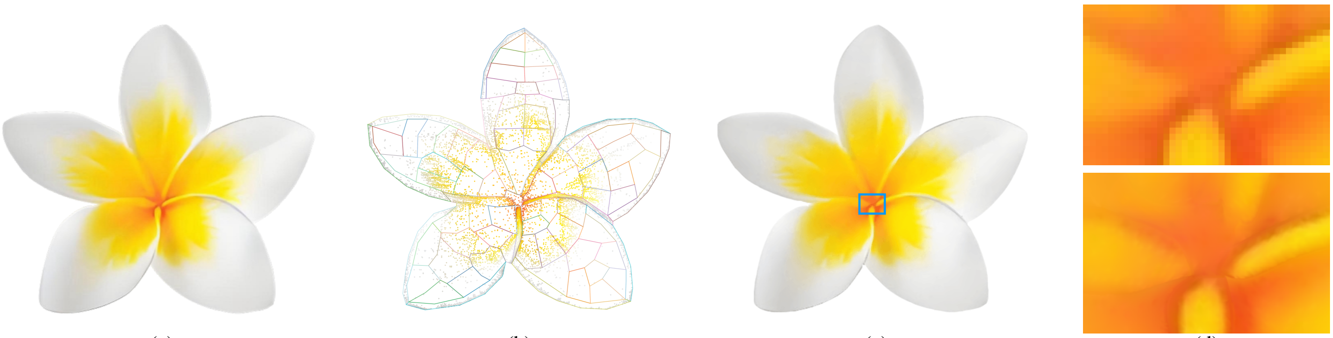
Pacific Graphics 2026
Y. He, N. Thürey, and L. Liu
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 7

Image Vectorization with Generalized Coons Patches

Yan Jing[1], Kai Hormann[2], Kaikai Qin[1], Chongyang Deng[1]†

[1]Hangzhou Dianzi University, China
[2]Università della Svizzera italiana, Switzerland



(a) (b) (c) (d)

Figure 1: *Our method converts a given raster image (a) to a vector image (c) using a refined parameter domain with adaptively inserted interior control points (b). While zooming into the raster image is prone to aliasing (d, top), the vector image can be enlarged arbitrarily without artifacts (d, bottom).*

Abstract

Patch-based image vectorization methods usually represent image regions using fixed-sided patches, such as triangular or quadrilateral patches. However, such patches are often inefficient for irregular or semantically complex regions, as many small patches are required to approximate them accurately. To solve this problem, we propose an image vectorization method based on generalized Coons patches, representing an input image as a set of shape-adaptive polygonal patches. Compared with methods based on conventional Coons patches, our method can represent images with fewer patches and substantially alleviate fold-over problems in irregular regions. To further improve the reconstruction accuracy inside each patch, an error-driven adaptive refinement mechanism is introduced. Interior control points are directly inserted at locations with large reconstruction errors, and their color attributes are determined through color fitting, without re-partitioning patches or modifying the patch topology. This mechanism is simple and flexible, and can progressively improve image reconstruction quality according to a prescribed error threshold. Experimental results demonstrate that our method achieves high-quality image reconstruction on various input images.

CCS Concepts

• ***Computing methodologies*** $\rightarrow$ ***Image processing; Parametric curve and surface models;***

1. Introduction

Raster images are the most common form of image representation and are widely used to store and transmit visual data acquired by imaging devices. They describe image content using a grid of pixels and are capable of depicting realistic scenes with rich details. However, due to their strong dependence on resolution, raster images are prone to aliasing and blurring artifacts in operations such as scaling, editing, and interactive manipulation, which limits their suitability for high-quality graphics processing. In contrast, vector images represent visual content through geometric primitives such as curves and surfaces, offering advantages in editability and resolution independence (see Fig. 1). As a result, vector images play an important role in graphic design, multimedia systems, and scal-

† Corresponding author.

able visual displays, and the conversion of raster images into vector representations with desirable geometric and color properties, known as image vectorization, has long been an active research topic [BB12, TG24].

Existing image vectorization methods have explored various representations for approximating raster images. Early approaches include polygonal decomposition and representations based on adaptive triangulation [SP06, DDI06, LL06]. Diffusion-curve-based methods represent smooth color variations through curve constraints and diffusion processes [OBW*08, JCW09]. Gradient-mesh-based methods model images using structured mesh layouts with geometry and color attributes attached to mesh vertices [SLWS07, LHM09]. Another important line of work relies on parametric surface or patch-based representations, including triangular patches, subdivision-based representations, and sparse Coons patch layouts [XLY09, LHFY12, ZZW14, HEK21, ZCX*22, CLL*20, HRK24].

These methods have improved the reconstruction quality and editability of vector images in different ways. However, for real-world images with complex color distributions and texture variations, it remains difficult to capture interior details while keeping the representation compact and well-structured. In particular, many surface- or patch-based methods rely on parameter domains with a fixed number of sides or regular structures, such as triangular patches, quadrilateral gradient meshes, or Coons patch layouts. Local approximation quality is therefore often improved by subdividing complex regions into multiple surface patches. In such representations, the expressive power within a region is closely related to the granularity of the patch layout. Capturing richer local variations usually requires more patches, which in turn increases the complexity of parameter-domain construction, patch management, and subsequent optimization. For regions with irregular shapes or highly varying colors, this preprocessing step can already be difficult. Moreover, since details are mainly controlled through patch subdivision, the modeling freedom inside a region is implicitly constrained by the parameter domain structure, making local error correction less direct and less flexible. Further improving reconstruction accuracy often requires additional refinement of the patch layout, leading to higher computational and storage costs. These limitations motivate a region-based vectorization method that can cover an image region with a single parameter domain, while introducing interior control variables to capture fine details without excessive structural complexity.

1.1. Contributions

In this paper, we propose a region-based image vectorization method based on generalized Coons patches (C^0 GC patches) to address the above challenges in modeling and representation [QZY*24]. The proposed method takes image regions as basic modeling units and constructs a parametric surface representation for each region independently, representing the entire raster image as a collection of C^0 GC patches. Unlike approaches that rely on regular parameter domains or fine-grained patch subdivision, our method adopts an arbitrary simple polygon consistent with the region topology as the parameter domain, allowing a single domain to fully cover an image region and significantly simplifying parameter domain construction and preprocessing. On this basis, interior details are adjusted by introducing interior control points. When large reconstruction errors or complex color variations occur locally, expressive power can be enhanced by adding control points at corresponding locations, without further subdividing surface patches or modifying the overall parameter domain structure. By decoupling region coverage from interior detail modeling, the proposed strategy maintains a compact structural representation while flexibly capturing complex variations within image regions.

The remainder of this paper is organized as follows. Section 2 reviews related work on image vectorization. Section 3 introduces the parametric patch-based image representation adopted in this work, based on generalized Coons patches. Section 4 presents an overview of the proposed algorithmic framework. Section 5 reports experimental results and analyzes the reconstruction performance of the proposed method. Section 6 discusses the limitations of the current approach. Finally, Section 7 concludes the paper.

2. Related Work

2.1. Curve-based methods

Curve-based vector representations constitute an important class of techniques for smoothly shaded image vectorization, among which diffusion curves are widely regarded as a representative primitive. This approach introduces a set of curves, either manually specified or automatically extracted, into the image plane and associates colors and related attributes on both sides of each curve. The continuous color variation of the image is then modeled as a diffusion process driven by these curve-based constraints, resulting in sharp structural features along the curves and smoothly shaded regions between them [OBW*08].

