Evolutionary Learning of Local Descriptor Operators for Object Recognition

Cynthia B. Perez
Centro de Investigación Cientifica y de Educación Superior de Ensenada (CICESE)
Km. 105 Tijuana-Ensenada
Ensenada B.C., México
cbperez@cicese.mx

Gustavo Olague
Centro de Investigación Cientifica y de Educación Superior de Ensenada (CICESE)
Km. 105 Tijuana-Ensenada
Ensenada B.C., México
olague@cicese.mx

ABSTRACT

Nowadays, object recognition is widely studied under the paradigm of matching local features. This work describes a genetic programming methodology that synthesizes mathematical expressions that are used to improve a well known local descriptor algorithm. It follows the idea that object recognition in the cerebral cortex of primates makes use of features of intermediate complexity that are largely invariant to change in scale, location, and illumination. These local features have been previously designed by human experts using traditional representations that have a clear, preferably mathematically, well-founded definition. However, it is not clear that these same representations are implemented by the natural system with the same representation. Hence, the possibility to design novel operators through genetic programming represents an open research avenue where the combinatorial search of evolutionary algorithms can largely exceed the ability of human experts. This paper provides evidence that genetic programming is able to design new features that enhance the overall performance of the best available local descriptor. Experimental results confirm the validity of the proposed approach using a widely accept testbed and an object recognition application.

Categories and Subject Descriptors

I.4.7 [**Image Processing and Computer Vision**]: Feature Measurement—*feature representation, invariants*; I.2.2 [**Artificial Intelligence**]: Automatic Programming—*program synthesis*

General Terms

Algorithms, Experimentation, Performance, Theory

Keywords

SIFT, Local Descriptors, Object Recognition, Matching

Montreal '09 Montreal, Canada

1. INTRODUCTION

Object Recognition is a fundamental process of visual perception that seems to be executed effortlessly by humans, as well as by many biological vision systems such as insects. In fact, we are constantly engaged in the process of recognizing multiple objects from a given visual scenario without paying too much attention to the internal mechanism of recognition. In this way, it is not easy to define the term "object recognition" in a simple, precise, and uncontroversial manner. Recently, a number of researchers have proposed the use of genetic programming (GP) to approach the challenging task of object recognition[4, 5, 6, 10, 11, 20, 24]. There are several reasons to approach such study through a genetic programming methodology. The first is to take advantage of an unsolved and challenging problem in order to create and study new genetic programming algorithms. Thus, the difficulty of the recognition problem is alleviated in benefit of improving the GP technique, see [23, 24]. In fact, such idea can be traced back to one of the earliest published work [6] where the primary goal was proof of concept rather than executing a rigorous experimental test. The work of Howard [6, 7] outlines the idea that genetic programming is able to design detectors that represent algebraic formulae and as a result the principles of detection can be discovered and reused. Such idea is original to the genetic programming paradigm in general and it represents a major advantage over other approaches such as neural networks. The second strategy is to investigate an application of genetic programming for object detection where it is desirable to keep focused on the standard GP methodology. The idea is to provide evidence to other researchers that genetic programming offers a powerful technique for knowledge discovery. This is the path that we are following in this research. A third approach is simply to make a combination of the first two. In other words, provide a new kind of genetic programming variant, while solving at the same time a problem of indisputable difficulty in the object recognition field. However, we believe that currently it is too early to make such a contribution.

Traditionally, most research on object recognition involves four stages: preprocessing, segmentation, feature extraction and classification that cover the whole visual chain. Recently, an approach based on local features adopts a simplified methodology for object recognition, where the preprocessing and segmentation stages are eliminated by only focusing on local and relatively small amounts of image information [18]. The Scale Invariant Feature Transform (SIFT) proposed by Lowe [13] provides a rich set of local feature vec-

tors that are said invariant to image translation, rotation, scaling; and are partially invariant to illumination changes and affine or perspective projection. Previous research [20] on local feature detection has introduced genetic programming as a suitable strategy that synthesizes interest point detectors. As explained by Lowe the scale invariant features are efficiently identified by using a stage filtering approach. The first part of the approach identifies key locations in scale space by looking for locations that are maxima or minima according to a difference of Gaussian function. Each point is used to generate a feature vector that describes the local image region sampled relative to its scale-space coordinate frame. SIFT is based on a model of the behavior of complex cells in the cerebral cortex of the mammalian vision system. Thus, it is appealing to use the idea that a genetic programming approach can synthesize novel mathematical expressions that improve the general SIFT behavior. This could be compared with the evolution of complex cells in an artificial cerebral cortex.

This paper is organized as follows: first, we recall a well known and accepted testbed used for comparison of local descriptors. Then, we outline the genetic programming approach that was implemented by correctly framing the problem statement through two essential aspects: the fitness function and the search space. Finally, we provide a complete set of 30 new results that are better than the current state-of-the-art according to the test. Also, we present a set of examples of an object recognition task using the best evolved operator.

