



Roger Johansson Blog

.NET, Go, Distributed Programming.

Genetic Programming: Evolution of Mona Lisa

Added FAQ here: <http://rogeralsing.com/2008/12/09/genetic-programming-mona-lisa-faq/>

Added Gallery here: <http://rogeralsing.com/2008/12/11/genetic-gallery/>

This weekend I decided to play around a bit with genetic programming and put evolution to the test, the test of fine art :-)

I created a small program that keeps a string of DNA for polygon rendering. The procedure of the program is quite simple:

- Setup a random DNA string (application start)
- 1. Copy the current DNA sequence and mutate it slightly
- 2. Use the new DNA to render polygons onto a canvas
- 3. Compare the canvas to the source image
- 4. If the new painting looks more like the source image than the previous painting did, then overwrite the current DNA with the new DNA
- 5. Repeat from 1

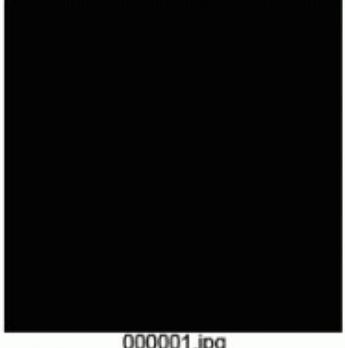
Now to the interesting part:

Could you paint a replica of the Mona Lisa using only 50 semi transparent polygons?

That is the challenge I decided to put my application up to.

The image below is the result of that test:

The number below each image is the number of generations it took to reach that specific painting.



000001.jpg



000136.jpg



000217.jpg



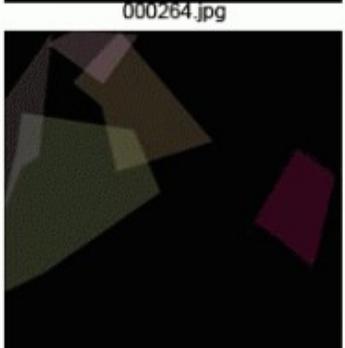
000264.jpg



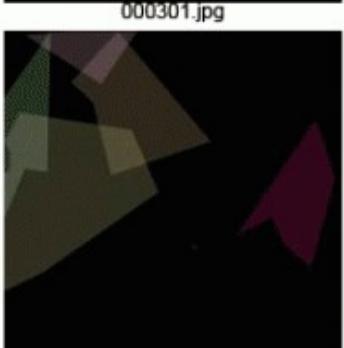
000301.jpg



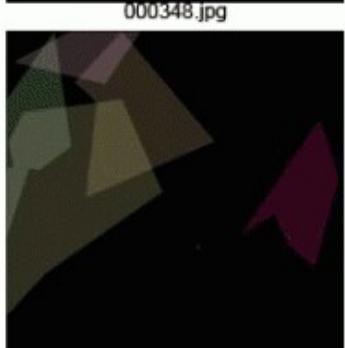
000348.jpg



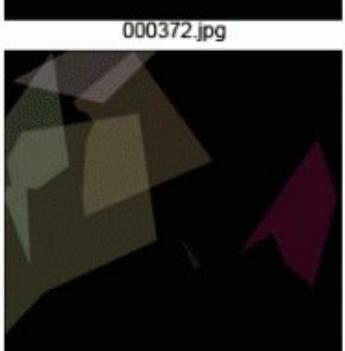
000372.jpg



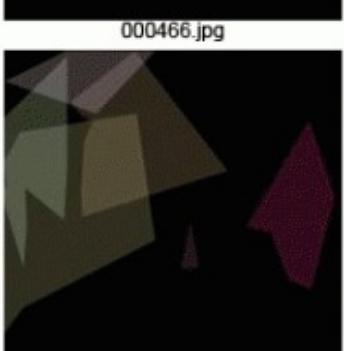
000466.jpg



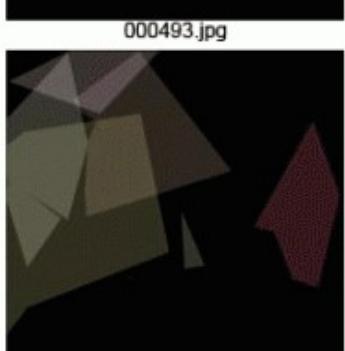
000493.jpg



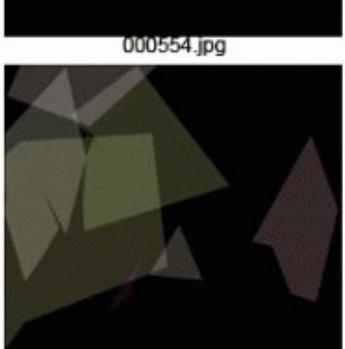
000554.jpg



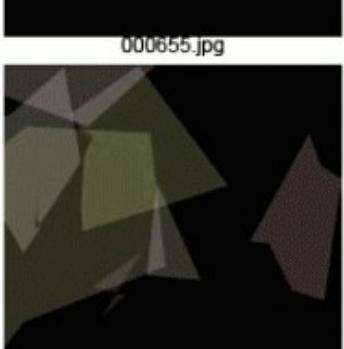
000655.jpg



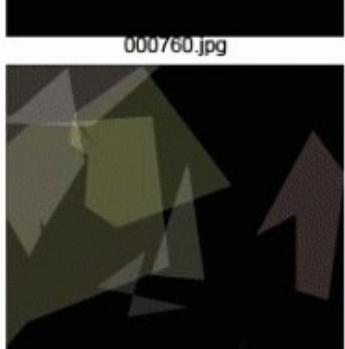
000760.jpg



000810.jpg



000977.jpg



001074.jpg



So what do you think?

[EDIT]

Added FAQ here: <http://rogeralsing.com/2008/12/09/genetic-programming-mona-lisa-faq/>

Added Gallery here: <http://rogeralsing.com/2008/12/11/genetic-gallery/>

SHARE THIS:



Like



16 bloggers like this.

RELATED:

[Genetic Gallery](#)

In "Evolution"

[Genetic Programming: Mona](#)

Lisa FAQ

In "Domain Specific Lan-
guages"

[Genetic Programming: Mona](#)

Lisa Source Code and Binaries

In ".NET"

December 7, 2008 Roger Johansson .NET, Evolution

374 thoughts on “Genetic Programming: Evolution of Mona Lisa”

BBQNinja

December 8, 2008 at 12:20 am

Any chance of source? This is REALLY cool.

Torkel

December 8, 2008 at 6:15 am

Very cool, I have been thinking about doing the same thing actually (but not using polygons, by using sort of line creatures).

What algorithm / library did you use for image comparison?

I have been pretty much obsessed with evolution stuff this past 1-2 years, read all Dawkins's books. I Can really recommend the selfish gene, and the blind watch-maker, and ancestors tale, plus Genome by Matt Ridley was great, really interesting how DNA really can be viewed as a hard drive littered with remains neutralized viruses and other self serving copying processes.

If I ever did PhD research it would be in evolution simulation :)

 **Andy Dwelly**

December 8, 2008 at 10:42 am

...interesting. If the representation of the polygons was small enough, you've probably got a new compression algorithm.

 **Searle**

December 8, 2008 at 10:47 am

Very nice! That's a vectorizer and image compressor in one. Even with scalable compression level, just add another 20 polygons.

I guess you have to trade time, though :-)

Same request: source would be nice – a toy to play with on christmas :-)

 **Andy**

December 8, 2008 at 10:53 am

Wow, your phenotype–genotype mapping seems particularly effective for this sort of thing!

