DESIGN ELEMENTS EXTRACTION BASED ON UNSUPERVISED SEGMENTATION AND COMPACT VECTORIZATION

Hong Qu, Yanghong Zhou, K.P. Chau and P.Y. Mok
Institute of Textiles and Clothing, The Hong Kong Polytechnic University, Hunghom, Hong Kong

ABSTRACT

Repeated design elements are abundant and ubiquitous in decorative patterns, which are now widely used in the process of design for objects, artworks and images found in our living environment. Extraction of repeated design elements from images of existing decorative patterns benefits understanding design, extracting compressed information for subsequent operation, e.g., design generation and vectorization. The early methods based on hand-crafted features were computationally inefficient and less accurate. Most deep learning (DL) based methods focus on natural environment images and are difficult to generalize to decorative images. Besides, DL-based methods require massive datasets with instance-level annotations, which are labor-intensive and hard to get. This paper proposes a novel scheme for design element extraction and vectorization. First of all, unsupervised segmentation is proposed to extract repeated design elements from images of unknown artworks without human assistance. We then distill the color information of the extracted repeated element based on statistics reflected in the color histogram of the input artwork. We develop an algorithm to remove redundant information extracted from images in order to get a compact vectorization result, reusable design element in vector format, at the end. To validate the proposed scheme, we conducted several experiments and the result demonstrated the effectiveness of our scheme and its potential for design generation application.

KEYWORDS

Image Understanding, Localization, Unsupervised Segmentation, Vectorization

1. INTRODUCTION

Artist-drawn patterns with distinctly colored regions and sharp, independent boundaries are widely used in the fashion industry, wrapping paper, and other digital media (Hoshyari et al., 2018, Dominici et al., 2020). A pattern is composed of discrete numbers of recurring elements called repeated elements, often arranged in regular or near-regular layouts over a solid background layer. Since the artist-drawn pattern has powerful decorative and communication functions, creating a pattern is popular in both professional and amateur. Many patterns are created in vector format for wide applications, which are scalable, reusable, and easily adjusted to fit new requirements (Zhang et al., 2009). Unfortunately, designing a completely new pattern requires substantial time and effort, particularly for the amateur. It will be wonderful to extract and vectorize repeated elements from existing patterns to support and release workload for people who want to create their graphics.

Object detection is a task to localize instances of particular object classes in an image (Inoue et al., 2018, Zou et al., 2019). Best-performing object detection methods are supervised learning-based and typically learned from natural images. Compared to natural images, the content of patterns is more straightforward and more abstract. The domain difference stops us from applying object detection methods to extract repeated elements.

Image segmentation algorithms also aim to partition an image into multiple segments and distinguish the different objects or regions of interest inside (Vitale et al., 2016). Since a labeled pattern dataset cannot be obtained in a short time and high cost, we consider using traditional segmentation algorithms or unsupervised segmentation algorithms. The traditional ones assign image pixels to a specific segment by analyzing low-level image features, such as color and texture, or based on graph theory and multivariate image analysis (Vitale et al., 2016, Felzenszwalb and Huttenlocher, 2004). They tend to generate an over-segmentation result with too many regions with flat shading and meaningless shape (Zhang et al., 2009). However, unsupervised

segmentation methods aim at partitioning an arbitrary input image into more general labels, such as foreground and background (Kanezaki, 2018, Kim et al., 2020). Kanezaki (2018) treated convolutional neural network (CNN) as a multi-level feature extractor, and she proposed a novel method alternately iterate segment images and network parameter learning. However, experiment results show the method still tends to suffer from over-segmentation over foreground content and occasionally fail in under segmentation. To overcome these problems, we repeatedly segment an input image three times and extract the stable background mask for subsequent application.

