Evolutionary Computation for Digital Art

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07.
https://doi.org/10.1145/3319619.3323375

Aneta Neumann, Frank Neumann
The University of Adelaide
Australia

Link to the Current Version

The current version is available at:

https://researchers.adelaide.edu.au/profile/aneta.neumann

Introduction and Motivation

Motivation

- Evolutionary Computation (EC) techniques have been frequently used in the context of computational creativity.
- Various techniques have been used in the context of music and art (see EvoMusArt conference and DETA track at GECCO).

Motivation

- Evolutionary algorithms have been frequently used to optimize complex objective functions.
- This makes them well suitable for generative art where fitness functions are often hard to optimize.
- Furthermore, objective functions are often subjective to the user.

Motivation

- In terms of novel design, evolutionary computation techniques can be used to explore new solutions in terms of different characteristics.
- Evolutionary algorithms are able to adapt to changing environments.
- This makes them well suited to be used in the context of artistic work where the desired characteristics may change over time.

This Tutorial

- Summary of results in the areas of
  - 2D and 3D artifacts
  - Animations
- Overview on our recent work to create unique generative art using evolutionary computation to carry out
  - Image transition and animation
  - Image composition
  - Diversity optimization for images

Outline

- Introduction and Motivation
- Evolving 2D and 3D Artifacts
- Aesthetic Features
- Evolutionary Image Transition
- Quasi-random Image Animation
- Evolutionary Image Composition
- Evolutionary Image Diversity Optimization
- Discrepancy-Based Evolutionary Diversity Optimization for Images
- Conclusions

Evolving 2D and 3D Artifacts

Evolving 2D and 3D Artifacts

- *Blind Watchmaker* (Dawkins, 1986) evolved 2D biomorph graphical objects from sets of genetic parameters (combined with Darwinism theory).
- Latham (1985) created *Black Form Synth*. These are hand-drawn "evolutionary trees of complex *forms*" using a set of transformation rules.

Evolving 2D and 3D Artifacts

- In 1991, Sims published his seminal SIGGRAPH paper.
- He introduced the expression-based approach of evolving images.
- He created images, solid textures, and animations using mutations of symbolic lisp expressions.

Evolving 2D and 3D Artifacts

- The mathematical expression is represented as a tree graph structure and used as the genotype.
- The tree graph consists of mathematical functions and operators at the nodes, and constants/variables at the leaves (similar to genetic programming).
- The resulting image is the phenotype.
- To evolve sets of images, it uses crossover and mutation.

Evolving 2D and 3D Artifacts (Sims, 1997)

- *In Galápagos* (Sims, 1997) created an interactive evolution of virtual "organisms" based on Darwinian theory.
- Several computers simulate the growth and characteristic behaviours of a population of abstract organisms.
- The results are displayed on computer screens.

Evolutionary Process (Sims, 1997)

- The offspring are copies and combinations of their parents.
- In addition, their genes are altered by random mutations.
- During evolutionary cycle of reproduction and selection, new organisms are created.

EC System (Sims, 1997)

- The EC system allows users to express their preferences by selecting their preferred display by standing on step sensors in front of those displays.
- The selected display is used for reproduction using mutation/crossover. The other solutions are removed when the new offspring is created.

Evolving 2D and 3D Artifacts (Latham, Todd, 1992)

- Latham, Todd (1992) introduced *Mutator* to generate art and evolve new biomorphic forms.
- The Mutator creates complex branching organic forms through the process of "surreal" evolution.
- At each iteration the artist selects phenotypes that are "breed and growth", and the solutions co-interact.

Other Selected Contributions

- Unemi (1999) developed *SBART*. This is a design support tool to create 2-D images based on user selection.
- Takagi (2001) describes in the survey research on interactive evolutionary computation (IEC) which categorises different application areas.
- Machado and Cardoso (2002) introduced *NEvAr*. *This* is an evolutionary art tool, using genetic programming and automatic fitness assignment.

Other Selective Contributions

- Draves (2005) introduced *Electric Sheep*. *The* system allows a user to approve or disapprove phenotypes.
- Hart (2009) evolved different expression-based images with a focus on colours and forms.
- Kowaliw, Dorin, McCormack (2012) explore a definition of creative novelty for generative art.