Dare I ask, what is it?

**Andy**

December 8, 2008 at 10:53 am

particularly, even!

**Malcolm**

December 8, 2008 at 10:58 am

I like this! Any chance of seeing the evolution as a video?

One thing that puzzles me, why aren't there 50 polygons in the initial images?

**Veera**

December 8, 2008 at 11:05 am

This is awesome.

**Luis Antón**

December 8, 2008 at 11:06 am

Pretty impressive. However, you said the DNA sequence was randomly created... why is the first screen black, instead of showing randomly placed triangles?

**Johannes**

December 8, 2008 at 11:59 am

It's not really a genetic algorithm that produced the picture, but a stochastic hill-climber. No recombination is done, the population size is one, so the genotype-phenotype mapping is not really important. Also it seems step 0 was not executed, as others already pointed out.

And finally, looking at the number of steps it did take to produce that drawing, the convergence speed is not very impressive.

**Saiyine**

December 8, 2008 at 12:13 pm

How long does it take to generate the final result?

Do the number of points in the polygons change overtime? If it does, how do you take that in account in your DNA sequence?

**Lance**

December 8, 2008 at 12:45 pm

That's very cool. How did you do the selection? Was it automated somehow, or did you manually compare each image to decide whether to keep/discard it?

**kc**

December 8, 2008 at 1:04 pm

I am curious about your fitness function. How, for example, did you decide that 466 looks more like the Mona Lisa than 372?

**shaun**

December 8, 2008 at 1:05 pm

very very cool.

would love to see the source too :)

**Jon Lee**

December 8, 2008 at 1:11 pm

I've implemented genetic algorithms before but only for optimization. Neat application of it!

Unfortunately, seems like it takes an exponential number of iterations to get finer and finer details.

Maarten

December 8, 2008 at 1:11 pm

Fantastic! Any chance you'll be releasing the source?

**programmer**

December 8, 2008 at 1:32 pm

Very cool. If I understand correctly, the last image is comprised of only 50 polygons?

**Adam**

December 8, 2008 at 1:32 pm

Maybe they were randomly placed black triangles initially?

**d**

December 8, 2008 at 1:35 pm

At almost 1 million evolutions, how long did this take?

**chris**

December 8, 2008 at 1:40 pm

nice, but i also wonder why there aren't 50 polygons in the first pictures...

**Tristan**

December 8, 2008 at 1:44 pm

Seems to me the real magic is in two words: "mutate slightly." That is the part of the code I would like to see :)

**Lee Spector**

December 8, 2008 at 1:51 pm

Beautiful! I second Andy's query about the genotype/phenotype mapping, and would also like to see what you're doing for mutation.



Andreas Håkansson

December 8, 2008 at 1:53 pm

As always you pull of another cool test! :-)



Mike

December 8, 2008 at 1:54 pm

Cool! What compression ratio do you get from the DNA compared to the original source image?



David B

December 8, 2008 at 1:57 pm

Brilliant idea!

As per the people asking about the algorithm, I'm curious of algorithm. I imagine the genes have 50 phenotypes, each one being 3 points and 4 colours (RGBA). To show the 'construction' of the image, you just draw all of the phenotypes in (a random) order?

The fitness function sounds like it'd be fun. Objective image comparison is hard to create and might fall apart at higher image sizes. :) Are you using line tracing or something like that?



Jasper

December 8, 2008 at 2:03 pm

Some more information would be cool. How long did it run? How did you compare the picture to the actual Mona Lisa?

You had a population of one, i feel that that might be best in this case, but i do not see why exactly.

**Louis Berman**

December 8, 2008 at 2:06 pm

Wow! I am impressed. Of course you left out a number of important details in your description. For starters you state the in step 3 you “Compare the canvas to the source image”. Does that mean a simple pixel by pixel comparison, luminance averaging within a fixed or titrating grid, facial recognition, or what? As always the fitness function(s) are at the heart of a GA/GP but you give no clue.

Also you took 900K generations to get there but you never mentioned your population size, rate of crossover, mutation, etc. Basically, I'd love to know as much about the algorithms you used as you'd be willing to share. A look at the code would be best although I agree with Andy Dwelly that if the storage size was significantly less than MPG4, etc. (and the calculation effort was small enough!) you might have an interesting and marketable compression algorithm. In that case I'd work hard to get it patented and to market ASP.

Again, great work and thanks for sharing....

**Max**

December 8, 2008 at 2:06 pm

One explanation could be, that one of the mutation possibilities is adding a new polygon to the generating DNA string.

**Paul**

December 8, 2008 at 2:20 pm

Luis: I'm guessing many of the polygons were initially degenerate in some way?

**Philipp Lenssen**

December 8, 2008 at 2:44 pm

Excellent! I once started a somewhat related project with mutating polygons nearing a likeness to a face. Here's a results gallery: <http://blogoscoped.com/archive/2007-10-09-n18.html>

Pingback: [Google Adsense Tips & News » Blog Archive » Polygons Evolving Into Mona Lisa](#)

**Anon**

December 8, 2008 at 2:56 pm

@Luis: probably because it starts off with DNA being initialized to a blank slate (even though it's not stated in the steps) and then generates the random polygons. So in this case, the black screen was a closer match than random polygons.

**Dan**

December 8, 2008 at 3:06 pm

This is seriously impressive. How long did it take to reach that final image? What algorithm did you use to compare the images?

Is it really Genetic Programming though? From your description it sounds more like some other kind of evolutionary algorithm. It would be interesting to run it again with a larger (>1) population and cross-over. I'd imagine it would converge much quicker.

**Stephanea**

December 8, 2008 at 3:19 pm

Very nice example of genetic programming. Would it be possible to have more details on the algorithm?

**Johan Hägg**

December 8, 2008 at 3:34 pm

I would say it is using Genetic Algorithm and if the algorithm is correct it sounds like single individual elitist GA. The easiest fitness function would be to do a per-pixel RGB comparison and optimising for lowest possible fitness value.

It might fall somewhat under the GP category if the number of points in the polygons is variable. The easiest approach would probably be to use triangles with each triangle represented as 3 positions and RGBA-components.

Going to try this out using Covariance Matrix Adaptation Evolution Strategies once I have more free time. This might actually result in a fast enough optimisation to make this a viable method for vectorising and compressing images with high complexity.

**Kalle**

December 8, 2008 at 3:40 pm

@Dan: The algorithm as describes sounds like a simple form of simulated annealing without a probability of accepting worse matches with a probability > 0 and no cooldown. Another algorith that could work is MCMC (Markov chain Monte Carlo), which is usually used with bayesian statistics and probabilities, but can easily applied as a generic optimizqation algorithm.

**John Leavitt**

December 8, 2008 at 3:42 pm

Source, please? Otherwise, I'm going to end up wasting time rolling my own instead of working.

**swaits**

December 8, 2008 at 4:01 pm

I doubt this is GP. Probably a GA.

Pingback: [Genetic Programming: Evolution of Mona Lisa - The WebZappr](#)

 **Alex**

December 8, 2008 at 4:02 pm

Compression through GA search is not a new idea. I have tried something similar to this myself a while ago.

This implementation works better because of the alpha-blended polygons though.

 **silveira**

December 8, 2008 at 4:12 pm

Show me the code.