2. PROBLEM STATEMENT

2.1 Testbed for Local Descriptors

Nowadays a new image recognition paradigm is applied using a set of invariant local features that are computed from an image sample and later are matched against a large database of images. This idea was first proposed by Schmid and Mohr [18] who showed that image matching through local features could be extended to general image recognition problems. They computed Harris corners to select interest points and instead of matching with a correlation window, they used a rotationally invariant descriptor. Today, this approach achieves widespread acceptance and SIFT is considered as the best representative algorithm due to its robustness against scale changes. However, the SIFT acceptance was significantly helped by the work of Mikolajczyk and Schmid [14] who designed a testbed for local descriptors. A performance evaluation of local descriptors was implemented through an experimental evaluation of interest region descriptors in the presence of real geometric and photometric transformations. This testbed is available through internet[1], with the binary code of all descriptors, as well as their complete data set used for evaluation. Today, this testbed is widely accepted as a standard performance evaluation. Thus, we decide to use it within our genetic programming approach in order to synthesize novel mathematical expressions that can outperform previous results using a widely accepted and standard testbed.

The test is based on the number of correct matches and the number of false matches obtained for an image pair, see Fig. 1. The idea is to create a Recall vs 1-Precision space using a set of metrics computed from the contingency table. Thus, the true positive (TP) refers to the correct matches being detected by the system; while, the false positive (FP) denotes the set of false matches being detected by the system. On the other hand, the false negative (FN) and true negative (TN) represent the correct and false matches not detected by the system respectively. In this particular problem the true negatives are never computed; therefore, it is better to work on the Precision-Recall space rather than the ROC (Receiver Operating Characteristics) space [1].

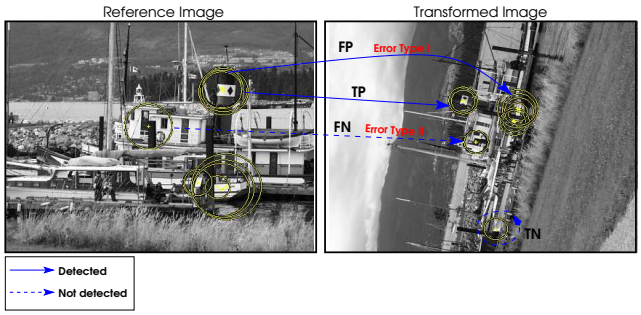


Figure 1: Interpretation of matching features.

The test consists on counting the matching of two regions A and B; if the distance between their descriptors D_A and D_B is below a threshold t. Thus, each descriptor from the reference image is compared with each descriptor from the transformed one in order to compute the number of correct and false matches. The value of t is varied to obtain the curves of Recall vs 1-Precision. Recall is the number of correctly matched regions with respect to the number of corresponding regions 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 A and B. Hence, the 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 given threshold. Furthermore, 1-Precision is computed as the number of false matched regions with respect to the total number of regions being matched. Notice that the false matches are computed from total matches minus the correct matches.

The above evaluation is used to create a set of graphs in the Recall vs 1-Precision space. Later, they compare visually a set of graphs computed with several descriptors in order to decide which is the best descriptor. A perfect descriptor would give a recall equal to 1 for any precision. In practice, recall increases while the threshold is relaxed at the cost of admitting a higher level of noise and decreasing the precision. Horizontal curves indicate that the recall is attained with a certain precision that is limited by the image properties. A drawback of the proposed approach is that it implies a subjective visual interpretation of the possible curve shapes; i.e., a problem occurs when two or more graphs overlap. Moreover, if we want to improve a local descriptor through optimization it is necessary to define a figure-of-merit function.

2.2 Performance Evaluation

The aim of our research is to show that genetic programming is a powerful methodology that is capable of improving

[1]http://www.robots.ox.ac.uk/ vgg/research/affine.

local descriptors that later could be used for object recognition tasks. In general, to apply evolutionary computing we need to devise a well defined fitness function together with the representation of the problem. In this work, we propose to use the F-measure as the fitness function in order to compare several descriptors [15]. This measure is widely used in the information retrieval community as a performance evaluation criterion [21]. The F-measure is based on the harmonic mean and it gives the best balance between precision and recall metrics. The general formula is defined in the following equation:

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

where p is precision $\{p : 0 \leq p \leq 1\}$, r is recall $\{r : 0 \leq r \leq 1\}$, and $\{\alpha : 0 \leq \alpha \leq \infty\}$. Note that in the case of $\alpha < 1$ the variable with a 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. At the time of writing, we are not aware of any work using the F-measure method for evaluating the performance measurement of local descriptors. In fact, there is only one previous work that attempts to improve local descriptors using statistical learning [22]. That work proposes the area under the ROC curve as the merit function; however, in our case is very cumbersome to use that criterion. Other criteria could be the true negative rate, true positive rate, weighted accuracy, G-mean, precision, and recall to mention but a few. However, we claim that the F-measure already offers a measure that is simple and reliable in the estimation of local descriptor performance.

