes. This difference in the training set allowed us to produce operators that achieve better estimation on real images, comparable with other state-of-the-art methods (more on this point in Section 4). Moreover, some of the evolved estimators operate in real-time and can be used for local image description and matching [58,55].

Turning to mathematics, GP has mostly been used to automatically derive models, functions or operators that can characterize a set of sample data with a minimum amount of error, a particular type of regression problem called *symbolic regression* [20]. In classical regression an optimization algorithm must find the optimal values for a set of coefficients for a function that was chosen by a human expert. On the other hand, in symbolic regression the problem consists on finding the mathematical expression of the function that best fits the training data, a harder task for which GP is particularly well-suited.

Symbolic regression represents one of the earliest successful applications of GP, and significant advances have continuously been developed [20,17,12,3]. Indeed, the success of GP in symbolic regression has prompted some researchers to characterize a correctly framed GP search as a tool for automatic scientific discovery [18,46]. Recently, other works have shown that GP can be used in other domains of mathematics. For instance, in [4] GP constructs approximate solutions for complex differential equations. In [51], GP is used to study special elements of finite algebras, and in [5] to design quantum circuits. Finally, [15] derives differential equation models using GP.

3. Theoretical background

The aim of this section is to present a concise introduction to Hölder regularity and the GP paradigm, the research areas that intersect in this paper. However, some details are omitted for brevity, but the interested reader should refer to [53,43] for a detailed introduction to Hölder regularity and GP respectively.

3.1. Hölder regularity

The notion of Hölder regularity allows one to characterize the singular structures contained within a signal [30]. A quantitative understanding of the regularity of a signal can be obtained from measuring Hölder exponents, either within a local region or at each individual point. In this paper, we focus on signals in $\mathbb{R}^2$ since the goal is to develop regularity-based techniques for image analysis. However, the following definition and discussion regarding Hölder exponents are applicable to the general case of $\mathbb{R}^n$ signals.

3.1.1. The pointwise Hölder exponent

Definition 1. Let $f : \mathbb{R}^2 \to \mathbb{R}$ and $x_0 \in \mathbb{R}^2$ such that f is not differentiable at x_0. The Hölder exponent of f at x_0 is the real number:

$$\alpha_p(x_0) = \liminf_{\|h\| \to 0} \frac{\log |f(x_0 + h) - f(x_0)|}{\log \|h\|}, \tag{1}$$

where $\|.\|$ denotes the Euclidean norm in $\mathbb{R}^2$. This definition is illustrated on Fig. 1: it states than, for any $\varepsilon > 0$, the graph of f in a sufficiently small neighbourhood of x_0 is included in all envelopes with equation $\pm|x - x_0|^{\alpha_p(x_0)-\varepsilon}$ but in no envelope with equation $\pm|x - x_0|^{\alpha_p(x_0)+\varepsilon}$.

Eq. (1) gives a bound on the amount by which a signal varies, or oscillates, around point x_0 within a local neighborhood $B(x_0, \eta)$. From the definition and Fig. 1, one sees that a small α (i.e. close to 0) translates into a wildly varying signal, while an α close to one corresponds to a smoother signal.

The reason for the restriction, f not differentiable at x_0, is the following. Suppose that $f(x) = \|x\| + \|x\|^{\frac{3}{2}}$. Then, the definition above would yield an exponent equal to 1 at $x_0 = 0$. However, we are interested in the *irregular* part of f, that is, the term $\|x\|^{\frac{3}{2}}$. In order to remove lower order smooth terms as the $\|x\|$ in our example, we need in the general case to subtract a polynomial from f rather than simply $f(x_0)$.

We will not consider the extra polynomial term in this work. This amounts to *clipping* all exponents to one. The main reason for this choice is that, for applications in image processing, only irregular parts in the image (that is, ones with exponents smaller than one) are of interest. In addition, the use of Eq. (1) simplifies the algorithmic implementation.

3.1.2. Estimation of the pointwise Hölder exponent

In order to use Hölder exponent in applications, one needs to have robust estimations methods from sampled data. This is a difficult task, as α is defined by a limit which would require an infinite resolution in order to be perfectly determined. Two main classes of estimators have been developed. The first one is *parametric*, that is, one makes assumptions about the nature
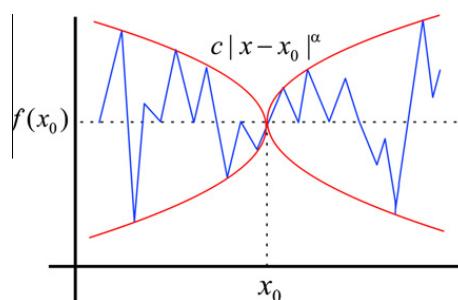

Fig. 1. Hölder envelope of a non-differentiable signal f at point x_0. This representation is for a 1D signal or a cross-section of a signal in 2D.

of the signal. Probably the most popular model in this frame is that of multifractional Brownian motion and its extensions, because it seems to adequately describe various phenomena [2]. In this class, rigorous convergence results may be obtained, along with speeds of convergence.

The second class are non-parametric estimators, which is the frame we place ourselves in. Our main motivation is that there is no accepted universal model for natural images that could be used as a basis for developing a parametric approach. Among the most well-known non-parametric estimator of Hölder exponents are wavelet-based methods, such as wavelet coefficients regression and wavelet leaders regression. These estimators usually produce reasonable results when the wavelets satisfy some specific regularity properties [16], and when the signal is not oscillating [29].

A more direct method for estimating α would be to use increments and directly implement Eq. (1) [31]. However, this is rather crude, as it does not make use of the whole information that is available. One obtains a better estimator by re-writing Definition 1 using the notion of oscillations. To explain this method, let us first recall the definition of oscillation.

Definition 2. The oscillation of f in the ball centered at t is:

$$osc_\tau(t) = \sup_{\|t-t'\|\leqslant\tau} f(t') - \inf_{\|t-t'\|\leqslant\tau} f(t').$$

One can characterize the Hölder exponent in terms of oscillations (when it is smaller than one). Indeed, for c a positive constant, $osc_\tau(t) \leqslant c\tau^\alpha$ implies directly that $|f(t) - f(t')| \leqslant c\|t - t'\|^\alpha$ for t' close to t. Conversely, assume that $|f(t) - f(t')| \leqslant c\|t - t'\|^\alpha$ for all t' close enough to t. Choose t_1 such that $\|t - t_1\| \leqslant \tau$ and $f(t_1) = \sup_{\|t-t'\|\leqslant\tau} f(t')$. Likewise, choose t_2 such that $\|t - t_2\| \leqslant \tau$ and $f(t_2) = \inf_{\|t-t'\|\leqslant\tau} f(t')$. Then,

$$osc_\tau(t) = f(t_1) - f(t_2) = f(t_1) - f(t) + f(t) - f(t_2) \leqslant 2c\tau^\alpha,$$

which is what we needed.

As a consequence, an alternative definition of the Hölder exponent is

$$\alpha_f(t) = \liminf_{\tau\to 0} \frac{\log osc_\tau(t)}{\log \tau}.$$

To estimate in practice $\alpha(t)$ with this formula, one assumes that the liminf above is in fact a limit. This suggests that, for small enough τ, the following relation should approximately hold

$$\log osc_\tau(t) = \alpha_f(t) \log \tau + k,$$

where k is a constant. One then estimates $\alpha_f(t)$ as the slope of the least-square regression of the logarithm of the oscillation as a function of the logarithm of the size τ.

The parameters in this method are the minimum and maximum sizes considered for the neighbourhood τ, and the *base* for incrementing τ between this extreme values; indeed, one verifies empirically that considering all sizes between the minimum and maximum is not optimal. Note that this algorithm, which is valid only for $\alpha < 1$, may be extended to the general case $\alpha > 0$ by replacing the oscillation by so-called higher-order modulus of continuity. It has been shown to achieve good result in real-world applications [29,28,58] and has proven to be superior, in some tests, to the wavelets-leaders method [27]. Therefore, we have chosen the oscillation method as the baseline reference method in this work. Fig. 2 shows a typical examples of the output that this algorithm produces.

In this paper, we use the parameter values published in [54], since they were set with the expressed goal of achieving the best performance on a set of synthetic images for which the underlying regularity was known a priori.
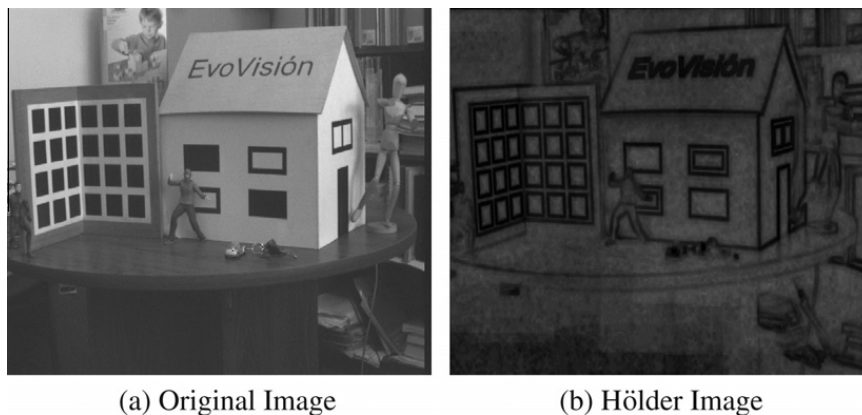

(a) Original Image   (b) Hölder Image

Fig. 2. Hölder exponent estimation using oscillation method.

3.1.3. Shortcomings and open questions

The brief introduction to regularity estimation given above reveals that the problem can be posed and analyzed in different ways. Directly using the definition of the Hölder exponent with either increments or oscillations, with the help of space-frequency decomposition through wavelets, and finally using a parametric method. In all cases, the proposed estimators are derived using formal and rigorous mathematical formulations. However, if we take a pragmatic perspective, we can see that implementing practical algorithms based on these methods presents several noteworthy challenges. Firstly, some of these methods are quite complex, both from a conceptual point of view and from an algorithmic one. Therefore, some implementations of these estimators tend to be relatively slow, which prohibits their use in domains that require real-time processing. Secondly, all of the estimators reviewed in the preceding section depend upon several important parameters that need to be correctly chosen and tuned to achieve a desired performance. However, for most users it will be difficult to establish the best parameters without a significant trial-and-error process. Therefore, an estimation method that limits these shortcomings, but that does not sacrifice performance, would expand the application domains of regularity-based image analysis.

