A Machine Learning application based on Giorgio Morandi Still Life Paintings to Assist Artists in the Choice of 3D Compositions

Guido Salimbeni , Frederic Fol Leymarie , William Latham.
Department of Computing, Goldsmiths, University of London, U.K.
Contacts: <g.salimbeni,ffl,w.latham@gold.ac.uk>

Abstract

The authors present a system built to generate arrangements of three-dimensional models for aesthetic evaluation, with the aim being to support an artist in their creative process. The authors explore how this system can automatically generate aesthetically pleasing content for use in the media and design industry, based on standards originally developed in master artworks. They then demonstrate the effectiveness of their process in the context of paintings using a collection of images inspired by the work of the artist Giorgio Morandi (Bologna, 1890–1964). Finally, they compare the results of their system with the results of a well-known Generative Adversarial Network (GAN).

1 Introduction

The position of the visual elements in an image to create a coherent and visually balanced whole is one of the aspects of the success of a painting or design [1]. Still-life paintings are a common means for artists to explore the potential of compositions and their impact on the viewer (Fig. 1). Unlike portraits or landscapes, where the preliminary work is done with sketches, use of still-life subjects allows artists to conduct an in-depth investigation of several potential compositions, by physically arranging all the elements and exploring several alternative solutions before starting the actual execution of the painting. Finding the optimal arrangement of visual elements for a painting is a manual and time-consuming activity that the painter pursues by applying general principles of design and personal aesthetic intuition. Still-life painters arrange the elements according to principles of perception applied to art including balance, contrast, emphasis, movement, rhythm and unity to create an aesthetically pleasing pictorial solution [2].

Paul Cezanne (Aix-en-Provence, 1839–1906) deliberately rearranged and tilted forward objects such as bowls or cups in his still-life scenes to give the composition the look that he wanted. By moving objects forward, Cezanne's "tilted" still-life paintings led ultimately to Cubism [3]. Giorgio Morandi (Bologna, 1890–1964), known in particular for his still-life painting compositions, would painstakingly place simple objects such as ceramic pots and jugs in different arrangements until he found one that pleased him that he would then paint from.

In our work, we have focused on core principles of design and used computational methods in a three-dimensional (3D) digital environment to assist the artist in the choice of a composition of high aesthetic value. By leveraging the features of a 3D game engine framework, our system can both analyze the rendered image on the screen and obtain information on the position, depth and volume of the objects in the 3D digital environment.

In this article, we demonstrate the effectiveness of our method by undertaking a specific task that involves the artistic choice of pictorial compositions. Our system uses a 3D digital prototype of the

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

objects to paint or render in design and provides a computational technique to assist the painter or the designer in the selection of the most aesthetically pleasing composition for the work. In this communication, we describe how our system uses an evolutionary algorithm (EA) and a combination of four artificial neural networks (NNs) to greatly simplify and speed up the usual bottleneck that is the choice of composition. Finally, we compare the results of our system with those produced by a GAN [4] currently used in commercial applications.

2 Related Work

During the past decade, architects and engineers have started to use digital generative design tools to optimize the creative and production processes. Previously, designers would perform most of their preliminary exploratory work with sketches [5], using a CAD (computer-aided design) application in the later stages of the design process. However, recent innovations from the field of artificial intelligence (AI), when applied within 3D software platforms, have allowed designers to experiment with alternative design solutions even in the early creative process stage. This can be considered the emergence of a new collaborative approach to design between the computer and the artist [6–8].

In the field of architecture, digital generative design is now allowing architects to explore thousands of design possibilities using tools integrated into CAD software [9]. For example, Autodesk's Fusion 360 Generative Design is one of the most recent software applications of such new design processes. This type of exploratory design process offers the potential to more rapidly produce innovative architectural projects that are feasible within constrained budgets and with available resources [10].

In other fields, many companies and research groups are experimenting with 3D-printing technology combined with generative design and AI. Different AI technologies have been applied to simplify the creative process, with significant and promising results [11–13]. For example, MX3D permits great reductions in cost, weight and production time for 3D printing design projects. DreamSketch, a 3D design application [14], has shown success in providing a procedural design approach that optimizes designs under functional requirements. Kowaliw et al. [15] have developed the EvoEco system, mimicking evolutionary principles, which automatically detects and improves creative designs.

Artists and researchers are also exploring the potential of joint generative design and AI technologies to support the creative process. Joanne Hastie uses a machine learning technique to generate new compositions for potential abstract paintings [16]. David Ha developed an art project where an NN draws sketches in a human-like manner [17]. Some research is focused on the design of fitness functions that can emulate human aesthetic preference by using machine learning technologies such as NN [18] and co-evolutionary algorithms [19,20].