Image Morphing (Banzhaf, Graf 1995)

- Banzhaf and Graf (1995) used interactive evolution to help determine parameters for image morphing.
- They combine IEC with the concepts of warping and morphing from computer graphics to evolve images.
- They used recombination of two bitmap images through image interpolation.

Aesthetic Measures

Aesthetic Measures

- Computational aesthetic is a subfield of artificial intelligence dealing with the computational assessment of aesthetic forms of visual art.
- Some general image features that have been used are:
  - Hue
  - Saturation
  - Symmetry
  - Smoothness

Aesthetic Measures

- Examples of aesthetic measurements:
  - Benford's Law
  - Global Contrast Factor
  - Information Theory
  - Reflectional Symmetry
  - Colorfulness

Aesthetic Measures (den Heijer, Eiben 2014)

- den Heijer and Eiben (2014) investigated aesthetic measures for unsupervised evolutionary art.
- Their Art Habitat System uses genetic programming and evolutionary multi-objective optimization.
- They compared aesthetic measurements and gave insights into the correlation of aesthetic scores.

Evolutionary Image Transition

Evolutionary Image Transition

- The main idea compromises of using well-known evolutionary processes and adapting these in an artistic way to create an innovative sequence of images (video).
- The evolutionary image transition starts from given image **S** and evolves it towards a target image **T**
- Our goal is to maximise the fitness function where we count the number of the pixels matching those of the target image.

Evolutionary Image Transition

Algorithm 1 Evolutionary algorithm for image transition

- Let S be the starting image and T be the target image.
- Set X:=S.
- Evaluate $f(X,T)$.
- while (not termination condition)
  - Obtain image Y from X by mutation.
  - Evaluate $f(Y,T)$
  - If $f(Y,T) \geq f(X,T)$, set $X := Y$.

Fitness function: $f(X,T) = |\{X_{ij} \in X \mid X_{ij} = T_{ij}\}|.$

Asymmetric Mutation

- We consider a simple evolutionary algorithm that has been well studied in the area of runtime analysis, namely variants of (1+1) EA.
- We adapt an asymmetric mutation operator used in Neumann, Wegener (2007) and Jansen, Sudholt (2010).

Asymmetric Mutation

Algorithm 2 Asymmetric mutation

- Obtain Y from X by flipping each pixel X_{ij} of X independently of the others with probability $c_s/(2|X|_S)$ if $X_{ij} = S_{ij}$, and flip X_{ij} with probability $c_t/(2|X|_T)$ if $X_{ij} = T_{ij}$, where $c_s \geq 1$ and $c_t \geq 1$ are constants, we consider $m = n$.

- for our experiments we set c_s =100 and c_t=50.

Example Images

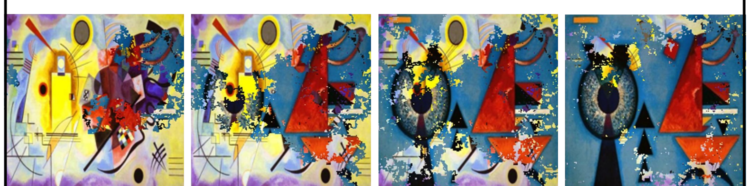
Starting image S (Yellow-Red-Blue, 1925 by Wassily Kandinsky) and target image T (Soft Hard, 1027 by Wassily Kandinsky)

Video: Asymmetric Mutation

Uniform Random Walk

- A *Uniform Random Walk* - the classical random walk chooses an element $X_{kl} \in N(X_{ij})$ uniformly at random.
- We define the neighbourhood $N(X_{ij})$ of X_{ij} as

$$N(X_{ij}) = \{X_{(i-1)j}, X_{(i+1)j}, X_{i(j-1)} X_{i(j+1)}\}$$



Uniform Random Walk

Algorithm 3 Uniform Random Walk

- Choose the starting pixel $X_{ij} \in X$ uniformly at random.
- Set $X_{ij} := T_{ij}$.
- while (not termination condition)
  - Choose $X_{kl} \in N(X_{ij})$ uniformly at random.
  - Set $i := k$, $j := l$ and $X_{ij} := T_{ij}$.
- Return X.

Video – Uniform Random Walk

Biased Random Walk

- A *Biased Random Walk* - the probability of choosing the element X_{kl} is dependent on the difference in RGB-values for T_{ij} and T_{kl}.

