Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding

Nick Cheney
Creative Machines Lab
Cornell University
Ithaca, New York
nac93@cornell.edu

Rob MacCurdy
Creative Machines Lab
Cornell University
Ithaca, New York
rbm7@cornell.edu

Hod Lipson
Creative Machines Lab
Cornell University
Ithaca, New York
hod.lipson@cornell.edu

Jeff Clune
What's your lab again???
University of Wyoming
Laramie, Wyoming
jclune@uwyo.edu

ABSTRACT

In 1994 Karl Sims showed that computational evolution can produce interesting morphologies that resemble natural organisms. Despite nearly two decades of work since, evolved morphologies are not obviously more complex or natural, and the field seems to have hit a complexity ceiling. One hypothesis for the lack of increased complexity is that most work, including Sims', evolved morphologies composed of rigid elements, such as solid cubes and cylinders, limiting the design space. A second hypothesis is that the encodings of previous work have been overly regular, not allowing complex regularities with variation. Here we test both hypotheses by evolving soft robots with multiple materials and a powerful generative encoding called compositional pattern-producing networks (CPPNs). Robots are selected for locomotion speed. We find that <insert different conclusions since this is not really true: evolution does take advantage of additional materials to produce faster, more diverse designs>. We also found that CPPNs evolve faster robots than a direct encoding control and that the CPPN morphologies appear more natural.

Categories and Subject Descriptors

ALIFE [**Artificial Life/Robotics/Evolvable Hardware**]: Robotics

General Terms

Terms

Keywords

Keywords

GECCO'13, July 6-10, 2013, Amsterdam, The Netherlands.

1. INTRODUCTION

In 1994, Karl Sims' work entitled "Evolving Virtual Creatures"[**?**] became an inspiration and a benchmark for the natural, complex morphologies that evolutionary computation is capable of prodcuing. But nearly two decades later, Sims' benchmark still holds - despite constant advances in evolutionary computation and computational resources. There any many hyoptheses as to why we have failed to outshine Sims' work.

One such hypothesis is that recent attempts have followed the inspiration set by Sims' to create morphologies from very limited number of predefined, rigid elements (such as solid blocks or cylinders). While this prodedure is able to abstractly model the segments of limbs found in natural creatures, it comes nowhere near the ability of biology to definte morphology at a cellular level and create arbitrarily complex shape and natural movement.

Another hypothesis is that the encodings of previous work also limited the design space. Direct encodings lack the regularity or evolvability necessary to reasonably achive impressive results, and overly regular indirect encodings constrict the design space - not allowing complex regularities with variation.

Here we take on both of these problems, and show that complex, natural appearing and behaving robots are achievable. We use VoxCAD, a voxel-based soft-body simulator to allow the specification of a robot's shape and material properties at a fine-grain resolution. We then use the powerful generative encoding in HyperNEAT to evolve the robot form and material makeup at each of these voxels.

We go on to demonstrate visually that the VoxCAD/HyperNEAT combination is able to provide a great diversity of forms, each with complex regularities with variation. We show that a direct encoding is unable to create forms that look as impressive or locomote as effectively. We also demonstrate the ability for this system to scale to larger sizes or higher resolutions, as well as to involve a greater diversity of materials. We show that diversity of form and behavior can also be increased (word choice?) with the use of various reward or penalty functions, suggesting that such a system also has the ability scale to create different forms for particular task requirements.

2. BACKGROUND

incorporte this into intro/background?

3. METHODS

what we used.

3.1 HyperNEAT (do we want to use that name? or call it CPPN-NEAT?)

it's awesome. people at this conference should have some background knowledge, but we still need a brief intro. brief, jeff!

