Image Signal Processor parameter tuning with surrogate-assisted Particle Swarm Optimization

Georey Portelli, Denis Pallez

Image Signal Processor parameter tuning with surrogate-assisted Particle Swarm Optimization

Geoffrey Portelli[1] and Denis Pallez[2]

[1] Université Côte d'Azur, CNRS, I3S, Sophia Antipolis, France
geoffreyportelli@gmail.com
[2] Université Côte d'Azur, CNRS, I3S, Sophia Antipolis, France
denis.pallez@univ-cotedazur.fr

Abstract. Evolutionary algorithms (EA) are developed and compared based on well defined benchmark problems, but their application to real-world problems is still challenging. In image processing, EA have been used to tune a particular image filter or in the design of filters themselves. But nowadays in digital cameras, the image sensor captures a raw image that is then processed by an Image Signal Processor (ISP) where several transformations or filters are sequentially applied in order to enhance the final picture. Each of these steps have several parameters and their tuning require lot of resources that are usually performed by human experts based on metrics to assess the quality of the final image. This can be considered as an expensive black-box optimization problem with many parameters and many quality metrics. In this paper, we investigate the use of EA in the context of ISP parameter tuning with the aim of raw image enhancement.

Keywords: Image Signal Processor · Parameter tuning · Particle Swarm Optimization.

1 Introduction

Image processing has been largely investigated for many years. As the quality of outputs is crucial for the whole computer vision chain, sophisticated mathematical theories and statistical methods have been developed in recent years [7]. These are a source of complex optimization problems. Moreover, new constraints for embedded, real-time computer vision systems necessitate the development of robust and flexible as well as cost-effective algorithms.

Image-related optimization problems tend to be highly nonlinear and computationally expensive. The objectives and constraints are usually related to image metrics that need to be calculated based on the images from the output of the whole computer vision chain. Thus each evaluation of such objectives requires the combination of applying the considered image processing techniques (denoising, sharpening, white balance, etc...) with optimization-related techniques. Moreover the computational costs will increase with the size of the images. All these factors lead to high computational costs [18].

Classical image processing methods tuned with EAs have been shown to outperform manually tuned classical methods such as image filtering, denoising or enhancement [17], and object detection [5], in terms of convergence speed and ability to deal with highly degraded image source. However, because of its complexity (non-linear process, number of parameters, etc...) and computing resource needs, authors focused on the optimization of a unique task [3,12,18]. For instance, [3] investigated the application of the Artificial Bee Colony algorithm (ABC) [10] where the artificial bees are moving to search for the optimal parameters of a pixel luminance transformation function to enhance the image contrast based on the fitness function.

But nowadays, embedded computer vision systems can be found in lots of devices such as digital cameras and small devices as smartphones, tablets, etc. Usually coupled with the digital sensor, an ISP can be found and applies sequentially many processing and filtering to the raw images like noise reduction, sharpening, white balance or color correction and thus aims at enhancing the image quality of final pictures (Fig. 1). Depending on the application, ISPs can have approximately 6 to 14 processing stages and each stage can have tens or even hundreds of parameters.

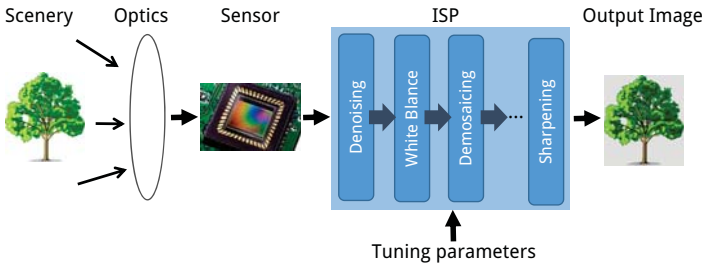


Fig. 1: Classical design of a digital vision system.