Biased Random Walk

Algorithm 4 Biased Random Walk

- Choose the starting pixel $X_{ij} \in X$ uniformly at random.
- Set $X_{ij} := T_{ij}$.
- while (not termination condition)
    - Choose $X_{kl} \in N(X_{ij})$ according to probabilities $p(X_{kl})$.
    - Set $i := k$, $j := l$ and $X_{ij} := T_{ij}$.
- Return X.

Biased Random Walk

We denote by T^r_{ij}, $1 \le r \le 3$, the rth RGB value of T_{ij} and define

$$\gamma(X_{kl}) = \max\left\{\sum_{r=1}^{3} |T^r_{kl} - T^r_{ij}|, 1\right\}$$

$$p(X_{kl}) = \frac{(1/\gamma(X_{kl}))}{\sum_{X_{st}\in N(X_{ij})}(1/\gamma(X_{st}))}.$$

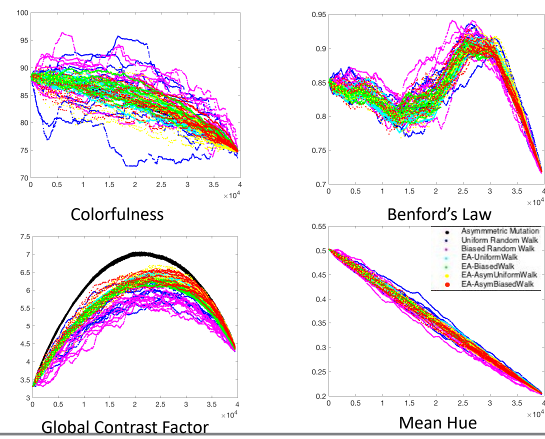
Mutation Based on Random Walks

- We use the random walk algorithms as part of our mutation operators.
- Each mutation picks a random pixel and runs the (biased) random walk for t_{max} steps.
- For our experiments we use 200x200 images and set t_{max}=100.

Random Walk Mutation and Biased Random Walk Mutation

Videos - Biased Random Walk Evolutionary Algorithm

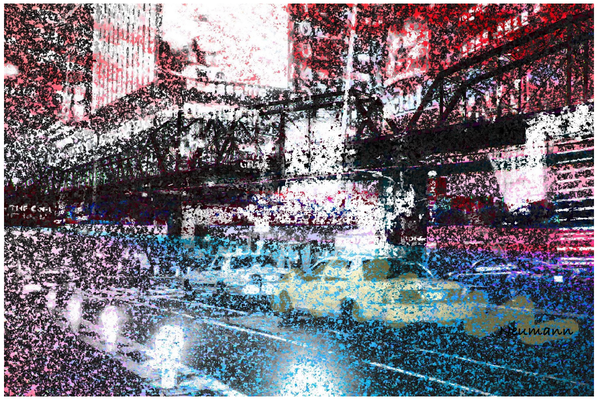
Feature Values During Transition:



SALA 2016 – Art Exhibition, Australia

© Aneta Neumann

SALA 2016 – Adelaide, Australia



© Aneta Neumann

Quasi-random Transition and Animation

Quasi-random Walks

- So far: Random walks as main operators for mutation and transition process
- Quasi-random walks give a (deterministic) alternative which is easy to control by a user.

Quasi-random Transition and Animation

General setting:

- There are k agents each of them painting their own image I^k through a quasi random walk. Quasi-random walk is determined by the sequence that the agent uses.
- Process starts with a common image X.
- All agents paint on this image overriding X and previous painting of other agents.
- This leads to complex animation processes.
- Image transition is a special case where all agents paint the same image I.

Agent Moves

Move of an agent:

- Each pixel has a sequence of directions out of from
  {left, right, up, down}.
- At each iteration, the agent moves from its current pixel p to the neighbor pixel p' determined by the current position in the sequence at p.
- It paints pixel p' with the current pixel in its sequence and increases the position counter at p by 1 (modulo sequence length).