Image vectorization, also known as image tracing, of a raster image is the process of converting a raster image into a vector image. Learning-based image vectorization methods are still restricted to fonts and simple icons domains (Reddy et al., 2021, Lopes et al., 2019). Most classic methods use closed Bezier curves (Bezigons) to describe intermediate representations generated by boundary detection or segmentation in image vectorization results. Since raster images are composed of square pixels, they naturally have jagged edges and are even more apparent when zooming in. Consequently, the resultant Bezigons are sometimes imperfect due to accumulated errors, inaccurate segmentation, and a lack of curve priors, especially for low-resolution images. Yang et al. (2015) focus on this problem proposes a new method that directly optimizes the Bezigons rather than using intermediate representations. However, their method assumes discontinuity and corner points are given. Besides, commercial software, such as Adobe Illustrator (2017), prefers generating many small Bezigons for ambiguous border areas to ensure fidelity. Even so, the function is not widely used at work, as it is tedious to remove the meaningless Bezigons before subsequent operations. Recently, Dominici et al. (2020) and Hoshyari et al. (2018) proposed schemes to generate vector images consistent with viewer expectations, and learning-based corner points algorithms were embedded. However, their methods assume the input image only contains a single closed region, which means color separation or segmentation approaches should be processed in advance. Our goal is to prepare vector repeated elements for design generation. Compared with high quality, efficiency and compactness are prior considering factors. Therefore, we propose to get compact vectorized results based on optimizing Potrace (Selinger, 2003).

The paper proposes a novel framework for repeated elements extraction and compact vectorization from existing patterns without any manual intervention. After analyzing the character of design patterns, we improve an existing unsupervised segmentation algorithm to first separate foreground elements and background layers. Later, we extract the repeated elements by removing the duplicate and incomplete elements. We assume the input image is an image containing many colorful elements. Since Potrace focuses on vectorizing black-on-white images, we combine color histogram results and the K-means algorithm to separate the image into independent color masks at first. Then, we delete redundant Bezigons and combine residual Bezigons ordered by area for compactness.

The rest of the paper is organized as follows. Section 2 describes the framework. Section 3 contains experimental results and quantitative analysis, and section 4 contains the concluding remarks.

2. FRAMEWORK

2.1 Overview

The input to our system is a set of design pattern images with distinctly colored regions. In this case, each image is created as a planar composition of two main layers: background and foreground. Typically, the foreground layer consists of more complicated shapes and color information, while the background layer usually contains one homogeneous region. We aim to vectorize repeated elements in the foreground for future design generation.

The first stage in our framework is repeated elements extraction jointly by an improved unsupervised segmentation method and a Canny edge detector (Canny, 1986). A redundant elements removal operation follows. Afterward, the system separates the image into different, independent color masks. Finally, a compact vector file is obtained after filtering and reordering all the Bezigons generated from the Potrace algorithm. The overall framework is shown in Figure 1.

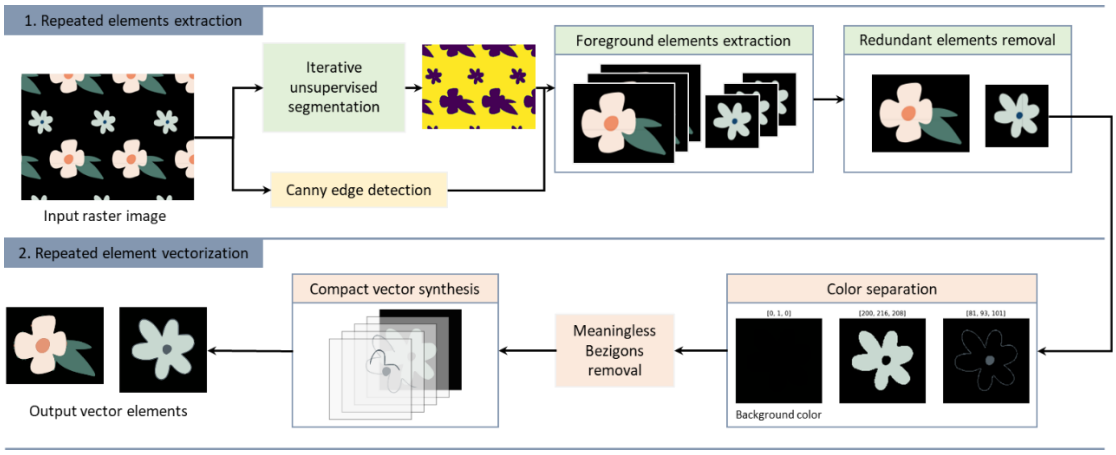

Figure 1. Framework overview: the framework consists of two stages. The top stage is repeated elements extraction, and the bottom stage is repeated element vectorization. The input of the framework is a raster pattern image, and the output is several vector elements