All those processing are based on methods that are parameterized. When enhancing only one raw image, many parameters have to be set and each of them could be continuous or discrete. Because of its size and its complexity, i.e. the amount of parameters to optimize (constrained and unconstrained) and several time consuming criteria, the problem to be solved combines fundamental issues like:

- large scale optimization: each stage embedded in the ISP can have few or even hundreds of parameters,
- constrained optimization: processing stages might have parameters that must fulfill specific inequality to be valid, for instance a specific relationship between two (or more) parameters or parameter values that must be within a given range,

- many objectives: several metrics can be used to asses the quality of the output image,
- expensive optimization: computational costs increase with the image size and ISP may need several minutes to process one raw image.

In addition, a gradient based search is not usable because the problem to optimize is a set of methods or procedures that might not be differentiable.

Usually, the parameters of ISPs are painstakingly tuned by a human expert based on image quality metrics. This painstaking work usually leads to one close-to-optimal solution and must be done for each product and variant, severely limiting the number of new configurations that can be evaluated.

Although EAs have been applied to image processing considering one unique task as showed previously, to our knowledge EAs have never been used to optimize a whole image processing chain made of several sequential processing stages as ISPs. Thus here, we investigate the usage of such EA to automatically optimize parameters of a typical ISP based on image quality metrics, for the enhancement of raw images.

2 Real-world problem: ISP

The optimization problem that is the tuning of the parameters of an ISP can be formally defined as following. Given a set of image transformations $\mathbb{T} = \{t_1, t_2, ..., t_k\}$, one ISP φ is one element of the power set of $\mathbb{T}$ ($\varphi \in \mathcal{P}(\mathbb{T})$). $|\varphi|$ represents the number of transformations that will be applied on a raw image I_R by φ. $\mathbb{P}$ is the set of parameters used by all $t_i \in \varphi$. As each transformation t_i is parameterized by a specific set of parameters $p_i \subseteq \mathcal{P}(\mathbb{P})$, a function p is defined and associates a transformation with its corresponding set of parameters, i.e. $p(t_i) = p_i$. Thus, φ is defined by $\mathbb{P}' = \bigcup_{t_i \in \varphi} p(t_i)$ and, the output image I obtained by applying all transformations contained in φ is defined by $I = \varphi(I_R, \mathbb{P}')$. Some image quality metrics m_i are measured on resulting image I.

The goal is to identify the parameter values of $\mathbb{P}'$ that will optimize all m_i metrics. Without any loss of generality, this optimization problem can be formalized by Eq. 1:

$$\begin{aligned} \underset{x \in \mathbb{R}^{|\mathbb{P}'|}}{\arg\min} \quad & M(x) = \{m_1(x), \ldots, m_n(x)\} \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 0, \ldots, m \quad m \geq 0, \\ & h_j(x) = 0, \quad j = 0, \ldots, p \quad p \geq 0 \end{aligned} \tag{1}$$

where g_i and h_i are respectively inequality and equality constraints on transformations parameters.

3 Objectives: image quality metrics

In the domain of camera-equipped mobile devices, a standardized suite of objective and subjective image quality metrics is available [1]. This suite also provides

standardized image targets that allow the metrics to be comparable between different camera devices. Here are considered the visual noise and the visual acutance (as defined in [1]) to assess the noise and the acutance of the image, respectively. Indeed, a human observer with little experience can visually perceive if an image subjectively looks soft or sharp, also if it looks really noisy or not. Thus, a visual validation of solutions given by an EA can be done contrary to more "abstract" metrics such as Root Mean Square Error or Signal over Noise Ratio that are informative but difficult to sense with human eyes, for instance.

The image noise can be view as the random variation of brightness or color information in images, usually related to the electronic noise. It can be produced by the digital sensor and circuitry in digital camera. The acutance of an image describes a subjective perception of sharpness that is related to the edge contrast of an image. It is linked to the amplitude of the derivative of brightness with respect to space.

The output value of visual noise function will actually decrease as the noise decrease in the measured image. Inversely, the output of the visual acutance function will increase with the sharpness of the measured image: the more the sharpness, the greater the visual acutance. One has to note that noise and acutance counteract with each other.

From a human point of view, a high value of visual acutance does not necessarily implies a subjectively good looking image. Preliminary tests with manual tuning of the ISP used here (data not shown) show that values of visual acutance $\sim$102.0 (and visual noise $\sim$0.01) gives best looking images. Thus not the raw value of the visual acutance function but its transformation $(x - 102.0)^2$ will be used as objective to be minimized. The visual noise as it is will be minimized.

