CPPN Based Evolution of Bitmap Facial Distortion

James Schneider
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816
Email: jschneider@knights.ucf.edu

Anthony Wertz
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816
Email: awertz@knights.ucf.edu

***Abstract*—Hello, my name is Picasso**

I. Introduction

It was a dark and stormy night...
Remove []

II. Background

Insert resume here

III. Approach

Not “reproach”

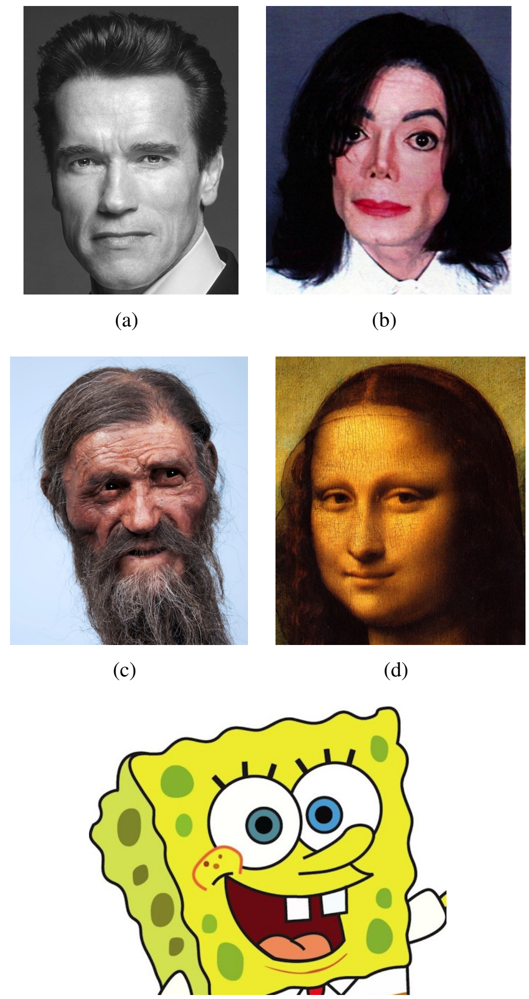
IV. Experiments

The experiments run utilized five different face images: three quite different real faces, one artist’s rendering of a face, and one unrealistic cartoon character’s. The majority of these faces are quite recognizable to an English audience, as is probably apparent in figures 1(a) through 1(e).

The diverse image set was chosen specifically to observe the effects of single distortions across a variety of faces. In many cases the same effect is achieved, though in some cases some quite diverse effects are seen (especially when dealing with the oddly proportioned cartoon figure). Of course the same mathematical function is being performed but the proportions of the face along with its structure have an impact on what the human observers interpret to have happened to the face.

Generally for each image the interactive evolutionary process is conducted by the tool operators (independently). The different faces are evolved individually to determine what effects a distortion chosen for one image has on the rest of the set. Throughout the evolution there were a few tools at the disposal of the operators. The most intuitive feature allows the operator to select the favored image or images to be evolved, which is followed by a single generation evolutionary step. This is a fairly slow heuristic compared with those used in many other evolutionary applications but it allows a tailored approach to the process.

In some instances the population might be highly homogeneous with only slight distortions between images. If this is not desirable, the cross correlation boost method can be used which will, over some user-specified number of generations, select images from the population which are not highly correlated with others in the selection. For the correlation the brightness map of each image is calculated and for each image it is determined whether it will be added one to the selection. Any given image is only added if there is a large amount of dissimilarity between it and the rest of the elements in the selection which is determined by a two dimensional correlation with each image already in the selection. This does a reasonable job of adding some additional diversity in the set.



(a) (b) (c) (d) (e)

Fig. 1. Face images used in the experimentation.

In cases where the distortions are changing quite slowly this can be helpful in acting as a spring-board for larger changes.

One pitfall of the cross-correlation approach is that it's quite slow, primarily because the output images have to be calculated for each generation. This process is sped up considerably as the evolutionary application uses scaled down versions of the original image, a process made possible by the use of a normalized image coordinate space, but the correlation and output calculation especially still take a noticeable amount of time to compute. The cross correlation method is useful to gain some diversity, but if a boost in complexity is desired the uncorrelated boost feature can be used which does not correlate any images, nor does it ever even calculate the output of the networks for all but the last set to be displayed to the operator. In this approach, members of the population are randomly assigned a fitness. The benefit here is that a large number of generations can be executed extremely which can be useful in order to gain complexity (or potentially to remove complexity, depending on how the evolutionary parameters are configured). The pitfall is that the fitness of the individuals is determined randomly so they're not always desirable. Cross correlation does a better job at adding momentum to the changes in a homogeneous set while uncorrelated boosting shuffles the phenotype space.