2.2 Repeated Elements Extraction

2.2.1 Foreground Elements Extraction

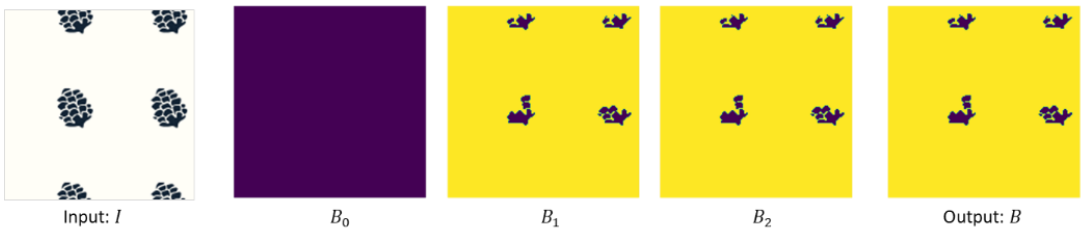

Figure 2. Iterative unsupervised segmentation

In order to separate the original image I into many foreground elements and a background layer, we adopt an unsupervised segmentation approach first presented in (Kanezaki, 2018) and later refined in (Kim et al., 2020). The approach can handle arbitrary images as the essence of the method is a clustering of image features. Ignoring the over-segmentation of the foreground, we find that the background mask is usually accurate. Therefore, instead extract foreground elements directly, we extract the background mask beforehand. To avoid occasional under-segmentation results (see B_0 in Figure 2), we first repeat the operation three times, getting a background mask $B_n, n \in [0, 1, 2]$ each time, then obtain the final background mask B by:

$$B' = \frac{1}{3} \cdot \sum B_n \tag{1}$$

$$B = \begin{cases} 1 \ \ if \ B'_{i,j} \geq 2 \\ 0 \ \ if \ B'_{i,j} < 2 \end{cases} \tag{2}$$

$$B_{i,j} \in B \tag{3}$$

where $B'_{i,j}$ and $B_{i,j}$ is the pixel value of B' and B, respectively. Here, when $B_{i,j}$ equals one means the corresponding pixel on I is the background, and the rest of I is foreground content. Later, we extract foreground elements following the bounding boxes obtained by OpenCV (Bradski, 2000). In case all extracted elements are incomplete, we do the same for the edge detection result of the original image. Then we collect all the extracted elements E as the input for the next operation. Besides, the RGB value C_{bk} of the original image's background is obtained by

$$C_{bk} = Mean(B \cdot I) \tag{4}$$

2.2.2 Redundant Elements Removal

A pattern consists of many highly similar elements, and it is unnecessary to vectorize each one. Therefore, before vectorization, we remove the redundant elements based on their similarity. Most element has an abstract and straightforward shape in which the local image feature descriptors, such as SIFT and DoG, fail to detect their key points and thus cannot compare the similarity between elements. The perceptual hashing algorithm encodes the result of the discrete cosine transform of the image to a snippet. Researchers can quickly compare the similarity between images by calculating the distance of their snippets.

Given all the elements E, for each pair of elements $e_i, e_j \in E$, we use a Perceptual Hashing algorithm (pHash) (Zauner, 2010) to encode and measure the similarity between them by calculating the distance $d_{i,j}$ and form all the distance to a set $D = \{d_{i,j}: \left| pHash(e_i) - pHash(e_j) \right|, \forall e_i, e_j \in E, i \neq j\}$. When the distance of a pair of elements is longer than the threshold $\theta = 0.8 * \max(D)$, we treat the smaller one as a redundant element and remove it. The procedure loops recursively until no pairs of elements satisfy the condition.

2.3 Repeated Element Vectorization

2.3.1 Color Separation

Before we vectorize all the extracted repeated elements, we separate a multi-color repeated element image into independent color masks. k-means clustering is a classical image segmentation algorithm where k is the number of clusters and should be given in advance (Jain, 2008). A color histogram represents the number of pixels with colors in each fixed list of color ranges. We calculate k through statistics the number of color ranges higher than a fixed threshold τ, in its color histogram after grayscale processing (see Figure 3). After this, we operate color separation by k-means clustering and successfully separate a multi-color image into k color masks. The nearest cluster center with C_{bk} is the final background color, we replace the value of C_{bk} to it and remove the corresponding color mask for computation reduction. Finally, we get a set of color masks denoted as $M_n, n \in \{1, \dots, k-1\}$ and their corresponding color values C_n and the background color C_{bk}.

