Swarm Intelligence for Automatic Color and Contrast Retrieval of Digital Images of Paintings

Akemi Gálvez
Toho University
Funabashi, Japan
University of Cantabria
Santander, Spain
galveza@unican.es

Eneko Osaba
TECNALIA, Basque Research & Technology Alliance (BRTA)
Derio, Spain
eneko.osaba@tecnalia.com

Iztok Fister
Faculty of Elect. Eng. & Comp. Sci.
University of Maribor
Maribor, Slovenia
iztok.fister2@uni-mb.si

Andrés Iglesias*
Toho University
Funabashi, Japan
University of Cantabria
Santander, Spain
iglesias@unican.es
*Corresponding author

Javier Del Ser
TECNALIA, Basque Research & Technology Alliance (BRTA)
Derio, Spain
University of Basque Country
Bilbao, Spain
javier.delser@tecnalia.com

Iztok Fister Jr.
Faculty of Elect. Eng. & Comp. Sci.
University of Maribor
Maribor, Slovenia
iztok.fister1@um.si

***Abstract*—We address the following problem: given an initial high-quality reference image and a variation of it, how to compute suitable values for color map and contrast such that, when applied to this variation, we get an image very similar visually to the reference image. This problem can be formulated as an optimization problem. Unfortunately, this leads to a continuous nonlinear optimization problem too difficult to handle by classical mathematical optimization techniques. To tackle this issue, we apply a powerful swarm intelligence method called cuckoo search algorithm. The method is tested on an illustrative example of a famous painting by artist Vincent Van Gogh. The experimental results show that the method performs very well, with a similarity error rate between the reference and the reconstructed images of only 5.73%. The method can be applied to any variation of the original painting regardless of its initial color map and contrast.**

***Index Terms*—artificial intelligence, swarm computation, cuckoo search algorithm, image processing, color retrieval.**

I. The Problem

Consider an initial digital image, called the *reference image*. Then, suppose we are given a variation of the reference image generated by changing its color map and other visual attributes in an unknown way. Our goal is to obtain optimal parametric values for some of such visual attributes such that, when applied to the variation, we obtain an image very similar to the reference image, according to a given metric. Here we restrict to only two visual attributes: color and contrast.

Color is one of the most important visual attributes of an image. Color models are represented by tuples of numbers [2]. In this paper we consider the RGB color model, where the different colors are defined as triplets $[R, G, B]$, representing the respective red, green, and blue color channels. We consider the values for each channel to be normalized on the unit interval $[0, 1]$, i.e., $R, G, B \in [0, 1]$. *Contrast* (denoted as C) is a visual attribute accounting for our ability to distinguish an object from others within the same field of view. There are many different definitions of contrast. In this paper we consider the *RMS (root-mean square) contrast*, given by:

$$C = \sqrt{\frac{1}{(M-1).(N-1)} \sum_{i=1}^{M} \sum_{j=1}^{N} (I_{ij} - \hat{I})^2} \qquad (1)$$

where (M, N) is the image size (in pixels), I_{ij} is the intensity of the pixel (i, j) of the image and $\bar{I}$ is the average intensity for all pixel values in the image. We assume that all intensities are normalized on the unit interval $[0, 1]$.

Let $\mathcal{R}$ denote the *reference image*, of size $M \times N$ (in pixels). It is represented by a $M \times N$ matrix of $[R, G, B]$ triplets, such that the color of pixel (i, j) is given by the triplet $[R_{ij}, G_{ij}, B_{ij}]$. $\mathcal{R}$ has a contrast parameter, C, given by Eq. (1). Let us consider now a variation $\mathcal{V}$ obtained from $\mathcal{R}$ after its RGB color map and other visual attributes are modified in an unknown way. We seek to determine the optimal values for the RGB color code and contrast such that, when applied to $\mathcal{V}$ via a transformation φ, the resulting image, $\mathcal{V}' = \varphi(\mathcal{V})$, is similar to $\mathcal{R}$, i.e., $\mathcal{R} \approx \mathcal{V}'$, according to a given metric defined by a similarity function, ϕ, as:

