Evolving Three Dimension (3D) Abstract Art: Fitting Concepts by Language

Anonymous Author(s)

Abstract

In computational creativity, digital arts, most in two dimension (2D) canvas, has been a dominating topic where evolution algorithms have recently shown great quality and efficiency. However, going from 2D to three dimension (3D) where computational approaches empowers creation of 3D art remains an open field. In this work, we extend the boundary of evolutionary algorithms to computational creativity in the spatial 3D art, by bridge evolution strategies (ES) and 3D rendering through customizable parameterization of scenes. We demonstrate that our approach is capable of placing semi-transparent triangles in 3D scenes that, when viewing from specific angles, renders into films that looks like what humans understand natural language. The flexibility to customize scenes allows a new way for the artist to express creativity ideas. The supplementary material for all figures can be found anonymously here: `https://drive.google.com/drive/folders/12ytrQsckoop_fgVcQ8euFDRf2R9YOkBz?usp=sharing`.

Introduction

In painting art, a trend of abstract art, in the course of modernism (Kuiper 2021) which focuses on abstract elements instead of traditional painting, has been influential since the beginning of 20-th century. Starting from Cubism art movement (Rewald 2014) and geometric abstraction (Dabrowski 2004), a trend of focus on abstraction leads to abstract expressionism (Paul 2004) and minimalist art (Modern 2018; Bertoni 2002). Although there are drastic differences among them, they have opened a new approach of painting art where the subjective appreciation of the object or the feeling could be expressed, such that once dominant traditional focus on representation is not the only standard anymore.

Like any instruments, computers, in the broader sense of computationally generated art, have also participated in this direction of abstract and minimalist art. Early works (Malkevitchn 2003; Verostko 1994) brought forward the concept of artists generating art by designing mathematically, or more precisely algorithmically, and emphasized the property of the said algorithm, such as its complexity (Kolmogorov 1965), is an important and intrinsic metric of the art (Schmidhuber 1997). More recently with the availability of genetic

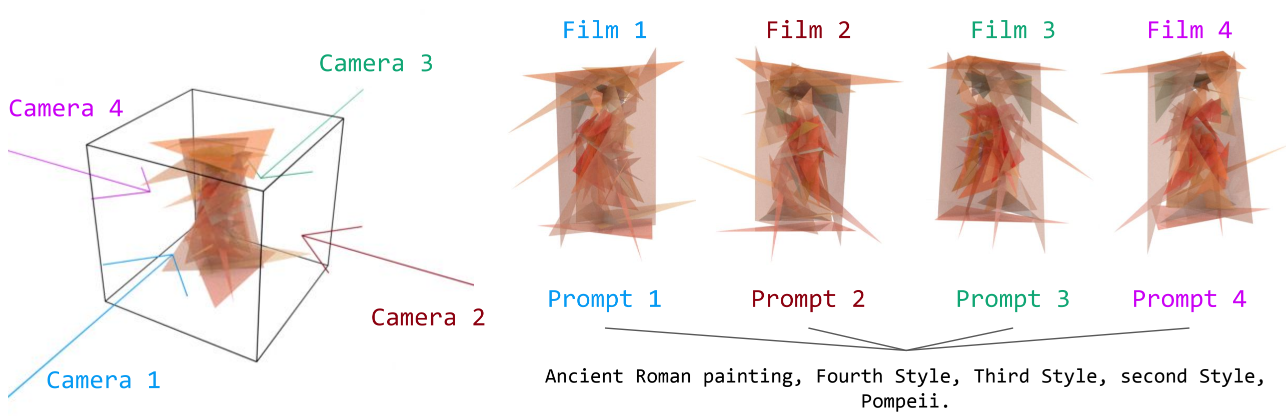


Figure 1: Our proposed method places semi-transparent triangles in three dimension (3D) spaces using Evolution Strategies (Tang, Tian, and Ha 2022; Hansen 2000; 2006). Leveraging ray-tracing based rendering Mitsuba 3 (Nimier-David et al. 2019; Jakob et al. 2022), the rendered film at possibly multiple cameras is compared with its corresponding, user-specified text prompt using distance between their representation embedded by CLIP (Radford et al. 2021). Such distances, aggregate by average, are used as the fitness in sense of Evolution Strategies, which optimize the parameters of triangles to archive better finesses.

algorithms, it is proposed that artists, instead of designing algorithmically, determine the rules that control how Evolution Strategies (ES) find such an algorithm which in turn generates the art. The computational community has thus explored the ES-driven art generation that ranges from simple (Johansson 2008; Alteredqualia 2008) to complex (Fogleman 2016; Cason 2016; Paauw and Van den Berg 2019; Shahrabi 2020; Tian and Ha 2022) art forms.

While painting has always been one of the most dominant art forms, arts concerning 3D objects is an equally important field that is essentially different from paintings. Arguably, among these arts the one with the longest tradition is sculpture (Rogers 2020) and architecture (Gowans et al. 2022) which starts from classical antiquity and remains pretty relevant today. Yet modern techniques and industries add new elements along this direction. Most notable are the advances in 3D computer graphics (Hemmendinger 2014) that simultaneously empower and are driven by electronic game (Lowood 2021) and computer animation (Britannica 2021). Like painting arts, since the 20-th century the trend of modernism leads to the sculpture to go beyond the realm of solid, representational form, and the artists started to produced "nonfunctional, nonrepresentational, three-dimensional works of art" (Rogers 2020). This particularly to spatial sculpture (Conroy 1977; Kricke 1976; Caro 1962) where space becomes the subject of the 3D artworks, and the viewing angle as well as the relation of objects comes to be an important part of the art.

As the computationally generated art mentioned above, there are also several explorations in the direction of 3D object generation. For 3D abstract, *spatial* art using genetic algorithm, early work explores rule based generation (Broughton, Tan, and Coates 1997; Coates, Broughton, and Jackson 1999) where the combination of rules are optimized, or the evaluation is done by human-in-the-loop (Cook 2007) and late ones focuses on parameterization of a single formulae (Chu 2021). A recent work (Hsiao, Huang, and Chu 2018) produces wire art that looks like predefined sketches by connecting vortex using 3D path finding algorithm. As for the border field of producing concrete, *volumetric* 3D object, the recent, high-quality advances in 3D generative model like NeRF (Mildenhall et al. 2021; Martin-Brualla et al. 2021) and text-to-image model like DALLE (Ramesh et al. 2022), Imagen (Saharia et al. 2022) and Parti (Yu et al. 2022) open the door to text-to-3D object, like DreamFields (Jain et al. 2022) DrameFusion (Poole et al. 2022) and Imagen Video (Ho et al. 2022), which are capably for generating high-quality video of concrete, *volumetric* objects.

The field of 3D art generation, with explorations mentioned above, remains an open one, in which we identify a missing gap we want to fill. Particularly, our goal focuses on computationally creating *spatial* 3D art as motivated by the trend of abstract art in modern sculpture. On the other side, unlike the early works where the artist needs to specify all the details, we would like to have a computational approach to enable autonomous optimization that makes the 3D art look like what human can easily integrate and input, which is the text, as shown by the success of recently text-to-image and text-to-video advances.

