A Hybrid Boundary Element and Boundary Integral Equation Method for Accurate Diffusion Curves

A Hybrid Boundary Element and Boundary Integral Equation Method for Accurate Diffusion Curves

Seungbae Bang
University of Toronto
Toronto, Canada
Amazon
Santa Clara, USA
seungbae@cs.toronto.edu

Kirill Serkh
University of Toronto
Toronto, Canada
kserkh@math.toronto.edu

Oded Stein
Columbia University
New York, USA
MIT
Cambridge, USA
ostein@mit.edu

Alec Jacobson
University of Toronto
Toronto, Canada
Adobe Research
Toronto, Canada
jacobson@cs.toronto.edu

ABSTRACT

In theory, diffusion curves promise complex color gradations for infinite-resolution vector graphics. In practice, existing realizations suffer from poor scaling, discretization artifacts, or insufficient support for rich boundary conditions. In this paper, we utilize the boundary integral equation method to accurately and efficiently solve the underlying partial differential equation.

CCS CONCEPTS

• **Computing methodologies** → **Rasterization**.

KEYWORDS

vector graphics, diffusion curves, boundary element method, boundary integral equation method

ACM Reference Format:
Seungbae Bang, Kirill Serkh, Oded Stein, and Alec Jacobson. 2022. A Hybrid Boundary Element and Boundary Integral Equation Method for Accurate Diffusion Curves . In *SIGGRAPH Asia 2022 Technical Communications (SA '22 Technical Communications), December 6–9, 2022, Daegu, Republic of Korea.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3550340.3564233

Figure 1: The original finite-difference method of Orzan et al. [2008] exhibits inaccuracies (left), e.g., around the eye, compared to our accurate hybrid, boundary-only method (right).

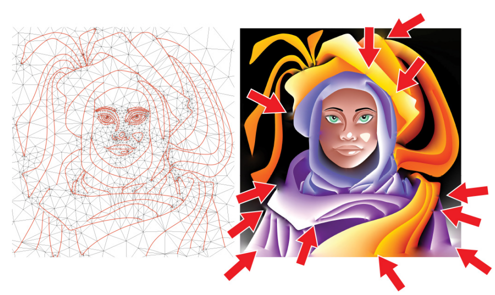

Figure 2: Bleeding artifacts are pervasive in the FEM results of TriWild [Hu et al. 2019], as shown in their Figure 12 (© Yixin Hu), reproduced here with arrows added.

1 INTRODUCTION

Diffusion curves are primitives for smoothly interpolating color data in vector graphics images, where the continuous color data is defined to be the solution to Laplace's equation with boundary values specified along vector graphics curves. Laplace's equation is the prototypical elliptic partial differential equation (PDE), and at first glance it would appear that any numerical method for elliptic PDEs could potentially be used to solve it, such as finite differences, finite elements, boundary elements or random walks. Unfortunately, in practice, diffusion curves present a number of complications which cause problems in many existing numerical methods.

Finite difference-based diffusion curve methods [Orzan et al. 2008] rely on lossy rasterization of boundary data onto a fixed pixel grid, which may either be too dense (and slow) or too coarse (and inaccurate and aliased) for a desired display resolution. Finite element methods similarly commit to a fixed, albeit adaptive, grid resolution which simultaneously determines the solution accuracy, solution smoothness, and boundary curve fidelity.

SA '22 Technical Communications, December 6–9, 2022, Daegu, Republic of Korea

ACM ISBN 978-1-4503-9465-9/22/12…$15.00
https://doi.org/10.1145/3550340.3564233

An alternative to discretizing the entire image domain is to employ boundary-only methods, where the color value at every point can be computed from calculations performed on the boundary alone. The boundary element method (BEM) discretizes only the boundary using boundary elements, and can then evaluate the solution at any point in the domain after a precomputation step which involves solving an integral equation. BEM, however, still requires discretization of the boundary into line segments, which can lead to resolution problems at the boundary, similar to those encountered in linear FEM.