4 Particle Swarm Optimization

The aim of the presented study is not to compare EAs nor to find the best EA for the considered problem. Here the aim is to test whether or not this kind of real-world problem can be tackled using EA. Among the numerous EA strategies that have been investigated by the community, here in the considered real-world application, we focus on Particle Swarm Optimization algorithms (PSO). However others EA strategies could be considered.

PSO is a bio-inspired meta-heuristic mimicking the social behavior of bird flocking or fish schooling [11] which has become very popular to solve multi-objective [15] and many-objective [6] optimization problems. Briefly, swarm particles will evolve in the solution space where their position will be updated at each iteration given their own vector speed computed from global best and local best encountered solutions.

PSO has many advantages, for instance, easy implementation, effective memory, efficient maintenance of the solution diversity [19]. Also, PSO makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. PSO does not use the gradient of the problem being optimized, which means PSO does not require that the optimization problem

be differentiable as is required by classic optimization methods such as gradient descent. However, as many others metaheuristics, PSO do not guarantee an optimal solution is ever found.

Among all the possible variants, we focus on SMPSO [14] to solve the optimization of the ISP. Its main features are :

- use of a strategy to limit the velocity of the particles,
- external archive to store the non-dominated solutions (leaders) and a density estimator (crowding distance),
- leader selection with a binary tournament from the leaders archive taking into account the crowding distance,
- mutation operator (polynomial mutation) that add turbulence.

It has been shown to exhibit high performances in various benchmarks and real-world problems [14], and suitable for large scale problems up-to 2048 variables [13]. Briefly as described in [14], the pseudo code of SMPSO is given in Algo. 1.

Algorithm 1: SMPSO pseudo-code (from [14])

```
initializeSwarm()
initializeLeadersArchive()
generation = 0
while generation < maxGenerations do
    computeSpeedVector()
    updateParticlesPosition()
    mutation()
    evaluation()
    updateLeadersArchive()
    updateLocalBest()
    generation ++
end
returnLeadersArchive()
```

In the original publication, the performances of SMPSO were stated on benchmark problems after 25000 function evaluations [14]. Such a number of function evaluations seems to be not compatible with expensive problems. Motivated by decreasing the computational costs in evolutionary optimization of expensive problems, Surrogate-Assisted evolutionary computation has been highly investigated (see for review [9]). Basically, surrogates are used together with the real fitness function with the aim of estimating the fitness of new individuals of which can be then selected upon a given criteria for a real evaluation. As the real-world problem presented here is expensive, we modified SMPSO in order to add surrogates model (SASMPSO). The difference is that, during the evolutionary algorithm, the swarm particles will search according to their estimated evaluation thanks to surrogate, for a given amount of resources ($maxNfeSurrogate$). Once those resources are depleted, all the current swarm particles are evaluated using the real fitness function and then used in the evolutionary algorithm.

Here, there is no particular selection of best candidates for real evaluations. The pseudo code of SASMPSO is given in Algo. 2.

Algorithm 2: SASMPSO pseudo-code

```
initializeSwarm()
initializeLeadersArchive()
generation = 0
computeSpeedVector()
updateParticlesPosition()
mutation()
evaluation()
updateLeadersArchive()
generation ++
while generation < maxGenerations do
    selectSurrogates()
    trainSurrogates()
    nfeSurrogate = 0
    while nfeSurrogate < maxNfeSurrogate do
        computeSpeedVector()
        updateParticulesPosition()
        mutation()
        estimationWithSurrrogate()
        updateLocalBest()
        nfeSurrogate ++
    end
    evaluation()
    updateLocalBest()
    updateLeadersArchive()
    generation ++
end
returnLeadersArchive()
```

Upon the real evaluation of the initial swarm population, surrogates used here are constructed and updated for each objectives, separately, following the method described in [8]. Briefly, four models as Gaussian Process, Radial Basis Function, and Multivariate polynomial regression of degree 1 and 2 have been used here even if other surrogate models can be considered too. First, surrogates are selected based on the minimum root mean-squared error while performing a crossvalidation with 80% of samples for the training set and 20% of samples for the testing set (function *selectSurrogates* in Algo.2). Then, the selected surrogate is trained considering all the samples (function *trainSurrogates* in Algo.2). The samples are from an external archive different from the archive containing the leaders. This archive is updated with all the real-evaluated individuals and will grow until the algorithm stops. Whenever this archive is updated with new real-evaluated individuals, surrogates are re-selected and re-trained with the aim of getting the most accurate estimation.