Our work builds on these earlier efforts and is the first, to our knowledge, to combine a 3D model manipulation inside a game engine framework together with a novel machine learning architecture to evaluate the fitness function. Our system is also the first to apply AI in the form of a deep learning architecture to assist the artist or designer with the selection and arrangement of 3D models for still-life paintings and designs.

3 Description of Generative System

Our system uses a genetic algorithm (GA) to generate a series of compositions of 3D models within Unity3D, a popular and sophisticated game engine, together with an automatic score to select the most visually balanced composition. The GA simulates an environment in which members of the population are selected for reproduction based on their fitness value in order to produce a subsequent generation [21]. In our system, the fitness value is the output of an architecture built from four NNs trained with a database of numerical 3D model data and images, each associated with an aesthetic judgment previously assigned by the artist/designer/user. The database consists of 300 sets of 3D models arranged on a table surface imitating the practice of Giorgio Morandi (Fig. 1).

We have manually placed each of the 300 series of 3D models based on their position in the original paintings. Through a digital process of crossover, mutation and selection based on the aesthetic value produced by the system independently of the user's intervention, the GA generates compositions with a consistently better aesthetic evaluation and, finally, proposes its best composition layout (the one with the highest score) for painting (Fig. 2).

Figure 1: Still life (vases and bottles) - Artist: Giorgio Morandi, Year 1948, Ca' Pesaro - Galleria Internazionale d'Arte Moderna.

The NN architecture is composed of four separate networks. The first is a fully connected NN (or FCN) that processes the front and top views of the 3D scene and assigns a positive or negative value to the composition by considering the volume and position of the 3D objects. The second is a convolutional neural network (CNN) that uses the perspective main camera as input and calculates the similarity of the current composition in Unity3D using the images of the initial database and their assigned composition value. The third is another FCN that takes the data directly from Unity3D and checks the symmetry of the 3D scene, any intersection of 3D models, the isolation of the objects, and how much screen space they cover.

These three networks each individually evaluate a specific aspect of the composition and then merge their outputs into a fourth NN that produces the final composition value, normalized between 0 and 1, which can be fed into the GA. Figure 5 shows the functional diagram of the entire system. We used 270 labeled images of the database to train the neural networks and evaluated the models' performances on the remaining 30 images. We repeated the process 10 times, each time selecting a different range of 270 images for training and 30 for testing. This cross-validation testing reported a mean accuracy of over 95% of the output.

4 Discussion

Our system represents a different approach to the generation of images that can be used by a painter to expand and expedite the creative decision of a pictorial composition. In Fig. 3 we present some images generated by our system, sorted according to their fitness value, and in Fig. 4 we show the outputs from a Generative Adversarial Network (DCGAN) [22]. In recent years, several generative model architectures have been designed for various applications. For our purpose, we used a DCGAN that mainly consists of convolutional layers and is particularly good at extracting useful features from images [23].

Our application can generate new compositions in which the 3D objects do not intersect because the system is aware of the volume and position of the models in the 3D space. The same ability is difficult to verify with the DCGAN. Moreover, the ability of our system to position the 3D model at a different point along the perspective axis is more evident than with DCGAN. The variety of composition solutions proposed by our system also includes the rotation of objects along their vertical axes. A DCGAN working only with 2D images cannot recognize the 3D attributes of the objects and so is not able to rotate them to produce greater variety in composition solutions.

5 Conclusion

Our system can create new compositions with 3D models not included in the original database since the criterion for aesthetic evaluation is less dependent on the object's type and shape than with DCGAN. However, the DCGAN can generate new images quickly, while the current state of our system requires the generation of several potential solutions before selection.

