General Primitives for Smooth Coloring of Vector Graphics

Vineet Batra
Adobe Systems
vbatra@adobe.com

Ankit Phogat
Adobe Systems
phogat@adobe.com

Mridul Kavidayal
Adobe Systems
kavidaya@adobe.com

Figure 1: A richly colored vector graphic illustration created using our method.

ABSTRACT

We propose a novel and intuitive method for coloring vector graphics which is easy to use and creates richly colored artwork with very little effort. Further, it preserves the underlying geometry of the vector graphic primitives, thereby, making it easy to perform subsequent edits. Our method builds upon the concepts of shape-coverage, color and opacity and thus is applicable to all vector graphics constructs including non-convex paths and text. Furthermore, our method is highly performant and provides real-time results irrespective of the number of coloring primitives used.

CCS CONCEPTS

• **Computing methodologies** → *Image-based rendering*;

KEYWORDS

Diffusion Coloring, Vector Graphics, Bilaplacian, Gradients

ACM Reference format:
Vineet Batra, Ankit Phogat, and Mridul Kavidayal. 2018. General Primitives for Smooth Coloring of Vector Graphics. In *Proceedings of SIGGRAPH '18 Posters, Vancouver, BC, Canada, August 12-16, 2018,* 2 pages.
https://doi.org/10.1145/3230744.3230786

SIGGRAPH '18 Posters, August 12-16, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5817-0/18/08.
https://doi.org/10.1145/3230744.3230786

1 INTRODUCTION

The simplest form of coloring 2D vector graphics is solid fill, which applies a constant color to all the pixels of the input primitive. For richer coloring, linear and radial gradients are commonly used in mainstream 2D vector graphic editing applications. In addition, gradient meshes are also available in some applications but are rarely used in practice due to complexity in creating and manipulating such meshes. Orzan et al. [Orzan et al. 2008] proposed a new vector-based primitive for creating smooth shaded images, called Diffusion Curves, by modelling coloring as a solution to the Poisson's equation over the entire plane. Their system computes analytical solution for each pixel and thus performance is dependent on both the number of curves and display density of the underlying surface; also, the resulting output is a raster image. Building upon this, Boye et al. [Boyé et al. 2012] introduce a vectorial solver based on Finite Element Method by modelling the intermediate representation as 'special quadratic patches'. However, their resulting mesh has to be converted to a raster to be used in different applications as this specialized mesh is not a commonly supported vector graphics construct; it also requires recomputation on change in viewing aperture. Recently, Sun et al. [Sun et al. 2014] proposed a new algorithm for random-access evaluation of diffusion curves and claim a constant-time solution in number of pixels but complexity is dependent on the number of input curves. Further, use of a lattice based structure inherently limits application to convex paths. In all these methods, the coloring primitives and underlying geometry are closely coupled, thereby making it difficult to predict and control the generated result. Furthermore, lack of a truly robust

and resolution independent method has limited their adaptability in user-facing applications.

We present a method for creating richly colored graphics artwork which overcomes aforementioned problems and offers a unified method to produce resolution independent output and can be used in all workflows. We model coloring as a solution to bilaplacian equation over the surface of a triangle mesh generated to represent the shape of the input primitive. Primarily, we propose a method that : a) is intuitive to use yet offers precise control on the quality of output and can generate photorealistic artwork b) is a unified model combining all color primitives- point, line and curve handles and supports opacity as well c) is highly performant for real-time feedback d) provides resolution independent rendering e) seamlessly applies to all vector graphics objects eg. path, text etc.

2 OUR APPROACH

The premise of our approach is separation of color and geometry primitives as it significantly enhances the predictability of output and ease of use. Both color and geometry can be modified independently of each other thereby making it easier for iterative refinement till desired results are achieved. Furthermore, we use a triangle mesh to model the generated colouring which can be directly used in most applications.