4.1 Encoding/decoding parameters

For simplicity, parameters are encoded as float valued parameters in the range $[0, 1]$. In order to deal with the different ranges of the ISP parameters (see section 5.1), a function is used to convert float values in $[0, 1]$ to proper parameters values compatible with the experimental ISP. To do so, Eq. 2 was used. Regarding the particular dependence between the parameters k and N of the denoising filter, Eq. 3 was used to compute the values of each k and N parameters before feeding the experimental ISP. One can note that N will not be tuned directly but k and n will be, thanks to Eq. 3.

$$x = (x_e * (max(x) - min(x))) + min(x)\,, \text{ with } \begin{cases} x \in [\text{original range}] \\ x_e \in [0, 1] \end{cases} \tag{2}$$

$$N = 4 * n + k\,,$$
$$\text{with } \begin{cases} k = 2 * (round((k_e * (max(k) - min(k))) + min(k))) - 1\,,\ k_e \in [0, 1] \\ n = round((n_e * (max(n) - min(n))) + min(n))\,,\ n_e \in [0, 1] \text{ and } n \in [2, 5] \end{cases} \tag{3}$$

4.2 Function evaluation

Algorithm 3: Function evaluation

f $(rawImage, param_e)$;
Input : Raw image, encoded ISP parameters
Output: *acutance*, *noise*
$param = decode(param_e)$
$srbg = experimentalISP(rawImage, param)$
$acutance = (visualAcutance(srgb) - 102)^2$
$noise = visualNoise(srgb)$

The function evaluation is defined as in Algo. 3. It takes as input the raw image ($rawImage$) and the ISP parameters. The encoded parameters ($param_e$) are decoded with the *decode* described function, into proper ranged values before feeding the experimental ISP. The output of the ISP, i.e. the processed image *srgb* is then used to compute the image quality metrics with the functions $visualAcutance$ and $visualNoise$. This gives the values of the two objectives that will be minimized by the EAs. One can note that the computation time of one function evaluation will be the sum of the ISP processing a raw image plus the metrics measure, that is 400 seconds in average.

5 Experimental methods

Because of its complexity and as a first step, we propose to simplify the problem of ISP tuning by using EAs to optimize the parameters of a two-filtering stages ISP thanks to two image quality metrics. We designed a simplified two-filtering stages ISP that can be tuned according to 8 parameters, with respect to the minimization of 2 objectives. The parameters are related to the denoising and sharpening filters that will be detailed here after. As previously described, objectives that are minimized are the visual noise and the visual acutance. In the following we detail the experimental ISP, the implemented image filters and their parameters.

5.1 The experimental ISP

As ISPs used in industry feature hundreds of parameters and are not open source, a simpler prototype has been developed. Based on commonly used procedures of raw image processing [16], we implemented a simple but fully functional ISP. Here as a proof of concept, only denoising and sharpening stages will be considered.

The denoise filter is based on the non-local means algorithm (NLM) proposed by [2]. Briefly, NLM filtering takes a mean of all pixels in the image, weighted by how similar these pixels are to the target pixel. This results in much greater post-filtering clarity, and less loss of detail in the image. Here is considered especially the implementation suggested in [4]. It depends on five parameters $\{\alpha, \beta, k, N, thr\}$ which are the lower/upper values $[\alpha, \beta]$ for the noise profile of the sensor itself that can vary in the 20 % range, k and N the size of the patches of the denoise filter, and thr a threshold.