2.3 Structure Representation

The idea of image descriptors has a long tradition in computer vision. The simplest descriptor is a vector of image pixels, where cross-correlation can be used to compute a similarity score between two descriptors. We can find other descriptors that are based on histograms, spatial-frequency techniques and image derivatives. Today, many researchers have a special interest in SIFT given its ability to detect objects under partial occlusion, invariant to rotation, scale and robust against a substantial range of affine distortion, noise, illumination change, and that a large number of keypoints could be extracted at near-real time performance. Zhang has recognized the complexity of improving such descriptor through a genetic programming approach [23]. Therefore, to keep the complexity low he has preferred to represent its descriptors with a set of primitive operators; other authors have used a similar approach, see [5, 6, 10, 11, 12].

In order to devise a structure representation that can be improved with genetic programming 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 performance of SIFT descriptors. SIFT consists of four stages and the local image descriptors are computed in the last stage. Such description of information is based upon image gradients of a patch of pixels within a local neighborhood. This patch is centered on the keypoint location, rotated with respect to the dominant orientation and scaled to the appropriate size. We propose in this work to name the gradient magnitude normally used in SIFT and its variants as the SIFT operator. We will use the acronym RDGP (Region Descriptor Operators with Genetic Programming) to indicate our evolved SIFT operators, see Fig. 2.

We will observe in the experiments that in fact this simple change has resulted in a set of operators that significantly outperforms other state-of-the-art SIFT versions. Since the publication of SIFT many authors have attempted to improve it using several ideas. For example, Ke and Sukthankar [8] proposed PCA-SIFT in order to represent SIFT features in a more compact way using the normalized gradient patch instead of the standard SIFT representation. Bay et al. [2] also designed a descriptor called SURF, which is a SIFT version specially designed for real time applications. Mikolajczyk and Schmid [14] suggested a descriptor called GLOH, which is a SIFT variant where the local region used to build the histogram is computed with polar coordinates and PCA was also employed. Dalal and Triggs [3] proposed also a SIFT variant called HOG that is based on evaluating well-normalized local histograms of image gradient orientations in a dense grid. Tola et al. [19] introduced a SIFT version named DAISY with the idea of computing fast dense matching in the case of wide baseline configurations using graph-cuts. In the experiments we compare our evolved descriptors against binary versions of SIFT, GLOH, and SURF using several keypoint detectors that are normally used to improve their overall performance. In all cases the new evolved descriptors are much better according to the standard testbed.

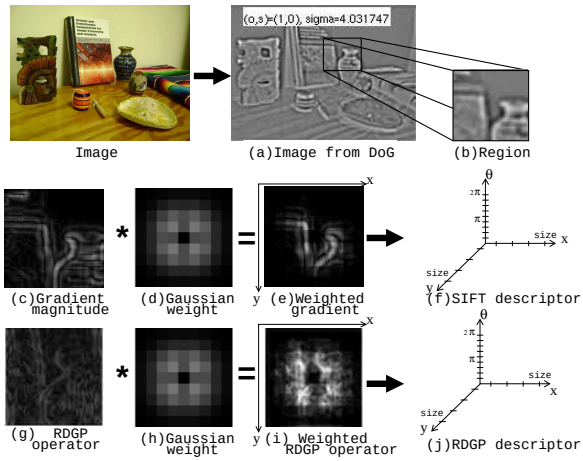


Figure 2: Local image descriptor operator used within SIFT: c) gradient magnitude g) RDGP.

3. EVOLVING SIFT OPERATORS WITH GP

In this section, we explain our genetic programming methodology that synthesize SIFT operators from 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 genetic programming is able to create composite operators even from scratch [12, 16, 17]. However, we believe that the choice of well selected function and terminal sets is of paramount importance for the final result [20]. 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.

3.1 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 [9] 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:

$$\begin{aligned} FuncSet &= \Big\{+, |+|, -, |-|, *, \div, \sqrt{I_t}, \\ &\frac{I_t}{2}, log_2(I_t), D_xG_\sigma, D_yG_\sigma, G_\sigma\Big\} \\ TermSet &= \{I, D_x, D_{xx}, D_{yy}, D_{xy}, D_y\} \end{aligned} \quad (2)$$

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 the image derivatives along direction u then $D_u = I{*}G_{u(\sigma=1)}$; G_σ are the Gaussian smoothing filters with $\sigma = 1$ or 2; D_uG_σ represents the derivative of a Gaussian filter with image blur σ.