3.2. Genetic Programming

Evolutionary computation encompasses a large group of search and optimization algorithms that base their core functionality on the principles of Neo-Darwinian evolutionary theory [9]. These techniques are population-based meta-heuristics, where candidate solutions are stochastically selected and modified 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. In general, evolutionary algorithms have proven to be a powerful tool in the search for novel solutions to longstanding challenges or provide novel insights to difficult and poorly understood problem domains, achieving excellent results by sometimes generating unorthodox designs [22,19,8]. Of current algorithms, GP is one of the most advanced forms of evolutionary search [20]. 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 of elements, 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, they provide the expressive power that GP can evolve and search for. The search space contains all of the different programs that can be constructed using the basic building blocks in T and F, and when a depth or size limit is enforced on the trees, this space is normally very large but finite. Therefore, when using a canonical GP, there are two main aspects that must be defined: (1) the elements within F and T; and (2) the evaluation function that guides the evolutionary search by providing structure to the fitness landscape.

The evolutionary loop in GP is similar to that of the more widely known genetic algorithm (GA), the main difference is the manner in which individuals are coded [9]. While a GA uses bit strings or parameter vectors with a constant and uniform length, GP uses tree structures that can be of different shapes and sizes. As a result, another important difference is the manner in which new solutions are constructed. Two main operations are normally used. First, crossover effectively swaps two randomly selected subtrees between two individuals that were chosen based on their fitness. Crossover works under the assumption that if two individual trees have a high fitness value, then a combination of both might produce program trees of higher fitness. The second operator is mutation, where a randomly chosen subtree is deleted and substituted by a randomly generated subtree. The main roles of mutation are to introduce diversity into the population and possibly provide a slight and gradual improvements.

4. Hölder exponent estimation as a GP optimization problem

4.1. 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,

$$K^o = \underset{K}{\operatorname{argmin}}\{Err[K(I), H_I]\},\tag{2}$$

where $Err[,]$ represents an error measure. In previous work [54], H_I was set using a prescribed regularity function, and synthetic images of multifractional Brownian motions that share the same underlying regularity were then constructed using the methods described in [1,6]. Using the synthetic images for training, GP evolved several estimators that outperform the baseline oscillations method by as much as one order of magnitude, this is illustrated in Fig. 3. However, the evolved estimators suffer from two significant limitations. First, even though they achieve an accurate estimation on synthetic images, they perform rather poorly when tested on real-world scenes. The problem is depicted in Fig. 4, the estimation obtained by the evolved operators is smoother then in ought to be considering the irregular regions that are contained within the test image. It appears that the evolved estimators are overfitted to the smooth regularity functions used to build the synthetic images. Second, in [54] the evolved estimators substantially outperform the oscillations method based on run-time. However, they cannot be used for real-time applications, they require between 1 and 12 s to process a 512×512 image.
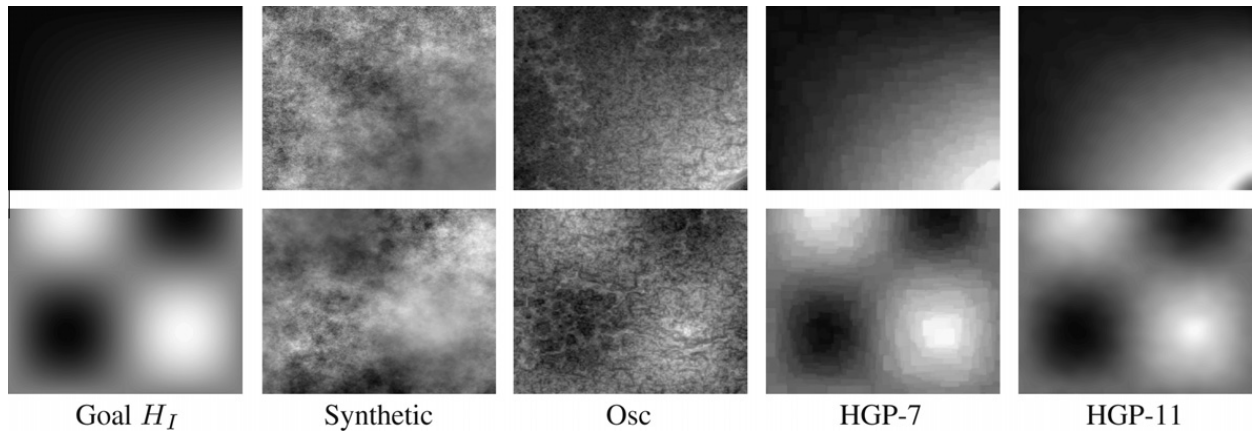
Fig. 3. Performance of evolved estimators compared with the oscillations method on synthetic images of multifractional Brownian motions, from Trujillo et al. [54]. The first column shows two different prescribed regularity functions H_I, a polynomial given by $H_I(x, y) = 0.1 + 0.8xy$ and a sine $H_I(x, y) = 0.5 + 0.2(sin(2\pi x))(cos(\frac{3}{2}\pi y))$. The second column are two synthetic images with the corresponding underlying regularity. The third column shows the estimated regularity obtained with the oscillations method, while the final two columns show the estimation produced by two evolved estimators (HGP-7 and HGP-11). Notice how the evolved estimators approximate quite well the prescribed regularity of the synthetic images, while the oscillations method performs rather poorly by comparison.
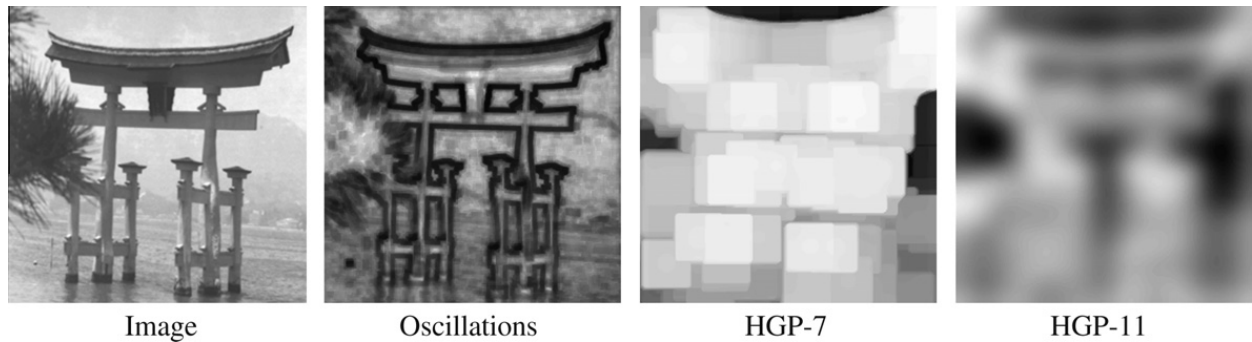

Fig. 4. Performance of evolved estimators compared with the oscillations method on a real image, from Trujillo et al. [54]. Notice how the evolved estimators produce overly smooth estimations, they are overfitted to the type of regularity used during training, see Fig. 3.

Therefore, in this work the ground truth H_I is established by estimating the Hölder exponent on a training set of real images using the oscillations method. Hence, the optimization problem is that of approximating, with the minimum amount of error, the estimation obtained by the oscillations method. The justification for this choice is based on three assumptions. First, since we already know that GP can produce better estimators that the oscillations method [54] (for synthetic Brownian motion images), then we assume that GP will also be able to closely approximate the oscillations method on a set of real images. Second, [54] showed that the evolved estimators are substantially faster than the oscillations method. Therefore, if the evolved estimators that are, in some sense, better then the oscillations method are also computationally faster, then it is reasonable to assume that estimators that approximate the oscillations method might be simpler and computationally faster. Third, since the oscillations method has achieved good results on several real-world tasks [58,55], evolved estimators that approximate the oscillations method should achieve similar performance.

It is important to point out that all assumptions discussed above are experimentally tested and validated in Section 5. Moreover, regarding the use of a finite set of training images, here we use what might be considered as *normal* images of real world scenes. While their choice might seem ad hoc, this will be a common objection for any machine learning approach. Therefore, to validate their usefulness, and show that the learning process was not overfitted to the training examples, we perform a substantial amount of tests on unseen images. On the other hand, we can expect that if the training set is modified then the GP search might converge to different types of solutions, as was the case in [54]. However, the results obtained suggest that the evolved estimators generalize well.

4.2. Genetic programming approach

The optimization problem defined in Eq. (2) can be solved using a GP search. For this task, we use a canonical tree-based GP with Koza style genetic operators [20,26] and a bloat control mechanism to limit the size and complexity of individual trees [49]. A general overview of the approach is illustrated in Fig. 6, where the basic evolutionary loop is depicted with its three main stages: fitness evaluation, fitness-based selection, and the production of new program trees.
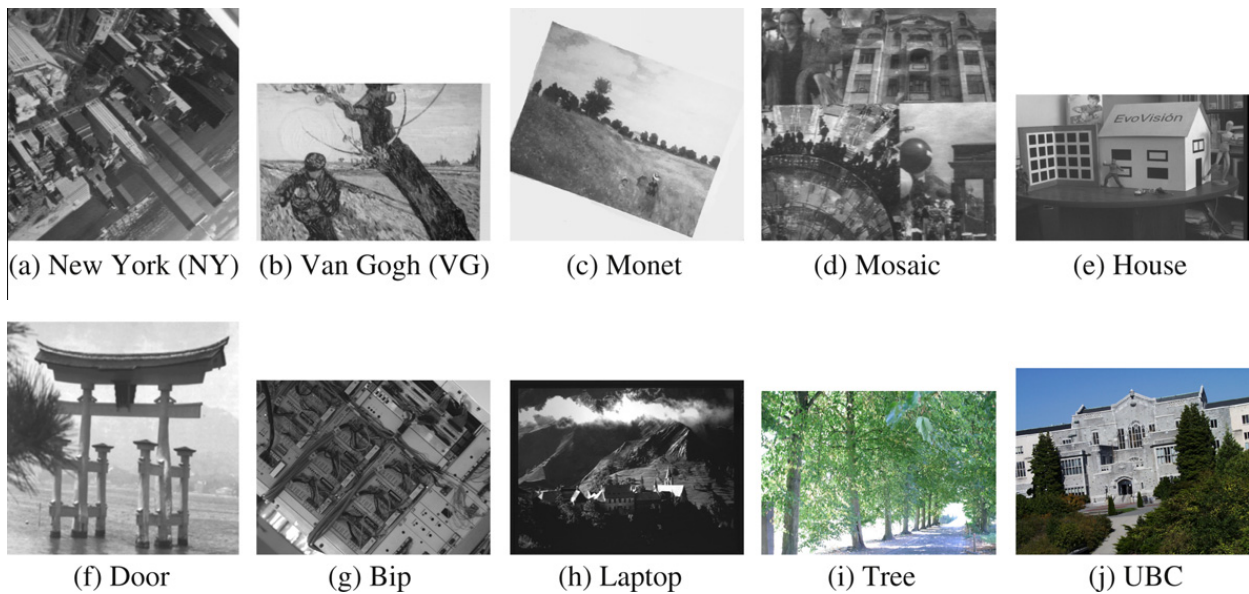
(a) New York (NY) (b) Van Gogh (VG) (c) Monet (d) Mosaic (e) House