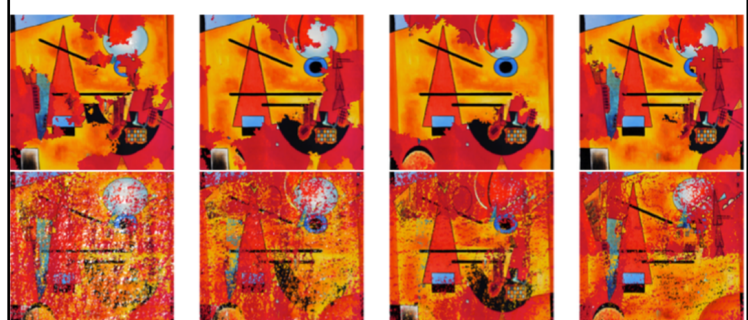
Algorithm

Algorithm 1 QUASI-RANDOM ANIMATION

Require: Start image Y of size $m \times n$. For each agent k, $1 \leq k \leq r$, an image I^k of size $m \times n$, sequence S^k and position counters $c^k(i,j) \in \{0, \ldots, |S^k|\}$, $1 \leq i \leq m$, $1 \leq j \leq n$.

$X \leftarrow Y$
for each agent k, $1 \leq k \leq r$ **do**
  choose $P^k \in m \times n$ and set $X(P^k) := I^k(P^k)$.
end for
$t \leftarrow 1$
while $(t \leq t_{\max})$ **do**
  for each agent k, $1 \leq k \leq r$ **do**
    Choose $\hat{P}^k \in N(P^k)$ according to $S_k(c(P^k))$.
    $X(\hat{P}^k) \leftarrow I^k(\hat{P}^k)$
    $c^k(P^k) \leftarrow (c^k(P^k) + 1) \mod |S^k|$.
    $P^k \leftarrow \hat{P}^k$.
  end for
  $t \leftarrow t + 1$
end while

2 Agents Symmetric and Asymmetric Sequences



Example Video: 4 Agents Symmetric Sequences

Example Video: 4 Agents Asymmetric Sequences

Evolutionary Image Composition

Key Idea

- Create a composition of two images using a region covariance descriptor.
- Incorporate region covariance descriptors into fitness function.
- Use Evolutionary algorithms to optimize the fitness such that a composition of the given two images based on the considered features is obtained.

Image Composition

Evolutionary Image Composition Using Feature Covariance Matrices

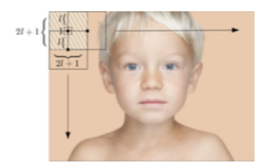
- Evolutionary algorithms that create new images based on a fitness function that incorporates feature covariance matrices associated with different parts of the images.
- Population-based evolutionary algorithm with mutation and crossover operators based on random walks.

Algorithm 1 $(\mu + 1)$ GA for evolutionary image composition

Require: S and T are images

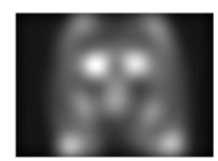
Initialise population $\mathcal{P} = \{P_1, \ldots, P_\mu\}$
while not termination condition **do**
  Select an individual $P_i \in \mathcal{P}$ uniformly at random
  if $rand() < p_c$ **then** ▷ Crossover
    Select $P_j \in \mathcal{P} \setminus P_i$ uniformly at random
    if $rand() < 0.5$ **then** ▷ See Section 4.2 for t_{cr}
      $Y \leftarrow \textsc{RandomWalkMutation}(X, Z, t_{cr})$
    else
      $Y \leftarrow \textsc{RectangularCrossover}(P_i, P_j)$
    $P_i \leftarrow \textsc{Selection}(P_i, Y)$
  else ▷ Mutation
    if $rand() < 0.5$ **then**
      $Y \leftarrow \textsc{RandomWalkMutation}(P_i, S, t_{max})$
    else
      $Y \leftarrow \textsc{RandomWalkMutation}(P_i, T, t_{max})$
    $P_i \leftarrow \textsc{Selection}(P_i, Y)$
    Adapt t_{max} ▷ See Section 4.1.
return $\mathcal{P}$ ▷ Result is a population of evolved images.

#3
square region of interest



$$\mathcal{G} = \left\{ (c,d) \;\middle|\; \begin{array}{l} c = (l+1) + pl,\ p = 0, 1, \ldots, \left\lfloor \frac{m-l}{l} \right\rfloor - 1 \\ d = (l+1) + ql,\ q = 0, 1, \ldots, \left\lfloor \frac{n-l}{l} \right\rfloor - 1 \end{array} \right\}$$

#4
saliency mask



#5
set of features

Set 1: $\left[i, j, r, g, b, \sqrt{\left(\frac{\partial I}{\partial i}\right)^2 + \left(\frac{\partial I}{\partial j}\right)^2}, \tan^{-1}\left(\left|\frac{\partial I}{\partial i}\right| / \left|\frac{\partial I}{\partial j}\right|\right)\right]^{\mathrm{T}}$;

Set 2: $[i, j, h, s, v]^{\mathrm{T}}$;

Set 3: $\left[h, s, v, \sqrt{\left(\frac{\partial I}{\partial i}\right)^2 + \left(\frac{\partial I}{\partial j}\right)^2}, \tan^{-1}\left(\left|\frac{\partial I}{\partial i}\right| / \left|\frac{\partial I}{\partial j}\right|\right)\right]^{\mathrm{T}}$.

