Automatic Synthesis of Computationally Efficient Interest Point Detectors

Ivor Uhliarik*
Supervised by: Zuzana Berger Haladová†

Faculty of Mathematics, Physics and Informatics
Comenius University
Bratislava / Slovakia

Abstract

With the arrival of the trend of integrating powerful graphics processing units into modern hand-held devices, performing complex computations is becoming feasible to the point that allows developers to deploy augmented reality-enabled smart-phone applications. This work aims to tackle the challenges of establishing an efficient pipeline of image processing tasks involved therein. We focus on the automatic synthesis of the interest point detection operator using a multiobjective genetic programming (MO-GP) framework that promotes properties suitable for detecting local features in cluttered scenes. In previous works, three properties chosen as the genetic programming (GP) search objectives have been used: stability, point dispersion, and information content. We seek to expand this approach with a fourth objective that emphasizes computational efficiency, taking parallelizability of algorithms into account. The produced operators are then validated using a set of images with appropriate content and compared with the results of existing approaches. Finally, the most promising Pareto-optimal operators are efficiently implemented in the Android RenderScript framework for use in Android mobile applications.

Keywords: augmented reality, genetic programming, interest point detection, multiobjective optimization, parallel computing

1 Introduction

Many applications of augmented reality heavily rely on the object recognition pipeline. The most frequently used method to recognize unmarked objects in unorganized, cluttered scenes is based on local image features. The process typically consists of three phases. First, salient regions in the image are detected using the *interest point (IP) detector*. Then, local features are computed for each interest point based on its local neighborhood. These features are assembled into feature vectors by *interest point descriptors*, which try to distinctively capture the nature of objects represented by interest points. Finally, these descriptions are *matched* with precomputed descriptions of objects in a database and the nearest match is declared as the recognized object.

This approach offers resilience toward scenes where the objects are occluded or otherwise distorted. Several human-designed algorithms have emerged and proven to perform well over the last few decades, and are still an active topic in research. The drawback of this system is, however, that the task of image recognition in a general case lacks a formal definition. This results in the variance of scenarios in which different algorithms perform well. For example, some interest point detectors focus on detecting corners, while other detect edges, ridges, or blobs. Moreover, existing algorithms considerably differ in their computational complexity [14].

In recent years, there has been effort to construct algorithms used in the image recognition pipeline in an automated way using evolutionary algorithms. Olague and Trujillo in [6] have proposed a multiobjective genetic programming approach to the synthesis of interest point detectors. To counter the bias introduced in existing human-designed algorithms, this approach promotes theoretical properties the interest point detector should maximize, which are discussed in the next chapters. The outcome of this work is a set of synthesized interest point detection algorithms that exceed the recognition performance of several human-designed algorithms.

Our work seeks to build up an MO-GP framework for the synthesis of interest point detectors similar to that in [6] and extend it with a proposed novel objective that maximizes computational efficiency. The focus is laid on the feasibility of the synthesized operators to be implemented in mobile devices with competent parallel processing power, such as the modern smart-phones, whose potential is often left unexploited.

The motivation of seeking to improve the detection phase of the object recognition pipeline rests on the low-level nature and ubiquity of IP detection. The resulting algorithms may be useful for all tasks in computer vision

*ivor.uhliarik@gmail.com
†zhaladova@gmail.com

that IP detection is a part of.

Note that this is still a work in progress. The purpose is to design a framework for performing experimental research.

In Section 2, we describe the previous and ongoing work in both human-designed interest point detection algorithms and their automatic synthesis. Later in Section 3, we delve into the four qualities of detectors our MO-GP maximizes. In Section 4, we describe the genetic programming concept and explain how we optimize multiple objectives therein. Section 5 displays the results we have acquired and finally, Section 6 sums up the conclusions of our work.