 **Mihai**

December 8, 2008 at 4:16 pm

The comparison function could be as simple as the quadratic error, $\text{sum}(\text{square}(pxy - qxy))$. I used it in fractal image compression. (Note the applications of this to fractal image compression? ;))

It's interesting how one can get close to the target in the presence of a fitness function. There's, however, a drawback to the algorithm that's not immediately apparent. The approach is entirely greedy and with this limited population of (two) images at each generation you won't be able to take the image through depression points. You don't notice this, because your tools (polygons) are coarse grained and hence the goal is not a very fine picture to begin with.

In any case, impressive.

 **Matto**

December 8, 2008 at 4:16 pm

Wow, that's great!

**Martin**

December 8, 2008 at 4:18 pm

Dude, any chance of posting this up on youtube as a proper video/animation?

Not sure about how effective this is as a genetic algorithm, but you could have a real gem of a video effect here!

**Hugh Bothwell**

December 8, 2008 at 4:20 pm

Nice! I would be interested in more detail:

What limit did you put on the maximum degree for each poly? Did you considered limiting the total number of vertices rather than the number of polygons? How does varying the poly limit affect the final image? (ie what is the limit of convergence for different poly counts?)

What did your mutation function look like – shift-vertex, add-vertex, tweak-color, tweak-alpha, what else? What relative-occurrence-probabilities did you use?

Do you use edge-crossing-detection to ‘untwist’ the poly, or just ignore crossovers? Do you limit new-vertex locations, ie midpoint-offset, or just use random-in-canvas? Did you consider a vertex-pruning mutator? Have you got any statistics on result-acceptance rates for different mutators? How do these rates change at different stages of convergence?

What do you do when you hit the max number of polys – drop the least-fit, or just stop creating new ones? If drop, how did you determine least-fit?

What color space did you use, RGB/LSV/? What measure did you use for image-comparison (sum of absolute pixel-differences?)? Have you considered using variable-resolution comparison to speed things up (ie start with a lo-res image and increase resolution as fitness increases)?

Is it possible to calculate per-poly best-fit alphas as part of the image comparison? If so, can we do the same for each color channel?

It sounds like you used population-1, ie random-walk search. Any idea what a valid crossover operator would look like – something like polygon-swapping with alpha adjustment?

Have you tried using smart/directed mutators – ie, pick one poly to mutate, cache the image-and-difference for everything-but, then try a set of mutations on the poly and use the one which increases the fitness the most?

Have you investigated how changing some of the parameters affects convergence speed in terms of either generations or overall computation time?

For fun – what about using this alg to morph from one image to another?



Kevin

December 8, 2008 at 4:23 pm

Very nicely done.



NoMoKeTo

December 8, 2008 at 4:26 pm

Could you please publish the result of the test as an SVG? That would be really interesting ;-)

(Sorry for my bad english)

Pingback: [Mona Lisa: Evolution durch künstliche Intelligenz KI - Von Alper Iseri - Roger Alsing, Mona Lisa, Lisa, Mona, Original, Bild, Programm, Vorgänger, Wenn, Vergleiche - meetinx-Blog](#)



Przemek Klosowski

December 8, 2008 at 4:40 pm

The representation of triangles needs 3 vertices each having X and Y coordinates, plus the color (R,G,B), so 9 numbers per triangle, times 50, i.e. 450 numbers. This could be 1800 bytes if they all were

single precision floats,
or we could do integer arithmetic, with say 16-bit
X/Y coordinates, and 24-bit color, i.e. 19 bytes per triangle, or 950 bytes total.

 **mempko**

December 8, 2008 at 4:48 pm

I would be curious to see this where every corner of a triangle has a different color and interpolate the polygon colors to create a gradient. You might be able to get a more faithful representation that way and possibly be able to reduce the amount of polygons.

 **Alex**

December 8, 2008 at 4:57 pm

To silveira: I do not have the code right here but I can sketch the approach: the gene size was variable, encoding a variable number of rectangles with a colour. The aprroach described in this page has a fixed number of arbitrary polygons and the colour combination function is some type of blending. I think the variable number of polygons is a better feature but this experiment shows that having general polygons instead of rectangles and specially blending colours (which makes the fitness function change more smoothly) is a good improvement. I wasn't trying to solve exactly the same problem though.

You can do a google search on ‘genetic algorithms image compression’ and you’ll see a few hits on the subject.

 **Martin**

December 8, 2008 at 4:59 pm

Having now made a movie out of it myself, it is a lot easier to see what is going on.

I think it would be interesting to place some limitations on the evolution:

- 1) each polygon is limited to up to n-sides, or
- 2) the total number of vertices is limited to some number.

Or both.

Also, I too would like to know more about your fitness algorithm. I would've thought that some regions would've gained fidelity (detail) more quickly than others, but it seems — from the samples you've provided — that detail sort of emerges together.



Eric Anderson

December 8, 2008 at 5:02 pm

Source or it didn't happen :P



Leif902

December 8, 2008 at 5:13 pm

Very nice; You should generalize it and apply it to other images as a way of converting from raster to vector formats.



BobR

December 8, 2008 at 5:14 pm

LHOOCube.



Thinking Out Loud

December 8, 2008 at 5:21 pm

Hmmmm.... I don't think this is a true test of Evolution, though... The algorithm always had access to the original "Design" of Mona Lisa to compare itself with. The process and theory of evolution states that there is no Designer and that things happened randomly (uncoordinated). So, for me, evolution requires MORE faith than believing in God or an intelligent Designer.



Ian

December 8, 2008 at 5:30 pm

Source Code or it Didn't Happen!

 **Andy Baio**

December 8, 2008 at 5:34 pm

Adding to the chorus of people wanting to see this rendered as a movie!

 **Robert**

December 8, 2008 at 5:39 pm

Post the source or gtfo

 **Roger Alsing** 

December 8, 2008 at 5:41 pm

OK OK , I Will try to make a movie out of this.

I'll even try to make a new run and output SVG or some other vector format out of the images :-)

Also, the polygons are 3-x points large, so they are not only triangles but can be big complex polygons.

(Saw a few comments about triangles in here)

And I'll blog about the various aspects on this topic as soon as the kids go to bed.

 **Robb Greathouse**

December 8, 2008 at 6:05 pm

Could you post the code? It would be interesting to see it. Especially the code used to compare to the original and estimate the match.

My understanding is that that is where the magic really happens and the most difficult thing to design.

Pingback: [uglychart.com: a blog about stocks » Blog Archive » links for December 8th](#)

**franco**

December 8, 2008 at 7:07 pm

source or no dice.
provided that, very nice.

**Andreas Håkansson**

December 8, 2008 at 7:36 pm

About that first black image. I would guess its the first generation and selecting a strand with all initial values as zero is just as good as a random starting strand.

Basically you need something to compare the next generation with and a black screen works? =)

**Johan Lindfors**

December 8, 2008 at 7:51 pm

Once again you manage to both impress and inspire! Extremely nice and interesting. I guess this was all done in 'M' :)

**realistic bird**

December 8, 2008 at 8:04 pm

Nice! I can't say I understand how you did this but it is very cool, bravo.



Just curious

December 8, 2008 at 8:21 pm

This is thoroughly awesome. It really is.
Just curious – how long did the computation take ?



Nick

December 8, 2008 at 8:43 pm

I wouldn't call this a genetic algorithm because you are not keeping multiple candidates at each iteration nor breeding them (that is, swapping DNA from multiple candidates). This is simply a straightforward hill climbing algorithm. Not that the method impacts on the results.

If it took about 1M iterations to get to your result with hill climbing, I would expect you could do much much better with a genetic algorithm provided you were smart about how DNA got swapped. If it was swapped spatially, somehow, then you could get a result where a candidate which was good at the face bred with a candidate which was good at the background and you got a candidate which was good at both.

