

Genetic programming as strategy for learning image descriptor operators

Cynthia B. Perez and Gustavo Olague*

CICESE Research Center, Centro de Investigación Científica y de Educación Superior de Ensenada, Ensenada, B.C., México

Abstract. Nowadays, object recognition based on local invariant features is widely acknowledged as one of the best paradigms for object recognition due to its robustness for solving image matching across different views of a given scene. This paper proposes a new approach for learning invariant region descriptor operators through genetic programming and introduces another optimization method based on a hill-climbing algorithm with multiple re-starts. The approach relies on the synthesis of mathematical expressions that extract information derived from local image patches called local features. These local features have been previously designed by human experts using traditional representations that have a clear and, preferably mathematically, well-founded definition. We propose in this paper that the mathematical principles that are used in the description of such local features could be well optimized using a genetic programming paradigm. Experimental results confirm the validity of our approach using a widely accepted testbed that is used for testing local descriptor algorithms. In addition, we compare our results not only against three state-of-the-art algorithms designed by human experts, but also, against a simpler search method for automatically generating programs such as hill-climber. Furthermore, we provide results that illustrate the performance of our improved SIFT algorithms using an object recognition application for indoor and outdoor scenarios.

Keywords: Local descriptors, genetic programming, SIFT, object recognition, F-measure, hill-climbing

1. Introduction

Description of interest regions is an attractive and successful paradigm of computer vision that is commonly applied in state-of-the-art systems for solving challenging tasks such as: object detection and recognition, image matching, robot navigation, image retrieval and visual data mining, to name but a few application domains. This work follows the idea of evolving local descriptor operators through a domain independent machine learning paradigm known as genetic programming. The application of genetic programming (GP) to image related problems has received an increase level of attention since Koza introduced GP as a problem solving strategy [26]. The first works can be traced back to 1993 where Tackett [55] applied GP for solving an object detection task. Then, Johnson et al. [23] evolved visual routines which were able to locate simple salient patterns from silhouettes of people. Also, Teller and Veloso in [56] developed PADO as a probe of concept that shows GP as a suitable paradigm for building a whole recognition system. Later, Poli [49] introduced GP into the classical problem of image segmentation. In 1999, Ebner [14] introduced a technique based on GP for evolving a task specific image operator, in particular to reproduce the well-known Moravec interest point detector. Also, in 1999

*Corresponding author: Gustavo Olague, CICESE Research Center, Centro de Investigación Científica y de Educación Superior de Ensenada, Km. 107 Carretera Tijuana-Ensenada, 22860, Ensenada, B.C., México. E-mail: olague@cicese.mx.

Howard et al. [22] described a GP approach, where binary classifiers were evolved with the goal of detecting targets within SAR imagery. Later, Zhang et al. [63] used GP to perform multi-class detection of small objects present in large images, with domain independent pixel statistics as the terminal sets. Then, Lin and Bhanu [31] performed object detection from SAR images through a GP-based approach using a cooperative coevolutionary framework; where simple features were combined in such a way that novel features are said to be synthesized. Hernández et al. [21] presented a linear-GP approach coupled with a support vector machine as a visual learning strategy for facial expression recognition. Recently, Song and Ciesielski [53] showed that a fast and accurate texture segmentation method can be developed based on classifiers evolved by GP. Trujillo and Olague [59] described a GP methodology that can be used to synthesize low-level image operators that detect interest points on digital images. Also, Olague and Trujillo [45] extended their work presenting an innovative method for designing interest point detectors using an evolutionary-computer-assisted design approach. Finally, Fu et al. [16] developed a GP-based method for edge detection in images. They evolved program detectors using different fitness functions, from which result different new models that compete with traditional window-based edge detectors. All above works describe methods for solving image related tasks, which are based on the direct evolution of programs or algorithms using a kind of inductive learning. Our work follows such GP-based methodology where tree structures, mathematical operators, are evolved in order to enhance the overall performance of local descriptors. Also, we introduce another optimization algorithm based on a hill-climber method with multiple re-starts using the same search space (functions, terminals), fitness function and mutation used in the GP algorithm. The idea is to compare our GP results not only with man-made algorithms but also, with the results of another search method since there is no optimization algorithm in the literature that learns descriptor operators beside our GP proposal described in this paper.

Research on local image descriptors has received a lot of attention in recent years due to its many advantages. For example, local features can be designed to be very tolerant to geometric and photometric deformations, occlusions, as well as to promote its distinctiveness. The basic idea is to detect interest points or regions that are covariant to a class of transformation. Afterwards, for each detected image patch, an invariant descriptor is computed. The objective of a local descriptor is to obtain a compact and complete feature description that captures numerically the information about the local structure around each interest point. In this way, the descriptors' vectors can be matched between different images exhibiting image deformations caused by changes in camera pose, brightness, and others such as: rotation, scaling, illumination changes, JPEG compression, affine transformation, eyefish lens deformation, and nonrigid deformation, to mention but a few. There exist several comparative studies on region descriptors [9,41,42]. In general, the best results are reported for methods or algorithms that use histograms to represent different characteristics of appearance or shape, like the Scale Invariant Feature Transform (SIFT) [36]. In particular, the SIFT descriptor is considered as the state-of-the-art of distribution-based descriptors.

1.1. Motivation and problem statement

The SIFT descriptor is a 3D histogram of gradient locations and orientations, where a scale-normalized image region is represented through the concatenation of gradient orientation histograms relative to several rectangular subregions, resulting in a 128-dimensional descriptor. First, the gradient magnitudes and orientations are computed within the interest region. The gradient magnitudes are then weighted with a Gaussian window overlaid over the region. In order to detect the scale-normalized patches, a salient region detection procedure is executed using as the saliency function, the scale-space Difference

of Gaussians (DoG), and the image regions (positions and scale) are selected by the local extrema at DoG. Thus, the regions are scale normalized to compute the derivatives of the image through pixel differences, which are used to calculate the image gradient and orientation for every pixel in the image. This information is then divided in subregions as in a rectangular grid in order to compute the histogram of gradient orientation, weighted by gradient magnitude, for each subregion. The SIFT local descriptor is the concatenation of the several gradient orientation histograms for all subregions. The gradient magnitude has been criticized [32,33] specially if the gradient computation is performed through pixel differences, which are very sensitive to noisy measurements. In this work, we are interested to test if the gradient magnitude based feature used within the SIFT algorithm could be replaced by a different feature, operator, that can offer a better or comparable performance. The idea is to propose a GP methodology that synthesizes mathematical operators which will be used instead of the gradient magnitude. A major difference with previous research is that such local features have been designed by human experts using traditional representations that have a clear and preferably mathematically well-founded definition. In this paper, we will show how a powerful machine learning strategy is able to create reliable composite operators that enhance the overall performance of the best available local descriptor. We will use the acronym RDGP (Region Descriptor Operators with Genetic Programming) to refer to our evolved SIFT operators using the GP and HILL for the optimized operators using the hill-climbing algorithm. Moreover, we will observe in the experiments that this simple change into the SIFT descriptor has resulted in a set of features that significantly outperform other state-of-the-art descriptors. Two main aspects will be addressed in the paper: the selection of a set of mathematical functions and genetic operations that are useful to define and explore the search space, as well as the fitness function that is used to measure the performance of local descriptors.

1.2. Research contributions

This paper describes an optimization based approach using GP and a hill-climbing algorithm as a machine learning strategy, which create composite image operators for improving the SIFT descriptor. This document enhances preliminary results using a new search method based on a greedy hill-climber and provides a clear explanation of our GP proposal [46–48]. In this regard, Perez and Olague, [46,47] briefly introduced the idea of using GP for learning SIFT descriptor operators using rotation and scale information during the learning stage; while in [48] we included illumination information in the learning stage due to the SIFT algorithm is partially invariant to illumination changes.

The main contributions of our work are the following:

1. This work proposes a GP strategy to automate the design of SIFT descriptor's operators.
2. This work provides an analysis about the descriptor operators of several state-of-the-art local descriptors.
3. We propose a quantitative criteria based on the F-Measure for evaluating local descriptors.
4. We introduce a new search method based on a greedy hill-climbing algorithm with multiple restarts for learning SIFT descriptor operators.
5. We introduce $RDGP_2$ and $HILL$ operators which outperform a set of state-of-art descriptors.
6. The usage of $RDGP_2$ and $HILL$ as a descriptor in an object recognition task produce fewer outliers.

The structure of this paper is as follows. In Section 2 we present an overview of the related work on SIFT-like and other local descriptors in order to understand the relevance of our contribution. In Section 3 we describe the necessary steps to formulate the optimization learning strategy for local descriptors through the F-measure and all building blocks of the algorithm. Section 4 presents the results of learning composite image operators with the testbed and an object recognition experiment. Finally, we conclude the paper in Section 5.

2. Related work on local descriptors

Today, many researchers have a special interest in SIFT descriptor due to its ability to detect objects under different viewing conditions; for that reason, they have improved the SIFT algorithm in different ways with different purposes [1,6,13,20,24,28,35,40,44,54,57] adding new information or changing its original histogram representation. The idea in this paper is to replace the gradient magnitude by a new mathematical operation applied to the image region that can outperform all other descriptors using a standard protocol. Such mathematical operation will be developed through GP learning optimization and a Hill-climbing algorithm with the goal of enhancing the information content which is one of the most important properties of SIFT-like descriptors. Indeed, the invariance of the information content determines the distinctiveness and robustness of local descriptors.

Table 1 presents an overview of some of the most important state-of-the-art descriptors. This table summarizes works directly derived from SIFT, which are marked with \checkmark and those that are based on similar concepts as the histogram or when we can identify a possible operator \times . All local descriptors presented in this table are arranged by its descriptor operator, data organization, vector dimension, and the kind of application where they were tested.

