# The Investigation of Deep Learning Models Utilized in Vector Graphics Manipulation

Chen Yang *

Graduate School of Architecture, Planning and Preservation, Columbia University, New York, United States

* Corresponding author: cy2581@columbia.edu

**Abstract.** Vector graphics, 2D or 3D, hold paramount significance across various professional domains, including graphic design, web design, architecture, and engineering. However, traditional methods of creating vector graphics are characterized by low efficiency. This review explores the integration of some deep learning models designed for 2D and 3D vector graphics generation and manipulation, summarizing their main tasks and methods. In terms of 2D vector graphics, this review examines advanced models, including Convolutional Neural Networks, Generative Adversarial Networks, and more, for diverse tasks such as font or icon generation and image manipulation. For 3D vector graphics, this paper assesses the progress achieved in models tailored for point cloud and image reconstruction, as well as 3D shape generation, using approaches such as Variational Autoencoders, Multi-Layer Perceptrons, and Transformers. This review also assesses their progress and limitations, acknowledging a comprehensive overview of deep learning models in vector graphic manipulation, and emphasizing their potential impact on the design industry while recognizing the challenges ahead.

**Keywords:** Computer Vision, Artificial Intelligence, Deep Learning, Vector Graphics, 3D Models.

## 1. Introduction

Vector graphics [1] is a visual form describing images composed directly of mathematical models of analytic or coordinate geometries, such as points, lines, meshes, and polygons, which are defined by their coordinates on a Cartesian Plane as p = (x, y) or p = (x, y, z). From a high level, vector graphics can be categorized into at least two categories, 2-dimensional (2D) and 3-dimensional (3D), and both have a wide range of applications. Scalable Vector Graphics (SVG) [2] is a universal 2D vector graphic format across graphic design or web design, while Computer-Aided Design (CAD) [3] is the utilization of the computer to aid in the graphic creation, shape modification, context analysis, or optimization of a design, often applying in 3D modeling of architecture or engineering. CAD files can be projected and converted into SVG effortlessly, and vice versa, via CAD programs, representing that both are vector graphics described by a collection of parametric primitives. Summarizing the use of artificial intelligence methods in these formats, which can boost the efficiency of the designer's workflow, is of great importance in understanding the influence of cutting-edge technology on the design industry.

Song, Y [4] utilizes the multi-round vectorization strategy and the differentiable rasterization to generate more accurate raster-based images guided by text inputs, which is a compromise from direct manipulation of 2D vector graphics, given the significant number of studies regarding the manipulation and application of raster-based images [5-7], of which the success is attributed to the efficiency of Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs) or Contrastive Language-Image Pretraining (CLIP) [8]. In works on an effective representation and manipulation of 2D vector images, several other models have been proposed. They have articulated the potential applied impact of artificial intelligence in vector graphic representation, such as fonts [9-13], animation, and logo design [14]. SVGs have excellent scaling capabilities and their digital sequential format makes them a great option for storing 2D spatial patterns. However, it is still challenging to extract the geometric information from these patterns to produce optimal results. This is also an important area of study for improving productivity in the design industry.

For 3D graphics, two categories can be classified: voxel-based (or point cloud-based) and shape-based methods [15, 16]. Point cloud [17] is the standard acquisition format for devices like Lidar cameras on iPhones, and they provide a straightforward way to perform geometric operations. Some early research [15, 18, 19] focused on transforming point cloud model to a shape-based model. Later studies such as [20] proposed the model architecture which outputs a sequence of operations in CAD programs, which can be discretized into polygon meshes or point clouds as needed. As for modification or generation of the 3D graphics, [16, 21] both introduced models of direct operations to shape-based graphics.

Though a great amount of research and studies conveyed the Deep Learning (DL) models and representation learning in creating or categorizing vector graphics, few have summarized their features and specificities. Besides, rare studies considered the relationship between 2D and 3D graphics, which are both inevitable to utilize by designers or modeling engineers. Those which mentioned the relationship [22] are trying to build a workflow either generating 2D to 3D or vice versa. This paper will also analyze the relationship between models of 2D and 3D graphics, especially emphasize the lack of the focus on DL models of 3D vector graphics recently. Moreover, the application will be the key element in the following contents. The goal of this review is to identify the importance of integrating Artificial Intelligence (AI) into 2D and 3D vector graphics in the design industry and its prospects.