$$\phi(\mathcal{R}, \mathcal{V}') = \lambda \sum_{i=1}^{M} \sum_{j=1}^{N} \left(\mathbf{\Delta}_{ij} - \mathbf{\Delta}'_{ij}\right)^2 + \mu(C - C') \qquad (2)$$

where the superscript $'$ denotes the output image, $\mathbf{\Delta}_{ij} = (R_{ij}, G_{ij}, B_{ij})|_{\mathcal{R}}$, $\mathbf{\Delta}'_{ij} = \left(R'_{ij}, G'_{ij}, B'_{ij}\right)|_{\mathcal{V}'}$, and λ and μ are scalar weights, one of which can be assumed to be unitary for simplicity. We denote as $\psi(\mathcal{R}, \mathcal{V}')$ the normalized version

2642-3596/20/$31.00 ©2020 IEEE
DOI 10.1109/CW49994.2020.00043

TABLE I: Cuckoo Search Algorithm with Lévy Flights.

```
begin
    Fitness function h(x), x = (x_1, ..., x_D)^T
    Generate initial population of P host nests x_i (i = 1, 2, ..., P)
    while (t < N_gen)
        Choose a cuckoo (say, i) randomly using Lévy flights
        Computer its fitness, F_i
        Select a new nest (say, j) randomly
        if (h(x_i) > h(x_j))
            Replace x_j by x_i
        end
        An amount (p_a) of poor nests are abandoned; new
                nests are built using Lévy flights
        Preserve the best solutions (elitism)
        Rank all solutions to determine the best one
    end while
    Return the best solution and fitness value
end
```

of function of $\phi(\mathcal{R}, \mathcal{V}')$, where values close to 0 mean that both images are very similar, and the value 0 indicating that they are visually identical. Then, the problem can be expressed as:

$$\underset{\{\boldsymbol{\Delta}', C'\}}{\text{minimize}} \ \left[\psi(\mathcal{R}, \mathcal{V}')\right] \tag{3}$$

Unfortunately, this is a very difficult nonlinear multivariate continuous optimization problem. The problem is so difficult that it cannot be solved by classical mathematical optimization techniques. To tackle this issue, we rely on the cuckoo search algorithm, described in next section.

II. Cuckoo Search Algorithms

The *cuckoo search* (CS) is a powerful metaheuristic algorithm originally proposed by Yang and Deb in 2009 [8]. Since then, it has been successfully applied to difficult optimization problems [3]–[7]. The algorithm is inspired by the obligate interspecific brood-parasitism of some cuckoo species that lay their eggs in the nests of host birds of other species to escape from the parental investment in raising their offspring. The CS algorithm is based on three idealized rules [8], [9]:

1) Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest;
2) The best nests with high quality of eggs (solutions) will be carried over to the next generations;
3) The number of available host nests is fixed, and a host can discover an alien egg with a probability $p_a \in [0, 1]$. For simplicity, this assumption can be approximated by a fraction p_a of the n nests being replaced by new nests.

The basic steps of the CS algorithm are summarized in Table I. It starts with an initial population of n host nests, randomly chosen within their upper and lower bounds, and it is performed iteratively. For each iteration t, a cuckoo egg i is selected randomly and new solutions $\mathbf{x}_i^{t+1}$ are generated by using the Lévy flight. The general equation for the Lévy flight is given by: $\mathbf{x}_i(g+1) = \mathbf{x}_i(g) + \alpha \oplus levy(\lambda)$, where g means the current generation, $\alpha > 0$ is the step size (associated with the scale of the optimization problem) and symbol $\oplus$ means the entry-wise multiplication. The second term of this expression is a transition probability modulated by the Lévy distribution as: $levy(\lambda) \sim t^{-\lambda}$, $(1 < \lambda \leqslant 3)$. The CS evaluates the fitness of the new solution and compares it with the current one. In case that the new solution brings better fitness, it replaces the current one. Also, a fraction of the worse nests, given by the probability p_a, are abandoned and replaced by new solutions to increase the exploration of the search space looking for more promising solutions. For each iteration step, all current solutions are ranked according to their fitness and the best solution reached so far is stored as the vector $\mathbf{x}_{best}$.