To bridge this gap, We propose to bridge evolution strategies (ES) and 3D rendering through customized parameterization of scenes. In doing so we leverage the recent advances in evolution algorithm applied to abstract art generation, as well as ray-tracing rendering, which is vital to the rendering of physically-sounding transparent objects. Two components are bridged by immediate mode, a paradigm in computer graphics where senses are parameterized. As such parameterization could be specified by the computation artist, such a flexibility to customize scenes allows a new way for the artist to express creative ideas at a high level. We demonstrate that our approach is capable of placing shapes in 3D scenes that, when viewing from specific angles, renders into a canvas that looks like images, or can be aligned with possibly multiple human understanding of natural language. This is powered by recent advances in deep learning, named CLIP which is the same used in DALLE (Ramesh et al. 2022), that is a power model connecting text and images. In doing so, the artist can freely express the idea by text, which is a more approachable way, and optionally, allowing a wider audience to participate in the 3d art creativity. A quick summary of our proposed method and some exemplary artifacts are shown in Figure 1 and Figure 2.

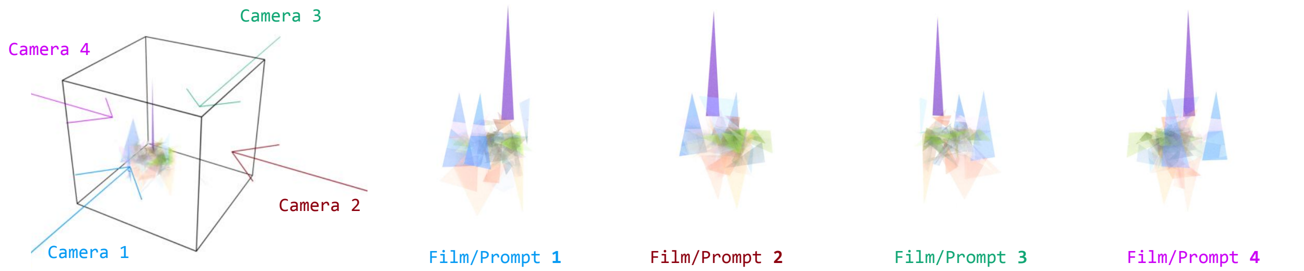
The rest of the paper is organized as fellows: we cover related works in Section , our method in Section , the experiments and artifacts in Section , and we finally conclude in Section .

Related Works

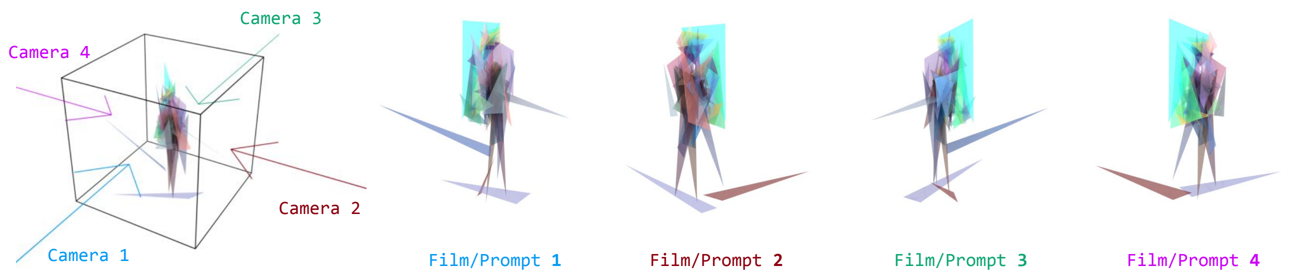
In this section we cover works that are background or related to our proposed methods.

Evolution Strategies (ES) (Beyer 2001; Beyer and Schwefel 2002), as an optimization method, has been applied to many problems. Inspired by biological evolution, it bears the similar idea of changing parameters and keeping the sets of parameters that are most fitting, in a serial and iterative way, so that at the end of evolution the best, or the most fitting solutions are left. A straightforward realization of this idea would be keeping randomly perturbing parameters and keeping ones only if the change leads to better fitness, which is unfortunately not preferred. Recent advances in ES have largely improved the performance. For example, PGPE (Sehnke et al. 2010) propose to estimate the gradients in linear time which can be used by gradient-based optimizers like Adam (Kingma and Ba 2014) and ClipUp (Toklu, Liskowski, and Srivastava 2020), and CMA-ES (Hansen 2000; 2006) estimate the covariance matrix of parameters, which provides better performance at the cost of quadratic running time.

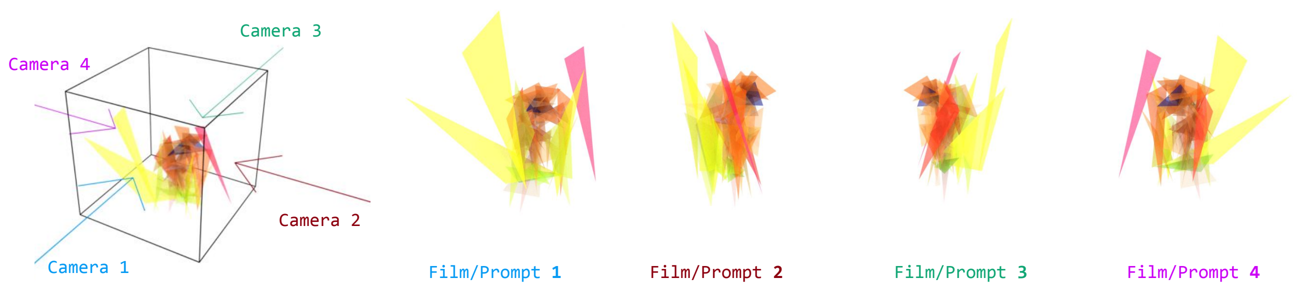
Notably, unlike gradient-based optimization, evolution strategies do not require the optimized problem to be differentiable, thus serving the role of black-box optimization solver where only the evaluation of a problem is feasible or needed. That said, its application is wide. Recent advances in neural evolution (Such et al. 2017) allows efficient optimization of neural networks, and EvoJAX (Tang, Tian, and Ha 2022) fully leverages the hardware acceleration for a wide range of evolution tasks.



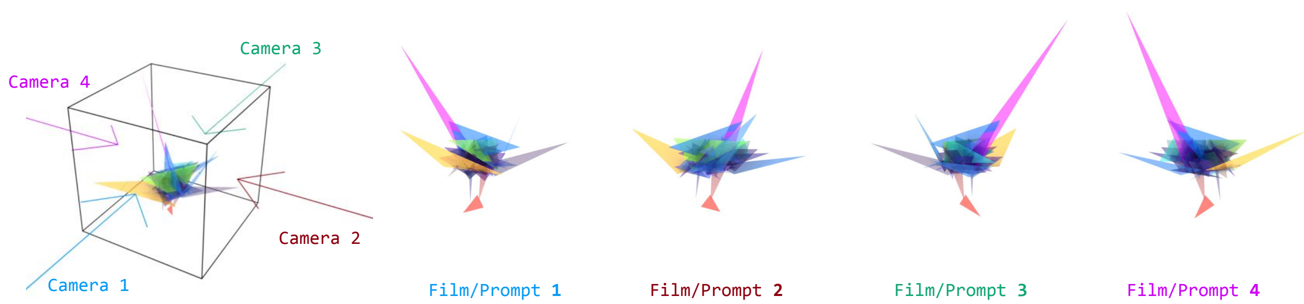
(a) The prompt for all camera/films is “Walt Disney World”



(b) The prompt for all camera/films is “A painting of Human”



(c) The prompt for all camera/films is “A bright, vibrant, dynamic, spirited, vivid painting of a dog.”