## 2. Methods

### 2.1. 2-Dimensional Vector Graphics

This section is categorized by the main task of each model, given their complicated interrelated relationships and methods. The vector graphics mentioned in this section will all be in the format of SVG. The brief summary of all models mentioned below is displayed in Table 1.

**Table 1.** The Summary of 2D Vector Graphic DL Models

| Model | TASK | Encoding Modality | Decoding Modality | Model Architecture | YEAR |
|---|---|---|---|---|---|
| Im2Vec [12] | Image Manipulation | img* | vec** | RNN | 2021 |
| SVG-VAE [11] | Font Generation | img* | img*/vec** | VAE | 2019 |
| SVGformer [10] | Image Manipulation | vec** | vec** | Transformer | 2023 |
| DeepSVG [14] | Image Manipulation | vec** | vec** | Transformer | 2020 |
| DeepVecFont [13] | Font Generation | img*/vec** | img*/vec** | LSTM + CNN | 2023 |
| CLIPVG [4] | Image Manipulation | img* | img * | CLIP + GAN | 2023 |

* “img” represents “raster-based image”

** “vec” represents “2D vector graphics”

#### 2.1.1. Font Generation

Font generation, which widely used 2D vector graphics, is a critical aspect in graphic design. SVG-VAE [11], taking inspiration from the generative models of rasterized images, focuses on font generation mainly and opens the possibilities on more complicated graphics. The model consists of a Variational Autoencoder (VAE) and an autoregressive decoder which contains 4 stacked Long Short-Term Memories (LSTMs) [23] and a Mixture Density Network (MDN) for generating SVG commands, which also allows it to be trained end-to-end in principle.

In addition, DeepVecFont [13] established a CNN model architecture based on previous works [9, 12], focusing on two major problems, in which one is that in the previous works usually only one modality of glyphs is considered when encoding, the other is the issue of location shifts caused by Mixture Distribution Models. To respond to these two issues, the authors built a learning strategy for dual- to deal with image-aspect and sequence-aspect features, where in this case, vector font synthesis could be viewed as a sequence generation problem where drawing commands are executed sequentially. By introducing a new approach to sort unstructured data, DeepVecFont can generate visually appealing vector outlines by using the input of original vector fonts.

### 2.1.2. Icons

Icons, like fonts, are something inevitable to discuss when speaking of the presentation of 2D vector graphics. DeepSVG [14] is proposed mainly for complicated generation and interpolation of SVG icons. It utilizes the VAE, containing an encoder as well as a decoder network, which are designed by taking into account the hierarchical structure of an SVG image and using Transformer-based model. The model is able to predict the commands in a fee-forward manner, which differs from the autoregressive strategy employed in previous work [11]. The authors also evaluate the performance of the model by conducting an ablation study, animation by interpolation and font generation.

### 2.1.3. Image Manipulation

Im2Vec [12] builds a model for 2D vector graphics which does not require the supervision of vector. It maps a raster-based image and then decodes it into a vector graphic sequence. It generates complicated graphics which are made of paths of various lengths and topologies. An additional model is added to predict the optimal number of control points for paths. At the end, the vector graphics are rasterized by implementing rasterizer and the rendering. The performance of reconstruction, generation, and interpolation is evaluated quantitatively by the authors. Additionally, they compare the obtained results with other models [9, 14] for these tasks.

SVGformer [10] is based on transformer architecture which has been developed to cater to various vector graphics downstream tasks. The embedding layers contain the continuous token component, the position information and the geometric component. The encoder module receives the embedding layer and produces a condensed representation with improved receptive field efficiently. Given a masked input, the decoder's objective is to reconstruct the original SVG commands, while also predicting their types. The model is evaluated on fonts dataset, icons dataset and downstream tasks without any supervision, which shows the results of good interpretability.