TABLE II: Parameters of our method and values used.

Symbol	*Meaning*	*Range*	*Used Value*
$\mathcal{N}_{gen}$	maximum number of iterations	2,000–15,000	10,000
$\mathcal{P}$	population size	20–100	50
p_a	ratio of replacement	0.001–0.9	0.01
α	step-size	0.001–0.1	0.01

III. Proposed Approach

Our method is based on the application of the CS algorithm to the optimization problem given by Eq. (3). We consider a initial population of $\mathcal{P}$ individuals or host nests (representing the candidate solutions), $\{\mathcal{N}_\kappa\}_{\kappa=1,\ldots,\mathcal{P}}$, defined as:

$$\mathcal{N}_\kappa = \left(\left\{(R_{ij}^\kappa, G_{ij}^\kappa, B_{ij}^\kappa)\right\}_{i=1,\ldots,M;j=1,\ldots,N}; \mu_\kappa\right) \tag{4}$$

where $R_{ij}^\kappa, G_{ij}^\kappa, B_{ij}^\kappa \in [0, 1]$ represent the RGB code for each pixel of the image, and $\mu^\kappa \in [0, 2]$ represents a weight factor for the contrast C of the source image, i.e., $C_\kappa = \mu_\kappa C$. This population is initialized with random values following a uniform distribution on their respective domains. This initial population is ranked according to Eq. (3). Then, the cuckoo search algorithm is applied iteratively for $\mathcal{N}_{gen}$ generations, until convergence is reached. The best solution at final generation is taken as the solution of the minimization problem.

A shortcoming of bio-inspired optimization techniques is the dependence on parameters that require proper tuning for good performance. Fortunately, the cuckoo search algorithm is particularly well suited for this problem. In contrast to most metaheuristic methods, our approach requires only a very few parameters. They are arranged in rows in Table II. For each parameter, the table shows (in columns) its symbol, meaning, range of values, and the parameter value chosen in this paper.

Finding proper values for these parameters is a difficult task because it is problem-dependent. Based on experiments, we found that $\mathcal{N}_{gen} = 10,000$ is enough to reach convergence, so this is the value used in this work. We also tested population sizes ranging from 20 to 100 individuals, and found that $\mathcal{P} = 50$ provides an adequate trade-off for this problem. We also set the value of p_a to 0.01. For α, we considered values ranging from 0.001, corresponding to a rather conservative approach based on small jumps near the current position, to 0.1, corresponding to a very aggressive approach of large jumps. After several executions, we concluded that values near 0.01 provide the best trade-off between both end options.

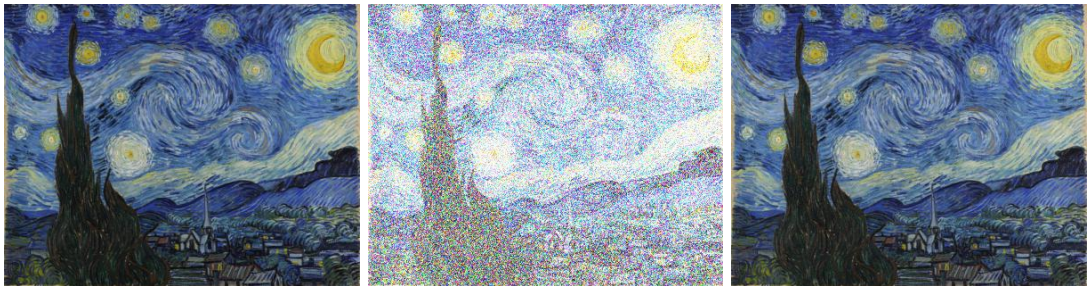

Fig. 1: *Starry Night* example: (left) Reference image; (middle) variation image; (right) reconstructed image from the variation.