(d) The prompt for all camera/films is “A vivid, colorful bird”

Figure 2: Several examples of the evolved 3D art produced by our method, where the evolution process places triangles inside the unit cube space visualized by black frame and sets triangles’ colors and transparencies, forming a spatial configuration. In each example shown here, four cameras look at the space from four sides, although this is an arbitrary decision and cameras can have different numbers and directions. The film from each camera, capturing the rendered images, is compared with the prompt. It could be observed that our method is capable of making a 3D art, which follows the spatial abstract art style, that looks like what humans can compose in natural language text.

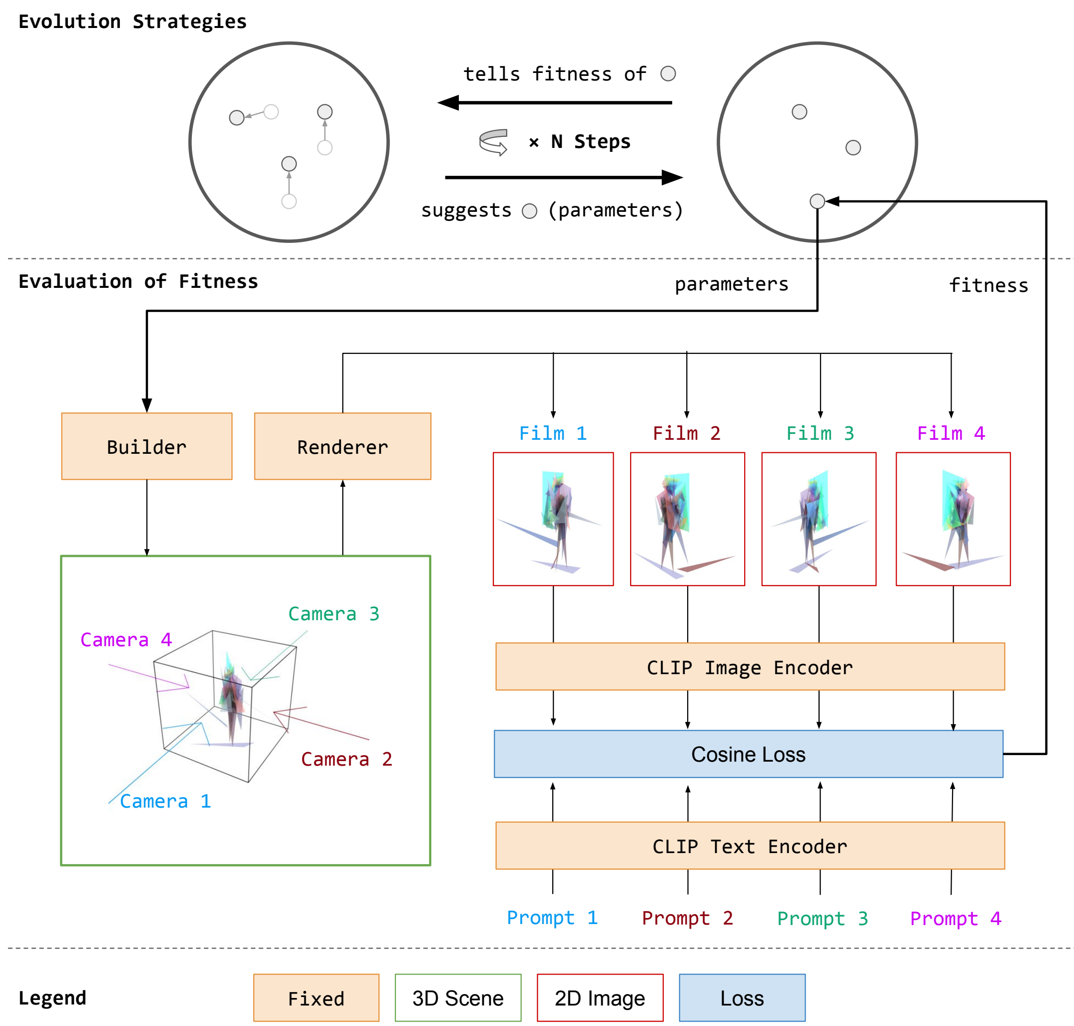


Figure 3: The architecture of our proposed method, which from parameters builds spatial 3D objects that compose semi-transparent triangles on the 3D space. The renderer renders the 3D space from different cameras producing corresponding images or "films" which are compared with provided text prompts, measured by the Cosine loss between the images and text prompts encoded by CLIP encoders. Such a loss is treated as the fitness, or the score, of the parameters, which are used by the evolution strategies in the optimization loop, where parameters leading to better fitness are found. The user of our proposed method specifies the text prompt and hyperparameters governing the behavior of the builder and the renderer, allowing themselves to express creativity in the whole process.

Computational Approach to Abstract Painting Art The computational approach to abstract and minimalist, painting art has been with a long history Even before the era of computing, early works discuss mathematical art (Malkevitchn 2003) where the connection between properties of art works and mathematics such as symmetry, polygon and octave in music, have been established. Since the inception of computers as a new means for human activity, algorithmic art (Verostko 1994) has been proposed a new framing of art, where artworks are not by human directly but by human designing a process, or an algorithm, that produces the artifact. Besides this new framing, the properties of the said algorithm themselves could also be a subject of artistic discussion, like argued in low-complexity art (Schmidhuber 1997) where the complexity of the said algorithm becomes a measure of the artwork. With the recent availability of genetic strategies, a new level of abstract is provided: artists now, instead of directly designing an algorithm process, could use the rules that controls the possible set of algorithm processes.

To find one algorithm process within the domain that produces the as good as possible artwork, the computational community has thus explore leveraging Evolution Strategies (ES) in art generation, due to the fact that ES belongs to the category of black box optimization and is suitable for problems where gradients are not available or hard to define. Such effort ranges from simple (Johansson 2008; Alteredqualia 2008) to complex (Fogleman 2016; Cason 2016; Paauw and Van den Berg 2019; Shahrabi 2020; Tian and Ha 2022) art forms.

3D Rendering The development of Computer Graphics (Foley et al. 1994; Shirley, Ashikhmin, and Marschner 2009) is largely associated with striving for better three dimension (3D) rendering (Watt 1993). One driven force behind that is the development of game (Gregory 2018) which naturally calls for high quality rendering in real-time (Akenine-Moller, Haines, and Hoffman 2019). Broadly speaking, two ways of rendering exist: first is rasterization (Shirley, Ashikhmin, and Marschner 2009), where polygons representing 3D objects are projected to pixels on 2D screen, which is fast, widely adopted, and often good enough. Another is ray-tracing (Glassner 1989; Spencer and Murty 1962; Appel 1968; Whitted 2005), where rays are traced back from camera, interacting with the objects it encounters accordingly to the rendering equation (Kajiya 1986), all the way till the light source, which enables a high degree of physically plausibility. In the practice of 3D rendering engines, two paradigms exist: One is retained mode graphics (Jin 2006) where the application issues since to graphic libraries, which is the dominating one given the efficiency. Another one is immediate mode paradigm (Radich and Satran 2019) where the application builds the scene and only issues drawing primitives to the graphic libraries. It is less efficient, but allows more flexibility and expression, which is helpful in creativity settings.

