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