Fig. 2: Ten images of the initial population of 50 images for the cuckoo search algorithm in this paper.

Fig. 3: (top-bottom, left-right) Evolution over the generations of the global best of the population from generation $g = 0$ to $g = 9,000$, with step-size $1,000$. The initial image at $g = 0$ corresponds to the variation image in Fig. 1 (middle).

IV. Experimental Results

To analyze the performance of our method, we consider the reference image shown in Figure 1(left), corresponding to a famous oil painting *Starry Night* (1889) by Dutch painter Vincent Van Gogh. It shows a rich color map, with prevalence of bluish tones: the relative percentage of the red, green, and blue channels is about 27.4%, 33.0%, and 39.6%, respectively. It is also a very representative work of a painting technique called *impasto*, which refers to the use of thickly textured, undiluted paint that looks almost three-dimensional on the canvas. When dry, impasto provides texture; the paint is seen sticking out from the canvas in globs when viewed from the side, as if it was coming out of the canvas. This three-dimensional texture makes the incident light to be reflected, and shadows are created. Fig. 1(middle) shows a variation image, where the RGB color map and other visual attributes are changed in a nonlinear and unknown way. As a result, we do not know how to transform this variation back into the reference image.

TABLE III: Similarity error value over the generations.

g	*Similarity*	*g*	*Similarity*	*g*	*Similarity*
0	0.97515	3,500	0.452605	7,000	0.205845
500	0.92959	4,000	0.404858	7,500	0.173082
1,000	0.718985	4,500	0.382751	8,000	0.14586
1,500	0.652483	5,000	0.331585	8,500	0.124525
2,000	0.556781	5,500	0.296667	9,000	0.119904
2,500	0.533526	6,000	0.265271	9,500	0.0814461
3,000	0.496339	6,500	0.227603	10,000	0.0573967

We solve this problem by applying our method in Sect. III to perform minimization of the similarity error function for the color map and contrast. To run the CS algorithm, we consider an initial population of 50 images, obtained from the variation image with RBG color map and contrast chosen randomly. Ten of such images are shown in Fig. 2 for illustration. Note the diversity of their visual attributes. Then, the CS algorithm is run for 10,000 iterations. Figure 3 shows the evolution of the global best of the population from $g = 0$ to $g = 9,000$ with step-size $1,000$. Note that these initial images are very far from the reference image. This corresponds to a highly explorative phase, where the colors and contrast of the images change drastically in order to perform extensive exploration of the search space. Over the time, the exploration through global search turns gradually into intensification through local search on the neighboring areas of the local minima, until reaching the later stages of the method, when only incremental changes are considered. As a result, the global best is slowly approaching to the reference image, as shown in last images in Fig. 3, which are all very similar to the reference image. In fact, it is even hard to distinguish them visually. Table I report the numerical values of the similarity error for the images in Fig. 3. As we can see, the initial image is very far from the reference, with a similarity error of about 97.5%. This high percentage decreases over the generations, until the method settles in the neighboring areas of the global solution and the local search phase starts. Finally, convergence to the global solution is achieved. This is clearly noticeable when the method reaches a similarity error of less than 10% for 9,500 iterations, and only small incremental improvements of the numerical value are obtained. After the method is run for 10,000 generations, the global best of the last iteration is the solution of the problem, shown in Fig. 1(right). Its similarity error is 0.0573967 (i.e., error percentage about 5.73%), which provides a very good similarity between the reference and the reconstructed images. This good matching between both images is clear by visual comparison of top and bottom pictures in Fig. 1. All computations have been performed on a 2.6 GHz Intel Core i7 processor with 32 GB of RAM, a graphical card Nvidia GeForce RTX 2060 with a 1.36 GHz processor and 6 GB GDDR6 of RAM. Regarding the CPU time, a single execution takes about 1–1.5 hours for the example in this paper. Although these times are competitive, the method is unsuitable for real-time applications.