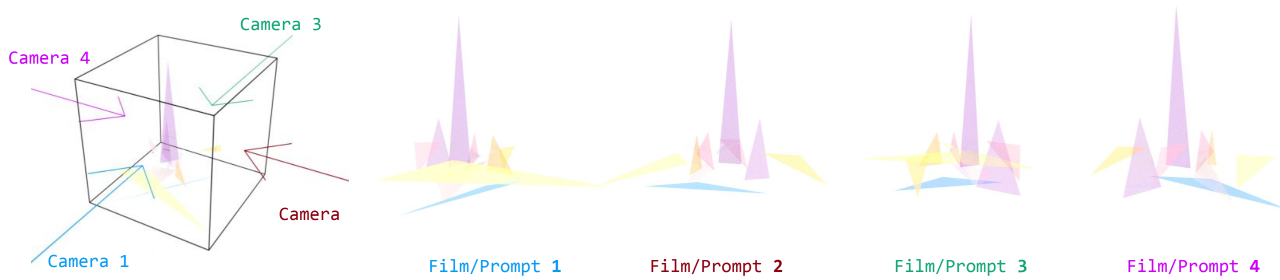
3D Generative Models and Computational Creativity One way of generating 3D objects since early days starts with not the objects themselves but 3D point clouds (Nguyen and Le 2013; Guo et al. 2020) where points in 3D space, with possible unit volume, is the modality of the data. The 3D point cloud is used in turn to generate the 3D shape by morphing the points (Mo et al. 2019; Li et al. 2021). Recently we have seen a surge of high quality generative models that produce data in 3D space. Especially in producing concrete, *volumetric* 3D objects, works in the line of NeRF (Mildenhall et al. 2021; Martin-Brualla et al. 2021) where the whole scene is represented by a radiance field parameterized by the neural base models. Research in this direction is not limited to just working on 3D space. Inspired by multi-modal, text-to-image works such as DALLE (Ramesh et al. 2022), Imagen (Saharia et al. 2022) and Parti (Yu et al. 2022), where creative yet high quality images are generated following the text prompts as guidance, text-to-3D objects have become possible, where works DreamFields (Jain et al. 2022) DrameFusion (Poole et al. 2022) and Imagen Video (Ho et al. 2022), which are capably for generating high-quality video of concrete, *volumetric* objects following the description given in text.

Beside addressing 3D space using generative models, similar problems have also been approached from a computational creativity point of view, in which the artistic creativity of the generated object is emphasized instead of the modeling of data in 3D modality. Early work explores rule based generation (Broughton, Tan, and Coates 1997; Coates, Broughton, and Jackson 1999) where the combination of rules are optimized, or the evaluation is done by either enabling human-in-the-loop (Cook 2007) or parameterizing a single formulae (Chu 2021). Also, recent works produce wire art (Hsiao, Huang, and Chu 2018) that looks like predefined sketches by first generating vortexes and then connecting them by leveraging 3D path finding algorithms. They are probably closest to our work, but crucial differences exist: As far as we know, we are the first work to address *spatial*, *abstract* 3D generation with the expressionism from modern neural based models.

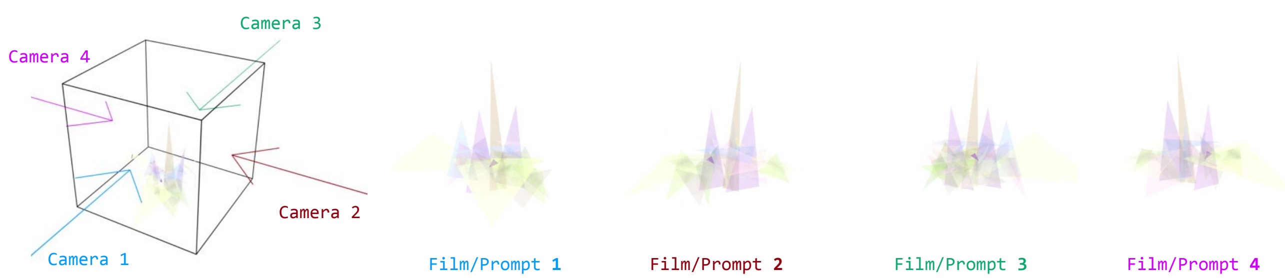
Methodology

We show the overall architecture of our proposed method in Figure 3. It contains two parts, namely the outer loop of evolution strategies, which is a black box optimization that suggests multiple sets of parameters and adjusts them based on the fitness, or how well each set of parameters are, and the inner, actual evaluation of such fitness. At the end of several steps of optimization, the results are parameters with better fitness. For evolution strategies we use CMA-ES (Hansen 2000; 2006) which estimates the covariance matrix of parameters, since it provides better performance than common alternatives like PGPE (Sehnke et al. 2010) while only incurring marginal increase of running time in our case. Engineering-wise, We use the EvoJAX implementation of CMA-ES, which is based on JAX (Bradbury et al. 2018) and runs on accelerators like GPU.

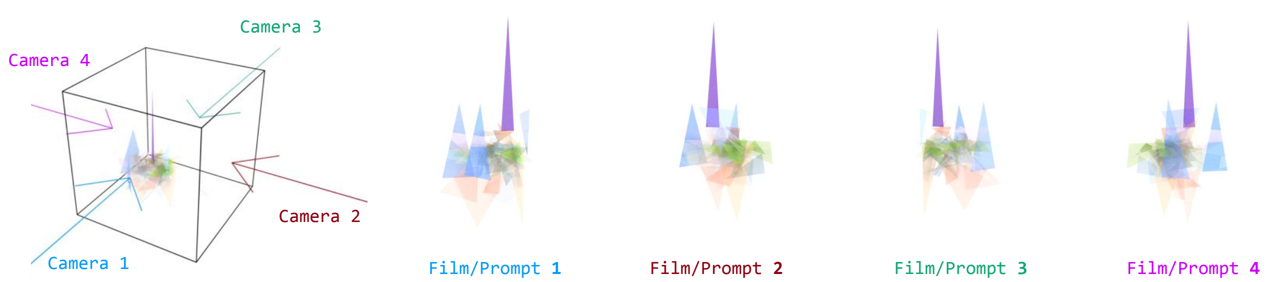
In our setup, the parameters literally parameterize the building of the 3D scene and the rendering of it. We first build the 3D scene and render it from possibly multiple cameras, using the ray-tracing renderer engine. While the actual spatial objects are parameterized by the parameters, how the builder and the renderer interpret these parameters are considered



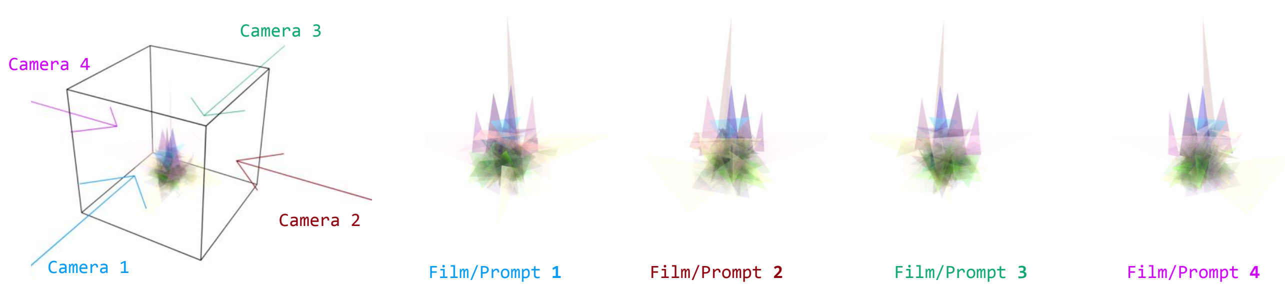
(a) “Walt Disney World” - 10 Triangles.