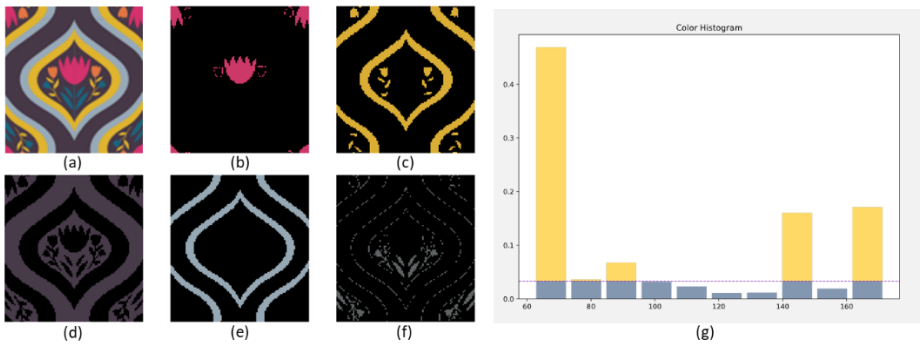

Figure 3. An example of color separation. (a) the input element image, (b)-(f) color masks after color separation, (g) color histogram, and in this image k equals five

2.3.2 Redundant Bezigons Removal

We first vectorize each color mask M_n by Potrace to V_n. Since raster images are composed of pixels with different density values, there are many redundant small Bezigons whose area is a small fraction of the overall image (see Figure 4). Therefore, the region V_n^i whose area a_n^i far less than the area of the whole image A will be removed, and a new Bezigon set Z can be obtained by $Z = \{Z_n: \frac{a_n^i}{A} \geq \omega\}$. In this paper, we used the value of $\omega = 0.05$.

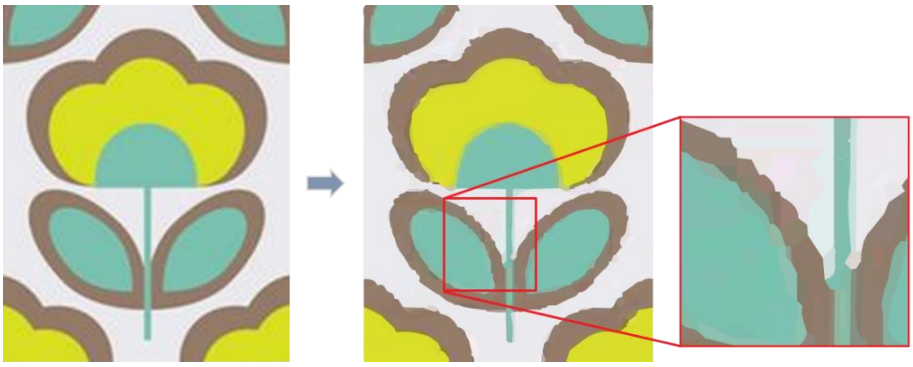

Figure 4. An example of Redundant small Bezigons

2.3.3 Compact Vector Synthesis

A vector image is a multi-layer composition of Bezigons, and each Bezigon exists in a separate layer. A Bezigon can be wholly determined by its geometric and color parameters in the final vector image. When we get the Bezigons set Z, we first order all the Bezigons by area to avoid occlusion between Bezigons. And then synthesize them together, filling them with their corresponding color. Moreover, we add a background rectangular filled with color value C_{bk} to the last layer of the vector image. An example is shown in Figure 5.

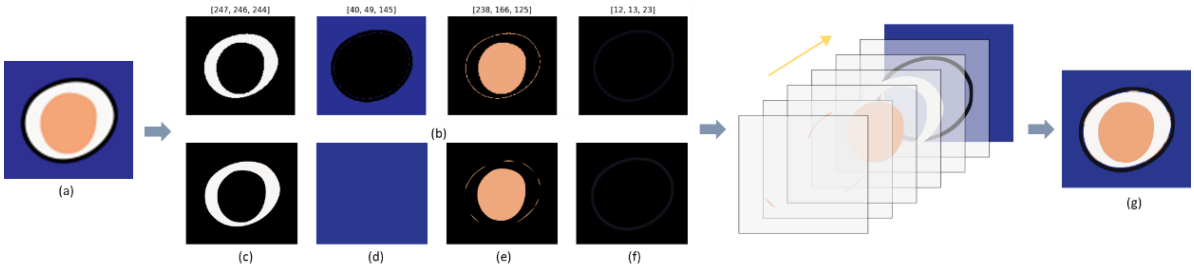

Figure 5. (a) is the input element image, (b) is its color separation results, (c)-(f) are the corresponding vectorization results by Potrace, and (g) is the final vector image our algorithm gets

