Skinning Vector Graphics with GANs

Ankit Phogat
Adobe
phogat@adobe.com

Matthew Fisher
Adobe
matfishe@adobe.com

Danny M. Kaufman
Adobe
kaufman@adobe.com

Vineet Batra
Adobe
vbatra@adobe.com

Figure 1: Multiple poses of dancing couple created form initial pose *(left most)*. Resolution independent, can be zoomed into.

ABSTRACT

We propose a novel method for editing vector graphics which enables users to intuitively modify complex Bézier geometry. Our method uses a Generative Adversarial Network (GAN) to automatically predict salient points for any arbitrary geometry defined by cubic Bézier curves, which are used as *handle* locations for a Linear Blend Skinning transformation. Further, we bind input geometry to a triangle mesh, to decouple the complexity of input geometry from mesh topology. Finally, to reconstruct Bézier curves from the transformed mesh, we formulate a linear optimization problem and solve it in performant manner to ensure real time feedback, without increasing the number of Bézier segments.

CCS CONCEPTS

• **[Computer Graphics]: Computational Geometry and Object Modeling**;

KEYWORDS

Free-form Deformation, Vector Graphics, Linear Blend Skinning

SIGGRAPH '19 Posters, July 28 - August 01, 2019, Los Angeles, CA, USA

ACM ISBN 978-1-4503-6314-3/19/07.
https://doi.org/10.1145/3306214.3338544

ACM Reference format:
Ankit Phogat, Matthew Fisher, Danny M. Kaufman, and Vineet Batra. 2019. Skinning Vector Graphics with GANs. In *Proceedings of SIGGRAPH '19 Posters, Los Angeles, CA, USA, July 28 - August 01, 2019,* 2 pages.
https://doi.org/10.1145/3306214.3338544

1 INTRODUCTION

Traditional vector graphics editing is based on piece-wise modification of cubic Bézier segments using control handles. This is a tedious workflow, demanding high level of expertise in order to achieve desired results. Paradigms such as Linear Blend Skinning [Liu et al. 2014] demonstrate significant potential to provide intuitive method for editing vector graphics. However, these techniques still present a steep learning curve, as users are required to plot handles manually. We build upon advancements in deep learning, specifically GANs[Goodfellow et al. 2014] [Isola et al. 2016], and propose a model to automatically predict LBS handle locations, thereby eliminating the learning curve associated with plotting handles, and enabling users to start interacting right away. A method to adapt LBS for Bézier geometry was given by [Liu et al. 2014], wherein the Bézier curves are sampled to generate mesh vertices. However, this lacks robustness required for real world vector graphics, as generated mesh[1] is overtly complex and system fails when solving for biharmonic weights [Jacobson et al. 2011]. Furthermore, non-linear optimization step for curves-fitting makes it inadequate for real

[1]Mesh generation fails for complex Bézier geometry, e.g 'Elephant' in result section

(a) (b) (c)

Figure 3: (a) Vector graphic (*445 curves*); (b) mesh using [Liu et al. 2014](*39,066 triangles*); our method (*494 triangles*) (c)

time performance. We propose an improved method, which creates a simple mesh using only the boundary of input curves [Figure 3]. As a result, our method can handle arbitrarily complex geometry. Finally, we propose an efficient and accurate method for re-fitting Bézier curves to transformed samples using a linear solve.

2 OUR APPROACH

A user starts interaction by clicking on the input vector graphic, at which point it is rasterized to grayscale and passed as input to a trained model to predict handle locations. The predicted handle locations, along with outline of the grayscale raster are used to generate a triangle mesh representation. The user can then move these handles to deform the mesh at which point the Bézier curves are refitted as per the deformed mesh.

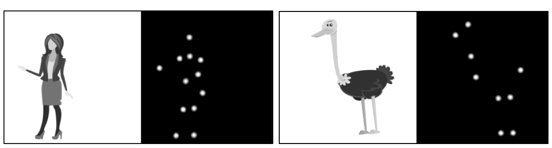


Figure 4: Grayscale images with predicted handles locations

2.1 Predicting Handle Locations

We formulate this as an image translation problem using Conditional GAN [Isola et al. 2016]; given a grayscale image, translate it to set of corresponding handle locations. [Figure 4]

2.1.1 Generator Architecture. The generator is an encoder decoder architecture and consists of:

- **Encoder** C64-C128-C256-C512-C512-C512-C512
- **Decoder** C512-C512-C512-C512-C256-C128-C64

Skip connections [Ronneberger et al. 2015] are used in generator.

2.1.2 Discriminator Architecture. The discriminator architecture is C64-C128-C256-C512, with 4x4 convolution filters and stride 2.

2.1.3 Losses.

- **Pixel Loss (L_{pix}):** Average per-pixel L_2 norm of difference between generated image and ground-truth.
- **Adversarial Loss (L_{adv})** is defined as:

$$L_{adv} = -\sum_i \log \mathscr{D}_\phi(\mathscr{G}_\theta(x_i)) \quad (1)$$

- **Perceptual Loss (L_p):** L_2 difference in a feature space, extracted from pre-trained VGG-19 (layers 2, 7 and 12).

Our final objective function becomes:

$$\mathscr{L} = w_{pix}L_{pix} + w_{adv}L_{adv} + w_pL_p \quad (2)$$

2.2 Mesh-Curve Binding

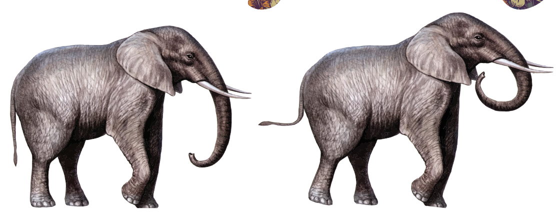
The next step is to compute the representative boundary or outline from the set of input Bézier curves. This polyline, along with predicted handle locations, is used to generate a triangle mesh using Constrained Delaunay Triangulation(CDT). This mesh is iteratively refined when solving for weights to ensure smooth distribution. Béizer curves are then adaptively sampled and for each sample we compute index of the containing triangle and barycentric coordinates within the triangle. A Bounding volume hierarchy (BVH) structure, using rectangles as bounding volumes, is used to accelerate this lookup.

2.3 Refitting Bézier Curves

Finally, we refit Bézier segments to the transformed mesh vertices and embedded samples subject to continuity constraints inferred from the input curves, and maintaining the regularity of geometric shapes. We use a two step approach to fit curves; first, Newton-Rhapson is used for an initial fit and re-parametrization of curve samples, which we then use to perform a LSQ solve (a milder form of G1 is used to keep it linear) to obtain final fitted curves. This formulation causes the resulting curve fitting to be highly performant and robust at the same time.

3 RESULTS

The following figures show the initial vector graphic and modified results using proposed method.



REFERENCES

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems.*

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2016. Image-to-Image Translation with Conditional Adversarial Networks. *arxiv* (2016).

Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine. 2011. Bounded Biharmonic Weights for Real-time Deformation. *ACM Trans. Graph.* (2011). https://doi.org/10.1145/2010324.1964973

Songrun Liu, Alec Jacobson, and Yotam Gingold. 2014. Skinning Cubic Bézier Splines and Catmull-Clark Subdivision Surfaces. *ACM Transactions on Graphics (TOG)* (2014).

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. (2015). arXiv:1505.04597