Ke and Sukthankar [24] introduced a descriptor named PCA-SIFT that concatenates the first order derivatives of every subregion into a gradient magnitude as in SIFT; however, its originality is based on reducing the feature vector dimension through PCA (Principal Component Analysis) data selection. Lazebnik et al. [28], proposed the Rotation Invariant Feature Transform (RIFT) that generalizes the SIFT descriptor using concentric rings from a circular normalized region, and its data is organized as a 2D histogram; like SIFT, the gradient magnitude can be identified as the RIFT's descriptor operator. As an application they use the RIFT descriptor for texture image retrieval and classification. Mikolajczyk and Schmid [41] proposed the Gradient Location and Orientation Histogram (GLOH) which uses polar coordinates instead of cartesian coordinates to organize the SIFT 3D histogram; again, the descriptor operator is based on the gradient magnitude like in SIFT. Stein and Hebert [54] introduced the Background-SIFT algorithm coined BSIFT, which incorporates background invariance into the SIFT algorithm using heat diffusion theory for the detection and description processes which could be identified as the base of its descriptor operator; this descriptor is used to detect objects on synthetic and natural images. Mortensen et al. [44], presented SIFT+GC that augments SIFT descriptor with a global context adding shape information in the description process; its operator and data organization are implemented like in the SIFT descriptor using the gradient magnitude, as well as 3D localization and orientation histogram. Abdel-Hakim and Farag [1] proposed the Colour-SIFT algorithm (CSIFT) which adds color invariance to the SIFT descriptor process using a Gaussian RGB color model from which we identify the color-based gradient magnitude as their descriptor operator and a 3D histogram as the way to organize the final information. Bay et al. [20] presented the SURF algorithm (Speed Up Robust Features) which was inspired from SIFT algorithm. They outperformed the computational time using integral images, a fast-hessian detector and Haar-wavelet responses to integrate the vector descriptor.

Dalal and Triggs [13] described a SIFT-like algorithm named Histogram of Gradient Orientation (HOG) descriptor, which is similar on capturing the gradient structure in a dense overlapping grid using cell-blocks for the problem of human detection. Bosch et al. [6] proposed the Pyramid Histogram of Gradient Orientation (PHOG) descriptor, inspired on the image pyramid representation of Lazebnik et al. [30] and the HOG descriptor [13]; they organized the feature information into 2D histograms applying PHOG descriptors for image classification. Tola et al. [57], introduced a local descriptor coined DAISY for dense wide-baseline matching inspired on SIFT and GLOH descriptors. They replaced the weighted

Table 1
Overview of local descriptors

Descriptor	Operator	Data organization	VD	Tested in	IoS
SIFT [36]	$\ \nabla\phi\ = \sqrt{\phi_x^2 + \phi_y^2}$	3D Histogram Cartesian Coord.	128	Object Recognition	✓
PCA-SIFT [24]	$\ \nabla\phi\ = \sqrt{\phi_x^2 + \phi_y^2}$	Directly 2D Gradient	36	Image Retrieval Texture Image Retrieval	✓
RIFT [28]	$\ \nabla\phi\ = \sqrt{\phi_x^2 + \phi_y^2}$	Orientation Histogram	32	and Classification [28] Butterflies Recognition [29]	✓
GLOH [41]	$\ \nabla\phi\ = \sqrt{\phi_x^2 + \phi_y^2}$	3D Histogram Polar Coord.	128	–	✓
BSIFT [54]	$I^{(k+1)}(x, y) = I^{(k)}(x, y) + \tau \nabla^2 I^{(k)}(x, y)$	3D Histogram	128	Object Recognition	✓
SIFT-GC [44]	$\ \nabla\phi\ = \sqrt{\phi_x^2 + \phi_y^2}$	3D Hist + 2D Hist Cart. + Log Polar	128+60	Insect Recognition	✓
CSIFT [1]	$\hat{E}_{\lambda^i w} = \sqrt{\hat{E}_{\lambda^i x}^2 + \hat{E}_{\lambda^i y}^2}$	3D Histogram	128	–	✓
SURF [20]	Haar Wavelet * G_σ $v = (\sum dx, \sum dy, \sum dx , \sum dy)$	4x4 Oriented Square Subregions	64	3D Reconstruction [20] Museum Obj. Recog. [4]	✓
HOG [13]	$\ \nabla\phi\ = \sqrt{\phi_x^2 + \phi_y^2}$	Cell-Blocks	–	Human Detection	✓
PHOG [6]	$\ \nabla\phi\ = \sqrt{\phi_x^2 + \phi_y^2}$	Histograms	$K \sum_{l \in L} 4^l$	Image Classification	✓
DAISY [57]	G_Σ from $G_\sigma^\Sigma = G_\Sigma * \left(\frac{\partial I}{\partial o}\right)^+$	Orientation Maps	200	3D Reconstruction	✓
SIFT-Flow [35]	$\ \nabla\phi\ = \sqrt{\phi_x^2 + \phi_y^2}$	3D Histogram	128	Motion Prediction and Transfer	✓
SIFT-Gabor [43]	Gradient through Gabor Filters	Histograms	–	Object Recognition	✓
$RDGP_2$ [46,47]	$\sqrt{D_x \left(\sqrt{\sqrt{D_{xy}(I)} - D_{xx}(I)} \right)}$	3D Histogram	128	Object Recognition	✓
Steerable filters [15]	$E_n(\theta) = [G_n^\theta]^2 + [H_n^\theta]^2$	Directly	–	Texture Analysis	×
HTD-TBD-EHD [38]	Bank of filters	Histogram	–	Texture Classification	×
SCD-CSD-CLD-DCD	HSV, HMMD & YCrCb			Image Retrieval	
SASI [10]	$\mu_{S,L}^{\bar{p},\bar{c}}(n), \sigma_{S,L}^{\bar{p},\bar{c}}(n)$	Clique Windows	2xQ	Image Retrieval	×
GIH [34]	$I(x,y)$	Geodesic-Intensity Histogram		Synthetic Deformation Non-affine Deformation	×
MOPs [7]	Haar Wavelet Coefficients	8x8 Patch Pixels	64	Panoramic Image Stitching	×
Wiccest [17]	$\hat{E}_{k1m}(x, y, \sigma_i) = G_{kl}(x, y, \sigma_i) * E_m(x, y)$	RGB Histogram	12x3	Object Recognition	×
Hölder [60]	$\alpha_p = \sup_s \{f \in C^s(x_o)\}$	Concentric Rings	129	Image Matching	×
LESH [52]	Local Energy Model	2D Histogram	128	Face Recognition	×
SMD [19]	$I(x,y)$	Stable Pairs	–	Image Matching	×
DLID [12]	$\ \nabla\phi\ = \sqrt{\phi_x^2 + \phi_y^2}$	2D Histograms	$L = L_1 \times L_2$	Image Deformations	×
WLD [11]	$\xi(I_c) = \arctan \left[\sum_{i=0}^{p-1} \left[\frac{I_i - I_c}{I_c} \right] \right]$	2D Histograms	–	Texture Classification Face Detection	×

VD: Vector Dimension; IoS: Inspired on SIFT.

sums of gradient norms by convolutions of the original image with several derivatives of Gaussian filters referring to the convolution results as convolved orientation maps. Hence, the descriptor operator for this descriptor could be related to those Gaussian filters. In particular, DAISY produces good 3D reconstructions. Liu et al. [35] proposed the SIFT-flow algorithm which consists of aligning images of complex scenes in order to achieve good dense matching; like in SIFT, they used the gradient magnitude and a 3D histogram in their algorithm. Moreover, SIFT-flow is applied to motion prediction from a single image and motion synthesis via transfer of moving objects. Finally, SIFT-Gabor proposed by Moreno et al. [43] performs an analysis of gradient information based on odd Gabor functions, whose parameters are selectively tuned instead of the computation of image derivatives with pixel difference methods; ex-

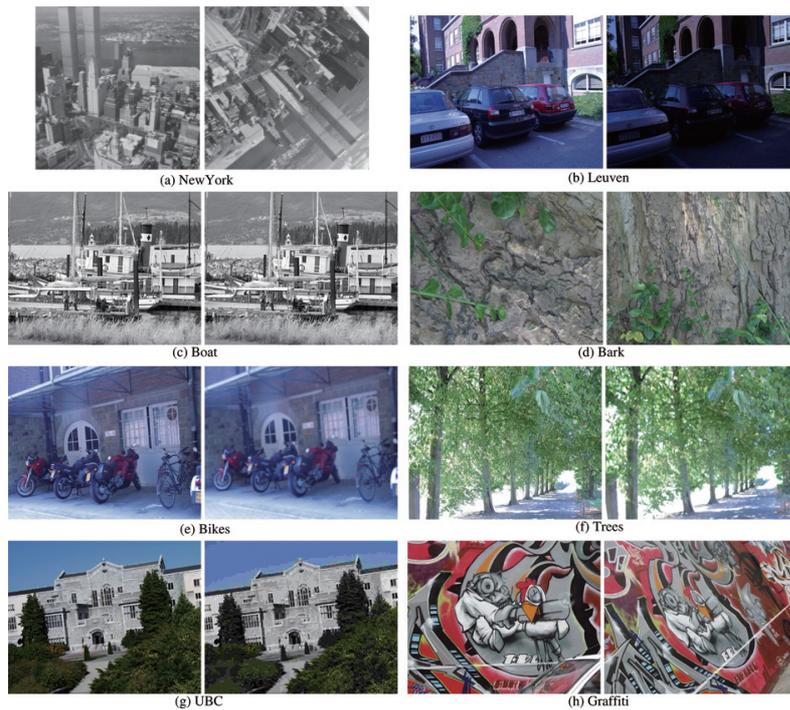


Fig. 1. Dataset which includes different image transformations. (a) Rotation; (b) Illumination; (c)&(d) Rotation + Scale; (e)&(f) Image blur; (g) JPEG Compression; (h) Affine transformation. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130594>)

periments on image matching and object detection are provided to illustrate the distinctiveness of their proposal.

On the other hand, there are others local descriptors not inspired directly on SIFT but where it is possible to identify the operator and data organization that is used to incorporate geometric and photometric invariance into the vector descriptor through different well founded theories. However, the descriptor operators for some of these works could be even more difficult to identify due to the nature of their description process. For instance, Freeman and Adelson [15] proposed steerable filters based on Gaussian derivatives which could be the base of the descriptor operator. Steerable filters, used as local descriptors, are applied to the image region in order to produce the descriptor vector which resulted from using directly the outputs of the filters. Manjunath et al. [38] presented color and texture descriptors based on the MPEG-7 standard applied to image retrieval and texture classification. Color descriptors are formed using histograms or dominant color information based on HSV, HMMD, YCrCb colour spaces. The histogram descriptors capture the global distribution of color whereas the dominant color descriptor is represented in a compact representation. Such texture descriptors use a bank of filters in a color space that could be identified as their descriptor operator.

