Novel Nature-Inspired Selection Strategies for Digital Image Evolution of Artwork

Novel Nature-Inspired Selection Strategies for Digital Image Evolution of Artwork

Doina Logofătu[1], Gia Thuan Lam,
Kristiyan Balabanov, Costin Badica

Working Group Soft Computing, Programming and Algorithms[1]
Faculty of Computer Science and Engineering
Frankfurt University of Applied Sciences
Frankfurt a.M., Germany

Outline

Outline

Main Idea

- Computer science has enhanced and/or replaced human labor in various fields
- There is an ongoing debate whether machines would ever possess traits unique to humans
- One example is creativity and forms to express it, such as art
- Art is often inspired by nature

Main Idea

- Computer science has enhanced and/or replaced human labor in various fields
- There is an ongoing debate whether machines would ever possess traits unique to humans
- One example is creativity and forms to express it, such as art
- Art is often inspired by nature

“Is possible to apply nature-inspired computing techniques, such as evolutionary algorithms, in the creation of art?”

Main Idea

- Computer science has enhanced and/or replaced human labor in various fields
- There is an ongoing debate whether machines would ever possess traits unique to humans
- One example is creativity and forms to express it, such as art
- Art is often inspired by nature

“Is possible to apply nature-inspired computing techniques, such as evolutionary algorithms, in the creation of art?”
“In a way, yes!”

Evolutionary Art

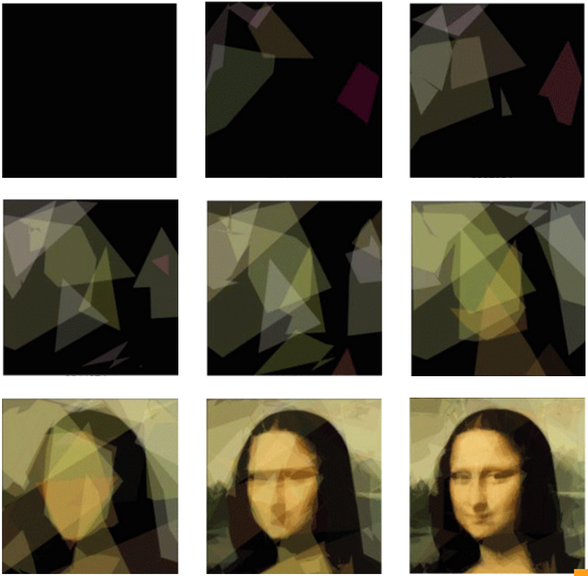
Example:

- Approximate paintings with decent quality using evolutionary strategies [6, 7, 8, 9, 10]
- Johansson's "Evolution of Mona Lisa" [4]

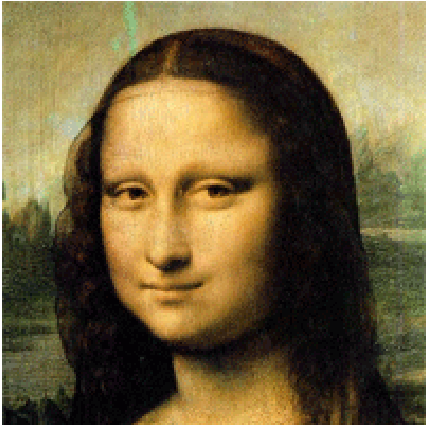
Approach:

- Use a finite list of translucent polygons
- A polygon consists of a finite number of points
- A polygon is initialized with random color and coordinates
- The polygons evolve by mutating their color and coordinates
- The procedure runs in discrete steps until a selection criteria is met

“Evolution of Mona Lisa”



Comparison with the Original



Original painting

Approximation after 1,000,000 iterations

Motivation

- Just a few implementations exist
- Most of them exhibit a slow convergence speed, i. e. the effort to result quality ratio decreases exponentially with the number of iterations
- Existing implementations emphasize on *mutation* rather than *selection*
- We propose a novel natural selection strategy based on *Simulated Annealing*, that offers better efficiency

Outline

Project Scope

Implementation Details

Experimental Results

Conclusion

Expected Input and output

- The input is a digital colored image of size $W \times H$ pixels
- The output is an approximation of the input created using a set of evolved polygons

Puppy Submarine Parrot

Original

Approximation

Solution Encoding Scheme

- A candidate solution S is represented as a finite set of n polygons
- A polygon has a fixed number of vertexes m
- Each polygon is colored using the *ARGB* (alpha red green blue) color space
- A vertex is identified by its x, y-coordinates
- A color is identified by the four values a (alpha), r (red), g (green), b (blue)
- All values are expressed as integers
- A solution is encoded as a sequence of integers representing the aforementioned values, where

$$0 \leq a, r, g, b \leq 255, \quad 0 \leq x < W \leq 200, \quad 0 \leq y < H \leq 200$$