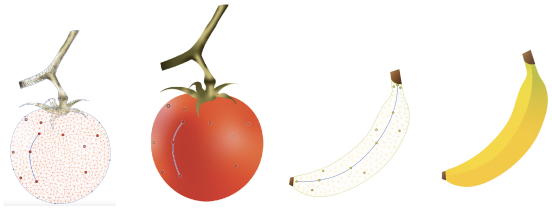

Figure 2: Figure (a) shows separate color and geometry primitives as well as the underlying mesh. Figure (c) depicts a barrier color curve. Figures (b) and (d) are resultant colorings

For intuitive control, a single color is assigned to a point on the curve which is diffused on both sides without need for post processing (blur)[Boyé et al. 2012; Orzan et al. 2008; Sun et al. 2014]. Thus, we formulate this problem as assigning colors to mesh vertices by solving the bilaplacian equation, using input color primitives to initialize boundary conditions, thereby ensuring smoothness of color across input shape. In addition, support for sharp boundaries is also provided in the form of 'barrier curves' by introducing a tear in mesh Figure [2c].

2.1 Mesh Generation

Input geometry is planarized and then an offset is applied for robustness, so that boundary curves do not get over-flattened during sampling. Points are adaptively sampled along the input boundary and color primitives are added as vertices and edges, to generate a triangle mesh using Conforming Delaunay Triangulation(CDT). A clipping path, identical to input boundary is applied to the triangle mesh, ensuring smooth edges.

2.2 Formulation

We model this as *per-color-channel* solution to the bilaplacian equation.

$$\Delta^2 w_i = 0 \tag{1}$$

w_i denotes the variable representing the weights of the i^{th} color component over all mesh vertices in the domain ($\mathcal{D}$). The final color at a mesh vertex $p \in \mathcal{D}$ is then an aggregation of all color components in the color space. For example, in RGBA color space, resultant color over the entire domain ($\mathcal{D}$) is defined as :

$$(w_R(p), w_G(p), w_B(p), w_\alpha(p)) \tag{2}$$

Equation (1) is equivalent to minimizing the Laplace energy:

$$\min_{w_i, i=1,\ldots,n} \sum_{i=1}^{n} \frac{1}{2} \int_{\mathcal{D}} (\Delta w_i)^2 dA \tag{3}$$

where n is the number of components in the color space(e.g. CMYK: 4) and an additional weight corresponding to the alpha component if present. Laplacian energy is minimized subject to boundary conditions which are added as set of linear constraints derived from the point, line and curve color primitives. Their corresponding strengths are used to establish boundary conditions on their neighbouring vertices.

2.3 Solver

Recalling huge literature on solving the bilaplacian equation using FEM is out of the scope of this article. We base our solver on Jacobson et al.[Jacobson et al. 2010]. Once weights are calculated for each vertex of the mesh, final color at the vertex is weighted combination of color components in the input color space.

3 RESULTS

Our method unifies all coloring primitives- linear, radial and curvilinear gradients. Further, user has the ability to finely control diffusion using higher order constraints which are added in the form of Neumann boundary conditions to the bilaplacian solve.

Figure 3: Artworks created using our model

REFERENCES

Simon Boyé, Pascal Barla, and Gaël Guennebaud. 2012. A Vectorial Solver for Free-form Vector Gradients. *ACM Trans. Graph.* 31, 6, Article 173 (Nov. 2012), 9 pages. https://doi.org/10.1145/2366145.2366192

Alec Jacobson, Elif Tosun, Olga Sorkine, and Denis Zorin. 2010. Mixed finite elements for variational surface modeling. In *Computer graphics forum*, Vol. 29. Wiley Online Library, 1565–1574.

Alexandrina Orzan, Adrien Bousseau, Holger Winnemöller, Pascal Barla, Joëlle Thollot, and David Salesin. 2008. Diffusion Curves: A Vector Representation for Smooth-shaded Images. *ACM Trans. Graph.* 27, 3, Article 92 (Aug. 2008), 8 pages. https://doi.org/10.1145/1360612.1360691

Timothy Sun, Papoj Thamjaroenporn, and Changxi Zheng. 2014. Fast Multipole Representation of Diffusion Curves and Points. *ACM Trans. Graph.* 33, 4, Article 53 (July 2014), 12 pages. https://doi.org/10.1145/2601097.2601187