Çarkacıoglu and Yarman-Vural [10] proposed a generic texture descriptor for image retrieval named SASI (Statistical Analysis of Structural Information) which is based on statistics of clique autocorrelation coefficients. The SASI's descriptor operator could be defined by the process required to obtain the clique window. Hence, the SASI's operator could be related with the computation of second order statistics from the autocorrelation coefficients μ and σ defined on their paper. Ling and Jacobs [34] proposed the Geodesic-Intensity Histogram (GIH) descriptor which captures invariance to synthetic deformation

and real non-affine deformation from the joint distribution of geodesic distances and intensity of sample points. Intensities values are considered as the GIH descriptor operators organized into 2D histograms. Geusebroek [17] proposed a compact colour descriptor named Wiccest that captures colour and texture information based on local histogram of colour edges. The Wiccest's descriptor operator $\hat{E}_{klm}(x, y, \sigma_i)$ is related with the convolution of opponent colour channels $E, E_\lambda, E_{\lambda\lambda}$, see [18], with a Gaussian filter $G_{kl}(x, y, \sigma_i)$. Trujillo et al. [60] proposed a local descriptor based on Hölderian regularity that is the base for their descriptor operator. Hölder descriptor organizes the region information derived from the pointwise signal regularity into four concentric rings obtaining a feature vector of 129 dimensions. Sarfraz and Hellwich [52] presented the LESH (Local Energy based Shape Histogram) descriptor used for face recognition. They generated the LESH descriptor by concatenating local histograms that accumulate the local energy (descriptor operator) along each filter orientation for different subregions of the image. Gupta and Mittal [19] presented the Stable Monotonic Change Invariant Feature Descriptor (SMD) which is invariant to a monotonic change in intensities and robust to Gaussian noise. Hence, the intensities values are considered as the SMD's descriptor operator. Moreover, they formed the vector descriptor computing intensity differences of points pairs. Cheng et al. [12] proposed a Deformable Local Image Descriptor (DLID) which is robust to general deformations such as: fisheye lens, nonrigid, affine and synthetic deformations. They used multiple support regions of different sizes for each interest point; then, they computed a histogram of gradient directions for each support region. Thus, the DLID's descriptor operator is the gradient magnitude obtained from the description process. Finally, Chen et al. [11] proposed the Weber Local Descriptor (WLD) based on Weber's Law theory for texture classification and face detection. The WLD's descriptor operator could be defined as the differential excitation $\xi(I_c)$ which uses the intensity differences between the neighbors of the current pixel.

3. Optimization of SIFT descriptor operators

3.1. Optimization using genetic programming

Genetic programming has become the main machine learning paradigm of evolutionary computation which was formalized by John Koza [26] in the early 1990s. This approach is evolving rapidly in the computer science community, where Poli et al. [50] presented a modern field guide to GP with the aim to provide a comprehensive review for beginners and experts. GP is an offshoot of genetic algorithms that automatically solves problems without requiring the designer to know or specify the form or structure of the solution in advance. GP is said to be a systematic, domain independent method; however, at least for real world applications the user usually needs to define the way in which GP evolves programs (expressions or formulae) often in a domain-specific language through a user-defined task.

GP is inspired from the principles of biological evolution and it is used to create programs that learn a user-defined function. Figure 2 shows our GP approach used to synthesize mathematical operators that optimize the SIFT local image descriptor. GP is a machine learning approach that looks into the search space of all possible computer programs composed of functions and terminals appropriate to the problem domain. GP learning, as in common supervised machine learning, is divided in two stages of training and testing. The GP evolution is usually performed by individuals encoded with a tree representation grouped in what is called a population. GP simulates natural evolution as a random process and therefore it can never guarantee results. GP's essential randomness, however, can lead it to escape traps which challenge deterministic methods. Moreover, GP has been very successful at evolving novel and unexpected ways of solving problems. GP starts with an initial population of randomly generated

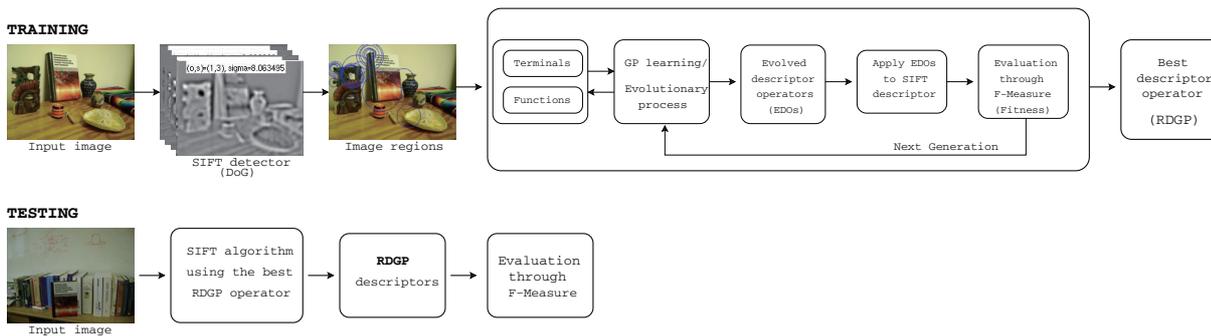


Fig. 2. Overview of our GP approach to local image descriptors. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130594>)

programs. Each individual computer program in the population is measured in terms of how well it performs in the particular problem environment. This measure is called the fitness function. The computer programs in the initial generation of the process will generally have poor fitness. Nonetheless, some individuals in the population will turn out to be somewhat more fit than others. The Darwinian principle of reproduction and survival of the fittest, as well as the genetic operations of crossover and mutation are used to create a new offspring population of individual computer programs from the current population of programs.

In this section, we describe the three major preparatory steps for applying GP to the evolution of SIFT operators; first, the definition of the set of terminals and primitive functions, as well as the way of combining them to find a solution; second, the fitness measure; and third, the parameters for controlling the algorithm and the criterion for terminating a run.

3.2. Representations, search space and genetic operations

In order to define the structure representation that will be evolved through GP learning we have reviewed the main aspects of SIFT. The idea is not to improve the whole algorithmic process; a task beyond the capacity of genetic programming, but to identify a key aspect that could enhance the overall quality performance of SIFT descriptors. SIFT consists of four stages and the local image descriptors are computed in the last stage. Such description of local information is based upon image gradients of a patch of pixels situated within a local neighborhood. This patch is centered on the key point location, rotated with respect to the dominant orientation, and scaled to the appropriate size. We propose in this work to replace the gradient magnitude, normally used in SIFT and SIFT-like proposals, with a new GP-evolved feature. Note, that all mathematical operators being evolved by the GP learning system are tested on thousands of small image patches computed during the key point localization stage.

In GP, programs are usually encoded as syntax trees 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 . Thus, the sets of functions and terminals represent the problem search space. Historically, differential operators have been used through a set of image derivatives computed up to a given order to describe the properties of a point neighborhood. The properties of local derivatives (local jet) were investigated by Koenderink and Van Doorn [25] and later produced a number of approaches such as: steerable filters and differential invariants that compute derivatives by convolution with Gaussian derivatives. We decide to use such ideas to establish our functional and terminal sets as follows:

$$\begin{aligned}
F &= \left\{ +, | + |, -, | - |, *, \div, \sqrt{I_t}, \right. \\
&\left. \frac{I_t}{2}, \log_2(I_t), D_x G_\sigma, D_y G_\sigma, G_\sigma \right\} \\
T &= \{I, D_x(I), D_y I, D_{xx}(I), D_{yy}(I), D_{xy}(I)\}
\end{aligned} \tag{1}$$

where I is the input image and I_t can be any of the terminals in T , as well as the output of any of the functions in F ; D_u symbolizes Gaussian image derivatives along direction u then $D_u(I) = I * G'_{u(\sigma=1)}$; G_σ are Gaussian smoothing filters with $\sigma = 1$ or 2 ; $D_u G_\sigma$ represents the derivative of a Gaussian filter with image blur σ . These sets ensure that the property of closure is achieved because all terminals and functions are defined as images in accordance with type consistency and evaluation safety. On the other hand, the property of sufficiency is not guaranteed. Our goal is the synthesis of SIFT operators using a set of structurally complementary operators that are combined to produce structural or functional properties not present in any individual component. It is widely accepted that GP is able to create composite operators; hence, the choice of well selected functions and terminal sets are of paramount importance for the final result. We are interested in obtaining synthetic operators that could be simple in structure, and at the same time each operator should be able to improve the overall performance of SIFT.

After the initial generation is created and the fitness has been assigned to each individual, GP probabilistically selects the best individuals from the population for breeding the parents to create the next generation using the genetic operations of selection, mutation and crossover. The selection function is responsible of choosing the best individuals for reproduction through mutation and crossover. The 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 crossover method needs a pair of selected parents to perform its genetic operation; first, one point of each parent is randomly selected as the crossover point; then, the subtrees are combined in order to create the new child.

3.3. Fitness function

The aim of our research is to show that GP is a powerful methodology that it is capable of improving local descriptors that later could be used for object recognition tasks. In general, to apply evolutionary computation we need to devise a well posed fitness function together with the representation of the problem. Today, the evaluation techniques described in the literature propose metrics related to matching descriptors that mostly work on the ROC (Receiver Operating Characteristic) space [8,11,34,40] as well as on the Recall vs 1-Precision space [12,19,20,24,41,43,60]. The appropriate use for each technique depends on the criteria for comparing local descriptors. Hence, the Recall vs 1-Precision technique is used for evaluating descriptors extracted from image pairs while ROC analysis is used in the context of image classification or retrieval from databases. Therefore, the Recall vs 1-Precision curves are better-suited for evaluating detection systems since it is not necessarily to predict true negatives for a given image pair. Thus, Agarwal et al. [2] stated that Recall vs 1-Precision curves are more appropriate than ROC curves for measuring the performance of object detection approaches.

In this work, we propose to use the F-measure as the fitness function in order to compare several descriptors since it is based on recall and precision metrics, see Fig. 3. Thus, the recall and precision metrics are obtained according to the testbed proposed by Mikolajczyk and Schmid [41] for evaluating local descriptors. This testbed is freely available through internet¹ and it is based on the number of correct and

¹<http://www.robots.ox.ac.uk/vgg/research/affine/>.

