Evolving Estimators of the Pointwise Holder Exponent with Genetic Programming

Leonardo Trujillo, Pierrick Legrand, Gustavo Olague, Jacques Lévy-Vehel

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]*Instituto Tecnológico de Tijuana, Av. Tecnológico S/N, Fracc. Tomás Aquino, Tijuana, B.C., México*
[b]*Université Victor Segalen Bordeaux 2 and the IMB, Institut de Mathématiques de Bordeaux, in UMR CNRS 5251*
[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*

Abstract

The analysis of image regularity using Hölder exponents can be used to characterize singular structures contained within an image, and provide a compact description of local shape and appearance. However, estimating the Hölder exponent is not a trivial task and current methods tend to be slow and complex. Therefore, the goal in this work is to automatically synthesize image operators that can be used to estimate the Hölder regularity of an image. We pose this task as an optimization problem and use Genetic Programming (GP) to search for operators that can approximate a traditional estimator, the oscillations method. In our experiments, GP was able to evolve estimators that achieve a low error and a high correlation with the ground truth estimation. Furthermore, most of the GP estimators are faster than the 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 we obtain a stable and robust matching that is comparable with state-of-the-art methods. In conclusion, the evolved estimators produced by GP could help expand the application domain of

Email addresses: `leonardo.trujillo.ttl@gmail.com` (Leonardo Trujillo), `pierrick.legrand@u-bordeaux2.fr` (Pierrick Legrand), `olague@cicese.mx` (Gustavo Olague), `Jacques.levy-vehel@inria.fr` (Jacques Lévy-Véhel)

Hölderian regularity within the fields of image analysis and signal processing.

1. Introduction

Image analysis entails the detection and extraction of meaningful descriptive features from digital images, to carry out higher level tasks such as object recognition, image indexing, and vision-based tracking, to mention but a few common examples. For most application domains, the most prominent and informative parts of an image correspond with those regions that exhibit an irregular structure with a high amount of local variation. Therefore, many works have addressed the problem of detecting and describing these salient image regions [35, 55, 52, 53]. One approach towards describing the local shape and appearance within an image is through the concept of signal regularity, an approach that can be used to characterize the singularities contained within non-differentiable signals [10, 32]. Therefore, regularity-based analysis has been used to describe local image patches [27, 53] and to detect salient image features [29].

In this paper, we focus on Hölder regularity, which can be used to quantify the singularity, or amount of irregularity, that is present at any given point, using what is known as the pointwise Hölder exponent [10, 32] (see Section 3.1 for a formal definition). While the Hölder exponent has shown to be a useful tool for image analysis [27, 53, 29], the process of computing the exponent is not trivial. 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 [48, 15, 29, 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 tend to be computationally complex and time consuming,

which limits their use in domains that require real-time processing. 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 might open new application domains for regularity-based image analysis.

Therefore, the goal is to find operators that can provide a positive answer to the above question, and to achieve this goal 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 can automatically synthesize solutions for complex tasks. Moreover, unlike black-box methods GP can produce solutions that are amenable to further analysis and understanding [52, 38], though this might prove to be a difficult endeavor in some cases [17]. GP has also proven to be quite flexible, with successful applications in various fields [20, 18], that include image analysis [23, 13, 7, 37, 36, 41], and applied mathematics [5, 46]. Indeed, the power and flexibility of GP comes from the fact that it solves two tasks simultaneously: searching for the desired functionality and also determining the structure of the final solution [19, 24].

For these reasons, we use GP to search for specialized image operators that estimate the pointwise Hölder exponent for digital images. The experimental results show that GP is capable of evolving highly competitive estimators, that are able to approximate the estimation produced by a traditional approach, the oscillations method, with a small amount of error and a high correlation. Moreover, the GP estimators are far more efficient, in terms of computation time they achieve a 50% improvement with respect to one method of estimation, and several orders of magnitude with respect to other approaches from current literature. The quality of the evolved estimators is verified using a common problem of modern computer vision, the description and matching of local image features[35]. In this task, results show that the evolved estimators can be used to construct meaningful and discriminative local descriptors; in fact, the estimators compare favorably with the original Hölder-based descriptor proposed in [53, 50]. Finally, a noteworthy result is that GP found a novel computational operator that extracts a measure of image regularity using a compact and simple operation that can be

3

implemented in real-time.

The remainder of this paper proceeds as follows. Section 2 contains a brief overview of related works that have applied GP to image analysis and mathematics. Then, Section 3 provides an introduction to Hölder regularity and brief introduction to genetic programming. In Section 4, we pose the task of estimating the Hölder regularity of an image as an optimization problem and present a GP approach to solve it. The experimental results are detailed in Section 5, and qualitative and quantitative comparisons are made between the evolved estimators and the oscillations method. Moreover, in Section 6 the evolved estimators are used to build local image descriptors and are compared with the canonical Hölder descriptor from [53]. Finally, a summary and concluding remarks are given in Section 7.