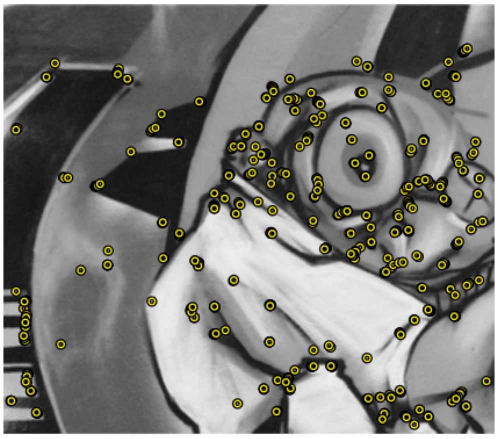

Figure 1: Example of points detected by the Harris interest point detector [14]

2 Related Work

Human-designed algorithms for interest point detection based on local features are still the most commonly used method in the detection phase of the recognition pipeline. These can be categorized into corner and edge detectors, blob detectors, and region detectors.

An example of a corner detector is the Harris detector proposed by Harris and Stephens [2]. The approach is based on the auto-correlation matrix that describes the gradient distribution in the local neighborhood of a point. The eigenvalues of this matrix represent the principal changes in the image signal. The point for which both of the eigenvalues are large is likely to be a corner. The output of the Harris detector is shown in Figure 1. Other corner-based interest point detectors include SUSAN [9] and FAST [7].

An example of a blob detector is the scale-invariant feature transform (SIFT) by Lowe [4], which is distinguished by its extension of the image space by sub-sampling and smoothing methods to form the scale-space. This allows for scale-invariant object detection.

A comprehensive comparison of human-designed interest point detectors is discussed in [14].

The already mentioned disadvantages of human-designed algorithms are apparent: each of these solutions maximizes ad-hoc objectives. There is no universal consensus as to what defines the salience of the detected interest points. Images with smooth corners may be overlooked by corner detectors, while the points in other images may be captured more meaningfully by corner detectors rather than blob detectors.

The ad-hoc fashion of the objectives of human-designed algorithms was challenged by the work of Olague and Trujillo in 2006 [11]. A (single-objective) genetic programming approach was proposed that synthesizes interest point detectors. The approach promotes detector *stability* and *point dispersion* using a single fitness function. The output consists of several generated algorithms in form of computational trees built up using low-level image transformations. Results of this work have been competitive to the human-designed state of the art algorithms. Later work by Trujillo and Olague in 2008 [12] showed new results that yielded performance better than many human-designed algorithms at the time using a similar setup. In 2011 [5], the same authors proposed a multi-objective GP approach, effectively splitting stability and point dispersion into two separate objectives (fitness functions). Several novel synthesized interest point detectors have been introduced. Finally, in 2012 [6], the work of Trujillo and Olague continued with the addition of a novel, third objective: *information content*.

As our proposal is based on the progressive work of Trujillo and Olague, the key concepts and algorithms will be further explained in the following sections. An example of a synthesized detector is shown in Figure 2.

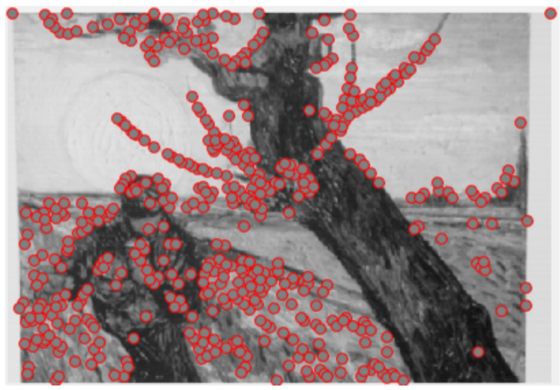

Figure 2: Example of points detected by an evolved operator synthesized by the MO-GP process using our framework

Similar work has been conducted in the synthesis of interest point *descriptors* by Liu et al. in 2013 [3]. The proposed solution uses MO-GP for the synthesis of image descriptors yielding feature vectors for detected interest points.

3 Qualities of IP Detectors

In this section, we describe the three qualities proposed by Trujillo and Olague in [6] a good interest point detector should exhibit. Next, we propose fourth, novel objective used in our work that extends the solution.