(f) Door (g) Bip (h) Laptop (i) Tree (j) UBC

Fig. 5. Some of the images used for training and testing.

4.2.1. Fitness evaluation

The goal of the GP search is to find an operator that minimizes the error between the regularity estimation it provides and a ground estimate established with the oscillations method, see Fig. 6. Here, the error measure from Eq. (2) is defined as the root mean square error (RMSE),

$$Err[K(I), H_I] = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(K(x_i) - H_{x_i} \right)^2}, \tag{3}$$

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

Therefore, if we consider a total of M training images, each with its corresponding Hölder image H_I, Fitness for individual operators K is given by

$$f(K) = \frac{1}{\frac{1}{M} \sum_{j=1}^{M} Err[\widehat{K}(I_j), \widehat{H}_{I_j}] + \epsilon}, \tag{4}$$

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. For example for the ith image pixel in image j, $x_{i,j}$, the normalized value is given by

$$\widehat{K}(x_{i,j}) = S \cdot \frac{K(x_{i,j})}{\sqrt{\sum_{i=1}^{N} K(x_{i,j})^2}}, \tag{5}$$

with $S = 10^3$ used as a scale factor so that fitness is the same order as 1.

The set of M training images contains four images of different scenes ($M = 4$). The name of the images used for training are New York, Van Gogh, Monet and Mosaic. Fig. 5 shows 10 images, the first four are used for training while the others are part of the larger testing set that contains thirty images.[1]

4.2.2. Search space

The search space for GP is established by the sets of Terminals and Functions, given by

$$F_{point} = \{+, |+|, -, |-|, |I_{out}|, *, \div, I_{out}^2, \bigcup \sqrt{I_{out}}, log_2(I_{out}), k \cdot I_{out}\},$$
$$F_{neighborhood} = \{G_1, G_2\}, \tag{6}$$
$$F = F_{point} \bigcup F_{neighborhood}, T = \{I\},$$

---

[1] Some of the images were obtained from the Lear team at INRIA Rhone-Alpes; see K. Mikolajczyk home page: http://lear.inrialpes.fr/people/mikolajczyk/.
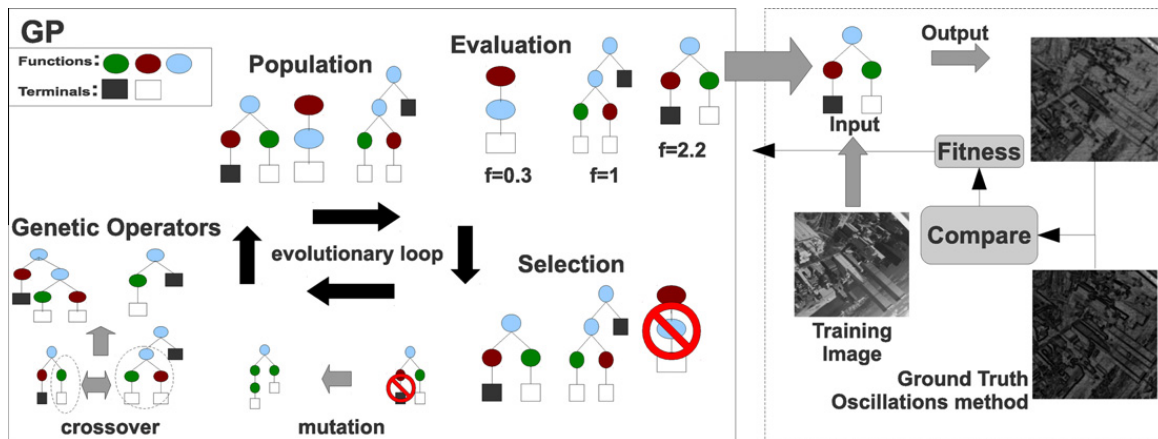
Fig. 6. The GP algorithm used to evolve estimators for the pointwise Hölder exponent.

Table 1
GP parameters used in all runs of the algorithm.

Parameter	Description
Population size	200 Individuals
Generations	200 Generations
Initialization	*Ramped Half-and-Half*, with six levels of maximum depth
Operator probabilities	Crossover $p_c = 0.85$, mutation $p_\mu = 0.15$
Bloat control	Dynamic depth
Maximum dynamic depth	11 Levels
Hard maximum depth	16 Levels
Selection	Stochastic universal sampling
Survival	Elitism

where I is the input image; I_{out} is either the input image I or the output from any function in F; G_σ are Gaussian smoothing filters. Finally, the constant scale factor $k = 0.05$ is included to allow the GP to combine a small fraction of some value, or term, with another. This function was useful in the evolutionary search for interest point detectors [56,57,59].

Set F is conceptually divided into two subsets, one contains point functions F_{point} and the other are functions that operate within a local neighborhood of each pixel $F_{neighborhood}$. Function in F_{point} operate on a pixel to pixel basis independently of neighboring pixels, these functions include all arithmetic operations, non-linear functions and a scalar product. Conversely, $F_{neighborhood}$ contains filters that use a convolution mask and operate on a group of neighboring pixels. These functions are simple Gaussian filters that allow the GP to incorporate information of the local neighborhood around each point. Such operators are crucial because Definition 1 explicitly considers signal variations within a local neighborhood. Finally, to avoid undefined operations during evolution, it is assumed that $I \in \Re^+$ and protected versions of the functions $\div, \sqrt{\cdot}$ and log are used [43].

5. Experimental results

This section provides a detailed description of the GP system, the experimental results, and comparisons with traditional estimators.

5.1. Implementation details

The parameters of the GP system are presented in Table 1, these were tuned empirically for best performance using nearly a hundred test runs which are not reported here due to length considerations. The GP algorithm was programmed using the Matlab toolbox GPLab [48], and estimation of the Hölder exponent with the oscillations method was done using the FracLab toolbox [32].

Given that each execution required several days of computation, and in some cases an entire week, the algorithm was executed only twenty times with the best parametric configuration. Nevertheless, since we are mainly interested in finding the best possible estimators we shall only focus on describing the best five runs. The best estimator produced in each run will
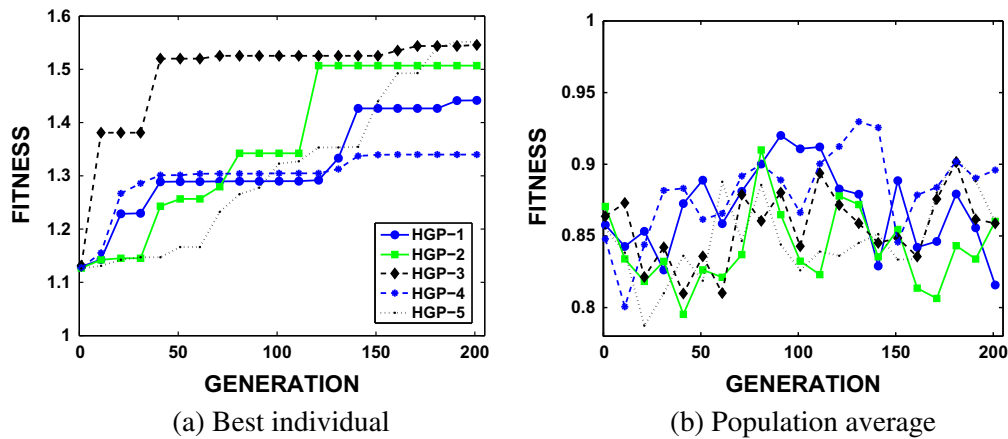
(a) Best individual (b) Population average

Fig. 7. Evolution statistics for the five best GP runs.

be named using the following convention: HGP-<R>. Where, the acronym *HGP* stands for *Hölder estimation with Genetic Programming* and <R> represents the run number.

5.2. Evolution of the GP search

The evolution of fitness from each run is presented in Fig. 7. The figure shows plots for the fitness of the best solution at each generation (a), and the average fitness of the entire population (b). We can see that the best fitness was achieved by HGP-5, HGP-3 and HGP-2, while HGP-4 has the lowest fitness. Table 2 presents the symbolic expression for each of the best estimators evolved by the GP.[2]

5.3. Comparisons

We present a comparison between the estimation of evolved operators and the oscillations method. Additionally, a run-time comparison with other state-of-the-art approaches is also presented.

5.3.1. Quantitative comparisons
First, Table 3 presents a quantitative comparison using the images in Fig. 5. Note that the first four images (NY,VG, Mosaic and Monet) are the ones used during training, while the other six are from the testing set. The comparison is based on the RMSE and 2D correlation coefficient; the table also states the size of each image in pixels.