As alluded to, the operator is able to modify the NEAT parameters at any time to affect the evolutionary process. If more complexity is desired, for example, the node or link creation parameters may be increased. If less complexity is desired, the same parameters may be decreased along with the increase of the node and link destruction parameters. The experiments run for this project included a variety of the methods described above.

Once a favorable distortion has been evolved, the distortion is stored then applied across all images in the set for comparison.

V. Results

As mentioned in the previous section a strictly interactive evolutionary approach was not adhered to: there were a few tools at the disposal of the operators. However the initial results somewhat necessitated these changes, an observation that will be described shortly. Figures 2 to 5 show four examples of interesting distortion generated in the evolutionary process.

Figures 2 and 3 are examples of interesting distortions attained prior to the implementation of the boosting process. The first, the "big eye" distortion, creates an odd elongation of the eye. This is one of the cases mentioned previously where the results, while performing the same operation, yield different artifacts per each image. Notice on figures 2(b), 2(c), and 2(e) it would appear that each individual's left eye is elongated, though on 2(d) and 2(f) it would appear to be the right. Just given the different head geometry and orientation the essence of what distortion is taking place is fundamentally altered. This is something which could perhaps be fixed by first warping the image into a predefined coordinate space which will be discussed later.

Figures 3(b) to 3(f), victims of the "multi-eye" distortion, are very interesting for a few reasons. First, in general it's a quite odd warping that has occurred. The face of the individuals seems repeated and warped in a very fluid manner yielding nicely connected mouths and sets of eyes. It's oddly comforting to see the features flow together to form a face that could be somewhat realistic, though far-fetched, with it's flowing hairlines and its oddly shaped but not completely disjoint features. But what seems most interesting here is the warping to the cartoon character 3(f). It wasn't surprising to see the combination of multiple faces in the same weaving fashion observed in the other images. What was surprising is the seeming lack of blending and well defined features. On the lower three-eyed face, for example, it was not at all expected that the eyes would be three independent clearly defined eyes as opposed to a blur as in the other faces. This is a feature consistent across all distortions of the cartoon which seems to be primarily due to the limited palette used in the image so very little blurring actually occurs. This has two interesting applications: first, the face evolution can probably be used with great success to form new cartoon character morphs because definition is maintained and limited blurring occurs; second, a realistically proportioned cartoon of a face can be used to see with greater ease exactly what the distortion is accomplishing, which might be useful in the search for generic interesting distortions.

The last two distortions seemed fairly interesting. The issue prior to incorporating the boosting though was that complexity didn't grow very well without a large number of generations. Even with a number of generations many interesting networks had little or no hidden nodes (the previous distortion networks figures 2(a) and 3(a) had none). As long as interesting distortions come out this isn't a huge issue but the lack of complexity was yielding a constrained set of distortions that reappeared on a run to run basis. The first resort was to meddle with the complexifying evolution parameters (node and link creation) but this resulted in quite a bit of parameter "thrashing" as the parameters would need to be modified sufficiently high or low to provoke or inhibit complexification. The boosting method seemed to offer a good approach allowing rapid complexification when needed and fine tuned adjustments when it wasn't needed.

The distortions represented in figures 4 and 5 are more complex examples. By comparing their networks (figures 4(a) and 5(a)) with those of the last two examples (figures 2(a) and 3(a)) the complexity difference is obvious. It's also obvious when evaluating the distortions that arise as a result.