Fitness Function

- All values in a solution's encoding are within the range $[0, 255]$
- In other words all values fit in a single byte
- The fitness function used performs a *byte-to-byte mapping* between a candidate solution and the desired outcome
- The input and output are converted into byte arrays B_1 and B_2 of size $4 \times W \times H$
- The output of the fitness function is the averaged byte difference computed as

$$f = \frac{1}{4 \times W \times H} \sum_{i=1}^{4 \times W \times H} |B_1[i] - B_2[i]|$$

Mutation Algorithms

Single Attribute Mutation:

- Change the value at a random position in a candidate solution's sequence to a random number
- Beyond some threshold add a random offset to the new value

Hill Climbing Mutation:

- Each iteration either a color attribute or a vertex attribute is randomly changed

Genetic Mutation:

- Change a random number of attributes (between none and all)

Individual Selection Algorithms

Best Selection:

- Mutate a solution while keeping the original
- Compare the fitness of the original and the mutant, and select the better one

Simulated Annealing Selection [13]:

- Mutate a solution while keeping the original
- Compare the fitness of the original and the mutant, and select the better one
- Dynamically change the acceptance criteria T for mutants over the course of the simulation
- The acceptance criteria in the i-th iteration is a product of the initial acceptance threshold T_0 and a decreasing factor

$$T(i) = T_0 \times c^i, \quad 0 < c < 1$$

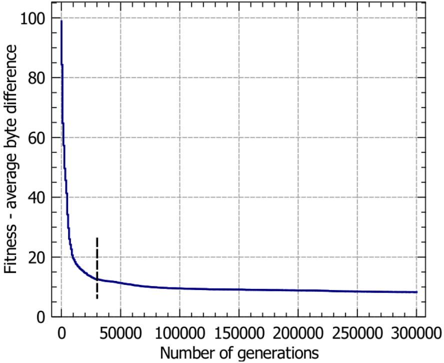
Outline

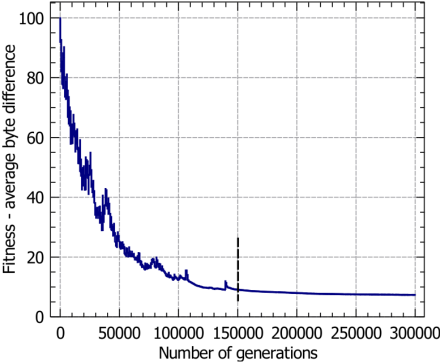
Conducted Experiments

- Tests included 20 random images of various sizes
  - 7 images of size 100 $\times$ 100 pixels
  - 7 images of size 150 $\times$ 150 pixels
  - 6 images of size 200 $\times$ 200 pixels
- An image consists of 50 polygons with 6 vertexes
- Each image was tested over a duration of 300,000 iterations
- 4 different combinations of mutation and selection algorithms were used
  - (A) Hill Climbing Mutation + Best Selection
  - (B) Hill Climbing Mutation + Simulated Annealing Selection
  - (C) Genetic Mutation + Best Selection
  - (D) Genetic Mutation + Simulated Annealing Selection

Selection Algorithms Comparison

- Basic Selection quickly gets stuck at a *local optimum*
- Simulated Annealing Selection has a better exploration [18] and obtains slightly better results





Gathered Data

No.	A	B	$\frac{A-B}{A} \times 100(\%)$	C	D	$\frac{C-D}{C} \times 100(\%)$
1	7.527450	6.487500	13.815435510	7.250175	6.358600	12.297289380
2	13.156640	11.747910	10.707397830	10.876520	10.624350	2.318462581
3	7.369000	6.373550	13.508617180	8.613675	6.247450	27.470562800
4	5.736825	4.642700	19.071960540	5.065475	4.343025	14.262236020
5	8.627625	7.431550	13.863316960	8.490500	6.626675	21.951887400
6	12.602330	12.098730	3.996088023	12.984300	10.185750	21.553337490
7	1.885150	0.591225	68.637774180	2.416125	0.672800	72.153758600
8	7.865344	6.700000	14.816186040	7.536533	6.950556	7.775153376
9	13.654380	11.987840	12.205125710	13.748460	12.695940	7.655492369
10	8.033178	6.307233	21.485207970	6.994689	6.838200	2.237254580
11	17.575620	16.759860	4.641463045	17.243740	16.979870	1.530276719
12	5.591567	4.717011	15.640624530	5.237422	5.066933	3.255208383
13	2.919456	2.035122	30.291054220	3.585878	2.488122	30.613311440
14	10.085630	9.006722	10.697504060	9.920333	8.065389	18.698404580
15	12.939250	11.703430	9.550932241	12.660190	11.810990	6.707593916
16	8.140325	6.146339	24.495164510	8.242873	6.639227	19.454940040
17	14.181910	13.340490	5.932996594	14.710340	14.385130	2.210818455
18	8.408981	8.353006	0.665657349	9.050606	8.957644	1.027135642
19	7.163481	7.036281	1.775673028	7.294994	7.091244	2.793011207
20	10.838990	10.428110	3.790713672	11.106470	10.759180	3.126952409