Table 4 presents comparative results for an extended test set of 30 images. These images are diverse, with scenes of people, buildings, outdoor areas, and computer generated graphics. Table 4 presents the average RMSE and the average squared 2D correlation coefficient between the ground truth estimation of the oscillations method and the evolved estimators. The comparative data presented in Tables 3 and 4 show that HGP-3 and HGP-2 consistently achieve the best correlation and the lowest error. Moreover, the GP search did not produce solutions that are overfitted to the limited set of training examples. While the training set only contains four images, which seems to be a small amount, in fact each image provides a large variety of singular and irregular structures. Thus, the GP is able to produce estimators that perform quite well on a variety of local image patterns.

5.3.2. Qualitative comparisons
Fig. 8 presents a qualitative comparison between the oscillations method and the HGP estimators. The first column contains the Hölder image computed with the oscillations method, and the next two columns show the estimation computed with an evolved estimator. In most cases, the similarity between the evolved estimators and the oscillations method is very high.

5.3.3. Runtime comparisons
Finally, we perform a comparisons of the runtime required to execute each of the evolved estimators in seconds, results are summarized in Table 5. For these tests we use a PC Laptop with a 64 bit AMD processor and 1 GB of system RAM, running Ubuntu 9.04 and Matlab R2007a. Each estimation method was executed thirty times on a test image of 640×480 pixels. It is important to note that in all cases we use the complete program trees that were generated by the GP without removing introns or simplifying the trees in any way. Additionally, the evolved estimators use non-optimal code which is implemented entirely in Matlab. For comparison, we include the runtime of four estimation methods included in the FracLab toolbox, the

---

[2] These expressions have been algebraically simplified for easier understanding.

Table 2
Symbolic expression for evolved estimators.

HGP-1:

$|G_1 * [|G_2 * \log(I)| - k \cdot (|2I - G_1 * G_1 * (G_2 * I + I)| \cdot G_1 * \sqrt{\log(2I + 21)})]|$

HGP-2:

$G_1 * |\log|G_1 * (k \cdot (I - G_1 * I))||$

HGP-3:

$G_1 * |G_2 * \log (k \cdot G_1 * | I - G_1 * G_1 * I|)|$

HGP-4:

$\log \left[(G_2 * I - I) \cdot \left| \log \left(\frac{I}{2}\right) \cdot \left(\frac{k \cdot (I + (\log(2I))^2)}{I - \sqrt{I + G_2 * I}} \right) \right| \right]$

HGP-5:

$\log \left[\sqrt{G_2 * \left(\frac{I - \sqrt{2I + I \cdot (G_1 * (I + G_2 * 2I))}}{\sqrt{|k \cdot (I + I^2) \cdot (G_1 * I - G_1 * G_1 * I)|}} \right)} \right]$

---

only freely available software for regularity analysis [32]. The estimators from FracLab are the oscillations method described above, the wavelet leaders method [16], an estimator based on Formula (1) that uses plain increments, and the parametric method of [2]. All of FracLab estimators use optimized code and some include fast C implementations of some crucial parts of the algorithms, an advantage over the HGP estimators.

The runtime comparisons suggest that the HGP estimators are in fact more efficient, in particular HGP-2 gives the fastest estimation with HGP-3 not far behind. There does not seem to be a practical trade-off between obtaining a good estimation and using a fast algorithm, because HGP-2 and HGP-3 also achieve the best quantitative results. In comparison with the traditional methods, the HGP estimators perform quite well. The oscillations method, for instance, is nearly three orders of magnitude slower. Similarly, wavelet leaders and the parametric method are slower by two orders of magnitude. The fastest algorithm in FracLab is the increments-based estimator, which uses an optimized C implementation. However, it still is 50% slower than HGP-2, a significant difference if we consider real-time applications. We would also stress that HGP-2 achieves a faster runtime despite using sub-optimal Matlab code and the complete GP trees, a noteworthy result. Indeed, the fast estimation given by this operator allowed us to perform real-time estimation of the Hölder exponent on a Unibrain Fire-i Fire-Wire camera. We implemented operator HGP-2 using the C++ vision library LibCVD,[3] which allowed us to obtain a frame-rate of 30 fps,[4] with a video resolution of 640 × 480; sample frames are shown in Fig. 9.

In summary, this section presents an extensive comparison between the evolved estimators and traditional methods. Indeed, we have shown that the GP approach can produce very good estimators of Hölderian regularity which closely approximate the estimation achieved by the oscillations method. Furthermore, the evolved HGP estimators are significantly faster than traditional methods, between 50% faster and up to several orders of magnitude faster. In particular, the best and fastest estimation was achieved by the HGP-2 operator, which is based on the logarithm of a scaled difference-of-Gaussians filter (see Table 2).

6. Application to local image description

Despite the encouraging results presented above, a question remains: can the HGP estimators provide a useful estimation for higher level applications? Here, this question is addressed by applying the evolved operators on a difficult computer vision problem.

Recently, many computer vision systems are based on the detection and description of local and sparse image features. The approach was introduced in [45,33] and consists of the following basic steps. First, small image regions centered around salient pixels, better known as *interest points*, are detected using specially designed image operators [60]. Then, each of these regions is described using compact numerical vectors that capture the main characteristics of local image shape and appearance, these vectors are called *local image descriptors* [38]. The set of local regions and their corresponding descriptors are then used to construct models of the objects, or scene, present within an image. When a new image is analyzed this process is repeated and the extracted features are compared with stored models. Therefore, to perform a recognition task, the vision system searches for local correspondences such as the one depicted in Fig. 10. The main advantages of this approach are: (1) it does not require traditional image segmentation, a difficult mid-level task; (2) the approach is robust to partial occlusions and to several types of image transformations; and (3) the total amount of information is sharply reduced because only a subset of image regions are analyzed and described.

Keeping to the problem of local description, many proposals have been made over the last 15 years. However, currently the Scale Invariant Feature Transform (SIFT) [33] is still widely accepted as the standard method in current literature [38]. The overall success of SIFT has led some researcher to develop improved versions of the algorithm, one related example is the GP-optimized SIFT [41]. Another descriptor was proposed in [58], based on sampling the pointwise Hölder exponent within a

---

[3] http://www.edwardrosten.com/cvd/index.html.
[4] This is the maximum frame rate of the Fire-i camera.

Table 3
Quantitative comparison of the evolved estimators and the oscillations method. The table shows the name and size of each test image, the RMSE and the 2D correlation value; bold indicates best.

Image	Measures	HGP-1	HGP-2	HGP-3	HGP-4	HGP-5
NY$_{(512 \times 512)}$	Error (10^{-3})	0.218	0.182	**0.116**	0.257	0.172
	Correlation	0.746	0.871	**0.917**	0.648	0.875
VG$_{(348 \times 512)}$	Error	0.357	0.317	**0.151**	0.456	0.311
	Correlation	0.860	0.890	**0.948**	0.658	0.895
Monet$_{(842 \times 842)}$	Error	0.265	0.201	0.249	0.105	**0.102**
	Correlation	0.696	**0.876**	0.860	0.872	0.869
Mosaic$_{(512 \times 512)}$	Error	0.143	0.107	**0.071**	0.133	0.107
	Correlation	0.746	0.826	**0.896**	0.625	0.842
House$_{(484 \times 768)}$	Error	0.677	0.484	0.924	**0.368**	0.712
	Correlation	0.2610	**0.845**	0.776	0.639	0.744
Door$_{(256 \times 256)}$	Error	0.226	0.307	**0.195**	0.366	0.341
	Correlation	0.864	0.8928	**0.940**	0.725	0.874
Bip$_{(768 \times 574)}$	Error	0.118	0.095	**0.094**	0.166	0.128
	Correlation	0.771	**0.805**	0.794	0.651	0.691
Lap$_{(768 \times 574)}$	Error	0.171	**0.090**	0.120	0.118	0.134
	Correlation	0.518	**0.834**	0.815	0.762	0.752
Tree$_{(1000 \times 700)}$	Error	0.177	**0.099**	0.390	0.183	0.210
	Correlation	0.677	0.866	**0.871**	0.590	0.859
UBC$_{(800 \times 640)}$	Error	0.182	**0.128**	0.135	0.160	0.154
	Correlation	0.697	0.916	**0.942**	0.607	0.904

Table 4
Quantitative comparison of the evolved estimators using the test set of 30 images; best results are indicated with bold.

Estimator	HGP-1	HGP-2	HGP-3	HGP-4	HGP-5
Error 10^{-3}	0.2184	**0.1479**	0.1740	0.1660	0.1789
Square correlation	0.4423	0.6917	**0.7503**	0.4316	0.6515

local region. The *Hölder descriptor* achieves comparable performance to SIFT on standard tests [58]. Those results confirm the ability of the Hölder exponent to effectively describe the local structure within a signal. One important limitation of SIFT, is its computational complexity which makes it ill-suited for real-time tasks.[5] Similarly, since the original Hölder descriptor relies on traditional estimation methods it also comes with a high computational overhead. In previous work, this shortcoming was addressed using dimensionality reduction with a GA [55]. In this work, however, we employ the evolved HGP estimators to construct the local Hölder descriptor and compare the performance with the descriptor based on the oscillations method.

6.1. Hölder descriptor

The process used to build the local Hölder descriptor proceeds as follows. First, a set of interest regions are extracted from an image. Second, the dominant gradient orientation within each region is computed, to preserve rotation invariance. Finally, the descriptor vector contains a sampling of the Hölder exponent on 129 concentric points using a polar grid and ordering them based on the dominant orientation.

6.1.1. Region extraction

The first step requires stable detection of salient image regions. The type of regions will depend on the requirement of the higher level application with respect to invariance. For instance, an interest point detector is sufficient when the scale of the image is not modified. Here, we use a detector optimized for geometric stability and global point separability, the *IPGP*2 detector which is the determinant of the Hessian matrix smoothed by a 2D Gaussian kernel [56,57]. All regions extracted with an interest point detector are assigned the same scale, $w_\lambda = 2.5$ pixels. For images where scale is a factor, we use the Hessian–Laplace detector [37], which searches for extrema in the scale space generated with a Gaussian kernel. After this step we are left with a set Λ of circular regions. The size of the image region used to compute the local descriptor (support region) is set to $s_\lambda = 5 \cdot w_\lambda$, where w_λ is the scale of the region. Then, all image regions are cropped and normalized to 61×61 pixels using bicubic interpolation.