The sharpening stage is based on the commonly used unsharp-mask technique [7] where the original image is combined to a mask obtained from the negative image that is blurred, or "unsharp". This combination creates an output image that is less blurry than the original. This unsharp-mask depend on three parameters $\{radius, amount, threshold\}$ which are mainly related to the parameters of the Gaussian function used to blur the original image. All the eight parameters and their ranges are detailed in Table 1.

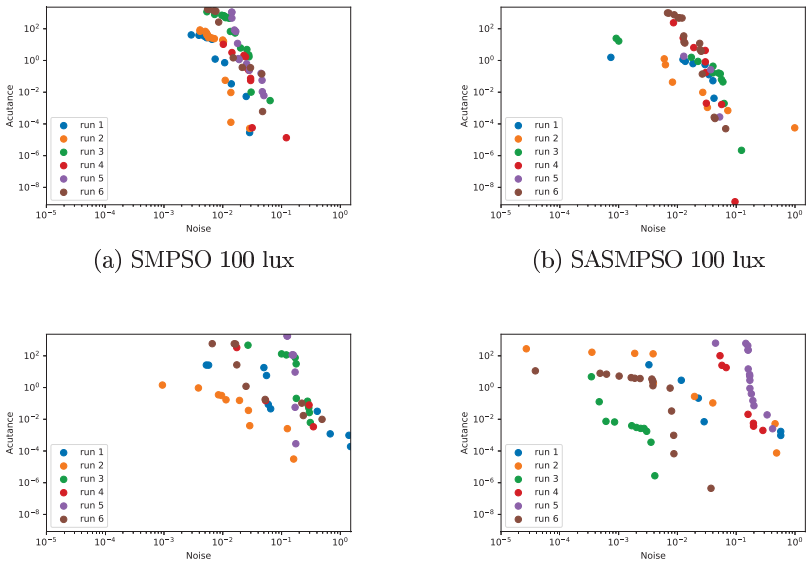
5.2 Raw images

Nowadays, lots of imaging devices allow access to the raw output of the digital sensor such as digital single-lens reflex camera and even some recent smartphones with high-end photographic capabilities. Thanks to the suite [1], pictures of image target suitable for measuring the visual noise and the visual actuance (standard deadleaves target) have been made in a controlled environment. Especially, two raw images of this target, one in low-light condition (20 lux) and one in normal-light condition (100 lux) have been considered. Typically, in low-light condition, the image is more noisy and should be more difficult to process than the normal-light condition image. Thus, the EA will optimize the parameters of

Table 1: Tunable parameters of the experimental ISP for denoising and sharpen filters.

Name	Range	type
α	$\alpha \pm 20\%$	float
β	$\beta \pm 20\%$	float
k	$[3, 5, 7, 9]$	integer
N	$N = f(k)$, with $N \leq 37$	integer
thr	$[0, 2]$	float
radius	$[0, 3]$	float
amount	$[0, 2]$	float
threshold	$[0, 1]$	float

the experimental ISP considering one of the two images at a time and results will be compared.



(a) SMPSO 100 lux

(b) SASMPSO 100 lux

(c) SMPSO 20 lux

(d) SASMPSO 20 lux

Fig. 2: Pareto fronts for each independent runs (one color per run), and for each algorithm and raw image combination, after 1000 nfe.

5.3 Performance evaluation

In this study, both SMPSO and the proposed SASMPSO are compared for the optimization of the experimental ISP parameters considering two raw images (100 lux and 20 lux).

Hyperparameters are the same for both algorithms and raw images. The initial population size and the leaders archive size are set to 50. When surrogates are involved, the maximum number of evaluations using the surrogates (*maxNfeSurrogate*) is set to 1000. The maximum number of real evaluations allowed is set to 1000. Of course one could also vary those hyperparameters to analyze the sensitivity of the algorithms. But this is not the aim of the presented study. Six independent runs are performed. We agree that such a low number of runs is not enough to get statistically significant conclusions, and will need to be completed. But here are preliminary results that we think are already interesting for the community.