Several subsequent works have extended or improved this framework from different aspects, including surface detail representation [JCW09], biharmonic formulations [IKCM13], fast evaluation and rasterization [STZ14], ray-traced diffusion curves [PJSH15], automatic or inverse diffusion-curve construction from raster images [XSTN14, ZDZ18], and more general Poisson-based vector graphics formulations [HSF*20]. Other curve- or constraint-based vector graphics representations have also been proposed to provide more direct control over smooth shading effects, such as controlled thin-plate splines [FSH11] and shading curves with explicit gradient control [LTKD15].

Since both geometric and color information are explicitly attached to curve structures, diffusion-curve-based methods exhibit considerable flexibility for interactive vector graphics design and editing. However, when applied to the automatic vectorization of raster images, such representations still face several limitations. On the one hand, rendering diffusion-curve-based vector images typically requires solving partial differential equations over the entire image domain, which poses challenges in computational efficiency and numerical stability. On the other hand, color propagation based on Poisson diffusion is inherently limited in its ability to faithfully capture subtle and highly non-uniform color variations commonly present in natural images.

2.2. Patch-based methods

Patch-based image vectorization methods represent smoothly shaded images by constructing continuous parametric surfaces within image regions, aiming to balance visual fidelity, compactness, and editability. Gradient meshes are a representative formulation, where traditional gradient meshes are composed of Ferguson patches arranged on regular quadrilateral grids, with geometry and color controlled at mesh vertices. Sun et al. [SLWS07] introduced optimized gradient meshes for image vectorization, and Lai et al. [LHM09] further proposed an automatic and topology-preserving gradient mesh generation method that enables a single mesh to represent regions with holes while reducing manual intervention through sparse linear system formulations. Subsequent extensions improved the flexibility and expressive power of gradient-mesh-based representations, including mesh colors [YKH10], locally refinable gradient meshes with branching and sharp color transitions [BLHK18], topologically unrestricted gradient meshes [LKSD17], and color interpolants for polygonal gradient meshes [HBK19].

To improve topological flexibility beyond regular quadrilateral meshes, Xia et al. [XLY09] introduced a triangular Bézier-patch-based representation built on planar triangulations, combined with thin-plate spline fitting to better model complex color variations. Chen et al. [CLL*20] combined parametric patches with detail color features and real-time thin-plate spline interpolation to improve local detail representation and interactive editing. Hettinga et al. [HEK21] further employed cubic Bézier triangles to construct curved triangular meshes and used mesh colors to represent intra-patch color variation, enabling efficient rendering and improved preservation of texture details. Subsequently, TCB-splines were incorporated into the triangular parametric patch framework to enhance boundary and color control, thereby improving shading quality and visual consistency across patches [ZCX*22].

A key goal of patch-based image vectorization is to use fewer patches while maintaining good reconstruction quality and editability. Li et al. [LJH13] applied cubic mean value coordinates to adaptive gradient meshes, enabling expressive color interpolation over irregular multi-sided patches with fewer elements. However, the patch layout is derived from mesh simplification rather than semantic image regions, which limits its region-level editability. Wei et al. [WZG*19] generated feature-aligned quadrilateral meshes using direction fields derived from image characteristics, improving the alignment between patch layouts and image geometry. Building on this idea, He et al. [HRK24] proposed a sparse layout of Coons patches with explicit support for T-junctions, achieving compact representations with smooth shading quality. However, these approaches typically rely on quadrilateral parameter domains, which often require irregular regions to be further decomposed into multiple patches. In contrast, our method generalizes the Coons construction to arbitrary polygonal parameter domains, allowing patch topology to directly adapt to region shapes and enabling a more flexible representation with fewer patches for complex image layouts.

2.3. Subdivision-surface-based methods

Subdivision-surface-based image vectorization methods typically represent an image as a subdivision limit surface defined over a control triangular mesh, where smooth color transitions are obtained at a global level through repeated application of subdivision rules. Rather than relying on explicit region partitioning or parametric patch stitching, these methods adopt a unified subdivision surface as the fundamental representation, thereby offering resolution independence and higher-order smoothness in theory. Zhou et al. [ZZW14] proposed a curvilinear feature-driven subdivision surface representation, in which explicit curvilinear structures in the image are embedded into the subdivision process on a triangular mesh. On the one hand, feature edges are modeled using curve representations that are consistent with the subdivision rules, ensuring the stability of feature curves across multiple levels of refinement. On the other hand, spatial positions and color attributes are jointly modeled within a unified subdivision framework, enabling an integrated representation of smoothly varying regions and prominent structural features. While this approach emphasizes the consistency between feature modeling and subdivision rules and can effectively preserve curve shapes and their surrounding color variations, its performance is to some extent dependent on the quality of feature extraction and the initial control mesh.

From the perspective of vector image editing, Liao et al. [LHFY12] introduced a piecewise smooth subdivision-based representation that likewise adopts subdivision surfaces defined on triangular meshes as the core model, while explicitly incorporating mechanisms to represent feature discontinuities. By duplicating vertices at feature locations and introducing tear edges, the method supports color discontinuities across features while maintaining smooth transitions over larger regions. Combined with feature-preserving mesh simplification and color optimization, this representation is further extended to a multiresolution form, enabling image editing and processing at different levels of abstraction. In general, image vectorization methods based on subdivision surfaces provide clear advantages in terms of smoothness and unified representation. However, their representational complexity and final reconstruction quality are often influenced by the reliability of feature detection, mesh construction, and subdivision strategies.

3. Parametric Patch-based Image Representation

In our image vectorization framework, each image region is represented as a surface defined over a polygonal parameter domain, on which both geometric positions and color attributes are jointly modeled. To this end, we adopt the C^0 generalized Coons (C^0 GC) patches introduced by Qin et al. [QZY*24] as our basic representation primitives. Unlike triangular Bézier-patch-based methods that rely on triangular parameter domains [XLY09], or classical Coons patches that require quadrilateral parameter domains [Coo67, HRK24], C^0 GC patches admit arbitrary polygonal parameter domains. Consequently, each patch can directly conform to the shape of a region obtained from image segmentation without additional triangulation or quadrilateralization. This property significantly simplifies parameter-domain construction and reduces preprocessing complexity.

Building upon this existing theoretical formulation, we extend

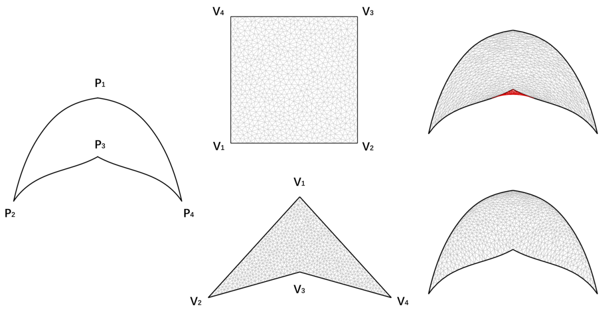


Figure 2: *For a given region $\mathcal{P}$ (left), the original Coons patch S_C (top right) is defined with respect to a square parameter domain (top center) and folds over in this example. The generalized Coons patch S_{GC} (bottom right) in (1) avoids this problem, since it uses a parameter domain Φ (bottom center) that better fits the region $\mathcal{P}$.*

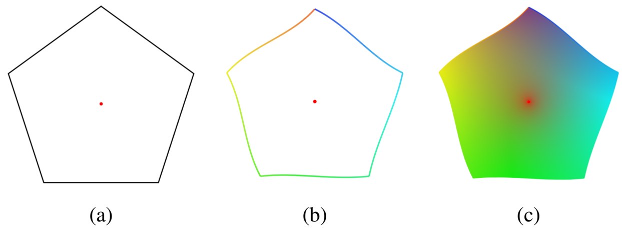


Figure 3: *We use a polygonal domain Φ with optional interior points (a) to define the boundary of an image region and the color along this boundary in terms of cubic B-splines, parameterized over the edges of Φ (b), and a C^0 GC patch $\hat{S}_{GC}$ to interpolate the boundary color as well as color given at the interior points (c).*

it to image vectorization by using generalized Coons patches to represent segmented image regions. We further propose an error-driven strategy for adaptively placing interior color control points, as described in Section 4.