3.1 Stability

Stability of an interest point detector is measured by its repeatability rate. In practice, this corresponds to the level of invariance of the detector toward affine transformations of the image. This objective is crucial in achieving good performance of interest operators in cluttered scenes. Given images I_1 and I_i, the set of point pairs (x_1, x_i) that are repeated in image I_i based on the image I_1 related by homography H_{1i} with maximum error of ε is

$$R_{I_i}(\varepsilon) = \{(x_1, x_i) | dist(H_{1i}x_1, x_i) < \varepsilon\}. \quad (1)$$

The overall repeatability rate is calculated as

$$r_{I_i}(\varepsilon) = \frac{|R_{I_i}(\varepsilon)|}{min(\gamma_1, \gamma_i)}, \quad (2)$$

where $\gamma_1 = |\{x_1\}|$ and $\gamma_2 = |\{x_i\}|$. $min(\gamma_1, \gamma_i)$ in the equation represents the total number of extracted points.

The concept of the measure of repeatability is depicted in Figure 3.

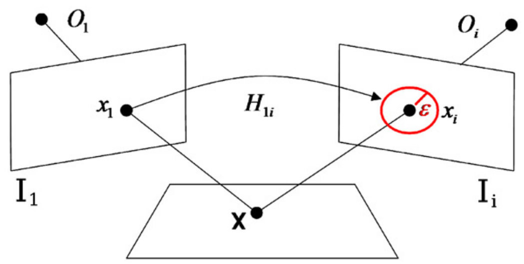

Figure 3: An illustration of the stability objective criterion [6]

3.2 Point Dispersion

The idea of uniform interest point dispersion across the coordinates of an image is simple: covering of more regions in an image is more likely to contain useful descriptions. We partition the image plane into a grid of J bins. The measure of point dispersion is defined using the entropy value of the spatial distribution of detected points X over the image plane I.

$$\mathscr{D}(I, X) = -\sum P_j * log_2(P_j), \quad (3)$$

where P_j is approximated by the 2D histogram of the positions of interest points within bin j.

3.3 Information Content

Information content is the measure of the uniform distribution of feature vectors of the detected points. To maximize information content, therefore to avoid the loss of discriminatory power of descriptors constructed for the detected points, we have to penalize correspondences between the positional point dispersion and the descriptor space. We can achieve this by implementing the same principle as in point dispersion, but in the space of descriptors, which is illustrated in figure 4. We partition the descriptor space Γ into partitions Υ_j and approximate the probability of the occurrence of a descriptor within Υ_j by a histogram of the descriptors $\gamma \in \Upsilon_j$ as q_j. The measure of information content can then be formulated as

$$\mathscr{I}(\Gamma) = -\sum q_j * log_2(q_j). \quad (4)$$

The choice of the descriptor algorithm used in this step is crucial for a positive effect of the objective. In [13], the SIFT descriptor has been used, which has led toward counter-intuitive MO-GP results. The problem lies in the manner in which SIFT builds the feature vector. SIFT constructs histograms of gradient orientations for a region around the interest point. This is completely appropriate when using the SIFT algorithm for interest point detection as well. In our case, though, the descriptor component is used separately. It may happen that neighboring points within regions containing curves or circles end up being drastically different in their feature vectors. For this reason, the Hölder descriptor described next is used.

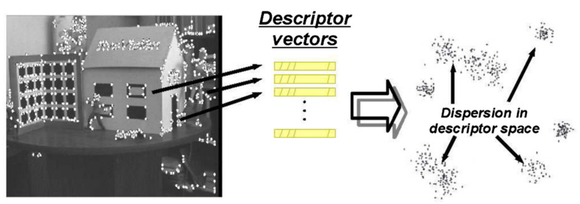

Figure 4: The effect of the objective of point dispersion on the interest point detector in a sample image [6]

3.3.1 Hölder descriptor

The Hölder descriptor is based on measuring the regularity of the region of an interest point. It follows the idea that most information is contained within irregular (singular) regions. The regularity of regions can be described by the pointwise Hölder exponent. In an image (2D signal), the exponent is defined as follows.

