ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.30, No-1-2, 2022

Design Space of Geometry-based Image Abstraction Techniques with Vectorization Applications

Lisa Ihde
Hasso Plattner Institute,
Digital Engineering Faculty,
University of Potsdam, Germany
lisa.ihde@student.hpi.uni-potsdam.de

Amir Semmo
Digital Masterpieces GmbH,
Potsdam, Germany
amir.semmo@digitalmasterpieces.com

Jürgen Döllner
Hasso Plattner Institute,
Digital Engineering Faculty,
University of Potsdam, Germany
juergen.doellner@hpi.uni-potsdam.de

Matthias Trapp
Hasso Plattner Institute,
Digital Engineering Faculty,
University of Potsdam, Germany
matthias.trapp@hpi.uni-potsdam.de

Figure 1: Result of vectorized pencil hatching using optimized tonal art maps based on the presented approach.

ABSTRACT

The paper presents a new approach of optimized vectorization to generate stylized artifacts such as drawings with a plotter or cutouts with a laser cutter. For this, we developed a methodology for transformations between raster and vector space. Moreover, we identify semiotic aspects of Geometry-based Stylization Techniques (GSTs) and the combination with raster-based stylization techniques. Therefore, the system enables also Fused Stylization Techniques (FSTs).

Keywords: Image stylization, Image processing, Vectorization, Non-photorealistic Rendering, Plotting

1 INTRODUCTION

1.1 Motivation

Artists use different tools and materials for creative expression, e.g., from traditional techniques such as oil painting or pencil hatching to modern digital fabrication. In recent years, research has focused on imitating these techniques to create computer graphics-based images using Non-photorealistic Rendering (NPR) techniques [Kyp13; Dev13]. The resulting images enjoy great popularity and are increasingly used and shared on social media platforms [Sem162].

While the visual quality of these results closely resembles the original techniques, they lack certain qualities during reproduction, e.g., using canvas printing or similar. However, by using fabrication techniques based on pen plotters — a commodity reproduction hardware that enjoys popularity in the maker culture — pencil hatching or stippling can be easily implemented using a variety of real pencils. The capability and functionality of these devices range from professional to hobby level, and are mainly self-built by participant of the maker scene [Mee20]. For the latter, the building instructions are freely available and partially customizable, thus not limited to certain spatial sizes or resolutions.

With respect to this, a pen plotter uses a vectorized image representation to produce a line drawing. Vector graphics add advantages compared to raster images. For example, the resolution of the graphic is then no longer limited and can be scaled up. In addition, connectivity information of graphic primitives facilitate editing processes. Furthermore and due to the geometric representation, vector graphics enable, next to printers or plotters, the use of production or fabrication devices such as laser cutters [Mue15].

1.2 Problem Statement

Mostly limited by input resolution, for a stylized photo (e.g., transformed using a pencil hatching stylization technique [Sem20]) it would be difficult to trace or reconstruction the individual lines or edges when automatically converted to a vector graphic. This challenge exists due to crossing of lines and used textures to represent the pen pressure (cf. Tonal Art Maps (TAMs) [Pra01]). To obtain an optimized and actually usable vector graphic representation, we need to

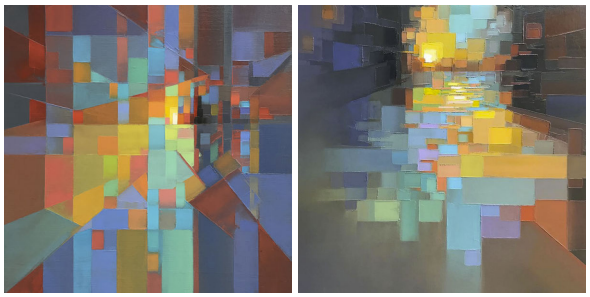


Figure 2: From "Landscapes" by Jason Anderson.

examine how a stylization technique is concretely constructed.