For completeness, we briefly review two variants of C^0 GC patches: a boundary-only formulation without interior control points, and a formulation that uses the interior control points supported by C^0 GC patches to capture interior details.

3.1. C^0 generalized Coons patches

In the two-dimensional setting, consider a region $\mathcal{P}$ with n corner points $\mathbf{p}_1,\dots,\mathbf{p}_n$, joined by n parametric curves $C_i\colon [0,1] \to \mathbb{R}^2$ with $C_i(0) = \mathbf{p}_i$ and $C_i(1) = \mathbf{p}_{i+1}$, where indices are cyclic over the range $(1,2,\dots,n)$, that is, $\mathbf{p}_{n+1} = \mathbf{p}_1$. Moreover, let Φ be an n-sided polygonal domain homeomorphic to $\mathcal{P}$, with vertices $\mathbf{v}_1,\dots,\mathbf{v}_n$ and edges $E_i = [\mathbf{v}_i, \mathbf{v}_{i+1}]$. Then, the C^0 *generalized Coons patch* $S_{GC}\colon \Phi \to \mathbb{R}^2$ that interpolates all boundary curves is defined as

$$S_{GC} = \sum_{i=1}^{n} (\lambda_i + \lambda_{i+1}) C_i\left(\frac{\lambda_{i+1}}{\lambda_i + \lambda_{i+1}}\right) - \sum_{i=1}^{n} \lambda_i \mathbf{p}_i, \qquad (1)$$

where $\lambda_1,\dots,\lambda_n\colon \Phi \to \mathbb{R}$ are a set of generalized barycentric coordinate functions for Φ.

This construction relies on generalized barycentric coordinates (GBCs) to blend the boundary curves with vertex-related correction terms over the polygonal domain. In this work, we use harmonic coordinates [JMD*07], because they are non-negative over arbitrary polygonal domains. Under these conditions, the surface patch defined by (1) forms a convex combination of boundary-curve contributions and vertex-related terms throughout the domain. Moreover, the C^0 GC patch S_{GC} remains well-defined over concave polygonal domains, which largely avoids the fold-over issues that may arise in classical Coons patches under concave quadrilateral configurations (see Fig. 2). As a result, the representation is more stable and robust for regions with complex or irregular boundaries.

3.2. C^0 generalized Coons patches with interior control points

Building on the boundary-driven construction above, C^0 generalized Coons patches can be naturally extended from ordinary polygonal domains to more general non-manifold topologies. In the two-dimensional setting, the interior of the parameter domain may contain isolated vertices, interior linear edges, or interior faces. As long as the adopted generalized barycentric coordinates are well defined over the corresponding topology, these structures can be incorporated into C^0 GC patches in a unified manner.

This property is particularly useful for image vectorization, because the color variation inside a segmented region cannot always be accurately represented by boundary information alone. In this work, we use the interior control points supported by C^0 GC patches to increase the fitting flexibility of the patch interior. Suppose we add to the n vertices of the polygonal domain Φ another m interior control points $\mathbf{v}_{n+1},\dots,\mathbf{v}_{n+m} \in \Phi$ and corresponding points $\mathbf{p}_{n+1},\dots,\mathbf{p}_{n+m}$ in the interior of $\mathcal{P}$. The associated C^0 *GC patch with interior control points* is then defined as

$$\hat{S}_{GC} = S_{GC} + \sum_{j=1}^{m} \lambda_{n+j} \mathbf{p}_{n+j}, \qquad (2)$$

where the λ_i are again GBCs, but this time with respect to the vertices of Φ *and* the interior control points and S_{GC} is the C^0 GC patch without interior control points in (1). We continue to use harmonic coordinates in this formulation, since they can also be defined for polygons with interior points. In this form, the interior control points serve as additional constraints that force the patch $\hat{S}_{GC}$ to interpolate $\mathbf{p}_{n+j}$ at $\mathbf{v}_{n+j}$ for $j = 1,\dots,m$ without changing the boundary behaviour.

In our image vectorization method, we extend this approach by attaching not only geometric point information to the control points and edges of Φ, but also color information. Hence, the curves C_i do not only parameterize the boundary of $\mathcal{P}$, but also define a color distribution along the boundary, and the interior control points $\mathbf{v}_{n+j}$ are additionally associated with color values. Hence, $\hat{S}_{GC}$ describes not only a planar patch $\mathcal{P}$, but also interpolates the colors given along its boundary and at the interior points $\mathbf{p}_{n+j}$ of $\mathcal{P}$ (see Fig. 3). In this model, the interior control points are particularly useful for fitting regions with complex interior color distributions, as they add flexibility for representing interior details and texture variations, while the prescribed boundary curves remain strictly interpolated and boundary continuity across shared edges is preserved.

4. Algorithm Overview

In this work, C^0 GC patches are used to model each image region independently, thereby converting the input raster image into a vector representation composed of multiple parametric surface patches. As illustrated in Fig. 4, the overall pipeline of the proposed method is summarized as follows and explained in detail in the forthcoming subsections.

Fig. 4 (a) shows the input raster image. First, the image is segmented into a set of non-overlapping regions with relatively homogeneous color appearance, as shown in Fig. 4 (b). For each region, a polygonal parameter domain consistent with its topology is manually constructed to serve as the parametric space for subsequent surface construction, as illustrated in Fig. 4 (c). After the parameter domain is obtained, the region boundaries are converted into continuous parametric representations. Specifically, the polygonal boundary polylines are taken as input and approximated by cubic B-spline curves using a Progressive Iterative Approximation [LWD04] algorithm, resulting in smooth and differentiable boundary representations. Fig. 4 (d) shows the fitted B-spline boundary curves, which are used as boundary constraints in the subsequent construction of C^0 GC patches.

Along the fitted B-spline boundary curves, boundary colors are sampled to fit the color of the B-spline control points, as shown in Fig. 4 (e). Based on the fitted boundary curves and sampled boundary colors, a C^0 GC patch is constructed over the polygonal parameter domain, yielding a baseline surface that is determined solely by the boundary information. This baseline model strictly interpolates the region boundaries but has limited degrees of freedom in the interior, and its reconstruction result is shown in Fig. 4 (f).

To further enhance the expressive capability within the interior of the region, interior control points are introduced in the parameter domain. The selection of these points is driven by pixelwise reconstruction errors, and the corresponding coefficients are iteratively optimized via a least-squares formulation. Fig. 4 (g) illustrates the parameter domain after adaptive insertion of interior control points, where the points are mainly concentrated in regions with pronounced color variations.

By alternately performing error-driven interior-point insertion and coefficient optimization, the interior representation capability of the C^0 GC patch is progressively enhanced, leading to a refined surface model. Its reconstruction result is shown in Fig. 4 (h). Compared with the baseline model, the refined C^0 GC patch is able to more accurately capture the interior color distribution while strictly preserving boundary interpolation.