CLIPVG [4] is the first image manipulation framework based on CLIP which requires no additional generative models. First, the input image is vectorized multiple times with varying precision, and then they are joined back into rasterized images using Diffvg [24]. The iterative algorithms are designed to be optimized in the direction of the text prompts. With the process of vectorization, the model archives better precision during generation and reconstruction compared to other one-shot cases.

## 2.2. 3-Dimensional Vector Graphics

This section will be categorized based on the main tasks that each model deals with, such reconstruction or shape generation. The brief summary of all models mentioned below are displayed in Table 2.

**Table 2.** The Summary of 3D Vector Graphic DL Models

| Model | TASK | Encoding Modality | Decoding Modality | Model Architecture | YEAR |
|---|---|---|---|---|---|
| AtlasNet [18] | Reconstruction | pc* / img** | mesh | MLP | 2018 |
| DeepMarchingCubes [15] | Reconstruction | pc* | mesh | CNN | 2019 |
| ShapeAssembly [16] | Shape Generation / Reconstruction | pc* | shape | VAE | 2020 |
| IM-NET [19] | Shape Generation / Reconstruction | pc* / img** | shape | CNN / VAEs / GANs | 2023 |
| DeepCAD [20] | Shape Generation / Reconstruction | shape | command seq*** | Transformer | 2023 |
| BRepNet [21] | Shape Generation | shape | shape | CNN | 2023 |

* “pc” represents “point clouds”
** “img” represents “raster-based image”
*** “seq” represents “sequence”

### 2.2.1. Point Cloud or Image Reconstruction

Given that point clouds are essentially data instead of editable shapes, some early models focused on the reconstruction from point clouds to shape-based graphics.

AtlasNet [18] is a new technique for creating 3D surfaces by presenting them as a set of parametric surface elements, instead of generating point clouds or voxel elements. Locally mapping a set of squares to the surface of 3D shapes is introduces as an approach to approximating the target surface. The model takes not only a latent shape representation but also 2D points sampled in the unit square as input, which can be sampled at any resolution, and it generate points on the surface. The output, the continuous image of a planar surface, has the advantages on surface generation of high resolution from point clouds and 2D images. The model is tested on the tasks of generating meshes and reconstruction form images or point clouds.

DeepMachingCubes [15] is a model which predicts explicit surface representations of arbitrary topology, such as meshes, point cloud or an image. Based on the previous work of Marching Cubes algorithm [25], the model proposes a differentiable representation which avoids the problem of backpropagation through Marching Cubes algorithm. Then it exploits the representation for its goal of the prediction of surfaces. The model is verified through end-to-end learning for the prediction of 3D shapes from point clouds.

IM-NET [19] is a decoder of implicit fields to generate shapes. It receives a feature vector that has been produced by a shape encoder as input, as well as a point coordinate, 3D or 2D. The decoder then outputs a value indicating whether the vector is inside or outside the shape, relative to its position. Embedding IM-NET into Generative Adversarial Networks (GANs) and autoencoders (AEs) leads to IM-GANs and IM-AEs, which both are evaluated by 2D or 3D shape generation and interpolation and 3D construction of 2D image.

### 2.2.2. 3D Shape Generation and Manipulation

The ways of generating and manipulating 3D shapes are versatile and based on various model architecture, which is also of great importance in as a branch of 3D vector graphics manipulation. Some of the models are introduced below.

Constructing shape structure using Shape Assembly [16] involves declaring cuboid part proxies and connecting them hierarchically. The authors also trained a generative model which is able to write ShapeAssembly programs. By utilizing both procedural and deep generative models, this approach takes advantage of the representation's strong points, resulting in outcomes that are modifiable and divergent. The input for this approach is a large dataset of 3D parts graphs, which is transformed into a Domain-Specific Language (DSL) for connected and hierarchical shapes A dataset of shape-program pairs is created to learn how to generate programs that can in turn generate editable shapes.