2. Related work

The present work is related with two application domains of GP: image analysis and applied mathematics.

Image analysis encompasses a large and 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 is difficult to define. Conversely, in many instances a large amount of experimental data is widely available and easy to obtain. These characteristics make image analysis problems appropriate candidates for machine learning and evolutionary approaches such as GP [23, 7]. For instance, GP has been used for image classification [21, 47], object detection and recognition [14, 9, 13], feature synthesis [22, 41], image segmentation [39, 45], and local image description [37, 38]. 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 [51, 52] and a multi-objective optimization problem [54, 36] and solved with GP. Another example can be found in [56], where GP was used to detect edge points, a fundamental problem for many computer vision systems. In those works, the operators produced by GP compute a measure of *edgeness* or *saliency*, which is a similar proposi-

tion 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 [49], we focused on the same high-level goal of evolving estimators of Hölder regularity. However, in [49] the ground truth for the learning process utilized images of multifractional 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 a 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 applied to the problem of local image description and matching [53, 50].

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* [19]. In classical regression an optimization algorithm must find the optimal values for a set of coefficients in 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 [19, 16, 11, 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 [17]. Recently, other works have shown that GP can be used in other domains of mathematics. For instance, [4] use GP to construct approximate solutions for complex differential equations. In [46] GP is used to study special elements of finite algebras, and in [5] to design quantum circuits. In summary, this paper is another concrete example of how GP can solve complex mathematical problems from applied domains.

3. Theoretical background

The aim of this section is to present a concise introduction to the concept of Hölder regularity and the GP paradigm, the two topics that intersect in this paper. However, some details are omitted for brevity, but the interested reader should refer to [48] and [40] for specific details regarding Hölder regularity and GP respectively.

3.1. Hölder regularity

The concept of Hölder regularity allows us to characterize the singular structures contained within a signal [28]. 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 is applicable to the general case of $\mathbb{R}^n$ signals.
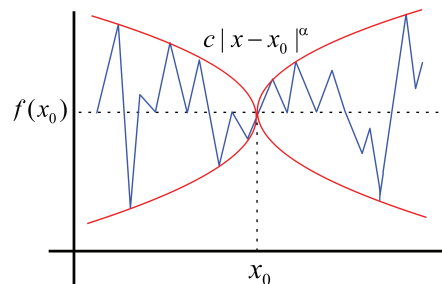

Figure 1: Hölder envelope of a non-differentiable signal f at point x_0. This representation is for a 1D signal or a cross-section of a signal in 2D.

3.1.1. The pointwise Hölder exponent

Here, we are interested in measuring the pointwise Hölder exponent which is defined as follows for a 2D signal.

Definition 1: Let $f : \mathbb{R}^2 \to \mathbb{R}$, $s \in \mathbb{R}^{+*} \setminus \mathbb{N}$ 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_n of degree $n < s$ and a constant c such that

$$\forall x \in B(x_0, \eta), |f(x) - P_n(x - x_0)| \leq c|x - x_0|^s ,\tag{1}$$

where $B(x_0, \eta)$ is the ball around x_0 with a radius η. The pointwise Hölder exponent of f at x_0 is $\alpha_p(x_0) = \sup_s \{f \in C^s(x_0)\}$.
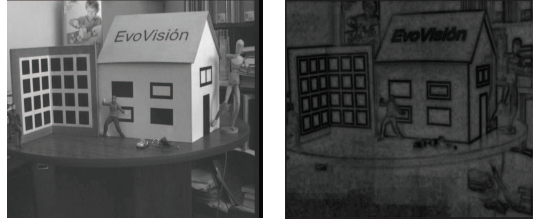
In the above definition, P_n represents the Taylor series expansion of function f. Equation 1 describes a bound on the amount by which a signal varies, or oscillates, around point x_0 within an arbitrary local neighborhood $B(x_0, \eta)$. Hence, when the singularity is large at x_0, with a large variation of the signal, then $\alpha_p \to 0$ as $x \to x_0$. Conversely, $\alpha_p \to 1$ when the variation of the signal $(f(x) - P_n(x - x_0)) \to 0$ as $x \to x_0$, thus the signal is smoother, or more regular, at x_0. Figure 1 shows the envelope that bounds the oscillations of f expressed by the Hölder exponent α_p at x_0. In summary, we can say that the Hölder exponents refines the concept of the Taylor series approximation of a function, by also characterizing the non-differentiable points [32].

3.1.2. Estimation of the pointwise Hölder exponent

As stated above, several estimators for the Hölder exponent have been developed. Probably the better known estimators are wavelet-based methods, such as the wavelet coefficients regression and the wavelet leaders regression. These estimators employ a wavelet decomposition of the analyzed signal and produce a reliable estimation when the wavelets satisfy some specified regularity properties [15].

Another approach is based on modeling an image using Choquet capacities [29]. A Choquet capacity can be understood as a measure which does not need to satisfy the additivity requirement. In this method, the multifractal spectrum of a sequence of Choquet capacities is used to estimate the pointwise regularity. This approach has been successfully applied to edge detection [29].