Figure 4, the "Gaussian V", has a quite large network for a distortion that seems fairly fluent and not terribly difficult to explain. For the real faces and the painting it pretty consistently spits out a two headed amalgam backed by a very interesting "V" shaped texture. Again, what's really interesting though is the cartoon figure where the distortion, due potentially to the lack of refined detail and the exaggerated

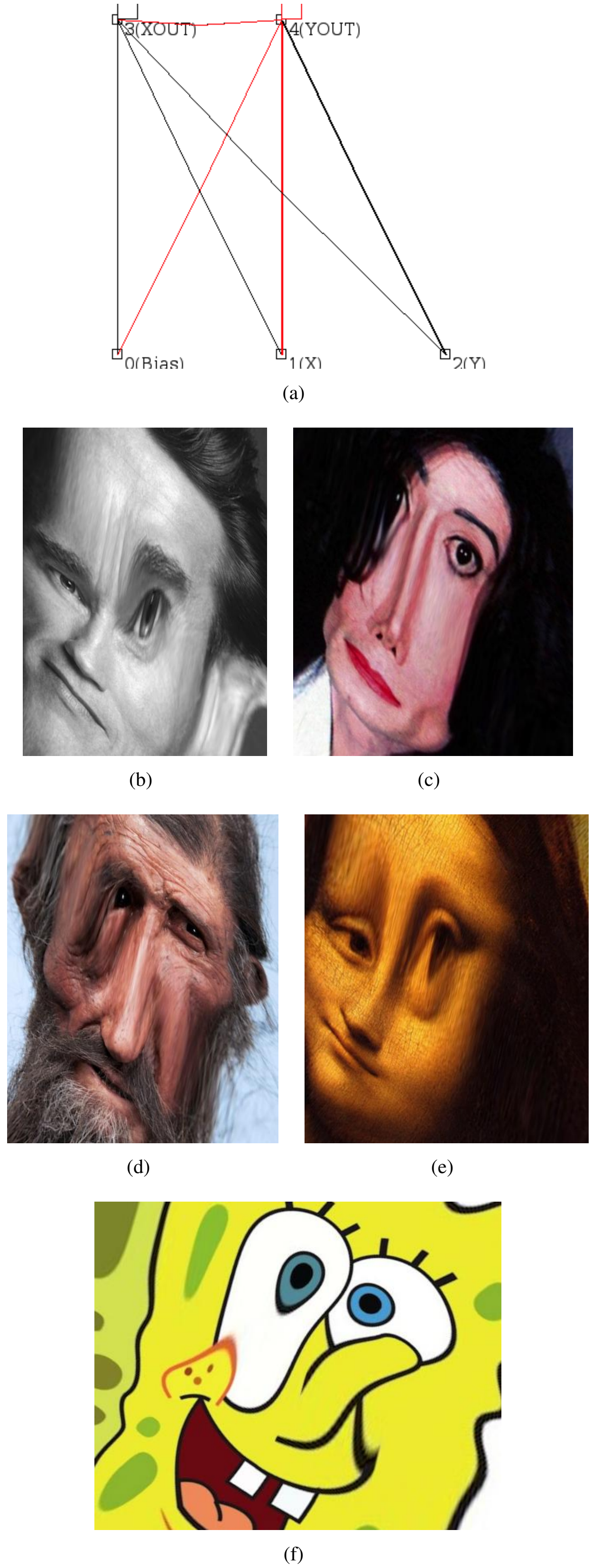
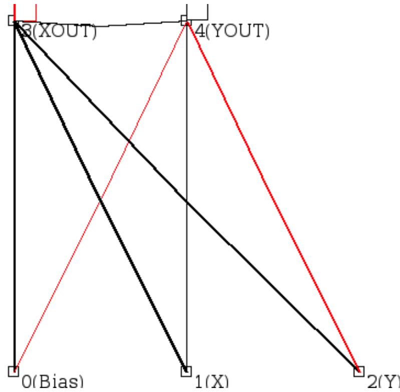
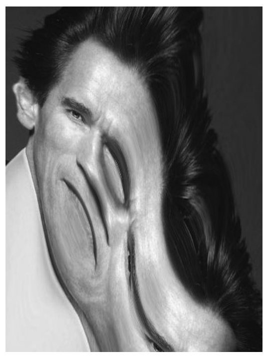


Fig. 2. Big-eye distortion: 2(a) the network and 2(b)-2(f) the distorted face set.