(b) “Walt Disney World” - 25 Triangles.

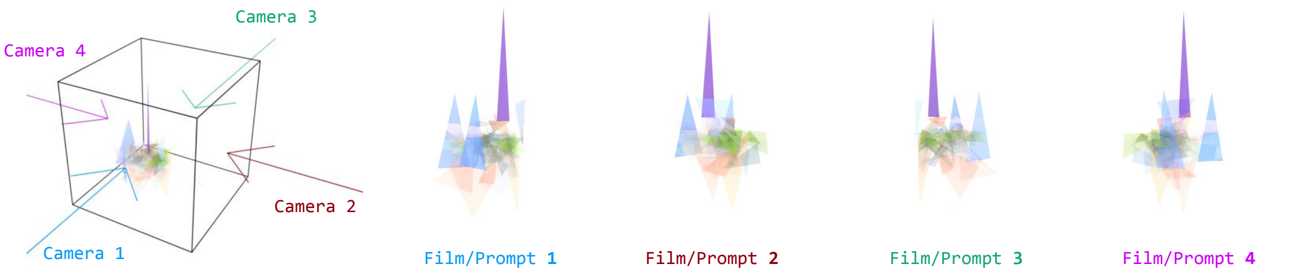


(c) “Walt Disney World” - 50 Triangles.

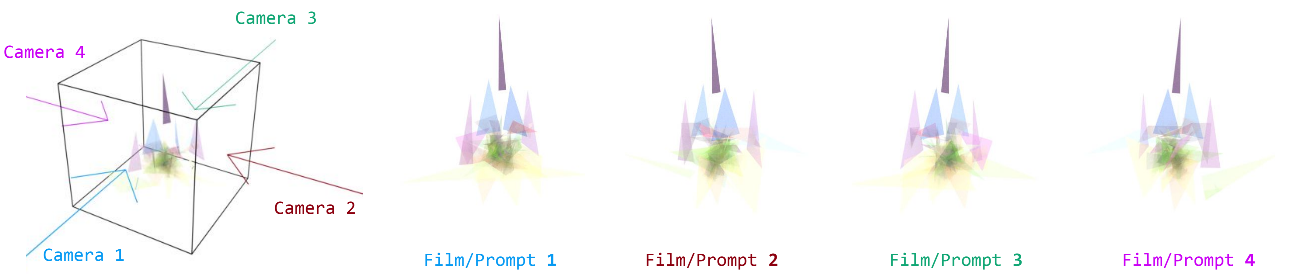


(d) “Walt Disney World” - 100 Triangles.

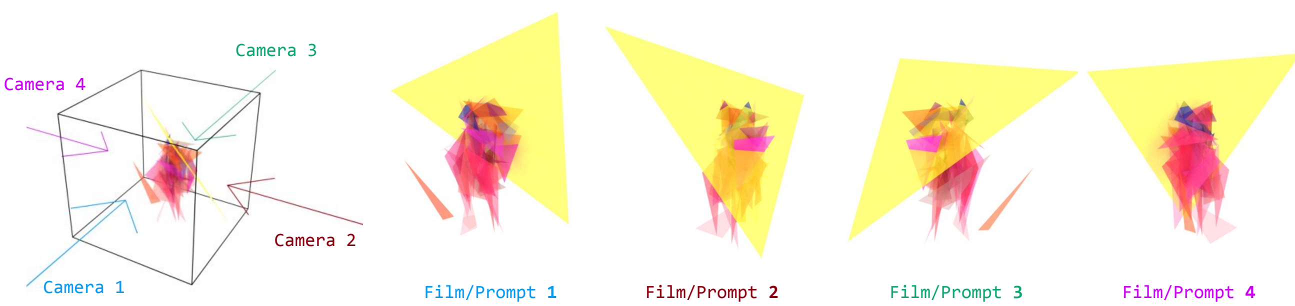
Figure 4: Our method generating with text prompts “Walt Disney World” with four cameras, with different numbers of triangles, namely 10, 25, 50 and 100 respectively. It could be shown that our method leverages the budgets of triangles in the increasing order of granularity, by first using triangles for general shape and then moving towards fine-grained details.