We propose a boundary-only method which not represent the solution on line segments approximating the boundary geometry. Instead, we sample directly from the exact spline representation of boundary curves using the boundary integral equation method (BIEM), and solve the associated integral equation in a way that allows us to color pixels at an arbitrary resolution. To evaluate the color data, we interpolate our smooth BIEM solution to a BEM discretization.

2 RELATED WORKS

When Diffusion Curves (DCs) were first introduced, Orzan et al. [2008] solved Laplace's equation using the Finite Difference (FD) Method. Despite its strengths of simplicity and easy parallelization, rasterizing the input curve to a pixel domain can lead to inaccurate results, as shown in Fig. 1 (left).

To overcome the problems of the FD method, the Finite Element Method (FEM) was employed to evaluate DCs [Boyé et al. 2012; Pang et al. 2011] since FEM can more precisely represent boundary geometry using constrained triangulation along curves. While the boundary can indeed be better represented, triangulation itself can become burden if the input curves are too numerous or have complex shapes. Using the powerful triangulation tool TriWild [Hu et al. 2019], we could not successfully generate a triangulation of example Fig. 1 with sufficient detail preserved. Even if triangulation succeeds, FEM still suffers from bleeding artifacts if the triangulation is not dense enough, as shown in Fig. 2.

The Boundary Element Method (BEM) [Sun et al. 2012] can be used to avoid triangulation by only discretizing boundary curves and re-formulating the problem as an integral equation. However, BEM still suffers from visible polyline discretization, as shown in Fig. 4 (left).

Diffusion curves can also be evaluated using stochastic methods. The fully meshless Walk on Spheres (WoS) [Sawhney and Crane 2020], on the other hand, does diffuse colors around obstacles. However, WoS has difficulties with Neumann boundary conditions, and this turns out to be a major limitation, since such boundary conditions turn out to be exceedingly useful in practice. For complicated collections of input curves, it is difficult to specify Dirichlet boundary conditions on every single curve. By specifying a zero Neumann boundary condition on a majority of the input curves, one only needs to specify Dirichlet boundary conditions on a small subset of curves to create a smooth and natural color interpolation on the domain, as shown in Fig. 3.

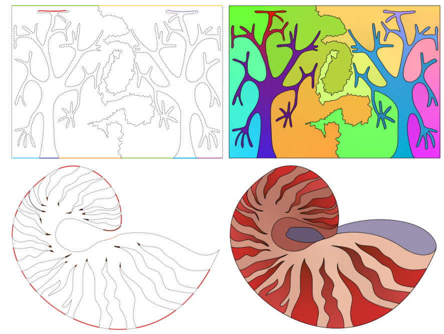

Figure 3: Input diffusion curves, with Dirichlet boundary conditions on colored curves and zero Neumann boundary conditions on dotted curves (left), and its solution (right) (top) and an example of a Nautilus shell (bottom). For the Nautilus shell example, a solid color inner shell region is overlayed.

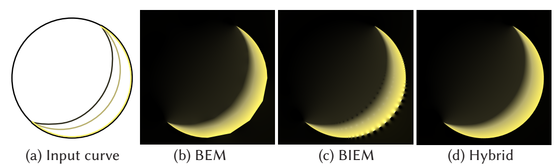

Figure 4: Result comparisons between BEM, BIEM, and our Hybrid method. BEM suffers from visible polyline, and BIEM shows dotted-looking artifacts near the boundary, whereas our method is free from both problems. (Note that we choose to have coarse discretization for clear distinction)

3 BOUNDARY VALUE PROBLEM

We consider the model problem of Laplace's equation on region $V \subset \mathbb{R}^2$ with a simple, closed boundary S:

$$\Delta u = 0 \text{ on } V, \quad \text{subject to one of} \quad \begin{cases} u = u^* \text{ on } S, \\ \frac{\partial u}{\partial n} = \psi^* \text{ on } S. \end{cases} \tag{1}$$

We discuss three approaches to solving this problem: the boundary element method (BEM), the boundary integral equation method (BIEM), and our newly proposed hybrid of BEM and BIEM.