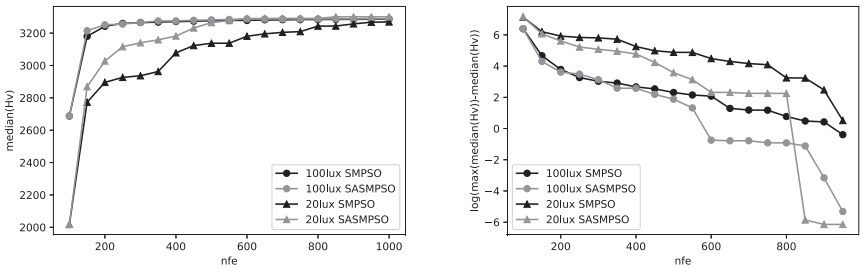
As the considered problem is black-box, the true Pareto front is unknown. Nevertheless, the performance between algorithms can be compared visually by the cumulative Pareto front across the 6 independent runs obtained after 1000 real evaluations, i.e. the non-dominated solutions over the concatenated solutions of the 6 runs. The idea beyond this is to have an overview of the exploration of the solution space.

The hypervolume indicator is computed to assess the differences in diversity and convergence. The hypervolume is the area between the non-dominated solutions and a reference point. Especially due to the low number of independent runs available, not the average but the median of the hypervolumes computed for the 6 runs is considered and is displayed as a function of the number of real function evaluations (nfe). The aim here is to provide a measure of central tendency: what is the most representative behavior of the hypervolume across the 6 runs. With such a low number of runs, the average will be highly sensitive to extreme values and variability unlike the median.

6 Results

Pareto fronts for each independent runs can be seen in Figure 2. One can note the variabilty between the runs of one algorithm given one raw image, especially for SASMPSO 20 lux (Fig. 2d). Because of these discrepancies and the low number of available runs, not the mean but the median of the hypervolume is used.

For each combination algorithm-raw image, the hypervolume is computed for each run and the median is plotted as a function of the increasing number of nfe (Fig. 3a). For all, the hypervolume is increasing with the nfe showing that there is minimization of the two objectives. One can speculate that there is convergence toward regions of the solution space containing better solutions. Also here, little differences can be seen between the two algorithm while considering the 100 lux raw image. SASMPSO seems to perform better than SMPSO while considering the 20 lux raw image. But as a squared difference is used as



(a) (b)

Fig. 3: (3a) median of the hypervolumes for the 6 independent runs. (3b) *log* of the difference between the maximum of the median of hypervolumes and the median of hypervolumes. SMPSO is in black, SASMPSO is in grey, and for the raw images, 100 lux is circle markers and 20 lux is triangle markers.

the Acutance fitness (see 3), objective values evolve within a huge range. As a consequence, the reference point use to compute the hypervolume is far from the final solutions and thus the differences in hypervolumes that could be observed toward the "convergence" are smothered. In order to reduce this bias, is represented the *log* of the difference between the maximal median hypervolume and the considered hypervolume ($log(max(medians) - median)$) in Figure 3b. As a result, it can be seen the decrease toward the highest hypervolume, i.e. the faster the decrease, the better. Thus, it is highlighted that SASMPSO (grey) is converging faster than SMPSO (black) toward this best encountered hypervolume. Also in both algorithms, the decrease toward the highest hypervolume seems to be slower while considering the 20 lux image compared to the 100 lux image (both circles versus both triangles curves until 800 nfe, Figure 3b). This supports the *a priori* about the increased difficulty of the processing due to the inherent higher noise level in the 20 lux image.

Figure 4 shows the cumulative Pareto fronts across the 6 independent runs, after 1000 nfe for SMPSO (black), SASMPSO (grey), considering the 20 lux raw image (triangles) or the 100 lux raw image (circles). The objectives are in *log* scale. Both algorithms reach solutions in the range of expected values for both objectives. The use of surrogates in SASMPSO seems to lead to a better convergence for both raw images. Even if the differences are low for the 100 lux raw image, SASMPSO seems to outperform SMPSO for the enhancement of the 20 lux raw image, although the processing of the 20 lux raw image was *a priori* more difficult to optimize due to its inherent higher noise level compared to the 100 lux raw image. For illustration, crops of the raw image and the processed image related to the kneepoint solution of the cumulative Pareto front of each conditions are showed in Figure 5.