Pointwise Hölder exponent definition for 2D signal f
Let $f : \mathbb{R}^2 \to \mathbb{R}$, $s \in \mathbb{R}^{+\star} \setminus \mathbb{N}$ and $x_0 \in \mathbb{R}^2$. Then $f \in C^s(x_0)$ if and only if $\exists \eta \in \mathbb{R}^{+\star}$, and a polynomial P of degree $< s$ and a constant c such that

$$\forall x \in B(x_0, \eta), |f(x) - P(x - x_0)| \leq c|x - x_0|^s, \quad (5)$$

where $B(x_0, \eta)$ is the neighborhood of x_0 with radius η. Now, the pointwise Hölder exponent of f at x_0 is defined as

$$\alpha_p(x_0) = \sup_s \{f \in C^s(x_0)\}. \tag{6}$$

The concept is better grasped when considering 1D signal as shown in figure 5. $\alpha_p(x_0)$ is the bound on the Hölder envelope—the amount by which a signal varies. Values close to zero indicate a wildly varying signal, while values close to one represent a smooth signal or a regular region. Estimators of the Hölder exponent based on genetic programming are proposed in [10]. In our work, we use the HGP-2 estimator. The application of the Hölder exponent to an image is depicted in Figure 6.

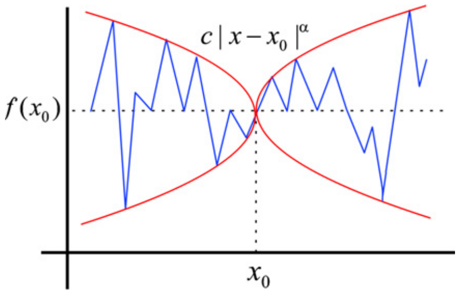

Figure 5: Visualization of the plot of the Hölder envelope of 1D signal f at point x_0 [10]

The Hölder descriptor is built by sampling the exponent at the position of each detected interest point and at equidistant positions lying on four circles of different radii around the interest point, with 32 samples per each circle. This yields a feature vector of 129 real numbers. While it is computationally costly to process such real-valued vectors in the matching phase, this problem does not concern our aim, since we only use this descriptor for the evaluation of synthesized interest point detectors, which is performed offline. The structure of the descriptor is shown in Figure 7.

Figure 6: The Hölder exponent applied to an image. The histogram of the result has been equalized for better visualization. One detected interest point is highlighted whose detail is shown in Figure 7.

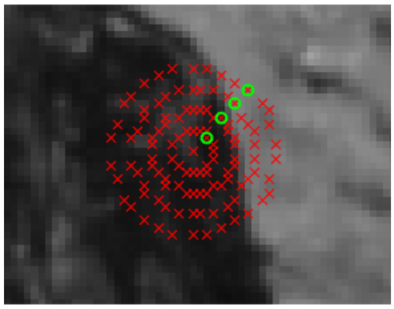

Figure 7: The structure of the Hölder descriptor applied to an image. The descriptor is formed by sampling the Hölder exponent in the image region centered at an interest point, and at 32 points per circle of 4 different radii around the interest point. The dominant gradient orientation is highlighted.

3.4 Computational Complexity

Our proposal stands on the definition of the objective promoting computational efficiency of the synthesized operators. As we focus on the implementation of the resulting operators on parallel processors[1], we should take the parallelizability of the algorithms into account.

The complexity theory defines complexity classes for problems considering their parallel nature. In particular, the class *NC* (Nick's Class) is defined as the set of problems decidable in *parallel* (polylogarithmic) time $(\log n)^{O(1)}$ on a polynomial number of processors $n^{O(1)}$ [1]. The problem with the approach using the complexity theory as a measure of fitness is that the definition is too rough for the distinction of the low-level operations used in this work (as explained in Section 4 and shown in equations 8 and 9). All of these operations are either performed independently on each point of an image, or on their local neighborhood. Therefore, all of these operations are trivially parallelizable.

The approach we propose is to empirically measure the time it takes for the synthesized algorithms to execute. This can be done in two ways.