		TRUE CONDITION		
		Correct Matches	Not Correct Matches	
PREDICTED	Detected	True Positive (TP) "Correct Matches Detected"	False Positive (FP) "Not Correct Matches Detected" ERROR TYPE I	$Recall = \frac{TP}{TP + FN}$
	Not Detected	False Negative (FN) "Correct Matches Not Detected" ERROR TYPE II	True Negative (TN) "Not Correct Matches Not Detected"	$Precision = \frac{TP}{TP + FP}$ $False\ Positive\ Rate = \frac{FP}{FP + TN}$ $1 - Precision = \frac{FP}{FP + TP}$

Fig. 3. Contingency table related to matching features. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130594>)

false matches obtained from an image pair (I_A, I_B) . Recall represents the number of correctly matched regions (TP) with respect to the number of corresponding regions (TP+FN) between two images of the same scene using an overlap error. This overlap error measures how well the regions correspond under a homographic transformation according to the ratio of intersection and union of two regions A and B . Hence, authors assume that a match is correct if there is a 50% of overlapping between regions and the distance between both descriptors is below a threshold. Finally, they compute 1-Precision considering the number of false matched regions (FP) with respect to the total number of matches (FP+TP) while we consider Precision due to the formulation of the F-Measure; thus, Precision is obtained from 1-Precision just computing $1 - (1 - Precision)$.

The F-Measure was originally proposed in the information retrieval community by Van Rijsbergen [61] as the Effectiveness function (E). The F-measure is based on the harmonic mean giving the best balance between Precision and Recall metrics and it is widely used as a performance evaluation criterion. Recently, it has been used in some computer vision applications such as object detection [2], visual surveillance for motion detection [27] and image segmentation [3,39]. In fact, our work is the first implementation of such a function for evaluating local descriptors [46,47]. The idea is to obtain a quantitative measurement for evaluating local descriptors in the Recall vs 1-Precision space. It is true that other criteria could be used for comparing quantitatively local descriptors such as the true negative rate, true positive rate, weighted accuracy, G-Mean, precision and recall to mention but a few. Indeed, there is only one previous work that used the area under the ROC curve as the merit function for attempting to improve local descriptors using statistical learning [62]; however, as was shown by [2,41] it is not suitable to use that criterion for the evaluation of local descriptors in the context of object recognition. Thus, the general formula is defined by the following equation:

$$F_{\alpha}(p, r) = \frac{(1 + \alpha) \cdot (p \cdot r)}{(\alpha \cdot p + r)}, \quad (2)$$

where p is Precision $\{p : 0 \leq p \leq 1\}$, r is Recall $\{r : 0 \leq r \leq 1\}$, and α is the parameter that controls the balance between p and r , $\{\alpha : 0 \leq \alpha \leq \infty\}$. Note that in the case of $\alpha < 1$ the variable with higher weight is p , while for the case of $\alpha > 1$ the variable with a higher weight is r , and when $\alpha = 1$ the precision and recall are well balanced.

Following the idea of using the F-Measure for learning local descriptors, we propose this equation as the fitness function for our GP system:

$$Q = \operatorname{argmax} \left\{ F_{\alpha}(P_i^x, R_i^x) = \sum_{i=1}^n \frac{(1 + \alpha) \cdot (p_i \cdot r_i)}{(\alpha \cdot p_i) + r_i} \right\} \quad (3)$$

where $Q: F_{\alpha}(P^s, R^s) \geq F_{\alpha}(P^t, R^t)$, with n representing the number of thresholds used in the testbed where $n = 20$ and the threshold vector used is [10,50,90,130,170,210,250,290,330,370,410,450,650,850,1050,1250,1450,1650,1850,2050] which was obtained from the testbed proposed by Mikolajczyk and Schmid [41]. Moreover, Precision data from an image pair is denoted by $P^x = (p_1, p_2, \dots, p_n)$ and Recall data by $R^x = (r_1, r_2, \dots, r_n)$ where x represents a possible solution and $s, t \subseteq x$. Hence, Q represents the ascendent's ranking order where the highest value corresponds to the best descriptor performance. Here, we claim that the F-Measure is a simple and reliable criterion that provides a significant evaluation for local descriptors.

3.4. Initialization, GP parameters and solution designation

Once we have defined the search space and the fitness function, the first step is to start the evolutionary process randomly. The initial population is created using the ramped half-and-half method proposed by Koza [26], which selects half of the individuals with the grow method and half with the full method to produce the population. The full method makes balanced trees according to the maximum initial depth while the grow method makes unbalanced trees allowing branches of varying length. Here, the size of individuals should not exceed a user specified maximum depth in order to avoid uncontrolling growing of trees over time; it helps to control bloat. The depth of a tree is defined as the length of the longest nonbacktracking path from the root to the endpoint. Tree depth is dynamically set using two maximum tree depths that limit the size of any given individual within the population, see Table 2. The dynamic max depth is a maximum tree depth that may be not surpassed by any individual unless its fitness is better than the best solution found so far. If 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 is a hard limit that no individual may surpass under any circumstances. Table 2 provides the GP runtime parameters used during the experimental tests. These parameters have canonical values that were set empirically after a number of tests. Selection is carried out using a tournament with lexicographic parsimony pressure, while keeping the best individual. We use the lexicographic parsimony pressure proposed by [37] which is a technique for controlling the bloat problem of genetic programming trees. They modify selection, preferring smaller trees only when fitness are equal in rank without affecting the fitness performance. Finally, the termination criteria was defined by a maximum number of generations; thus, the evolutionary process reaches an optimum operator for each single run.

3.5. Optimization using the hill-climbing algorithm

The hill-climbing optimization technique is an iterative algorithm that starts with a random solution to a problem, then, attempts to find better solutions by selecting the best choice in a greedy fashion until no further improvements can be found [51]. Unfortunately, hill-climbing often gets stuck in a local maxima since it continually moves in the direction of increasing value without look ahead beyond the immediate neighbors of the current state. Hence, one of the solutions is to include random re-starts when a local maximum is found. In this work, we use a hill-climbing algorithm with random re-starts with the

Table 2
RDGP algorithm set up

Parameters	Description
Generations	50
Population size	50 individuals
Initialization	Ramped Half-and-Half
Crossover	0.90
Mutation	0.10
Tree depth	Dynamic depth selection
Dynamic max depth	7 levels
Real max depth	9 levels
Selection	Stochastic Universal Sampling
Elitism	Keep the Best Individual, 1/50

aim to observe the results obtained by another optimization technique that produce SIFT operators. We use the same terminals and functions set, see Eq. (1), same fitness function, see Eq. (3), same number of candidate solutions and the same mutation procedure used in our GP algorithm. The Algorithm 1 shows the pseudocode of our hill-climbing algorithm where σ_0 represents a candidate solution (Random Parent); *score* means the fitness of ‘x’ candidate solutions; T, F are the terminals and functions set used in our GP algorithm; N is the total candidate solutions being evaluated ($N = 75000$) which is the same total number of individuals evaluated in our GP algorithm (50 generations, 50 individuals, 30 runs); x represents the number of candidate solutions ($x = 5$); *Mutation* is the procedure that generates ‘x’ candidate solutions through mutation; *Evaluate* is the procedure that evaluates ‘x’ candidate solutions as we do with each individual using the GP algorithm; *RandomC* is the procedure that creates a random candidate solution; finally, v is the variable used to establish the local maximum.

Algorithm 1 Greedy hill-climbing for searching SIFT operators

```

 $[\sigma_0, score(\sigma_0)] \leftarrow RandomC(F, T)$ 
while  $n \geq N$  do
   $\sigma \leftarrow Mutation(F, T, \sigma_0, x)$ 
   $score \leftarrow Evaluate(\sigma)$ 
   $[\sigma_{best}, score(\sigma_{best})] \leftarrow max(score)$ 
  if  $score(\sigma_{best}) > score(\sigma_0)$  then
     $\sigma_{best} \leftarrow \sigma$ 
     $score(\sigma_{best}) \leftarrow score(\sigma)$ 
  else
    if  $v > 10$  then
       $[\sigma_0, score(\sigma_0)] \leftarrow RandomC(F, T)$ 
       $v \leftarrow 0$ 
    else
       $v \leftarrow v + 1$ 
    end if
   $n \leftarrow n + 1$ 
end if
end while
return  $\sigma_{best}$ 

```

4. Results

This section presents two experimental frameworks used to evaluate the impact of our approach in the automated design of SIFT operators. The first experiment describes our two optimized algorithms that synthesize composite operators using the image region matching testbed [41]. The second experiment provides evidence that the two best optimized operators could improve significantly the SIFT algorithm by minimizing the total number of outliers in an object recognition task.

4.1. SIFT descriptor operators through GP and hill-climbing

GP learning and hill-climbing are performed using a matching protocol that is designed for matching thousands of interest regions between a pair of images. We compare the efficiency of our best GP image operator against three state-of-the-art descriptors: the original SIFT, GLOH and SURF and the best operator produced by our hill-climbing algorithm. Moreover, we compare also the efficacy of improving the descriptor's performance after applying different interest point detectors. The implementation for learning local descriptor operators through GP was programmed on Matlab with the GP toolbox GPLAB² while the core platform for SIFT features was programmed in Matlab/C.³ Also, the hill-climbing algorithm was programmed on Matlab and use the same SIFT algorithm. For training, we select the boat image pair, see Fig. 1(c), because this sequence presents rotation and scale changes. Later, we test the best evolved operator produced by GP, and the best operator produced by the hill-climbing algorithm against all other images presented in Fig. 1 showing that both best descriptor operators produce outstanding results.

The GP learning algorithm was executed 30 times using 50 generations and 50 individuals for each run; while the hill-climbing performed 75000 evaluations of candidate solutions for a fair comparison with the GP. Then, we obtained 30 best descriptor operators using the GP algorithm from which we selected the best descriptor operator coined $RDGP_2$ for further tests. Thus, each GP experiment takes about 24 hours for training while for testing depends on the SIFT run time that is about 6 seconds. Notice that the GP learning process does not limits the application to vision tasks since we only replace the gradient magnitude for an evolved mathematical expression into the SIFT algorithm. On the other hand, the hill-climbing took a similar time, that is about one month to produce the best result which we call *HILL* operator.