V. Conclusions and Future Work

We introduced a new method to determine the values of the color map and contrast that, when applied to a variation of a reference image, yield a reconstructed image similar to the reference. The method is based on the cuckoo search algorithm, a popular swarm intelligence method for optimization. The method is applied to a challenging example of a Van Gogh oil painting. Our graphical and numerical results show that the method performs very well, and is able to replicate the reference image with high accuracy. Applications of this method arise in image cleaning and restoration of digital images, old or damaged pictures, and in image inpainting [1].

The proposed method has some limitations. The most critical one is the CPU time, as it requires the computation of thousands of images. The use of graphical cards and parallel programming can alleviate this problem, but not completely. A second problem is the fact that only a few visual attributes (the color and the contrast) are considered in this method, so its applicability is limited to only these two features. Solving these problems are our plans for future work in the field.

Acknowledgment

A. Gálvez and A. Iglesias thank the projects PDE-GIR of European Union's Horizon 2020 research and innovation program, Marie Sklodowska-Curie grant agreement, reference number 778035, and #TIN2017-89275-R from the Agencia Estatal de Investigación (AEI) of the Spanish Ministry of Science, Innovation and Universities (Computer Science National Program) and European Funds EFRD (AEI/FEDER, UE). E. Osaba and J. Del Ser would like to thank the Basque Government for its funding support through the EMAITEK and ELKARTEK programs. J. Del Ser also thanks the Consolidated Research Group MATH-MODE (IT1294-19) by the Department of Education of the Basque Government.

References

[1] Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image Inpainting. *Proc. ACM SIGGRAPH 2000*, New Orleans, ACM Press, 417–424 (2000).

[2] Gonzalez, R.C., Woods, R.E.: *Digital Image Processing. 4th Edition*. Pearson Prentice Hall, N.J. (2017).

[3] Gálvez, A., Iglesias, A., Cabellos, L.: Cuckoo search with Lvy flights for weighted Bayesian energy functional optimization in global-support curve data fitting. *The Scientific World Journal*, **2014**, Article ID 138760, 11 pages (2014).

[4] Gálvez, A., Fister, I., Fister Jr., I., Osaba, E., Del Ser, J., Iglesias, A.: Cuckoo search algorithm for border reconstruction of medical images with rational curves. *Lecture Notes Comput. Sci.*, **11655**, 320–330 (2019).

[5] Gálvez, A., Iglesias, A.: Memetic improved cuckoo search algorithm for automatic B-spline border approximation of cutaneous melanoma from macroscopic medical images. *Advanced Engineering Informatics*, **43**, 101005 (2020).

[6] Iglesias, A., Gálvez, A.: Cuckoo search with Lévy flights for reconstruction of outline curves of computer fonts with rational Bézier curves. Proc. of IEEE CEC, IEEE CS Press, Los Alamitos, CA, 2247–2254 (2016).

[7] Iglesias, A., Gálvez, A., Suárez, P., Shinya, M., Yoshida, N., Otero, C., Manchado, C., Gómez-Jauregui, V.: Cuckoo search algorithm with Lvy flights for global-support parametric surface approximation in reverse engineering. *Symmetry* **10**(3), Article 58 (2018).

[8] Yang, X.S., Deb, S.: Cuckoo search via Lévy flights. In: *Proc. World Congress on Nature & Biologically Inspired Computing (NaBIC)*. IEEE (2009) 210-214.

[9] Yang, X.S., Deb, S.: Engineering optimization by cuckoo search. *Int. J. Math. Modelling and Numerical Optimization*, **1**(4) (2010) 330-343.