
Volume 29, issue 7, June 2011
ISSN 0262-8856
Contains Editor's Choice Article
ELSEVIER
image
AND
vision
COMPUTING
Available online at www.sciencedirect.com
ScienceDirect

Image and Vision Computing 29 (2011) 484–498

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Evolutionary-computer-assisted design of image operators that detect interest points using genetic programming ☆

Gustavo Olague [a,*], Leonardo Trujillo [a,b]

[a] Proyecto Evovisión, Departamento de Ciencias de la Computación, División de Física Aplicada, Centro de Investigación Científica y de Educación Superior de Ensenada, Km. 107 Carretera Tijuana-Ensenada, 22860, Ensenada, BC, Mexico

[b] Instituto Tecnológico de Tijuana, Av. Tecnológico S/N, Fracc. Tomás Aquino, Tijuana, B.C., Mexico

ARTICLE INFO

Article history:
Received 3 May 2010
Received in revised form 29 January 2011
Accepted 22 March 2011

Keyword:
Interest points
Computer assisted design
Evolutionary computation
Genetic programming
Evolutionary computer vision

ABSTRACT

This work describes a way of designing interest point detectors using an evolutionary-computer-assisted design approach. Nowadays, feature extraction is performed through the paradigm of interest point detection due to its simplicity and robustness for practical applications such as: image matching and view-based object recognition. Genetic programming is used as the core functionality of the proposed human-computer framework that significantly augments the scope of interest point design through a computer assisted learning process. Indeed, genetic programming has produced numerous interest point operators, many with unique or unorthodox designs. The analysis of those best detectors gives us an advantage to achieve a new level of creative design that improves the perspective for human-machine innovation. In particular, we present two novel interest point detectors produced through the analysis of multiple solutions that were obtained through single and multi-objective searches. Experimental results using a well-known testbed are provided to illustrate the performance of the operators and hence the effectiveness of the proposal.

1. Introduction

Computer vision (CV) is concerned with the development of artificial systems that can automatically analyze and interpret visual information, and current systems have obtained impressive performance on many high-level tasks that include object recognition [15], object detection [2], image classification [7], image retrieval [36], object categorization [12], and 3D reconstruction [17,57]. Nevertheless, the answer to the conundrum of artificial vision from the point of view of machine intelligence presents fundamental problems with underlying difficulties that attract continuously the interest of researchers from diverse fields of research, such as pattern recognition, artificial intelligence, and cognitive science, to name but a few. In particular, researchers from evolutionary computation are working actively in numerous theoretical and practical CV problems, see [3,4,39].

In this way, for many domains of technological and scientific endeavor, solutions to particular problems are traditionally the outcome of a detailed design process undertaken by a group of human experts, and in this respect CV is not an exception. Problem solving requires that a scientist or engineer makes a series of design choices in order to produce a final solution [42]. Therefore, the attributes of a particular solution will depend upon the initial assumptions that are made, and on the overall understanding that the human expert possesses regarding the nature of the problem domain. One shortcoming for this approach to problem solving is that sometimes, when a different type of solution is desired, or required, then the design process must be changed and executed once more, using a different set of assumptions and analytical perspectives. As a result, a large number of competing proposals can exist for what appear to be very basic and simple problems; for example, the problem of interest point detection [64]. The design of candidate solutions can also be understood as an informed search process. For instance, in the scenario described above the search is guided by human expertise and operates within a domain-specific space of possible solutions. When the domain of a problem is well-known, then some useful properties of the search space could conceivably be inferred in order to improve the search process. However, even if this is the case, and for difficult problems it cannot be assumed, the space is normally very large, complex and non-linear [42].

In this work, we follow the genetic programming framework to extend the traditional approach for problem solving just described. Indeed, from all evolutionary-based methodologies inspired by biological evolution genetic programming provides a framework to find computer programs that perform a user-defined task. This technique is a powerful machine learning approach that is still largely unknown in the computer vision literature. This paper takes a further step in the traditional way of designing CV programs using what we call an evolutionary computer assisted design (E-CAD) concept. The

☆ This paper has been recommended for acceptance by Maja Pantic, PhD.

* Corresponding author. Tel.: +526461750500 x 23429.
E-mail addresses: olague@cicese.mx (G. Olague), leonardo.trujillo.ttl@gmail.com (L. Trujillo).

0262-8856/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2011.03.004

idea is to explore the design space of a very specific and useful task found in many machine vision systems, such as the interest point detection, using the genetic programming technique. Later, when numerous design solutions are available the decision maker could create novel human designs from the set of human-competitive machine intelligence solutions. Thus, the human designer is given the possibility to create new designs that are far away of the abilities of a designer using more traditional approaches. Indeed, the advantage is clear because as we will show in the experiments the designer starts a new design stage from a whole set of competitive designs; thus, closing the human-machine invention cycle within a repeating process that can be easily reconfigured to produce new suitable designs according to the requirements of the task. This is possible because a priori knowledge is easily incorporated within the genetic programming framework that uses an adaptive learning paradigm to approach the curse of dimensionality.

The goal of finding the optimal solution for real-world problems is usually a quite arduous task; in particular, our work follows the approach of artificial evolution. Thus, one of the main keys to set up such a proposal is to begin with a clearly specified problem where it is often much easier to propose a performance or evaluation criteria that could help to define a measure of optimality. In this case, the search could then be carried out automatically by a computer algorithm that exploits the information that such measures of performance can return. In this way, the search could be carried out using evolutionary computation (EC), a population based meta-heuristic based on the principles of artificial Darwinism that have shown great success at exploring large search spaces; thus, producing solutions that are well adapted to the prescribed objectives [9,19,24]. Indeed, EC is based on the core principles of biological evolution, a natural process that exhibits an adaptive power that by far outstrips that of any human-engineered system [54]. Currently, a large amount of experimental evidence exists that confirms the ability of EC to outperform man-made solutions in many domains, such as antenna design, mathematical proofs, and even CV [4,25,26,44,56,62]. In many cases, the stochastic nature of EC allows it to sample large portions of the search space, and sometimes produce solutions that might not be evident to a human expert. For example, the case of network design in photogrammetry in which the design of a specific network, considered as not atypical, was rediscovered by means of evolutionary computing [38,41]. Moreover, in our previous work on the design of interest point detectors genetic programming was able to rediscover Beaudet's detector [60]. However, we do not suggest that the use of machine learning in general, or EC in particular, should completely substitute the design work that a human expert could perform. On the contrary, we agree with the argument that a more complete strategy would cooperatively include both methodologies, thereby blending the complimentary skills of each [55], in what others have called a computer assisted design (CAD) process [34,42].

In this paper, we employ an E-CAD based approach in the search for optimal image operators that detect low-level features known as interest points [64]. We use genetic programming, one of the more advanced forms of EC, to automatically synthesize candidate solutions that can be represented using tree structures. The evolutionary search is guided by two performance criteria, the geometric and photometric stability of detected points given by the repeatability rate [50], and a measure of how disperse the set of detected points are over the image plane [64]. In order to achieve a design these objectives are concurrently considered using two different techniques: first, both criteria are included into a single objective function and the search returns the best single solution found; and second, we pose a multi-objective problem that searches for a diverse set of Pareto optimal solutions. In each case, we use E-CAD to propose novel interest point detectors using the operators that the evolutionary algorithm generates. The first one is characterized by its simplicity and the high performance it achieves on standard tests; we call it the *Gaussian Intensity Neighborhood* (GIN) interest point detector. On the other hand, the other is a parameterized operator for interest point detection that allows for fine grained control of the amount of point dispersion without sacrificing the geometric stability; we call it the *Multi-Objective Parameterized* (MOP) interest point detector, and to our knowledge it is unique in CV literature.

The remainder of this paper is organized as follows. Section 2 presents the basic problem of interest point detection, reviews previous work and defines the performance criteria. Genetic programming is introduced in Section 3, and a review of applications to CV is outlined. Section 4 describes the single objective approach to evolutionary-computer-assisted design of interest point operators and introduces the Gaussian Intensity Neighborhood detector. Then, the multi-objective approach is presented in Section 5 and the Multi-Objective Parameterized interest point detector is explained. Finally, Section 7 contains a brief summary and outlines possible lines of future research.

2. Interest point detection

Currently, many CV systems employ a local approach to feature extraction and description, by focusing on small and highly invariant features called interest points [31,33,49,50,64]. It is also important to understand that the performance of these systems directly depends on the quality of the underlying detection and description algorithms that are used. Keeping to the former, there are dozens of proposed interest point detectors available in CV literature, most of which are the direct product of a human-based approach to problem solving and/or design.

Using the taxonomy of local features given in [50,64], we can say that interest points are detected by algorithms that focus on image intensity values and only make weak assumptions regarding the underlying structure of the observed scene [64]. Interest points are salient image pixels that are unique and distinctive; i.e., they are quantitatively and qualitatively different from other points, and they normally represent only a small fraction of the total image area [32,50].

2.1. Problem definition

A measure of how salient or interesting each pixel is can be obtained using a mapping of the form $K(x):\mathbb{R}^+\to\mathbb{R}$ which we call an interest point operator. Each interest point detector will employ a different operator K; in this way, a *detector* refers to the complete algorithmic process that extracts interest points, while an *operator* only computes the corresponding interest measure. Applying K to an image I produces what can be called an *interest image* I^*, see Fig. 1. Afterwards, most detectors follow the same basic process: non-maxima suppression that eliminates pixels that are not local maxima, and a thresholding step that obtains the final set of points. Therefore, a pixel $\mathbf{x}$ is tagged as an interest point if the following conditions hold,

$$K(\mathbf{x}) > max\{K(\mathbf{x_W})\,|\,\forall \mathbf{x_W}\in\mathbf{W}, \mathbf{x_W}\neq\mathbf{x}\}\wedge K(\mathbf{x}) > h, \tag{1}$$

where $\mathbf{W}$ is a square neighborhood of size $n\times n$ around $\mathbf{x}$, and h is an empirically defined threshold. The first condition in Eq. (1) accounts for non-maximum suppression and the second is the thresholding step, the process is shown in Fig. 1. Experiments in the current work use $n=5$, while h depends on the operator.

2.2. Previous proposals

In this section only a brief overview of previous work is given, a thorough discussion is beyond the scope of this paper. For example, there exists a group of interest point detectors that employ operators that are based on the *auto-correlation* or *second-moment matrix*. This

Fig. 1. A look at interest point detection: left, an input image I; middle, *interest image* I^*; right, detected points after non-maximum suppression and thresholding superimposed on I.

mathematical concept captures local properties of the image gradient around each point. For instance, previous detectors have used the determinant and trace of this matrix [13,14,16], as well as an eigenvalue analysis [51].

Other operators employ measures pertaining to the local curvature around each point. For instance, [1] used the determinant of the Hessian matrix as the interest operator, and in [23] an operator is proposed that uses the gradient magnitude and the intensity with which the gradient changes direction. These operators, and others, have been shown to be mathematically equivalent in [37].

Other examples include operators that exploit color information, temporal information, and simple intensity variations around each point. For a complete discussion and comparison of these and other such methods, we refer the interested reader to [40,50,62,64].