4.1. Region segmentation and parameter domain construction

For each input image, we first decompose it into a set of non-overlapping regions, and each region is modeled independently in the subsequent vectorization process. In the current study, region segmentation is performed manually using Computer Vision Annotation Tool [S*20]. Specifically, the image is annotated into regions with relatively homogeneous color appearance, and the resulting segmentation is used as the basis for subsequent boundary extraction, boundary fitting, and C^0 GC patch construction.

After segmentation, we construct a polygonal parameter domain for each region. At present, this step is also performed manually in an interactive manner, by selecting a sequence of vertices to form a polygon whose topology and boundary layout are compatible with those of the target region. The purpose of this polygon is not to exactly reproduce the geometric boundary of the image region, but to provide a simple and valid parameter domain for subsequent B-spline boundary fitting and C^0 GC patch construction. Once the parameter domain is determined, each polygon edge is associated with a corresponding boundary segment of the target region, thereby establishing the geometric correspondence between the parameter domain boundary and the image region boundary.

It should be noted that this work does not require parameter domains with the same number of sides to share a unified canonical shape. In principle, if multiple regions use polygonal parameter domains with the same number of sides, a single canonical polygon could be adopted as their common parameter domain. This would allow the triangulation, generalized barycentric coordinates, and related interpolation structures to be reused, thereby significantly reducing the computational cost. However, we do not adopt this strategy because fold-overs of C^0 GC patches are theoretically possible when the parameter domain is substantially mismatched with the shape of the target region. Fig. 5 shows an artificially constructed example in which a regular hexagonal parameter domain causes a local fold-over. When a fixed canonical parameter domain is used for image regions with significantly different shapes, the correspondence between the parameter domain boundary and the target boundary may increase the risk of local fold-overs. Therefore, we construct a polygonal parameter domain for each region according to its actual shape, so that the overall shape of the parameter domain is as close as possible to that of the target region. This shape-adaptive construction avoids the fold-over shown in Fig. 5 and improves the stability of the C^0 GC patch construction. We did not observe any fold-overs in our experiments.

In addition, we do not restrict the number of sides of the polygonal parameter domain. Regions with simple boundary structures can be represented by parameter domains with fewer sides, whereas regions with more complex boundary structures can be assigned parameter domains with more sides to better match their topology and geometric layout. In this work, region segmentation and parameter-domain construction are both treated as preprocessing steps. Our main focus is not on automatic region extraction or automatic parameterization, but on the subsequent C^0 GC patch representation, interior control point insertion, and reconstruction refinement. Nevertheless, our framework is compatible with other segmentation and parameter domain generation methods, and these components could be replaced by automatic alternatives in future work.

4.2. PIA-based B-spline boundary fitting

To convert discrete region boundaries into continuous, smooth, and differentiable curve constraints, we represent each region boundary using cubic B-spline curves. For each edge of the polygonal pa-

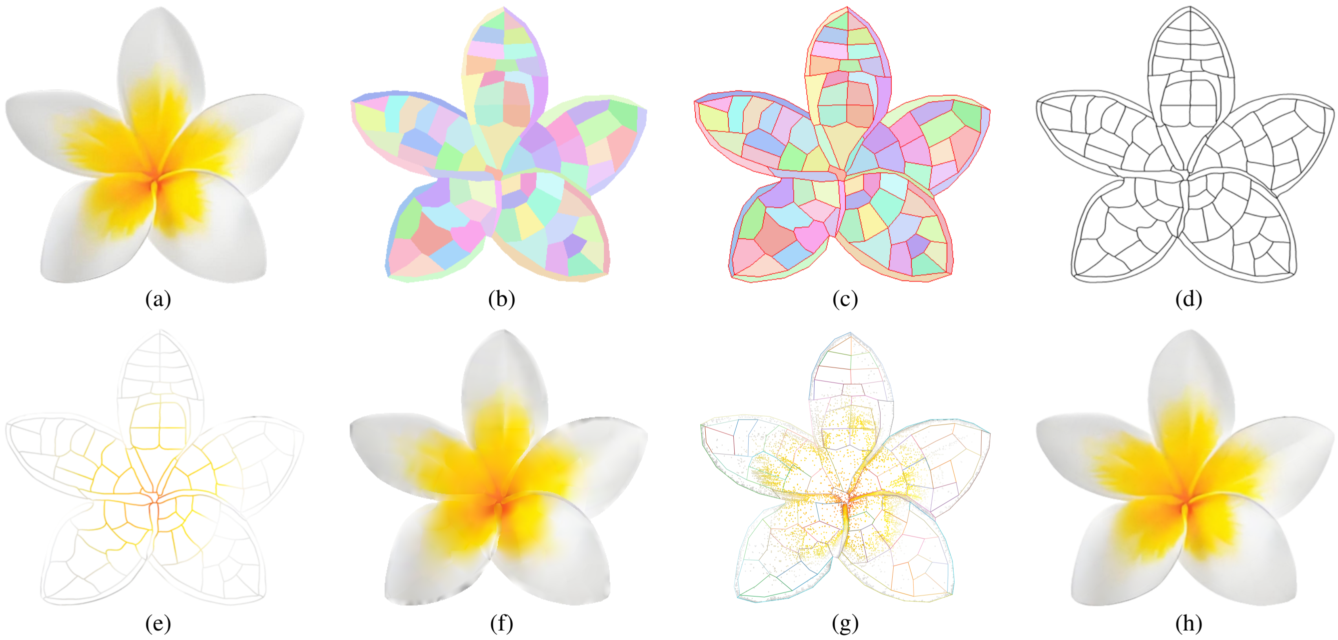


Figure 4: *Pipeline of our region-based C^0 GC patch construction with adaptive interior control points. Given an input image (a), we first segment it into regions (b) and fit a polygonal parameter domain to each segmented region (c). We then fit B-spline curves to the region contours (d) and sample the colors along those boundary curves (e). As the C^0 GC patch that interpolates this data (f) does not capture high frequent color variations inside the regions well, we augment the parameter domains by adaptively inserting interior control points via least-squares refinement (g), leading to a faithful reproduction of the input image as a C^0 GC patch with interior control points.*

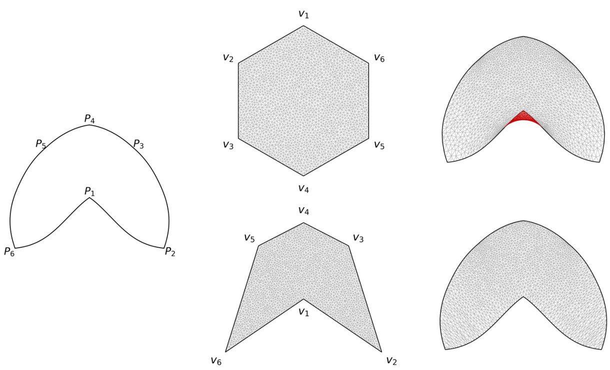


Figure 5: *Influence of the parameter domain shape on the C^0 GC patch that interpolates the region $\mathcal{P}$ (left). While a regular hexagon domain (top center) leads to a patch that folds over (top right), the patch defined with respect to a shape-adapted domain (bottom center) avoids this problem (bottom right).*

rameter domain, a sequence of pixel-level boundary samples corresponding to that edge is first extracted from the segmentation result and used as the target data for curve fitting.

Each parametric boundary curve is represented as a cubic B-spline [dB76] with m segments and defined as

$$C(u) = \sum_{i=0}^{m+2} \mathbf{P}_i N_{i,3}(u), \qquad u \in [U_3, U_{m+3}], \tag{3}$$

where $\mathbf{P}_0, \ldots, \mathbf{P}_{m+2}$ are the control points and $N_{i,3}(u)$ are the cubic B-spline basis functions over the uniform open knot vector $U = [U_0, \ldots, U_{m+6}]$. The boundary samples are mapped to the parameter domain using chord-length parameterization, establishing a correspondence between data points and curve parameters.