Another example is [2], which presents estimators for the generalized multifractional Brownian motion, signals where the Hölder exponent can vary from point to

(a) Original Image   (b) Hölder Image

Figure 2: Hölder exponent estimation using oscillation method.

point in an erratic manner. These signals provide realistic models of real-world phenomena. The method is based on generalized quadratic variations (GQV), an estimator for which the Central Limit Theorem holds under certain conditions.

Finally, probably the most direct estimator is based on analyzing local signal oscillations, a simple and direct process. This estimator has achieved good result in real-world applications [27, 26, 53] and has proven to be superior, in some tests, to the wavelets-leaders method [25]. Therefore, we have chosen the oscillation method as the baseline reference method, described in greater detail next.

3.1.3. Oscillation based method

The most direct estimator of the Hölder exponent consists on analyzing the oscillations of a signal around each point. This method is derived directly from Definition 1 as follows [48]. The Hölder exponent for a non-differentiable function $f(t)$ at t is the $sup(\alpha_p) \in [0,1]$, for which a constant c exists such that $\forall\, t'$ in a neighborhood of t,

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

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

$$osc_\tau(t) = \sup_{t',t'' \in B(t,\tau)} |f(t') - f(t'')| . \tag{3}$$

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

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

8

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

Hence, 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 τ at which the oscillations 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 point x_0 only on intervals of the form $B(x_0, \tau)$. 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, the corresponding Hölder image.

The above estimation method depends on several parametric choices. In this paper, we use the same values published in [49], 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.

3.1.4. 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 (oscillations), spatial-frequency decomposition (wavelet-based methods), using Choquet capacities, and modeling the signal as a type of Brownian motion (GQV). 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

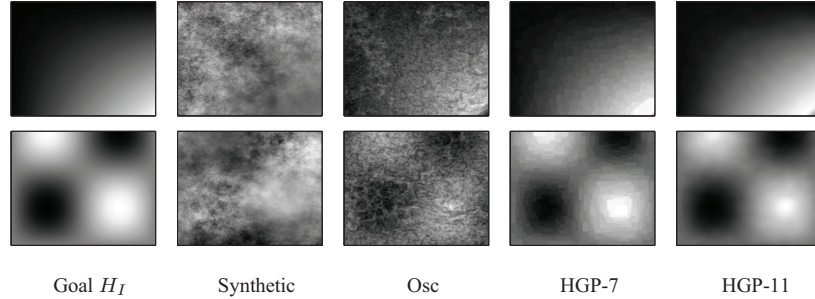Goal H_I Synthetic   Osc   HGP-7   HGP-11

Figure 3: Performance of evolved estimators compared with the oscillations method on synthetic images of multifractional Brownian motions, from [49]. 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.

Image   Oscillations   HGP-7   HGP-11

Figure 4: Performance of evolved estimators compared with the oscillations method on a real image, from [49]. Notice how the evolved estimators produce overly smooth estimations, they are overfitted to the type of regularity used during training, see Figure 3.

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 evolution [8]. 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. Of current algorithms, GP is one of the most advanced forms of evolutionary search [19]. 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), where the main difference lies with the manner in which individuals are coded, see [8]. While a GA uses bit strings or parameter vectors with a constant and uniform length, GP uses tree structures that can be of different sizes within the population. As a result, another important difference is the manner in which new program trees 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 even higher fitness. The second operator is mutation, where a randomly chosen subtree is deleted and substituted by a new subtree that was also randomly generated. The main roles of mutation are to introduce diversity into the population and possibly
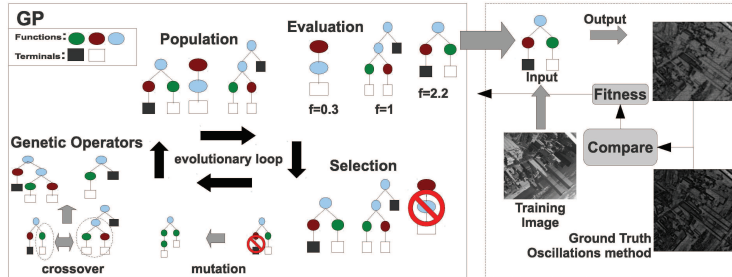
11

Figure 5: The GP algorithm used to evolve estimators for the pointwise Hölder exponent.

provide a slight improvement to program trees that already exhibit good performance.

4. Hölder exponent estimation as a GP optimization problem

In this work, the proposal is to evolve image operators that estimate the pointwise Hölder exponent. In order to do so, we pose the following 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,

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

where $Err[,]$ represents an error measure. In a previous work [49], 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 Figure 3. However, the evolved estimators suffer from two significant limitations that prohibit their use in real-world problems. 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 Figure 4, the estimation obtained by the evolved operators is smoother then in ought to be considering

12

the many 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 in the training set. Second, in [49] the evolved estimators substantially outperform the oscillations method based on run-time. However, they cannot be used for real-time applications they require between one and twelve seconds to process a 512×512 image.

