Towards High-fidelity Artistic Image Vectorization via Texture-Encapsulated Shape Parameterization

Ye Chen[1] Bingbing Ni[1,2*] Jinfan Liu[1] Xiaoyang Huang[1] Xuanhong Chen[1,2]
[1]Shanghai Jiao Tong University, Shanghai 200240, China
[2]USC-SJTU Institute of Cultural and Creative Industry
{chenye123, nibingbing, chen19910528}@sjtu.edu.cn

Abstract

*We develop a novel vectorized image representation scheme accommodating both shape/geometry and texture in a decoupled way, particularly tailored for reconstruction and editing tasks of artistic/design images such as Emojis and Cliparts. In the heart of this representation is a set of sparsely and unevenly located 2D control points. On one hand, these points constitute a collection of parametric/vectorized geometric primitives (*e.g.*, curves and closed shapes) describing the shape characteristics of the target image. On the other hand, local texture codes, in terms of implicit neural network parameters, are spatially distributed into each control point, yielding local coordinate-to-RGB mappings within the anchored region of each control point. In the meantime, a zero-shot learning algorithm is developed to decompose an arbitrary raster image into the above representation, for the sake of high-fidelity image vectorization with convenient editing ability. Extensive experiments on a series of image vectorization and editing tasks well demonstrate the high accuracy offered by our proposed method, with a significantly higher image compression ratio over prior art.*

1. Introduction

How to represent images is a fundamental problem in the field of computer vision. Traditionally, images are represented by pixels stored on fixed and discrete grids (*i.e.*, raster images). One main advantage of such representation is that, there is virtually no limit to its ability to express image details with enough pixels. Nevertheless, pixel-based representation is plagued by numerous limitations. On one hand, it suffers from excessive redundancy due to its storage on fixed grids, which inherently constrains image resolution, along with inevitable information loss during image re-scaling. On the other hand, most importantly, pixel-based representation stores all the information of an image in the RGB values of the pixels, resulting in a high coupling between the representation of image geometry and texture. Hence, it is challenging to edit the shape/geometry and texture of a raster image separately.

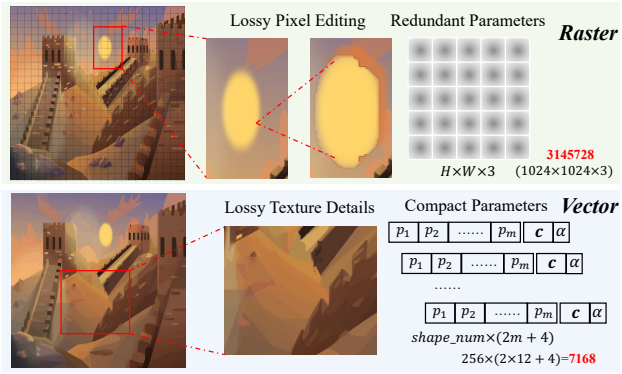

Figure 1. **Illustration of our motivation. Raster images** that store pixel values on fixed and discrete grids suffer from excessive parameter redundancy and a high coupling between image geometry and texture, resulting in challenging image editing. **Vector images** represent visual concepts with vectorized geometric primitives in a very compact parameter format, endowed with great editability. However, vector images are not suitable for expressing image texture details. This work explores a **compact vectorized image representation** that decouples images in geometric and texture space, tailored for high-fidelity texture-rich image vectorization, facilitating easy image editing.

Vector images embody an alternative paradigm for representing visual information, which describe images with a collection of parametric/vectorized geometric primitives (*e.g.*, curves and closed shapes), defined as control points in the continuous space. Compared to representing images with fixed discrete grids, vector images present many advantages. 1) Resolution-Agnostic: Pixel-based image rescaling often introduces unexpected noise/artifact that is hard to identify and remove. In contrast, modeling images with continuous parameters enables the storage and generation of images in arbitrary resolutions without in-

*Corresponding author: Bingbing Ni

formation loss. 2) Easy Editing: Vector images have explicit/explainable parameter format, facilitating convenient image editing ability by simply modifying shape or color parameters in a structured way. 3) Lossless Compression: The shape primitives in vector images are typically defined by mathematical functions, allowing for the expression of complex shapes with a minimal number of control points. Therefore, the number of parameters required to store a vector image is significantly lower than that of a raster image.

Unfortunately, while vector images possess indisputable advantages in representing geometric information, they fall short in effectively capturing the texture details found in images, which restricts their applications to simple images (*e.g*. Emojis, Fonts and Icons). As depicted in Fig. 1, current image vectorization representations [7, 15, 17, 25, 26] suffer from significant loss of fine texture details when confronted with regions containing intricate textures. The underlying reason can be attributed to the fact that traditional vectorization representations store a single color within each shape, rather than capturing the distribution of colors. Consequently, this representation inherently exhibits inefficiency in conveying rich image textures.

To address both limitations, we propose a memory-efficient texture-encapsulated shape-vectorized image representation that decomposes an image into dual-parameterized geometry and texture space, facilitating high-fidelity representation of complicated image geometry and texture as well as decoupled editing ability. Our novel image representation features the following designs. In a nutshell, geometry is encoded with closed Bézier shapes parameterized into the coordinates of control points, similar to contemporary image vectorization algorithms [7, 15, 17, 26]. In the meantime, to represent texture, we follow the emerging implicit neural representation scheme [22, 29], which is parameterized into a lightweight MLP that takes local latent encodings of coordinate information as input and predicts local texture (*i.e*. RGB values). Drawing inspiration from distributed implicit representations [23], these local texture codes are spatially distributed into each control point of our geometric representation, namely, each control point stores a latent texture code which takes charge of the local coordinate-to-RGB mappings within the surrounding region. To this end, an image is parameterized to a set of coordinates of control points assigned with corresponding latent texture codes and a lightweight MLP shared across the whole image space, which fully capture image characteristics including color, texture, and geometry, in a decoupled way. Compared to current methods that store texture parameters in fixed grids, our shape control point anchored distributed texture representation scheme substantially minimizes storage overhead, while assigning more resources for describing those more crucial local visual regions (patches).