On the one hand, there are raster-based stylization techniques such as watercolor [Bou06], oil-painting [Sem161], Cartoon [Win06], or pencil hatching [Sem20]. On the other hand, there are Geometric Stylization Techniques (GSTs), which denote image stylization techniques that output a number of geometric primitives to represent the stylization result. In addition, there are also mixtures of both techniques, e.g., the artist Jason Anderson and his work on "Landscapes" (Figure 2). His work shows a real-world example of combining geometry-based and raster-based image stylization techniques. We denote such combinations as Fused Stylization Technique (FST). The two classes of stylization techniques can be differentiated of the different data representations of the results, namely vector and raster.

1.3 Approach and Contributions

For the approach, a system generates image stylization results based on a digital photo. Therefore, we develop a general approach of semiotic aspects to classify different geometry-based stylization techniques. The system provides different geometric stylization techniques. In addition, we also investigate the combination with raster-based techniques and how fused stylization techniques can be achieved. This is based on the elaboration of stylization operations between vector-based and pixel-based representation. To summarize, this paper makes the following contributions:

1. It presents the concept and semiotic aspects for Geometric Stylization Techniques (GSTs).

2. It Implementation of framework with various geometry-based stylization techniques, which is demonstrated by different application examples.

3. It presents the concept of fused stylization techniques, which combines geometry-based and image-based stylization techniques.

The remainder of this paper is structured as follows. Section 2 reviews related work and present background to image-based and geometry-based abstraction techniques that represents the basis for our approach. Section 3 presents the design space and fundamental concept of geometry-based image abstraction techniques. Based on this, Section 4 demonstrate it application by means of different examples. Section 5 discusses the presented approach and describes future research directions. Finally, Section 6 concludes this work.

2 BACKGROUND

For the classification of stylization techniques, a comprehensive review exists in the area of NPR [Kyp13]. Kyprianidis et al. presented a taxonomy of artistic stylization techniques for images and video using Stroke-based Rendering (SBR). The algorithms are grouped by elementary artistic rendering primitive and their placement (e.g. regions, strokes, stipples, tiles).

Mould and Rosin developed a standard benchmark data set consisting of 20 photos [Mou17]. A list of image characteristics was taken into account, ranging from the level of detail to contrast and visual clutter.

Kumar *et al*. provided an extensive survey on NPR techniques such as image abstraction, artistic stylization, line drawing, engraving, color enhancement, pencil drawing, dithering, stippling, halftoning, hatching, and mosaicking [Kum19]. Furthermore, they developed a respective benchmark considering technology, algorithms, parameter and design.

NPR techniques have been successfully classified and have benchmarks. We now consider geometry-based techniques and their vectorized representation in more detail. Bertin's Visual Variables (1967/83) include seven properties, which distinguish one graphic object from another [Ber67]. The variables consist of shape, hue, value, position, size, orientation, and texture. They are used in cartographic design, graphic design, and data visualization. This research forms a starting point in the study of geometry-based image stylization, whereby mainly vector-based representations of image stylization are generated. For this we want to find shapes and other properties to optimize vectorization. Selinger described a polygon tracer and how the algorithm generates Bézier curves as vector outline for fonts or logos [Sel03].

There are only a few research about more complex things like a vectorization of an NPR technique. Glöckner *et al*. introduced an optimized vectorization of GPU-based stylized images using intermediate data representations [Glö20]. The approach allows interaction manipulation of parameters and support for various stylization pipelines.

Song *et al*. [Son08] proposed arty shapes and mixed it with NPR techniques [Son08]. Song et al. generates an image segmentation to be able fitting best shapes to each segment. Afterwards, an NPR technique such as

oil or crayon painting is applied. This approach supports the creation of synthetic artworks. Overall, this research demonstrates a contribution to FSTs, which we are further expanding with our contribution.

3 CONCEPT

This section first describes the notation of GSTs (Section 3.1), describes their semiotic aspects (Section 3.2), and briefly outlines a general approach towards a framework that is used for implementation (Section 3.3).