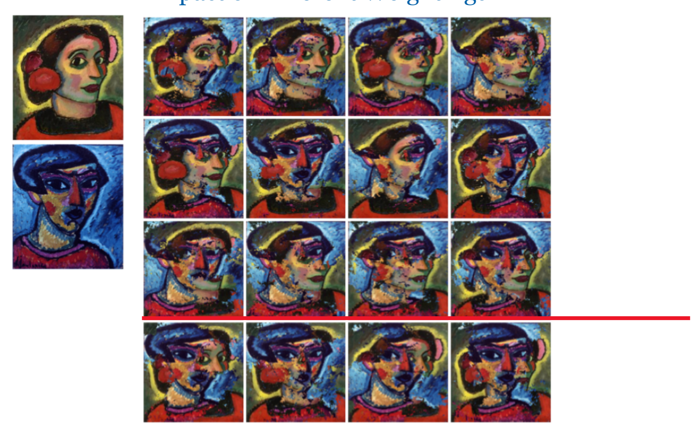
#6
covariance-based fitness function

$$f(X,S,T) = \sum_{(c,d)\in\mathcal{G}} \Big(w^{S}_{(c,d)} \operatorname{dist}\left(\Lambda^{X}_{\mathcal{R}_{(c,d)}}, \Lambda^{S}_{\mathcal{R}_{(c,d)}}\right) + w^{T}_{(c,d)} \operatorname{dist}\left(\Lambda^{X}_{\mathcal{R}_{(c,d)}}, \Lambda^{T}_{\mathcal{R}_{(c,d)}}\right)\Big),$$

Impact of Different Features

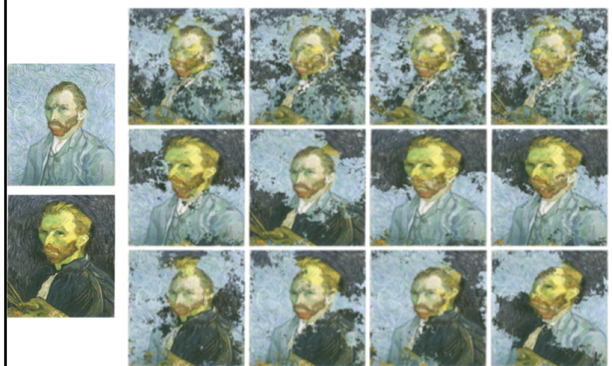
Image composition with different features. Rows 1, 2 and 3 correspond to Feature Sets 1, 2 and 3, respectively.

Impact of Different Weightings



Rows 1, 2 and 3 correspond to $w^{S}_{(c,d)}$ set to 0.25, 0.5 and 0.75 and $w^{T}_{(c,d)}$ set to 0.75, 0.5 and 0.25, respectively. In the last row the weights were set using an image saliency algorithm. The saliency algorithm strikes a consistent balance between notable regions in both images.

Impact of Distance Metrics



Rows 1, 2 and 3 correspond to distance metrics $dist_E$, $dist_A$ and $dist_L$, respectively.

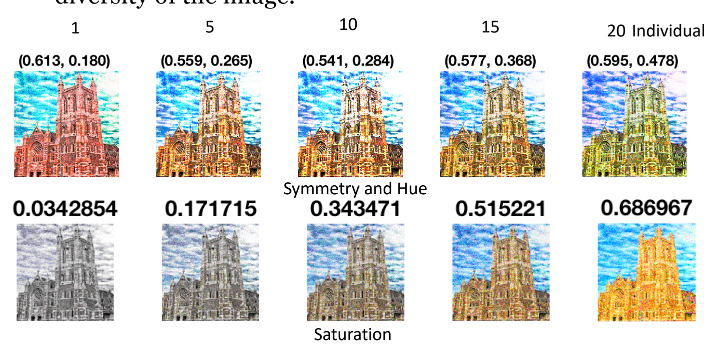
Variants of Image Composition

Image composition with Feature Set 1, saliency-based weighting and a Log-Euclidean distance measure.