---

[5] Many works deal with this topic, and faster implementations have been proposed such as SURF and GPU-SIFT.
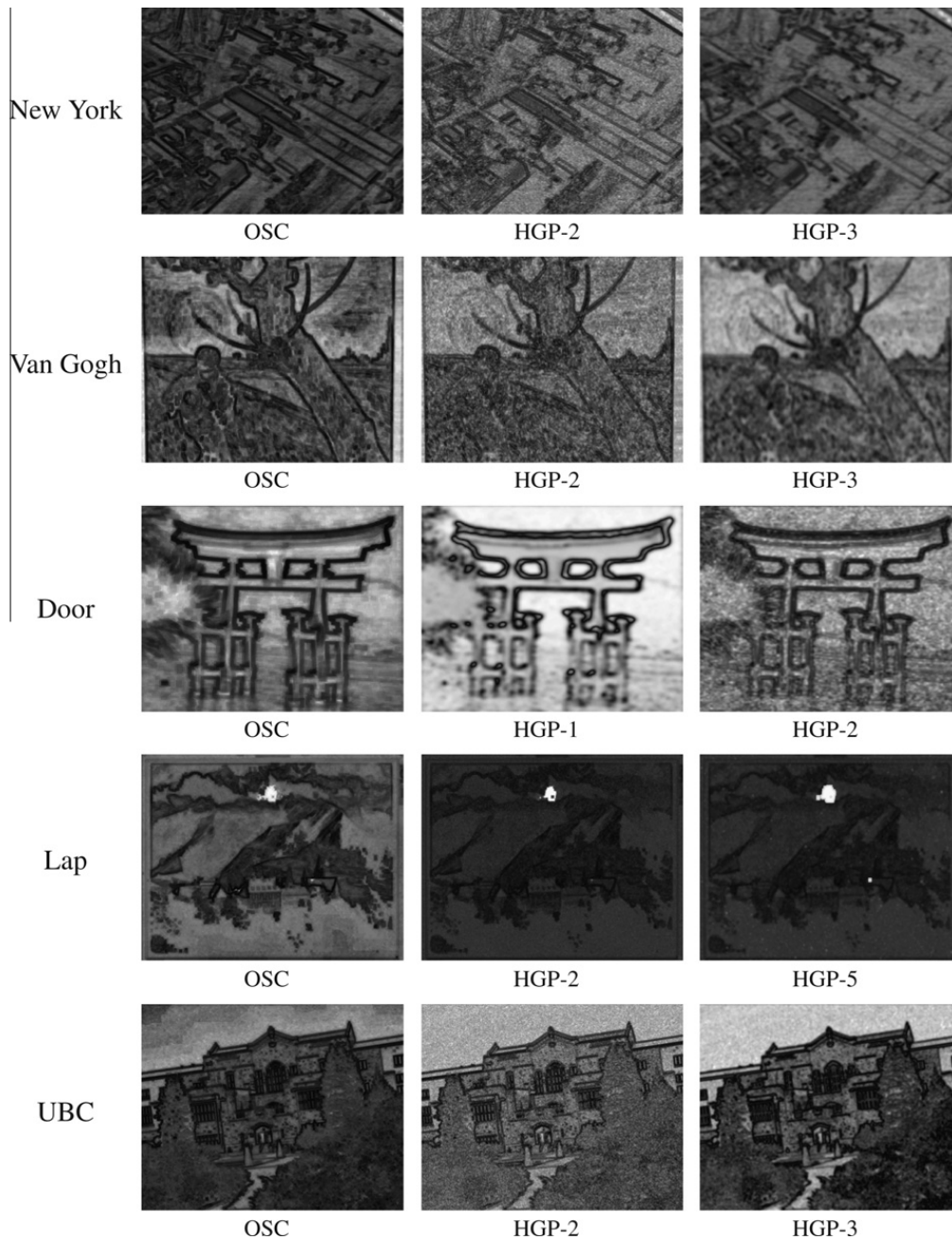
Fig. 8. Qualitative comparisons between the evolved estimators and the oscillations method. First column contains the Hölder image computed with the oscillations method, and each row corresponds to a different test image.

Table 5
Runtime comparison between the evolved estimators and traditional approaches. The estimators are tested on a 640×480 image, all results are shown in seconds and represent the average over thirty executions; bold indicates best.

Evolved estimators	HGP-1	HGP-2	HGP-3	HGP-4	HGP-A
Time	0.62	**0.19**	0.30	0.76	0.54

FracLab		Osc.	Wavelets	Increments	Parametric
Time		360	66.30	0.30	31.5

6.1.2. Dominant orientation

For rotation invariance, the dominant gradient orientation is computed and used as a reference for the subsequent sampling process. A histogram is constructed using gradient orientations within the interest region, similar to what is done in [33]. The histogram peak is obtained $\forall \lambda \in \Lambda$ and a corresponding dominant orientation ϕ_λ is assigned. Each region is described by a 4-tuple $\lambda = (x_\lambda, y_\lambda, s_\lambda, \phi_\lambda)$, that respectively define the region center, scale, and dominant orientation.
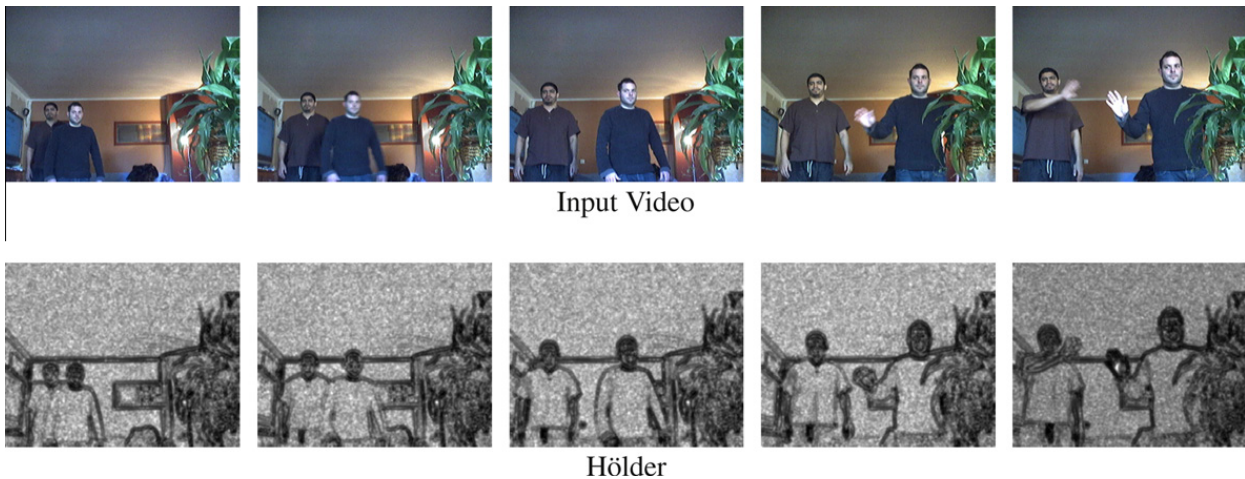
Fig. 9. Sample frames captured for a realtime estimation of the Hölder exponent using the evolved HGP-2A estimator. The top row are the input video frames and the bottom row shows the estimated Hölder regularity.
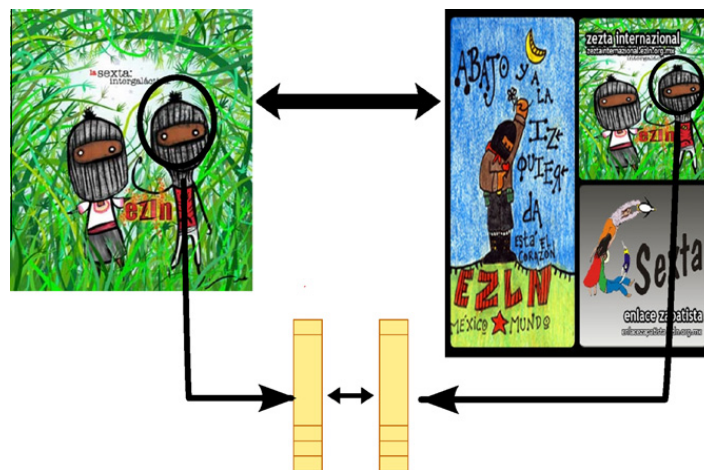

Fig. 10. The matching process using locally salient features. A local descriptor is computed for an interest region detected on the left image. Then, a correspondence is sought between it and the local descriptors extracted from the image on the right. The figure shows how a correct match between descriptive vectors can assure a correct match between corresponding regions.

6.1.3. Descriptor

After the salient image regions are detected and described with λ, it is possible to construct a local descriptor $\delta_\lambda, \forall \lambda \in \Lambda$. As stated before, the descriptor is a uniform sampling of the pointwise Hölder exponent within each region using a circular grid, see Fig. 11. The first element of δ_λ is the Hölder exponent α_p computed at the region center (x_λ, y_λ). Next, the Hölder exponent of points on the perimeter of four concentric rings is sampled, with radii of $\frac{1}{4} \cdot s_\lambda$, $\frac{1}{2} \cdot s_\lambda$, $\frac{3}{4} \cdot s_\lambda$ and s_λ respectively. A total of 32 points on each ring are sampled, starting from the position given by ϕ_λ, all uniformly spaced and ordered counterclockwise. Therefore, the feature vector δ_λ has 129 dimensions; for comparison, the SIFT descriptor has 128 dimensions.