3.2 Fitness Function

An appropriate fitness function is decisive to the GP process success. Thus, our fitness function is based on a well balance precision and recall data as explained earlier:

$$\begin{aligned} Q = \text{argmax} &\left\{F_\alpha(P^x, R^x) = \sum_{i=1}^{n} \frac{(1+\alpha)\cdot(p_i \cdot r_i)}{(\alpha \cdot p_i) \ + \ r_i}\right\} \\ where \quad Q : F_\alpha(P^s, R^s) \quad &\geq \quad F_\alpha(P^t, R^t) \end{aligned} \quad (3)$$

with n representing the number of thresholds. Precision data from an image pair are denoted by $P^x = (p_1, p_2, ..., p_n)$ and recall data by $R^x = (r_1, r_2, ..., r_n)$; where x represents a possible solution and $s, t \subseteq x$. The ranking order of Q is ascendent where the highest value of Q corresponds to the descriptor with the best performance. It is also possible to use the mean of Q instead of the total sum of F_α.

4. EXPERIMENTAL RESULTS

This section describes the implementation setup, the training and testing sets, and the results of our proposed GP algorithm. The section is divided in two parts: first, we describe the process of learning and testing using the standard testbed; second, we present the application of our best evolved descriptor in an object recognition task.

4.1 Learning and Testing SIFT Operators

We learn and test the evolved operators on real images with different geometric and photometric transformations and for different scene types. The implementation of the GP approach was programmed on Matlab, using the GPLAB [2]. Also, the core platform of our algorithm is based on SIFT features programmed in Matlab/C [3]. Table 1 specifies the GP runtime parameters used during the experimental tests. The image sequence used for training was the Boat set with combined image rotation and scale transformations. Scale changes lie in the range of 2-2.5 and image rotation in the range 30 to 45 degrees. We selected the pair of images with the most significant change for training instead of the whole sequence, because of the computational complexity of computing the evaluation test for a single descriptor. Figure 3 shows the final results of the 30 experiments that were executed; each GP experiment takes about 24 hours for training while for testing it is about 6 seconds. We can observe that the average performance is superior to all man-made descriptors including SIFT, GLOH, and SURF. The SIFT and GLOH descriptors are considered to achieve the best performance from about 10 local descriptors according to Mikolajczyk and Schmid [14]. Also, we identify the second of all thirty tests as the best evolved descriptor and we named RDGP2. This operator was discovered in the last series of genetic operations through a crossover that we can appreciate in Figure 4. Table 2 shows the 30 operators written as S-expressions using prefix notation, together with their final fitness. It is clear from these results that we have found a new local optimum, RDGP2, with the GP approach. We will use the rest of the sequences in the testbed to verify that the best evolved operator is still better than man-made designs. Figure 5 shows final results of the testing including the New York (rotation), Bark (rotation and scale changes), Trees (blur changes), and UBC (JPEG compression) images. Again, rotations are obtained by rotating the camera around its optical axis in the range 30 to 45 degrees. Scale change and blur sequences were acquired by varying the camera zoom and focus, respectively. The JPEG sequence is generated with a standard XV image browser with the image quality parameter set to 5 percent as in [14]. The images are either of planar scenes or the camera position was fixed during acquisition; therefore, they are always related by a homography. We can appreciate that several detectors were used in order to improve the man made descriptor designs using the Hessian-affine, Harris-affine, and Hessian-Laplace detectors. In all cases we keep the DoG detector originally used in the SIFT descriptor as the detector used by RDGP2. We observe that RDGP2 is the best operator, because it improves the overall performance of the SIFT descriptor across very different transformations.

Table 1: 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

4.2 Object recognition aplication

As a further test we decide to implement an object recognition application similar to one proposed by Lowe [13]. The test consists of a set of photographs acquired with a SONY Cyber-shot 7.2 MP digital camera using indoor and outdoor scenarios of texture and non-texture objects, see Figure 6. We compute the epipolar geometry using RANSAC (RAN-

[2]http://gplab.sourceforge.net/index.html, GPLAB a genetic programming Toolbox for Matlab.
[3]http://vision.ucla.edu/vedaldi/code/sift/sift.html

Table 2: RDGP's Operators Training Results.