3.1 Boundary Integral Equation

Both the BEM and the BIEM reformulate the underlying PDE over the volume V as boundary integral equations over the boundary S. The key idea is to use a representation involving the free-space Green's function, which ensures that the candidate solution always satisfies the PDE. The Green's function for Laplace's equation in two-dimensional Euclidean space, as well as its directional derivative, are well known to be

$$G(p, q) = -\frac{\log(\|p - q\|)}{2\pi} \tag{2}$$

3.1.1 Integral Equation. Using the free space Green's function, we can convert the boundary value problem Eq. 1 into its Boundary Integral Equation (BIE) formulation. Consider the so-called single layer potential, which represents our candidate solution u as an integral of the Green's function over a boundary density σ:

$$u(x) = \int_S G(p,x)\sigma(p)dS(p), \quad \forall x \in V. \tag{3}$$

Letting x approach the boundary S, we obtain the following BIE, which we can solve for the unknown density $\sigma(p)$ on the boundary given Dirichet boundary values $u^*(q)$:

$$u^*(q) = \int_S G(p,q)\sigma(p)dS(p), \quad \forall q \in S. \tag{4}$$

3.2 Boundary Element Method

We can apply the boundary element method to discretize BIEs in order to solve them numerically. Suppose, without any loss of generality, that the boundary consists of a single curve S. We begin by discretizing S into line segments $\overline{S}_j$. Then we assume that the density value σ_j is constant on each line segment. The Integral equation 4 can be expressed as:

$$u^*(q) = \sum_{j=1}^{s} \int_{\overline{S}_j} G(p,q)dS(p)\,\sigma_j, \tag{5}$$

where s is the number of boundary elements, and the integrals $\int_{\overline{S}_j} G(p,q)dS(p)$ are computed analytically using well-known formulas that depend on $\overline{S}_j$ being a line segment. We have s unknowns σ_j, and so we need at least s equations to determine a unique solution.

In matrix form, this system of equations is

$$\overline{\mathbf{u}}^* = \overline{\mathbf{G}}\overline{\boldsymbol{\sigma}}, \tag{6}$$

3.3 Boundary Integral Equation Method

The Boundary Integral Equation Method (BIEM) can accurately represent continuous functions defined on curved boundaries without any lossy approximations to the boundary geometry, in contrast to how BEM approximates S with linear segments. Functions are represented using carefully chosen discretizations based on quadrature formulas, and are interpolated by mapping their values at the discretization points to the coefficients of spectral expansions. The rapid convergence of quadrature-based approximations means that functions can be represented with minimal loss of accuracy.

The integral equation 4 can be written as a system of equations by discretizing the boundary data at Gauss-Legendre nodes:

$$u_i^* = \int_S G(p,q_i)\sigma(p)dS(p), \tag{7}$$

where, without loss of generality, we assume the geometric boundary curve is given by a function $\gamma(t)\colon [-1,1] \to \mathbb{R}^2$, $q_i = \gamma(t_i)$ are the sampled Gauss-Legendre quadrature points, and $u_i^* = u^*(q_i)$.

Ultimately, this procedure can be written in matrix form:

$$\mathring{\mathbf{u}}^* = \mathring{\tilde{\mathbf{G}}}\mathring{\boldsymbol{\sigma}}, \tag{8}$$

4 HYBRID METHOD

Our proposed method combines the advantages of the BEM and BIEM approaches into a hybrid technique.

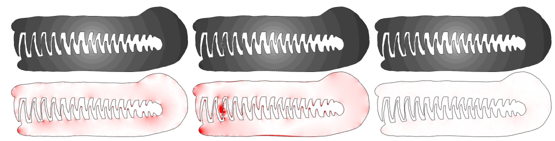

Figure 5: Accuracy comparison between BEM, BIEM, and our Hybrid method. The boundary values are constructed by placing single source Green's function in the middle of the figure. The solutions should exactly match the potential induced by that Green's function. The bottom row shows the error compared to the ground truth, highlighted in red color.

4.1 Comparison between BEM and BIEM

BEM has the limitation that the number of degrees of freedom representing the piecewise constant density σ is bounded by the number of elements in the spatial discretization of the boundary curves. BIEM is free from this limitation, and the number of degrees of freedom in the representation of the continuous density σ is decoupled from the number of quadrature points used for *evaluation*. On the other hand, BIEM has the limitation that it is inaccurate when curves are close-to-touching in the *solution* stage, and has artifacts in the induced potential near the quadrature points in the *evaluation* stage. BEM, however, is free from both of these problems, since it uses analytic integration along line segments.