6.2. Experimental tests

For evaluation, standard image sequences provided by the Visual Geometry Group and the Lear team at INRIA are used. Table 6 gives further details regarding the images used in the experimental tests, and information can be obtained from each groups website or in [38]. There is a total of eight different sequences, including four with rotation transformations (NY, BG, Mars, and Monet), two with illumination change (Graph and Mosaic), one with JPEG compression (UBC) and another one with scale changes (Laptop). From each sequence there is one reference image and several test images, each progressively transformed. To test the descriptor, the strategy is to use the reference image and one transformed image, then the detected regions between both images are matched using the descriptor vectors, see Fig. 10. Because we possess prior knowledge regarding the transformation between the reference and the transformed image, we can effectively determine if the matches produced by the local descriptors are correct [38].
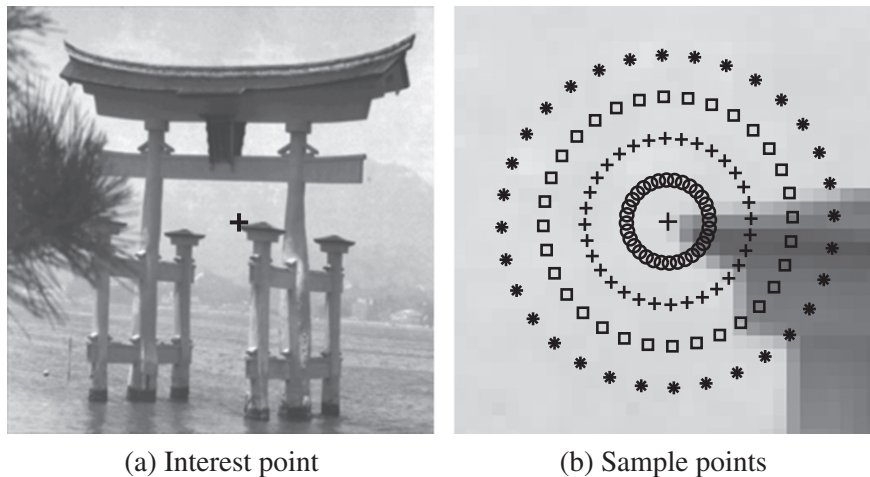
(a) Interest point   (b) Sample points

Fig. 11. Sample points used to build the Hölder descriptor around a detected interest point.

Table 6
Test pairs used to evaluate the performance of the evolved estimators on the task of region matching.

Sequence	Transformation	# Of test image
New York	Rotation	11
Van Gogh	Rotation	7
Mars	Rotation	10
Monet	Rotation	6
Graph	Illumination change	6
Mosaic	Illumination change	12
UBC	JPEG compression	2
Laptop	Scale change	7

In this work, two image regions λ_1 and λ_2 are matched if $d(\delta_{\lambda_1}, \delta_{\lambda_2}) < t_\delta$ and if δ_{λ_2} fulfills the nearest-neighbor criterion for δ_{λ_1}, where $d(\cdot, \cdot)$ is the Euclidean distance. The performance of a descriptor with regards to local matching can then be assessed by varying t_δ to obtain *Recall* versus *1-Precision* curves, which characterize the matching process between two images [38]. *Recall*/*1-Precision* provides information regarding the number of correct and false matches between two images. *Recall* is the number of correctly matched regions with respect to the number of corresponding regions between two images of the same scene. The number of false matches relative to the total number of matches is given by 1-*Precision*. A perfect descriptor would give a *Recall* equal to 1 for any 1-*Precision*. *Recall* and 1-*Precision* are defined as

$$Recall = \frac{\#correct\_matches}{\#correspondences},$$

$$1 - Precision = \frac{\#false\_matches}{\#correct\_matches + \#false\_matches}.$$

6.3. Results

Test results are organized according to the type of transformation that each sequence presents: rotation, illumination change, JPEG compression, and scale change. For each sequence we show the base image and one test image, as well as the corresponding *Recall*/1-*Precision* curve for each of the evolved estimators along with the performance of the original Hölder descriptor.

6.3.1. Rotation

As noted above, there are four test cases for rotation transformations, Fig. 12 for the NY pair of images, Fig. 13 for the VG pair, Fig. 14 for the Mars pair, and Fig. 15 for the Monet pair. In these tests HGP-2 and HGP-3 consistently achieve a high performance, practically the same as the original Hölder descriptor. These trends are consistent with the results shown in Table 3, where these estimators obtained the lowest error and highest correlation values.

6.3.2. Illumination change

There are two test cases for illumination change, Fig. 16 for the Graph images and Fig. 17 for the Mosaic pair. For these tests almost all of the estimators produce a comparable performance to that of the original Hölder descriptor, with the
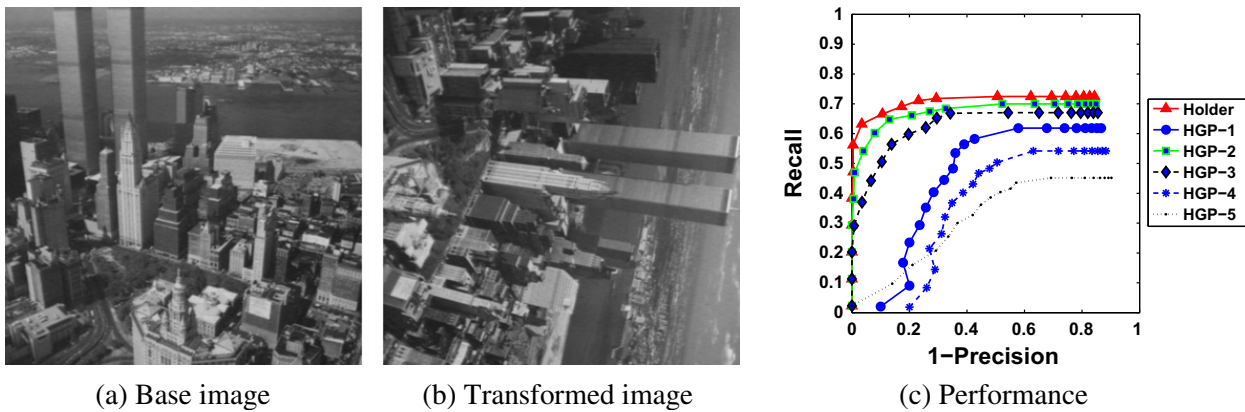
(a) Base image (b) Transformed image (c) Performance

Fig. 12. *Recall*/1-*Precision* curves for the New York image pair. The transformation between (a) and (b) is a rotation.
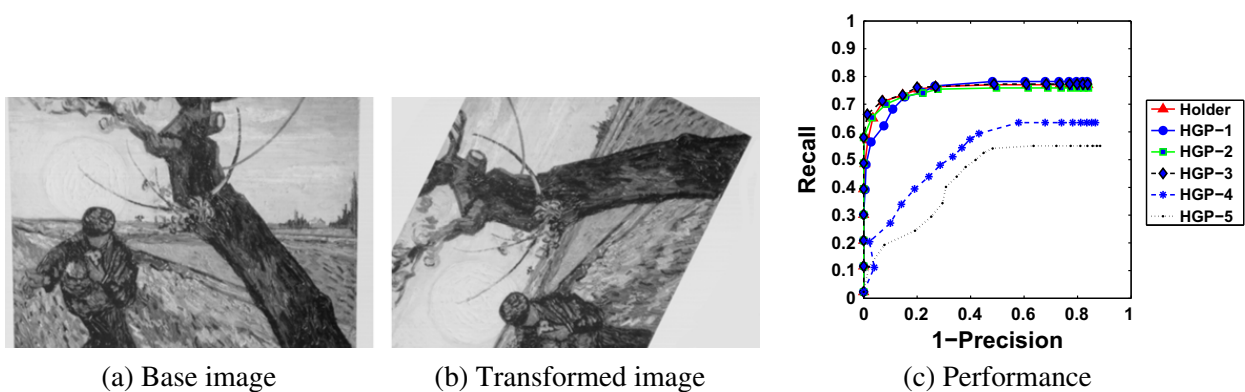

(a) Base image (b) Transformed image (c) Performance

Fig. 13. *Recall*/1-*Precision* curves for the Van Gogh image pair. The transformation between (a) and (b) is a rotation.
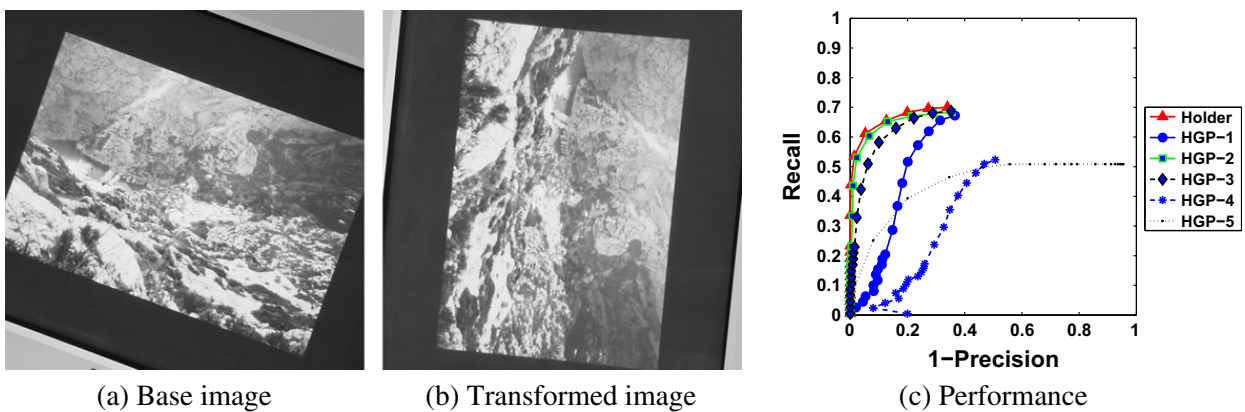

(a) Base image (b) Transformed image (c) Performance

Fig. 14. *Recall*/1-*Precision* curves for the Mars image pair. The transformation between (a) and (b) is a rotation.

following notable observations. First, only HGP-1 and HGP-4 produce notably inferior matching scores. Second, HGP-2 outperforms the original descriptor on the Graph test.

