10 December 2019

Aston University
Birmingham

Artificial Intelligence for Visual Art

Anikó Ekárt
Computer Science
a.ekart@aston.ac.uk

ALICE

Agenda

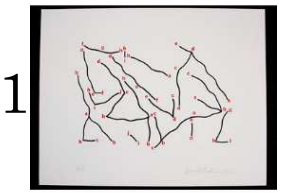
- Early computer art
- Computer art today
- Interactive evolutionary art
- Automatic programming → automatic art ?
- Let the computer learn what is “nice” art

Early Computer Art Example

Harold Cohen's (1928-2016) **AARON**

Created in 1973
Written initially in C and then LISP

In Victoria & Albert museum:

1971 

1982 

1984 

2003 

- AARON grew out of its creator's need 'to understand what art is'. In the process of designing a program to work with random variables, making its own decisions on colouring and composition, he explored the potential inherent in different programming languages.

- Cohen insisted that display labels and image credits for prints produced by AARON should read "Digital print by AARON, a computer program written by Harold Cohen"

COMPUTER ART TODAY

The Lumen prize

In its 8th year in 2019

"The Lumen Prize is a visionary undertaking that is helping the world better understand new media art."

Anne Spalter - 2015 Lumen Artist

Selected Lumen prize winners

2019
Still image award:
Drawing Operations
by Sougwen Chung

2019
BCS AI award:
Lichtsuhende