2.3. Performance criteria

In the CV community a consensus about the evaluation of interest point detectors is that probably the best approach to measure the performance of each detector is to use experimental criteria and a set of easily reproducible tests [35,50,58]. Other methods have been proposed such as using an analysis based on mathematical axioms [22] without success. The idea of using genetic programming is congruent with the more useful approach of experimental evaluation of interest points since such an evolutionary algorithm attempts to automate the trial and error process; thus, providing an avenue towards creative designs.

2.3.1. The repeatability rate

Currently, the most common measure for evaluating interest points is the repeatability rate, which quantifies how invariant is the detection process with respect to basic image transformations. In this way, the repeatability rate is computed between two images taken from different viewpoints; thus, one is the base image and the other is its corresponding transformed counterpart. Therefore, after detecting interest points on the base image it is possible to calculate if the same scene features are retroprojected as interest points on the transformed image. Moreover, the repeatability rate is represented as a well-behaved function that varies from 0 to 100, where lower values indicate that detection is unstable with respect to the mapping and higher values indicate a more stable detection process. Therefore, if two viewpoints are related by a planar homography; thus, computing a repeatability score is a straightforward task [50]. Hence, the repeatability rate is given by

$$r_{I_i}(\epsilon) = \frac{\left|R_{I_i}(\epsilon)\right|}{min(\gamma_1, \gamma_i)}, \tag{2}$$

where R_{I_i} is the set of point pairs that lie in the common part of both images and correspond within an error ϵ, and $\gamma_1 = |\{x_1^c\}|$ and $\gamma_i = |\{x_i^c\}|$ are the total number of points extracted from the base image I_1 and the transformed image I_i.

2.3.2. Point dispersion

Another performance criterion is to consider the amount of dispersion that the interest points have over the image plane. Although these criteria will greatly depend upon the underlying structure of the imaged scene, some authors have stated that it is an important determining factor when choosing a method for point detection in specific problem domains [8,32,65].

A measure for point dispersion can be obtained by using the entropy computed from the partition $\{\mathcal{I}_j\}$ of the spatial dimensions of the image plane I. Hence, $\mathcal{D}$ is the entropy value of the spatial distribution of detected interest points X within the image and it is calculated as follows:

$$\mathcal{D}(I, X) = -\sum P_j \cdot log_2\left(P_j\right), \tag{3}$$

where P_j is approximated by the 2D histogram of the position of IPs within I. In this work, the image is divided into a 2D grid where each bin has a size of 8×8 pixels.

3. Genetic programming

The evolutionary computation (EC) paradigm consists of the development of computer algorithms that base their core functionality on the basic principles of Darwinian evolution. These techniques are population-based meta-heuristics, where candidate solutions are stochastically selected and modified in order to produce new solutions and thus to explore the search space of a particular problem. The selection process favors those individuals that exhibit the best performance, and the entire process is carried out iteratively until a predefined termination criterion is reached, such as a maximum number of iterations. The general strategy in artificial evolution involves the following steps:

1. An encoding scheme that allows an evolutionary algorithm to represent a set, or *population*, of problem solutions, where a single solution is normally referred to as an *individual*. In GP a tree-based structure is commonly applied.
2. An evaluation function f that quantifies the performance of each individual given the objectives of a specific problem, and assigns a *fitness* value accordingly.
3. A set of operations are applied to individuals in the population that are chosen with a probability based on fitness, which are thereafter used to create a new population of individuals that are tested in the following iteration.
4. A mechanism that produces new solutions through *variation* of the subset of selected individuals. This is accomplished with two methods: *Recombination* that allows two individuals to exchange

information and thus to create a new candidate solution; and *mutation* which slightly modifies the information contained within a single solution.

5. A stochastic *survival* strategy that decides which individuals will appear in the following iteration, or *generation*, of the algorithm.

Genetic programming (GP) is arguably the most advanced and complex technique used in EC, a generalization of the better-known and more widely used genetic algorithms (GAs) [9,24,29,45]. In canonical GP, each individual solution is represented through a parse tree because such structures can express simple computer programs, functions, or mathematical operators. For instance, trees are equivalent to the S-expressions used in the LISP language. Thus, individual trees are made up of internal and leaf nodes, which are defined by a set of primitive elements also called function set *F*, and terminal set *T*. These sets define the search space of possible solutions that the GP can produce; and even when a maximum depth or size limit for individual trees is enforced, normally the search space is very large but finite. Thus, the selection function is charged of choosing the best individuals for reproduction through mutation and recombination. An individual selected for mutation operation, randomly selects a node (mutation site), which is deleted to substitute this part of the tree with a new expression in order to obtain a new individual. On the other hand, the recombination (crossover) method needs two selected individuals called parents to perform its genetic operation; thus, one node of each parent is randomly selected as the crossover point; then, the subtrees are combined in order to create a new individual called child. In this work, selection is carried out using a tournament with lexicographic parsimony pressure, while keeping the best individual using as survival strategy the stochastic universal sampling method. The termination criteria were defined by a maximum number of generations; thus, the evolutionary algorithm reaches a local optimum operator for the case of the mono-objective test and a set of local Pareto fronts in the multi-objective case. Most parameters involved have canonical values and some others were set empirically after a number of tests. In accordance with the five main features of the evolutionary algorithms described above, Fig. 2 presents a basic GP algorithm with: a tree-based representation for individuals (1); a module that *evaluates* all the individuals in each population (2); a module that performs *Population management* (3 and 5); and a module that performs *Variation* of individual solutions (4) [68].

3.1. Computer vision applications

In recent years, GP has received a growing interest as a methodology for solving CV problems because of its ability to synthesize specialized image operators that can detect image features, or construct new features which can then be used in higher-level tasks [3,4,39]. It is possible to identify three types of GP-based approaches: (1) those that employ GP to detect low-level features which have been predefined by human experts, such as corners or edges [21,44,60–62,67] and recently one regarding vegetation indices used on remote sensing [46,47]; (2) those that construct novel low-level features which are specific to a particular problem domain, and do not need to be interpretable to a human expert [18,27,28,30]; and (3) those approaches that use GP to directly solve a high-level recognition problem [20,53,66]. For example, keeping to the latter two groups, works have addressed the problem of object detection [66], image classification [18,27], texture segmentation [53], and the analysis of synthetic aperture (SAR) images [20,28,30].

However, such approaches often produce solutions that can be unintuitive and in many cases lack a proper semantic interpretation. For these reasons, researchers in other fields might become tentative, or even skeptical, of solutions that are generated by GP. On the other hand, the works from the first group, those that attempt to detect features defined by human experts, by definition will not be hampered by the problem of semantic interpretation and here we can find examples also for object detection [43,44], and the analysis of multi-spectral images [47]. Furthermore, we believe that when appropriate fitness criteria are given, and when a comprehensive analysis of the obtained results is carried out, then it is possible to derive a better understanding of the logic behind the solutions that a GP produces, and also to obtain deeper insights regarding the nature of the problem itself.

4. E-CAD of interest operators using a single objective function

In this section we describe how we implemented a GP algorithm with a single objective method that automatically generates image operators for interest point detection. However, only a brief review is given, a complete description of the method can be read in our previous works on this subject [60–62].

In general, the goal is to design the GP candidate operators that maximize the overall stability of the detection process, as well as the amount of dispersion of the set of detected points. Hence, two

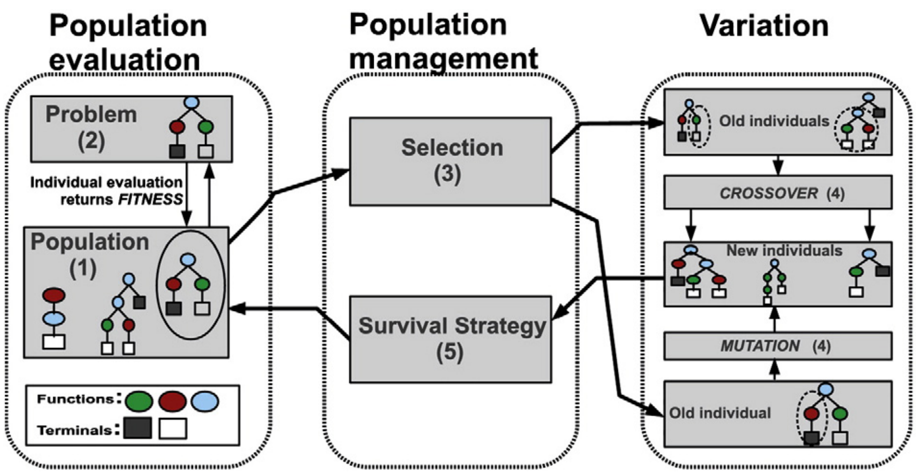

Fig. 2. A high-level view of the basic GP algorithm. It is possible to observe three main modules for: *Population evaluation*, *Population management*, and *Variation*.

important aspects must be described in detail: first, the search space (i.e., sets F and T); and second, the evaluation function that combines both objectives into a single measure.

4.1. Search space

In order to define an appropriate search space, the function and terminal sets contain operations that are widely used by previously proposed detectors. Hence, these sets are given by,

$$F = \left\{+, |+|, -, |-|, |I_{out}|, *, \div, I_{out}^2, \sqrt{I_{out}}, log_2(I_{out}), EQ(I_{out}), k\cdot I_{out}\right\}$$
$$\cup \left\{\frac{\delta}{\delta x}G_{\sigma_D}, \frac{\delta}{\delta y}G_{\sigma_D}, G_{\sigma=1}, G_{\sigma=2}\right\},$$
$$T = \left\{I, L_x, L_{xx}, L_{xy}, L_{yy}, L_y\right\}, \tag{4}$$

where I is the input image, and I_{out} can be any of the terminals in T, as well as the output of any of the functions in F; $EQ(I)$ is a histogram equalization operation; L_u are Gaussian image derivatives along direction u; G_σ are Gaussian smoothing filters; $\frac{\delta}{\delta u}G_{\sigma_D}$ represents the derivative of a Gaussian function[1]; and finally a scale factor $k = 0.05$. Note that the sets of functions and terminals cannot be considered as final. However, most interest point detectors can be evolved with these arguments. In previous work, we showed that evolved operators match previous detectors and hence that genetic programming was able to rediscover Beaudet's detector, DoG filter, and Laplacian without explicitly incorporating knowledge about the way of obtaining specific operators. However, the functions and terminals contain elements that artificial evolution could use to solve the problem. Harris is an example of an operator that was never found by the evolutionary process.

4.2. Evaluation function

As explained above, it is expected that the fitness function should promote the emergence of suitable operators that perform invariant detection of highly repeatable points, and that the set of detected points could be highly dispersed over the image plane. In this first part of the work, both objectives are combined in a multiplicative manner, as follows

$$f(K) = r_{K,J}(\epsilon)\cdot\phi_x^{\alpha}\cdot\phi_y^{\beta}\cdot N_{\%}^{\gamma}, \tag{5}$$

where $r_{K,J}(\epsilon)$ represents the average repeatability rate of an individual operator K computed from a set J of progressively transformed images with a localization error ϵ, and the terms ϕ_u promote a high point dispersion. The final term

$$N_{\%} = \frac{extracted\ \ points}{requested\ \ points}, \tag{6}$$

is a penalizing factor that reduces the fitness value for detectors that return less than the number of requested points. The terms ϕ_u behave like sigmoidal functions within a specified interval,

$$\phi_u = \begin{cases} \dfrac{1}{1+e^{-a(H_u - c)}}, & \text{when} \quad H_u < H_u^{max}, \\ 0 & \text{otherwise}. \end{cases} \tag{7}$$

Where H_u is the entropy value of the spatial distribution of detected interest points along direction u, on the reference image I_1 of the training set J, given by

$$H_u = -\sum P_j(u) log_2\left[P_j(u)\right], \tag{8}$$

with $P_j(\cdot)$ approximated by the histogram of interest point localizations. Values for $H_{u=\{x,y\}}^{max}$ are set empirically using the reference image of the training sequence, further details are provided in [62].