Leveraging our developed representation, we accomplish remarkably high-fidelity image vectorization (*i.e*. parameterized reconstruction and editing) with a straightforward zero-shot learning algorithm based on self-supervision. Unlike previous algorithms that rely on differentiable rasterization to estimate gradients and optimize control point coordinates, which often encounters large gradient error, we efficiently constrain the Bézier shapes to match crucial areas in image space by constructing a geometric field, thus encouraging accurate parameter fitting and fast convergence. We extensively experiment the proposed framework for a series of image vectorization tasks including Emoji [1], Icon [2] and our developed Clipart benchmarks. It is demonstrated that our method achieves much better image vectorization quality with a significantly reduced number of parametric shapes, compared to prior art. In addition, compared to image reconstruction methods based on implicit neural representations, our method achieves better reconstruction results with only $10\times$ fewer latent codes, which significantly demonstrates the effectiveness and efficiency of our representation. Moreover, our method can edit images by simply editing the shape and texture parameters in a decoupled way.

2. Related Works

Image Vectorization. Image vectorization is an important research area in computer graphics. Traditional approaches [8, 33] typically utilize a two-stage algorithm that involves an initial step of image segmentation, followed by targeted vectorization of the segmented regions. In the era of deep learning, several neural network-based methods [7, 26] are proposed inspired by the differentiable rasterization method DiffVG [15]. Im2Vec [26] uses a variable-complexity closed Bézier path as the fundamental graphic primitive. Chen *et al*. [7] utilize several geometric primitives like triangle and rectangle to fit the image. Nevertheless, these methods all face challenges when it comes to vectorizing complex images, as predicting a large number of control points simultaneously solely based on pixel loss is very difficult. LIVE [17] is a remarkable work that generates compact vectorized representations for complex images with a meticulously designed layer-wise path initialization technique. Although LIVE can generate decent vectorization results for complex images in almost any domain, it has inherent limitations in representing image textures due to a lack of exploration of internal color distributions within closed Bézier shapes. As a result, it relies on stacking local shapes to express texture details. Du *et al*. [9] propose to use linear gradients to define the spatially varying color within local image regions, endowed with the ability to edit images easily in a structured way. However, the linear gradient layers rely heavily on an intricate layer configuration strategy. In this work, we in-

troduce a texture-encapsulated shape-vectorized representation to capture complex image textures, achieving high-fidelity image vectorization with a simplistic framework.

Implicit Neural Representation. Implicit neural representation is widely applied in the field of 3D vision [3, 5, 11, 12, 21, 28], which models 3D shapes and appearances with MLPs that map coordinates to signals. Particularly, NeRF [22] inspires a series of outstanding works [4, 13, 16, 19, 24, 32]. Recent advances [5, 10, 20, 23] investigate the utilization of latent representations to balance between memory usage and computational expenses of heavy MLPs.

While implicit neural representation achieves success in 3D tasks, its applications in 2D domain [6, 14, 29–31] are relatively unexplored. Deep Image Prior [31] is a pioneering work to represent images using deep ConvNets with abundant neural parameters. SIREN [29] parameterizes 2D images with MLP and novel periodic activation functions. However, the MLPs of SIREN are highly redundant and inefficient, and unacceptable reconstruction errors occur as the hidden layers are further reduced. This work proposes an efficient and flexible representation by encapsulating latent codes into crucial shape parameters, which can be optimized with a zero-shot learning framework.

3. Methodology

3.1. Overview

The overall framework is illustrated in Fig. 2. Our framework decouples the raster image $\boldsymbol{I} \in \mathbb{R}^{H\times W\times 3}$ in geometric space and texture space using a texture-encapsulated shape-vectorized representation, *i.e.*, a parameterized shape representation (Sec. 3.2) and a shape-anchored implicit neural texture representation (Sec. 3.3).

We utilize parameterized closed Bézier shapes (defined as coordinates of control points, as implemented in [15, 17]) to represent the fundamental geometric information in the image. We establish a geometric field based on image edge points to constrain the coordinates of control points of all Bézier shapes, enabling Bézier shapes to efficiently cover regions with rich geometric information in the image space.

Then we introduce a shape-anchored implicit neural texture representation to explore image textures. Specifically, our framework assigns an optimizable latent code z to each control point in the shape representation. Then a shape-aware interpolation function ϕ (no parameters) and a trainable lightweight MLP f_θ (with θ as its parameters) are utilized to represent the texture in an implicit coordinate-to-RGB way, where the texture information at arbitrary position can be spatially interpolated and retrieved.

To conclude, we parameterize the input image as a set of Bézier shapes attached with trainable texture latents and a lightweight MLP f_θ, which can be formulated as:

$$\boldsymbol{I} \sim \{\mathbb{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_n\}, \theta\}, \tag{1}$$

where n denotes the shapes needed to represent an image. And $\boldsymbol{s}_i$ takes the form:

$$\boldsymbol{s}_i = \{(\boldsymbol{p}_{i1}, \boldsymbol{z}_{i1}), (\boldsymbol{p}_{i2}, \boldsymbol{z}_{i2}), ..., (\boldsymbol{p}_{im}, \boldsymbol{z}_{im})\}, \tag{2}$$

where $\boldsymbol{p}_* = (x_*, y_*) \in \mathbb{R}^2$ denotes the coordinate of control points and $\boldsymbol{z}_* \in \mathbb{R}^d$ represents the latent texture code assigned to the corresponding control point. m denotes the number of control points for each shape. More details are elaborated in the following sections.