6.3.3. JPEG compression

To test the performance of the matching process with respect to distortions introduced by JPEG compression we employ the images shown in Fig. 18. Some of the evolved estimators perform quite well on this test, including HGP-1, HGP-2, and HGP-3; in all cases performance is above that achieved by the oscillations method. Indeed this was not expected for two reasons. First, it is normally assumed that image regularity will be changed drastically when JPEG compression is applied, however some of the evolved estimators can cope quite well. Second, the performance of most image descriptors is normally degraded by JPEG compression, even SIFT [38]. However, using the HGP estimators the Hölder descriptor achieves good results.
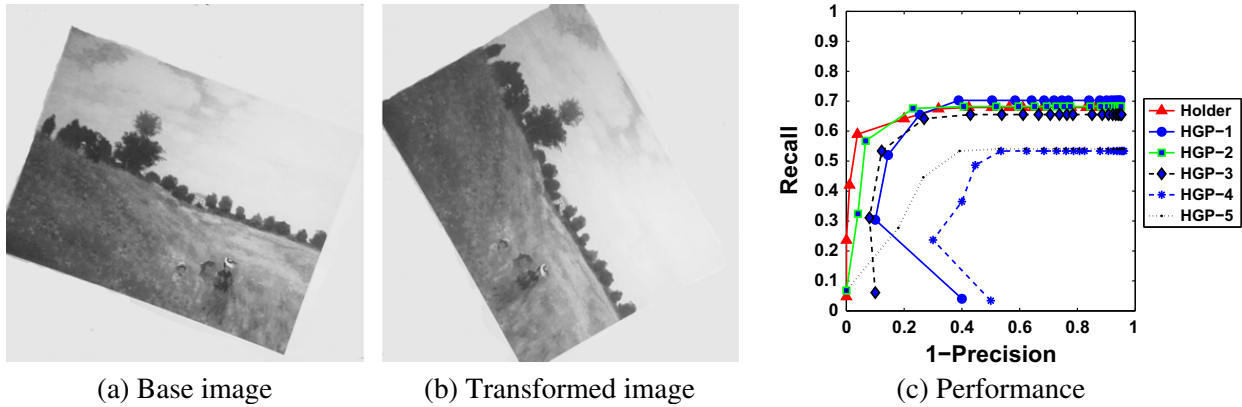
(a) Base image   (b) Transformed image   (c) Performance

Fig. 15. *Recall*/1-*Precision* curves for the Monet image pair. The transformation between (a) and (b) is a rotation.
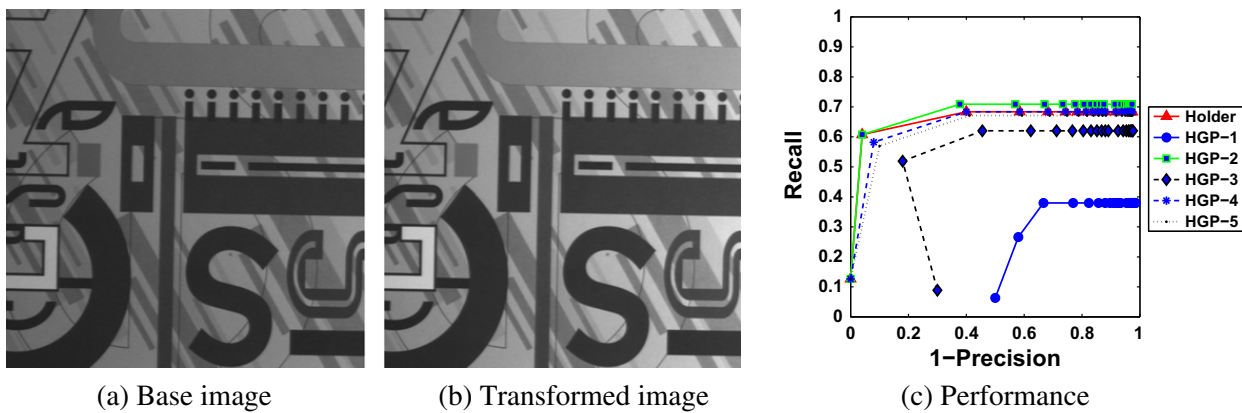

(a) Base image   (b) Transformed image   (c) Performance

Fig. 16. *Recall*/1-*Precision* curves for the Graph image pair. The transformation between (a) and (b) is illumination change.

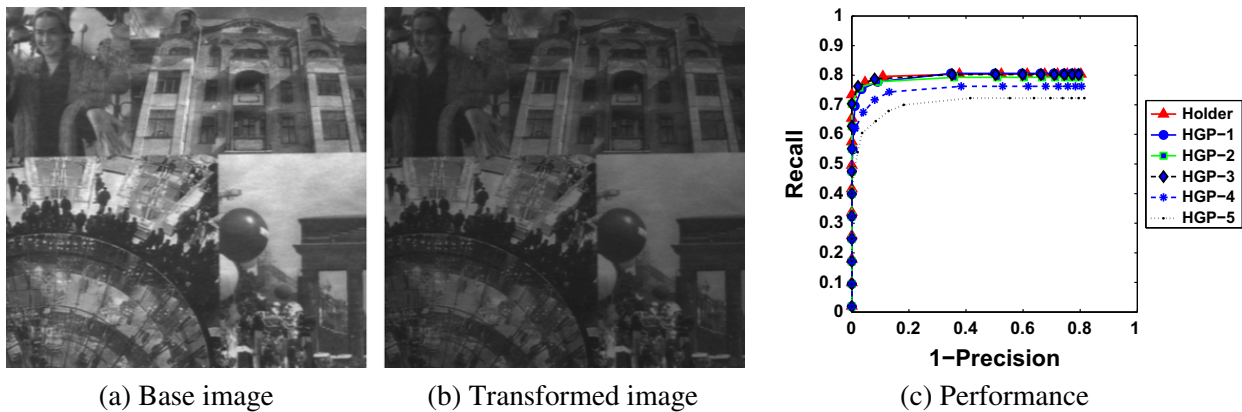(a) Base image   (b) Transformed image   (c) Performance

Fig. 17. *Recall*/1-*Precision* curves for the Mosaic image pair. The transformation between (a) and (b) is illumination change.

6.3.4. Scale change

The performance of the matching process with respect to scale changes was evaluated using the images shown in Fig. 19. In this test, only HGP-1, HGP-3 and HGP-2 obtain similar performance to that achieved by the oscillations method.

In general, these tests show that some of the evolved estimators can be used to effectively describe local shape and appearance, and achieve a similar performance to the oscillations method [58]. In particular, estimators HGP-2 and HGP-3 obtain a high performance across all tests, and in some instances are better than the original descriptor. The performance of these operators was indeed expected, given their low RMSE and high correlation with the oscillations method. In fact, using HGP-2 we can build a very simple, efficient and robust image descriptor that can be computed in a small fraction of the time that more traditional methods require. This could allow us to develop faster algorithms for real-time computer vision systems.
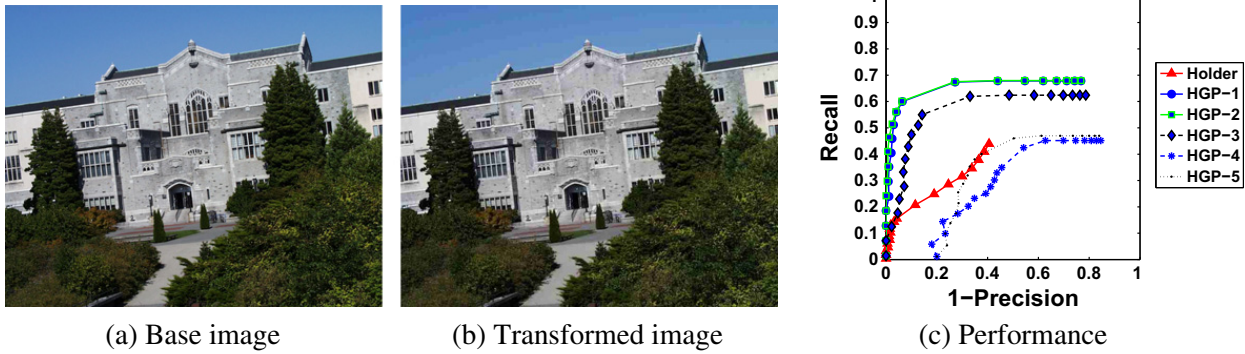
(a) Base image   (b) Transformed image   (c) Performance

Fig. 18. *Recall*/1-*Precision* curves for the UBC image pair. The transformation between (a) and (b) is JPEG compression.

(a) Base image   (b) Transformed image   (c) Performance

Fig. 19. *Recall*/1-*Precision* curves for the Laptop image pair. The transformation between (a) and (b) is a scale change or zoom.

7. Summary and concluding remarks

In this paper, the task of developing a new estimator of image regularity is posed as an optimization problem and solved using Genetic Programming. The goal is to synthesize image operators that can approximate the oscillations method for Hölder exponent estimation. Additionally, the evolved estimators should also be simpler, easier to implement, and exhibit a lower runtime. Indeed, a standard implementation of GP was able to solve this problem and satisfy the desired criteria. This allowed us to develop a real-time estimator of Hölder regularity for a live video signal, the first such algorithm in current literature.

To test the estimators in a real-world task, they are applied to one of the most important problems in modern computer vision, describing locally salient image features. Standard experimental tests and performance criteria showed that the evolved HGP estimators are capable of extracting a meaningful description characterization of local shape and appearance. Such results can surely help expand the applicability of regularity-based techniques to other problems in image processing and computer vision.

From among the evolved estimators, one in particular called HGP-2 achieved the best results, based on estimation error, correlation, runtime, and image description. It is interesting to note the simple logic behind this estimator, it basically relies on computing the logarithm of the absolute value of a difference-of-Gaussian filter. This suggests that the Hölder exponent at each point can be effectively approximated by one of the most basic feature detection methods used in computer vision, proposed by David Marr in he's seminal works [36]. The result is both surprising and logical, because the basic concept of irregular or singular structures is obviously related to the type of saliency that Marr so effectively modeled. However, given the complexity of some of the traditional estimators we did not expect that such a simple operator could replicate, and in some sense improve upon, their performance.

The use of the HGP-2 estimator for local image description reveals that this task can be solved using much simpler and direct algorithms than those currently employed. Even a cursory review on this topic reveals that state-of-the-art methods employ complex models and algorithms when building discriminant feature descriptors [38]. One exception is the Hölder descriptor [58], which is based on a simple sampling algorithm. However, the original version of the Hölder descriptor also relies on a computationally slow estimation process, which severely limits its usefulness. However, the HGP estimators allow us to enhance the Hölder descriptor by reducing the runtime of the estimation process. Hence, all that is required is to estimate the Hölder exponent using the HGP-2 operator and then sample the exponent on a fixed polar grid. This simple approach allows us to generate a robust and invariant descriptor that achieves state-of-the-art performance and can be implemented in real-time.