Descriptor	Fitness	Individual's Expression	Descriptor	Fitness	Individual's Expression
$RDGP_1$	7.4158	$sqrt(sqrt(D_x(sqrt(D_x(sqrt(D_{xy}($ $image)))))))$	$RDGP_2$	7.4859	$sqrt(D_x(sqrt(dif(sqrt(D_{xy}($ $image)),D_{xx}(image)))))$
$RDGP_3$	7.1812	$G_2(G_2(sqrt(D_x(D_y(D_x(D_x($ $image)))))))$	$RDGP_4$	7.3928	$G_2(absdif(G_2(absdif(absdif(D_y$ $(image),D_{xx}(image))),D_y(Log(D_x($ $D_x(image))))))),Half(D_x(D_y(image)))))$
$RDGP_5$	7.4053	$G_1(sqrt(G_2(sqrt(sqrt(dif(sqrt(G_1($ $D_y(image))),div(D_{xx}(image)),$ $absadd(D_x(image),D_y(image)))))))))$	$RDGP_6$	6.9470	$sqrt(dif(sqrt(sqrt(dif(D_{xy}($ $image)),sqrt(D_{xy}(image))))))$ $,sqrt(D_x(image))))$
$RDGP_7$	6.9666	$sqrt(div(absadd(D_{yy}(image)),D_{xx}($ $image))),G_1(Half(D_{xx}(image))))))$	$RDGP_8$	7.1544	$sqrt(sqrt(D_x(G_2(D_{xy}(image))))))$
$RDGP_9$	7.1557	$sqrt(dif(D_{xx}(image)),D_y(Log($ $sqrt(D_{xx}(image)))))))$	$RDGP_{10}$	6.8833	$G_1(dif(sqrt(sqrt(Log(D_{xx}(image$ $)))))),Log(G_2(D_{xy}(image))))))$
$RDGP_{11}$	7.3736	$Half(G_2(G_2(sqrt(D_{xx}(Log(D_{xy}($ $image)))))))))$	$RDGP_{12}$	7.1020	$sqrt(dif(sqrt(D_{xy}(image))),D_y(Log($ $D_{xx}(image))))))$
$RDGP_{13}$	6.9170	$dif(absdif(D_{xy}(image)),div(Half($ $D_x(image)),absadd(D_x(image),$ $Half(D_y(image)))))),D_{xy}(image)))$	$RDGP_{14}$	6.8704	$absadd(Half(D_{yy}(image))),Log(absadd($ $sqrt(D_{yy}(image))),D_{xy}(D_{yy}(image)))))))$
$RDGP_{15}$	6.9203	$absadd(absadd(Half(Log($ $D_{yy}(image)))),Log(Half(Log($ $D_x(image))))),Half(D_{yy}(image))))$	$RDGP_{16}$	7.1264	$sqrt(div(dif(D_{xx}(image)),D_x(image)$ $),absadd(absadd(D_{yy}(D_y(image))),$ $D_{yy}(image))),D_x(image))))$
$RDGP_{17}$	6.8778	$sqrt(D_y(sqrt(Log(Log(D_{xx}($ $image)))))))$	$RDGP_{18}$	6.9633	$absdif(sqrt(G_2(sqrt(sqrt(D_{xx}($ $image))))))),D_{xx}(image)))$
$RDGP_{19}$	7.1225	$sqrt(D_y(add(D_{xy}(image)),Log($ $D_{xx}(image))))))$	$RDGP_{20}$	6.9024	$div(Log(D_{yy}(image))),Log(G_1($ $D_x(G_2(D_x(image))))))$
$RDGP_{21}$	6.8824	$Log(Log(G_2(D_{xy}(image)))))$	$RDGP_{22}$	6.9230	$absdif(D_x(image),sqrt(G_2($ $sqrt(D_{xx}(image))))))$
$RDGP_{23}$	7.0466	$absadd(D_{yy}(image)),sqrt(add($ $D_{yy}(image)),D_{xx}(D_x(image))))))$	$RDGP_{24}$	6.8824	$Log(Log(G_2(D_{xy}(image)))))$
$RDGP_{25}$	7.0207	$absdif(sqrt(absadd(absdif(sqrt($ $D_{xy}(image))),sqrt(D_{yy}(image)))),$ $D_{xx}(D_y(image))))),D_{xx}(image)))$	$RDGP_{26}$	7.1978	$sqrt(absdif(dif(D_y(Log(D_{xx}($ $image)))),D_{yx}(image))),G_2(D_x($ $G_1(D_{xx}(image)))))))$
$RDGP_{27}$	7.0570	$sqrt(sqrt(absdif(D_x(image),sqrt($ $D_y(image)))))$	$RDGP_{28}$	7.0433	$sqrt(dif(sqrt(G_2(absadd(D_{xx}(image)),$ $D_{yy}(D_{xx}(image)))))),D_y(image)))$
$RDGP_{29}$	6.9063	$Half(absadd(absadd(Half(sqrt($ $Log(D_{xy}(image))))),D_y(image)),$ $absdif(sqrt(D_{xy}(image))),G_2($ $absadd(D_{xy}(image)),D_{xx}($ $image)))))))$	$RDGP_{30}$	7.0529	$dif(sqrt(Log(div(D_{xx}(image)),$ $D_{xy}(image))))),D_{yy}(image)))$