3.1 Geometric Stylization Techniques

A Geometric Stylization Technique (GST) is a type of stylization or abstraction of digital images by means of geometric primitives and their appearance attributes (e.g., color, outline, texture). Therefore, a number of geometric primitives are generated and can be created as a vector representation. These are particularly suitable for line-art production by a pen plotter, laser cutter, programmable embroidery machines or CNC carving. For feasibility, we implement several GSTs, including Pencil Hatching [Pra01], Scribble [Lo19], Voronoi Stippling [Sec03], and Shape Packing [Col03].

3.2 Semiotic Aspects

To characterize GSTs, we examined over 30 artists that use geometry abstraction characteristics in their artwork as well as more than 15 GSTs for characteristic properties. As a result, we identified eight major categories as semiotic aspects (Figure 3), which are briefly described in the following.

Shape Types: We distinguish in three different shapes. Besides points (e.g., in stippling), the shape type can also be one of three line types: single straight lines, poly-lines, and curves. A circular shape is thus represented by a curve. In addition, three different types of polygons are also used: convex polygons, non-convex polygons, and general polygons.

Outline: A geometric primitive can have an outline that can be regular or sketchy by style and comprise stippling patterns.

Fill: The fill of a geometric primitive can be solid using a single, have a gradient, or a texture. The latter two can be parameterized with respect to orientation, scaling, and offset transformations.

Shape Size: For the shape size, we basically distinguish between uniform and non-uniform shape sizes used within a single artwork.

Shape Orientation: Similar to size, the orientation of a geometric primitive is either the same (uniform) or different (non-uniform) within a single artwork.

Shape Type Mixture: There are GSTs that either consist of collection of the same shape types (uniform) or combine different ones (non-uniform).

Shape Placement: The coverage of geometric primitives among themselves can be distinguished into overlapping or non-overlapping.

Placement Approach: The application of operations can be applied local (e.g., by segmentation) or global for the entire artwork.

3.3 Data Processing Operations

The implementation of GSTs usually comprises a number of data processing operation. For the classification of such operations, we distinguish between two basic data representations they operate on: Vector (V) and Raster (R). Based on these two representations, four transformations can be performed: Vector-to-Raster, Raster-to-Vector, Raster-to-Raster, and Vector-to-Vector (Figure 4). In the following, the four main operations are described by input, output, and examples.

Vector-2-Raster (V2R): These type of operations use geometric primitives with appearance information as input and output a number of raster data layers. These are usually implemented using rasterization techniques or diffusion curves [Orz08].

Raster-2-Vector (R2V): These kind of operations implements the inverse process to rasterization. It takes raster data layers as input and computes geometric primitives with associated appearance information. Exemplary implementations are tracing/vectorization with and without intermediate representations [Glö20].

Vector-2-Vector (V2V): These operations enable transformations in the same vector space, e.g. primitive filtering, geometry amplification, tessellation, subdivision, or geometric mapping stages. Thus, input and output consist of geometric primitives with appearance information.

Raster-2-Raster (R2R): Similar to V2V, input and output of the operations consist of the same, only this time a number of raster data layers such as color, depth, surface orientation, segments, structure, or flow. These operations are used for image segmentation, Neural Style Transfer (NST), algorithmic stylization techniques, as well as blending or compositing.

Considering suitable input and output, the four transformations can be combined to obtain optimized results for the implementation of GSTs. However, these can