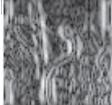
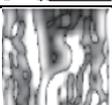
Table 3 shows the five best GP descriptor operators obtained from the 30 experiments and the five best descriptor operators produced by the hill-climbing algorithm. This table presents each solution in a prefix notation and its mathematical expression along with its fitness score. It also illustrates a corresponding image region produced by applying each $RDGP$ and *HILL* formula over a sample image region. We observed that $RDGP_1$ and $RDGP_2$ enhance with more details the original image, while $RDGP_4$ and $RDGP_5$ are quite similar to SIFT's weighted gradient, and the $RDGP_{11}$ is not human interpretable. On the other hand, the first three *HILL* operators shows a binarized example of the original image while the $HILL_4$ seems to be an abstract art in grayscale. Finally, the image that corresponds to the $HILL_5$ operator looks like a good segmentation of the object. Moreover, it is remarkable that most composite operators use the square root as one of their basic operations; this, is quite interesting because we see ageing this behaviour using the hill-climbing algorithm and such operation was never used in the synthesis of interest point detectors [58,59].

As an example of a typical GP run we present four statistical plots describing the process that synthesize the $RDGP_2$ operator. Figure 4 provides four plots illustrating the maximum, median, and average fitness; the population diversity; the evolution of the tree structural complexity; and finally, the variations of applying the crossover and mutation operations. Population diversity plots the percentage of unique individuals for each generation; while structural complexity shows the parameters related to tree size. In addition, Fig. 5 illustrates the fitness evolution and corresponding tree structure for $RDGP_1$, $RDGP_4$, $RDGP_5$ and $RDGP_{11}$ descriptor operators. The runs that produce $RDGP_2$, $RDGP_4$ and $RDGP_5$

²<http://gplab.sourceforge.net/index.html>.

³<http://vision.ucla.edu/vedaldi/code/sift/sift.html>.

Table 3
Final results of five best learned RDGP's and HILL's operators

Image region without any operation		Image region after SIFT's weighted gradient		
Descriptor	Fitness	Individual's Expression	Mathematical Expression	Image Region
				
$RDGP_1$	7.4158	$\text{sqrt}(\text{sqrt}(D_x(\text{sqrt}(D_x(\text{sqrt}(D_{xy}(\text{image})))))))$	$\sqrt{D_x(\sqrt{D_x(\sqrt{D_{xy}(I)})})}$	
$RDGP_2$	7.4859	$\text{sqrt}(D_x(\text{sqrt}(\text{subtract}(\text{sqrt}(D_{xy}(\text{image})), D_{xx}(\text{image}))))$	$\sqrt{D_x(\sqrt{\sqrt{D_{xy}(I)} - D_{xx}(I)})}$	
$RDGP_4$	7.3928	$\text{Gauss2}(\text{absdif}(\text{Gauss2}(\text{absdif}(\text{absdif}(D_y(\text{image})), D_x(D_x(\text{image}))), D_y(\text{Logarithm}(D_x(D_x(\text{image}))))), \text{Half}(D_x(D_x(\text{image}))))))$	$G_{\sigma=2} G_{\sigma=2}(D_y(I) - D_{xx}(I) - D_y(\log(D_{xx}(I)))) - \frac{D_{xx}(I)}{2} $	
$RDGP_5$	7.4053	$\text{Gauss1}(\text{sqrt}(\text{Gauss2}(\text{sqrt}(\text{sqrt}(\text{subtract}(\text{sqrt}(\text{Gauss1}(D_y(\text{image}))), \text{divide}(D_{xx}(\text{image}), \text{absadd}(D_x(\text{image}), D_y(\text{image}))))))))$	$G_{\sigma=1}\sqrt{G_{\sigma=2}\sqrt{G_{\sigma=1}(D_y(I)) - \frac{D_{xx}(I)}{ D_x(I)+D_y(I) }}}$	
$RDGP_{11}$	7.3736	$\text{Half}(G_2(G_2(\text{sqrt}(D_{xx}(\text{Log}(D_{xy}(\text{image})))))))$	$\frac{G_{\sigma=2}(G_{\sigma=2}\sqrt{(D_{xx}(\log_2(D_{xy}(I))))})}{2}$	
$HILL_1$	6.2854	$\text{add}(D_{xx}(\text{image}), \text{sqrt}(\text{sqrt}(D_x(\text{image}))))$	$D_{xx}(I) + \sqrt{\sqrt{D_x(I)}}$	
$HILL_2$	6.1779	$\text{add}(D_y(\text{image}), \text{sqrt}(\text{Gauss1}(\text{sqrt}(\text{Gauss2}(D_{xx}(\text{image}))))))$	$D_{yy}(I) + \sqrt{G_{\sigma=1}\sqrt{G_{\sigma=2}(D_{xx}(I))}}$	
$HILL_3$	6.1389	$\text{absdif}(D_y(\text{image}), \text{subtract}(D_{xy}(\text{image}), \text{Log}(\text{Half}(\text{Gauss1}(D_{xx}(\text{image}))))))$	$ D_y(I) - [D_{xy}(I) - \log(\frac{G_{\sigma=1}(D_{xx}(I))}{2})] $	
$HILL_4$	6.1345	$\text{sqrt}(\text{Half}(\text{sqrt}(\text{Gauss2}(D_{xx}(\text{image}))))))$	$\sqrt{\frac{G_{\sigma=2}(D_{xx}(I))}{2}}$	
$HILL_5$	6.1245	$\text{absadd}(\text{sqrt}(\text{Log}(\text{subtract}(D_{xx}(\text{image}), \text{Half}(D_y(\text{image}))))), D_y(\text{image}))$	$ \sqrt{\log(D_{xx}(I) - \frac{D_y(I)}{2})} + D_y(I) $	

reached their maximum fitness during the end, while those producing $RDGP_1$ and $RDGP_{11}$ found its best operator during the 40th generation. On the other hand, the five best $HILL$ operators were obtained at different moments along the algorithm execution and we observed that all best solutions at least they had the *sqrt* or *logarithm* function. Moreover, the hill-climbing algorithm, during the training, got in a local maximum about every 100 candidate solutions.

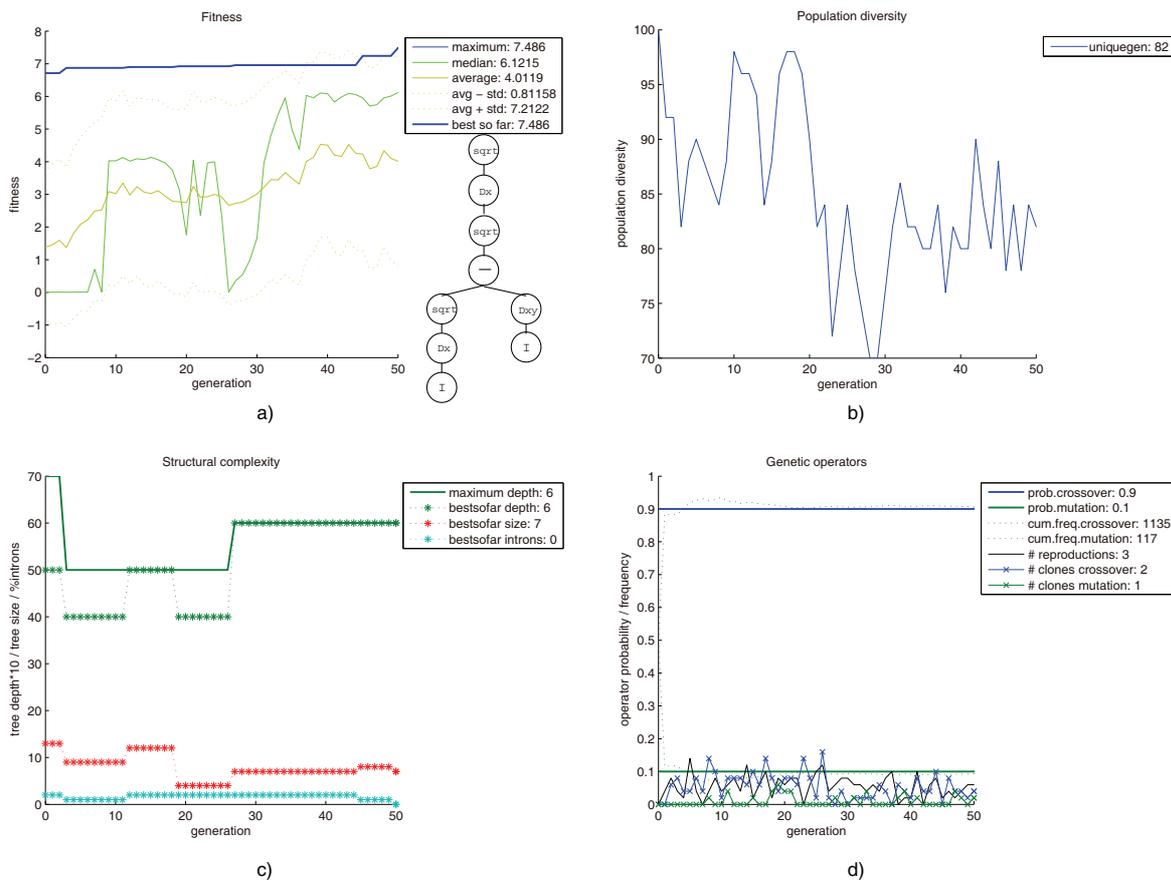


Fig. 4. These charts show the process of creating $RDGP_2$: a) Fitness plot and tree representation; b) Population diversity; c) Tree structural complexity; and d) Frequency of applying mutation and crossover. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130594>)

4.2. Experimental evaluation of local descriptors

The test of local descriptors consists on evaluating our proposed $SIFT-RDGP_2$ descriptor against our $SIFT-HILL_1$ descriptor and three state-of-art descriptors: $SIFT$, $GLOH$, and $SURF$; using several image transformations, as well as different detectors for a reliable comparison. The overall results are presented in Fig. 6; $GLOH$ and $SIFT$ were included in our tests because they are the two best descriptors according to an exhaustive experimental evaluation proposed by Mikolajczyk et al. [41]; while $SURF$ is considered as one of the fastest descriptor algorithms used for real world applications and its code is available and ready to be used in the testbed [20]. Moreover, other descriptors could be easily compared to our $SIFT-RDGP_2$ or $SIFT-HILL_1$ descriptor using this widely accepted testbed.