3.2. Parameterized Shape Representation

We use Bézier shapes defined by control points to represent image geometry. Common image vectorization methods tend to optimize the control points with pixel loss by utilizing the gradients approximated by differentiable renderer (*e.g.*, DiffVG [15]), which suffers from the issue of gradient sparsity. In contrast, we introduce a geometric field to optimize the coordinates of control points. For the input raster image, we use the Canny operator to extract the edge points $\mathbb{E} = \{\boldsymbol{e}_i\}$ and generate the corresponding geometric field. More concretely, we think that the closer a point is to the edge of an image, the more geometric information it contains. Hence, we define the geometric field as a probability field, which describes the importance of each query point in describing the image geometry. The ground-truth geometric field G_e defined by edge points can be formulated as:

$$G_e(\boldsymbol{q}) = 1 - tanh(\min_i \frac{||\boldsymbol{q} - \boldsymbol{e}_i||_2^2}{\sigma^2}), \tag{3}$$

where $\boldsymbol{q} = (x, y) \in \mathbb{R}^2$ denotes any query coordinate in the image space and σ is used to control the range of the field. We utilize the geometric field to optimize the coordinates of control points of all Bézier shapes. Specifically, we use control points in $\mathbb{S}$ to densely sample a set of Bézier curve points $\mathbb{B} = \{\boldsymbol{b}_i(t)\}$:

$$\boldsymbol{b}_i(t) = \boldsymbol{B}_{\boldsymbol{p}_{i1}\boldsymbol{p}_{i2}\cdots\boldsymbol{p}_{im}}(t), \tag{4}$$

where $\boldsymbol{b}_i(t)$ is a curve point of Bézier shape $\boldsymbol{s}_i$ conditioned on t and $\boldsymbol{B}$ is the Bézier function. Then we compute the predicted geometric field G_b:

$$G_b(\boldsymbol{q}) = 1 - tanh(\min_{i,t} \frac{||\boldsymbol{q} - \boldsymbol{b}_i(t)||_2^2}{\sigma^2}). \tag{5}$$

We use the binary cross-entropy loss to compare the distributions of G_e and G_c:

$$\mathcal{L}_g = BCE(G_b(*), G_e(*)). \tag{6}$$

In addition, we use the chamfer distance loss to encourage the Bézier shapes to fit the key edge points in the image:

$$\mathcal{L}_c = \sum_{\boldsymbol{b}\in\mathbb{B}} \min_{\boldsymbol{e}\in\mathbb{E}} \|\boldsymbol{b} - \boldsymbol{e}\|_2^2 + \sum_{\boldsymbol{e}\in\mathbb{E}} \min_{\boldsymbol{b}\in\mathbb{B}} \|\boldsymbol{e} - \boldsymbol{b}\|_2^2. \tag{7}$$

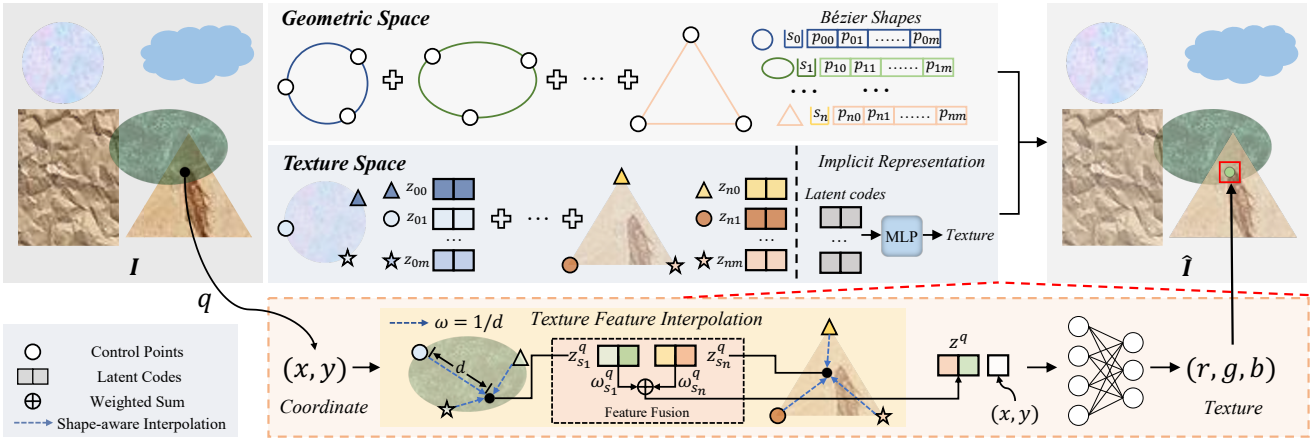

Figure 2. **Overview of our framework.** We propose a novel texture-encapsulated shape-vectorized representation to decouple the raster image in geometric and texture space. We utilize Bézier shapes defined as coordinates of control points to represent image geometry and a shape-anchored implicit neural representation to explore image texture efficiently by distributing latent texture codes into control points. With such representation, the texture feature at arbitrary coordinate can be spatially interpolated and fused within Bézier shapes and the texture information can be retrieved through a very lightweight MLP.