3.2 VoxCAD

also very cool. rob can proably talk about it in more depth than I can, though I'm not sure (especially for this confrence) that we want to be dedicating a lot of page space to the simulator. We simulated different materials that would contract (like muscles), would remain stiff (like bone), or would be maluable (like ???, why am a drawing blank here, there must be something in the body that's soft but not muscle? organs aren't really a great analogy here...). Thus voxels (for example contractable ones) are like cells or fibers (for example muscle fibers). Voxels are not like cupcakes... unless those cupcakes were voxels.

3.2.1 MATERIALS

green = osciallotry phase contraction/expansion
light blue = passive soft material
red = counterphase contraction/expansion
dark blue = passive stiff material
In treatments spanning number of materials used, materials were added in the order above (ex: two material treatments consisted of green and light blue)

3.3 GAlib

we used it too. it's a publicly avalible, out of the box GA library from MIT in the late 90's. (Does it even deserve it's own subsection? or just a line in the direct encoidng part of results?)

4. RESULTS

we overloaded the wyo cluster with these babies, and here's what we got:

lots of figures and pretty pictures, link to videos (in footnotes?).

Show images of different kinds of locomotion for proof of diveristy? (Scootcher, Jumpers, Walkers, etc.)

4.1 Direct v. Generative Encoding

hyperneat is awesome! we used all the same stuff (same materials, same LEO genotype->phenotype encoding, same simulator, etc.) with a direct encoding, and the results looked like multi-color spaghetti and didn't really move.

Figure of side-by-side champs for direct v. generative

Also Plot of material distribution for direct v. generative?

4.2 Material Types

soft-robots are meant to be soft. we tried make them use hard stuff, but they resisted, it just wasn't the point. It may also be due to using only two phases of actuatable materials...

Figure of side-by-side champs for each material param

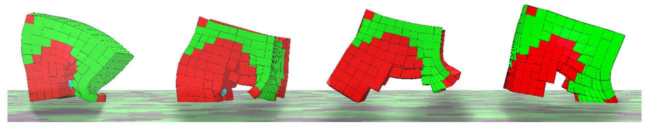
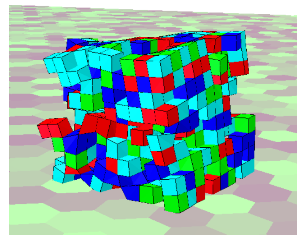


Figure 1: time series of generative encoding



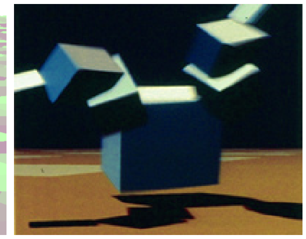


Figure 2: (left) image of direct encoding (time series not applicable)

Figure 3: (right) if we do want to make any comparisons, this is an image Sims has of one of his creatures on his own website

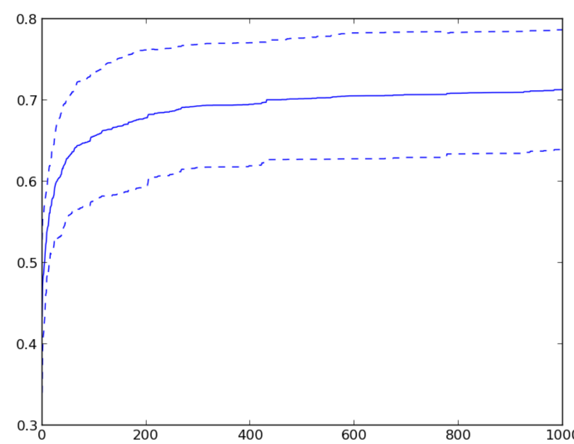


Figure 4: fitness of direct encoding. TODO: decide on final data set, plot against generative encoding, use friendly labels, proof it in black and white

Plot of material distribution for each material param

Also plot fitness? Or just say not signficantly different than all materials?

4.3 Penalty Fuctions?

give us different robot behaviors and material distributions. Can we quantatively describe behavior yet?

Figure of side-by-side champs for each penalty type

Plot of material distribution for each penalty type

Also plot fitness? Or just say not signficantly different than no penalty?