Figure 6(b) shows the results over the Bark images that have rotation + scale changes. Here, $GLOH-heslap$ improves significantly with respect to $GLOH-DOG$. It is possible that a similar improvement could be achieved in our descriptor if we change the detector. On the other hand, in a different image with a rotation transformation (Newyork), we observe that $GLOH-DOG$ is better than $SIFT-DOG$; however, $GLOH-hesaff$ scores a lower F-measure. In the case of illumination changes $GLOH-haraff$ and

Table 4
Summary of Descriptor's Performance Evaluation using the F-measure

Descriptor	NewYork	Leuven	Bark	Bikes	Trees	UBC	Graffiti
GLOH hessaff/haraff/heslap	4.3270	3.5459	13.5464	4.0373	1.3454	3.3758	4.1158
SIFT-harris/heslap/hesaff	5.6893	4.0065	6.2301	3.8642	2.2035	3.8000	3.6787
SIFT-DoG	5.2918	4.8373	7.0021	1.9075	0.4214	2.8942	0.4092
GLOH-DoG	6.0782	5.9319	7.6658	2.7105	0.6927	2.9981	0.6696
SURF	5.4379	5.1307	3.3079	4.7212	3.3130	5.0078	1.7021
RDGP ₂ -DoG	11.1728	8.1092	15.0334	5.6935	5.1599	6.1477	1.3371
HILL-DoG	10.9384	8.8381	14.6722	5.3765	5.1804	5.8921	1.3035
Difference (%) between the best man-made descriptor and the best optimized descriptor	45.59% or 83.82%	32.88% or 48.99%	9.89% or 10.98%	17.08% or 20.59%	36.05% or 56.37%	18.54% or 22.76%	67.51% or 207.81%
Difference (%) between the two optimized descriptors	2.14% or 2.18%	8.25% or 8.98%	2.40% or 2.46	5.57% or 5.89	0.39% or 0.39	4.15% or 4.33	2.51% or 2.57

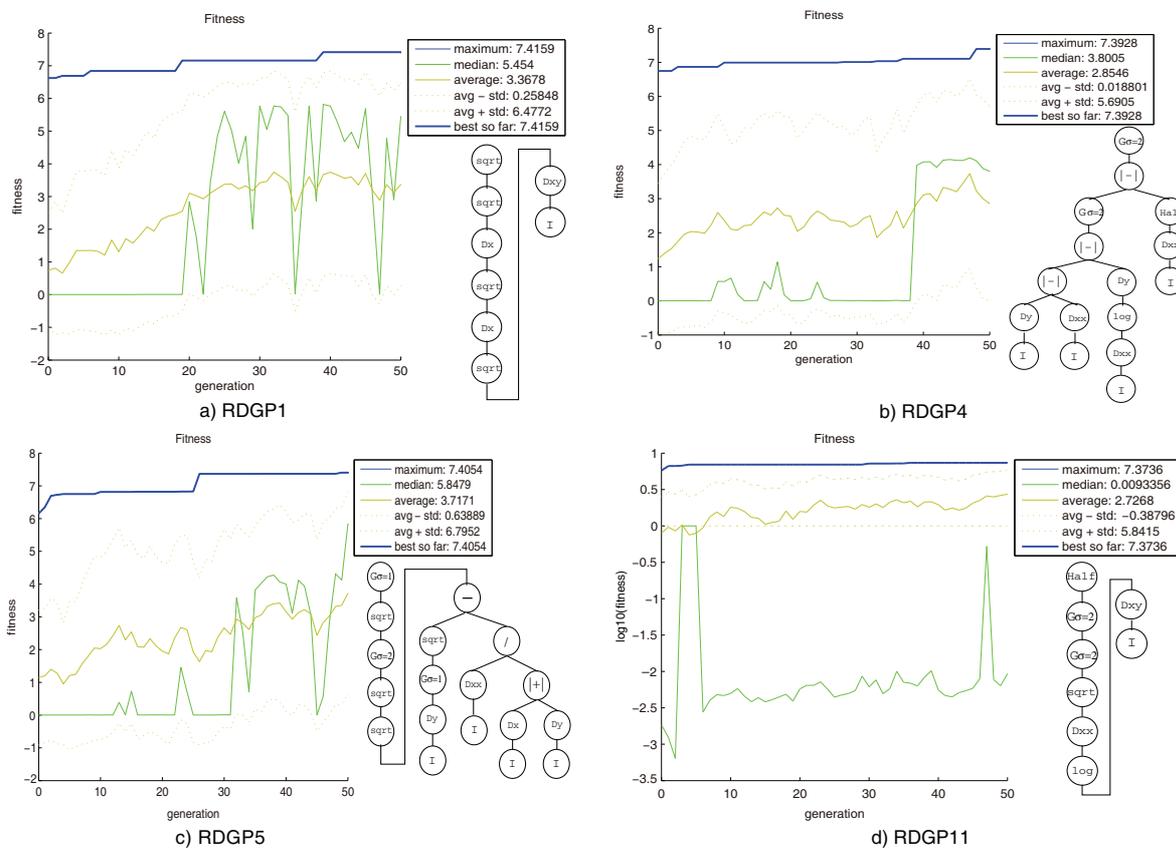


Fig. 5. This figure illustrates the RDGP₁, RDGP₄, RDGP₅ and RDGP₁₁ fitness plots with their corresponding tree structure. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130594>)

SIFT-hesaff scores lower values with respect to its corresponding versions using DOG. Thus, for this kind of transformations DOG represents a good choice.

Figures 6(e) and 6(f) present the results of the test considering blur changes. In Fig. 6(f) the hesaff and haraff detectors present a lower number of matches; however, the performance achieved by their

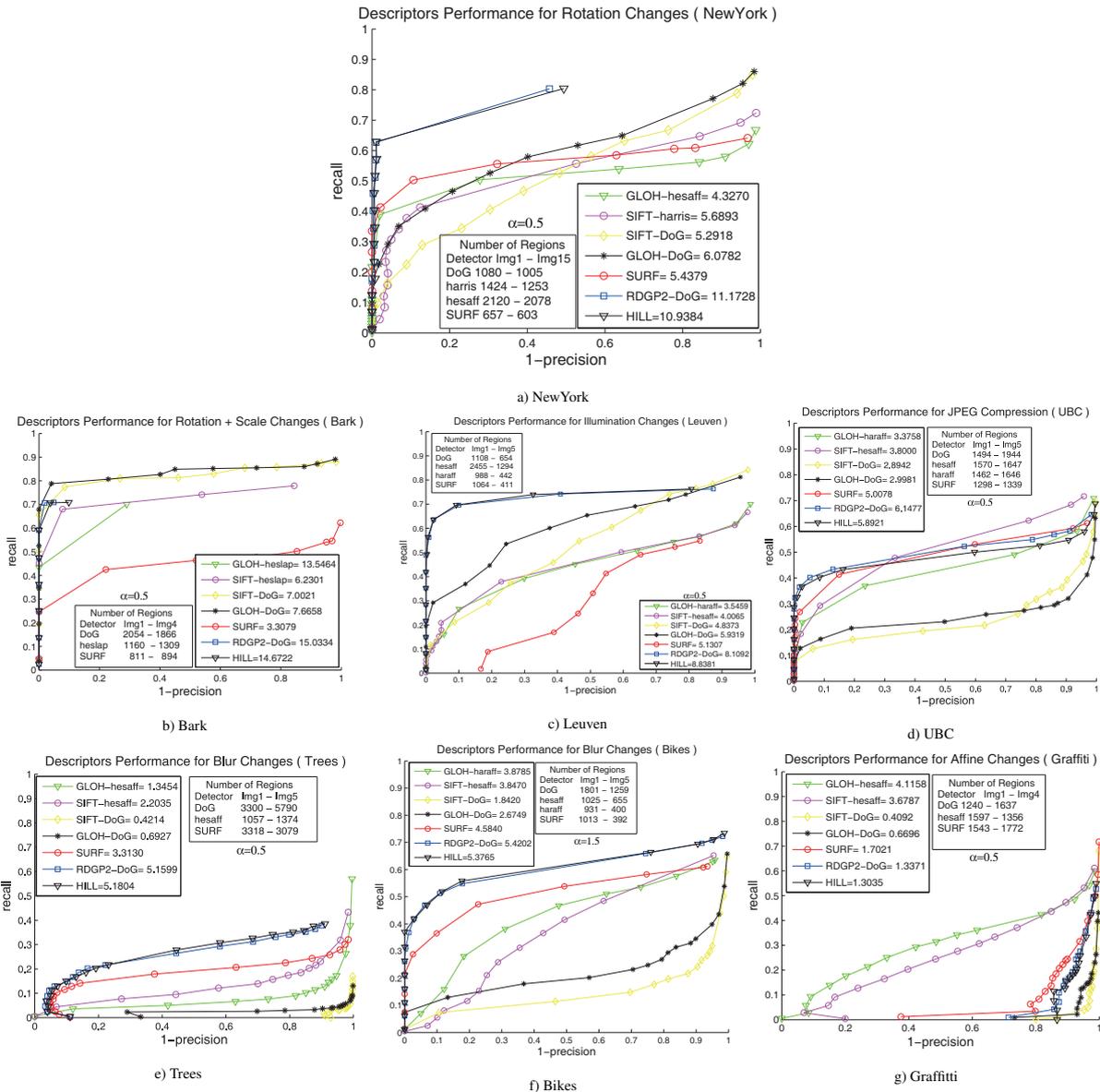


Fig. 6. Experimental results showing the best solution compared against several state-of-the-art descriptors and detectors using different image transformations. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130594>)

descriptors is higher than those using *DOG*. Figure 6(e) provides similar results with *SURF* as the third best descriptor. In the case of *JPEG* compression all detectors provide similar results for matching; however, *GLOH-haraff* and *SIFT-hesaff* score a higher F-measure than their corresponding *DOG* versions. *SURF* is again the third best descriptor. So far all tests showed that *SIFT-RDGP₂* and *SIFT-HILL* is by far the best descriptor. Nevertheless, we present the Graffiti images to illustrate results considering affine changes. Here, *GLOH-hesaff* and *SIFT-hesaff* achieves the higher scores; however, we remark that *RDGP₂-DOG* and *HILL* score higher than *SIFT-DOG* and *GLOH-DOG*. It is very likely that after using a more suitable detector the final results could be improved. On the other hand, we observed along these



Fig. 7. Images used for indoor and outdoor object recognition. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130594>)

results that both optimized descriptors have similar results where $SIFT-RDGP_2$ descriptor still being better.