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 main assumptions. First, since we already know that GP can produce better estimators that the oscillations method [49] (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, in [49] we also 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, such as local image description [53, 50], then evolved estimators that approximate the oscillations method should exhibit a similar performance on the same problem.

It is important to point out that all three of the above a priori assumptions are experimentally tested and validated in the experimental work below, see 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 [49]. However, the results obtained suggest that indeed the evolved estimators generalize quite well.
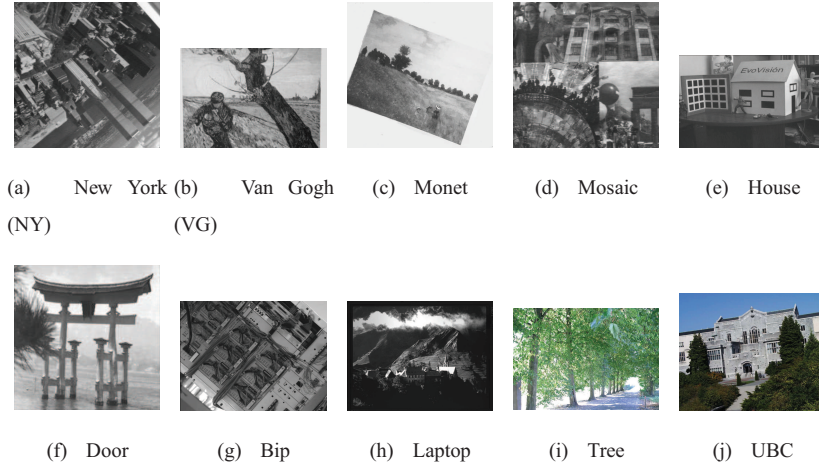
13

Figure 6: Some of the images used for training and testing.

4.2. Genetic programming approach

The optimization problem defined in Equation 5 can be solved using a GP search. For this task, we have chosen to use a canonical tree-based GP with Koza style genetic operators [19, 24] and a bloat control mechanism to limit the size and complexity of the individual trees [44]. A general overview of the GP approach is illustrated in Figure 5, where the basic evolutionary loop is depicted with its three main stages: fitness evaluation, fitness-based selection, and the production of new program trees by crossover and mutation.

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 Figure 5. Here, the error measure from Equation 5 is defined as the root mean square error (RMSE),

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

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

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

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

$$\widehat{K}(x_{i,j}) = S \cdot \frac{K(x_{i,j})}{\sqrt{\displaystyle\sum_{i=1}^{N} K(x_{i,j})_i^2}} \; , \tag{8}$$

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

Then, we must define a set of M training images, and here we use a set with four images of different scenes ($M = 4$). The name of the images used for training are New York, Van Gogh, Monet and Mosaic. Figure 6 shows 10 images, the first four are used for training while the others are part of the larger testing set that contains thirty different images [1].

4.2.2. Search space

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

$$\begin{aligned}
F_{point} &= \left\{ +, |+|, -, |-|, |I_{out}|, *, \div, I_{out}^2, \right\} \\
&\bigcup \left\{ \sqrt{I_{out}}, log_2(I_{out}), k \cdot I_{out} \right\} \; , \\
F_{neighborhood} &= \{ G_1, G_2 \} \; , \\
F = F_{point} &\bigcup F_{neighborhood} \; , \quad T = \{ I \} \; ,
\end{aligned} \tag{9}$$

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

---

[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/ .

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 6 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.

term, with another. For instance, in [12] two terms are combined to obtain a measure of saliency for image pixels, and one of the terms is scaled by a similar factor. Also, this function proved to be useful in the evolutionary search for interest point detectors [51, 52, 54].

The Functions 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. On the other hand, the functions in $F_{neighborhood}$ are filters that use a convolution mask and operates 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, as suggested in [40].

5. Experimental results

This Section provides a detailed description of the GP system, the experimental results, and comparisons with traditional estimators.
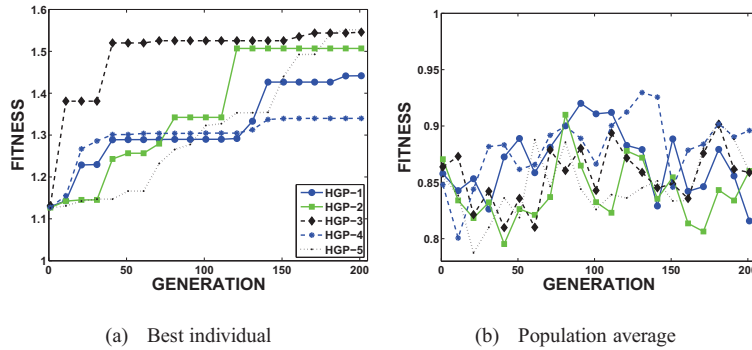
16

(a) Best individual	(b) Population average

Figure 7: Evolution statistics for the five best GP runs.