To achieve a stable approximation of the boundary shape while maintaining numerical robustness, we employ the Progressive Iterative Approximation algorithm to fit the B-spline control points [LWD04]. Starting from an initial estimate of the control points, PIA iteratively adjusts their positions so that the curve progressively approaches the target boundary samples. Let $C^{(k)}(u)$ denote the curve at iteration k. For each boundary sample $\mathbf{d}_j$ with corresponding parameter value u_j, the error increment at iteration k is defined as

$$\Delta\mathbf{e}_j^{(k)} = \mathbf{d}_j - C^{(k)}(u_j). \tag{4}$$

Based on these error terms, the control points are updated in an error-driven manner, which can be uniformly written as

$$\mathbf{P}_j^{(k+1)} = \mathbf{P}_j^{(k)} + \Delta\mathbf{e}_j^{(k)}, \tag{5}$$

where $\Delta\mathbf{e}_j^{(k)}$ denotes an error-driven displacement term associated

Figure 6: *Comparison of color fitting with and without offset sampling. Given an image (top left), we first detect hard (red) and soft (green) edges (top right). Without offset sampling, the C^0 GC patch without interior control points has severe artefacts (bottom left), which are resolved by our offset sampling strategy (bottom right).*

with control point $\mathbf{P}_j$, accumulated from the local fitting residuals at iteration k. Through this progressive error-driven correction process, PIA exhibits stable and smooth convergence behavior in practice [LWD04] .

To ensure geometric consistency along shared boundaries, each common boundary between adjacent regions is fitted only once and is reused by the neighboring regions. Throughout this procedure, discrete region boundaries are transformed into smooth and differentiable B-spline curves, which serve as boundary constraints for the subsequent C^0 GC patch construction.

4.3. B-spline color fitting

After obtaining the geometric B-spline boundary curves, we keep their control point positions fixed and fit RGB color attributes to the corresponding control points. Following the B-spline representation in (3), the color variation along the same boundary curve is represented using the same parameterization and basis functions as

$$c(u) = \sum_{i=0}^{m+2} \mathbf{Q}_i N_{i,p}(u), \qquad u \in [U_3, U_{m+3}], \tag{6}$$

where $\mathbf{Q}_i \in [0,1]^3$ denotes the RGB color control point associated with the geometric control point $\mathbf{P}_i$.

To handle different types of shared boundaries, we classify them into soft and hard edges according to the boundary color gradient and the mean color difference between adjacent regions. Specifically, we compute the Sobel gradient magnitude of the input image, evaluate the local average gradient at boundary samples, and use the median value as the boundary gradient strength G_e. We also compute the mean color difference D_e between the two adjacent regions. If both G_e and D_e exceed their corresponding thresholds, the edge is classified as a hard edge, otherwise as a soft edge.

For soft edges, since the colors on both sides usually vary smoothly, we sample colors directly near the boundary curve and fit one set of color control points for the shared edge. For hard edges, boundary color errors may propagate into the patch interior and cause larger fitting errors, which may require additional interior control points for compensation (see Fig. 6). Therefore, we adopt an inward offset sampling strategy: for each boundary sample, we erode the label mask of the current region and search for the nearest valid pixel inside the eroded mask as the color sample. In this case, the same geometric B-spline curve is shared, but two sets of color control points are fitted for both sides of the hard edge.

Finally, given the sampled parameters u_j and their corresponding colors $\hat{c}_j$, the color control points are determined by solving a bounded least-squares problem,

$$\min_{\mathbf{Q}_0,\ldots,\mathbf{Q}_{m+2} \in [0,1]^3} \sum_j \|c(u_j) - \hat{c}_j\|_2^2.$$

This strategy preserves smooth color transitions across soft edges and reduces the influence of boundary color errors near hard edges.

4.4. C^0 GC patch construction and refinement

This section presents a region-based modeling and refinement framework based on generalized Coons patches for improving interior reconstruction accuracy. For each segmented region, a baseline C^0 GC patch is first constructed from the prescribed boundary constraints. The model is then progressively refined by inserting interior control points in the parameter domain and optimizing their RGB values, so as to compensate for residual reconstruction errors and enhance the expressive power inside the region.

Let u be an arbitrary point in the parameter domain, and let $\hat{y}(u)$ denote the predicted color value given by the C^0 GC patch at u. When no interior control points are introduced, the prediction inside the region is entirely determined by the boundary curves and boundary control points. This baseline model strictly interpolates the region boundary while providing a smooth interior extension. However, for regions with complex interior color variations, the boundary-only model often has limited expressive capability and may lead to noticeable reconstruction errors in the interior.

To improve the interior reconstruction quality, we adopt an error-driven iterative refinement strategy. At each iteration, we first evaluate the pixel-wise reconstruction errors between the current C^0 GC reconstruction and the input raster image, and select the pixel with the largest error. The corresponding location of this pixel in the parameter domain is then inserted as a new interior control point. In this way, the positions of interior control points are determined adaptively according to the current most significant reconstruction errors, rather than being fixed in advance.

Suppose that M interior control points have been inserted so far, and let $P_1, \ldots, P_M$ denote the unknown RGB colors of these interior control points. Their locations are fixed once selected from the corresponding maximum-error pixels in the parameter domain, while their colors are solved by minimizing the objective function

$$\min_{P_1,\ldots,P_M} \sum_{i \in \Omega} \|\hat{y}_i(P) - y_i\|_2^2 + w \sum_{m=1}^{M} \left\|P_m - y_m^*\right\|_2^2, \tag{7}$$

where Ω is the set of fitting pixels, y_i and $\hat{y}_i(P)$ are the ground

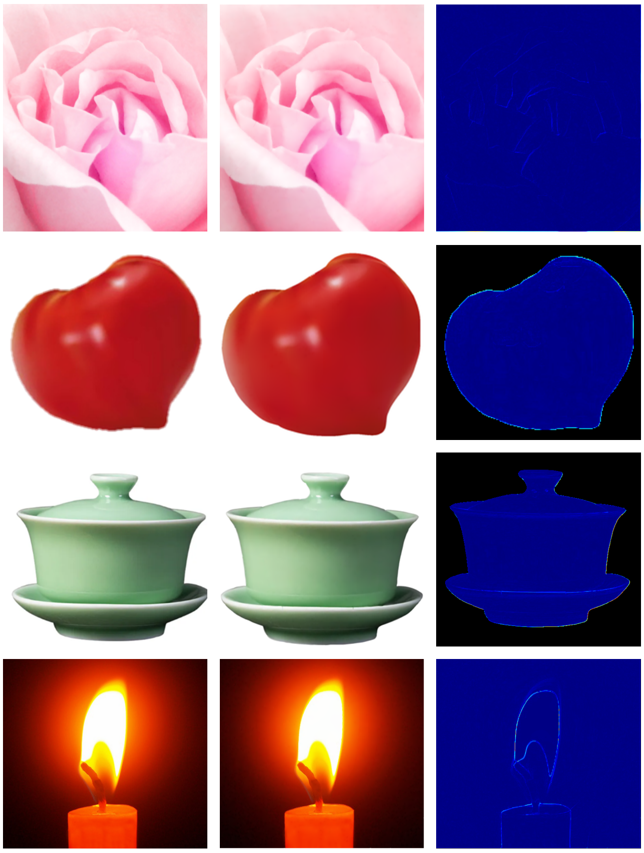


Figure 7: *Vectorization results for different images (cf. Table 2). Each row shows the original image (left), the reconstructed result (center), and the reconstruction error map (right).*