The first involves executing the complete algorithm on a set of training images and measuring the total time of computation of the tree[2]. The disadvantage is that the measuring capabilities are dependent on and limited to the host machine performing the GP search, and hence all synthesized algorithms are biased toward it.

The second way which is used in our work is based on taking apriori measurements of each atomic operation we use in the GP search. We are not constrained to the host machine, as these atomic operations may be imple-

[1] Using the CPU, GPU, and DSP units present in a device in a heterogeneous manner.

[2] As generated by the MO-GP algorithm explained in Section 4

mented on the target platform containing multiple processing units, or within an emulated customizable environment. The measurements are also performed on a large set of diverse images.

Therefore, for all $op \in F \cup T$, where F and T are the sets of functions and terminals used in the GP (defined in Section 4) let $C_{op} \in \mathbb{R}$ be the measured average time it takes for op to execute. We define the computational complexity cost to be the sum of all atomic operations present in a computational tree A of the generated algorithm, as formulated in Equation 7. Note that this is a measure we are trying to minimize.

$$\mathscr{C}(A) = \sum_{op \in A} C_{op}. \tag{7}$$

4 Genetic Programming

Genetic programming is a branch of evolutionary algorithms performing symbolic regression. It is a biologically inspired optimization method based on the iterative stochastic generation of computer programs to perform a given task. With each iteration, a *population* of computer programs (computational trees) is generated or modified in a manner that aims to produce better results by preferring individuals with better survivability or *fitness* to solve the problem at hand.

This approach is similar to the genetic algorithms. Here, however, each individual consists of a computational tree with a dynamic, flexible structure. The computational trees are composed of nodes which are either *functions* (internal nodes) or *terminals* (leaves). These form the search space of the GP algorithm. The terminals act as input data that traverse upward across the tree, being transformed by every function they pass through. When the data reach the root node, the computation stops, yielding a result. Each individual in the population in this work is an image interest operator constructed by this form of symbolic regression.

In our work, we use the sets of functions F and terminals T as shown in equations 8 and 9.

$$\begin{aligned} F = \{&+, |+|, -, |-|, |I_{out}|, \times, \div, I_{out}^2, \sqrt{I_{out}}, \\ &log_2(I_{out}), k * I_{out}, \frac{\partial}{\partial x} G_D, G_{\sigma=1}, G_{\sigma=2}\}, \end{aligned} \tag{8}$$

$$T = \{I, L_x, L_{xx}, L_{xy}, L_{yy}, L_y\}. \tag{9}$$

Here, I_{out} is either one of the terminals in T, or the result of any function in F. $k = 0.05$ is a constant, G_D is the application of the Gaussian smoothing filter along direction D, and L_u is the Gaussian image derivative along direction u.

The general simplified pipeline of the computation performed by the genetic programming search is described in the following pseudo-code:

```
P := generate_population()
while stop condition is not met:
    for individual I in P:
        Fit[I] := fitness(I)
    S := selection of individuals
         from P with the highest
         fitness
    crossover(S)
    mutation(S)
    E := selection of individuals
         from P with the lowest
         fitness
    P := P - E
```

When the GP search process is complete, the individuals of the population in the last iteration are considered to be good candidates of interest image operators.

An important remark is that the results of the GP search in out work are *interest point operators*. An interest point operator may be defined as a function $K(x) : \mathbb{R}^+ \to \mathbb{R}$. We may obtain the *interest image* by applying K to a regular image. Then, we say the point x is an interest point if the conditions in Equation 10 hold.

$$K(x) \geq max\{K(x_W) | \forall x_W \in W, x_W \neq x\} \wedge K(x) > h \tag{10}$$

Here, W is a square neighborhood around point x. These two conditions represent non-maxima suppression and thresholding, respectively. In this work, we do not use a fixed value of h, but rather we choose the 500 points with the highest response to the interest operator.