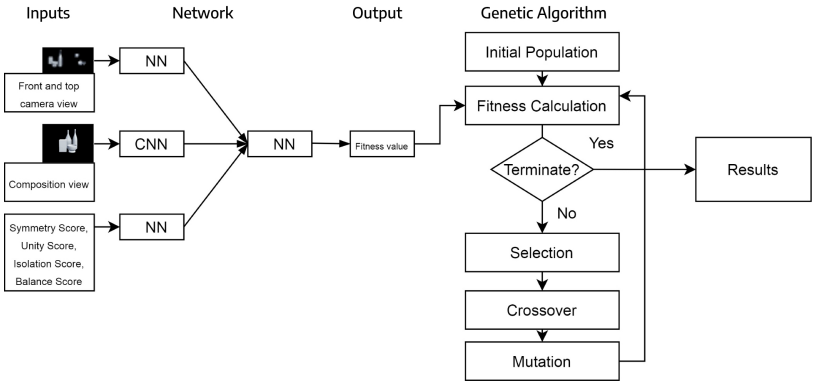

Figure 2: Architecture

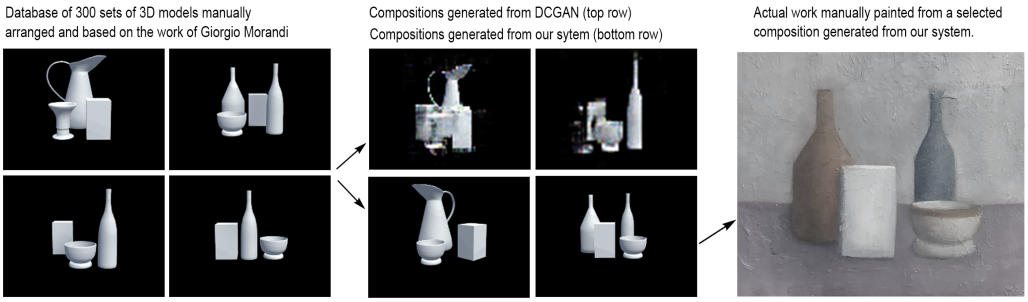

Figure 3: From an initial database to a final 3D composition and painting.

Finally, we quantified the differences between the results given by the DCGAN and the proposed system using the CNN of our architecture. The mean predicted value of the CNN for 100 images produced by the system was 95%, whereas the same test for 100 images produced by the DCGAN returned a mean value of 28%. This indicates that the compositions generated by our system are in general more similar in style to the samples in the database than the ones generated by the DCGAN.

Our experiments show that our system is able to suggest to the painter aesthetically pleasing pictorial solutions for still-life paintings generated with computational constraints that take into consideration general principles of design and aesthetic artistic intuition. This system currently works using a GA as the optimization filter, which usually requires the user to wait until all iterations conclude. Another approach, which we plan to explore in the future, would be to add reinforcement learning solutions to the system to speed up the selection process. Such an augmented system would also be applicable to other fields, including design, game content generation and architecture.

6 References and Notes

1. Rudolf Arnheim. Art and visual perception: A psychology of the creative eye. (Univ of California Press, 1965).
2. See Arnheim [1].
3. Erle Loran and Paul Cézanne. Cézanne's composition: analysis of his form with diagrams and photographs of his motifs. (Univ of California Press, 2006).
4. Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. (arXiv preprint arXiv:1701.00160, 2016).
5. Alexandre Menezes and Bryan Lawson. How designers perceive sketches. (Design Studies, 27(5):571–585, 2006).
6. Sivam Krish. A practical generative design method. Computer-Aided Design, 43(1):88–100, 2011.
7. Gabriela Goldschmidt. The dialectics of sketching. (Creativity research journal, 4(2):123–143, 1991).

8. Han Jong Jun and JS Gero. Emergence of shape semantics of architectural shapes. (Environment and Planning B: Planning and Design, 25(4):577–600, 1998).

9. Ipek Dino. Creative design exploration by parametric generative systems in architecture. (METU Journal of Faculty of Architecture, 29(1):207–224, 2012).

10. Ronny Kühne, Markus Feldmann, Sandro Citarelli, Uwe Reisgen, Rahul Sharma, and Lukas Oster. 3d printing in steel construction with the automated wire arc additive manufacturing. (ce/papers, 3(3-4):577– 583, 2019).

11. Carlos M Fernandes, Antonio M Mora, Juan J Merelo, and Agostinho C Rosa. Kants: A stigmergic ant algorithm for cluster analysis and swarm art. (IEEE transactions on cybernetics, 44(6):843–856, 2013).

12. Chien-Hung Liu and Chuan-Kang Ting. Evolutionary composition using music theory and charts. (In 2013 IEEE Symposium on Computational Intelligence for Creativity and Affective Computing (CICAC), pages 63–70. IEEE, 2013).

13. Mario García-Valdez, Juan J Merelo, Leonardo Trujillo, Francisco Fernández-de Vega, José C Romero, and Alejandra Mancilla. Evospace-i: a framework for interactive evolutionary algorithms. (In Proceedings of the 15th annual conference companion on Genetic and evolutionary computation, pages 1301–1308. ACM, 2013).

14. Rubaiat Habib Kazi, Tovi Grossman, Hyunmin Cheong, Ali Hashemi, and George W Fitzmaurice. Dreams- ketch: Early stage 3d design explorations with sketching and generative design. (In UIST, volume 14, pages 401–414, 2017).

15. Taras Kowaliw, Alan Dorin, and Jon McCormack. Promoting creative design in interactive evolutionary computation. (IEEE transactions on evolutionary computation, 16(4):523, 2012).