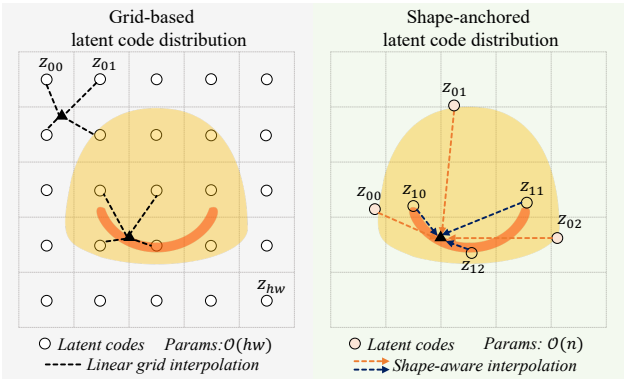

Figure 3. An example of the comparison between our shape-anchored latent code distribution and general grid-based latent code distribution. "n" denotes shape number. Our strategy greatly reduces storage overhead. Moreover, our shape-aware interpolation restricts the interpolation process to the interior of shapes, making the interpolation smoother and accelerating convergence because the texture inside shapes tends to be relatively smooth.

With Eqn.(6&7), we can iteratively optimize the coordinates of control points to constrain Bézier shapes to cover the crucial areas with rich geometric information.

3.3. Shape-anchored Implicit Neural Texture Representation

We use implicit neural representation in the continuous parameter domain to explore complex image texture. Following recent progress in image reconstruction via implicit neural representation [6], we also distribute some latent codes in the image spatial positions.

However, we abandon the usual strategy that assigns latent codes to fixed and discrete grid points, which is very inefficient and inflexible because it not only leads to storage redundancy but also makes the training process extremely complex (as shown in Fig. 3). For our problem, we notice that there are some smooth regions in the image space that can be fitted with only a small amount of latent codes, while more latent codes are needed to be allocated for those regions with sharp gradients/changes (*e.g.*, the edge points), which significantly aligns with our utilization of geometric representation in Sec. 3.2 in that texture-rich regions well correspond to edge regions. Based on the above considerations, we distribute the texture latent codes on the control points of the Bézier shapes in a distributed manner, which significantly improves the utilization efficiency of latent codes and reduces storage costs.

Specifically, in our texture representation, the image texture information is parameterized by a set of latent codes stored in control points and a lightweight mapping function ($f_\theta : [\boldsymbol{x}, \boldsymbol{z}] \mapsto \boldsymbol{c}$) which decodes the latent codes to texture values (RGB), where $\boldsymbol{x}$ and $\boldsymbol{z}$ are the coordinate and the corresponding latent vector/texture feature at query point in the continuous image space. $\boldsymbol{z}$ is obtained via a shape-aware interpolation method, *i.e.*, interpolating the latent codes of the shape that covers the query point. Assume a query point $\boldsymbol{q} \in \mathbb{R}^2$ is within a shape $\boldsymbol{s}_i$, the shape-aware interpolation function ϕ is formulated as:

$$\phi(\boldsymbol{s}_i, \boldsymbol{q}) = \sum_{j=1}^{m} \frac{w_j^i(\boldsymbol{q})}{\sum_{j=1}^{m} w_j^i(\boldsymbol{q})} \boldsymbol{z}_{ij}, \tag{8}$$

with

$$w_j^i(\boldsymbol{q}) = \frac{1}{\|\boldsymbol{q} - \boldsymbol{p}_{ij}\|_2^2}, \tag{9}$$

where $(\boldsymbol{p}_{ij}, \boldsymbol{z}_{ij}) \in \boldsymbol{s}_i$, as defined in Eqn. 2.

For points in the image space, we adopt a divide-and-conquer strategy to assign them with latent vectors. Specifically, for points covered by Bézier shapes, we can obtain their latents by interpolating the latent codes of the control points of the shapes they belong to by utilizing Eqn. 8. For the points that are not covered by any Bézier shape, we assign them a default latent vector. Note that a pixel may be covered by several Bézier shapes, thus we propose an inverse distance weighting feature fusion method (IDW-Fusion) to fuse the latent vectors interpolated by all shapes. We can formulate this process as:

$\mathbf{1}^d$, **if** $\widetilde{\mathbb{S}}^{\boldsymbol{q}} = \emptyset$, (10) with:

$$w^{\boldsymbol{s}_i}(\boldsymbol{q}) = \frac{m}{\sum_{j=1}^{m} \|\boldsymbol{q} - \boldsymbol{p}_{ij}\|_2^2}, \tag{11}$$

where $\widetilde{\mathbb{S}}^{\boldsymbol{q}} \subset \mathbb{S}$ denotes the set of all shapes that cover point $\boldsymbol{q}$, which can be obtained with the Winding Number Algorithm. The RGB value at point $\boldsymbol{q}$ can be predicted as:

$$\boldsymbol{c}^{\boldsymbol{q}} = f_\theta([\boldsymbol{q}, \boldsymbol{z}^{\boldsymbol{q}}]), \tag{12}$$

where $[\cdot, \cdot]$ means concatenation.

Compared to previous image vectorization methods that use only one color within a Bézier shape, our shape-anchored implicit neural texture representation models the color distribution inside the Bézier shape efficiently. In addition, in our texture representation, the number of latent codes is resolution-agnostic, which achieves a significant compression of parameters and efficient texture representational capability with the content-adaptive latent code distribution.