Which isn't to say that such an approach might not take more computation over all, keeping around and breeding multiple candidates takes time.

Pingback: [Extremely cool: genetic programming evolves picture of Mona Lisa « Later On](#)



Julian Togelius

December 8, 2008 at 8:58 pm

I agree with everybody else here:

1. Cool stuff

2. What's the fitness function?

3. What's the mutation operator?

 **Rich**

December 8, 2008 at 9:21 pm

Hey, instead of so narrowly directing the evolution, you could do something more like real life, by coupling a facial recognition program to the fitness testing. See what kinds of ugly mugshots you get coming out!

Pingback: [ILLUZIVE – Personal » Blog Archive » Genetic Programming: Evolution of Mona Lisa « Roger Alsing Weblog](#)

 **kaiser**

December 8, 2008 at 10:10 pm

It's not evolution it's intelligent design ...

 **rip**

December 8, 2008 at 10:18 pm

excellent :D

 **PsySal**

December 8, 2008 at 10:18 pm

This is really cool, dunno about applications but really surprising how good it looks with only 50 polys...

 **TimT**

December 8, 2008 at 10:23 pm

also would like to see/use the source ;)

**Roger Alsing** ♀

December 8, 2008 at 10:28 pm

I will release the source tomorrow, I just have to clean it up so I don't have to feel too ashamed of the actual code quality.

(Threading it so that it won't lock up your computer etc)

The app is written in C# 3 using .NET framework 3.5.
So that's the requirements for using/compiling it.

Just so you know :-)

Im off to bed.

**v_krishna**

December 8, 2008 at 10:33 pm

source or more information, please? (in particular, I'm interested in how you compared particular states with the actual mona lisa)

and, like others said, this isn't really genetic programming (as there is only a population of 1), is it? more like a genetic algorithm?

**noname**

December 8, 2008 at 10:48 pm

It is really impressive, but I would also like to know more details about the algorithm. I think it would be also interesting to let fight several runs of the same test to find better result with same number of polygons (or similar with less number). I also vote for canvas version here ;)
And maybe we can all try to optimise the algorithm

**noname**

December 8, 2008 at 10:50 pm

and i would also like to see simillar test but with more limitations (3 sides – triangles only for example).

 **Fester**

December 9, 2008 at 12:42 am

Man, you are really a genius. That is a way too great.

 **Benjamin Smith**

December 9, 2008 at 12:46 am

Fascinating stuff! Would love more details!

 **Dan**

December 9, 2008 at 1:15 am

@Kalle, sounds closer to Evolution Strategies, particularly the (1+1) variant.

Pingback: [The World's Most Famous Painting : LikeItHateIt](#)

 **David Goscinski**

December 9, 2008 at 3:39 am

Interesting, but it isn't evolution. You just showed intelligent design. You had an end result that you compared your program to. So you had an ultimate goal in mind. You didn't really do it "randomly" though. So thanks for showing designs have designers. :)

 **Krylen**

December 9, 2008 at 3:55 am

Incredible! As a programmer of genetic algorithms myself I must tip my hat to your ingenuity. For even I, with no 3D programming experience could write something to combine 50 semi transparent polygons, however the ingenuity of

thought, the stroke of brilliance to apply such a seemingly impossible task to such simple constraints, that! is true genius.

tips hat

Pingback: [Genetic programming – Hack a Day](#)



cydeweys

December 9, 2008 at 4:15 am

Just to echo a lot of other people, I'd really love to see the source. I've been fascinated by genetic algorithms ever since I first heard of them eight years ago, and I've long wanted to experiment with them myself, but the initial hurdle of coming up with a GA framework on my own has proved to be too much. If you could provide the source for this project (say, under the GNU GPL), that'd be a great starting point for me and many others. I'd start by evolving a different image, and then from there, who knows what I'd do?

Thanks a lot.



prfx

December 9, 2008 at 4:24 am

I'm impressed.



AE

December 9, 2008 at 4:48 am

While I find this very interesting (and, frankly, cool), I hardly think this could be used for a new image compression algorithm — at least not in any practical manner.

First, it's likely to be hugely costly in terms of memory and computation in order to do so (almost 1M generations, not to mention an unknown population size). It's probably also quite likely that the polygons evolved in the process are non-trivial. What you're really looking at is the building of an SVG. Therefore, it's possible

that this could be used either by itself or perhaps in tandem with an existing algorithm to convert raster images to vector representations.

I'm skeptical as to whether there's any commercial viability to such code. If I'm proven wrong, then kudos! But not every project is meant to be commercialized.

Good work!

Pingback: [The mona lisa « Stever's blog](#)

Pingback: [Matrix67: My Blog » Blog Archive » 强大的遗传算法：用50个半透明多边形重现蒙娜丽莎](#)

 **huyichen**

December 9, 2008 at 5:23 am

Then is life designed?

 **Karl**

December 9, 2008 at 6:40 am

I'm wondering if the 50 polygons are particularly suited to rendering a face. Might 50 polygons have a harder time rendering "scream?" Would 50 polygons be really good at rendering a picasso?

 **MrE**

December 9, 2008 at 6:52 am

For all those people who want the code, why don't you just follow the procedure that's explained at the top of the page?

If you can't do that by yourself, then what the heck are you going to do with code except get confused?

**jeremiah**

December 9, 2008 at 6:53 am

<http://opensource.arc.nasa.gov/project/javagenes/>

Nasa has had an open sourced genetic framework up for about 5 years now.

might be a good starting point for you genetic developers out there.

**Alto**

December 9, 2008 at 6:55 am

Sorry but I think you are not using a genetic algorithm at all like Johannes and Johan mentioned. Please change the title and description to refelect this since you are misleading other people.

**Daniel Duder**

December 9, 2008 at 7:10 am

For MrE

Why don't you write the code based on the procedure and then explain it to us such that we won't get confused with your code

**echelon**

December 9, 2008 at 7:14 am

Are you sure this is GP or GA? Unless some of your polygons are being drawn off the canvas, I'm not quite following. Since the image is slowly converging by *adding* polygons and not mutating their positions, it looks like hill climbing to me.

**Bummed**

December 9, 2008 at 7:21 am

Damn. I did this same thing a year and half ago. Didn't think there would be a

Wired article about it.



Dred Poirate Nanicoar

December 9, 2008 at 7:32 am

> So what do you think?

Spontaneous reaction: Oh wow.



kroll

December 9, 2008 at 7:55 am

Very cool. I agree with previous posters though: this is a hill climber, not an EA. I'm guessing a hill climber is likely to work well in the given fitness landscape. Depends on the fitness function of course, but it would likely have no local maxima, making a hill climber more effective.

Still: Seeing Mona Lisa slowly appearing like that is really cool.



kakaz

December 9, 2008 at 8:28 am

It is interesting to note, that 10^6 generations is equivalent for example $57 * 10^3$ years of evolutions of drozofila. So: how complex structure may be evolved by this short time...



Gods1Child

December 9, 2008 at 8:35 am

Eerie and very cool!

**Jan**

December 9, 2008 at 8:48 am

Really cool, nice work Roger!

**Itay**

December 9, 2008 at 8:53 am

Kakaz: Because each iteration happens immediately after the other.. no need to wait for the old generation to “reproduce”.

**psycho sparky**

December 9, 2008 at 8:54 am

Is nothing sacred ?

Sodding cubists !@#

**Irregular Activity**

December 9, 2008 at 9:00 am

That is a phenomenal piece of work, colour me very impressed, am very much looking forward to the source tomorrow – or should that be today?