5.1. Implementation details

The GP system uses the parameters presented in Table 1, which were tuned empirically for best performance. The GP algorithm was programmed using the Matlab toolbox GPLAB [43], and estimation of the Hölder exponent with the oscillations method was done using the FracLab toolbox [30].

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 of the GP. The best estimator produced in each run will 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 Figure 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. These mathematical expressions have been algebraically simplified for easier understanding.

17

New York — OSC, HGP-2, HGP-3

Van Gogh — OSC, HGP-2, HGP-3

Door — OSC, HGP-1, HGP-2

Lap — OSC, HGP-2, HGP-5

UBC — OSC, HGP-2, HGP-3

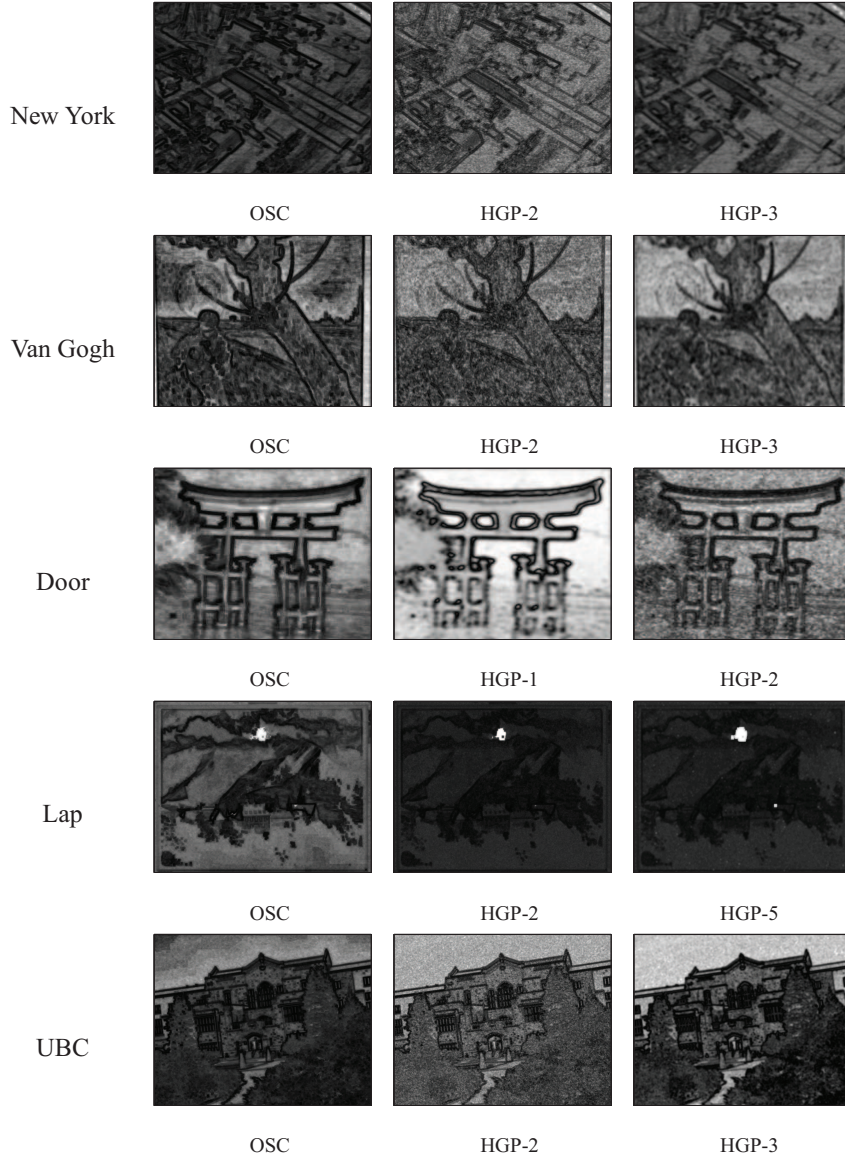Figure 8: Qualitative comparisons between the evolved estimators and the oscillations method. First column contains the Hölder image computed with the oscillations method, and each row corresponds to a different test image.

5.3. Comparisons

In what follows, we present a comparison between the estimation of the Hölder exponent produced by the evolved operators, and the estimation obtained with the os-

18

Table 2: Symbolic expression for evolved estimators.

HGP-1:

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

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:

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

HGP-5:

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

cillations method. Moreover, we also present a run-time comparison with other state-of-the-art approaches.

5.3.1. Quantitative comparisons

First, in Table 3 we present a quantitative comparison using the images in Figure 6. 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, and the table also shows the size of each image in pixels.

Then, in Table 4 we use an extended test set of 30 different images for a more

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

reliable comparison. These images were selected to be as diverse as possible, 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 shows that HGP-3 and HGP-2 consistently achieve the best correlation and the lowest RMSE. Moreover, we can see that 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

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

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**	**Choquet**	**GQV**
Time		360	66.30	0.30	31.5

image patterns.