3.4. Parameterized Representation Optimization

Given an input raster $\boldsymbol{I} \in \mathbb{R}^{H \times W \times 3}$, our task is to reconstruct it with the above proposed parameterized representation in a zero-shot learning framework (*i.e.*, single-image optimization). We have to optimize the parameters in a self-supervised manner because we only have raster images for optimization without any image-to-vector or image-to-parameter labeling. Thus we have to rasterize our parameterized representation into a raster image $\hat{\boldsymbol{I}} \in \mathbb{R}^{H \times W \times 3}$, with the same size as the input image and then measure the pixel errors. Specifically, for each pixel $[i, j]$ of the output image $\hat{\boldsymbol{I}}$, we query its color with our parameterized representation by viewing it as a point in the image space with coordinate $\boldsymbol{x}_{ij} = [\frac{i}{H} - 0.5, \frac{j}{W} - 0.5]$ as in [18]. Then we can obtain the corresponding texture feature $\boldsymbol{z}^{\boldsymbol{x}_{ij}}$ and use f_θ to approximate the RGB value:

$$\hat{\boldsymbol{I}}[i, j] = f_\theta([\boldsymbol{x}_{ij}, \boldsymbol{z}^{\boldsymbol{x}_{ij}}]). \tag{13}$$

Then we can utilize the pixel-wise mean square error loss to optimize the parameters, which is formulated as:

$$\mathcal{L}_r = \|\boldsymbol{I} - \hat{\boldsymbol{I}}\|_2^2. \tag{14}$$

After optimization, the input image is parameterized into decoupled geometric and texture space, facilitating high-fidelity reconstruction with easy editing by simply editing the shape and texture parameters in a decoupled way.

Optimization objectives. The input image is decomposed into our parameterized representation in a zero-shot learning algorithm. As shown in Alg. 1, we firstly randomly initialize all parameters of our representation including the control points and corresponding latent codes of all shapes (*i.e.*, $\mathbb{S}$) and the MLP parameters (*i.e.*, θ). Then we iteratively optimize all the parameters with a combination of $\mathcal{L}_g$, $\mathcal{L}_c$ and $\mathcal{L}_r$, and the optimal problem of our framework can be expressed as:

$$\min_{\{\mathbb{S}, \theta\}} \mathcal{L}_r + \mathcal{L}_g + \mathcal{L}_c. \tag{15}$$

Algorithm 1: Zero-shot Representation Learning

Input : $\boldsymbol{I}$, n, m, d, $iters$;
Random Init: $\mathbb{S}$, θ ; // Parameters
Generate: $\mathbb{E}$, G_e, $\mathbb{Q}$;
// Edge points, Geometric field, All pixel points
for i *in range(iters)* **do**
  $\mathbb{B}$ = sample_curve_points($\mathbb{S}$);
  // pred geometric field
  G_b = geometric_field($\mathbb{B}$);
  // $\widetilde{\mathbb{S}}$ for all pixel points
  $\widetilde{\mathbb{S}}^{\mathbb{Q}}$ =winding_number_parallel($\mathbb{S}$, $\mathbb{Q}$);
  // latent vector for all pixel points
  $\boldsymbol{z}^{\mathbb{Q}}$ = latent_interp($\widetilde{\mathbb{S}}^{\mathbb{Q}}$, $\mathbb{Q}$);
  // rasterize
  $\hat{\boldsymbol{I}} = f_\theta(\mathbb{Q}, \boldsymbol{z}^{\mathbb{Q}})$;
  // compute loss
  $\mathcal{L} = \mathcal{L}_r(\boldsymbol{I}, \hat{\boldsymbol{I}}) + \mathcal{L}_g(G_b, G_e) + \mathcal{L}_c(\mathbb{B}, \mathbb{E})$;
  update parameters $\mathbb{S}$, θ ;
end
Output: Parameterized image representation:$\{\mathbb{S}, \theta\}$.

4. Experiments

4.1. Experimental Setups

Datasets. Current image vectorization algorithms are limited to handling images with simple geometric structures like Emoji [1] and Icon [2], and there is a lack of exploration for images with complex textures. In this paper, in addition to comparing our method with existing methods on commonly used Emojis and Icons, we also introduce a **Clipart Dataset** consisting of 200 clipart images with complex shapes, textures, and rich backgrounds, which is very challenging for image vectorization task.

n	Dataset	DiffVG	LIVE	NPA	**Ours**
5	Emoji	0.0212	0.0019	0.0049	**0.0007**
	Icon	0.0573	0.0026	0.0093	**0.0009**
10	Emoji	0.0092	0.0016	0.0020	**0.0006**
	Icon	0.0285	0.0024	0.0017	**0.0007**

Table 1. **Image vectorization results on Emoji&Icon datasets.** n means the number of shapes. MSE results are reported. Our method achieves significantly better results than state-of-the-art.

Implementation Details. By default, we use four segments for each closed Bézier shape, thus each shape contains 12 control points (*i.e.*, $m = 12$). The number of shapes n is set to 5 and 128 for Emoji/Icon dataset and Clipart dataset respectively. Definitely, more shapes lead to better vectorized results. The control factor σ of the geometric field is set to 0.005. We set the dimension of latent codes as $d = 16$, and the MLP f_θ is implemented as a linear layer with input dimension 18 and output dimension 3. With the help of a highly parallel optimization process, we can reduce the time it takes to optimize an image to the order of minutes.

4.2. Image Vectorization

We evaluate our texture-encapsulated shape representation on image vectorization task by measuring the differences between the input raster images and the rasterized vectors. The comparison with SOTA methods (*i.e.*, DiffVG [15], LIVE [17] and NPA [7]) are performed both quantitatively and qualitatively on Emoji, Icon and Clipart dataset.

Emoji&Icon Datasets. The quantitative comparisons on Emoji and Icon datasets are shown in Tab. 1. Our method outperforms all previous image vectorization methods on both benchmarks, especially when fewer Bézier shapes are utilized. The results fully demonstrate the compactness of our texture-encapsulated shape representation, especially in the efficient utilization of shapes. Qualitative comparisons are shown in Fig 4. We can see that our representation achieves high-fidelity image vectorization results even with a minimal number of shapes. Note that LIVE also generates very compact representation, but it relies heavily on the initialization strategy and still struggles to generate regular shapes when dealing with shape intersections. We also show some decoupled editing examples in Fig. 5. Our method can easily transform geometric shapes and replace the textures within the corresponding shapes with specific ones like frosted glass texture.