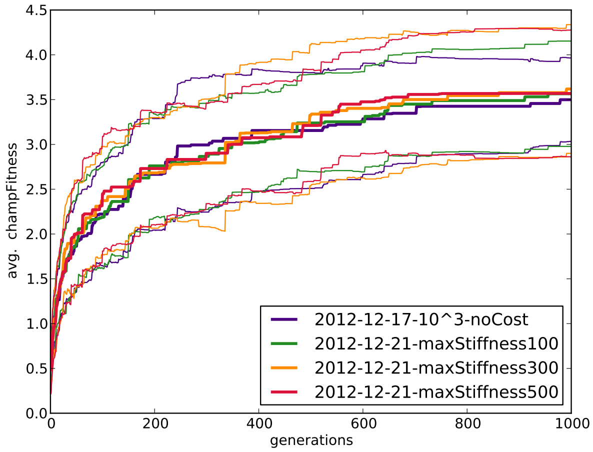


Figure 5: sweep of max stiffness (delete me?!). TODO: decide on final data set, delete?, use friendly labels, proof it in black and white

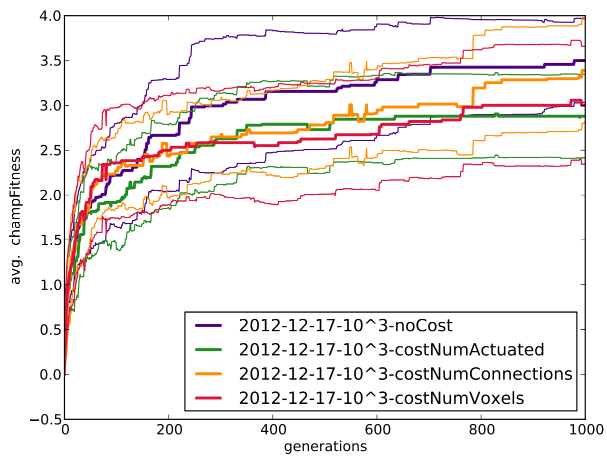


Figure 6: sweep accross penalty functions - plotting distance, not penalized fitness (delete me?!). TODO: decide on final data set, delete?, use friendly labels, proof it in black and white

4.4 SIZE

does this deserve it's own subsection? Or is it just a statment we make (not even include plots?) Or is scalability something we lean on more?

5. DISCUSSION

softbots = aweomse!

These thing are not at all trivial to design. Multiple designers - including those with and without engineering backgrounds - tried to design soft-robots by hand with the given matrials and simulator. All noted the unexptected difficulty

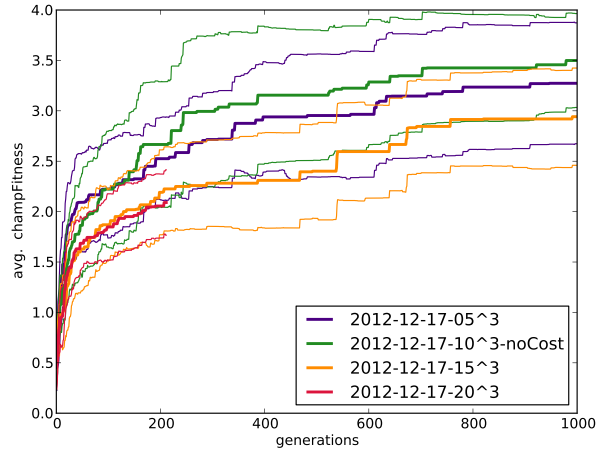


Figure 7: sweep across sizes (delete me?!). TODO: decide on final data set (with complete 20^3), delete?, use friendly labels, proof it in black and white

of such a task. None were able to design an instance that exceeded the performance of the evolutionary algorithm.

6. CONCLUSION

we are the best.

7. ACKNOWLEDGMENTS

thanks for the moneys!

8. REFERENCES