Evolutionary Diversity Optimisation
for Images

Key Idea

- Produce diverse image sets using evolutionary computation methods.
- Use the $(\mu + \lambda)$–EA_D for evolving image instances
- Select the individuals based on their contribution to diversity of the image.



Evolution of Artistic Image Variants Through Feature Based Diversity Optimisation

- We use $(\mu + \lambda) - EA_D$ to evolve diverse image instances.
- Knowledge on how we can combine different image features to produce interesting image effects.
- Study how different combinations of image features correlate when images are evolved to maximise diversity.

Algorithm 1 The $(\mu + \lambda) - EA_D$ algorithm μ = 20 and λ = 10

```
input: an image S.
output: a population P = {I_1, ..., I_μ} of image variants.
{Initialise with μ mutated copies of source image}
P = {mutate(S), ..., mutate(S)}
repeat
  randomly select C ⊆ P where |C| = λ
  for I ∈ C do
    produce I' = mutate(I)
    if valid(I') then
      add I' to P
    end if
  end for
  while |P| > μ do
    remove an individual I = arg min_{J∈P} d(J, P)
  end while
until Termination condition reached
```

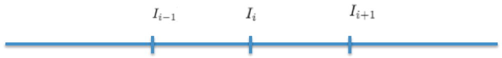
#1 starting image

#2 pixel-based mutation

#3 image validity check

Image has mean squared error to starting image less than 10

#4 feature diversity measure



$$f(I_1) \leq f(I_2) \leq \ldots \leq f(I_k),\ f(I_i) \neq f(I_1) \neq f(I_k)$$

$$d_{f_i}(I_i, P) = (f(I_i) - f(I_{i-1})) \times (f(I_{i+1}) - f(I_i))$$

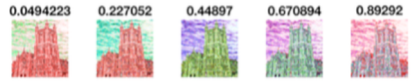
$$d'(I, P) = \sum_{i=1}^{k} (w_i \times d_{f_i}(I, P))$$

[Gao, Nallaperuma, F. Neumann, PPSN 2016,arxiv2016]

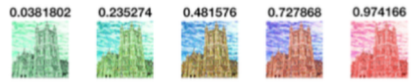
Single Dimensional Feature Results

#5 features

1 5 10 15 20 Individuals

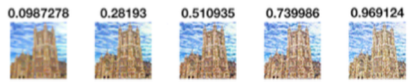


(a) Hue



(b) SD Hue

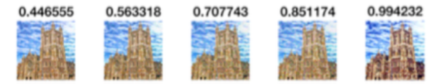
(c) Saturation



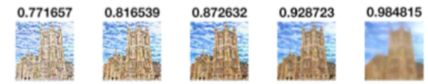
(d) Global Contrast Factor

Single Dimensional Feature Results

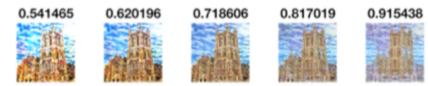
1 5 10 15 20 Individuals



(e) Information Theory

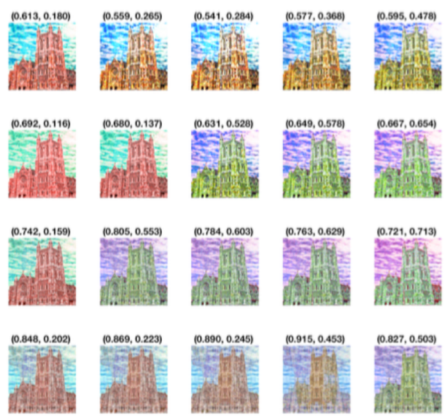


(f) Smoothness



(g) Symmetry

Two-Dimensional Feature Experiments



a) Symmetry and Hue 20 Individuals

Discrepancy-Based Evolutionary Diversity Optimization for Images

Discrepancy-Based Evolutionary Diversity Optimization

- New approach for discrepancy-based evolutionary diversity optimization
- Investigate the use of the star discrepancy measure for diversity optimization for images and TSP instances
- Introduce an adaptive random walk mutation operator based on random walks
- Compared the previously approach for images and TSP instances [Alexander, Kortman, A. Neumann, GECCO 2017]