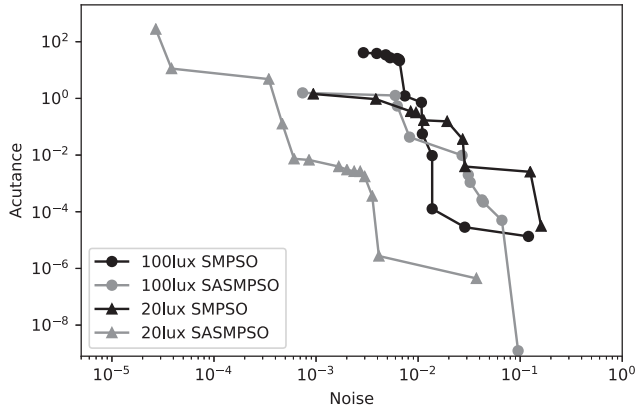
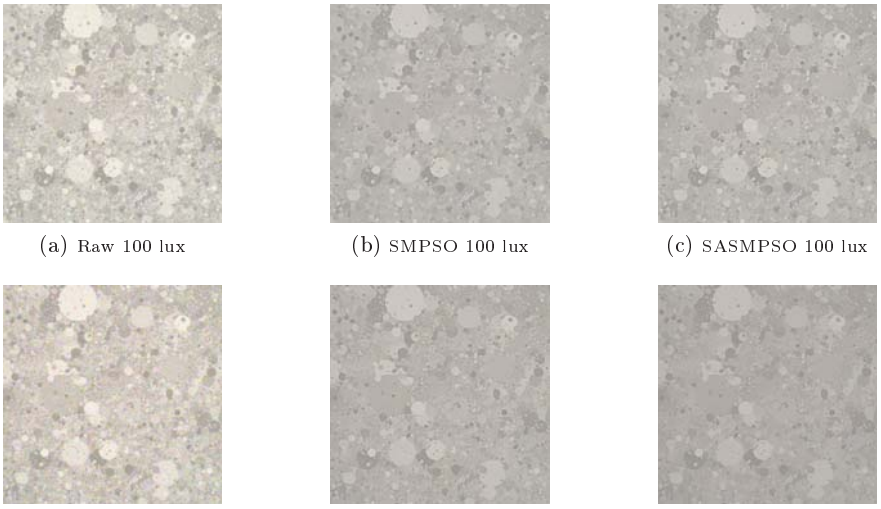


Fig. 4: Cumulative Pareto fronts across 6 independent runs after 1000 function evaluations for each algorithms and raw images combination. SMPSO is in black, SASMPSO is in grey. And for the raw images, 100 lux is circle markers and 20 lux is triangle markers.

7 Discussion and conclusions

We presented in this study the use of an EA to optimize the parameters of a set of filters for the image enhancement, i.e. for the optimization of an ISP. Commonly in the literature, EAs were used in the context of image processing for the optimization or the design of single filters for image enhancement [3,12,18,5]. To our knowledge, this is the first time that EAs are used to optimize a set of filters for image enhancement. Among the diversity of available EAs we focused on PSO, especially SMPSO [14] and the proposed simple surrogate-assisted SMPSO, SASMPSO, to optimize the parameters of an experimental ISP for the enhancement of raw images in normal-light and low-light conditions.

Despite the fact that the presented results can be considered as preliminary, results show that both EA succeeded in converging toward solutions in the expected range of values for the objectives, for both light conditions. Results show also that the use of surrogates allows to "converge" quicker. One could argue that only a simplified ISP have been used here and actual complete ISP are far more complex and challenging. We agree that the presented results obtained using the simplified ISP need to be confirmed with complementary experiments using more complex ISP (more filters, more parameters, more image quality metrics). Indeed, besides the black-box and expensive aspects, complete ISPs can have approximately tens processing stages, each of them parameterized with tens or even hundreds of parameters. Moreover, one could consider that the sequence of processing stages itself, can be optimized which highly increase the complexity of the problem.