What is important to remark about the fitness function is that it primarily promotes a high repeatability score, and penalizes operators that obtain entropy values for point dispersion that lie outside the specified bound. The training sequence used is the Van Gogh set of progressively rotated images used as a reference test in [50,60,62]; all images are of size 348 × 512 pixels and samples are shown in Fig. 3. The training sequence has one base image and 16 progressively rotated images by 11.25° clockwise; however, only eight transformed images are used for fitness evaluation with a rotation angle of 22.5° between them.

4.3. Experimental results

This section gives an overview of 32 different executions of the GP algorithm, from which a total of 17 useful interest operators were obtained; see [62]. The reason for the discrepancy between the number of runs and the number of operators, is the fact that some runs produced the same operator (3 times), and in others the operator that was found did not achieve stable performance on the rest of the images that were not included during training (12 times).

4.3.1. Run-time parameters

The parameters of the GP algorithm are provided in Table 1 and were used in all of the experiments. The first five parameters were set empirically with canonical values. The next three help to limit the size of the evolved programs. *Tree depth* is dynamically set up using two maximum tree depths that limit the size of any given individual within the population. The *dynamic max depth* is a maximum tree depth that may not be surpassed by any individual unless its fitness matches or surpasses the fitness of the best individual found so far. When this happens, the *dynamic max depth* is augmented to the tree depth of the new fittest individual. Conversely, it is reduced if the new best individual has a lower tree depth. The *real max depth* parameter is a hard limit that no individual may surpass under any circumstance. Finally, selection is carried out using a tournament with lexicographic parsimony pressure. The implementation of the previously described algorithm was programmed on Matlab using the GP toolbox GPLAB[2]; a version written in C language was also implemented with similar results using the LilGP system[3] and the Vision-*something*-Libraries VXL.[4]

4.3.2. Convergence of the GP algorithm

Here, we present the convergence plots of the GP algorithm, by averaging over all 32 runs and computing the standard deviation at each generation, the results are plotted in Fig. 4. Fig. 4(a) shows how the fitness of the best individual in the population improves during the run. It can be seen that fitness mostly improves during the initial generations, and only slightly at the end of most runs, a characteristic which is typical to evolutionary algorithms. Nevertheless, there is a clear pattern of progressive improvements and overall optimization. This is also shown in Fig. 4(b) with the average population fitness, showing a pattern similar to that of the best fitness, but with a higher variance. Fig. 4(c)–(d) presents the evolution dynamics of the size of the best solution found; these figures plot, respectively, the number of nodes and the depth of the corresponding program tree. For the former, Fig. 4(c) shows that even if the size of the tree tends to increase with each generation, the variance is quite large. Hence, it can be argued that code bloat was effectively limited in some runs. This is also evident in Fig. 4(d), where the depth of the best tree does not tend to the maximum allowed depth of 7 levels; thus, the mean value asymptotically reaches a maximum depth of 6.

[1] All Gaussian filters are applied by convolution.

[2] http://gplab.sourceforge.net/, GPLAB a Genetic Programming Toolbox for Matlab.
[3] http://garage.cse.msu.edu/software/lil-gp/.
[4] http://vxl.sourceforge.net/.

Fig. 3. Samples from the Van Gogh sequence used during training.

4.4. Gaussian Intensity Neighborhood interest point detector

In our previous work, we presented a general analysis of the types of image operators that the GP produced [60,62]. It was seen that the GP generates a variety of operators that employ several types of operations; for instance, some operators employ a Difference-of-Gaussian operation, a Laplacian, or sometimes more complex operators such as the sum of second order derivatives or the determinant of the Hessian matrix. In the present paper, however, we describe a single solution found, and utilize it in order to develop a novel interest point detector. For this purpose, we have selected one of the simplest operators generated with the GP, given by

$$K(\mathbf{x}) = G_2 * G_2 * G_1 * \left(\frac{I(\mathbf{x})}{G_2 * I(\mathbf{x})}\right)^2.$$

The above operator represents the exact solution generated by the GP. However, it can be simplified to

$$K_*(\mathbf{x}) = G_{\sigma_1} * \left(\frac{I(\mathbf{x})}{G_{\sigma_2} * I(\mathbf{x})}\right)^2, \tag{9}$$

with $\sigma_1 > \sigma_2$. This operator returns higher values for pixels that are brighter than the weighted average intensity of surrounding pixels computed through a Gaussian mask.

On the other hand, the inverse can be written as

$$K_*^-(\mathbf{x}) = G_{\sigma_1} * \left(\frac{G_{\sigma_2} * I(\mathbf{x})}{I(\mathbf{x})}\right)^2, \tag{10}$$

where K_*^- returns higher values for pixels that are darker than neighboring pixels. However, a similar ordering of pixels can be obtained by using the approximation

$$K_*^-(\mathbf{x}) \propto 1 - K_*(\mathbf{x}), \tag{11}$$

which would require less calculations once K_* is computed. In both cases, only two parameters need to be set, σ_1 and σ_2. The latter controls the size of the local neighborhood that is considered, while the former is simply a blur factor that eliminates high frequency oscillations from the response to K_*.

Based on the above, we propose the *Gaussian Intensity Neighborhood* (GIN) interest point detector, that tags a pixel $\mathbf{x}$ as an interest point if the following conditions hold

$$\begin{aligned} &K_*(\mathbf{x}) > max\left\{K_*(\mathbf{x_W}) \mid \forall \mathbf{x_W} \in \mathbf{W}, \mathbf{x_W} \neq \mathbf{x}\right\} \wedge K_*(\mathbf{x}) > h_1 \vee \\ &K_*^-(\mathbf{x}) > max\left\{K_*^-(\mathbf{x_W}) \mid \forall \mathbf{x_W} \in \mathbf{W}, \mathbf{x_W} \neq \mathbf{x}\right\} \wedge K_*^-(\mathbf{x}) > h_2. \end{aligned} \tag{12}$$

The threshold values h_u set up the ratio between the intensity of an interest point $\mathbf{x}$ and the intensity of neighboring pixels weighted by the Gaussian function. For instance, if we omit the effects of G_{σ_1} and set $h = 2$, then an interest point $\mathbf{x}$ must have an intensity value that is twice as large as the weighted average intensity of pixels located within the Gaussian neighborhood with a standard deviation of σ_2 centered around $\mathbf{x}$. The second smoothing operation performed by G_{σ_1} helps to provide a more stable response to each operator by eliminating high frequency oscillations. In summary, the GIN detector identifies points that are darker or brighter than the Gaussian average of pixels that surround it. Therefore, the GIN detector is similar, in principle, to the detectors proposed in [48,52,59], where the relative intensity of a pixel with respect to neighboring pixels is the determinant factor during the detection process.

In Fig. 5 we present an example of the types of points that the GIN detector extracts from an image, showing the interest image that K_* and K_*^- generate, the points extracted with each operator independently, and all of the points detected with GIN.

In Table 2 we present experimental results that confirm the stability and invariance of the GIN detector by computing the average repeatability rate it achieves on the Van Gogh training sequence, and on four test sets which were also used in [62]. The test sequences are shown in Fig. 6 and have the following characteristics: (1) Mars sequence of 18 images with rotation transformations; (2) New York sequence of 35 images with rotation transformations; (3) Graph sequence of 12 images with illumination change; and (4) Mosaic sequence of 10 images with illumination change. Table 2 also shows the average number of detected points, and the size of all the images from each sequence. In all of the experiments presented here, the parameter values for the GIN detector were set to: $\sigma_1 = 2$, $\sigma_2 = 1$, $h_1 = 1$ and $h_2 = 1$.

In order to help contrast with our previous work we show results of selected interest points detectors that illustrate their performance as well as the complexity of their corresponding evolved operators. Table 3 provides some numbers about the performance of such detectors using the testbed, while Table 4 shows the operators written in prefix notation as obtained with genetic programming. Finally, Fig. 7 shows the interest points that were detected with GIN on some of the images used for testing.

Table 1
General parameter settings for our GP framework.

Parameters	Description and values
Population size	50 individuals.
Generations	50.
Initialization	Ramped half-and-half.
Crossover	Standard crossover.
Crossover and Mutation prob.	Crossover prob. $p_c = 0.85$; mutation prob. $p_\mu = 0.15$.
Tree depth	Dynamic depth selection.
Dynamic max depth	5 levels.
Real max depth	7 levels.
Selection	Tournament selection with lexicographic parsimony pressure.
Survival	Always keep the best solutions in the population (elitism).
Fitness function parameters	$a_x = 7$, $c_x = 5.05$, $a_y = 6$, $c_y = 4.3$, $\alpha = 20$, $\beta = 20$, $\gamma = 2$.

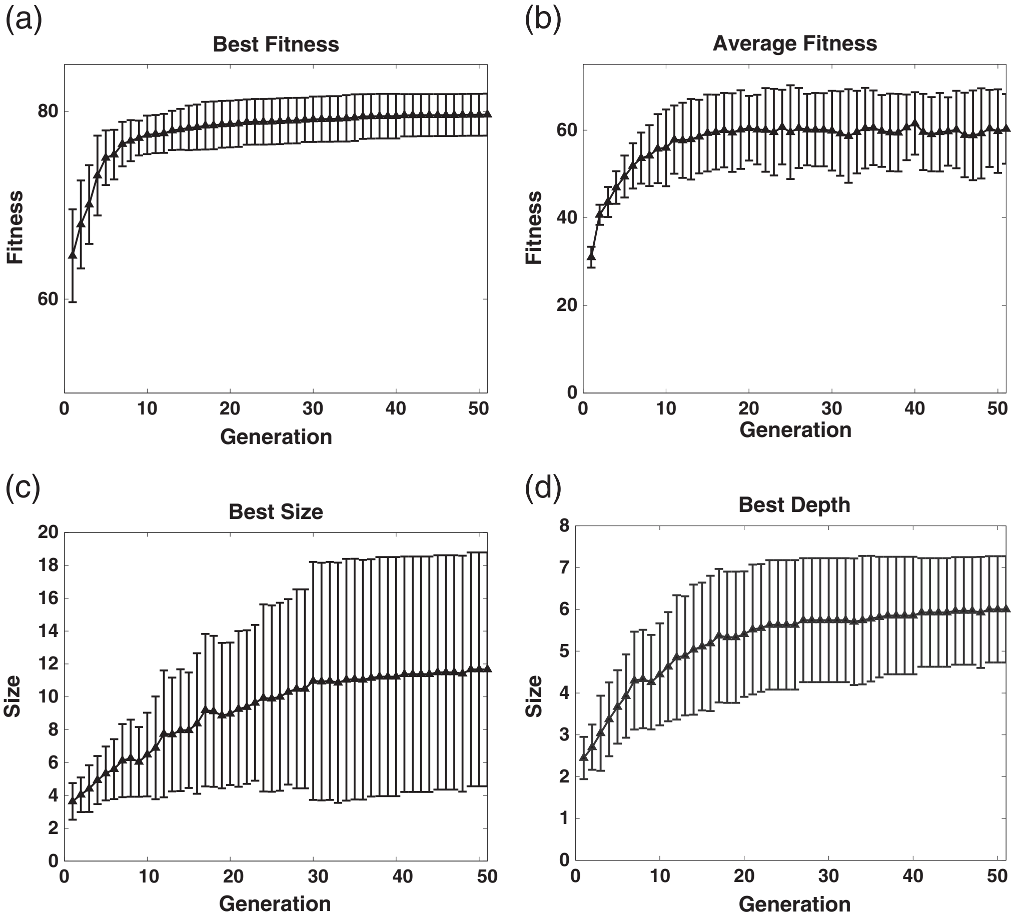

Fig. 4. Convergence plots of the GP algorithm, showing mean value from the 32 runs, and standard deviation across 50 generations: (a) Best fitness; (b) Average population fitness; (c) Size of best solution; and (d) Depth of best solution.