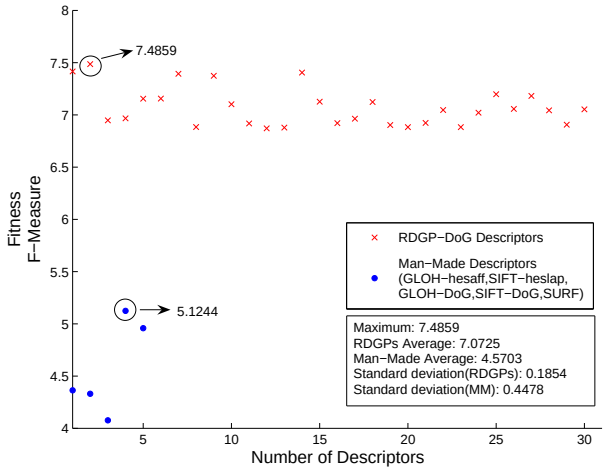


Figure 3: Man-Made and RDGPs Descriptors

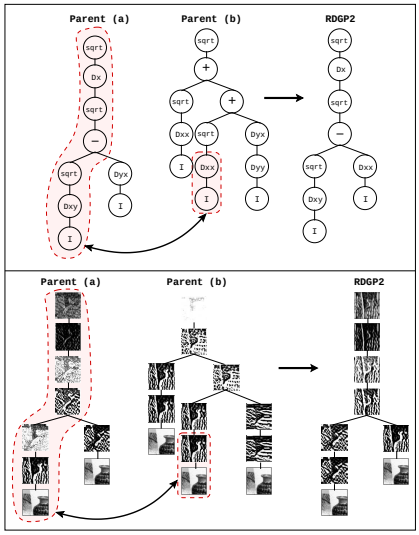
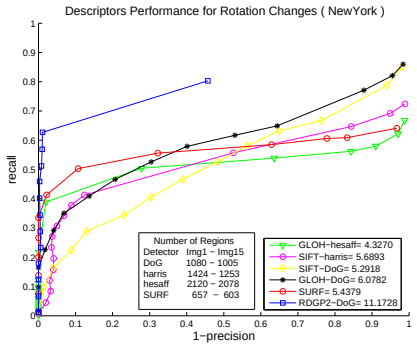


Figure 4: Example of the RDGP2 operator.

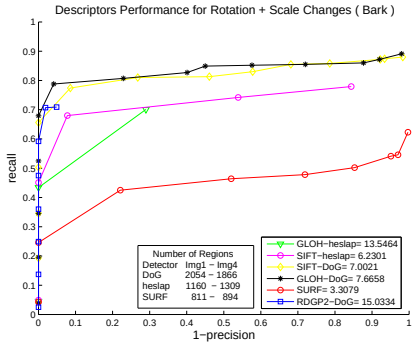
dom SAmple Consensus) algorithm in order to identify the inliers and outliers of the matched features. Figure 7 shows green lines representing the correct matches; while, red lines correspond to the false matches. It is easy to observe that the total number of false matches is significantly reduced, while keeping almost all correct matches. Table 3 provides final results of the recognition tasks, where we can appreciate the error percentage produced by RDGP2 and SIFT descriptors. Thus, the RDGP2 descriptor could be seen as performing a filtering stage over the bad matches.

5. CONCLUSIONS

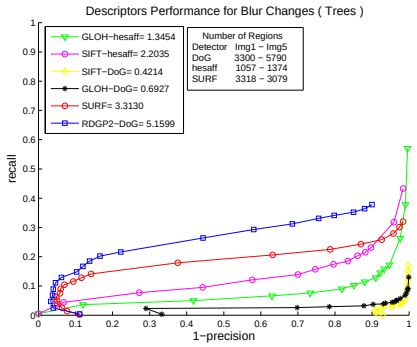
This article presented a genetic programming method to synthesize novel descriptor operators that surpass the overall performance of SIFT-like descriptors. We have demonstrated the effectiveness of our GP approach through an extensive experimental study using a standard testbed and an object recognition application. Finally, we would like to