16. Luba Elliott. Art, intelligence and creativity. (The RUSI Journal, 164(5-6):110–119, 2019).

17. David Ha and Douglas Eck. A neural representation of sketch drawings. (arXiv preprint arXiv:1704.03477, 2017).

18. Shumeet Baluja, Dean Pomerleau, and Todd Jochem. Towards automated artificial evolution for computer-generated images. (Connection Science, 6(2-3):325–354, 1994).

19. Gary R Greenfield. Co-evolutionary methods in evolutionary art. (In the Art of Artificial Evolution, pages 357–380. Springer, 2008).

20. Thomas E Cook. Gauguin: generating art using genetic algorithms and user input naturally. (In Proceedings of the 9th annual conference companion on Genetic and evolutionary computation, pages 2647–2650. ACM, 2007).

21. John R Koza. Genetic programming. 1997. (Proceedings of the Second Annual Conference, pages 431–438, Stanford University, CA, USA, 13-16 July 1997).

22. Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. (arXiv preprint arXiv:1511.06434, 2015).

23. Foster, D., 2019. Generative deep learning: teaching machines to paint, write, compose, and play. O'Reilly Media.

24. Jeff Phoenix. Python for Machine Learning: The Art of Machine Learning & Data Science - 2019 - ASINS

Appendix A – Technical details

This deep learning architecture (Fig. 5) was implemented using the Python programming language, under the Keras machine learning library together with the TensorFlow backend. We implemented the code in Python, which is well suited for machine learning applications due to its versatility and large amount of available libraries and coded solutions already implemented. After training, the NNs are converted into bytes formats (which store the weights and architecture of the trained models) that Unity3D can use to make predictions. The system uses a plug-in called TensorFlowSharp to facilitate the integration of machine learning models developed in TensorFlow with the Unity3D API written in C-Sharp. Each of the four NNs is compiled using the categorical cross-entropy loss function and produces an output value in the range from 0 to 1.

The first NN (from the top left in Fig. 5) takes as input the combined images of the front and top camera views of the scene, as obtained from Unity3D. It is composed of three fully connected layers with 128 neurons each and a final fully connected layer with two neurons and a Softmax activation function.

The second NN (from the top left) takes as input the image from the main camera view of the scene from Unity3D. It is a CNN composed of one convolutional layer with 32 units, followed by three convolutional layers of 64 units and three MaxPooling layers, a fully connected layer with 64 neurons and a final fully connected layer with two neurons and a Softmax activation function.

The third NN (from the top left) takes as input four scores, generated directly from Unity3D, that account for symmetry, visual unity, visual isolation and visual balance. This NN is composed of two fully connected layers with 32 neurons and 16 neurons each and a final fully connected layer with two neurons and a Softmax activation function.

These three networks join into a final fourth NN that takes their outputs as inputs. It is composed of two fully connected layers with 32 neurons and 16 neurons each and a final fully connected layer with two neurons and a Softmax activation function.

Each of the four NNs has been trained independently but share the same target variable.

Figure 4: Samples of compositions generated from our system

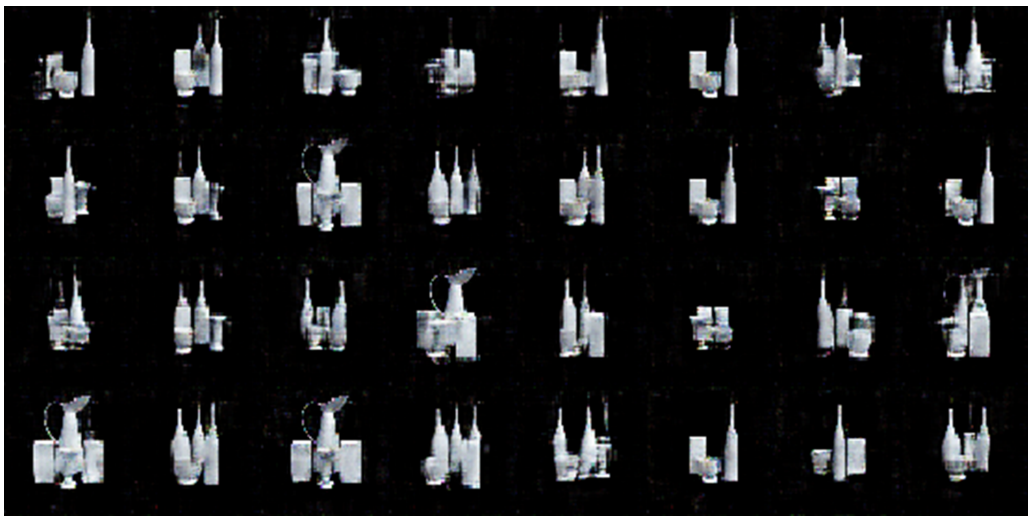

Figure 5: Samples of compositions generated from DCGAN