(a) Raw 100 lux (b) SMPSO 100 lux (c) SASMPSO 100 lux

(d) Raw 20 lux (e) SMPSO 20 lux (f) SASMPSO 20 lux

Fig. 5: Raw images and processed images related to the kneepoint solutions of the Pareto fronts showed in Figure 4

So, the final problem combines many of current challenges as black-box, expensive, large scale, many objectives optimization which have been and are currently investigated separately or partially combined, by the community. Going toward the optimization of a complete ISP will require the development of new surrogate-assisted EAs that can handle all those challenges at once.

References

1. Ieee cpiq 1858-2016 - ieee standard for camera phone image quality (2016), `https://standards.ieee.org/standard/1858-2016.html`
2. Buades, A., Coll, B.: A non-local algorithm for image denoising. In: In CVPR. pp. 60–65 (2005)
3. Chen, J., Yu, W., Tian, J., Chen, L., Zhou, Z.: Image contrast enhancement using an artificial bee colony algorithm. Swarm and Evolutionary Computation **38**, 287–294 (2018)
4. Darbon, J., Cunha, A., Chan, T.F., Osher, S., Jensen, G.J.: Fast nonlocal filtering applied to electron cryomicroscopy. In: 2008 5th IEEE International Symposium on biomedical imaging: from nano to macro. pp. 1331–1334. IEEE (2008)
5. Ebner, M.: Engineering of computer vision algorithms using evolutionary algorithms. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) Advanced Concepts for Intelligent Vision Systems. pp. 367–378. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
6. Figueiredo, E., Ludermir, T., Bastos-Filho, C.: Many objective particle swarm optimization. Information Sciences **374**, 115 – 134 (2016).

https://doi.org/https://doi.org/10.1016/j.ins.2016.09.026, http://www.sciencedirect.com/science/article/pii/S0020025516308404
7. Gonzalez, R.C., Woods, R.E.: Digital image processing. third edit (2008)
8. Habib, A., Singh, H.K., Chugh, T., Ray, T., Miettinen, K.: A multiple surrogate assisted decomposition based evolutionary algorithm for expensive multi/many-objective optimization. IEEE Transactions on Evolutionary Computation (2019)
9. Jin, Y.: Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm and Evolutionary Computation **1**(2), 61–70 (2011)
10. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Tech. rep. (2005)
11. Kennedy, J., Eberhart, R.: Particle swarm optimization (pso). In: Proc. IEEE International Conference on Neural Networks, Perth, Australia. pp. 1942–1948 (1995)
12. Munteanu, C., Rosa, A.: Towards automatic image enhancement using genetic algorithms. In: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512). vol. 2, pp. 1535–1542. IEEE (2000)
13. Nebro, A.J., Durillo, J.J., García-Nieto, J., Barba-González, C., Del Ser, J., Coello, C.A.C., Benítez-Hidalgo, A., Aldana-Montes, J.F.: Extending the speed-constrained multi-objective pso (smpso) with reference point based preference articulation. In: International Conference on Parallel Problem Solving from Nature. pp. 298–310. Springer (2018)
14. Nebro, A.J., Durillo, J.J., Garcia-Nieto, J., Coello, C.C., Luna, F., Alba, E.: Smpso: A new pso-based metaheuristic for multi-objective optimization. In: 2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM). pp. 66–73. IEEE (2009)
15. Reyes-Sierra, M., Coello, C.C., et al.: Multi-objective particle swarm optimizers: A survey of the state-of-the-art. International journal of computational intelligence research **2**(3), 287–308 (2006)
16. Rob, S.: Processing raw images in matlab (2014), http://rcsumner.net/raw_guide/RAWguide.pdf
17. Sahoo, M.: Classical and evolutionary image contrast enhancement techniques: Comparison by case studies. In: Computational Intelligence in Data Mining, pp. 37–44. Springer (2017)
18. Yang, X.S., Papa, J.P.: Bio-inspired computation and applications in image processing. Academic Press (2016)
19. Zhang, Y., Gong, D.W., Geng, N.: Multi-objective optimization problems using cooperative evolvement particle swarm optimizer. Journal of computational and theoretical nanoscience **10**(3), 655–663 (2013)