4.1 Multiobjective Approach

This work aims to design and implement a framework for obtaining results that are optimal with respect to multiple optimization functions, or objectives. The MO-GP system avoids the need of manual tuning of parameters of a single fitness function to achieve results. Instead, it comes with the flexibility of altering or inserting new objectives and handles the trade-off optimization in a consolidated way. Moreover, with a single run of the GP search algorithm, we are able to obtain several non-dominant (near-optimal) results, which saves us a lot of time, considering the computational complexity of the overall MO-GP search.

The principle of the multiobjective search is intertwined with Pareto's economic theory. Given k objectives, the multiobjective space in which trade-offs and domination relations are considered, is k-dimensional and individual samples are k-dimensional vectors. In this space, we focus on finding solutions that are Pareto-optimal, i.e. are not dominated by any other vector. Considering a maximization problem, an objective vector f^i dominates objective vector f^j if no component of f^i is smaller than its counterpart in f^j and at least one component is larger. This can be seen in Figure 8 which optimizes objectives $f1$ and

$f2$: the white samples are non-dominated, as there is no other sample in a dominant relation with them. On the other hand, the black samples are all dominated by at least one of the white samples. The set of non-dominated (near-optimal) solutions is called the *Pareto-front*.

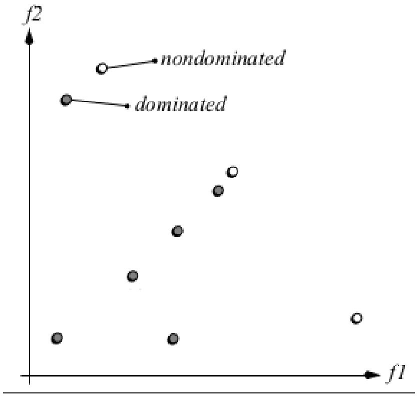

Figure 8: An example of a maximization SPEA2 search space [15]. The objective space specific to this work is depicted later in Figure 9. With all four objectives employed, this space is four-dimensional. The Pareto front represents the set of near-optimal IP detection operators that are efficient in terms of both object matching and computational complexity.

Approximation algorithms may have to deal with two problems: finding the true (optimal) Pareto-front and sampling the Pareto-front with uniform distribution across the objective space. Several multiobjective evolutionary algorithms (MOEAs) exist that solve these problems. In our work, we use Strength Pareto Evolutionary Algorithm 2 (SPEA2) [15].

5 Results

With an MO-GP framework deployed, we are able to conduct the experiments of automatic interest point operator synthesis, optimizing multiple objectives at the same time. In comparison with [6], we extend the objectives being optimized by our novel objective promoting less computationally expensive algorithms, as defined in Section 3. The solution proposed in [6] serves as a basis of comparison.

5.1 Environment Setup

To be able to objectively compare our results with the work in [6], we use the same environment and similar parameters of the MO-GP search process. The MO-GP parameters are presented in Table 1. There are only two differences. First, we set the crossover and mutation probabilities to 50%. Second, we do not limit the tree depth, as our novel objective already prefers results with lower computational cost, which is partially dependent on the tree size.

Parameter	Value
Population size	200
Generations	50
Initialization type	Ramped half-and-half
Crossover probability	0.5
Mutation probability	0.5
Mating selection	Binary tournament
SPEA2 archive size	100
SPEA2 selection size	100

Table 1: Table of parameters of the MO-GP search environment

The four fitness functions are formed of the quality measures defined in Section 3, where the first three are inverted so as to match the minimization goal. Also, the outputs of objectives 2 through 4 are empirically proportionally transformed and scaled to maintain a similar range of all four fitness values.

We use the GPLAB MATLAB toolbox [8] to perform the GP search, and the SPEA2 implementation available at the PISA website[3]. The training image dataset consisting of rotated Van Gogh images has been obtained from the Learning and Recognition in Vision team of Inria[4]. The repeatability rate MATLAB script from the Visual Geometry Group at Oxford University[5] has been used.

5.1.1 Computational costs

For our computational cost objective, we have empirically evaluated the costs of atomic operations as shown in Table 2. These costs have been measured by averaging multiple runs of the operations over a random subset of 500 images of size 256x256 from the SUN scene category database[6].