**amoe**

December 9, 2008 at 9:15 am

Every mention of evolution shouldn't turn into a god vs. science debate, but I'm going to fuel the fire anyway. To those saying that this is no proof of evolution: true, the algorithm did have access to the 'finished product'. However, the final form provided to the algorithm (AFAIK) is a raster image, not the polygons from which it is formed by the algorithm. In the same way, the final form dictated by evolution is survival, but the method to obtain that isn't completely specified.

Pingback: [Evolution Proof](#)

 **Eddie**

December 9, 2008 at 9:39 am

Yes I agree it's not a genetic algorithm, but I think the convergence rate is fine. Why is that an issue? He's come up with a good, 50-polygon approximation with a "population size" of 1 and less than a million iterations. This is a reasonable amount of work compared to the true GAs I've seen with populations of thousands and needing thousands of iterations. But, more to the point, why optimize a one-shot calculation? That would be premature ...

To the ID straw-clutchers, this is not evidence that mankind evolved through ID. Evolution is not purely random; it occurs within an epigenetic landscape which includes (at least) the fixed laws of physics and chemistry. This calculation uses an epigenetic landscape of the Mona Lisa.

A good example to demonstrate true evolution to the non-believers would be to have a million people rate the output of the algorithm and evolve it using their opinions. Then the output would still be "designed", but by a million people rather than one all-powerful "designer".

Now create a statistical approximation of the ratings you got, and use that instead of people to rate the pictures. I bet you'd get a picture out which was aesthetically pleasing but had no designer. That'd be a nail in the coffin for those who believe ID because they don't understand evolution :)

Pingback: [Geeko! » Dibuja la Mona Lisa con 50 polígonos](#)

 **Vincent**

December 9, 2008 at 9:56 am

In your face, creationists.

Pingback: [La Mona Lisa dibujada con 50 polígonos | ...: Cristian Eslava | Diseño](#)

Gráfico / Web | Maquetación | Formación ...

**ik**

December 9, 2008 at 10:00 am

How about creating three sets of polygons, each set for one channel in YCbCr -color space. They could still be alpha blended in this stage.

Pingback: [appler's me2DAY](#)

**Weird**

December 9, 2008 at 10:43 am

Could the same algorithm be easily converted to use gradient meshes?

Pingback: [Creative Computation | Genetic Algorithm Paints Mona Lisa](#)

**Licenser**

December 9, 2008 at 10:55 am

Wow, this is certenly impressive. Admited, I personaly won't ever need somthing like that then again screw practical use! It is just cool ;).

I too can't wait to have a look at the source (even so I'll have a hard time to run it w/o windows) but I doubt it will take long and this algorithm is ported to a few other languages :)

Well done!

**vokimon**

December 9, 2008 at 11:00 am

What about using it as morphing tool just by changing the goal?

**Evolutionist**

December 9, 2008 at 11:01 am

It doesn't really have much to do with nature's evolution when you set up a specific goal, and discard everything that doesn't take you one step closer to it. This is a misconception that many people on both sides seem to have.

**Nick Begich III**

December 9, 2008 at 11:29 am

This could offer an interesting approach to image compression or even more importantly raster to vector conversion.

**Gearoid**

December 9, 2008 at 11:36 am

very impressive. Do you use 50 polygons at the start? , it seems as though the algorithm adds them progressively.

**Pete Austin**

December 9, 2008 at 11:50 am

An etch-a-sketch picture is ONE polygon with a huge number of sides. I've seen TV interviews with artists who can produce superb quality pictures just using this child's toy, so it's quite surprising that this code needs 50x as many polygons for the Mona Lisa.

As previous comments have said, you should try to limit the number of sides or vertexes to make this impressive.

**Eike**

December 9, 2008 at 12:00 pm

"I have been pretty much obsessed with evolution stuff this past 1-2 years, read all Dawkins's books. I Can really recommend the selfish gene, and the blind watchmaker, and ancestors tale, plus Genome by Matt Ridley was great, really in-

teresting how DNA really can be viewed as a hard drive littered with remains neutralized viruses and other self serving copying processes.”

Problem with these two guys is that they write nicely, but have very little actual knowledge of what they’re talking about. Neither has any background in genetics – let alone modern genetics (genomics/postgenomics) – that goes beyond undergrad level. Ridley is a passable zoologist, while Dawkins just writes nicely, but is not even a biologist at all and has fscked up the meaning of “gene” thoroughly and beyond redemption, thank you very much NOT: when Dawkins says “gene”, it may be anything between a single allele and a multigene complex like the bacterial flagellum and its chemical “rotary motor”.

If they were correct, how come life ever progressed beyond protists? The price you pay for multicellularism is death, and that’s where it is hard to argue for “selfishness”. Sexual reproduction, as far as can be told, evolved not coincident with terminal mortality.

The present example brings the point across rather nicely: no single “DNA sequence” benefits from being too “selfish”. They have to tread the middle ground between resource-hogging but replicating abundantly, and evolving but dying out for lack of replication carefully.

Try Nature Reviews Genetics – there, you’ll find the real hardcore stuff. Dawkins and Ridley are, from a philosophical perspective, rather annoying logical positivists, with all weaknesses for their theories this implies: they only tell you what fits. That there are counterexamples galore, they’ll never tell you.

 Caligula

December 9, 2008 at 12:08 pm

Hm. I like that.

@Evolutionist:

Anthropomorphized “goal”: reproduction and survival (micro and macro). Fitness algorithm: reproduction and survival. Generation production: reproducing and surviving, or the opposite.

How “much” does it have to do with “nature’s evolution” before it has *some-

thing* to do with it? And what's your point, anyway? It's still G(evolutionary)P– which is all that's claimed.

**ibsteve2u**

December 9, 2008 at 12:09 pm

So survival is no longer a specific goal, Evolutionist?

**Eike**

December 9, 2008 at 12:26 pm

“It doesn’t really have much to do with nature’s evolution when you set up a specific goal, and discard everything that doesn’t take you one step closer to it. This is a misconception that many people on both sides seem to have.”

Indeed. You can only assume temporary local optima; the “global optimum” is a misconception that smacks of ID, the pareidolia of the intelligent observer.

“Un dimanche après-midi à l’Île de la Grande Jatte” might be a better work to start with. But notwithstanding, you’d need an optimization against locally differing *variables*.

You might *not* need randomness except in the “mutation” and as a starting condition however. The oldest biogenic stromatolites are more than half as old as this planet itself, so the first *nonrandom structures* created *by organisms* must be older.

(Dawkin’s one *really* great idea was the extended phenotype. For anything else, refer to current peer-reviewed journals.)

**The Professor**

December 9, 2008 at 12:34 pm

For those interested in genetic algorithms, I have a program (in C++) which encodes words into a Boggle grid. The source is freely available to view and play with. It is also written using templates so you can modify it easily to work on other genetic optimization problems.

Look for the “Inverse Boggle Encoder” at <http://professorguy.com/ideas>

Pingback: [Evolving Art with Nature Inspired Algorithms](#)

 **BearOfNH**

December 9, 2008 at 12:38 pm

Marvelous!

Are the polygons always convex? Looking at the samples I don't see any that aren't.

 **TaraLi**

December 9, 2008 at 12:42 pm

Evolutionist: The goal and discard is simply substituting for the “fittest” criterion of Nature's “survival of the fittest”. However, earlier criticism is correct to some degree – this is a very limited version of “genetic programming”.