4.2 Combination of BEM and BIEM

We propose to combine these two methods, inheriting the strengths of both. We discretize both the solution σ and the boundary data u^* at Gauss-Legendre nodes, as in BIEM. However, we also introduce the BEM in two places. In order to evaluate integrals of the form Eq. 3 in the *solution* stage, we interpolate the density to a BEM-like approximation, which corrects the shortcoming of BIEM for close-to-touching curves. Once we have solved for the solution $\mathring{\boldsymbol{\sigma}}$ at the quadrature nodes, we evaluate the potential by once again interpolating to a BEM-like approximation, which corrects the shortcoming of BIEM with respect to artifacts in the induced potential.

We begin by discretizing the boundary data at Gauss-Legendre nodes. We then discretize the boundary curve S into s line segments $\overline{S}_j$. If the density values $\overline{\boldsymbol{\sigma}} \in \mathbb{R}^s$ on these line segments are known, then we can write Eq. 4 as

$$u^*(q_i) = \sum_{j}^{s} \int_{\overline{S}_j} G(p,q_i)dS_j(p)\sigma_j, \quad \text{for each quadrature point } i. \tag{9}$$

In matrix form:

$$\mathring{\mathbf{u}}^* = \mathring{\overline{\mathbf{G}}}\overline{\boldsymbol{\sigma}}, \tag{10}$$

Where $\mathring{\mathbf{u}} \in \mathbb{R}^g$ are given boundary value on quadrature points, $\overline{\boldsymbol{\sigma}} \in \mathbb{R}^s$ are density value on line segments of boundary, and $\mathring{\overline{\mathbf{G}}} \in \mathbb{R}^{g\times s}$.

Since we choose to discretize the solution σ at Gauss-Legendre nodes like in the BIEM, we recover the density values $\overline{\boldsymbol{\sigma}}$ by using Legendre polynomial interpolation. Computing the coefficients of the Legendre expansion of σ by $\mathbf{c} = \mathring{\mathbf{P}}^{-1}\mathring{\boldsymbol{\sigma}}$, we can evaluate the density value on the midpoint of each line segment $\overline{S}_j$ by the formula $\overline{\sigma} = \overline{\mathbf{P}}\mathbf{c}$,where $\overline{\mathbf{P}} \in \mathbb{R}^{s\times g}$ is the Legendre interpolation matrix constructed by evaluating the Legendre polynomials at $\overline{\mathbf{t}}$, which is a vector of curve parameter values corresponding to the midpoints of the line segments $\overline{S}_j$. Hence, we have the relation $\overline{\sigma} = \overline{\mathbf{P}}\mathring{\mathbf{P}}^{-1}\mathring{\sigma}$.

We can thus express our system in matrix form in terms of $\mathring{\sigma}$ as:

$$\mathring{\mathbf{u}}^* = \underbrace{\mathring{\overline{\mathbf{G}}}\,\overline{\mathbf{P}}\mathring{\mathbf{P}}^{-1}}_{\mathring{\mathbf{G}}_\mathrm{H}}\,\mathring{\sigma}, \tag{11}$$

where $\mathring{\mathbf{G}}_\mathrm{H} \in \mathbb{R}^{g\times g}$. In order for $\mathring{\mathbf{G}}_\mathrm{H}$ to have full rank, the number of quadrature points g must be $\leq$ the number of line segments s. Note that, regardless of the size of s, the dimensionality of the system is $g \times g$. This is beneficial for us, as the matrix that needs to be inverted is much smaller than the corresponding matrix for BEM, $\overline{\mathbf{G}} \in \mathbb{R}^{s\times s}$.

Once we solve the system, we have, by Legendre polynomial interpolation, a density value $\sigma(p)$ that can be evaluated anywhere on the curve. At the *evaluation* stage, we employ the BEM-like approach and now we can use an arbitrary number of line segments e, that is independent both of the number line segments s used at *solution* stage and the number of quadrature points g used to represent the solution.