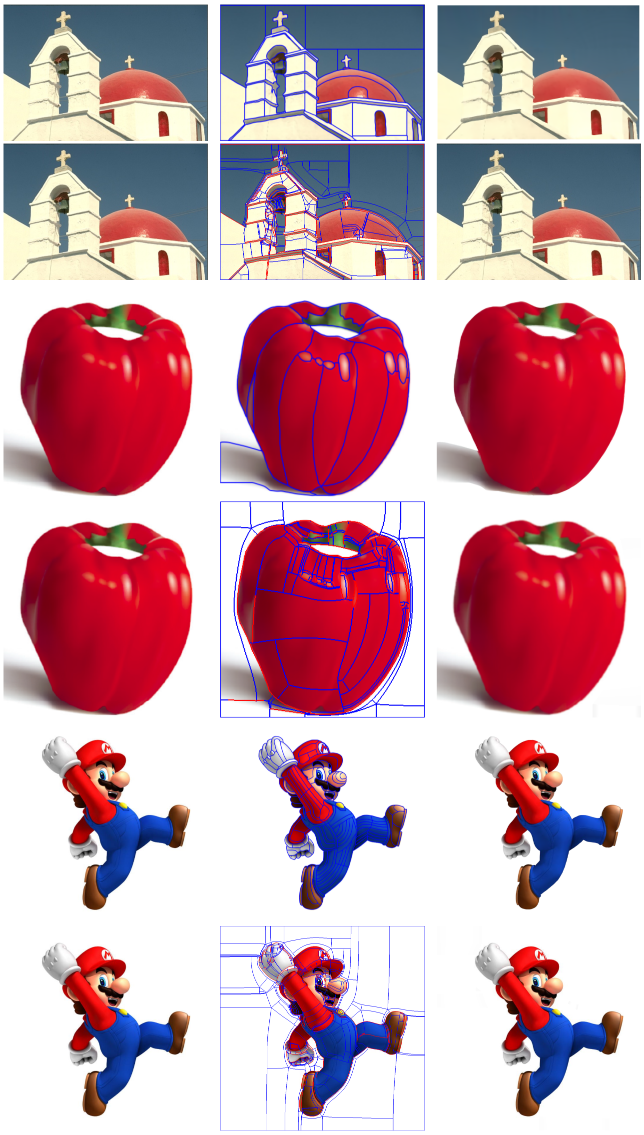


Figure 8: *Comparison of our method (top row for each example) to He et al. [HRK24] (bottom rows). While the vectorized results (right) of the input images (left) are visually indistinguishable, our method needs far fewer patches (center); cf. Table 2.*

truth and the reconstructed RGB values, respectively, and y_m^* is the ground truth RGB value at the selected pixel corresponding to the m-th interior control point. The first term measures the reconstruction error over all fitting pixels, while the second term is a regularization term that encourages each interior control point color to remain close to the ground truth color of its corresponding selected pixel, thereby improving numerical stability. Here, w is a regularization weight balancing the data fidelity term and the regularization term. We set $w = 30$ in all our experiments.

After solving for the color values of the newly inserted interior control points, the C^0 GC patch is updated and rendered again, and a new error map is computed with respect to the input image. Then, in the next iteration, another maximum-error pixel is selected, and the above process of control-point insertion, color optimization, and reconstruction update is repeated. Through this progressive refinement procedure, the expressive capability of the patch interior is continuously enhanced, while the boundary interpolation property is strictly preserved.

Compared with inserting all interior control points at once, our progressive strategy adaptively places them in regions with large reconstruction errors, effectively improving the reconstruction of complex color variations.

The refinement process continues until the global reconstruction error falls below a prescribed threshold or the maximum number of iterations is reached. In all our experiments, the error threshold and the maximum number of iterations were fixed to 5 and 2000, respectively. The resulting refined C^0 GC patch preserves strict boundary interpolation while providing a more accurate reconstruction of the interior color distribution and fine-scale details, thereby significantly improving the quality of image vectorization.

Table 1: *Characteristics of our test data and performance parameters of our method; see Figs. 1 and 7 for results.*

Image	Resolution	Original size (KB)	Added points	Patch #	Average error	MSE	Storage (KB)	Time breakdown (s)			
								Mesh/HC	Evaluation	Optimization	Total
plumeria	477×441	116	8554	85	0.94	1.91	64.80	369	571	351	1291
flower	505×561	288	10723	168	1.26	2.62	78.00	655	734	292	1681
tomato	255×214	32	1230	17	1.18	4.66	9.70	44	88	43	175
cup	457×390	120	9270	77	2.28	4.11	71.90	409	704	468	1581
candle	521×482	298	12681	164	2.51	4.46	77.63	748	893	539	2180

Table 2: *Performance comparison; see Fig. 8.*

Image (resolution)	Method	Patch #	MSE	Storage (KB)
church (481×321)	Our	56	3.48	51.95
	[HRK24]	540	2.48	59.27
pepper (256×270)	Our	22	4.03	31.40
	[HRK24]	190	1.26	44.64
Mario (700×700)	Our	146	2.63	86.50
	[HRK24]	799	1.96	39.90

5. Experimental Results

We have implemented our method in Python on a Windows 11 laptop with an Intel Core i7-14700HX CPU and 16 GB RAM. Figure 1 and 7 show some representative results and Table 1 reports the corresponding statistics.

To identify the main computational bottlenecks, we further divide the runtime into three stages. Mesh/HC includes initial and adaptive triangulation and harmonic-coordinate computation. Evaluation includes C^0 GC patch evaluation at mesh vertices, dense-sample localization and interpolation, and reconstruction-error evaluation. Optimization includes the assembly and solution of the least-squares systems for fitting the interior control-point colors. As shown in Table 1, evaluation constitutes the largest portion of the runtime. This is mainly because the GC patches must be repeatedly evaluated at the mesh vertices and dense sample points after each interior control point is inserted. Mesh/HC is another major computational component, since the parameter-domain mesh and harmonic coordinates need to be updated during each adaptive refinement iteration.

5.1. Comparison

We compared the proposed method to the most recent state-of-the-art method by He et al. [HRK24] taking three aspects into account: reconstruction quality, the number of patches, and the ability to largely avoid fold-over.

Table 2 reports the statistics of this comparison, including the number of patches, the reconstruction error, and the storage size. These data and the visual results in Fig. 8 show that our method achieves high reconstruction quality while using significantly fewer patches. This is mainly because our method represents each region using a shape-adaptive polygonal patch, allowing irregular regions to be represented more directly and thereby reducing unnecessary patch subdivision. For storage, instead of directly storing the absolute color values of the interior control points, we store the differences between these values and the colors predicted by the corresponding Coons patches. Since these differences are small and regularly distributed, they can be efficiently compressed using `np.savez_compressed`. For example, for the church image, the file size would increase from 51.95 KB to 70.00 KB without this residual-based storage scheme. The Mario example has a relatively large storage size because it contains more patches than the other examples in Table 2 and requires more interior control points to achieve its relatively low MSE.