Clipart Dataset. The Clipart dataset contains clipart images with complex texture information, which is very challenging for image vectorization task. We use this dataset primarily to demonstrate the efficiency of our texture-encapsulated shape representation in expressing complex textures, especially when the number of shapes is limited. NPA [7] is limited in its ability to simultaneously optimize a large number of shape parameters only with gradients approximated by differentiable rasterization method, which prevents it from producing acceptable results on this dataset. Therefore, we only compare our method with DiffVG [15] and LIVE [17]. The quantitative results are shown in Tab. 2. Our method performs significantly better than DiffVG and LIVE when utilizing the same number of shapes. In addition, despite the layer-wise representation of LIVE is highly compact, we are still able to achieve comparable reconstruction results by further reducing the number of shapes by half. The qualitative results are visualized in Fig. 6. We can observe that our representation models the texture details much better than other methods, and effectively prevents local texture artifacts caused by the stacking of redundant shapes.

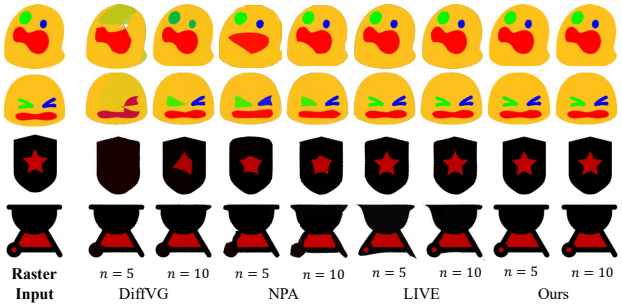

Figure 4. **Qualitative comparison on Emoji&Icon datasets.** n means the number of shapes. Our method can accurately represent the image geometry even with limited number of shapes.

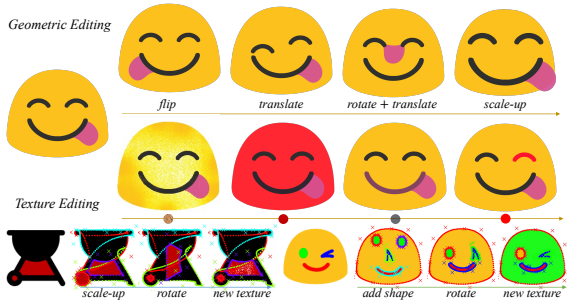

Figure 5. We showcase the editability of our parameterized representation. We can edit the image geometry and image texture in a decoupled way simply by editing the corresponding parameters. In the bottom row, we show the shape decompositions with the "x" marks indicating the control points.

4.3. Learning Implicit Image Representations

Considering that we utilize implicit neural representation to parameterize images, we also make comparisons with general implicit image representations. SIREN [29] is a classical method to learn implicit representations for images parameterized by neural networks. In this section, we compare our method with SIREN on the task of image reconstruction with continuous representation. To compare with methods that store latent codes on fixed grids, we also utilize the framework of LIIF [6] to perform zero-

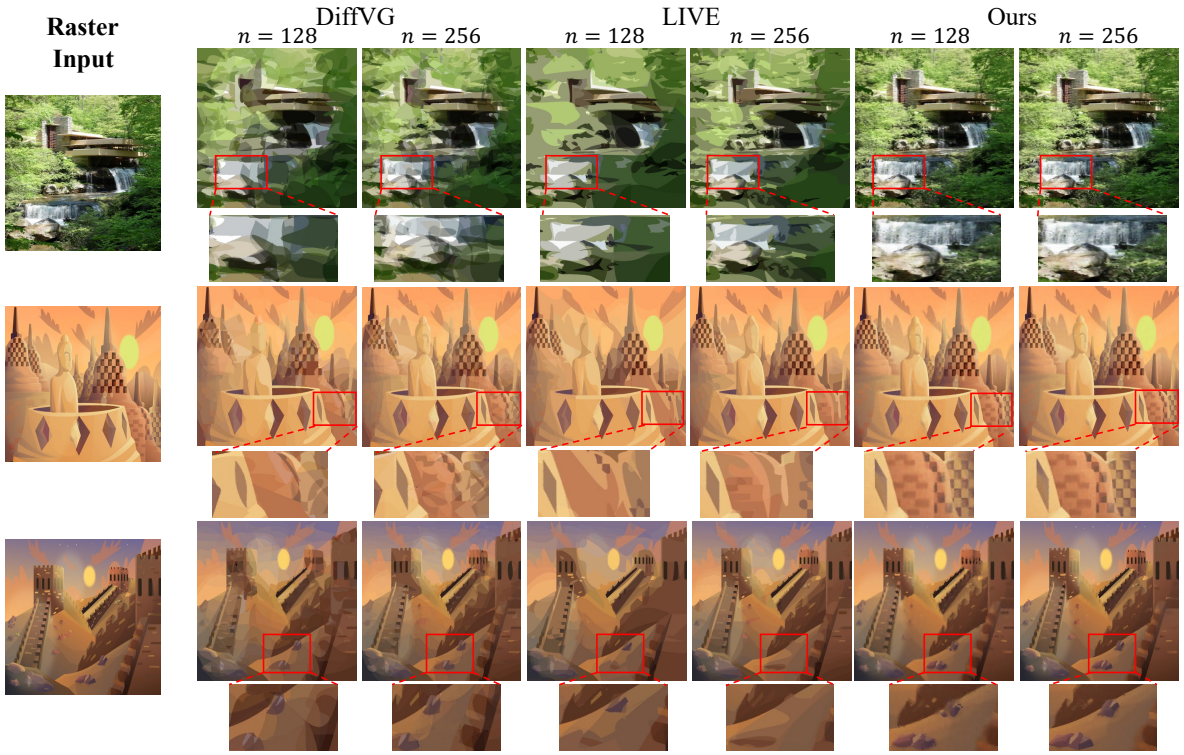

Figure 6. **Qualitative comparison on Clipart dataset.** n means the number of shapes. We use red boxes to emphasize the differences. Our representation can express complex image details with a small number of shapes. Please zoom in for more details.