5.3.2. Qualitative comparisons

In Figures 8 we present 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 methods is very high. Indeed, GP produces a very good approximation of the ground truth Hölder exponent.

5.3.3. Runtime comparisons

Finally, we perform a comparisons of the runtime required to execute each of the evolved estimators in seconds, the results are summarized in Table 5. For these tests we use a PC Laptop with a 64 bit AMD processor and 1GB 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 only freely available

software for regularity analysis [30]. The estimators from FracLab are the oscillations method described above, the wavelet leaders method [15], the estimator based on Choquet capacities [29], and the GQV method [2]. All of the FracLab estimators use optimized code and include fast C implementations of some crucial parts of the algorithms, this gives them an advantage over the evolved 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 GQV are slower by two orders of magnitude. The fastest algorithm in FracLab is the Choquet estimator, which uses a highly 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 FireWire camera. We implemented operator HGP-2 using the C++ vision library LibCVD[2], which allowed us to obtain a frame-rate of 30 fps[3] with a video resolution of 640×480; sample frames are shown in Figure 9.

In summary, this Section presents an extensive comparison between the evolved estimators and more traditional methods. Indeed, we have shown that the GP approach can produce very good estimators of Hölderian regularity which closely approximates 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

---

[2]http://www.edwardrosten.com/cvd/index.html

[3]This is the maximum frame rate of the Fire-i camera, but the estimator could process information faster if required.

Input Video

Hölder

Figure 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.

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, local region description and matching.

Recently, many computer vision systems are based on the detection and description of local and sparse image features. The approach was introduced in [42, 31] and consists on the following basic steps. First, small image regions centered around salient pixels, better known as *interest points*, are detected using specially designed image operators [55]. 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* [35]. The set of local regions and their corresponding descriptors are then used to construct models of the objects, or scene, present within the image. When a new image is analyzed this process is repeated and the extracted features are compared with the stored models. Therefore, to perform a recognition task, the vision system searches for local correspondences such as the one depicted in Figure 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 using compact descriptors.

Keeping to the problem of local description, many proposals have been made over the last fifteen years. However, currently the Scale Invariant Feature Transform (SIFT) [31] is still widely accepted as the standard method in current literature [35]. The overall success of SIFT has led some researcher to develop improved versions of the algorithm, one related example is the GP-optimized SIFT [38]. Another descriptor was proposed in [53], based on sampling the pointwise Hölder exponent within a local region. The *Hölder descriptor* achieves comparable performance to SIFT on standard tests, see [53]. 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 many real-time tasks [4]. 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 genetic algorithm [50]. In this work, however, we employ the evolved HGP estimators to construct the local Hölder descriptor and compare the performance with the original descriptor that uses the oscillations method. Because we already know that HGP estimators are substantially faster, if they achieve a similar performance on benchmark tests then we can have a fast algorithm for local image description.

6.1. Hölder descriptor

The process used to build the local descriptor using Hölderian regularity proceeds as follows. First, a set of interest regions are extracted from an image. Second, the dominant gradient orientation within each region is computed, thus preserving rotation invariance. Finally, the descriptor vector contains a sampling of the Hölder exponent

---

[4]Many works deal with this topic, and faster implementations have been proposed such as SURF and GPU-SIFT
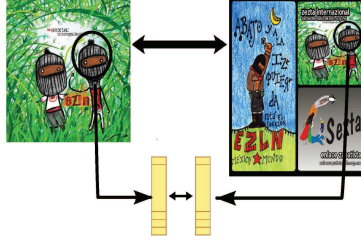
Figure 10: The matching process using locally salient features. A local descriptor is computed for an interest region detected on the left image. Then, a correspondence is sought between it and the local descriptors extracted from the image on the right. The figure shows how a correct match between descriptive vectors can assure a correct match between corresponding regions.

on 129 concentric points using a polar grid and ordering them based on the dominant orientation within the region. A detailed description of each step is given next.

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 $IPGP2$ detector which is the determinant of the Hessian matrix smoothed by a 2D Gaussian kernel [51, 52]. 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 of [34], 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, and the size of the image region used to compute the local descriptor 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 a size 61×61 pixels using bicubic interpolation.

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 [31]. The histogram peak is obtained $\forall \lambda \in \Lambda$ and a corresponding dominant orientation ϕ_λ is assigned. In this way, each region is described by a 4-tuple $\lambda =$
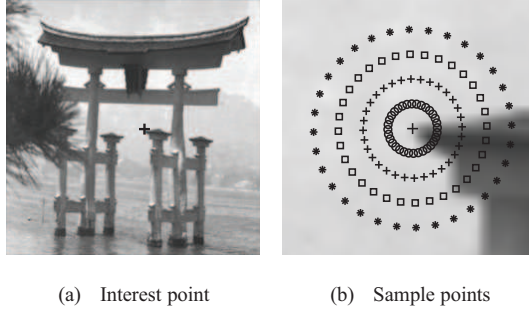
(a) Interest point   (b) Sample points

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

$(x_\lambda, y_\lambda, s_\lambda, \phi_\lambda)$, where each element respectively defines the region center, scale, and dominant orientation.