 **Another evolutionist**

December 9, 2008 at 12:59 pm

I agree with Evolutionist – this is goal-directed evolution (more like – egads – intelligent design). A truly evolutionary algorithm would compare each picture and the LAST picture against a shifting set of environmental conditions. The order emerges from that process over time.

However, as a demonstration that a randomised process can produce a highly non-random outcome, its certainly nice. As others have identified, the exact criteria for accepting or rejecting each picture as closer than the last is critical. As ecological modelers say – the devils in the assumptions.

 **Some geek around**

December 9, 2008 at 1:14 pm

Guys, since no sources, it was just a “CorelDraw!” So do not bother. :-P

 **Jeremy**

December 9, 2008 at 1:22 pm

sigh As someone with a degree in AI, (and written GAs that did vaguely the reverse... sought a set of transparent rectangles to best classify images) I've got to apologize for all the misguided, and even downright stupid comments that have been posted.

Your description was perfectly adequate to code an equivalent. I'm not at all surprised by the result, though it is a wonderful demonstration.

It's a shame people keep trying to read complexity into the algorithm, rather than appreciating it's sheer simplicity. I hope this becomes a textbook example.

Finally, All GA's are “hill climbers”, (variants of simulated annealing) just with different randomizing functions and population sets. Just because you only evolved one code doesn't make it any less a GA. There might have been efficiencies in using a thousand parallel codes and cross-breeding, but it's mathematically the same thing.

Kudos to you.

 **Eduard**

December 9, 2008 at 1:23 pm

I would love to see the source code and make it trully genetic. We have here a climbing hill solution for 1.000.000 individuals. What would be the result for a 1000 generations of 1000 individuals with true genetic algorithm? What will it be with 10 generations of 100.000 individuals each? Looks funny to try.

Pingback: [neonascent » Blog Archive » Evolution of Mona Lisa](#)

 **urodacus**

December 9, 2008 at 2:11 pm

My favourite is iteration 0004333.jpg

Pingback: pligg.com

Pingback: [Genetic Programming: Mona Lisa FAQ « Roger Alsing Weblog](#)



Roger Alsing

December 9, 2008 at 7:35 pm

Comments have been locked.

Please see the Mona Lisa FAQ post instead.

<http://rogeralsing.com/2008/12/07/genetic-programming-evolution-of-mona-lisa/>

Pingback: [Evolving the Mona Lisa by natural selection | Cow's Blog](#)

Pingback: [Geweldig nutteloze AI | GeekSpeak](#)

Pingback: [/dev/Kico » Blog Archive » Programação Genética: recriando a Mona Lisa com apenas 50 polígonos!](#)

Pingback: [BloGals » Archive du blog » Programmation Génétique – Genetic Programming](#)

Pingback: [What Good Is Half A Machine? | MetaFilter](#)

Pingback: [P3næus monodon » Blog Archive » I, For One, Welcome Our New Software Overlords.](#)

Pingback: [Top Posts << WordPress.com](#)

Pingback: [Top Posts << WordPress.com](#)

Pingback: [links for 2008-12-09](#)

Pingback: [Genetic Programming: Evolution Of The Mona Lisa | The Current Buzz - Tech](#)

Pingback: [Hypnotizing Rabbit » Blog Archive » Genetic programming and the Mona Lisa](#)

Pingback: [Shorter Than Ever | Lycanthropia](#)

Pingback: [Evolution of Mankind Art - Evolution of Mona Lisa Through Genetic Programming](#)

Pingback: [Querystring » Genetic programming](#)

Pingback: [Bram.us » The Evolution of the Mona Lisa - Genetic Programming](#)

Pingback: [The Mona Lisa recreated using 50 transparent polygons... | About Colon Blank :: The Procrastinators Wet Dream!](#)

Pingback: [遗传算法与蒙娜丽莎 << Ryanchen's Weblog](#)

Pingback: [Evolution of Mona Lisa – today and tomorrow](#)

Pingback: [Genetic Programming – The Prophecy Forums](#)

Pingback: [Free Gadget News » Algorytmy ewolucyjne w obrazkach](#)

Pingback: [Another Evolutionary Painting « Gems Sty](#)

Pingback: [Genetic Programming: Evolution of Mona Lisa « Roger Alsing Weblog :: MezzoMondo](#)

Pingback: [Genetic Programming Recreates Mona Lisa « MRod says:](#)

Pingback: [leipies-bloggy » Blog Archive » Genetic search and Art](#)

Pingback: [Auto\(- \)evolution « allzutaegliches](#)

Pingback: [A evolução da Monalisa | ArrobaZona.com | Sua zona de fragmentos digitais](#)

Pingback: [Bunch of links – #5 « PICDIT](#)

Pingback: [Image Evolution using Javascript and `canvas` | Phil Nelson Writes Here](#)

Pingback: [Evolving the Mona Lisa « The Maas:Media](#)

Pingback: [ویگردی در دنیای مجازی » روند تدریجی تکامل را در یک عکس تجربه کنید](#)

Pingback: [Coding is like gardening... with less mud » Genetic Algorithms: Approximating Fine Art](#)

Pingback: [Genetic Algorithms in Perl « Dabbler](#)

Pingback: [Extremely Late Renaissance » Mona Lisa Underdrawn](#)

Pingback: [Evolving the Mona Lisa « Twisted One 151's Weblog](#)

Pingback: [Google Adsense Tips & News » Blog Archive » Polygons Evolving Into Your Custom Picture](#)

Pingback: [Wordpress Update | Rhinosphere](#)

Pingback: [A single snowflake in the fractal explosion of technology](#)

Pingback: [links for 2008-12-11 : Eggplantia5](#)

Pingback: [jon abad dot com · I'm told that I am the internet.](#)

Pingback: [הרי חדשות » Blog Archive » 11/12/2K8](#)

Pingback: [Navigator Technology » Blog Archive » Genetic Programming: Evolution of Mona Lisa](#)

Pingback: [Genetic Gallery « Roger Alsing Weblog](#)

Pingback: [Mona Lisa aus Polygonen in 900.000 Schritten : Netzfischer.EU](#)

Pingback: [ZihuAztlan » Blog Archive » La Mona Lisa mediante Programación Genética](#)

Pingback: [links for 2008-12-11 | chrisgrice.com](#)

Pingback: [Friday Procrastination: Link Love : OUPblog](#)

Pingback: [links for 2008-12-12 | /dev/random](#)

Pingback: [links for 2008-12-12 « Mike's Blog](#)

Pingback: [Evolutionary misconceptions « Roger Alsing Weblog](#)

Pingback: [Evolution of Mona Lisa « Adam Kapler](#)

Pingback: [Even more totally awesome stuff « Circadian Rhythms](#)

Pingback: [Friday Links #29 | Blue Onion Software *](#)

Pingback: [calltoreason.org » My photo rendered using polygons](#)

Pingback: [piece o plastic – the revolution will be blogged » ruff linkage 200850](#)

Pingback: » [Genetic Programming: Evolution of Mona Lisa](#) « [Roger Alsing Weblog](#)
[mulei.info 老木, 一路向西](#)

Pingback: [Col's Rational World](#) » [Blog Archive](#) » [Yay! Evolving art!](#)

Pingback: [Mona lisa remake | Mundo Gump](#)

Pingback: [Besserwisser blog](#) « [My Blog](#)

Pingback: [Modulo Errors](#) » [Blog Archive](#) » [Genetic Algorithms with Processing](#)