(a)



(b)

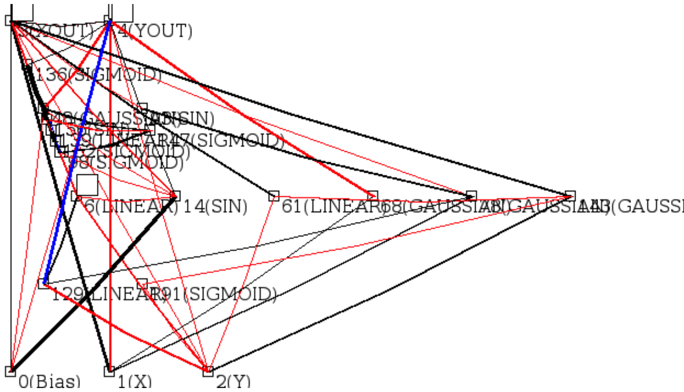
(c)

(d)

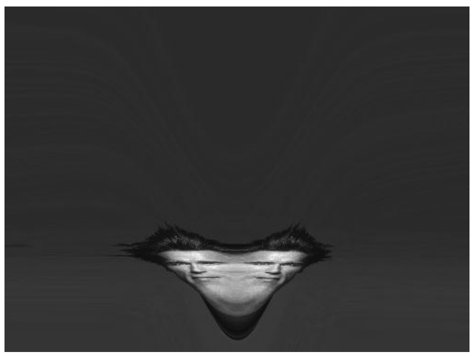
(e)

(f)

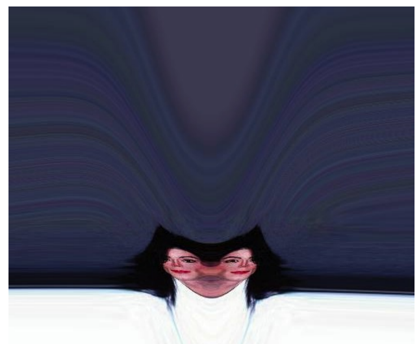
Fig. 3. Multi-eye distortion: 3(a) the network and 3(b)-3(f) the distorted face set.



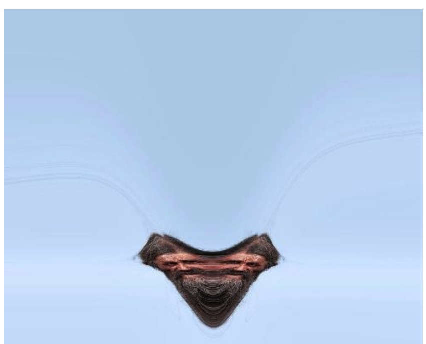
(a)



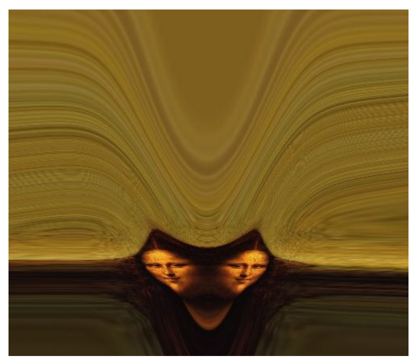
(b)



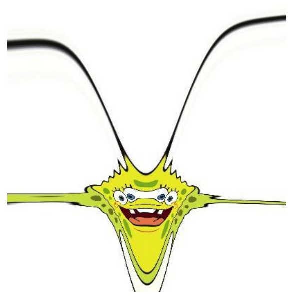
(c)



(d)

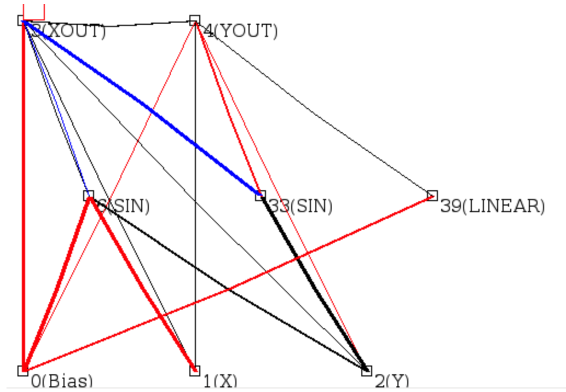


(e)

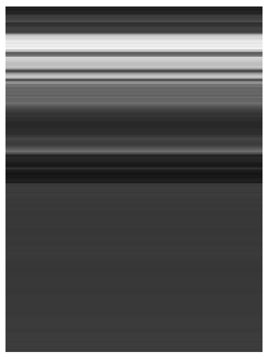


(f)

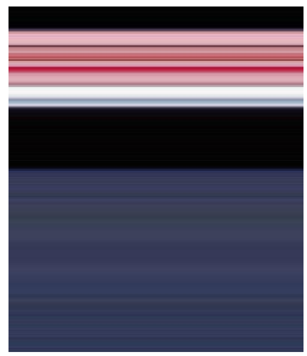
Fig. 4. Gaussian V distortion: 4(a) the network and 4(b)-4(f) the distorted face set.