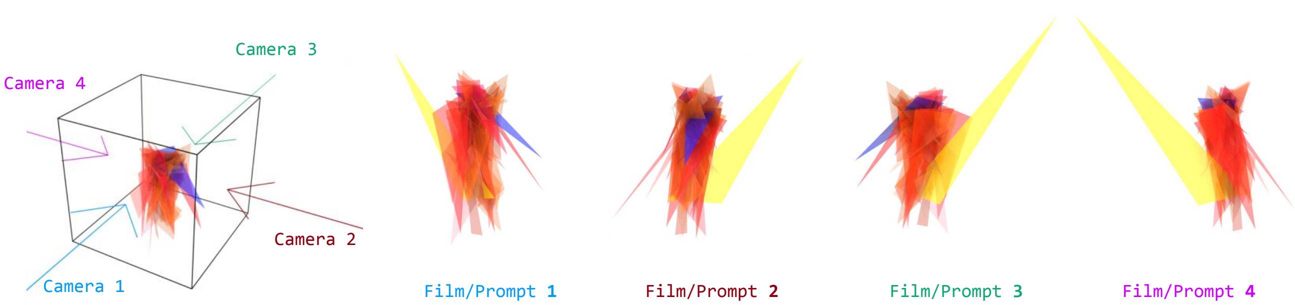
(a) “Walt Disney World” - Run 1



(b) “Walt Disney World” - Run 2

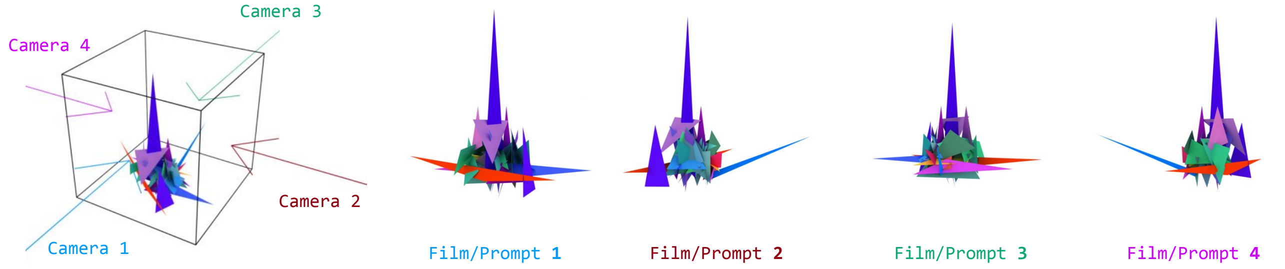


(c) “A bright, vibrant, dynamic, spirited, vivid painting of a dog” - Run 1

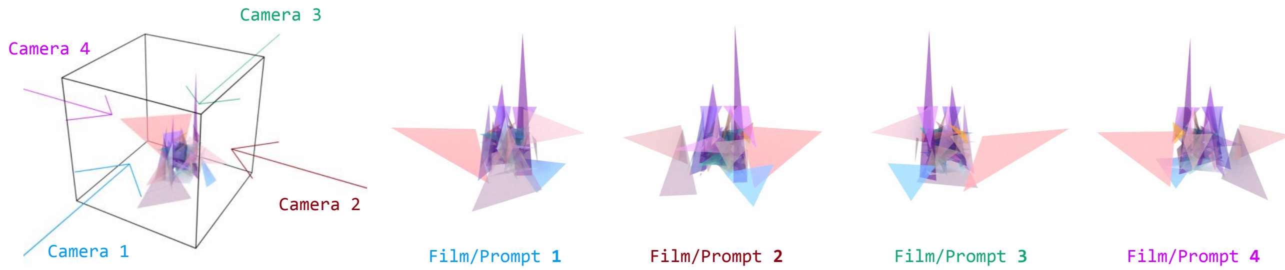


(d) “A bright, vibrant, dynamic, spirited, vivid painting of a dog” - Run 2

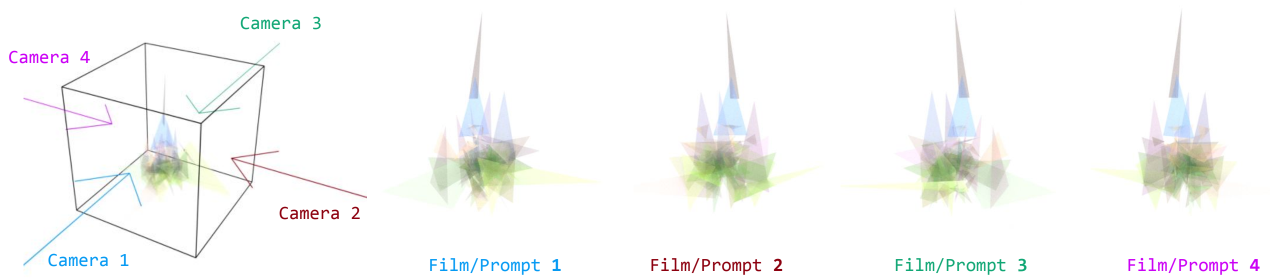
Figure 5: Our method generating two configurations, each with two independent runs. The first configuration is with text prompts “Walt Disney World” with four cameras, while the second configuration is with “A bright, vibrant, dynamic, spirited, vivid painting of a dog”. Different runs lead to equally plausible yet largely different 3D art. An artist user could thus be “in-the-loop” by choosing from different variants from these runs.



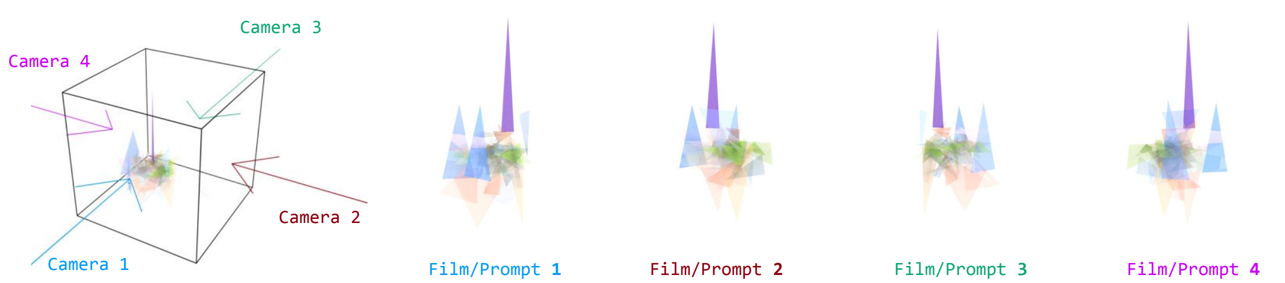
(a) Fixed transparency of 0%.



(b) Fixed transparency of 50%.



(c) Fixed transparency of 80%.



(d) Learnable transparency.

Figure 6: Our method generating with text prompts "Walt Disney World", with four settings of transparency. We show here three fixed transparency of 0%, 50% and 80%, compared with the default setting of learnable transparency. While the fixed transparency setting allows more global control of the scene, the learnable one provides great flexibility in how triangles are related to the space.

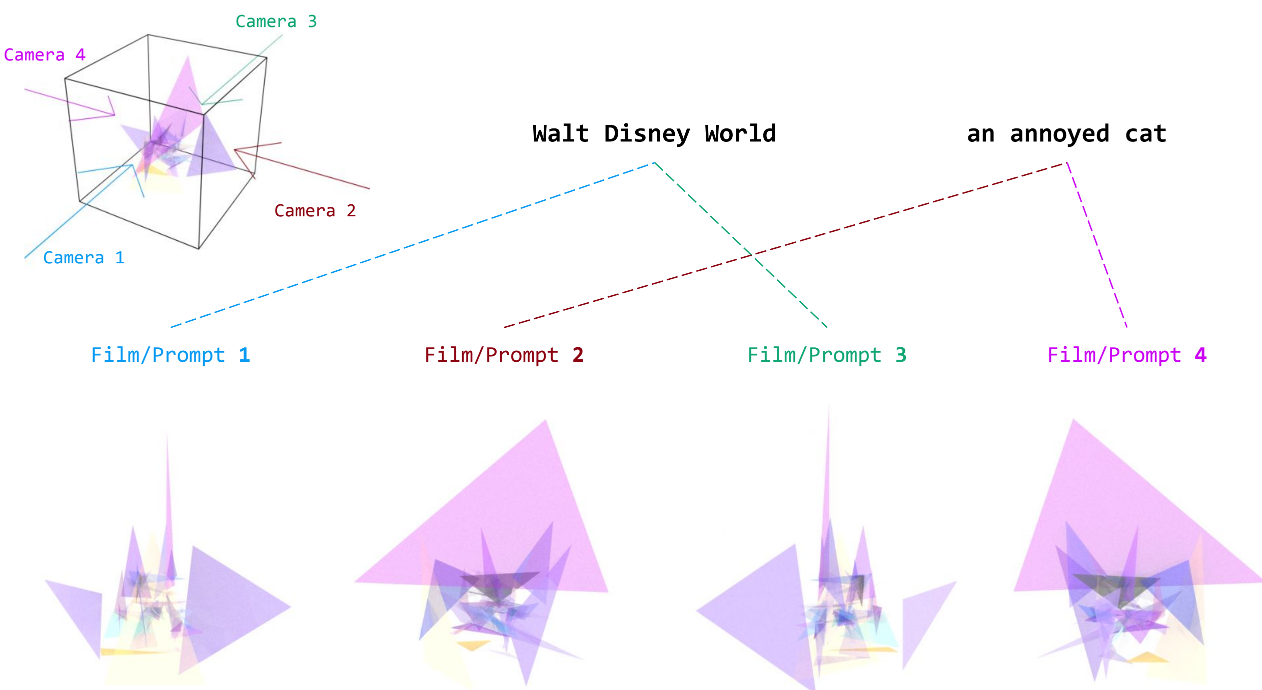


Figure 7: Our method generating with different text prompts at cameras. The text prompt for camera 1 and 3 is "Walt Disney World" and for camera 2 and 4 is "an annoyed cat". Our method produces one 3D art, and successfully allows it to look differently from different angles.

hyperparameters that the artist users could control. Finally, the rendered images from each camera, or "films" following photography terms, are compared with the text prompts semantically, which is done by computing the Cosine loss between images and texts encoded by CLIP encoders. The mean loss of all pairs of images and texts is given back to the aforementioned evolution strategy for adjusting the parameters accordingly.