Acknowledgements

References

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

[2] A. Ayache, J. Lévy-Véhel, On the identification of the pointwise hlder exponent of the generalized multifractional brownian motion, Stochastic Processes and their Applications 111 (1) (2004) 119–156.

[3] R.M.A. Azad, C. Ryan, Abstract functions and lifetime learning in genetic programming for symbolic regression, in: GECCO '10: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2010, pp. 893–900.

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

[5] H. Barnum, H. Bernstein, L. Spector, Quantum circuits for or and of ors, Journal of Physics A: Mathematical and General 33 (45) (2000) 8047–8057.

[6] O. Barriere, Synthése et estimation de mouvements browniens multifractionnaires et autres processus á régularité prescrite. définition du processus auto-régulé multifractionnaire et applications, Ph.D. thesis, Ecole centrale de Nantes et Universite de Nantes, France, 2007.

[7] S. Cagnoni, E. Lutton, G. Olague, (Eds.), Genetic and Evolutionary Computation for Image Processing and Analysis, EURASIP Book Series on Signal Processing and Communications, vol. 8, Hindawi Publishing Corporation, 2008.

[8] F.-C. Chang, H.-C. Huang, A refactoring method for cache-efficient swarm intelligence algorithms, Information Sciences 192 (2012) 39–49.

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

[10] M. Ebner, A real-time evolutionary object recognition system, in: L. Vanneschi, S. Gustafson, A. Moraglio, I.D. Falco, M. Ebner (Eds.), Genetic Programming, Springer, 2009, pp. 268–279.

[11] K. Falconer, Fractal Geometry: Mathematical Foundations and Applications, Wiley, 1990.

[12] S. Gustafson, E.K. Burke, 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, IEEE, 2005, pp. 912–919.

[13] B. Hernández, G. Olague, R. Hammoud, L. Trujillo, E. Romero, Visual learning of texture descriptors for facial expression recognition in thermal imagery, Computer Vision and Image Understanding, Special Issue on Vision Beyond the Visual Spectrum 106 (2–3) (2007) 258–269.

[14] D. Howard, S.C. Roberts, R. Brankin, Target detection in sar imagery by genetic programming, Advances in Engineering Software 30 (5) (1999) 303–311.

[15] H. Iba, Inference of differential equation models by genetic programming, Information Sciences 178 (23) (2008) 4453–4468.

[16] S. Jaffard, Wavelet techniques in multifractal analysis, in: Fractal Geometry and Applications: A Jubilee of Benoit Mandelbrot, Proceedings of Symposia in Pure Mathematics, vol. 72, 2004, pp. 91–151.

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

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

[19] J. Koza, Human-competitive results produced by genetic programming, Genetic Programming and Evolvable Machines 11 (3) (2010) 251–284.

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

[21] J.R. Koza, M.A. Keane, J. Yu, H. Forrest, I. Bennett, W. Mydlowec, Automatic creation of human-competitive programs and controllers by means of genetic programming, Genetic Programming and Evolvable Machines 1 (1–2) (2000) 121–164.

[22] J.R. Koza, M.J. Streeter, M.A. Keane, Routine high-return human-competitive automated problem-solving by means of genetic programming, Information Sciences 178 (23) (2008) 4434–4452.

[23] K. Krawiec, Genetic programming-based construction of features for machine learning and knowledge discovery tasks, Genetic Programming and Evolvable Machines 3 (4) (2002) 329–343.

[24] K. Krawiec, B. Bhanu, Visual learning by coevolutionary feature synthesis, IEEE Transactions on Systems, Man, and Cybernetics, Part B 35 (3) (2005) 409–425.

[25] K. Krawiec, D. Howard, M. Zhang, Overview of object detection and image analysis by means of genetic programming techniques, in: Frontiers in the Convergence of Bioscience and Information Technologies 2007, FBIT 2007, Jeju Island, Korea, October 11–13, IEEE Computer Society, 2007, pp. 779–784.

[26] W.B. Langdon, R. Poli, Foundations of Genetic Programming, Springer-Verlag, New York, New York, 2002.

[27] P. Legrand, Debruitage et interpolation par analyse de la regularite hölderienne, Application a la modelisation du frottement pneumatique-chaussee, Ph.D. thesis, Université de Nantes, France, 2004.

[28] P. Legrand, J. Lévy-Véhel, Local regularity - based image denoising, ICIP03, Spain, IEEE International Conference on Image Processing, September 14–17, 2003, pp. 377–380.

[29] P. Legrand, J. Lévy-Véhel, Local regularity-based interpolation, in: WAVELET X, Part of SPIE's Symposium on Optical Science and Technology, vol. 5207, 2003.

[30] J. Lévy Véhel, Fractal approaches in signal processing, in: C. Evertsz, H.-O. Peitgen, R. Voss (Eds.), Fractal Geometry and Analysis: The Mandelbrot festschrift, World Scientific, 1996, pp. 755–775.

[31] J. Lévy-Véhel, Fractal Image Encoding and Analysis, Springer Verlag, 1998. pp. 299–341 (Chapter Introduction to the Multifractal Analysis of Images).

[32] J. Lévy-Véhel, P. Legrand, Thinking in Patterns, World Scientific, 2004. pp. 321–322 (Chapter Signal and Image Processing with FRACLAB. <http://fraclab.saclay.inria.fr/homepage.html>).

[33] D.G. Lowe, Object recognition from local scale-invariant features, Proceedings of the International Conference on Computer Vision (ICCV), 20–25 September, Kerkyra, Corfu, Greece, vol. 2, IEEE Computer Society, 1999, pp. 1150–1157.

[34] M.T. Mahmood, A. Majid, T.-S. Choi, Optimal depth estimation by combining focus measures using genetic programming, Information Sciences 181 (7) (2011) 1249–1263.

[35] S. Mallat, A Wavelet Tour of Signal Processing, 2nd ed., Elsevier, San Diego, CA, 1999.

[36] D. Marr, Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, Henry Holt and Co., Inc., New York, NY, 1982.

[37] K. Mikolajczyk, C. Schmid, Scale & affine invariant interest point detectors, International Journal of Computer Vision 60 (1) (2004) 63–86.

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

[39] G. Olague, L. Trujillo, Evolutionary-computer-assisted design of image operators that detect interest points using genetic programming, Image and Vision Computing 29 (7) (2011) 484–498.

[40] C.B. Pérez, G. Olague Learning invariant region descriptor operators with genetic programming and the f-measure, in: 19th International Conference on Pattern Recognition (ICPR 2008), December 8–11, 2008, Tampa, Florida, USA, IEEE, 2008, pp. 1–4.

[41] C.B. Perez, 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, ACM, New York, NY, USA, 2009, pp. 1051–1058.

[42] R. Poli, Genetic programming for feature detection and image segmentation, in: T.C. Forgarty, (Ed.), AISB Workshop Evolutionary Computing, 1996, pp. 110–125.

[43] R. Poli, W.B. Langdon, N.F. McPhee, A field guide to genetic programming, 2008. <http://lulu.com and freely available at http://www.gp-field-guide.org.uk> (with contributions by J.R. Koza).

[44] C. Puente, G. Olague, S. Smith, S. Bullock, A. Hinojosa-Corona, M. González-Botello, A genetic programming approach to estimate vegetation cover in the context of soil erosion assessment, Photogrametric Engineering and Remote Sensing 77 (4) (2011) 363–376.

[45] C. Schmid, R. Mohr, Local grayvalue invariants for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (5) (1997) 530–534.

[46] M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data, Science 324 (5923) (2009) 81–85.

[47] S. Se, D. Lowe, J. Little, Global localization using distinctive visual features, in: Proceedings of the International Conferences on Intelligent Robots and Systems (IROS), Lausanne, Switzerland, 2002, pp. 226–231.

[48] S. Silva, J. Almeida, Gplab–a genetic programming toolbox for matlab, in: L. Gregersen, (Ed.), Proceedings of the Nordic MATLAB Conference, 2003, pp. 273–278.

[49] S. Silva, E. Costa, Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories, Genetic Programming and Evolvable Machines 10 (2) (2009) 141–179.

[50] A. Song, V. Ciesielski, Texture segmentation by genetic programming, Evolutionary Computation 16 (4) (2008) 461–481.

[51] L. Spector, D.M. Clark, I. Lindsay, B. Barr, J. Klein, Genetic programming for finite algebras, in: GECCO '08: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2008, pp. 1291–1298.

[52] X. Tan, B. Bhanu, Y. Lin, Fingerprint classification based on learned features, IEEE Transactions on Systems, Man, and Cybernetics, Part C 35 (3) (2005) 287–300.

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

[54] L. Trujillo, P. Legrand, J. Lévy-Véhel, The estimation of hölderian regularity using genetic programming, in: GECCO '10: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2010, pp. 861–868.

[55] L. Trujillo, P. Legrand, G. Olague, C. Pérez, Optimization of the hölder image descriptor using a genetic algorithm, in: GECCO '10: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2010, pp. 1147–1154.

[56] L. Trujillo, G. Olague, Synthesis of interest point detectors through genetic programming, in: M. Cattolico, (Ed.), Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), vol. 1, Seattle, Washington, ACM, July 8–12, 2006, pp. 887–894.

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

[58] L. Trujillo, G. Olague, P. Legrand, E. Lutton, Regularity based descriptor computed from local image oscillations, Optics Express 15 (10) (2007) 6140–6145.

[59] L. Trujillo, G. Olague, E. Lutton, F. Fernández de Vega, Multiobjective design of operators that detect points of interest in images, in: M. Cattolico (Ed.), Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), ACM, Atlanta, GA, New York, NY, USA, 2008, pp. 1299–1306.

[60] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: a survey, Foundations and Trends in Computer Graphics and Vision 3 (3) (2008) 177–280.

[61] Y.-H. Wang, Image indexing and similarity retrieval based on spatial relationship model, Information Sciences 154 (1-2) (2003) 39–58.

[62] Y. Zhang, 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, ACM, New York, NY, USA, 2005, pp. 795–802.