3. EXPERIMENTAL RESULTS

We evaluated the proposed framework using 46 images downloaded online. Each image is different in size contains more than two repeated patterns. Meanwhile, each pattern has one or more repeated elements random layout on the planar. We first extracted the repeated patterns through our previous work (Qu et al., 2020) and then input them into our proposed framework. During the experiment, our method requires no additional operations. The experimental results demonstrate that our framework can successfully extract repeated elements and vectorize them into a compact form. Figure 6 shows part of our results, our framework can successfully extract almost all the elements inside and vectorize them (column (a) to (d)). The images here are of different sizes, proving that our method can handle input images of any size. Besides, some failed cases are shown in Figure 6. In column (e), our framework fails to detect elements with a triangular appearance, and in column (f), our compact vectorization algorithm misses the flower heart.

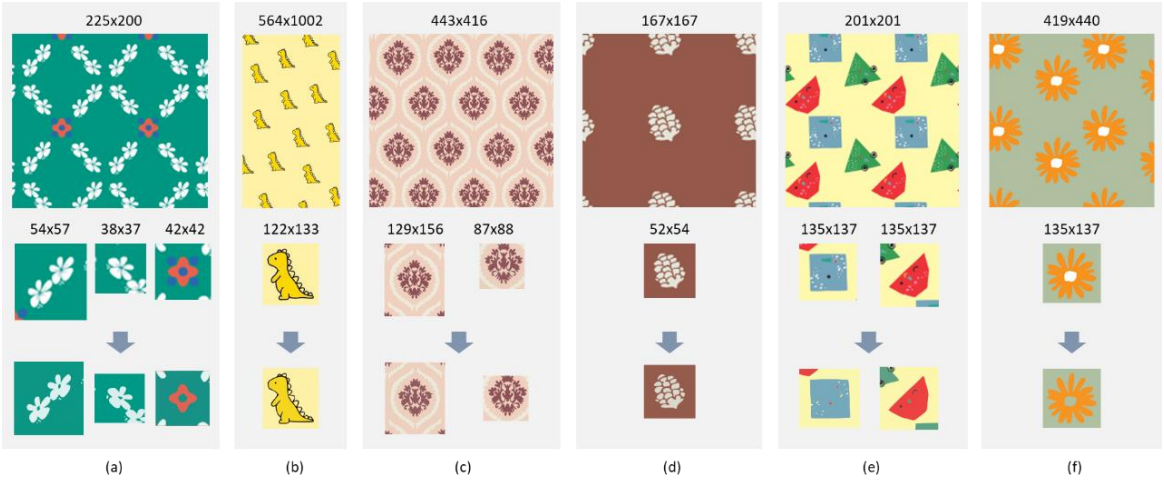

Figure 6. Part of the results of our framework

To analyze the efficiency of our method, we record the processing time of each step and summarize them in Table 1. From Nonetheless, the efficiency of our method is still within acceptable limits.

Table 1, we found that most of the time is spent on extracting repeated elements. The reason is the unsupervised segmentation network has no fixed parameters, and each time needs to be retrained to get the segmentation results. Nonetheless, the efficiency of our method is still within acceptable limits.

Table 1. Time cost record

	Repeated elements extraction	Repeated elements vectorization	Overall
Time cost (s)	22	4	26

To prove the compactness of our results, we counted the number of Bezigons in our vector images, and the average number of our results is less than 30. However, the number of commercial software is more than 50. To further prove that our method is suitable for reuse and re-creation, we randomly process the vectorized images obtained (see Figure 7). The experimental results prove that our method has the potential to assist design generation.