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, depicted in Figure 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, we use standard image sequences provided by the Visual Geometry Group and the Lear team at INRIA. Detailed information regarding the image sequences can be obtained from each groups website or in [35]. Table 6 gives further details regarding the images used in the experimental tests. 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

26

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

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 Figure 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 [35].

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 $d(,)$ 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 help characterize the matching process between two images [35].

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 represented by *1-Precision*. A perfect descriptor would give a *Recall* equal to 1 for any *1-Precision*. *Recall* and *1-Precision* are defined as

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

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

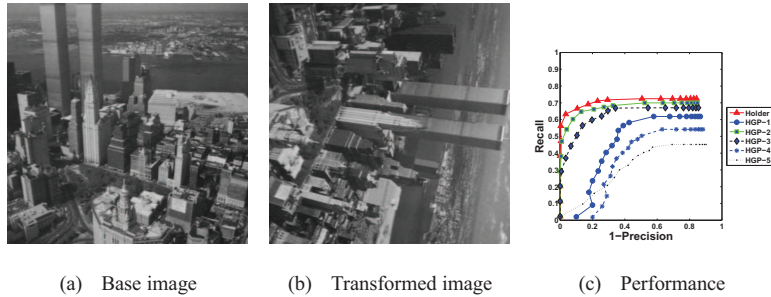| (a) Base image | (b) Transformed image | (c) Performance |

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

6.3. Results

We organize the test results 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, Figure 12 for the NY pair of images, Figure 13 for the VG pair, Figure 14 for the Mars pair, and Figure 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 transformations with illumination change, Figure 16 for the Graph images and Figure 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 following notable observations. First, only HGP-1 and HGP-4 produce notably inferior matching scores. Second, HGP-2 actually outperforms the original descriptor in the Graph test.
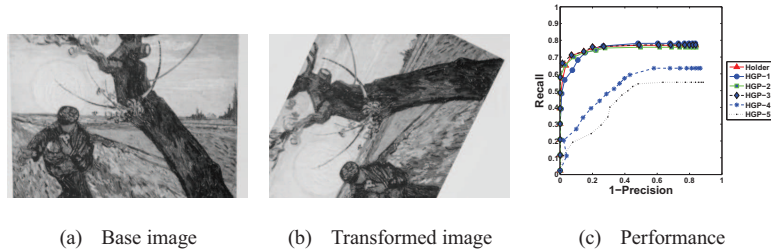
28

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

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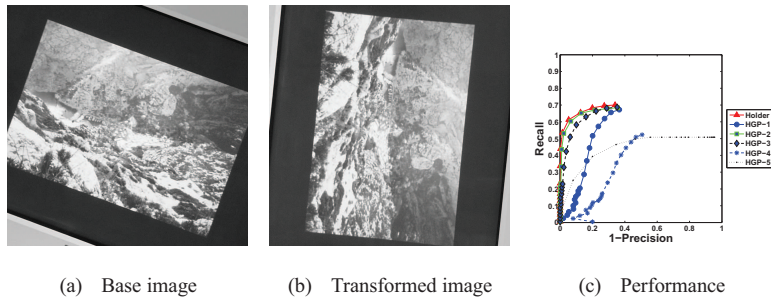
Figure 14: *Recall/1-Precision* curves for the Mars image pair. The transformation between (a) and (b) is a rotation.

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 Figure 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 with this quite well. Second, the performance of most image descriptors is normally degraded by JPEG compression, even for the SIFT method [35]. However, using the HGP estimator the Hölder descriptor can achieve remarkably good results. Therefore, the evolved estimators are able to produce an image descriptor that is invariant to JPEG compression.

29

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

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

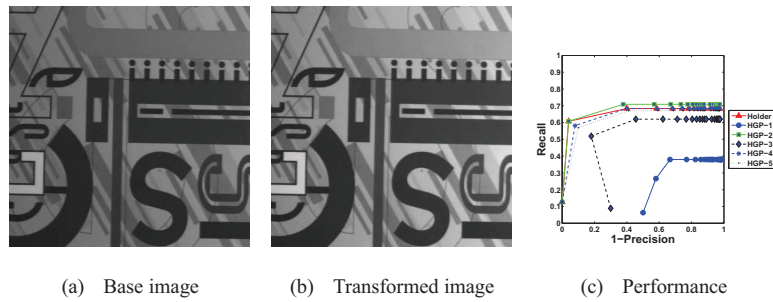| (a)   Base image | (b)   Transformed image | (c)   Performance |