5.2 Comparison of Results

It has been expected that by extending the results in [6] with our novel objective, we yield interest operators with similar properties. The assumption is that computational complexity conflicts with other objectives in the Pareto front. This means that the more computationally efficient the operators are, the less compliant they are with the other objectives. Figure 9 shows the Pareto front of two objectives: point dispersion and stability, which gives an overview of the idea. Dispersion and stability are in conflict, which is a desired situation, as we obtain several near-optimal results with different trade-offs, all of which are useful for experimental work.

The assumption of obtaining operators with similar properties to those in [6] is confirmed as shown in figures

[3]http://www.tik.ee.ethz.ch/sop/pisa/
[4]http://lear.inrialpes.fr/people/mikolajczyk/
[5]http://www.robots.ox.ac.uk/~vgg/research/
[6]http://sun.cs.princeton.edu/

Operation	Time in milliseconds
f_abs	0.3237
f_const_times	0.3203
f_div	0.1594
f_gauss_1	1.6819
f_gauss_2	1.3688
f_gauss_x	1.5951
f_gauss_y	1.5466
f_log2	1.1669
f_minus	0.1358
f_minus_abs	0.1362
f_plus	0.1340
f_plus_abs	0.1385
f_square	0.3301
f_square_root	0.4282
f_times	0.1403
t_gauss_x	1.2296
t_gauss_xx	1.3685
t_gauss_xy	1.3422
t_gauss_y	1.1930
t_gauss_yy	1.3373

Table 2: Table of the evaluated computational cost of atomic operations

10 and 11. The synthesized operator in Equation 11 exposes high point dispersion fitness.

$$\frac{\partial}{\partial x} G_x(\log_2(G_{\sigma=2}(k \cdot L_y))) \tag{11}$$

This is similar to the *(c)* operator evolved in [6], as it also achieves high point dispersion[7] as depicted in Figure 9. This operator uses different tree nodes, but is of similar depth and is also comprised of Gaussian filter applications as shown in Equation 12:

$$G_{\sigma=2} * (\frac{L_y}{L_{yy}}) \tag{12}$$

6 Conclusion

The aim of this work has been to implement an MO-GP framework for the synthesis of interest point detectors. The framework has successfully been implemented in a way that allows for feasible modification and insertion of objectives.

We have extended the work of Olague and Trujillo [6] by designing and implementing a fourth objective optimizing computational efficiency. This objective has been designed in a way that focuses on client runtime (mobile devices, devices with parallel computation power) rather than the host machine, while at the same time offers easy means of redefinition of the measurement of computational cost if desired.

[7] Here, the fitness is minimized, so high point dispersion measurement represents a low value in the graph

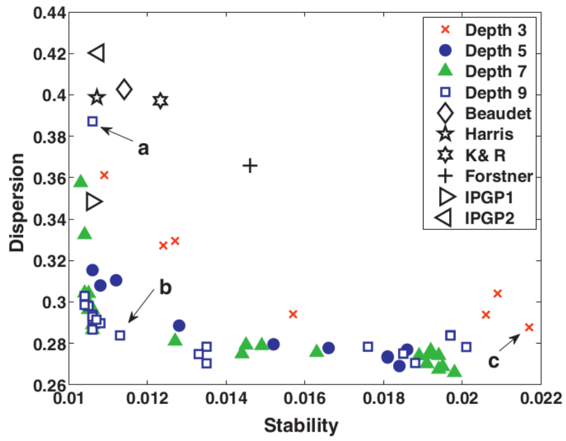

Figure 9: The Pareto front of stability and point dispersion in the work of [6]. The graph includes human-designed algorithms (Beaudet, Harris, K & R, Forstner) and operators resulting from SO-GP search presented in [11] (IPGP1, IPGP2).

In comparison, our results show high degree of similarity to those of the original work. Our intention was not to improve the results in the originally proposed three objectives, but rather to experiment by plugging the novel objective into our framework.