Table 4 presents a summary of the descriptors performance using the F-Measure, where the best optimized descriptor improved about a 45.59% for rotation, 9.89% for rotation and scale, 32.88% for illumination, 26.56% for blur and 18.54% for JPEG compression; while for the affine transformation $GLOH$ and $SIFT$ were better due to their *hesaff* detector. In addition, the difference between both optimized descriptors was not significant.

4.3. Application to object recognition

In this section, we describe an object recognition application similar to the one proposed by Lowe [36]. The goal is to show the $RDGP_2$ and $HILL$ performances within a real-world application using outdoor and indoor scenarios. The test consists of a set of photographs acquired with a SONY Cyber-shot 12.1 MP DSC-W230 digital camera. Thus, we select two images, one that is considered as the base or object image and the second representing the scene where the object is placed with some image transformations. Later, the descriptors from the two images are computed; and then, they are matched using an efficient nearest neighbor indexing called Best Bin First proposed by Lowe and Beis [5]. We additionally compute the epipolar geometry using the RANSAC (Random Sample Consensus) algorithm in order to identify the inliers and outliers of the previous matched features.

For indoor scenarios, we have eight photographs from which three are objects and the rest are the scenes, see Fig. 7a). Moreover, Fig. 8 illustrates an example of the descriptor's matching, where green

Table 5
Matching Error between *SIFT-RDGP₂* and SIFT-gradient and SIFT-HILL descriptors for indoor scenarios

Descriptor	Detection stage		Matching stage (using descriptors)			
	Image object	Image scene	Total	Corrects	Incorrects	Error (%)
<i>Fig. 7a): I(a) & II(b)</i>						
<i>SIFT – RDGP₂</i>	1495	2124	76	44	32	42.11%
SIFT-gradient			142	53	89	62.68%
SIFT-HILL			70	42	28	40.00%
<i>Fig. 7a): I(a) & IV(a)</i>						
<i>SIFT – RDGP₂</i>	1495	3680	67	44	23	34.33%
SIFT-gradient			118	52	66	55.93%
SIFT-HILL			65	37	28	43.07%
<i>Fig. 7a): I(a) & III(b)</i>						
<i>SIFT – RDGP₂</i>	1495	6175	73	46	27	36.99%
SIFT-gradient			144	57	87	60.42%
SIFT-HILL			70	38	32	45.71%
<i>Fig. 7a): I(b) & III(a)</i>						
<i>SIFT – RDGP₂</i>	1619	3660	158	127	31	19.62%
SIFT-gradient			188	131	57	30.32%
SIFT-HILL			140	109	31	22.14%
<i>Fig. 7a): I(b) & IV(b)</i>						
<i>SIFT – RDGP₂</i>	1619	5350	62	39	23	37.10%
SIFT-gradient			107	57	50	46.73%
SIFT-HILL			54	32	22	40.74%
<i>Fig. 7a): II(a) & III(a)</i>						
<i>SIFT – RDGP₂</i>	2133	3660	48	22	26	54.17%
SIFT-gradient			124	30	94	75.81%
SIFT-HILL			47	21	26	55.31%

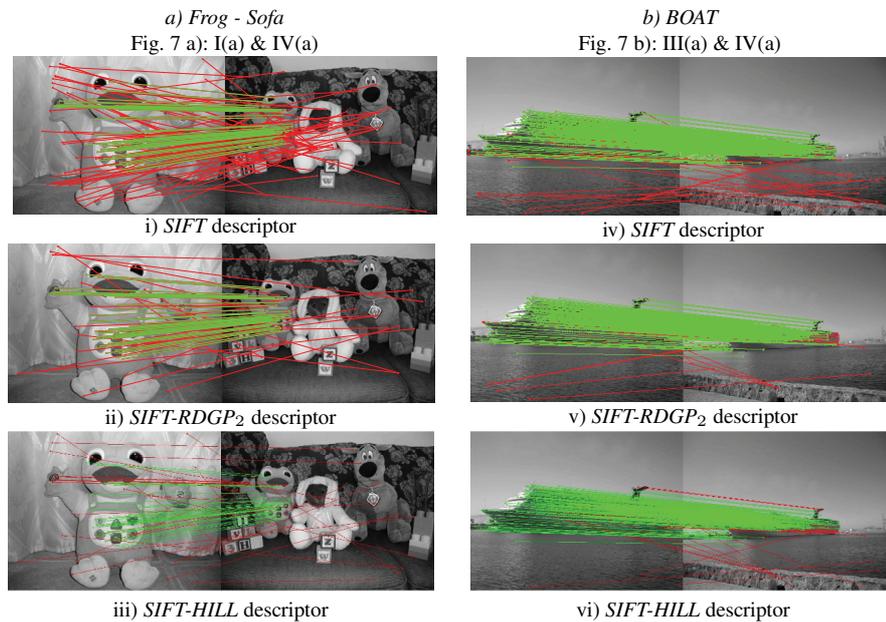


Fig. 8. Example of matching descriptors for indoor and outdoor scenarios. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130594>)

Table 6
Matching Error between *SIFT-RDGP*₂, *SIFT-gradient* and *SIFT-HILL* descriptors for outdoor scenarios

Descriptor	Matching error (%)				Detected regions	
	Average	Median	Min	Max	Img1	Img2 (average)
BOAT (52)						
<i>SIFT-RDGP</i> ₂	35.61%	28.53%	11.91%	84.26%	1806	1437
<i>SIFT-gradient</i>	41.70%	36.32%	15.12%	89.90%		
<i>SIFT-HILL</i>	35.82%	28.57%	12.10%	85.26%		
CEARTE (34)						
<i>SIFT-RDGP</i> ₂	32.64%	21.97%	6.45%	78.13%	924	1637
<i>SIFT-gradient</i>	42.05%	36.59%	10.91%	86.15%		
<i>SIFT-HILL</i>	33.39%	23.90%	7.45%	78.57%		
HUSSONGS (28)						
<i>SIFT-RDGP</i> ₂	43.38%	37.25%	12.50%	71.15%	2196	2536
<i>SIFT-gradient</i>	55.40%	54.06%	19.35%	82.49%		
<i>SIFT-HILL</i>	46.99%	39.15%	14.07%	87.65%		
PAPAS (36)						
<i>SIFT-RDGP</i> ₂	55.26%	54.89%	9.63%	83.02%	2966	2659
<i>SIFT-gradient</i>	67.76%	69.89%	17.62%	94.66%		
<i>SIFT-HILL</i>	60.14%	60.66%	8.40%	87.50%		

lines represent the correct matches, while red lines represent the errors being produced during the matching process. Hence, we observed that *SIFT-RDGP*₂ and *SIFT-HILL* produce better results than *SIFT* because a lower number of false matches. In addition, Table 5 describes the matching error between these descriptors, where *RDGP*₂ and *HILL* reduce the total error matching along the six tests. However, the *SIFT-HILL* presented some false negatives (FN) and some false positive (FP) matches in most cases for indoor and outdoor scenarios, e.g., the false matches in the right hand of the frog must be positive matches. Thus, *SIFT-RDGP*₂ descriptor is a most reliable descriptor than *SIFT-HILL* descriptor for outlier rejection.

In the case of the outdoor scenarios, we collected 150 photographs for recognizing particular tourist attractions situated in Ensenada, B.C., Mexico. These photographs are organized in four different categories such as: BOAT with 52 photos, CEARTE with 34, HUSSONGS with 28 and PAPAS with 36 images; see Fig. 7b), for an example of some of these photographs. Table 6 shows a summary of the matching error between *SIFT-RDGP*₂, *SIFT-HILL* and *SIFT* obtained from the 150 images. Here, our optimized descriptors produce a lower number of false matches than *SIFT* for all cases; however, the *SIFT-HILL* descriptor presented some false negatives (FN) matches as we can observe on the top of the boat. Moreover, the maximum errors computed during recognition represent the matching over those images with the highest complexity, while the minimum errors represent the matching over those images with less distortions. For example, localizing the entire boat showed in Fig. 7b)-I(a) within II(a) would be a difficult task due to we see not enough information in the second image that is the stern of the boat.

5. Conclusions

In this paper, we have described two optimization frameworks for learning composite image operators. Our best optimized descriptor operators represent an improvement over a patented descriptor algorithm called *SIFT* (Scale Invariant Feature Transform) and other man-made descriptors such as *GLOH* and *SURF*. The original idea was to find through *GP* a set of mathematical expressions that could be equal or

better than the weighted gradient magnitude that is applied within the SIFT descriptor. Later, we introduced a new optimization framework based on a hill-climber method in order to compare our GP results with another optimization algorithm. In addition, for comparing both methodologies along with man-made descriptors we considered two different recognition protocols: 1) we have used a standard testbed based on image region matching to evaluate the performance of our evolved mathematical operators, and 2) the best image operator for each algorithm was further tested on real world situations showing that the total number of outliers is greatly reduced. However, the *SIFT-HILL* descriptor presented some false negatives and false positives during the tests, being the *SIFT-RDGP₂* the best reliable descriptor for real applications. Moreover, our GP proposal could be implemented easily in any SIFT-like or similar descriptor, where some kind of mathematical operator is applied. As a matter of fact, in order to use the first protocol in our GP framework, we propose to include the F-Measure in the evaluation process to obtain not only a graphical result as is commonly performed, but also a quantitative measure as required in a GP optimization framework. Thus, our proposed technique opens a research avenue towards evolutionary learning of local descriptors.

Acknowledgments

This research was funded by CONACYT through the project “Evolución de Cerebros Artificiales en Visión por Computadora”, and by Ministerio de Ciencia y Tecnología, Spain, research project TIN2011-28627-C04-03. First author supported by scholarship 0416442 from CONACYT. Second author thanks the support of Junta de Extremadura Government.