4.5. Discussion

It is evident that the GIN detector performs well on these tests, something that was anticipated given the good performance that other evolved operators also achieve [62]. Moreover, the basic logic that GIN follows is similar to other intensity based methods [48,52,59]. However, there is a sense in which GIN is clearly unique and is because it was developed using an E-CAD approach that successfully integrates artificial evolution and human expertise into a single process. We believe, that such an approach can be extended to other CV problems, especially when the following conditions are fulfilled:

- The problem is of unquestionable difficulty.
- Many different types of solutions are indeed possible, and no theoretical arguments can significantly constrain the size of the search space for possible solutions.
- Reliable evaluation criteria do exist, which are clear, unambiguous and easily computed.

The point detection problem clearly fulfills the above conditions, and the development of the GIN detector confirms the validity of the proposed approach.

In what follows, we describe how the proposed GP algorithm can be extended by using a posteriori combination of optimization objectives with a Pareto-based multi-objective evolutionary algorithm. In this optimization scenario, the evolutionary-CAD process once again provides an appropriate framework for the development of another interest point operator that exhibits special qualities that make it unique within current literature.

5. E-CAD using a multi-objective approach

In the preceding section, it was shown that the single objective GP indeed produced operators that obtain very good performance scores. However, the fitness function in Eq. (5) is dominated by the repeatability term, and the amount of point dispersion is mostly used as a constraint on the search process. At first, this appears to be a reasonable choice because several researches emphasize the importance of stable detection in many real-world systems. However, we suggest that in order to fully exploit the powerful search capabilities that GP offers, then, a less constrained and more thorough exploration of the search space is desirable. One way to achieve this is to use a posteriori articulation of objective preferences, where the optimization process considers multiple criteria independently and the decision maker must select the final trade-off between the objectives after the search process has concluded [5]. Furthermore, one can assume that for many images the proposed performance criteria, stability and point dispersion, will indeed represent conflicting objectives; i.e., they cannot be optimized simultaneously. Therefore, we should not constrain the search process by prescribing the manner in which the objectives should be considered during optimization.

The principles described above form the basis of Pareto-based multi-objective optimization, where a set of optimal solutions is desired instead of the single solution that is normally sought when a single objective is optimized. In these circumstances, population-based approaches, such as evolutionary algorithms, have proven to be a successful choice to carry out the search process because they can provide a diverse sampling of the set of Pareto optimal solutions for difficult problems [5]. Therefore, in this section we describe a multi-

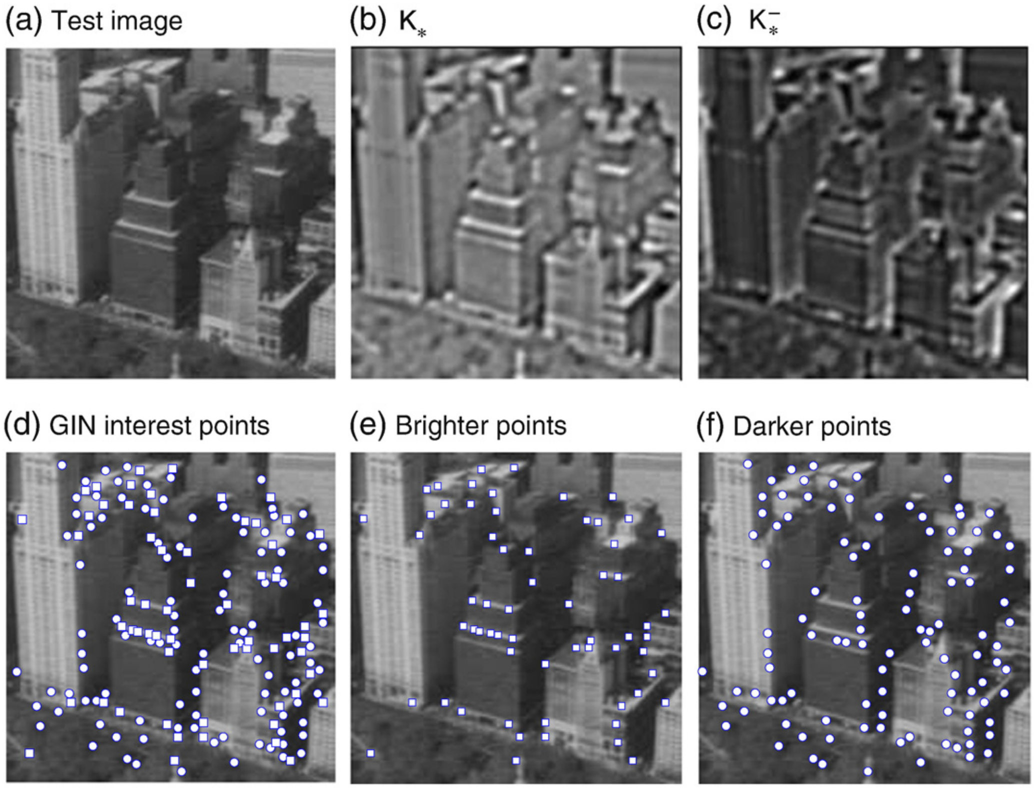

Fig. 5. Interest points detected with the GIN detector: (a) test image; (b,c) interest image computed with K_* and K_*^-; (d) total interest points extracted from the image; (e,f) points detected with K_* and K_*^- respectively.

objective GP (MO-GP) algorithm for the automatic synthesis of interest operators, preliminary results for this approach were given in [63] where information content is also included during the analysis. In the current paper the experiments focus on stability and dispersion because these results lead us to the interest point detector that we called MOP using our E-CAD approach.

5.1. Multi-objective optimization

Multi-objective (MO) optimization, as a separate field of research, is considerably more complex than single criterion optimization. The cardinal difference between single and MO optimization problems is the manner in which the concept of optimality is defined. On the one hand, optimality is trivial in single objective problems because the evaluation function resides in a mono-dimensional space. On the other hand, in MO problems optimality is based on dominance relations among different solutions in a multidimensional space.

In MO optimization two different and complimentary spaces exist: one for decision variables and another for their evaluation on the objective functions. In the case of real valued functions, these two spaces are related by the mapping $\vec{f}:\mathcal{R}^n \rightarrow \mathcal{R}^k$. The set of constraints on $\vec{f}(\mathbf{x}) = [f_1(x), \ldots, f_k(x)]$ define a feasible region $\Omega \subset \mathcal{R}^n$ in the decision space along with its corresponding image $\Lambda \subset \mathcal{R}^n$ on the objective function space, this is depicted in Fig. 8. The optimum is found at the frontier of the objective space called the *Pareto front*, while their corresponding decision variable values in Ω are called the *Pareto optimal set*.

The concept of *Pareto Dominance*, specified in objective space, is defined as follows. An objective vector $\vec{f^i}$ is said to dominate another objective vector $\vec{f^j}, \vec{f^i} \succ \vec{f^j}$, if no component of $\vec{f^i}$ is larger (considering a minimization problem) than the corresponding component of $\vec{f^j}$, and at least one component is smaller. For some problems, the objectives are in conflict between them, and thus a single optimal solution cannot exist. On the contrary, there is a set of multiple feasible solutions and all of them are optimal in the Pareto sense.

5.1.1. Multi-Objective Evolutionary Algorithms (MOEAs)

When a MO problem lacks a closed form solution, it is necessary to rely on computational search methods in order to obtain an approximation for the true Pareto optimal set. In such cases, a MO optimization algorithm should fulfill the following characteristics:

1. *It must converge towards the true Pareto front*. This is analogous to the global optimum in a mono-objective problem, and it is difficult to achieve when the objective functions are discontinuous or irregular.
2. *It must representatively sample the true Pareto front*. This means that the set of solutions should be diverse and widely dispersed along the entire Pareto front. Nonetheless, depending on the structure of the objective space some regions of the front may not be reachable.

Over the past two decades the EC paradigm has proven to be a good choice for solving MO optimization problems, with techniques that are called *Multi-Objective Evolutionary Algorithms* (MOEAs) [5].

Table 2
Average repeatability rate computed for the GIN detector on five different image sequences, and the average number of interest points extracted from each image in the sequence. In all of the experiments the parameters of the GIN detector were set to: $\sigma_1 = 2$, $\sigma_2 = 1$, $h_1 = 1$ and $h_2 = 1$.