by Dave Murray-Rust and Rocio von Jungenfeld

Selected Lumen prize winners

2018
Still image category:
Overload (Consequence)
by Mark Lyons

2016
Still image category:
Fifty sisters
by Jon McCormack

2015

Founder's prize: Electric sheep by Scott Draves

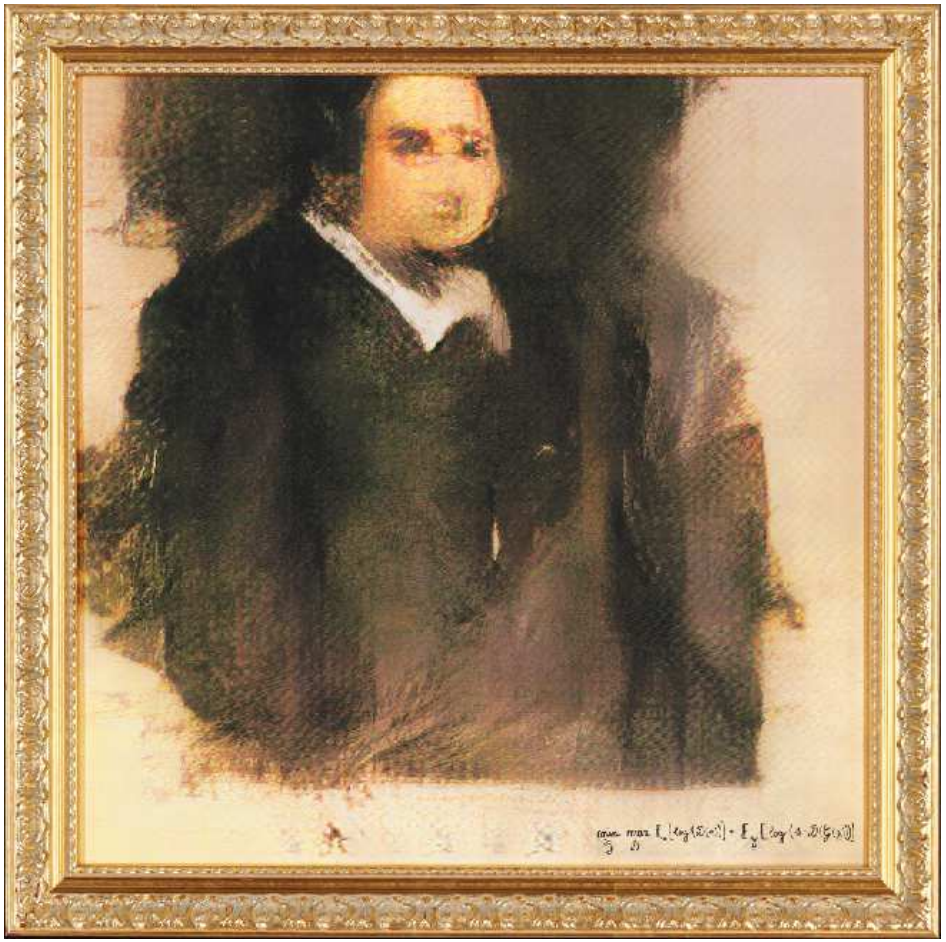
3D PRINTS

Software tools that generate 3D printable objects from mathematical descriptions

Art or not art?

In the *Prints & Multiples* sale at Christie's on 23-25 October 2018, *Portrait of Edmond Belamy* sold for an incredible $432,500

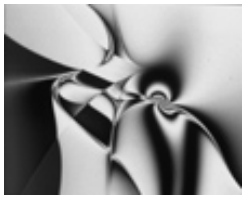
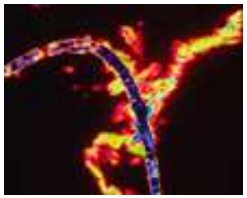
Created by a generative adversarial network (GAN)



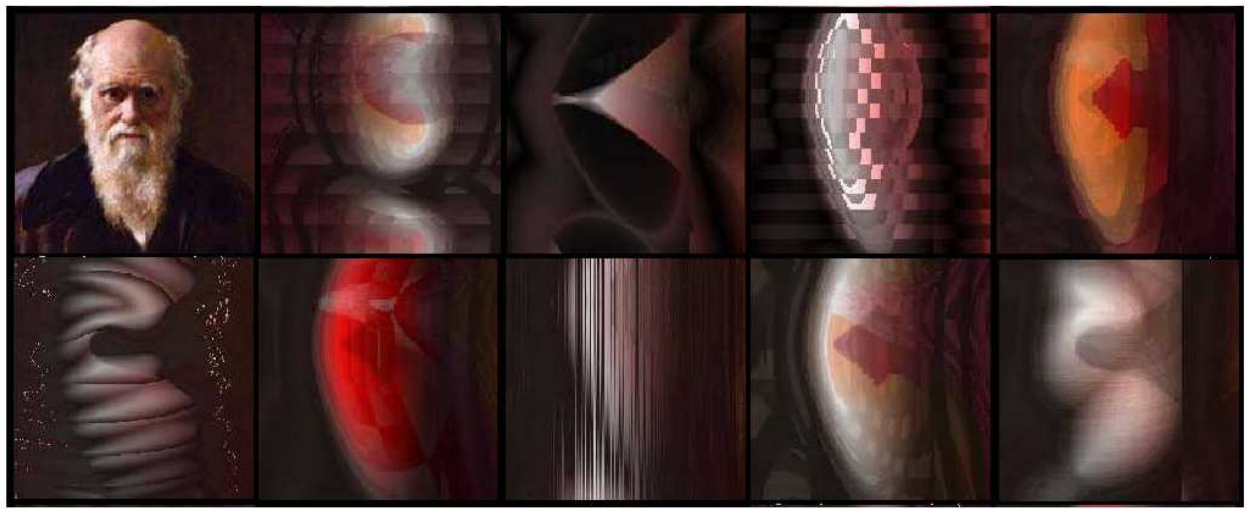
Evolutionary computer art

- Based on using some form of evolutionary algorithm to generate images/video
- Pioneered by **Karl Sims** (1991), exhibition at Pompidou Centre, Paris



- Steve DiPaola and Liane Gabora (2009), Incorporating characteristics of human creativity into an evolutionary art algorithm



Evolutionary interactive art

- The process:

1. Computer creates a number of images
2. User looks at the images
3. User selects the nicest images
4. Computer creates more images based on those selected by user
5. Repeat the above until user is happy with result

Any potential problems?