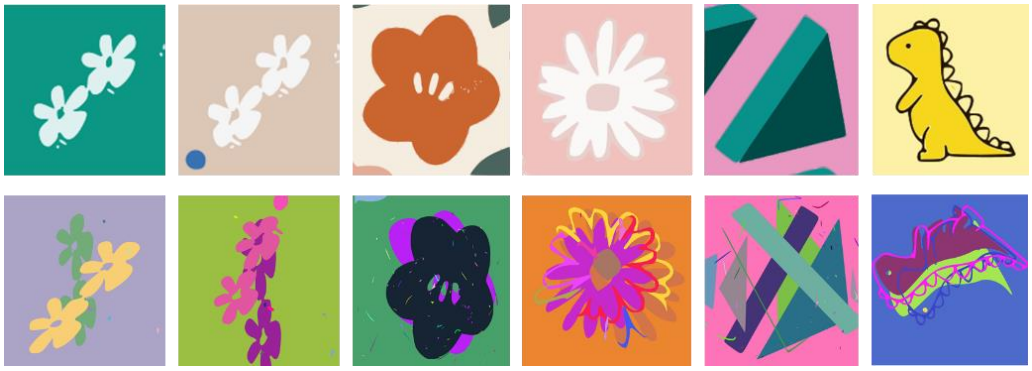

Figure 7. Random processing of the vector elements of our framework to generate new designs

4. CONCLUSION AND FUTURE WORK

This paper proposes a novel framework for repeated elements extraction and compact vectorization aiming to serve design generation. The repeated elements extraction phase creatively applies an unsupervised segmentation method to extract a pattern image's foreground elements and background color information. In addition, after noting that commercial vectorization software tends to produce many meaningless Bezigons

to ensure the fidelity of vector results, we propose an optimization method. The proposed method first counts the number of primary colors by analyzing the color histogram result of the image, then separates the image into the given number of color masks. In the subsequent vectorization, the method removed redundant Bezigons and sorted the remaining Bezigons in order of area to synthesize the final compact vector image. Although these operations will lose details in some images, they do not affect subsequent reuse and recreation. In the future, we plan to add supplementary information to the unsupervised segmentation algorithm to accelerate its convergence and replace our compact vectorization algorithm with a learning-based one.

ACKNOWLEDGEMENT

The work described in this paper was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. 152161/17E and 152112/19E).

REFERENCES

Adobe. 2017. *Adobe Illustrator*.

Bradski, G. 2000 The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Canny, J. 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis machine intelligence*, 679-698.

Dominici, E. A., et al. 2020. Polyfit: Perception-aligned vectorization of raster clip-art via intermediate polygonal fitting. *ACM Transactions on Graphics (TOG)*, 39, 77: 1-77: 16.

Felzenszwalb, P. F.andHuttenlocher, D. P. 2004. Efficient graph-based image segmentation. *International journal of computer vision*, 59, 167-181.

Hoshyari, S., et al. 2018. Perception-driven semi-structured boundary vectorization. *ACM Transactions on Graphics (TOG)*, 37, 1-14.

Inoue, N., et al. Cross-domain weakly-supervised object detection through progressive domain adaptation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 5001-5009.

Jain, A. K. Data clustering: 50 years beyond k-means. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2008. Springer, 3-4.

Kanezaki, A. Unsupervised image segmentation by backpropagation. *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2018. IEEE, 1543-1547.

Kim, W., et al. 2020. Unsupervised learning of image segmentation based on differentiable feature clustering. *IEEE Transactions on Image Processing*, 29, 8055-8068.

Lopes, R. G., et al. A learned representation for scalable vector graphics. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. 7930-7939.

Qu, H., et al. 2020. Repeated pattern extraction with knowledge-based attention and semantic embeddings. *14th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing*.

Reddy, P., et al. Im2vec: Synthesizing vector graphics without vector supervision. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 7342-7351.

Selinger, P. 2003. Potrace: a polygon-based tracing algorithm. *Potrace (online), http://potrace. sourceforge. net/potrace. pdf (2009-07-01)*, 2.

Vitale, R., et al. 2016. Segmentation techniques in image analysis: A comparative study. *Journal of Chemometrics*, 30, 749-758.

Yang, M., et al. 2015. Effective clipart image vectorization through direct optimization of bezigons. *IEEE Transactions on Visualization and Computer Graphics*, 22, 1063-1075.

Zauner, C. 2010. Implementation and benchmarking of perceptual image hash functions.

Zhang, S.-H., et al. 2009. Vectorizing cartoon animations. *IEEE Transactions on Visualization and Computer Graphics*, 15, 618-629.

Zou, Z., et al. 2019. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*.