Editing mesh outputs can be troublesome. Thus, some models are studying the direct manipulation of CAD models. DeepCAD [20] proposed a generative model directly toward CAD designs. It revolves around a representation of CAD command sequences, which is tailored to be fed into neural networks and leads to a natural objective function for training. The model architecture outputs the predicted command sequence, which generate a CAD model with smooth surface and editable features unlike point cloud 3D graphics. Additionally, BRepNet [21] is a model architecture developed to perform directly on boundary representation (B-rep) shapes, avoiding to approximate the meshes or point clouds. In BRepNet, the convolutional kernels are defined using coedges, which are topological entities. The learnable parameters in the kernel help to multiply adjacent feature vectors from a small group of faces, edges, and coedges. By performing convolution, the concealed states are extracted, which can then be utilized for face segmentation. It has been demonstrated by the outcomes that BRepNet is capable of providing more precise segmentation of shapes in comparison to mesh and point cloud methods.

## 3. Discussion

Despite extensive studies and the proposal of substantial models, the drawbacks and limitations have not received widespread discussion. Especially in terms of the relationship between 2D and 3D vector graphics and their application in the industry, although many of them envision the prospect, they lack the point of views in some critical aspects. In the following section, the current progress and application of research will also be discussed.

### 3.1. Progress

The current research in 2D vector graphics manipulation focuses more on specific branch areas of graphic design, for instance icons, logos, animation or image manipulation. All models were tested on various datasets and have the benefit of providing a vector graphic representation for imagery which often relate to design concept or ideas. Studies like [11] proposed immediate opportunities such as combining the current studies with the new attention-based model [26], which has been utilized in some models [10] mentioned above. Some models [4] have the capability to control the image generation with the concept of layers, empowering it to a edit partial region of one image. These are all leading directions that DL models of 2D vector graphics are focusing on.

As for the 3D vector graphics, some models focused on transformation from images or point clouds to meshes or other shape-based graphics, which could be manipulated later on programs like Maya, while some other models were proposed to generate or modify 3D vector graphics. While the amount is still few, some latest models are able to generate command sequences which give them control to CAD programs, such as AutoCAD or Rhino, to generate editable models for the designers. Models were tested and compared on various dataset. All these ensure a foreseeable future of application for general users to operate.

### 3.2. Limitation and Challenges

Though there is great progress in dealing with vector graphics based on DL models, some limitation and challenges of the current models must be considered. Some models, regardless 2D or 3D related, were only tested on limited datasets, which could be too simple and basic as opposed to the complicated scenarios in practice. For example, SVG-Fonts [11] was used on the experiments of [13, 14]. Although the dataset includes thousands of fonts, the results in the papers only show some English letters, causing the overall results are not intuitive enough for readers to understand their performance. This gives rooms for those models to be verified in broader application situations and on other languages other than English. In terms of 3D related models, dataset proposed in [27] were tested on [16] although many forms in the dataset are simple geometries. In comparison, the shapes used in the design industry could easily go up to as many as millions of meshes, which makes the tests that have been done still need examination on more complicated scenarios.

And overall speaking, no matter what inputs that the models are taken, their lack of controllable input parameters weakens the controllability of results in practice, making it hard for people to comprehend how the models make their decisions and produce outcomes. The DL models are generally conceived as black boxes, which do not fully empower users to control the generation process, causing the randomness in the results. This could be a deal breaker in many cases, such as the high precision requirement in the industry design, which often requires the precision of millimeters on their 2D or 3D models. Authorizing users to control the parameters or influence factors of the process is something that should be paid more attention to when designing these model architectures.

### 3.3. Prospects and Application

Nonetheless, the DL models above are revolutionary and have huge potential in application. Some other studies that could be dig in includes the 2D and 3D vector graphic transformation, given such a topic is critical in many fields such as architectural design, where architects often move 2D blueprints

to 3D models or export from 3D models to 2D drawings, all in vector graphic formats. Augmented Reality is also an intriguing area which could utilize the power of converting images or pointes cloud to usable 3D shapes in the virtual environments. With evolving by time, it is not hard to see a marvel future beyond these studies.

## 4. Conclusion

In summary, this paper introduces some leading DL models that focus on manipulating 2D and 3D vector graphics. Their methods and ideas to accomplish their main tasks are briefly summarized, where these models often use techniques like VAEs, CNNs, GANs and Transformers. In the discussion section, their current progress, limitation, and prospects are introduced. The analysis shows the great potential in this field, ranging from cross-model transformation to the application in the industry, where addressing these methods are essential for advancing this interdisciplinary field.