- User gets tired or bored

- So, wouldn't it be nice if …
  - the computer generated the pictures that **we wanted** without us having to do anything

The quest for automatic programming

- Arthur Samuel, late 1950s:

  How could computers learn to solve problems without being explicitly programmed?

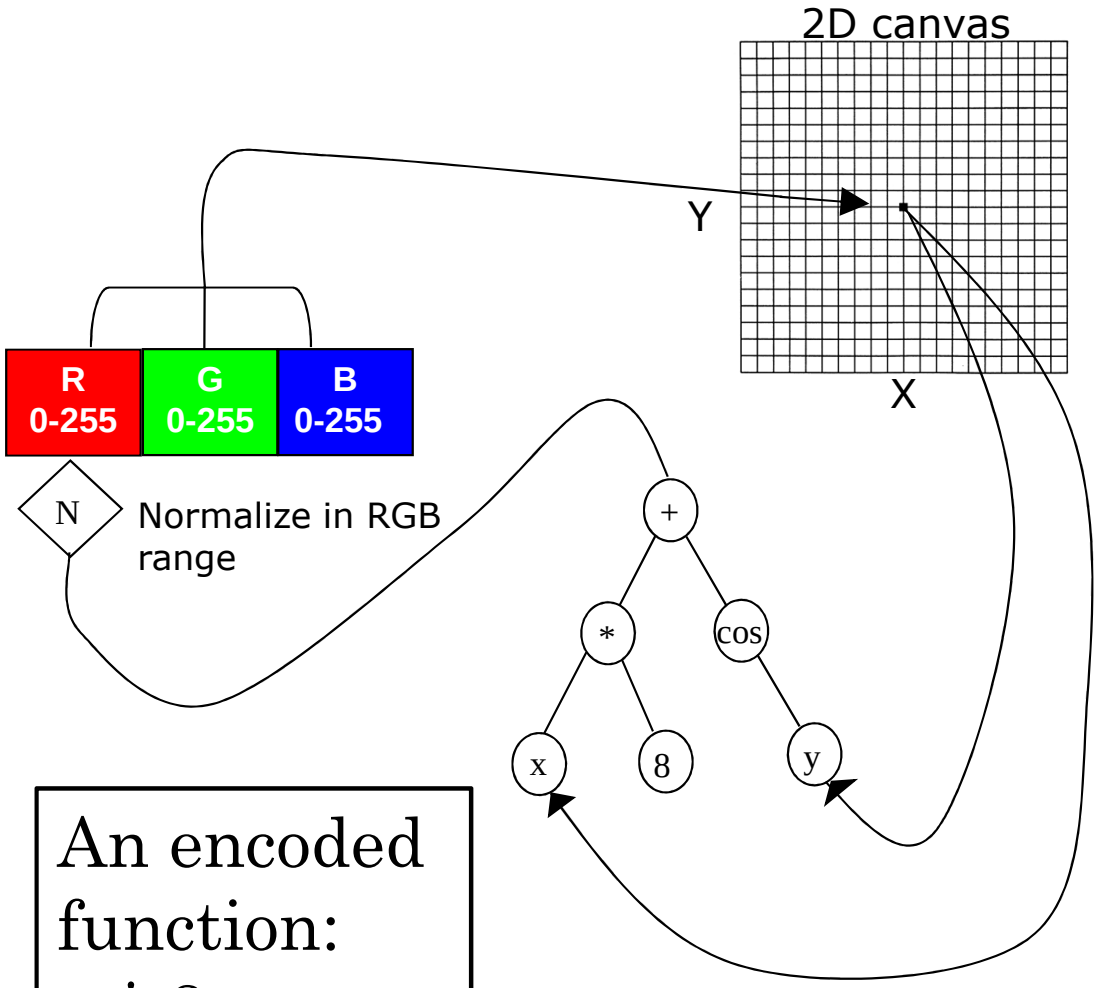
- A system is capable of automatically creating programs if
  - it starts from a high-level specification of problem requirements
  - it produces an executable program
  - it automatically determines the number of necessary steps that the program should take
  - it produces results that are competitive with those produced by **human** programmers, engineers, mathematicians, and designers