Discrepancy-Based Evolutionary Diversity Optimization for Images

#1

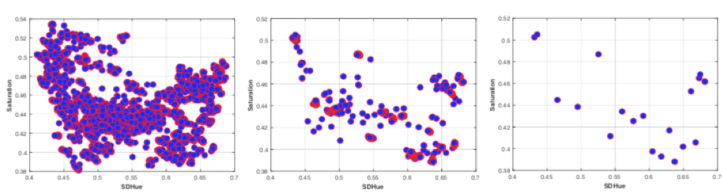
Start Image

#2

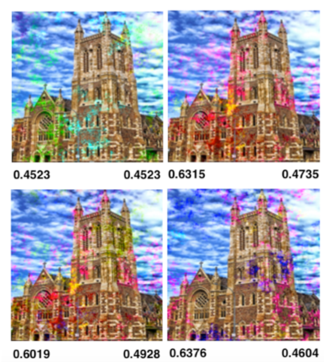
Features

#2

Features



Discrepancy-Based Evolutionary Diversity Optimization for Images



#4

Experimental settings

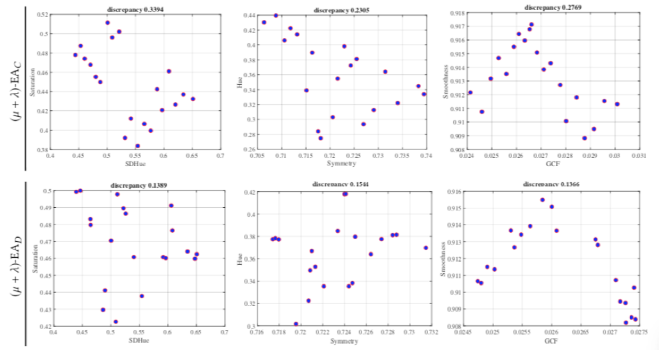
#5

Results

Discrepancy-Based Evolutionary Diversity Optimization for Images

#5

Results



Discrepancy-Based Evolutionary Diversity Optimization for Images

#5

Results

	$(\mu+\lambda)$-EA_C(1)				$(\mu+\lambda)$-EA_D (2)				$(\mu+\lambda)$-EA_T(3)			
	min	mean	std	stat	min	mean	std	stat	min	mean	std	stat
(f1, f2)	0.2014	0.3234	0.0595	$2^{(-)},3^{(-)}$	0.1272	0.2038	0.1157	$1^{(+)}$	0.1119	0.1530	0.0269	$1^{(+)}$
(f3, f4)	0.1964	0.2945	0.0497	$2^{(-)},3^{(-)}$	0.1574	0.2280	0.0592	$1^{(+)},3^{(-)}$	0.1051	0.1417	0.0179	$1^{(+)},2^{(+)}$
(f5, f6)	0.1997	0.2769	0.0344	$2^{(-)},3^{(-)}$	0.1363	0.2025	0.0538	$1^{(+)}$	0.1457	0.1800	0.0234	$1^{(+)}$
(f1, f2, f3)	0.3389	0.4327	0.0613	$2^{(-)},3^{(-)}$	0.1513	0.3335	0.1062	$1^{(+)}$	0.2253	0.2814	0.0422	$1^{(+)}$
(f1, f4, f3)	0.2754	0.3395	0.0483	$2^{(-)},3^{(-)}$	0.2100	0.3118	0.1309	$1^{(+)}$	0.2224	0.2600	0.0123	$1^{(+)}$
(f5, f4, f2)	0.4775	0.6488	0.0841	$2^{(-)},3^{(-)}$	0.2021	0.3007	0.1467	$1^{(+)}$	0.1983	0.2229	0.0125	$1^{(+)}$

Literature

- R. Dawkins (1986): The Blind Watchmaker - Why the Evidence of Evolution Reveals a Universe without Design, W. W. Norton & Company.
- W. Latham (1985): *Black Form Synth*. Offset lithograph, E.293-2014, Victoria and Albert Museum, London, UK. http://collections.vam.ac.uk/item/O1276894/black-form-synth-offset-lithograph-latham-william/
- K. Sims (1991): Artificial evolution for computer graphics. In Proc. Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '91). ACM Computer Graphics, 25(4): pp. 319–328. https://dl.acm.org/citation.cfm?doid=122718.122752
- K. Sims (1997): *Galápagos*. Installation at the NTT InterCommunication Center in Tokyo, Japan. http://www.ntticc.or.jp/en/archive/works/galapagos/
- S. Todd and W. Latham (1992): Evolutionary Art and Computers. Academic Press, London.
- T. Unemi (1999): SBART2.4: Breeding 2D CG Images and Movies, and Creating a type of Collage. In: The International Conference on Knowledge- based Intelligent Information Engineering Systems, pp. 288– 291. https://ieeexplore.ieee.org/document/820180/