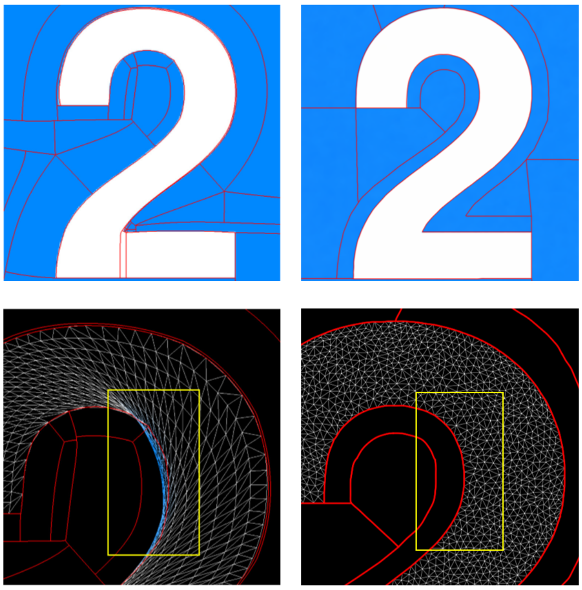


Figure 9: *Comparison of our method (right) to He et al. [HRK24] (left). While the Coons patches of quadrilateral patches (top left) over rectangular domains can fold over (bottom left), our C^0 GC patches over shape-adapted domains avoid this problem.*

We further compare our method with classical Coons patch representations. Since a classical Coons patch is defined over a rectangular parameter domain, it may suffer from fold-over when the prescribed boundary curves describe a concave or geometrically complex region in the image domain. Fig. 9 shows a representative

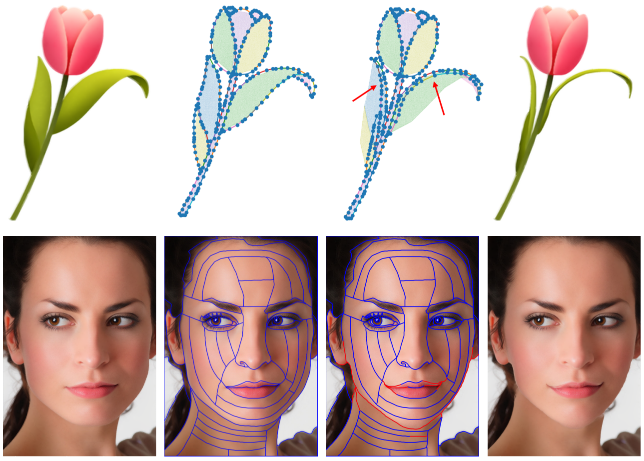


Figure 10: *The vectorized result (left) can be edited easily (right) by moving the control points of the fitted B-spline boundary curves without changing the polygonal parameter domains (center).*

Figure 11: *Our method allows users to turn a vectorized red pepper (left) into an orange pepper (right) by simple color editing.*

example. For He et al.'s method [HRK24], fold-overs may appear in the generated mesh, resulting in a distorted patch layout. In contrast, our method uses a polygonal parameter domain that better conforms to the shape of the target region, thereby avoiding this problem to a large extent.

5.2. Editing of Vectorized Results

A major advantage of the proposed representation with few patches is editability. Since an image can be represented as a set of C^0 GC patches, users can model regions of interest as independent patches according to the semantic structure of the image or specific editing requirements. Since the boundary of each patch is represented by B-spline curves, the shape of a selected region can be modified intuitively by adjusting the control points of the corresponding B-spline boundary curves during subsequent editing. Once the boundary curves are changed, the geometry of the associated patches is updated accordingly, automatically producing deformed vectorized image results, as shown in Fig. 10. This editing capability is particularly important in vector graphics applications, where users often wish to further manipulate image shapes after vectorization. In our framework, such modifications can be performed directly on the spline-based boundary representation without rerunning the entire vectorization pipeline.

In addition to geometric deformation, our representation also supports direct color editing. For each C^0 GC patch, the color distribution is determined by the color values defined on the B-spline boundary curves and the colors of the interior control points. Therefore, changing these color control values is sufficient to modify the appearance of the corresponding region, while the patch geometry and topology remain unchanged, as shown in Fig. 11.

6. Limitations

Although the proposed C^0 GC-patch-based image vectorization method demonstrates stable and high-quality reconstruction results, several limitations remain.

First, the vectorization quality depends to some extent on the quality of the initial region segmentation. Since each region is modeled independently using a parametric surface patch, inaccurate or inconsistent segmentation may introduce suboptimal region boundaries, which cannot be fully corrected by subsequent patch refinement alone.

Second, for regions with highly complex color variations or rich fine-scale details, achieving accurate reconstruction often requires introducing a larger number of interior control points to enhance interior expressiveness. While this improves reconstruction quality, it also increases model complexity and representation size.

These limitations suggest that improving the robustness of region segmentation and developing more efficient strategies for allocating interior control points in complex regions are promising directions for future work.

7. Conclusions

We presented an image vectorization method based on generalized Coons patches. The proposed method first segments the input image into several regions and constructs a parametric surface representation for each region independently, thereby converting the raster image into a vector representation composed of multiple C^0 GC patches. Based on a boundary-interpolating C^0 GC patch, we further introduce an adaptive interior control point refinement strategy based on least-squares optimization to enhance the ability of each patch to represent interior color variations. Experimental results show that the proposed method achieves stable and high-quality reconstruction on images with different content complexities while using a relatively small number of patches.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive suggestions and comments. This work was supported by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (2026C04052) and the National Natural Science Foundation of China (NSFC) under the project number 61872121.

References

[BB12] Barla P., Bousseau A.: Gradient art: Creation and vectorization. In *Image and Video-Based Artistic Stylisation*, Rosin P., Collomosse J., (Eds.), vol. 42 of *Computational Imaging and Vision*. Springer, London, 2012, ch. 8, pp. 149–166. doi:10.1007/978-1-4471-4519-6_8. 2

[BLHK18] Barendrecht P. J., Luinstra M., Hogervorst J., Kosinka J.: Locally refinable gradient meshes supporting branching and sharp colour transitions. *The Visual Computer 34*, 6 (June 2018), 949–960. doi:10.1007/s00371-018-1547-1. 3

[CLL*20] Chen K.-W., Luo Y.-S., Lai Y.-C., Chen Y.-L., Yao C.-Y., Chu H.-K., Lee T.-Y.: Image vectorization with real-time thin-plate spline. *IEEE Transactions on Multimedia 22*, 1 (Jan. 2020), 15–29. doi:10.1109/TMM.2019.2922126. 2, 3

[Coo67] Coons S. A.: *Surfaces for Computer-aided Design of Space Forms*. Tech. Rep. MIT-LCS-TR-041, Massachusetts Institute of Technology, Cambridge, MA, June 1967. URL: https://hdl.handle.net/1721.1/149362. 3

[dB76] de Boor C.: Splines as linear combinations of B-splines. In *Approximation Theory II*, Lorentz G. G., Chui C. K., Schumaker L. L., (Eds.). Academic Press, New York, 1976, pp. 1–47. 6

[DDI06] Demaret L., Dyn N., Iske A.: Image compression by linear splines over adaptive triangulations. *Signal Processing 86*, 7 (July 2006), 1604–1616. doi:10.1016/j.sigpro.2005.09.003. 2

[FSH11] Finch M., Snyder J., Hoppe H.: Freeform vector graphics with controlled thin-plate splines. *ACM Transactions on Graphics 30*, 6 (Dec. 2011), Article 166, 10 pages. doi:10.1145/2070781.2024200. 2

[HBK19] Hettinga G. J., Brals R., Kosinka J.: Colour interpolants for polygonal gradient meshes. *Computer Aided Geometric Design 74* (Oct. 2019), Article 101769, 15 pages. doi:10.1016/j.cagd.2019.101769. 3