GENETIC PROGRAMMING

- The newest evolutionary computation method (earliest reference 1985)
- The key idea is to apply principles of Darwinian evolution on computer code
- Applies the principles of **artificial evolution** on computer programs represented as instruction sequences, trees or graphs
- Various forms of the so-called **genetic operators** of crossover, mutation can be defined
- **Survival of the fittest** principle is applied, where fitness is usually assessed as a mathematical function computed on the performance of the computer program

The underlying mechanisms of evolutionary art

A TWO DIMENSIONAL IMAGE REPRESENTATION

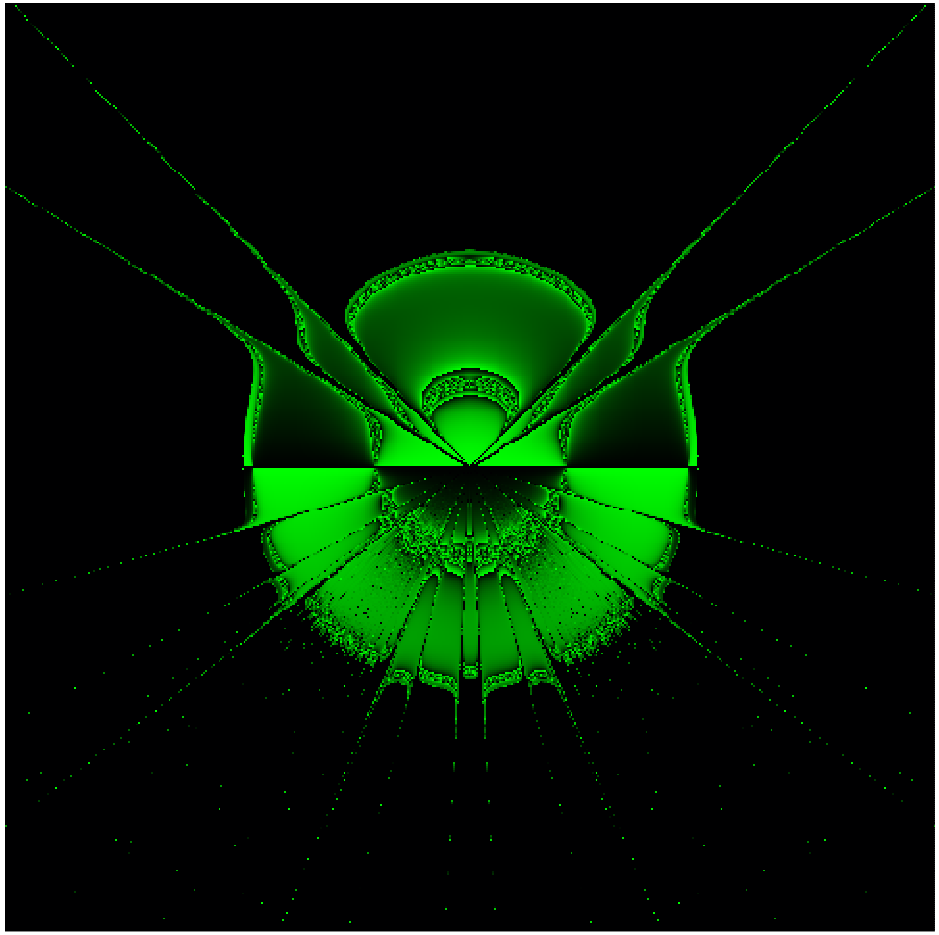


An encoded function:
x * 8 + cos y

Each point on the canvas will have a colour calculated from the three component functions for R,G and B (or a single greyscale value)

Each function is encoded as a tree

Image and associated expressions: spider



- Prefix notation expressions:

R:(MIN (COT (CUBRT THETA)) THETA)

G:(MULTIPLY (SEC (LOG (SEC (LOG (LOG (ADD (SEC (SEC (SQRT THETA))) (ADD (SEC (COSH THETA)) R)))))) Y)

B:(COT 0.7418012)

Robot design activity

via manual mutation

What is "nice"?