Pingback: [This week on C9: Oxite, Mona Lisa, Pool hacks, and Coding4Fun Gifts | This Week On Channel 9 | Channel 9](#)

Pingback: [This week on C9: Oxite, Mona Lisa, Pool hacks, and Coding4Fun Gifts | Games Money](#)

Pingback: [Top Posts](#) « [WordPress.com](#)

Pingback: [All about link building.](#)

Pingback: [Matthew Lynch vs Rationality](#) » [Blog Archive](#) » [Fun with Genetic Algorithms](#)

Pingback: [blueblog » Blog Archive » genetische Evolution von Bildern](#)

Pingback: [Living Code » Blog Archive » Drawing with opacity](#)

Pingback: [Besserwisser post on EvoLisa « Dan Byström's Weblog](#)

Pingback: [Microsoft NZ MSDN Blog : MSDN Flash Christmas Special](#)

Pingback: [Nigel Parker's Outside Line : MSDN Flash Christmas Special](#)

Pingback: [This week on C9: Oxite, Mona Lisa, Pool hacks, and Coding4Fun Gifts | CHARGED's Digital Lifestyle at Work or Play](#)

Pingback: [Hugh's ramblings » Blog Archive » Genetic Vectorizer](#)

Pingback: [RSS aggregator » Blog Archive » Interesting genetic programming exercise](#)

Pingback: [Genetic Programming Example « Rare Intellect Blog](#)

Pingback: [Optimizing away « Dan Byström's Weblog](#)

Pingback: Alt Gr

Pingback: [turn that shit UP » Blog Archive » Genetic Programming: Evolution of Mona Lisa « Roger Alsing Weblog](#)

Pingback: [kierandelaney.net » Blog Archive » Building The Mona Lisa](#)

Pingback: [rhapsodhy.hu » about wealth and taste](#)

Pingback: [Free of All » Blog Archive » Genetic programming](#)

Pingback: Alt Gr

Pingback: [Tweak and Geek » Blog Archive » TG#16.5: Mona Lisa Evolution](#)

Pingback: [Interesting stuff for December 2008 « The Outer Hoard](#)

Pingback: [blog@kde» Blogarchiv » Genetische Programmierung: Evolution der Mona Lisa](#)

Pingback: [Quirky Sunday Links](#)

Pingback: [Pinceladas de evolución | la ciudadela](#)

Pingback: [Complicated Algorithm, Neat Images | Slaptijack](#)

Pingback: [The Evolution of Mona Lisa | Nature Notes from Harold Stiver](#)

Pingback: [Evolucija Mona Lise](#)

Pingback: [Evolving Genetic Algorithms in Lisp | So Jake Says](#)

Pingback: [Kangaroo sequences, cocaininated bees and stupid robots « It's Alive!!](#)

Pingback: [» Blog Archive » Evolutionary Computing](#)

Pingback: [Real-Time Rendering » Blog Archive » This, That, and the Other](#)

Pingback: [West Karana » The Evolution of Ebony](#)

Pingback: [Genetic Program Creates Image of Mona Lisa at The Tech Record](#)

Pingback: [Lousy Canuck » Python Evolution - part 1](#)

Pingback: [The World of Stuff - Blog Archive - We are the champions](#)

Pingback: [dholm.com » Blog Archive » Tumblelog: 090112](#)

Pingback: [EvoLisa: Optimizations and Improved quality « Roger Alsing Weblog](#)

Pingback: [Me made out of Genetic Programming! | Oscar and Friends](#)

Pingback: [C Hees . info » Mona Lisa with Genetic Programming](#)

Pingback: [New Adventures in Software » Practical Evolutionary Computation: An Introduction](#)

Pingback: [Brian Low » EvoLisa Video](#)

Pingback: [Hevolisa « e7ektroblog](#)

Pingback: [STNDE | Image evolution | Javascriptで画像マイニング](#)

Pingback: [The evolution of a media player « Subjects Keeping Me Awake](#)

Pingback: [Evolving vectorization – Westhoffswelt – Welcome to the real world](#)

Pingback: [oranchak.com » Blog Archive » Simulated evolution parlor tricks](#)

Pingback: [Genetic Algorithms.., A Primer « The Green Destiny](#)

Pingback: [Evolution of Mona Lisa « Fatalis To Browser](#)

Pingback: [Today's Mindboggler: 1000:1 compression ratios? « OpticalFlow](#)

Pingback: [Genetic Algorithm Part I « Eager to Code, Enjoy to Debug ~ Embark into Each Stage with Your Heart](#)

Pingback: [c.onscio.us/ly :: Grrr](#)

Pingback: [Evolutionary Compression « Roger Alsing Weblog](#)

Pingback: [LCS and GBML » Blog Archive » Evolution of Mona Lisa](#)

Pingback: [Lousy Canuck » Python Picture Evolution part 2, finally](#)

Pingback: [Russian Coding 4 Fun : Мона Лиза у меня неплохо получилась](#)

Pingback: [lbrandy.com » Blog Archive » Genetic Algorithms: Evolving Human Faces](#)

Pingback: [New Adventures in Software » Watchmaker 0.6.0 - Evolutionary Computation for Java](#)

Pingback: [{5} Setfive - Talking to the World » Blog Archive » Monkeys and shakespeare: genetic algorithms with Jenes](#)

Pingback: [Hodgepodge « Tish Tosh Tesh](#)

Pingback: [Tagz | "Genetic Programming: Evolution of Mona Lisa « Roger Alsing Weblog" | Comments](#)

Pingback: [Logic Nest · The Consequences of Trusting Computers](#)

Pingback: [mental_floss Blog » Evolving Mona Lisa](#)

Pingback: [Künstliche Intelligenz – Durch Evolution zur Mona Lisa | Das](#)

Meinungs-Blog

Pingback: [Artistic Evolution » Blog Archive » The Application of an Idea](#)

Pingback: [InVisible Blog » links for 2009-10-25](#)

Pingback: [Whole](#)

Pingback: [Masuto Studios » Blog Archive » Monalisa por un algoritmo genetico](#)

Pingback: [WinSoft.se » Making fun things from the knowledge of evolution](#)

Pingback: [Painting with Computers | Flirting With Models](#)

Pingback: [pings from the void » Evolving Geometry. Or what I did with my Sunday.](#)

Pingback: [Interesting Reading... – The Blogs at HowStuffWorks](#)

Pingback: [Tim](#)

Pingback: [La Mathematica non è un Pantone « Macworld Online](#)

Pingback: [AC » Archive » genetic programming 多边形近似](#)

Pingback: [Some applications ! : Ben's Blog](#)

Pingback: [Mona Lisa replicated in software “using only 50 semi transparent polygons”](#)

Pingback: [\[links\] Link salad’s face is at first just ghostly | jlake.com](#)

Pingback: [Friday Links: Zach Attack Edition](#)

Pingback: [PHP Hackery » Blog Archive » How To Heat Your House With PHP](#)

Pingback: [27遗传程序：蒙娜丽莎的演化 - 新鲜事 - 演化 - 蒙娜丽莎 - 米叻！](#)

Pingback: [Reproducing images with primitive shapes. \(Graphics optimization problem\) | DeveloperQuestion.com](#)

Pingback: [The Mario Genome! « Kele's Science Blog](#)

Pingback: [Mona Lisa, algoritmos genéticos y HTML5: la evolución de una sonrisa | Shft](#)

Pingback: [Genetic Algorithm picture generation | sharpsgenetic](#)

Pingback: [Nice Vector Images photos](#)

Pingback: [Dabbler : Genetic Algorithms in Perl](#)