Name	r_J	No. of points	Size in pixels	Training
Van Gogh	92.99	774.9	348 × 512	Yes
Mars	94.65	1309.3	842 × 842	No
New York	88.51	972.35	512 × 512	No
Graph	88.61	642.00	512 × 512	No
Mosaic	98.00	569.61	512 × 512	No

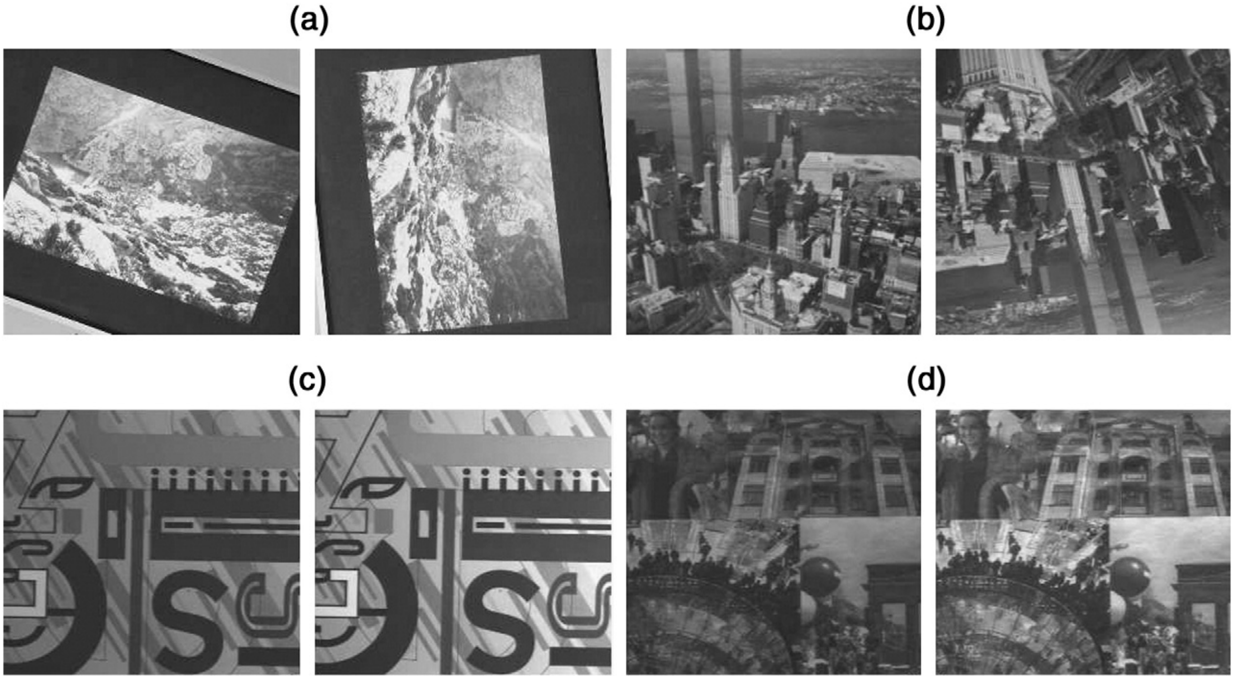

Fig. 6. The reference image (right) and a transformed image (left) for each testing sequence. (a) Mars, (b) New York, (c) Graph, and (d) Mosaic.

Currently, many flavors of MOEAs exist, including third generation algorithms, such as NSGA-II [10], PESA [6], and SPEA2 [69], however all of them address three main design issues. First, fitness assignment considers the MO nature of the problem in order to bias the search towards the Pareto front by considering dominance relations between individual solutions. Second, because a uniform sampling of the Pareto front is desired, diversity preservation is incorporated into the search process. In MOEAs, diversity is procured in objective space, using kernel methods [10], clustering methods [69], histogram methods [6], or by applying the concept of *-dominance* [11]. Finally, all state-of-the-art MOEAs implement elitism using population archiving. For a comprehensive review on this topic the reader is referred to [5,68]. In the present work, the *improved* Strength Pareto Evolutionary Algorithm [69] is chosen as the basis for our GP-MO algorithm.

5.1.2. Improved Strength Pareto Evolutionary Algorithm

The *improved Strength Pareto Evolutionary Algorithm* (SPEA2) is a third generation MOEA, an improved version of the second generation SPEA. Empirical results suggest that SPEA2 outperforms other MOEAs on a comprehensive set of difficult benchmark tests [11,68,69]. Those results showed that SPEA2 obtains a good approximation of the true Pareto front and maintains a set of solutions that are highly distributed in objective space. The fitness assignment used in SPEA2 [69] accounts for dominance and non-dominance relations between individuals in the current population and individuals from past generations. Diversity preservation is carried out using a k-th nearest neighbor clustering algorithm that penalizes individuals that reside in densely populated regions of objective space. It uses a fixed-size archiving approach, and a truncation scheme promotes diversity by removing individuals that have the minimum distance to their neighbors. Finally, it preserves boundary solutions by using a carefully designed selection operator.

5.2. Objective functions

Recalling Fig. 2, the implementation of a MO-GP search does not require much modification, only the appropriate alterations to the basic processes. Specifically, *Population management* is managed by the SPEA-2 algorithm. However, the *Variation* and *Population evaluation* modules work as in the mono-objective case. The only additional modification is in fitness assignment, where $\vec{f}(K)$ is a vector containing each of the possible objectives. Hence, in order to set up the design as a minimization problem, we define the following objective (cost) functions:

- Stability: $f_1(K) = \dfrac{1}{r_{K,J}(\phi) + \phi}$.
- Point dispersion: $f_2(K) = \dfrac{1}{exp(\mathcal{D}(I,X) - C_1)}$.

Table 3
Average repeatability rate achieved by other evolved detectors. These results were computed on seven sequences, including the training and testing sets, that were obtained from the testbed. The acronym IPGP refers to the Matlab implementation while C-IPGP to the C language system.

Detector	Van Gogh	Monet	Mars	New York	Graph	Mosaic	Leuven
IPGP1	96.41	85.17	91.42	88.41	92.04	93.11	69.06
IPGP2	93.74	94.80	86.26	83.47	95.97	93.79	63.18
Harris	90.71	92.96	89.90	89.75	98.00	94.43	70.42
C-IPGP1	98.33	86.73	96.00	94.24	94.63	96.51	70.74
C-IPGP2	97.75	89.40	95.53	94.64	93.67	96.45	71.22
C-IPGP5	96.49	86.69	96.00	95.37	93.76	95.98	76.21
C-IPGP6	95.90	82.72	95.85	96.57	95.03	95.82	76.08

Table 4
The following detectors are provided to illustrate the complexity of the evolved detectors in contrast with the Harris operator using prefix notation. In the case of results obtained with LilGP the constant $k = 0.25$.

Operator	Symbolic expression
IPGP1	$G_{\sigma=2}(-(G_{\sigma=1}(I),I))$
IPGP2	$-(G_{\sigma=1}(*(L_{xx},L_{yx})),G_{\sigma=1}(*(L_{xx},L_{yx})))$
Harris	$-(-(*(G_{\sigma=2}(I^2_{out}(L_x)),G_{\sigma=2}(I^2_{out}(L_y))),I^2_{out}(G_{\sigma=2}(*(L_x,L_y)))),$ $k\cdot I_{out}(+(G_{\sigma=2}(I^2_{out}(L_x)),G_{\sigma=2}(I^2_{out}(L_y))))),$
C-IPGP1	$G_{\sigma=1}(G_{\sigma=2}(G_{\sigma=2}(G_{\sigma=2}(+(L_{xx},L_{yy})))))$
C-IPGP2	$G_{\sigma=1}(G_{\sigma=1}(G_{\sigma=2}(G_{\sigma=2}(-(G_{\sigma=2}(G_{\sigma=1}(I)),G_{\sigma=1}(G_{\sigma=1}(I)))))))$
C-IPGP5	$G_{\sigma=2}(G_{\sigma=2}(-(I,G_{\sigma=2}(G_{\sigma=2}(abs(-(k\cdot I_{out}(L_{xy}),I)))))))$
C-IPGP6	$G_{\sigma=2}(G_{\sigma=2}(G_{\sigma=2}(-(I,G_{\sigma=1}(I)))))$

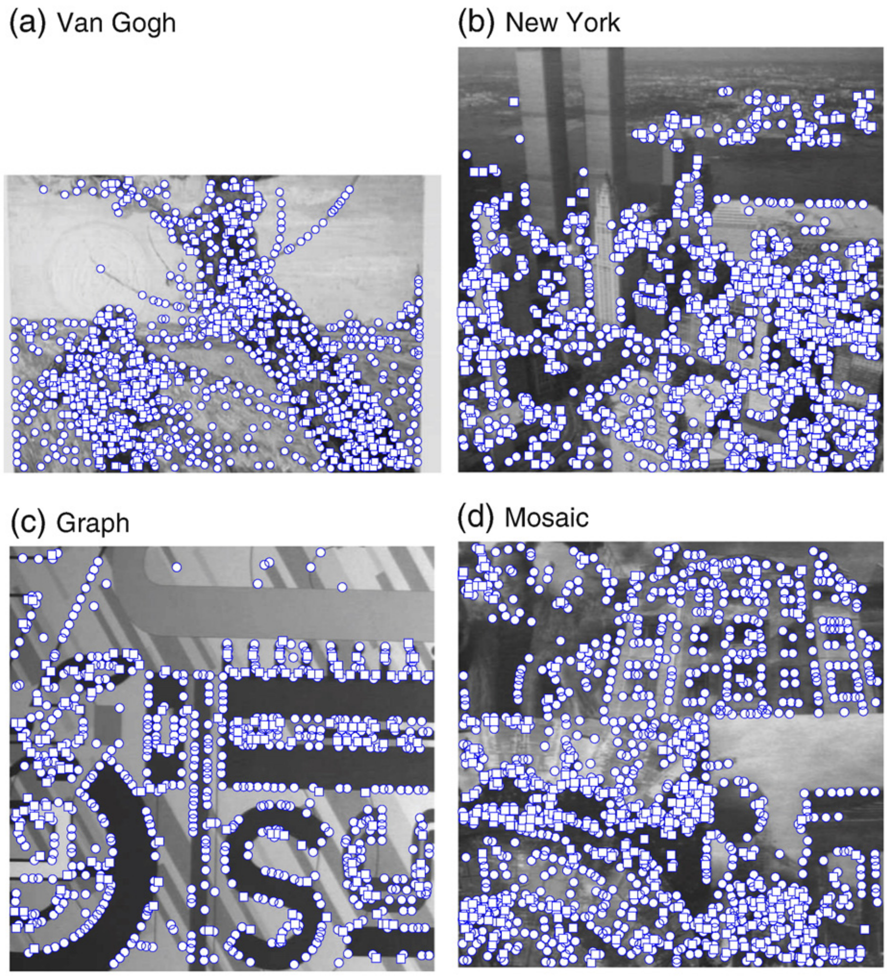

Fig. 7. GIN interest points from some of the images used for testing. Square points are detected with K_*, and circle points with K_*^-.

The constants were set experimentally to $\phi = 0.001$ and $C_1 = 10$, the former avoids a division by zero and the latter shifts the function response closer to the origin to simplify the equation. The following experimental runs take into account both criteria concurrently to look for Pareto optimal solutions.

5.3. Experimentation

This section provides results of the experimental set-up and describes the proposed MO-GP algorithm.

5.3.1. Implementation details

The implementation is similar to that used with the single objective approach, with the following important modifications. The selection and survival mechanisms depend on the SPEA2 algorithm, and we use the code made available by the *Platform and Programming Language Independent Interface for Search Algorithms*[5] project. This implementation allows a simple file based communication between SPEA2 and the GPLAB toolbox; the parameters are given in Table 5.

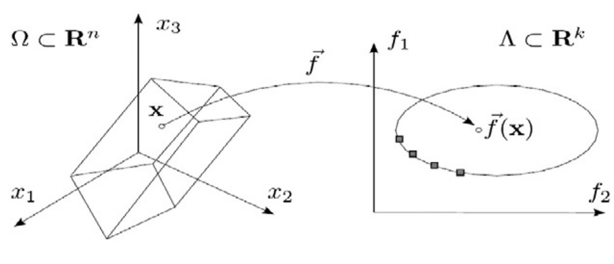

Fig. 8. *Decision* and *Objective* Space for MO optimization. A solution parameterization $\mathbf{x}$ is mapped by a vector function $\vec{f}$ into a vector in objective function space. The highlighted points on the boundary of Λ are elements of the Pareto front.