Results Summary

Simulated Annealing Selection offers and improvement of:

- 0.67% to 68.64% for (A) and (B) with 14.98% on average
- 1.03% to 72.15% for (C) and (D) with 13.95% on average

Hill Climbing Mutation offers and improvement of:

- 20.51% on average for images of size 100×100
- 15.68% on average for images of size 150×150
- 7.70% on average for images of size 200×200

Genetic Mutation offers and improvement of:

- 24.57% on average for images of size 100×100
- 10.25% on average for images of size 150×150
- 5.89% on average for images of size 200×200

Outline

Conclusion

- The size of the image is relevant regarding the improvement offered by our method
- For larger images more polygons with more vertexes are needed
- An improvement as small as 0.67% using Simulated Annealing Selection is equivalent to thousands of additional iterations needed by Best Selection to obtain the same result
- Simulated Annealing Selection outperforms Best Selection

Thank You for Your Attention!

¿Questions?

- Fujiwara, Y., Sawai, H.: Evolutionary computation applied to 3D image reconstruction. Evolutionary Computation, USA (1997).
- Ho, S. Y., Chen, Y. C.: An efficient evolutionary algorithm for accurate polygonal approximation. Pattern Recognition, vol. 34, pp. 2305–2317, Pergamon (2001).
- Santamaría, J. et. al.: Performance evaluation of memetic approaches in 3D reconstruction of forensic objects. Soft Computing, pp. 883, Springer-Verlag (2009).
- Genetic Programming: Evolution of Mona Lisa, `https://rogerjohansson.blog/2008/12/07/genetic-programming-evolution-of-mona-lisa`,last accessed 2018/01/31.
- Dianne, H.: Mona Lisa - A Life Discovered. Simon & Schuster (2015).
- Mona Lisa Thing, `https://github.com/mackstann/mona/tree/master`, last accessed 2018/01/31.

- Evolve: Image Evolution, http://alteredqualia.com/visualization/evolve, last accessed 2018/01/31.
- Evolisa, https://github.com/miezekatze/evolisa, last accessed 2018/01/31.
- Genetic Lisa, https://github.com/peterbraden/genetic-lisa, last accessed 2018/01/31.
- Genetic Programming: Mona Lisa FAQ, https://rogerjohansson.blog/2008/12/09/genetic-programming-mona-lisa-faq, last accessed 2018/01/31.
- Luke, S.: Essentials of Metaheuristics 2nd edition. Lulu (2013).
- Evolution Strategies, http://www.bionik.tu-berlin.de/institut/xs2evost.html, last accessed 2018/01/31.

- Kirkpatrick, S., Gelatt, C.D.J., Vecchi, M.P.: Optimization by Simulated Annealing. Science, vol. 220, issue 4598, pp. 671–680 (1983).
- Russell, S., Norvig, P.: Artificial Intelligence - A Modern Approach. 2nd edition. Upper Saddle River, New Jersey: Prentice Hall, pp. 111–114 (2003).
- Azizi, N., Zolfaghari, S.: Adaptive temperature control for simulated annealing: A comparative study. Computers & Operations Research, pp. 2439–2451 (2004).
- Schneider, J., Puchta, M.: Investigation of acceptance simulated annealing - A simplified approach to adaptive cooling schedules. Statistical Mechanics and its Applications, vol. 389, pp. 5822-5831 (2010).
- Xinchao, Z.: Simulated annealing algorithm with adaptive neighborhood. Applied Soft Computing, vol. 11, issue 2, pp. 1827–1836 (2011).

Li, K., Li, M., Chen, H.: The Empirical Analysis of Exploration-Exploitation Tradeoff for Uncertain Environment. 2014 Seventh International Symposium on Computational Intelligence and Design, pp. 274–279, Hangzhou (2014)