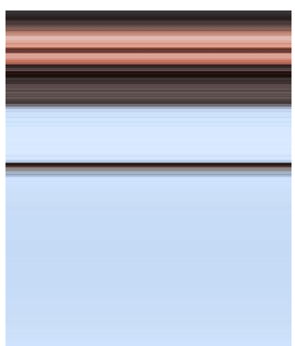
(a)



(b)

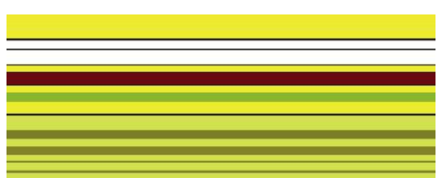


(c)



(d)

(e)



(f)

Fig. 5. Color extraction distortion: 5(a) the network and 5(b)-5(f) the distorted face set.

features, generates something more a kin to a new creature than the old one somewhat warped. The cartoon has turned in to some sort of bug or alien creature. This is a perception that's completely different than what's gathered from the same distortion applied to the other images.

The final distortion depicted in figure 5 is a fairly simple network that creates from an image of a face a linear texture using the colors extracted from the original image. Some how this network seems to have grabbed a single column from the original image, flipped the upper and lower halves, stretched one and compressed the other, then tiled it across the image. This is done with two sine nodes and one linear node. It's a pretty interesting outcome that doesn't seem at all intuitive given the network. It does serve to demonstrate the inherent difficulty in understanding CPPNs based purely on their makeup, even simple CPPNs with three hidden nodes.

VI. Discussion

The experimentation and results yielded some useful findings regarding the work and its usefulness.

First, sending in raw face images alone without any uniformity in or designation of features (such as the head dimensions, eye sizes and locations, et cetera) will not necessarily yield similar results across multiple faces. There were some examples presented showing different eyes elongated for example. Given the networks are configured to flow from the output destination to the source pixel location there is no coherent way to indicate the location of features since those are not known until the deformation is performed. One possibility to look at is reversing the flow of information so that the source locations are input and the destination locations are output. This may cause some issues with sparsely populated image generation which, depending on the distortion, may be difficult or even impossible to fix with blending and interpolation techniques. Another possibly more practical method might be to detect the features in the face image then allow warping to be done on the features individually and recombined to form a face. This has the disadvantage of compartmentalizing individual components though.

Second, interesting results can be obtained with very little complexity but the results tend to converge to similar designs. Even a few hidden nodes can greatly increase the diversity in the results. But to get past this initial no diversity hump the boosting effort seemed fairly useful. Another option might have been to simply start the network off with a few hidden nodes but that approach requires some assumptions in determining what might compose a desirable network, a task that's fairly complicated even with few nodes. For example, is figure 5 desirable? One may suppose it depends largely on the application; for general face deformations it's probably not very interesting. It's difficult to deduce beforehand that two sine nodes and one linear won't make something interesting and it's difficult to deduce that they won't ever create something interesting. The other aspect of boosting that is of interest is the part played by the cross correlation. It seemed to perform well at discriminating close to duplicates but they had to be very close for it to be useful. This may very well not be the best similarity indication available. Even if it were there would need to be large speed improvements made before it could be considered practical. This may very well come out of simply moving it away from Python into C or C++.

Third, the technique as it is currently implemented may have limited practical use on real images or detailed drawings but it seems to have some clear applications in less detailed use cases. The cartoons generated, for instance, show some potential of the system to evolve cartoons and, if expanded into a three dimensional space, perhaps also structures. This could be an interesting addition if trying to generate unique user-generated data (in games, for example).

VII. Future Work

Curing cancer, world domination, etc

VIII. Conclusion

Fin.

References

[1] N. Markey. (2005, October) Tame the beast — the b to x of bibtex. [Online]. Available: http://tug.ctan.org/tex-archive/info/bibtex/tamethebeast/

[2] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *Poop*, vol. 20, pp. 569–571, November 1999.

[3] B. D. Cullity, *Introduction to Magnetic Materials*. Reading, MA: Addison-Wesley, 1972.

[4] A. Castaldini, A. Cavallini, B. Fraboni, P. Fernandez, and J. Piqueras, "Midgap traps related to compensation processes in CdTe alloys," *Phys. Rev. B.*, vol. 56, no. 23, pp. 14 897–14 900, 1997.