5.3.2. Results

With the MO-GP we have generated the Pareto fronts shown in Fig. 9, different sets of solutions were obtained by changing the maximum allowed depth for individual program trees. For comparative purposes, the figure also shows the performance of four well known interest point detectors: Harris and Stephens [16], Beaudet [1], Kitchen and Rosenfeld [23] and Förstner [14]. Additionally, the plot also shows the performance of two operators that were evolved using the single objective approach described in the previous section: K_{IPGP1} and K_{IPGP2} [60,62].

The plot in Fig. 9 shows that the Pareto optimal solutions found by the MO-GP dominate, in the Pareto sense, all of the other detectors included in the comparison. However, some of those operators achieve comparable performance with respect to the stability criterion, but are far less competitive when point dispersion is considered. Indeed, Fig. 9 confirms that previous proposals were more concerned with achieving a stable and invariant detection process. Moreover, the two operators that were evolved using the single objective approach are also biased towards the stability criterion,

[5] http://www.tik.ee.ethz.ch/sop/pisa/.

Table 5
General parameter settings for the MO-GP algorithm.

Parameters	Description and values
Population size	200 individuals.
Generations	50 generations.
Initialization	Ramped half-and-half.
Crossover and Mutation prob.	Crossover prob. $p_c = 0.85$; mutation prob. $p_\mu = 0.15$.
Max depth	3, 5, 7 and 9 levels.
Archive size	The SPEA2 archive size: 100.
Selection size	The amount of individuals selected by SPEA2: 100.

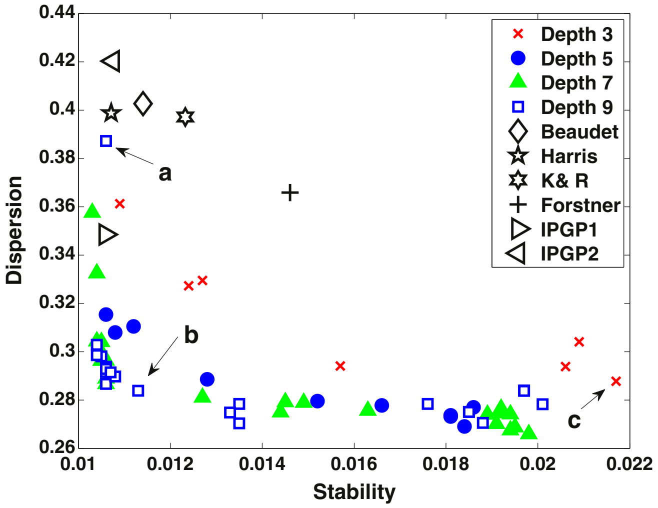

Fig. 9. The Pareto fronts found with each of the maximum allowed depths. The plot also shows the performance obtained by four other detectors: Harris and Stephens [16], Beaudet [1], Kitchen and Rosenfeld [23] and Förstner [14]. Additionally, two operators that were evolved using the single objective approach described in Section 4 are also shown: K_{IPGP1} and K_{IPGP2} [60,62]. Finally, the plot also identifies three operators located at what could be considered as extreme points on the Pareto front: (**a**), (**b**) and (**c**).

something that was anticipated given the manner in which Eq. (5) incorporates the dispersion terms.

Fig. 9 also identifies three operators located at what could be considered as extreme points on the Pareto front: (**a**), (**b**) and (**c**). The differences between these operators can be qualitatively seen in Fig. 10 where the interest image and the corresponding interest points are shown for the Van Gogh image. The conflict between both objectives is evident. For instance, operator (**a**) detects points that are cluttered together and very repeatable, while operator (**c**) detects very sparse points that are unstable. Perhaps, the best trade-off is given by operator (**b**), it obtains a good compromise between both objectives. In Table 6 the mathematical expressions derived by the MO-GP are shown, with only slight simplifications in order to simplify their interpretation.

5.4. Multi-Objective Parameterized interest point detector

In this section we derive a novel interest point detector by analyzing some of the solutions generated by the MO-GP. First, let us consider operator (**c**) which is inversely proportional to the local curvature around each point computed along the y direction. Interest points are indeed highly dispersed, however stability is poor because it only considers one principal direction. The same operation computed over the x direction produced a similar performance on both objectives.

On the other hand, operators (**a**) and (**b**) are in fact very similar since both of them contain an absolute sum of three terms, see Table 6. Note, a non-linear logarithmic term that depends on the intensity of each point. Finally, a third term that describes the same ratio used within the GIN detector through the K_*^- operator, see Eq. (10). Indeed, the first and last terms in operators (**a**) and (**b**) are the same, the difference between them lies in the second term, a simple DoG filter. The difference can be expressed as a single scale factor that modulates the second term, and we can therefore write a more general expression for both (**a**) and (**b**) as follows,

$$K_{MO} = G_2 * \left| G_1 * log\left(G_1 * I^2\right) + W \cdot G_2 * |G_1 * I - I| + \frac{G_1 * I}{I} \right|^2,$$

where W is the scale factor that controls the amount of point dispersion, and for convenience we can write this operator as

$$K_{MO} = G_2 * \left| K^1_{MO} + W \cdot K^2_{MO} + K^3_{MO} \right|^2, \quad (13)$$

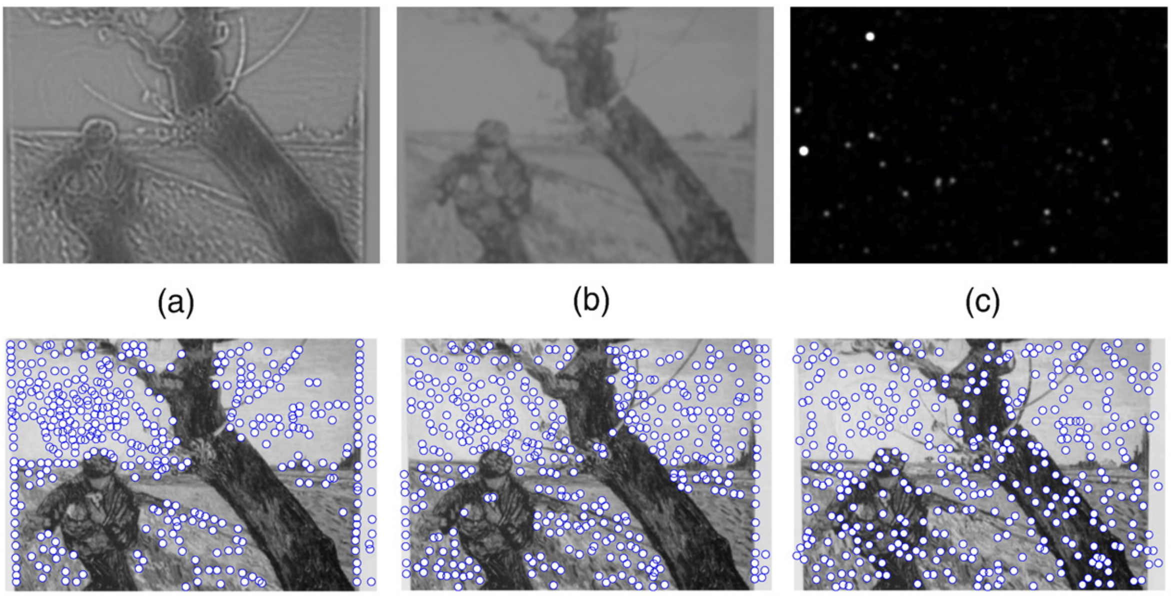

Fig. 10. Interest image (first row) and interest points (second row) on the Van Gogh image obtained with operators (**a**), (**b**) and (**c**) that are located on the Pareto front of Fig. 9.

Table 6
Symbolic expression for operators (**a**), (**b**) and (**c**) located on the Pareto front of Fig. 9.

Operator	Symbolic expression
Operator (**a**)	$G_2 * \left\lvert G_1 * log(G_1 * I^2) + G_2 * (G_1 * I - I) + \frac{G_1 * I}{I} \right\rvert^2$
Operator (**b**)	$G_2 * \left\lvert G_1 * log(G_1 * I^2) + k \cdot G_2 * \lvert G_1 * I - I \rvert + \frac{G_1 * I}{I} \right\rvert^2$
Operator (**c**)	$G_2 * \left(\frac{L_y}{L_{yy}}\right)$

where $K^1_{MO} = G_1 * log(G_1 * I^2)$, $K^2_{MO} = G_2 * |G_1 * I - I|$ and $K^3_{MO} = \frac{G_1 * I}{I}$. However, in practice the third term K^3_{MO} is several orders of magnitudes smaller than the first two, and operator K_{MO} can therefore be simplified to

$$K_{MO} = G_2 * \left\lvert K^1_{MO} + W \cdot K^2_{MO} \right\rvert^2. \quad (14)$$

Without any noticeable difference in performance. Operator K_{MO} is the interest operator used with our proposed *Multi-Objective Parameterized interest point detector* (MOP). In order to understand the usefulness of this detector, and the characteristics that make it unique, lets further analyze K_{MO}.

As stated above, K^2_{MO} is a DoG filter and it therefore enhances image borders and edges. Therefore, if we assume that on many real-world images borders and edges are not equally distributed over the image then we can understand how W regulates the dispersion of interest points by determining the relative importance of K^2_{MO} when the interest measure is computed. The usefulness of W comes from the fact that the difference in performance between (**a**) and (**b**) is based on the amount of dispersion that they produce. Hence, the stability is basically equivalent between them. We can then use W to modify the amount of dispersion without incurring in loss of stability for our detector.

Let us now test the effect of W during detection, for this we use the Van Gogh sequence and plot the stability and dispersion relative to W which we vary within the range of $[-1, 1]$ with 0.05 increments; the results are plotted in Fig. 11.

In Fig. 11(a) we can clearly appreciate a discontinuity at $W > 0$; i.e., for $W \leq 0$ the detector is not stable, and the opposite is true when $W > 0$. Fig. 11(b) shows how the dispersion of interest points is affected by W. In this case, the best dispersion is obtained when $W = 0.05$ which basically corresponds with operator (**b**). If we consider $W \in (0, 1]$, then we can conclude the following:

- Stability is mostly unaffected by W, with a very good repeatability rate.
- Point dispersion varies proportionally with respect to W.

The MOP detector effectively provides a parameter that allows for a fine control over the amount of point dispersion while geometric stability remains unaffected. Similar performance patterns are obtained when we apply the same test to other image sequences; Fig. 12 shows the results for the New York and Mars sequences.

In order to illustrate the qualitative effects of W over all points detected by MOP we present Fig. 13 that shows detected interest points on three different images using $W \in \{0, 0.05, 0.5, 1\}$. In every image, we can observe that the dispersion of interest points varies with respect to W, although for some images this difference is more noticeable than for others.

In CV, the most widely used detector was proposed by Harris and Stephens [16], and it also contains a tunable parameter k,

$$K_{\text{Harris\&Stephens}}(\mathbf{x}) = det(A) - k \cdot Tr(A)^2,$$

where A is the autocorrelation matrix. Therefore, for comparison Fig. 14 shows the results of conducting the same test on $K_{\text{Harris\&Stephens}}$ that were carried out for K_{MO} in Fig. 11.