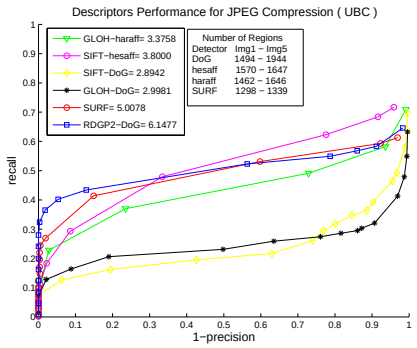
a) Rotation changes



b) Rotation + Scale changes

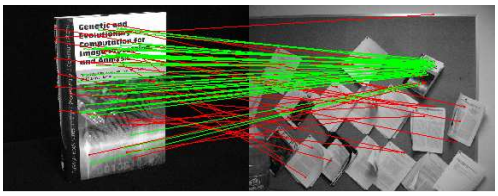


c) Blur changes

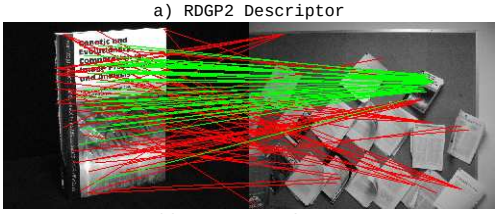


d) JPEG compression

Figure 5: Performance evaluation of RDGP2, SIFT, GLOH and SURF descriptors.

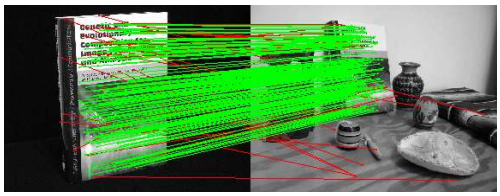


a) RDGP2 Descriptor

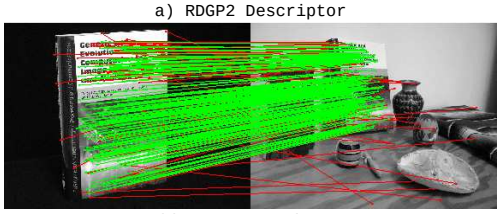


b) SIFT Descriptor

i) Book-Cork images

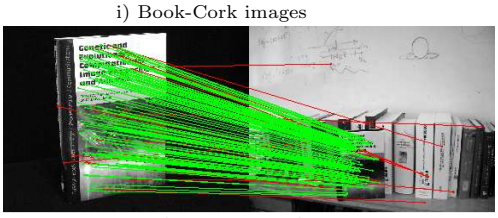


a) RDGP2 Descriptor

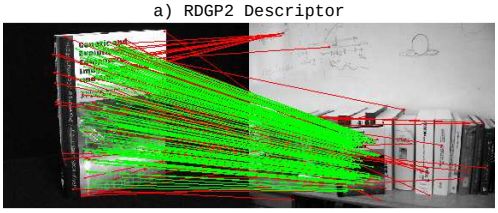


b) SIFT Descriptor

ii) Book-Objects images

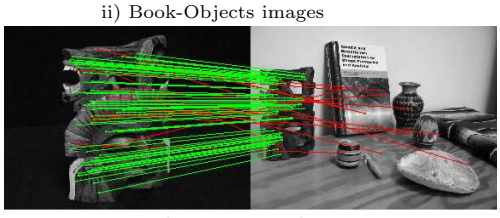


a) RDGP2 Descriptor

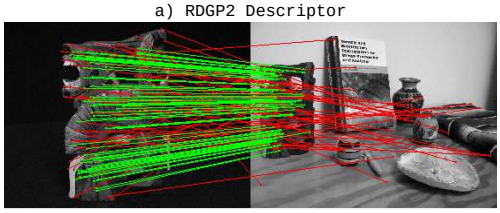


b) SIFT Descriptor

iii) Book-Books images

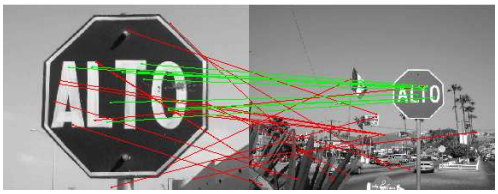


a) RDGP2 Descriptor

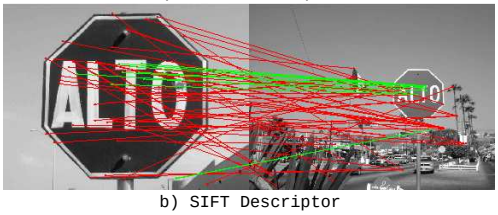


b) SIFT Descriptor

iv) Mask-Objects images

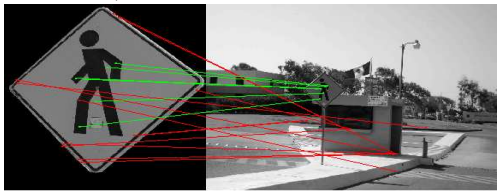


a) RDGP2 Descriptor

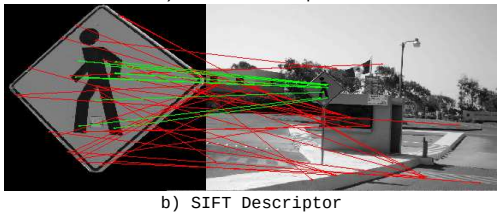


b) SIFT Descriptor

v) Stop Sign-Costero Blvd. images



a) RDGP2 Descriptor



b) SIFT Descriptor

vi) Pedestrian-UABC images

Figure 7: Matching descriptors from indoor and outdoor scenarios.