As mentioned in the introduction, our approach is still a work in progress. We have designed and developed the framework serving for conducting experiments. The parameters of the GP search are not definitive and heavily impact the results. Moreover, each run of the GP search algorithm takes roughly 24 hours of computation. For these reasons, we will be publishing noteworthy results of our experiments at a dedicated website[8].

6.1 Future work

The resulting operators of this work are yet to be implemented efficiently in the heterogeneous parallel computing platform Android RenderScript.

Also, to improve the recognition in cluttered scenes, a different training image set biased toward this phenomenon may be experimented with.

Finally, we recognize that this work may be extended to generate interest point detectors usable in object recognition in 3D scenes with very little work.

References

[1] Raymond Greenlaw, H. James Hoover, and Walter L. Ruzzo. *Limits to Parallel Computation: P-completeness Theory*. Oxford University Press, Inc., New York, NY, USA, 1995.

[8] http://davinci.fmph.uniba.sk/~uhliarik4/mogp/detector

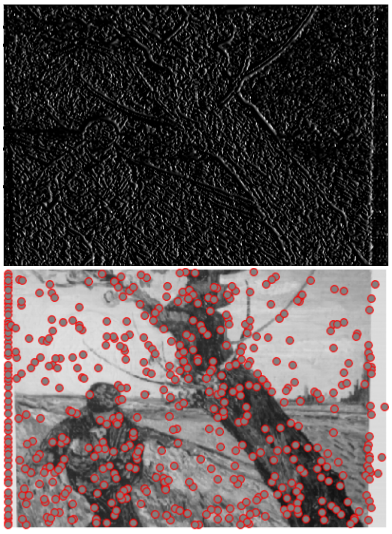

Figure 10: Interest image with detected points as the result of the synthesized operator in Equation 11, exhibiting high point dispersion measurement

(c)

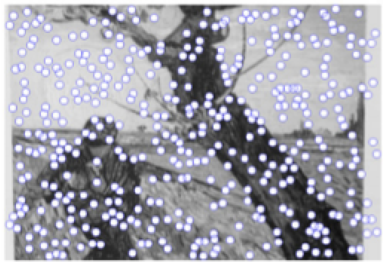

Figure 11: Interest image and points detected by the *(c)* operator synthesized by the MO-GP process in [6]

[2] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[3] Li Liu, Ling Shao, and Xuelong Li. Building holistic descriptors for scene recognition: A multi-objective genetic programming approach. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, pages 997–1006, New York, NY, USA, 2013. ACM.

[4] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[5] Gustavo Olague and Leonardo Trujillo. Evolutionary-computer-assisted design of image operators that detect interest points using genetic programming. *Image Vision Comput.*, 29(7):484–498, June 2011.

[6] Gustavo Olague and Leonardo Trujillo. Interest point detection through multiobjective genetic programming. *Appl. Soft Comput.*, 12(8):2566–2582, August 2012.

[7] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

[8] Sara Silva and Jonas Almeida. Gplab-a genetic programming toolbox for matlab. *Proceedings of the Nordic MATLAB conference*, pages 273–278, 2003.

[9] Stephen M Smith and J Michael Brady. Susana new approach to low level image processing. *International journal of computer vision*, 23(1):45–78, 1997.

[10] Leonardo Trujillo, Pierrick Legrand, Gustavo Olague, and Jacques Lévy-Vehel. Evolving Estimators of the Pointwise Holder Exponent with Genetic Programming. *Information Sciences*, 209:61–79, 2012. Submitted.

[11] Leonardo Trujillo and Gustavo Olague. Synthesis of interest point detectors through genetic programming. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, pages 887–894, New York, NY, USA, 2006. ACM.

[12] Leonardo Trujillo and Gustavo Olague. Automated design of image operators that detect interest points. *Evol. Comput.*, 16(4):483–507, December 2008.

[13] Leonardo Trujillo, Gustavo Olague, Evelyne Lutton, and Francisco Fernández de Vega. Multiobjective design of operators that detect points of interest in images. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, GECCO '08, pages 1299–1306, New York, NY, USA, 2008. ACM.

[14] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, July 2008.

[15] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm. Technical report, 2001.