n	Method	MSE↓	LPIPS↓	SSIM↑
128	DiffVG [15]	0.0089	0.3741	0.7763
	LIVE [17]	0.0090	0.3424	0.7916
	Ours	**0.0009**	**0.2492**	**0.8803**
256	DiffVG [15]	0.0035	0.3271	0.8092
	LIVE [17]	0.0028	0.2894	0.8631
	Ours	**0.0006**	**0.2218**	**0.9014**

Table 2. **Image vectorization results on Clipart dataset.** Pixel MSE, LPIPS and SSIM are reported. n means the number of shapes. The LPIPS is computed based on VGG [27]. Our method achieves significantly better reconstruction results than state-of-the-arts, especially when limited number of shapes are used.

shot image reconstruction. Specifically, instead of training a heavy encoder to generate latent codes, we directly distribute randomly initialized latent codes on fixed grid points and iteratively fit the implicit parameters (including latent codes and an MLP-based decoding function) for each image. We demonstrate the efficiency of our texture-encapsulated shape representation by comparing the image reconstruction errors and the corresponding number of parameters required for the implicit representation on the Clipart dataset. The results are shown in Tab. 3. We can see that our method achieves better image reconstruction results with a significantly higher parameter compression ratio than other methods. When using the same order of magnitude of parameters, our reconstruction results are obviously better than other methods. Some visualization results are shown in Fig. 7. For SIREN, more hidden layers lead to better reconstruction quality, but it lacks the ability to reconstruct images with lightweight networks. For methods that store latent codes on fixed grids, a large number of latent codes are needed to achieve acceptable reconstruction results. In contrast, our representation captures rich image textures with only a small number of latent codes and a very lightweight MLP because we assign more resources for describing crucial local visual regions by flexibly encapsulating texture parameters in shape control points. Both quantitative and qualitative results demonstrate that our method learns efficient continuous representations of images.

4.4. Ablation Study

In this section, we explore the crucial designs and hyperparameters of our framework. For simplicity, we only use MSE as the metric for all quantitative comparisons.

Component Analyses. We first investigate the effectiveness of each training objective. The MSE results are shown in Tab. 4. We see that the geometric field ($\mathcal{L}_g$) effectively improves the reconstruction results among all datasets, and the chamfer distance loss ($\mathcal{L}_c$) further improves reconstruction qualities in the complex Clipart dataset. The results demonstrate that, compared to only using pixel loss, our

Method	MSE↓	LPIPS↓	SSIM↑	Codes↓	Params.↓
SIREN-1	0.0058	0.4424	0.7035	-	17152
SIREN-3	0.0020	0.2642	0.8325	-	49920
Grid-/4	0.0010	0.2602	0.8506	16384	262198
Grid-/16	0.0043	0.4896	0.6724	1024	16438
Ours	**0.0009**	**0.2492**	**0.8803**	1536	27702

Table 3. **Comparison with general implicit image representations.** "Codes" denote the number of latent codes utilized. "Params" denote the number of parameters. "SIREN" does not distribute latent codes. "SIREN-$*$" means the SIREN version with $*$ hidden layers. In "Grid-/$*$", the "$*$" denotes the proportion of the spatial dimension of the original image to the grids storing latent variables. Our representation achieves comparable reconstruction results with a significant parameter compression. When using the same order of magnitude of parameters, our method achieves obviously higher reconstruction quality.

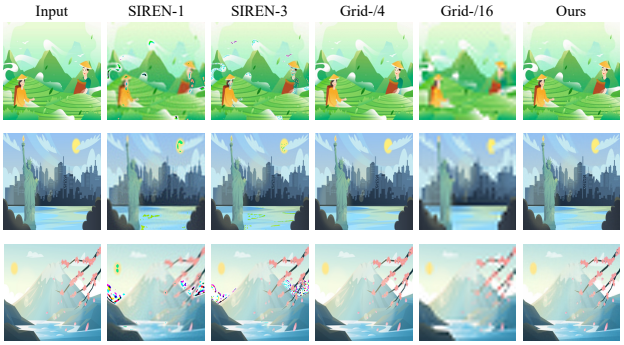

Figure 7. **Qualitative comparison with general implicit image representations.** Compared to SIREN and the usual strategy to store latent codes on grids, our representation achieves clearer and more faithful reconstruction results with fewer parameters.

Losses	Emoji	Icon	Clipart
$\mathcal{L}_r$	0.0013	0.0018	0.0043
$\mathcal{L}_r + \mathcal{L}_g$	0.0008	0.0011	0.0013
$\mathcal{L}_r + \mathcal{L}_g + \mathcal{L}_c$	0.0007	0.0009	0.0009

Table 4. **Component Analyses on the training objectives.** MSE results on several datasets are reported.

method makes the shapes more accurately fit key geometric regions by explicitly constraining the coordinates of control points utilizing geometric information in the image.