Figure 16: *Recall/1-Precision* curves for the Graph 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 Figure 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 [53]. In particular, estimators HGP-2 and HGP-3 obtain a high performance across all tests, and in some instances are better than the 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
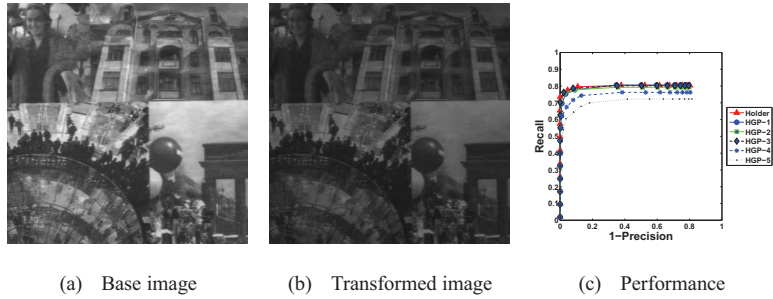
30

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

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

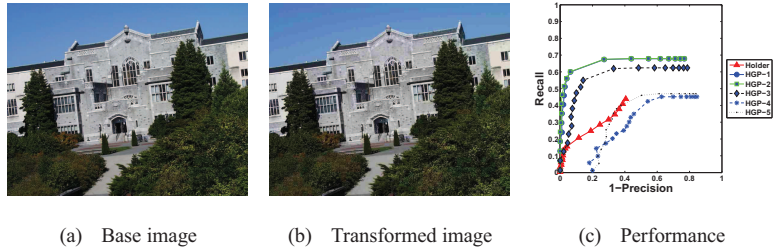|		
(a) Base image	(b) Transformed image	(c) Performance

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

small fraction of the time that more traditional methods require, such as SIFT. This could allow us to develop faster algorithms for real-time computer vision systems.

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.

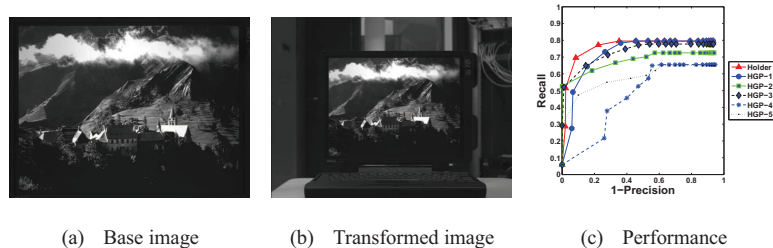(a) Base image   (b) Transformed image   (c) Performance

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

To test the estimators in a real-world task, we applied them 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 indeed capable of extracting a meaningful descriptive characterization of local shape and appearance using faster and simpler estimation methods. 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 [33]. 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.

On the other hand, 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

32

descriptors [35]. One exception is the Hölder descriptor by [53], 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.

References

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

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

[3] Azad, R. M. A., Ryan, C., 2010. 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, pp. 893–900.

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

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

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

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

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

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

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

[11] Gustafson, S., Burke, E. K., Krasnogor, N., 2005. 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, pp. 912–919.

[12] Harris, C., Stephens, M., 1988. A combined corner and edge detector. In: Proceedings from the Fourth Alvey Vision Conference. Vol. 15. pp. 147–151.

[13] Hernández, B., Olague, G., Hammoud, R., Trujillo, L., Romero, E., 2007. 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), 258–269.

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

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

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

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

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

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

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

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

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

[23] Krawiec, K., Howard, D., Zhang, M., 2007. 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, pp. 779–784.

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

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

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

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

[28] Lévy Véhel, J., 1996. Fractal Approaches in Signal Processing. In: Evertsz, C., Peitgen, H.-O., Voss, R. (Eds.), Fractal Geometry and Analysis : the Mandelbrot festschrift. World Scientific.

[29] Lévy-Véhel, J., 1998. Fractal Image Encoding and Analysis. Ch. Introduction to the Multifractal Analysis of Images, pp. 299–341.

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

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

[32] Mallat, S., 1999. A wavelet tour of signal processing, 2nd Edition. Elsevier, San Diego, CA.

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

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

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

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

[37] Pérez, C. B., Olague, G., 2008. 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, pp. 1–4.

[38] Perez, C. B., Olague, G., 2009. 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, pp. 1051–1058.

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

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

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

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

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

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

[45] Song, A., Ciesielski, V., 2008. Texture segmentation by genetic programming. Evol. Comput. 16 (4), 461–481.

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

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

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

[49] Trujillo, L., Legrand, P., Lévy-Véhel, J., 2010. 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, pp. 861–868.

[50] Trujillo, L., Legrand, P., Olague, G., Pérez, C., 2010. 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, pp. 1147–1154.

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

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

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

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

[55] Tuytelaars, T., Mikolajczyk, K., 2008. Local invariant feature detectors: a survey. Found. Trends Comput. Graph. Vis. 3 (3), 177–280.

[56] Zhang, Y., Rockett, P. I., 2005. 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, pp. 795–802.