Pingback: [Sza.lonna.hu » Időszaki szumma](#)

Pingback: [genetic painting agent | Mathisonian](#)

Pingback: [links for 2011-02-12 « Boskabout](#)

Pingback: [Genetic Algorithm Examples « puremango.co.uk](#)

Pingback: [HTML5 » Blog Archive » Mona Lisa, algoritmos genéticos y HTML5: la evolución de una sonrisa](#)

Pingback: [Marbled Rye and Evolutionary Algorithms « The Happy Technologist](#)

Pingback: [chiplynch.com » Marbled Rye and Evolutionary Algorithms](#)

Pingback: [A Arte dos Algoritmos | Teoria da Conspiração](#)

Pingback: [Genetic Programming: Evolution of Mona Lisa |](#)

Pingback: [SO COOL! | BradLeclerc.com](#)

Pingback: [Qball's Weblog » EvO – Image vectorization using evolution](#)

Pingback: [Beating PNG « Bainbridge Code](#)

Pingback: [Pinceladas de evolución | laciudadela.net](#)

Pingback: [Nature of Code Final | ITP Blog](#)

Pingback: [RESTART IT HERE | MonaLisa in Nerd style | Massimo Meijer's inspiration blog](#)

Pingback: [“进化”出来的蒙娜丽莎 « T客网 | Techpot](#)

Pingback: [Generating Faces from Random Polygons | Monkeyologist](#)

Pingback: [Reproducing images with primitive shapes. \(Graphics optimization problem\)](#)

Pingback: [Pareidoloop – algorithmic face generation | CreativeJS](#)

Pingback: [Pareidoloop :: iobound](#)

Pingback: [СЛУЧАЙНЫЕ ЛИЦА | Altsoph](#)

Pingback: [Computer Computer Interaction | Inventing Interactive](#)

Pingback: [What qualifies something as AI? | A.I. Spark](#)

Pingback: [What are some interesting and relatively simple to implement projects to learn Neural Networks? | A.I. Spark](#)

Pingback: [Should real-valued neural network inputs be scaled based on minimum/maximum values or by variance? | A.I. Spark](#)

Pingback: [Genetic Algorithms: Links and resources \(1\) « Angel "Java" Lopez on Blog](#)

Pingback: [Pareidoloop | Robot Monkeys](#)

Pingback: [How can I avoid premature convergence to a local optimum on my genetic algorithm? | A.I. Spark](#)

Pingback: » [What is Computer Science? ACM@LETU](#)

Pingback: [Genetics | Ene mene mu ...](#)

Pingback: [Class 11: Digitisation – Links « Waving at the Machines / ITP](#)

Pingback: [Thought this was cool: Genetic Programming: Evolution of Mona Lisa | Roger Alsing Weblog « CWYAlpha](#)

Pingback: [BrightestYoungThings – NYC – Rise and Shine: The Internet Told Me So ...](#)

Pingback: [BrightestYoungThings – Planet – Rise and Shine: The Internet Told Me So ...](#)

Pingback: [Guerrilla Monkey – Bookmarks](#)

Pingback: [Winter Break Project | leadiv](#)

Pingback: [BrightestYoungThings – DC – Rise and Shine: The Internet Told Me So](#)

...

Pingback: [Drain the Main Brain](#)

Pingback: [Poster 08 – Creative, Easy Charts and Graphs Using Flash and XML #hewebe10 | Elemental Cabal of Aardwolf mud](#)

Pingback: [» Quinta clase #Optimización de estructuras de hormigón](#)

Pingback: [An aid to AID: SMS image xfer service « Chewy Chunks](#)

Pingback: [Ross Churchley › Mildly Interesting | Image Evolution](#)

Pingback: [Ross Churchley › Blog › Image Evolution](#)

Pingback: [Introduction to Simulated Annealing: I](#)

Pingback: [Home Manchester | Playing with HOME data](#)

Pingback: [Genetic Algorithm | Robert Murrish](#)

Pingback: [Tile-Based Image Compression | Pointers Gone Wild](#)

Pingback: [Mona Lisa in 50 Evolutionary Polygons | Beyond the Beyond | Wired](#)

Pingback: [Image Estimation using Stochastic Hill Climbing | CCTV Computer Vision Blog](#)

Pingback: [documenting gleitzeit paul jaisini invisible](#)

Pingback: [Paul Jaisini Invisible Paintings from 1994](#)

Pingback: [Natural selection at work | Jeremy Yoder](#)

Pingback: [parse calm record into small settlement controlling matlab - Zietlow Gyan](#)

Pingback: [Boo! The optics behind “ghost” imaging | Skulls in the Stars](#)

Pingback: [How to: Twitter image encoding challenge | SevenNet](#)

Pingback: [Solution: Twitter image encoding challenge #dev #it #computers | Technical information for you](#)

Pingback: [Artificial intelligence – interesting topic | Zabbo Answers](#)

Pingback: [GLSL-Projectron | // andy hall](#)

Pingback: [Fixed Twitter image encoding challenge #dev #it #asnwer | Good Answer](#)

Pingback: Quora

Pingback: [Fix: What are good examples of genetic algorithms/genetic programming solutions? #development #solution #fix | Good Answer](#)

Pingback: [How to: What are good examples of genetic algorithms/genetic programming solutions? #answer #computers #solution | IT Info](#)

Pingback: [indico.io | Regressor Based Image Stylization](#)

Pingback: [Weekend Reading CIV : Blogcoven](#)

Pingback: [Artefact Two: Beauty | konrad maselko](#)

Pingback: [Pareidolia e identidad - Ártica1](#)

Pingback: [「遺伝的アルゴリズム」がどのような仕組みなのかが2分でわかるムービー - GIGAZINE | 4oCH.NET](#)

Pingback: [Painting with Triangles \(1\) | Incoherent Waves](#)

Pingback: [Odkłamywanie ewolucji \(2\) – Another blog in the net](#)

Pingback: [How can I tell when two cubic Bézier curves intersect? - MathHub](#)

Pingback: [Polygen – a genetic algorithm for approximating an image with polygons | 神刀安全网](#)

Pingback: [evolving mona lisa @ 2afuture.me](#)

Pingback: [« Le chien obèse » par le créateur de Simon's Cat !!](#)

Pingback: [A First Stab at Evolution | Adventures In Evolution](#)

Pingback: [An enigma in triangles | Adventures In Evolution](#)

Pingback: [Evolution Art – Industrial Assets](#)

Pingback: [Python趣味代码整合之提升学生编程兴趣 - HLike](#)

Pingback: [My First Attempt At Genetic Programming – Denis Bogdan](#)

Pingback: [非エンジニアでも直感的にわかる「遺伝的アルゴリズム」とは？ | SiTest\(サイテスト\) ブログ](#)

Pingback: [My Attempt at Generating the Mona Lisa – Soliton Science](#)

Pingback: [Algoritmi genetici: il fascino del calcolo evolutivo – Setup & Draw](#)

Pingback: [The Mona Lisa Facts - 19 Amazing Facts About The Mona Lisa | KickassFacts.com](#)

Pingback: [19 fatos surpreendentes sobre Mona Lisa | Detetive Curioso](#)

Pingback: [畳み込みニューラルネットワークの識別能力を用いて抽象画を描く - Perception Engines](#)

Pingback: [Week10: Final Project Idea – Yeonhee Lee](#)

Pingback: [The Mona Lisa – by Leonardo Da Vinci – The World of Arts](#)

Comments are closed.

Blog at WordPress.com.