References

- [1] A. Abdel-Hakim and A. Farag, C sift: A sift descriptor with color invariant characteristics, *IEEE Conference on Computer Vision and Pattern Recognition* **2** (2006), 1978–1983.
- [2] S. Agarwal, A. Awan and D. Roth, Learning to detect objects in images via a sparse, part-based representation, *Transactions on Pattern Analysis and Machine Intelligence* **26**(11) (2004), 1475–1490.
- [3] P.A. Arbelaez and L.D. Cohen, A metric approach to vector-valued image segmentation, *International Journal of Computer Vision* **69**(1) (2006), 119–126.
- [4] H. Ba, B. Fasel and L. Van Gool, Interactive museum guide: fast and robust recognition of museum objects, In *International Workshop on Mobile Vision*, 2006.
- [5] J. Beis and D. Lowe, Shape indexing using approximate nearest-neighbour search in high-dimensional spaces, *Conference on Computer Vision and Pattern Recognition*, 1997, pp. 1000–1006.
- [6] A. Bosch, A. Zisserman and X. Munoz, Representing shape with a spatial pyramid kernel, In *ACM International Conference on Image and Video Retrieval*, 2007, pp. 401–408.
- [7] M. Brown, R. Szeliski and S. Winder, Multi-image matching using multi-scale oriented patches, In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 510–517.
- [8] G. Carneiro and A.D. Jepson, Phase-based local features, In *European Conference on Computer Vision* **2350** (2002), 282–296.
- [9] G. Carneiro and A.D. Jepson, Multi-scale phase-based local features, In *IEEE Conference on Computer Vision and Pattern Recognition*, (Vol. 1), 2003, pp. 736–743.
- [10] A. Çarkacıoğlu and F. Yarman-Vural, Sasi: A generic texture descriptor for image retrieval, *Pattern Recognition* **33**(11) (2003), 2615–2633.
- [11] J. Chen, S. Shan, G. Zhao, X. Chen, W. Gao and M. Pietikäinen, A robust descriptor based on weber’s law, In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–7.
- [12] H. Cheng, Z. Li, N. Zheng and J. Yang, A deformable local image descriptor, In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
- [13] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, In *IEEE Conference on Computer Vision and Pattern Recognition*, (Vol. 2), 2006, pp. 886–893.

- [14] M. Ebner and A. Zell, Evolving a task specific image operator. In *First European Workshops, EvoIASP'99 and EuroEcTel'99*, volume 1596, In R. Poli et al. (eds.), 1999, pp. 74–89.
- [15] W. Freeman and E. Adelson, The design and use of steerable filters, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(9) (1991), 891–906.
- [16] W. Fu, M. Johnston and M. Zhang, Genetic programming for edge detection: A global approach, In *IEEE Congress on Evolutionary Computation*, 2011, pp. 254–261.
- [17] J. Geusebroek, Compact object descriptors from local colour invariant histograms, In *British Machine Vision Conference*, (Vol. 3), 2006, pp. 1029–1038.
- [18] J. Geusebroek, R. van den Boomgaard, A. Smeulders and H. Geerts, Color invariance, *IEEE Trans Pattern Anal Machine Intell* **23**(12) (2001), 1338–1350.
- [19] R. Gupta and A. Mittal, Smd: A locally stable monotonic change invariant feature descriptor, In *European Conference on Computer Vision*, (Vol. 5303), 2008, pp. 265–277.
- [20] T. Tuytelaars H. Bay and L. Van Gool, Surf: Speeded up robust features, In *European Conference on Computer Vision*, (Volume LNCS 3951), 2006, pp. 404–417.
- [21] B. Hernández, G. Olague, R. Hammoud, L. Trujillo and E. Romero, Visual learning of texture descriptors for facial expression recognition in thermal imagery, *Computer Vision and Image Understanding, Special Issue on Vision Beyond the Visual Spectrum* **106**(2–3) (2007), 258–269.
- [22] D. Howard, S.C. Roberts and R. Brankin, Target detection in sar imagery by genetic programming, *Advances in Engineering Software* **30**(5) (1999), 303–311.
- [23] M.P. Johnson, P. Maes and T. Darrell, Evolving visual routines, In *ARTIFICIAL LIFE IV, International Workshop on the Synthesis and Simulation of Living Systems*, MIT Press, 1994, pp. 198–209.
- [24] Y. Ke and R. Sukthankar, Pca-sift: A more distinctive representation for local image descriptors, In *IEEE Conference on Computer Vision and Pattern Recognition*, (Vol. 2), 2005, pp. 506–513.
- [25] J.J. Koenderink and A.J. van Doorn, Representation of local geometry in the visual system, *Biological Cybernetics* **55** (1987), 367–375.
- [26] J.R. Koza, *Genetic programming: On the programming of computers by means of natural selection*, The MIT Press, 1992.
- [27] N. Lazarevic-McManus, J.R. Renno, D. Makris and G.A. Jones, An object-based comparative methodology for motion detection based on the f-measure, *Computer Vision and Image Understanding* **111** (2008), 74–85.
- [28] S. Lazebnik, C. Schmid and J. Ponce, A sparse texture representation using affine-invariant regions, In *IEEE Conference on Computer Vision and Pattern Recognition*, (Vol. 2), 2003, pp. 319–324.
- [29] S. Lazebnik, C. Schmid and J. Ponce, Semi-local affine parts for object recognition, In *British Machine Vision Conference*, (Vol. 2), 2004, pp. 959–968.
- [30] S. Lazebnik, C. Schmid and J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, In *IEEE Conference on Computer Vision and Pattern Recognition* 2006, pp. 2169–2178.
- [31] Y. Lin and B. Bhanu, Evolutionary feature synthesis for object recognition, *IEEE Trans. on Systems, Man and Cybernetics, Part C, Special Issue on Knowledge Extraction and Incorporation in Evolutionary Computation* **35**(2) (2005), 156–171.
- [32] T. Lindeberg, On scale selection for differential operators, In *Conference on Image Analysis*, 1993, pp. 857–866.
- [33] T. Lindeberg, Edge detection and ridge detection with automatic scale selection, *International Journal of Computer Vision* **30**(2) (1998), 117–154.
- [34] H. Ling and D. Jacobs, Deformation invariant image matching. In *International Conference on Computer Vision*, (Vol. 2), 2005, pp. 1466–1473.
- [35] C. Liu, J. Yuen, A. Torralba and J. Sivic, Sift flow: dense correspondence across different scenes, In *European Conference on Computer Vision*, 2008, pp. 28–42.
- [36] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision* **60**(2) (2004), 91–110.
- [37] S. Luke and L. Panait, Lexicographic parsimony pressure. In *Genetic and Evolutionary Computation Conference*, 2002, pp. 829–836.
- [38] B.S. Manjunath, J. Ohm, V. Vasudevan and A. Yamada, Color and texture descriptors, *IEEE Transactions on Circuits and Systems for Video Technology* **11**(6) (2001), 703–715.
- [39] D.R. Martin, C.C. Fowlkes and J. Malik, Learning to detect natural image boundaries using local brightness, color and texture cues, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(5) (2004), 530–549.
- [40] K. Mikolajczyk and C. Schmid, A performance evaluation of local descriptors, In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 525–531.
- [41] K. Mikolajczyk and C. Schmid, A performance evaluation of local descriptors, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(10) (2005), 1615–1630.

- [42] P. Moreels and P. Perona, Evaluation of features detectors and descriptors based on 3d objects, *International Journal of Computer Vision* **73**(3) (2007), 263–284.
- [43] P. Moreno, A. Bernardino and J. Santos-Victor, Improving the sift descriptor with smooth derivative filters, *Pattern Recognition Letters* **30**(1) (2009), 18–26.
- [44] E. Mortensen, H. Deng and L. Shapiro, A sift descriptor with global context, In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 184–190.
- [45] G. Olague and L. Trujillo, Evolutionary-computer-assisted design of image operators that detect interest points using genetic programming, *Image and Vision Computing*, 2011.
- [46] C.B. Perez and G. Olague, Learning invariant region descriptor operators with genetic programming and the f-measure, In *International Conference on Pattern Recognition*, 2008, pp. 1–4.
- [47] C.B. Perez and G. Olague, Evolutionary learning of local descriptor operators for object recognition, In *Genetic and Evolutionary Computation Conference*, 2009, pp. 414–419.
- [48] C.B. Perez and G. Olague, Evolving local descriptor operators through genetic programming, In *Proceedings of the EvoWorkshops on Applications of Evolutionary Computing: EvoIASP*, 2009, pp. 1051–1058.
- [49] R. Poli, Genetic programming for feature detection and image segmentation, In *Workshop on Evolutionary Computation*, (Vol. 1143), In T. Fogarty (ed.), 1996, pp. 110–125.
- [50] R. Poli, W. Langdon and N. Freitag, A field guide to genetic programming, Published via <http://lulu.com> and freely available at <http://www.gp-field-guide.org.uk>, 2008.
- [51] S.J. Russell and P. Norving, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2003.
- [52] S. Sarfraz and O. Hellwich, Head pose estimation in face recognition across pose scenarios, In *International Conference on Computer Vision Theory and Applications*, 2008, pp. 235–242.
- [53] A. Song and V. Ciesielski, Texture segmentation by genetic programming, *Evolutionary Computation, Special Issue on Evolutionary Computer Vision* **16**(4) (2008), 461–481.
- [54] A. Stein and M. Hebert, Incorporating background invariance into feature-based object recognition, In *IEEE Workshops on Application of Computer Vision (WACV/MOTION'05)*, 2005, pp. 37–44.
- [55] W.A. Tackett, Genetic programming for feature discovery and image discrimination, In *International Conference on Genetic Algorithms*, 1993, pp. 303–309.
- [56] A. Teller and M. Veloso, Pado: Learning tree structured algorithms for orchestration into an object recognition system, Technical report, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 1995.
- [57] E. Tola, V. Lepetit and P. Fua, A fast local descriptor for dense matching, In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
- [58] L. Trujillo and G. Olague, Using evolution to learn how to perform interest point detection, In *International Conference on Pattern Recognition*, 2006, pp. 211–214.
- [59] L. Trujillo and G. Olague, Automated design of image operators that detect interest points, *Evolutionary Computation, Special Issue on Evolutionary Computer Vision* **16**(4) (2008), 483–507.
- [60] L. Trujillo, G. Olague, P. Legrand and E. Lutton, Regularity based descriptor computed from local image oscillations, *Optics Express* **15**(10) (2007), 6140–6145.
- [61] C.J. Van-Rijsbergen, *Information retrieval*, ed. Butterworth-Heinemann, (Second edition), 1979.
- [62] S. Winder and M. Brown, Learning local image descriptors, In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
- [63] M. Zhang, V.B. Ciesielski and P. Andrae, A domain-independent window approach to multiclass object detection using genetic programming, *Journal on Applied Signal Processing* **8** (2003), 841–859.

Copyright of Intelligent Data Analysis is the property of IOS Press and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.