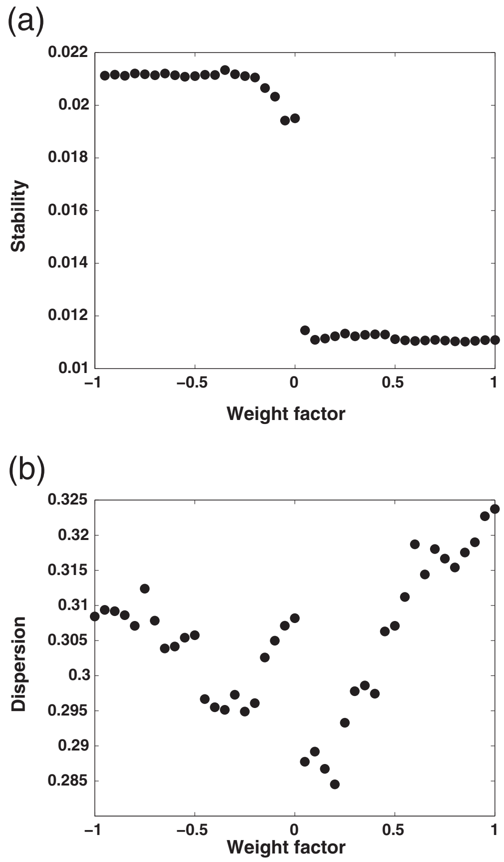

Fig. 11. The effects of scale factor W on the performance of K_{MO}. (a) Stability: the best performance is achieved with $W \in (0, 1]$. (b) Dispersion: the minimum corresponds with operator (**b**). Notice that point dispersion is effectively controlled by W without adversely affecting the geometric stability of the detection process.

In both plots shown in Fig. 14, the performance of $K_{\text{Harris\&Stephens}}$ is very sensitive to values of $k > 0.3$; in fact performance is so bad that the corresponding points are outside the range shown. Moreover, in both cases the overall performance seems largely unaffected for values of $k \leq 0.3$, which suggests that the parameter does not provide fine control over any of the objectives. Finally, in the case of point dispersion, $K_{\text{Harris\&Stephens}}$ never reaches a comparable performance to that achieved by the MOP detector.

5.5. Discussion

The MOP detector proposed in this section was effectively designed using an E-CAD approach that successfully integrates a MO problem statement based on Pareto optimality, a GP evolutionary search process, and a detailed analysis of the Pareto set of solutions. Such an approach is definitely not standard, particularly in CV problems. However, the MOP detector gives encouraging evidence regarding the validity and utility that such a design process could offer. Through the MO-GP, we were able to construct a parameterized detector that allows a user to control the dispersion of detected points without sacrificing geometric stability, a characteristic that effectively makes it unique in current literature.

6. Conclusions and future perspectives

In this work an evolutionary computed assisted design process was described as an innovative method for designing interest point detectors. Indeed, two novel image operators were proposed as a byproduct of the analysis of multiple results achieved by single and

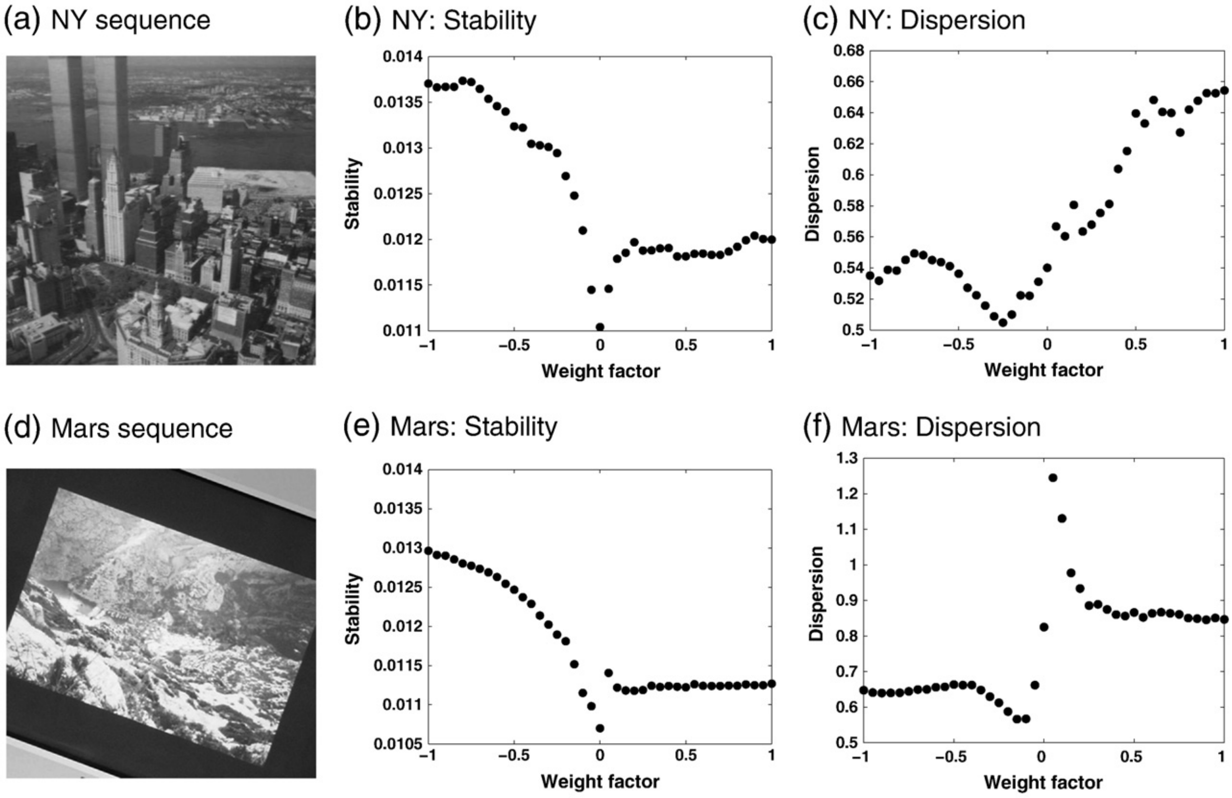

Fig. 12. Effects of scale factor W on the performance of K_{MO} when applied to two test sequences: New York (first row) and Mars (second row).

multi-objective genetic-programming searches. In particular, the GIN detector was designed from the pool of results obtained with the single objective formulation. Such a detector identifies points that are darker or brighter than the Gaussian average of pixels that surround it; hence, it resembles other detectors that follow similar assumptions but with a significant difference; it was designed with an E-CAD

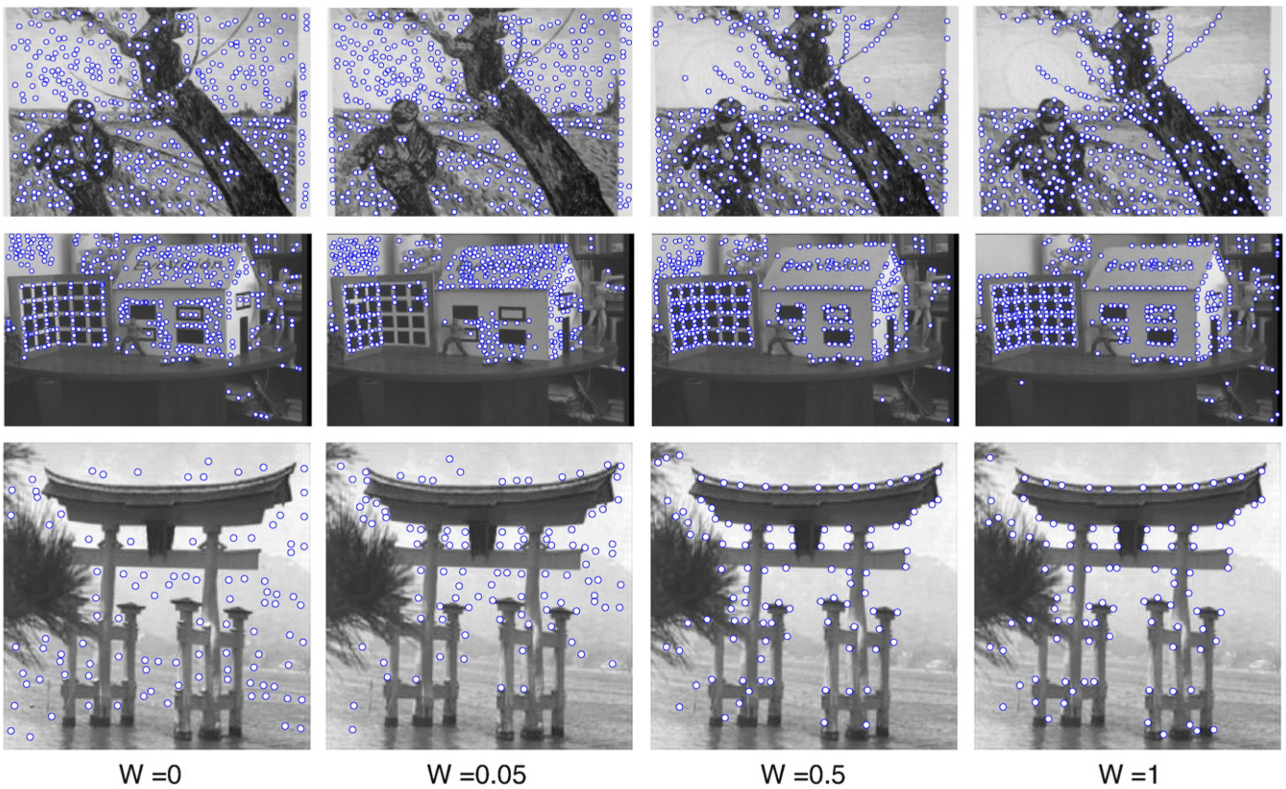

Fig. 13. MOP points detected on three test images with $W \in \{0, 0.05, 0.5, 1\}$.

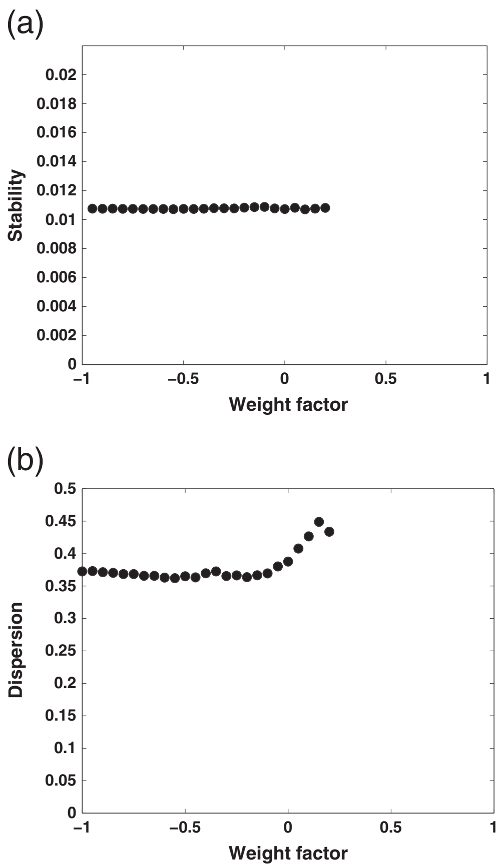

Fig. 14. Effects of scale factor k on the performance of $K_{\text{Harris\&Stephens}}$ when applied to the Van Gogh sequence.

approach that successfully integrates evolutionary learning and human expertise into a single process. This new approach was later extended into a multi-objective framework to produce a parameterized interest point detector (MOP) that presents unique characteristics that help to fine-tune the point distribution across the image without sacrificing the repeatability rate. We claim that this approach could be extended to other CV problems that are of unquestionable difficulty, with no analytical solution, but with reliable and easily computed metrics. Finally, we believe that it is trivial to extend the human-machine learning framework in order to design specialized operators for different domains by incorporating the requirements of specific applications.