Table 3: Matching Error between $RDGP_2$ descriptor and SIFT descriptor.

	DETECTION STAGE		MATCHING STAGE (using descriptors)			
Descriptor	Image Object	Image Scene	Total	Corrects	Incorrects	Error (%)
Book-Cork						
$RDGP_2$	1588	1418	83	50	33	**39.75 %**
SIFT			112	55	57	**50.89 %**
Book-Objects						
$RDGP_2$	1588	2044	255	214	41	**16.08 %**
SIFT			318	240	78	**24.53 %**
Book-Books						
$RDGP_2$	1588	1576	169	146	23	**13.61 %**
SIFT			253	167	86	**33.99 %**
Mask-Objects						
$RDGP_2$	1375	2044	91	73	8	**34.78 %**
SIFT			138	84	40	**68.96%**
Pedestrian-UABC						
$RDGP_2$	189	1579	23	15	18	**62.06 %**
SIFT			58	18	52	**81.25 %**
Stop-Costero Blvd.						
$RDGP_2$	419	1603	29	11	18	**19.78 %**
SIFT			64	12	54	**39.13 %**

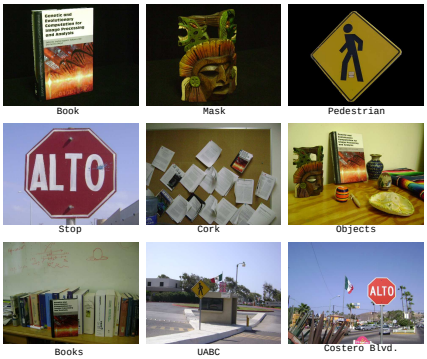


Figure 6: Database used in the object recognition application.

mention that the proposed technique opens a research avenue towards evolutionary learning of local descriptors.

6. REFERENCES

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. *In Proceedings of the Seventh European Conference on Computer Vision. LNCS: 2353*, pages 97 – 101, 2002.

[2] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *ECCV*, 3951:404–417, 2006.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR - Volume 1*, pages 886–893, Washington, DC, USA, 2005.

[4] M. Ebner. An adaptive on-line evolutionary visual system. *IEEE Workshop on Pervasive Adaptation*, pages 84–89, 2008.

[5] B. Gokberk, M. Irfanolglu, L. Akarun, and E. Alpaydin. Learning the best subset of local features for face recognition. *Pattern Recognition*, 40:1520–1532, 2006.

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

[7] D. Howard, S. C. Roberts, and C. Ryan. The boru data crawler for object detection tasks in machine vision. *EvoWorkshops. LNCS: 2279*, pages 222–232, 2002.

[8] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *CVPR, 27 June - 2 July, Washington, DC*, volume 2, pages 506–513. IEEE Computer Society, 2004.

[9] J. J. Koenderink and A. J. V. Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(3):367–375, 1987.

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

[11] Y. Lin and B. Bhanu. Evolutionary feature synthesis for object recognition. *IEEE Trans. on Systems, Man, and Cybernetics-Part C: Apps and Revs*, 35(2):156–171, 2005.

[12] Y. Lin and B. Bhanu. Object detection via feature synthesis using mdl-based genetic programming. *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, 35(3):538–547, 2005.

[13] D. Lowe. Object recognition from local scale-invariant features. *ICCV*, pages 1150–1157, 1999.

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

[15] C. B. Perez and G. Olague. Learning invariant region descriptor operators with genetic programming and the f-measure. *International Conference on Pattern Recognition*, December 8-11 2008.

[16] 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.

[17] J. R.Koza. *Genetic Programming*. The MIT Press., 1993.

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

[19] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *CVPR, 23-28 June, Anchorage, AK*. IEEE Computer Society, 2008.

[20] L. Trujillo and G. Olague. Synthesis of interest point detectors through genetic programming. *Genetic and Evolutionary Computation Conference*, pages 887–894, 2006.

[21] C. Van-Rijsbergen. *Information retrieval*. Ed. Butterworth-Heinemann. Second Edition, 1979.

[22] S. Winder and M. Brown. Learning local image descriptors. *IEEE Conf. on CVPR*, pages 1–8, 2007.

[23] M. Zhang. Genetic programming for object detection: A two-phase approach with an improved fitness function. *Electronic Letters on Computer Vision and Image Analysis*, 6(1):27–43, 2007.

[24] M. Zhang, V. B. Ciesielski, and P. Andreae. A domain-independent window approach to multiclass object detection using genetic programming. *EURASIP Journal on Applied Signal Processing*, 2003(8):841–859, 2003.