Literature

- H. Takagi (2001): Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. Proc. IEEE 89(9), pp. 1275–1296. https://ieeexplore.ieee.org/document/949485/
- P. Machado and A. Cardoso (2002): All the truth about NEvAr. Appl. Intell. 16, 2, pp. 101–118. https://doi.org/10.1023/A:1013662402341
- S. Draves (2005): The electric sheep screen-saver: A case study in aesthetic evolution. EvoMUSART. Vol. 3449 of Lecture Notes in Computer Science. Springer, pp. 458–467. https://doi.org/10.1007/978-3-540-32003-6_46
- D. Hart (2007): Toward greater artistic control for interactive evolution of images and animation. In Applications of Evolutionary Computing, EvoWorkshops 2007, volume 4448 of Lecture Notes in Computer Science, Springer, pp. 527–536. https://doi.org/10.1007/978-3-540-71805-5_58
- T. Kowaliw, A. Dorin, and J. McCormack (2012): Promoting Creative Design in Interactive Evolutionary Computation. IEEE Trans. Evolutionary Computation 16(4), pp. 523–536. https://ieeexplore.ieee.org/document/6151108/
- J. Graf and W. Banzhaf (1995): Interactive evolution of images. In Proc. Conference on Evolutionary Programming, pp. 53–65.
- E. den Heijer and A. E. Eiben (2014): Investigating aesthetic measures for unsupervised evolutionary art. Swarm and Evolutionary Computation 16, pp. 52–68. https://doi.org/10.1016/j.swevo.2014.01.002

Literature

- A. Neumann, B. Alexander, and F. Neumann (2017): Evolutionary Image Transition Using Random Walks. In: Computational Intelligence in Music, Sound, Art and Design, EvoMUSART 2017, Lecture Notes in Computer Science, 230-245. https://doi.org/10.1007/978-3-319-55750-2_16
- A. Neumann, B. Alexander, and F. Neumann (2016): The Evolutionary Process of Image Transition in Conjunction with Box and Strip Mutation. In: Neural Information Processing, ICONIP 2016. https://doi.org/10.1007/978-3-319-46675-0_29
- A. Neumann, F. Neumann, and T. Friedrich: Quasi-random Agents for Image Transition and Animation. In: submitted for publication, CoRR abs/1710.07421. Submitted for publication. http://arxiv.org/abs/1710.07421
- A. Neumann, Z. L. Szpak, W. Chojnacki, and F. Neumann (2017): Evolutionary Image Composition Using Feature Covariance Matrices. In: Genetic and Evolutionary Computation Conference, GECCO 2017, ACM Press, 817-824. http://doi.acm.org/10.1145/3071178.3071260
- B. Alexander, J. Kortman, and A. Neumann (2017): Evolution of Artistic Image Variants Through Feature Based Diversity Optimisation. In: Genetic and Evolutionary Computation Conference, GECCO 2017, ACM Press, 171-178. http://doi.acm.org/10.1145/3071178.3071342

Literature

- A. Neumann, W. Gao, C. Doerr, F. Neumann, M. Wagner (2018): Discrepancy-Based Evolutionary Diversity Optimization. In: Genetic and Evolutionary Computation Conference, GECCO 2018, ACM Press
- G. Greenfield (2006): Robot paintings evolved using simulated robots. In Workshops on Applications of Evolutionary Computation, pages 611–621. Springer
- K. Matkovic, L. Neumann, A. Neumann, T. Psik, W. Purgathofer (2005): Global Contrast Factor - a new approach to image contrast. Computational Aesthetics, 2005:159–168
- D. Hasler, S.E. Suesstrunk (2003): Measuring colorfulness in natural images. In Electronic Imaging 2003, pages 87–95. International Society for Optics and Photonics
- J.-M. Jolion (2001): Images and Benford's law. Journal of Mathematical Imaging and Vision, 14(1):73–81