[HEK21] Hettinga G. J., Echevarria J., Kosinka J.: Efficient image vectorisation using mesh colours. In *Smart Tools and Applications for Graphics* (Oct. 2021), Frosini P., Giorgi D., (Eds.), STAG 2021, The Eurographics Association, pp. 139–150. doi:10.2312/stag.20211484. 2, 3

[HRK24] He K., Roerdink J. B. T. M., Kosinka J.: Image vectorization using a sparse patch layout. *Graphical Models 135* (Oct. 2024), Article 101229, 10 pages. doi:10.1016/j.gmod.2024.101229. 2, 3, 8, 9, 10

[HSF*20] Hou F., Sun Q., Fang Z., Liu Y.-J., Hu S.-M., Qin H., Hao A., He Y.: Poisson vector graphics (pvg). *IEEE Transactions on Visualization and Computer Graphics 26*, 2 (Feb. 2020), 1361–1371. doi:10.1109/TVCG.2018.2867478. 2

[IKCM13] Ilbery P., Kendall L., Concolato C., McCosker M.: Biharmonic diffusion curve images from boundary elements. *ACM Transactions on Graphics 32*, 6 (Nov. 2013), Article 219, 12 pages. doi:10.1145/2508363.2508426. 2

[JCW09] Jeschke S., Cline D., Wonka P.: Rendering surface details with diffusion curves. *ACM Transactions on Graphics 28*, 5 (Dec. 2009), Article 117, 8 pages. doi:10.1145/1618452.1618463. 2

[JMD*07] Joshi P., Meyer M., DeRose T., Green B., Sanocki T.: Harmonic coordinates for character articulation. *ACM Transactions on Graphics 26*, 3 (July 2007), Article 71, 9 pages. doi:10.1145/1276377.1276466. 4

[LHFY12] Liao Z., Hoppe H., Forsyth D., Yu Y.: A subdivision-based representation for vector image editing. *IEEE Transactions on Visualization and Computer Graphics 18*, 11 (Nov. 2012), 1858–1867. doi:10.1109/TVCG.2012.76. 2, 3

[LHM09] Lai Y.-K., Hu S.-M., Martin R. R.: Automatic and topology-preserving gradient mesh generation for image vectorization. *ACM Transactions on Graphics 28*, 3 (July 2009), Article 85, 8 pages. doi:10.1145/1531326.1531391. 2, 3

[LJH13] Li X.-Y., Ju T., Hu S.-M.: Cubic mean value coordinates. *ACM Transactions on Graphics 32*, 4 (July 2013), Article 126, 10 pages. doi:10.1145/2461912.2461917. 3

[LKSD17] Lieng H., Kosinka J., Shen J., Dodgson N. A.: A colour interpolation scheme for topologically unrestricted gradient meshes. *Computer Graphics Forum 36*, 6 (Sept. 2017), 112–121. doi:10.1111/cgf.12862. 3

[LL06] Lecot G., Lévy B.: Ardeco: Automatic Region DEtection and COnversion. In *17th Eurographics Symposium on Rendering* (2006), Akenine-Moeller T., Heidrich W., (Eds.), EGSR '06, The Eurographics Association. doi:10.2312/EGWR/EGSR06/349-360. 2

[LTKD15] Lieng H., Tasse F. P., Kosinka J., Dodgson N. A.: Shading curves: Vector-based drawing with explicit gradient control. *Computer Graphics Forum 34*, 6 (Sept. 2015), 228–239. doi:10.1111/cgf.12532. 2

[LWD04] Lin H., Wang G., Dong C.: Constructing iterative non-uniform B-spline curve and surface to fit data points. *Science in China, Series F: Information Sciences 47*, 3 (May 2004), 315–331. doi:10.1360/02yf0529. 5, 6, 7

[OBW*08] Orzan A., Bousseau A., Winnemöller H., Barla P., Thollot J., Salesin D.: Diffusion curves: a vector representation for smooth-shaded images. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), Article 92, 8 pages. doi:10.1145/1360612.1360691. 2

[PJSH15] Prévost R., Jarosz W., Sorkine-Hornung O.: A vectorial framework for ray traced diffusion curves. *Computer Graphics Forum 34*, 1 (Feb. 2015), 253–264. doi:10.1111/cgf.12510. 2

[QZY*24] Qin K., Zhou Y., Ying C., Li Y., Deng C.: C^0 generalized Coons patches for high-order cage-based deformation. *ACM Transactions on Graphics 43*, 6 (Dec. 2024), Article 220, 15 pages. doi:10.1145/3687972. 2, 3

[S*20] Sekachev B., et al.: *opencv/cvat: v1.1.0*. Zenodo, Aug. 2020. doi:10.5281/zenodo.4009388. 5

[SLWS07] Sun J., Liang L., Wen F., Shum H.-Y.: Image vectorization using optimized gradient meshes. *ACM Transactions on Graphics 26*, 3 (July 2007), Article 11, 7 pages. doi:10.1145/1276377.1276391. 2, 3

[SP06] Swaminarayan S., Prasad L.: Rapid automated polygonal image decomposition. In *35th Applied Imagery and Pattern Recognition Workshop* (2006), AIPR '06, IEEE Computer Society, pp. 1–6. doi:10.1109/AIPR.2006.30. 2

[STZ14] Sun T., Thamjaroenporn P., Zheng C.: Fast multipole representation of diffusion curves and points. *ACM Transactions on Graphics 33*, 4 (July 2014), Article 52, 12 pages. doi:10.1145/2601097.2601187. 2

[TG24] Tian X., Günther T.: A survey of smooth vector graphics: Recent advances in representation, creation, rasterization, and image vectorization. *IEEE Transactions on Visualization and Computer Graphics 30*, 3 (Mar. 2024), 1652–1671. doi:10.1109/TVCG.2022.3220575. 2

[WZG*19] Wei G., Zhou Y., Gao X., Ma Q., Xin S., He Y.: Field-aligned quadrangulation for image vectorization. *Computer Graphics Forum 38*, 7 (Oct. 2019), 171–180. doi:10.1111/cgf.13826. 3

[XLY09] Xia T., Liao B., Yu Y.: Patch-based image vectorization with automatic curvilinear feature alignment. *ACM Transactions on Graphics 28*, 5 (Dec. 2009), Article 115, 10 pages. doi:10.1145/1618452.1618461. 2, 3

[XSTN14] Xie G., Sun X., Tong X., Nowrouzezahrai D.: Hierarchical diffusion curves for accurate automatic image vectorization. *ACM Transactions on Graphics 33*, 6 (Nov. 2014), Article 230, 11 pages. doi:10.1145/2661229.2661275. 2

[YKH10] Yuksel C., Keyser J., House D. H.: Mesh colors. *ACM Transactions on Graphics 29*, 2 (Mar. 2010), Article 15, 11 pages. doi:10.1145/1731047.1731053. 3

[ZCX*22] ZHU H., CAO J., XIAO Y., CHEN Z., ZHONG Z., ZHANG Y. J.: TCB-spline-based image vectorization. *ACM Transactions on Graphics 41*, 3 (June 2022), Article 34, 17 pages. doi:10.1145/3513132. 2, 3

[ZDZ18] ZHAO S., DURAND F., ZHENG C.: Inverse diffusion curves using shape optimization. *IEEE Transactions on Visualization and Computer Graphics 24*, 7 (July 2018), 2153–2166. doi:10.1109/TVCG.2017.2721400. 2

[ZZW14] ZHOU H., ZHENG J., WEI L.: Representing images using curvilinear feature driven subdivision surfaces. *IEEE Transactions on Image Processing 23*, 8 (Aug. 2014), 3268–3280. doi:10.1109/TIP.2014.2327807. 2, 3