The parameterization of 3D scenes, or how 3D render interprets the parameters and build the scene, is a core part of the creativity we express in our setup. Since our goal focuses on the computational creativity of *spatial* 3D art as motivated by the trend of abstract art in modern sculpture, we consider placing semi-transparent triangles in plastic material in the 3D space. Concretely, each of N triangles is associated with 13 learnable parameters, namely the position of its three vertices (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3) and the color and transparency (R, G, B, A), thus making totally $13N$ parameters. With the help of a ray-tracing renderer, it is not only possible to archive photo-realistic rendering of these semi-transparent triangles since the light ray could pass through and bounce between them many times, but also retain the possibility of reproducing a solution in the real world. The choice of semi-transparent triangles is inspired by a recent work on 2D (Tian and Ha 2022), but going to 3D, as in our setting, makes our whole new pipeline necessary since the technique and the optimization dynamics are completely different.

In practice, for ray-tracing rendering we use physic-based Mitsuba 3 renderer (Nimier-David et al. 2019). For each triangle, the bidirectional scattering distribution function (BSDF) for rendering is set as a mixture of BK7 Glass, a thin dielectric material, and a *Lambertian*, an ideally diffuse material of corresponding R, G, B value, mixed with ratio A. Besides physically correct rendering, Mitsuba 3 also allows GPU-powered paralleling sampling, which largely accelerates the ray-tracing rendering. Notably, Mitsuba 3 is capable of differentiable rendering, but we do not leverage such capacity and leave differentiable rendering as an orthogonal research direction for future study.

As we expect the pipeline to produce a scene that, when rendered from different cameras (we call what a camera produces "films"), looks like corresponding text semantically, We measure it using CLIP, which provides an image and a text encoder that projects images and text into a shared, comparable latent space with Cosine distance or loss. Note that with multiple pairs of camera and text, which means we could make the produced 3D object look like (or different) from different directions. In doing so, each film is encoded and compared with corresponding text which is also encoded, and the means of Cosine loss of all such pairs are used as fitness, which is given back to the evolution strategies.

Finally, since both the rendering and the evolution strategies we use are fully run-able on GPU, the computation is fast and in our experience can be tens of times faster than on CPU, thus fully leveraging the modern hardware accelerators.

Experiments

In this section we showcase our method with several experiments. In Figure 2, we show several examples of the evolved 3D art produced by our method, each with $1,200$ steps of evolution and a population of 128 in CMA-ES. As shown here, our method demonstrates that a wide range of text prompts can be handled by our method, producing spatial, abstract art that is both novel and consistent with human interpretation. Even given the abstract nature defined by the scene, our method could still handle both the spatial shape (first two examples) and the color (last two examples).

In the rest of this section, we investigate how turning several important hyperparameters could impact the finally generated 3D art, showing the dynamics of our method which could be served as a guidance for the artist users.

Different Number of Triangles The number of triangles could be used as a kind of "budget" that our method used to allocate in occupying the 3D space. In Figure 4, we show our method generating with 10, 25, 50 and 100 triangles, respectively. Our method thus places triangles in the increasing order of granularity, where general outline is first emphasized and details are then filled. Note that the number of parameters increases in proportion to the number of triangles, requiring more computation time in rendering and optimization. Thus, this is a balance that artist users should make.

Different Runs of the Same Configuration One important aspect of computational creativity is the ability to produce variants of the art given the same instruction. Such a property not only allows artist to be "in-the-loop" to choose from a wide range of variants, but also shows the capacity of the generative model. In Figure 5, we show two configurations, each with two independent runs. It is shown that different runs lead to equally plausible yet largely different 3D art. In doing so, our method could support artist be "in-the-loop" of the creativity process.

Fixed v.s. Learnable Transparency Unlike 2D art, in 3D art the transparency matters a lot, especially in the spatial setting as we focus on. In Figure 6, we demonstrate several setting of transparency, including the fixed transparency and the default setting of learnable transparency. It shows that while fixed transparency allows a more consistent global outlook, it nonetheless limits the expression by forcing large and small triangles contributing the same to the images. In contrast, learnable transparency gives our method flexibility in how the triangles are related to the 3D space.

Different Text Prompts at Different Cameras While in many examples we show the same text prompt for cameras, this is completely not a requirement imposed by our method. On the contrary, our method allows pairs of texts and cameras in an arbitrary combination. Such a capacity allows a wider range of creativity from users to, for example, generate a 3D art that looks differently from different angles. In Figure 7 we demonstrate one such case, where our method generates 3D art that looks like "Walt Disney World" from two directions but "an annoyed cat" from the other two directions, even these views are of the same, single 3D art. We argue that our method is the first to be capable of helping artists in such a creative process that previously requires lots of manual work (Hsiao, Huang, and Chu 2018).

Conclusion

In this work we address the problem that is previously not studied — generating 3D, abstract, and spatial art that is semantically aligned with human interpretation. In doing so, we propose to leverage evolution strategies (ES) with ray-tracing rendering of parameterized 3D scenes, along with CLIP method for measuring the semantic similarity. We demonstrate that our approach is capable of producing 3D arts through several experiments, and provides the flexibility for artists or users to fine-tune for the desired result.

Nonetheless, our proposed method is best suited as a call for further future study in computational approaches in 3D art. For example, it remains unclear whether optimization using differentiability of the renderer would lead to a different dynamic and thus art style. Also, the designing of the parameterized scene is a time-consuming one requiring the extensive knowledge of 3D rendering and optimization, so whether it would be improved through (semi-)automation process should be studied too.

References

Akenine-Moller, T.; Haines, E.; and Hoffman, N. 2019. *Real-time rendering*. AK Peters/crc Press.

Alteredqualia. 2008. Evolution of mona lisa in javascript and canvas. https://alteredqualia.com/visualization/evolve/. Last accessed 1 Dec 2021.

Appel, A. 1968. Some techniques for shading machine renderings of solids. In *Proceedings of the April 30–May 2, 1968, spring joint computer conference*, 37–45.

Bertoni, F. 2002. *Minimalist architecture*. Birkhäuser.

Beyer, H.-G., and Schwefel, H.-P. 2002. Evolution strategies–a comprehensive introduction. *Natural computing* 1(1):3–52.

Beyer, H.-G. 2001. *The theory of evolution strategies*. Springer Science & Business Media.

Bradbury, J.; Frostig, R.; Hawkins, P.; Johnson, M. J.; Leary, C.; Maclaurin, D.; Necula, G.; Paszke, A.; VanderPlas, J.; Wanderman-Milne, S.; and Zhang, Q. 2018. JAX: composable transformations of Python+NumPy programs.

Britannica, T. E. o. E. 2021. computer animation. https://www.britannica.com/topic/computer-animation. Last accessed 1 Nov 2022.

Broughton, T.; Tan, A.; and Coates, P. 1997. The use of genetic programming in exploring 3d design worlds. In *CAAD futures 1997*. Springer. 885–915.

Caro, S. A. 1962. Early one morning. https://artsandculture.google.com/asset/early-one-morning-sir-anthony-caro/AQEGPfzosYQ1tg. Last accessed 1 Nov 2022.

Cason, K. 2016. Genetic draw. https://github.com/kennycason/genetic_draw. Last accessed 1 Dec 2021.

Chu, E. 2021. Evolving evocative 2d views of generated 3d objects. *arXiv preprint arXiv:2111.04839*.

Coates, P.; Broughton, T.; and Jackson, H. 1999. Exploring three-dimensional design worlds using lindenmayer systems and genetic programming. *Evolutionary design by computers* 323–341.

Conroy, S. B. 1977. Spatial sculpture. https://www.washingtonpost.com/archive/lifestyle/1977/07/03/spatial-sculpture/3264a1b3-cc62-40d0-b11d-1a9127d4ed5d/. Last accessed 1 Nov 2022.