Nice vs complex

- **Expectation:** more complex functions are nicer, generate more "detailed" images
- **Circle:** simple, yet perfect shape in Ancient Greek culture

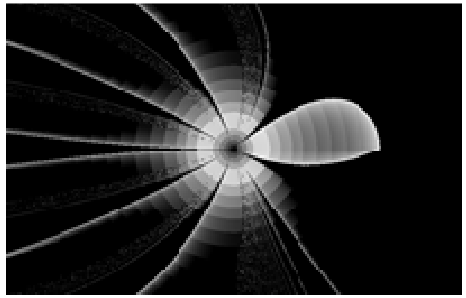
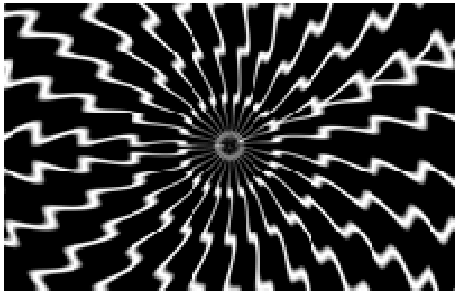
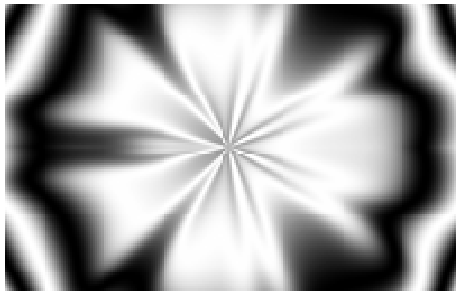
R: (MULTIPLY (COSEC X) (MIN (MIN 0.932 (COT THETA)) (MAX R (SIN X))))

G: (AVG (COSEC (HYP X (LOG THETA))) (AVG (ADD (SEC 0.591) (SUBTRACT Y 0.176)) (AVG 0.893 (AVG Y THETA))))

B: (COT (ADD 0.0050 THETA))

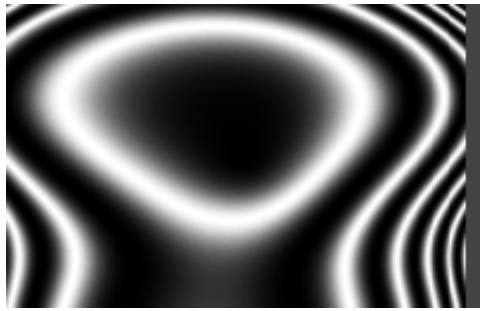
"Nice" Evo::art examples



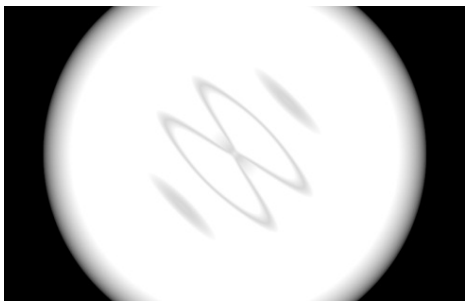

How does the computer know what is "nice"?

- Let the human artist work
- Get the computer to "watch and learn"
- Let the computer generate pictures and hope that the human artist likes the result



User's best

Machine learning
followed by
Automatic evolution



Computer generated,
liked by human

Current AI and Art @ Aston

- Edward Easton, PhD student, 3D Aesthetics in Virtual Reality (VR)
- Matthew Barlow, BSc final year student, Evolving compositions of patterns found in artist provided images
- Sean Corcoran, BSc final year student, VR interactive art experience
- Li He, BSc final year student, Using deep convolutional generative adversarial networks (DCGAN) to overlap styles, starting from traditional Japanese portrait prints (in the style of Ukio-e)
- Chairing the International Conference on Computational Intelligence in Music, Sound, Art and Design in 2019 & 2020
- Cover of ***AI matters*** magazine in 2015