Table 1: Computation time comparison between BEM solve and Hybrid. Once solved, evaluation step become identical if the number of segments are set equal.

	curves	BEM solve	Hybrid solve	eval
cherry	32	0.10s	0.008s	13.9s
red pepper	109	1.47s	0.079s	64.7s
person with purple cloak	326	32.7s	0.831s	326.7s

5 FAST MULTIPOLE METHOD

Consider the evaluation of the BEM integrals over m diffusion curves, for a total of $N = ms$ BEM line segments. Directly evaluating the BEM integrals at the midpoints of all N BEM line segments would require $O(N^2)$ operations. Greengard and Roklin [Greengard and Rokhlin 1987] demonstrated that the task could be done in $O(N)$ operations in finite precision by introducing the Fast Multipole Method (FMM). We employed FMM to accelerate the evaluation of the BEM integrals.

6 RESULTS

We have implemented the main algorithm of our method in C++ with [Jacobson et al. 2018].

Our method is free from both the visible polyline discretization problem of BEM for a system of the same size, and also from the artifacts around quadrature points that are found in BIEM (see Fig. 4). Our method shows the most accurate results when the number of degrees of freedom in the solution stage and the evaluation stage are both kept fixed (see Fig. 5).

Table 1 shows the computation time compared with BEM an our hybrid method. Note the computation time is without FMM.

Figure 6: Results generated with 4K resolution. Blur scalar field was computed with Diffusion curve and applied as post process (Best viewed in a high-resolution digital screen)

All the timings are computed on a MacBook Pro laptop with an Intel 2.4GHz Quad-Core i9 Processor and 16GB RAM. Fig. 6 shows high resolution image generated using our hybrid method + FMM. Diffusion curve data are from [Orzan et al. 2008] and [Liu 2009] .

7 CONCLUSIONS AND FUTURE WORKS

Our proposed method brings the diffusion curve to a new level of representation that can truly support high resolution. Despite its many desirable features, our method still has some limitations and room for future improvements.

Our aim of ensuring accurate computations can become burdensome computationally, because messy or wild curve data will exhibit a lot of intersecting and overlapping curves, which will require heavy adaptive subdivision to resolve. We developed a preprocessing step to deal with ill-posed curves, but it is difficult to distinguish between an intentional curve placed by an artist and unintended ill-posed curves. It will be useful to have a version of our algorithm with softer and less stringent accuracy requirements, which would allow for more wild and ill-posed data.

REFERENCES

Simon Boyé, Pascal Barla, and Gael Guennebaud. 2012. A vectorial solver for free-form vector gradients. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–9.

Leslie Greengard and Vladimir Rokhlin. 1987. A fast algorithm for particle simulations. *Journal of computational physics* 73, 2 (1987), 325–348.

Yixin Hu, Teseo Schneider, Xifeng Gao, Qingnan Zhou, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2019. TriWild: robust triangulation with curve constraints. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.

Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. https://libigl.github.io/.

Yijun Liu. 2009. *Fast multipole boundary element method: theory and applications in engineering*. Cambridge university press.

Alexandrina Orzan, Adrien Bousseau, Holger Winnemöller, Pascal Barla, Joëlle Thollot, and David Salesin. 2008. Diffusion curves: a vector representation for smooth-shaded images. *ACM Transactions on Graphics (TOG)* 27, 3 (2008), 1–8.

Wai-Man Pang, Jing Qin, Michael Cohen, Pheng-Ann Heng, and Kup-Sze Choi. 2011. Fast rendering of diffusion curves with triangles. *IEEE Computer Graphics and Applications* 32, 4 (2011), 68–78.

Rohan Sawhney and Keenan Crane. 2020. Monte Carlo geometry processing: A grid-free approach to PDE-based methods on volumetric domains. *ACM Transactions on Graphics* 39, 4 (2020).

Xin Sun, Guofu Xie, Yue Dong, Stephen Lin, Weiwei Xu, Wencheng Wang, Xin Tong, and Baining Guo. 2012. Diffusion curve textures for resolution independent texture mapping. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–9.