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

Dabrowski, M. 2004. Geometric abstraction. https://www.metmuseum.org/toah/hd/geab/hd_geab.htm. Last accessed 1 Dec 2021.

Fogleman, M. 2016. Primitive pictures. https://github.com/fogleman/primitive. Last accessed 1 Dec 2021.

Foley, J. D.; Van Dam, A.; Feiner, S. K.; Hughes, J. F.; and Phillips, R. L. 1994. *Introduction to computer graphics*, volume 55. Addison-Wesley Reading.

Glassner, A. S. 1989. *An introduction to ray tracing*. Morgan Kaufmann.

Gowans, A.; Collins, P.; Scruton, R.; and Ackerman, J. 2022. architecture. https://www.britannica.com/topic/architecture. Last accessed 1 Nov 2022.

Gregory, J. 2018. *Game engine architecture*. AK Peters/CRC Press.

Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; and Bennamoun, M. 2020. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence* 43(12):4338–4364.

Hansen, N. 2000. Invariance, self-adaptation and correlated mutations in evolution strategies. In *International Conference on Parallel Problem Solving from Nature*, 355–364. Springer.

Hansen, N. 2006. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation* 75–102.

Hemmendinger, D. 2014. computer graphics. https://www.britannica.com/topic/computer-graphics. Last accessed 1 Nov 2022.

Ho, J.; Chan, W.; Saharia, C.; Whang, J.; Gao, R.; Gritsenko, A.; Kingma, D. P.; Poole, B.; Norouzi, M.; Fleet, D. J.; et al. 2022. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*.

Hsiao, K.-W.; Huang, J.-B.; and Chu, H.-K. 2018. Multi-view wire art. *ACM Transactions on Graphics (TOG)* 37(6):1–11.

Jain, A.; Mildenhall, B.; Barron, J. T.; Abbeel, P.; and Poole, B. 2022. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 867–876.

Jakob, W.; Speierer, S.; Roussel, N.; and Vicini, D. 2022. Dr.jit: A just-in-time compiler for differentiable rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)* 41(4).

Jin, Z. 2006. Retained-mode parallel graphics rendering for cooperative design environments. In *2006 10th International Conference on Computer Supported Cooperative Work in Design*, 1–6. IEEE.

Johansson, R. 2008. Genetic programming: Evolution of mona lisa. https://rogerjohansson.blog/2008/12/07/genetic-programming-evolution-of-mona-lisa/. Last accessed 1 Dec 2021.

Kajiya, J. T. 1986. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, 143–150.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kolmogorov, A. 1965. Three approaches to the quantitative definition of information. *Problems of information transmission* 1(1):1–7.

Kricke, N. 1976. Spatial sculpture. https://artsandculture.google.com/asset/spatial-sculpture/2gF69CmELoj4Fw. Last accessed 1 Nov 2022.

Kuiper, K. 2021. Modernism. https:

//www.britannica.com/art/Modernism-art. Last accessed 1 Dec 2021.

Li, R.; Li, X.; Hui, K.-H.; and Fu, C.-W. 2021. Sp-gan: Sphere-guided 3d shape generation and manipulation. *ACM Transactions on Graphics (TOG)* 40(4):1–12.

Lowood, H. E. 2021. electronic game. https://www.britannica.com/topic/electronic-game. Last accessed 1 Nov 2022.

Malkevitchn, J. 2003. Mathematics and art. https://www.ams.org/publicoutreach/feature-column/fcarc-art1. Last accessed 1 Dec 2021.

Martin-Brualla, R.; Radwan, N.; Sajjadi, M. S.; Barron, J. T.; Dosovitskiy, A.; and Duckworth, D. 2021. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7210–7219.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* 65(1):99–106.

Mo, K.; Guerrero, P.; Yi, L.; Su, H.; Wonka, P.; Mitra, N.; and Guibas, L. J. 2019. Structurenet: Hierarchical graph networks for 3d shape generation. *arXiv preprint arXiv:1908.00575*.

Modern, T. 2018. Minimalism. https://www.tate.org.uk/art/art-terms/m/minimalism. Last accessed 1 Dec 2021.

Nguyen, A., and Le, B. 2013. 3d point cloud segmentation: A survey. In *2013 6th IEEE conference on robotics, automation and mechatronics (RAM)*, 225–230. IEEE.

Nimier-David, M.; Vicini, D.; Zeltner, T.; and Jakob, W. 2019. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG)* 38(6):1–17.

Paauw, M., and Van den Berg, D. 2019. Paintings, polygons and plant propagation. In *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*, 84–97. Springer.

Paul, S. 2004. Abstract expressionism. https://www.metmuseum.org/toah/hd/abex/hd_abex.htm. Last accessed 1 Dec 2021.

Poole, B.; Jain, A.; Barron, J. T.; and Mildenhall, B. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Radich, Q., and Satran, M. 2019. Retained mode versus immediate mode. https://learn.microsoft.com/en-us/windows/win32/learnwin32/retained-mode-versus-immediate-mode. Last accessed 1 Nov 2022.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Rewald, S. 2014. Cubism. *The Metropolitan Museum of Art*. https://www.metmuseum.org/toah/hd/cube/hd_cube.htm. Last accessed 1 Dec 2021.

Rogers, L. R. 2020. Sculpture. https://www.britannica.com/art/sculpture. Last accessed 1 Nov 2022.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.; Ghasemipour, S. K. S.; Ayan, B. K.; Mahdavi, S. S.; Lopes, R. G.; et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*.

Schmidhuber, J. 1997. Low-complexity art. *Leonardo* 30(2):97–103.

Sehnke, F.; Osendorfer, C.; Rückstieß, T.; Graves, A.; Peters, J.; and Schmidhuber, J. 2010. Parameter-exploring policy gradients. *Neural Networks* 23(4):551–559.

Shahrabi, S. 2020. Procedural paintings with genetic evolution algorithm. https://github.com/IRCSS/Procedural-painting. Last accessed 1 Dec 2021.

Shirley, P.; Ashikhmin, M.; and Marschner, S. 2009. *Fundamentals of computer graphics*. AK Peters/CRC Press.

Spencer, G., and Murty, M. 1962. General ray-tracing procedure. *JOSA* 52(6):672–678.

Such, F. P.; Madhavan, V.; Conti, E.; Lehman, J.; Stanley, K. O.; and Clune, J. 2017. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567*.

Tang, Y.; Tian, Y.; and Ha, D. 2022. Evojax: Hardware-accelerated neuroevolution. *arXiv preprint arXiv:2202.05008*.

Tian, Y., and Ha, D. 2022. Modern evolution strategies for creativity: Fitting concrete images and abstract concepts. In *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*, 275–291. Springer.

Toklu, N. E.; Liskowski, P.; and Srivastava, R. K. 2020. Clipup: A simple and powerful optimizer for distribution-based policy evolution. In *International Conference on Parallel Problem Solving from Nature*, 515–527. Springer.

Verostko, R. 1994. Algorithmic art. http://www.verostko.com/algorithm.html. Last accessed 1 Dec 2021.

Watt, A. 1993. 3d computer graphics.

Whitted, T. 2005. An improved illumination model for shaded display. In *ACM Siggraph 2005 Courses*. 4–es.

Yu, J.; Xu, Y.; Koh, J. Y.; Luong, T.; Baid, G.; Wang, Z.; Vasudevan, V.; Ku, A.; Yang, Y.; Ayan, B. K.; et al. 2022. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*.