## References

[1] Chapman, N.P., J. Chapman, and I. NetLibrary, Digital multimedia. Worldwide series in computer science. 2000, Chichester; New York: Wiley.

[2] W3C. Scalable Vector Graphics (SVG) 1.1 (Second Edition). 2011; Available from: https: //www.w3.org/TR/SVG11/.

[3] SARCAR, M.M.M., K. M. RAO, and K.L. NARAYAN, Computer Aided Design and Manufacturing. 2008: PHI Learning.

[4] Song, Y., et al. CLIPVG: text-guided image manipulation using differentiable vector graphics. in Proceedings of the AAAI Conference on Artificial Intelligence. 2023.

[5] Lee, J.R., L. Wang, and A. Wong, Emotion net nano: An efficient deep convolutional neural network design for real-time facial expression recognition. Frontiers in Artificial Intelligence, 2021. 3: p. 609673.

[6] Zhang, M., et al. An end-to-end deep learning architecture for graph classification. in Proceedings of the AAAI conference on artificial intelligence. 2018.

[7] LeCun, Y., et al., Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998. 86 (11): p. 2278 - 2324.

[8] Kim, G., T. Kwon, and J.C. Ye. Diffusion clip: Text-guided diffusion models for robust image manipulation. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[9] Campbell, N. D. F. and J. Kautz, Learning a manifold of fonts. ACM Trans. Graph., 2014. 33 (4): p. Article 91.

[10] Cao, D., et al. SVGformer: Representation Learning for Continuous Vector Graphics using Transformers. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

[11] Lopes, R.G., et al. A learned representation for scalable vector graphics. in Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

[12] Reddy, P., et al., Im2Vec: Synthesizing Vector Graphics without Vector Supervision, in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021, IEEE Computer Society. p. 7338 - 7347.

[13] Wang, Y. and Z. Lian, DeepVecFont: Synthesizing high-quality vector fonts via dual-modality learning. ACM Transactions on Graphics (TOG), 2021. 40 (6): p. 1 - 15.

[14] Carlier, A., et al., Deepsvg: A hierarchical generative network for vector graphics animation. Advances in Neural Information Processing Systems, 2020. 33: p. 16351 - 16361.

[15] Liao, Y., S. Donne, and A. Geiger. Deep marching cubes: Learning explicit surface representations. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[16] Jones, R.K., et al., Shape assembly: Learning to generate programs for 3d shape structure synthesis. ACM Transactions on Graphics (TOG), 2020. 39 (6): p. 1 - 20.

[17] Achlioptas, P., et al. Learning representations and generative models for 3d point clouds. in international conference on machine learning. 2018. PMLR.

[18] Groueix, T., et al., Atlasnet: A papier-mâché approach to learning 3d surface generation. arxiv 2018. arXiv preprint arXiv:1802.05384, 1802.

[19] Chen, Z. and H. Zhang. Learning implicit fields for generative shape modeling. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[20] Wu, R., C. Xiao, and C. Zheng. Deepcad: A deep generative network for computer-aided design models. in Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[21] Lambourne, J.G., et al. Brepnet: A topological message passing system for solid models. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[22] Girdhar, R., et al. Learning a predictable and generative vector representation for objects. in Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11 - 14, 2016, Proceedings, Part VI 14. 2016. Springer.

[23] Hochreiter, S. and J. Schmid Huber, long short-term memory. Neural Comput, 1997. 9 (8): p. 1735 - 80.

[24] Li, T.-M., et al., Differentiable vector graphics rasterization for editing and learning. ACM Transactions on Graphics (TOG), 2020. 39 (6): p. 1 - 15.

[25] Lorensen, W.E. and H.E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, in Seminal graphics: pioneering efforts that shaped the field. 1998. p. 347 - 353.

[26] Vaswani, A., et al., Attention is all you need. Advances in neural information processing systems, 2017. 30.

[27] Mo, K., et al. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.