Acknowledgments

First author dedicate this work to his brother José Agustin who passed away while the first draft of this paper was written. Fig. 1 shows a photograph of a painting done during his early years that demonstrate remarkable artistic talent that was later shown on his activities as a legist and his love to jurisprudence.

References

[1] P.R. Beaudet, Rotational invariant image operators, Proceedings of the 4th International Joint Conference on Pattern Recognition (ICPR),Tokyo, Japan, 1978, pp. 579–583.
[2] G. Bradski, A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly Media, 2008.
[3] S. Cagnoni, E. Lutton, G. Olague, Editorial introduction to the special issue on evolutionary computer vision, Evolutionary Computation 16 (4) (2008) 437–438.
[4] 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.
[5] C. Coello, D.V. Veldhuizen, G. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers, New York, New York, 2002.
[6] D. Corne, J.D. Knowles, M.J. Oates, The Pareto envelope-based selection algorithm for multi-objective optimisation, PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature, Springer-Verlag, London, UK, 2000, pp. 839–848.
[7] L. da Fountoura-Costa, R. Marcondes-Cesar, Shape Classification and Analysis: Theory and Practice, CRC Press, Taylor and Francis Group, LLC, 2009.
[8] A.J. Davison, N.D. Molton, Monoslam: real-time single camera slam, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (6) (2007) 1052–1067 member-Ian D. Reid and Member-Olivier Stasse.
[9] K. De Jong, Evolutionary Computation: A Unified Approach, The MIT Press, 2001.
[10] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-ii, IEEE Trans. Evolutionary Computation 6 (2) (2002) 182–197.
[11] K. Deb, M. Mohan, S. Mishra, A fast multi-objective evolutionary algorithm for finding well-spread Pareto-optimal solutions, KanGAL report 2003002, Indian Institute of Technology, Kanpur, India, 2003.
[12] S.J. Dickinson, A. Leonardis, B. Schiele, M.J. Tarr, Object Categorization: Computer and Human Vision Perspectives, Cambridge University Press, New York, NY, USA, 2009.
[13] W. Förstner, A feature based correspondence algorithm for image matching, International Archives of Photogrammetry and Remote Sensing 26 (3) (1986) 150–166.
[14] W. Förstner, E. Gülch, A fast operator for detection and precise location of distinct points, corners and centres of circular features, ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data, 1987, pp. 149–155.
[15] D.A. Forsyth, J. Ponce, Computer vision: a modern approach, Prentice Hall Professional Technical Reference, 2002.
[16] C. Harris, M. Stephens, A combined corner and edge detector, Proceedings from the Fourth Alvey Vision Conference. Vol. 15, 1988, pp. 147–151.
[17] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, New York, NY, USA, 2003.
[18] 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 106 (2–3) (2007) 258–269.
[19] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, 1975.
[20] D. Howard, S.C. Roberts, C. Ryan, Pragmatic genetic programming strategy for the problem of vehicle detection in airborne reconnaissance, Pattern Recognition Letters 27 (11) (2006) 1275–1288.
[21] W. Jaśkowski, K. Krawiec, B. Wieloch, Multitask visual learning using genetic programming, Evolutionary Computation 16 (4) (2008) 439–459.
[22] C.S. Kenney, M. Zuliani, B.S. Manjunath, An axiomatic approach to corner detection, International Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2005.
[23] L. Kitchen, A. Rosenfeld, Gray-level corner detection, Pattern Recognition Letters 1 (December 1982) 95–102.
[24] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, MA, USA, 1992.
[25] J.R. Koza, Human-competitive results produced by genetic programming, Genetic Programming and Evolvable Machines 11 (3/4) (2010) 251–284.
[26] 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.
[27] 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.
[28] K. Krawiec, B. Bhanu, Visual learning by coevolutionary feature synthesis, IEEE Transactions on Systems, Man, and Cybernetics, Part B 35 (3) (2005) 409–425.
[29] W.B. Langdon, R. Poli, Foundations of Genetic Programming, Springer-Verlag, New York, New York, 2002.
[30] Y. Lin, B. Bhanu, Evolutionary feature synthesis for object recognition, IEEE Transactions on Systems, Man and Cybernetics, Part C, Special Issue on Knowledge Extraction and Incorporation in Evolutionary Computation 35 (2) (2005 May) 156–171.
[31] 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.
[32] J. McGlone, et al., (Eds.), Manual of Photogrammetry, American Society of Photogrammetry and Remote Sensing, 2004.
[33] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (10) (2005) 1615–1630.
[34] J. Miles, L. Hall, J. Noyes, I.C. Parmee, C.L. Simons, A forward look at computational support for conceptual design, Intelligent Computing in Engineering and Architecture, 13th EG-ICE Workshop 2006, Ascona, Switzerland, June 25–30, Lecture Notes in Computer Science, vol. 4200, Springer, 2006, pp. 492–499.
[35] P. Moreels, P. Perona, Evaluation of features detectors and descriptors based on 3d objects, International Journal of Computer Vision 73 (3) (2007) 263–284.
[36] H. Müller, P. Clough, T. Deselaers, B. Caputo, Image CLEF: Experimental Evaluation in Visual Information Retrieval, Springer, 2010.
[37] Noble, A., 1989. Descriptions of image surfaces. Ph.D. thesis, Department of Engineering Science, Oxford University.
[38] G. Olague, Automated photogrammetric network design using genetic algorithms, Photogrammetric Engineering & Remote Sensing 68 (5) (2002) 423–431.

[39] G. Olague, S. Cagnoni, E. Lutton, Preface: introduction to the special issue on evolutionary computer vision and image understanding, Pattern Recognition Letters 27 (11) (2006) 1161–1163.
[40] G. Olague, B. Hernández, A new accurate and flexible model based multi-corner detector for measurement and recognition, Pattern Recognition Letters 26 (1) (2005) 27–41.
[41] G. Olague, R. Mohr, Optimal camera placement for accurate reconstruction, Pattern Recognition 35 (2002) 927–944.
[42] I.C. Parmee, Strategies for the integration of evolutionary/adaptive search with the engineering design process, in: D. Dasgupata, Z. Michalewicz (Eds.), Evolutionary Algorithms in Engineering Applications, Springer-Verlag, Berlin, Germany, 1997, pp. 453–477.
[43] C.B. Perez, G. Olague, Learning invariant region descriptor operators with genetic programming and the f-measure, IEEE International Conference on Pattern Recognition, 2008, pp. 1–4.
[44] C.B. Perez, G. Olague, Evolutionary learning of local descriptor operators for object recognition, GECCO '09: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2009, pp. 1051–1058.
[45] R. Poli, W.B. Langdon, N.F. Mcphee, A Field Guide to Genetic Programming, March 2008.
[46] C. Puente, G. Olague, S.V. Smith, S. Bullock, M.A. Gonzalez, A. Hinojosa, Genetic programming methodology that synthesize vegetation indices for the estimation of soil cover, GECCO '09: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2009, pp. 1593–1600.
[47] C. Puente, G. Olague, S.V. Smith, S. Bullock, A. Hinojosa, M.A. Gonzalez, A genetic programming approach to estimate vegetation cover in the context of soil erosion assessment, Photogrammetric Engineering & Remote Sensing 77 (4) (2011) 363–376.
[48] E. Rosten, R. Porter, T. Drummond, Faster and better: a machine learning approach to corner detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (1) (2010).
[49] C. Schmid, R. Mohr, Local grayvalue invariants for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (5) (May 1997) 530–534.
[50] C. Schmid, R. Mohr, C. Bauckhage, Evaluation of interest point detectors, International Journal of Computer Vision 37 (2) (2000) 151–172.
[51] J. Shi, C. Tomasi, Good features to track, Proceedings of the 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94), June 1994, Seattle, WA, USA, IEEE Computer Society, 1994, pp. 593–600.
[52] S.M. Smith, J.M. Brady, Susan—a new approach to low level image processing, International Journal of Computer Vision 23 (1) (1997) 45–78.
[53] A. Song, V. Ciesielski, Texture segmentation by genetic programming, Evolutionary Computation 16 (4) (2008) 461–481.
[54] L. Spector, An essay concerning human understanding of genetic programming, in: R. Riolo, W. Worzel (Eds.), Genetic Programming: Theory and Practice, Kluwer Academic Publishers, Boston, MA, USA, 2003, pp. 11–24.
[55] L. Spector, Evolution of artificial intelligence, Artificial Intelligence 170 (18) (2006) 1251–1253.
[56] L. Spector, D.M. Clark, I. Lindsay, B. Barr, J. Klein, Genetic programming for finite algebras, GECCO '08: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2008, pp. 1291–1298.
[57] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2011.
[58] P. Tissainayagam, D. Suter, Assessing the performance of corner detectors for point feature tracking applications, Image and Vision Computing 22 (8) (2004) 663–679.
[59] M. Trajkovic, M. Hedley, Fast corner detection, Image and Vision Computing 16 (2) (1998) 75–87.
[60] 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), Seattle, Washington, July 8–12. Vol. 1, ACM, 2006, pp. 887–894.
[61] L. Trujillo, G. Olague, Using evolution to learn how to perform interest point detection, Proceedings of the 18th International Conference on Pattern Recognition (ICPR), 20–24 August, Hong Kong, China. Vol. 1, IEEE Computer Society, 2006, pp. 211–214.
[62] L. Trujillo, G. Olague, Automated design of image operators that detect interest points, Evolutionary Computation 16 (4) (2008) 483–507.
[63] 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), Atlanta, GA, July 12–16, ACM, New York, NY, USA, 2008, pp. 1299–1306.
[64] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: a survey, Found. Trends Comput. Graph. Vis. 3 (3) (2008) 177–280.
[65] G. Yang, C.V. Stewart, M. Sofka, C.-L. Tsai, Registration of challenging image pairs: initialization, estimation, and decision, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (11) (2007) 1973–1989.
[66] M. Zhang, V.B. Ciesielski, P. Andreae, A domain-independent window approach to multiclass object detection using genetic programming, EURASIP Journal on Applied Signal Processing, Special Issue on Genetic and Evolutionary Computation for Signal Processing and Image Analysis July 2003 (8) (2003) 841–859.
[67] Y. Zhang, P.I. Rockett, Evolving optimal feature extraction using multi-objective genetic programming: a methodology and preliminary study on edge detection, GECCO '05: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation. ACM, New York, NY, USA, 2005, pp. 795–802.
[68] E. Zitzler, M. Laumanns, S. Bleuler, A tutorial on evolutionary multiobjective optimization, in: X. Gandibleux, et al., (Eds.), Metaheuristics for Multiobjective Optimisation, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, 2004.
[69] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization, Evolutionary Methods for Design, Optimisation, and Control, 2002, pp. 19–26.