A second experiment is conducted to explore how the texture feature fusion method affects the image reconstruction results. More concretely, we compare our inverse distance weighting feature fusion method (*i.e.*, IDW-Fusion) with the ablated version of directly adding the features interpolated by all shapes together (*i.e.*, Sum-Fusion). We also investigate the effectiveness of attaching the coordinates of query points to the features. The results are shown in Tab. 5. We can observe that the IDW-Fusion method is very effective by considering the varying degrees of influence of each shape on the query point. Notably, concatenating the features with coordinates is effective and crucial in our method because points in the background do not belong to any explicit shapes and the position information can serve as the discriminative features for these points.

	Emoji	Icon	Clipart
IDW-Fusion	0.0007	0.0009	0.0009
Sum-Fusion	0.0013	0.0012	0.0016
w/o coords	0.0015	0.0020	0.0042

Table 5. **Component Analyses on the feature fusion method.** MSE results on several datasets are reported.

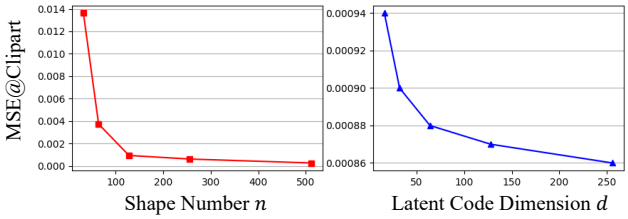

Figure 8. **Parameter Analyses on the shape number and latent code dimension.** MSE results on Clipart dataset are reported.

Parameter Analyses. We explore how shape number n and latent code dimension d affect the reconstruction results. The MSE results on Clipart dataset are shown in Fig. 8. We can see that more shapes lead to better reconstruction results and our method can achieve competitive results with only 64 shapes in the complex Clipart dataset. In addition, we can see that higher dimensions only marginally improve the reconstruction capability. Considering the computational and storage costs it incurs, $d = 16$ is an efficient setting.

5. Conclusion

This work presents a novel vectorized image representation to decompose images into parameterized shape and texture space. Along with our representation, we introduce a straightforward zero-shot learning framework in a self-supervised manner for image vectorization task. Extensive experimental results on various benchmarks and tasks prove that our representation achieves high-fidelity image reconstruction with a significantly high image parameters compression, endowed with convenient image editing by simply editing corresponding shape and texture parameters in a decoupled way.

6. Acknowledgment

This work was supported by National Science Foundation of China (U20B2072, 61976137). This work was also partially supported by Grant YG2021ZD18 from Shanghai Jiaotong University Medical Engineering Cross Research. This work was partially supported by STCSM 22DZ2229005.

References

[1] Note emoji. https://github.com/googlefonts/noto-emoji. Accessed: 2021-09-30. 2, 5

[2] creativestall. https://thenounproject.com/creativestall/. 2, 5

[3] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020. 3

[4] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 3

[5] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 608–625. Springer, 2020. 3

[6] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021. 3, 4, 6

[7] Ye Chen, Bingbing Ni, Xuanhong Chen, and Zhangli Hu. Editable image geometric abstraction via neural primitive assembly. In *ICCV*, pages 23514–23523, 2023. 2, 6

[8] James Richard Diebel. *Bayesian Image Vectorization: the probabilistic inversion of vector image rasterization*. Stanford University, 2008. 2

[9] Zheng-Jun Du, Liang-Fu Kang, Jianchao Tan, Yotam Gingold, and Kun Xu. Image vectorization and editing via linear gradient layer decomposition. *ACM Transactions on Graphics (TOG)*, 42(4):1–13, 2023. 2

[10] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 3

[11] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 3

[12] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 3

[13] Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, and Gerard Pons-Moll. Control-nerf: Editable feature volumes for scene rendering and manipulation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4340–4350, 2023. 3

[14] Jaewon Lee and Kyong Hwan Jin. Local texture estimator for implicit representation function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1929–1938, 2022. 3

[15] Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *TOG*, 39(6):1–15, 2020. 2, 3, 6, 7

[16] Jinxian Liu, Ye Chen, Bingbing Ni, Jiyao Mao, and Zhenbo Yu. Inferring fluid dynamics via inverse rendering. *arXiv preprint arXiv:2304.04446*, 2023. 3

[17] Xu Ma, Yuqian Zhou, Xingqian Xu, Bin Sun, Valerii Filev, Nikita Orlov, Yun Fu, and Humphrey Shi. Towards layer-wise image vectorization. In *CVPR*, pages 16314–16323, 2022. 2, 3, 6, 7

[18] Xu Ma, Yuqian Zhou, Huan Wang, Can Qin, Bin Sun, Chang Liu, and Yun Fu. Image as set of points. *arXiv preprint arXiv:2303.01494*, 2023. 5

[19] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 3

[20] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14214–14223, 2021. 3

[21] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4743–4752, 2019. 3

[22] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2, 3

[23] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2, 3

[24] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 3

[25] Antoine Quint. Scalable vector graphics. *IEEE MultiMedia*, 10(3):99–102, 2003. 2

[26] Pradyumna Reddy, Michael Gharbi, Michal Lukac, and Niloy J Mitra. Im2vec: Synthesizing vector graphics without vector supervision. In *CVPR*, pages 7342–7351, 2021. 2

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[28] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[29] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020. 2, 3, 6

[30] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P Srinivasan, Jonathan T Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2846–2855, 2021.

[31] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018. 3

[32] Peng Wang, Yuan Liu, Zhaoxi Chen, Lingjie Liu, Ziwei Liu, Taku Komura, Christian Theobalt, and Wenping Wang. F2-nerf: Fast neural radiance field training with free camera trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4150–4159, 2023. 3

[33] Tian Xia, Binbin Liao, and Yizhou Yu. Patch-based image vectorization with automatic curvilinear feature alignment. *ACM Transactions on Graphics (TOG)*, 28(5):1–10, 2009. 2