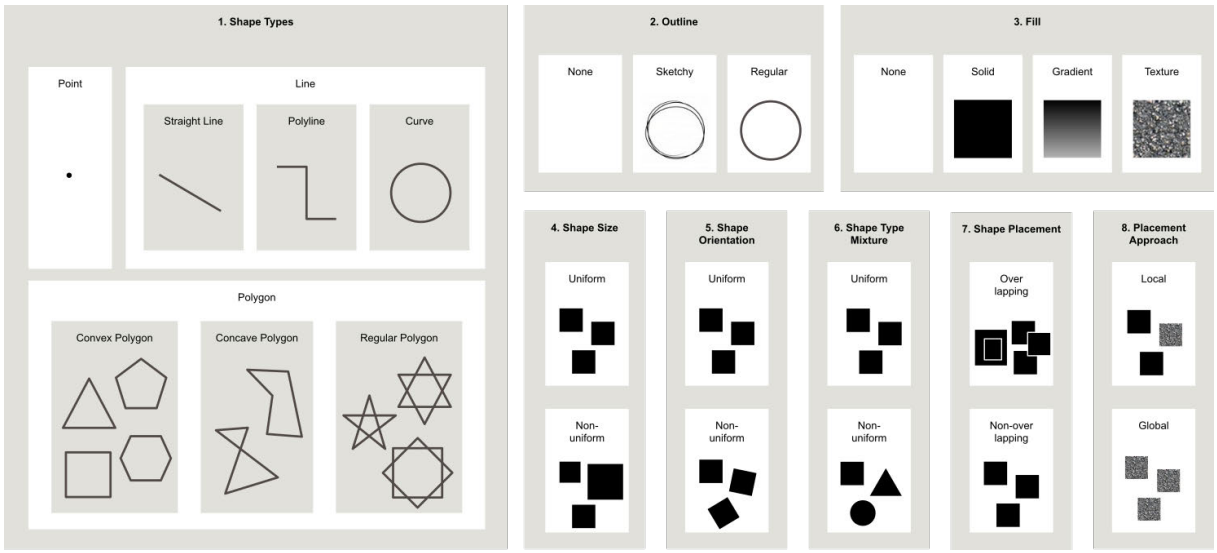


Figure 3: Summarizing presentation of the semiotic aspects of GSTs.

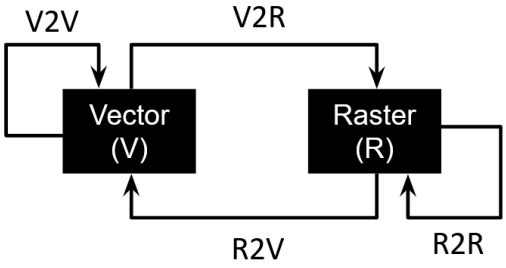


Figure 4: Classification of data processing operation between of vector and raster image representations.

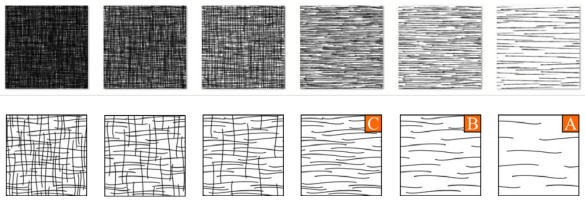


Figure 5: Top row: Conventional grid-based TAMs. Bottom row: new TAMs consisting of three line patterns (A, B, and C) only.

also form the basis for FSTs, i.e., the combination of GSTs and image-based stylization techniques [Kyp13]. This enables the implementation of a framework to imitate works by artists such as Kandinsky, Macdonald-Wright, Vasarely, Gleizes, Malevich, Feitelson, Miró, Picasso, Matisse, and Anderson.

4 APPLICATIONS

This section demonstrates the concept of GSTs by means of different application examples. For each example, we first describe the basic algorithm and following thereto the respective semiotic aspects.

4.1 Vectorized Pencil Hatching

Algorithm. The pencil-hatching approach by Webb *et al.* combines and aligns TAMs to simulate hatching strokes [Pra01]. To achieve vectorized stroke representations, we synthesized vectorized TAMs for enable clear lines in the vectorization step (Figure 5). These vectorized TAMs consist of three vector graphics, that yield another three variation by drawing lines.

For the vectorization, we generate line paths based on the edges in the image. The edges in the image were created with the help of our modified TAMs depending on the luminance in the input image. For the vector representation we support a color gradient of the lines (Figure 1). Therefore, we read luminance and color value of each node at position of input image and set the style of the path as linear gradient with luminance as opacity and individual color for each node. Listing 1 shows that stop-color specifies the color, stop-opacity the luminance, and offset the position on the path. The vectorized result consisting of lines can then also be produced with a plotter (Figure 6).

```
<svg ...>
  <defs>
    <linearGradient id="linearGradientvectorizeEdgePass">
      <stop style="stop-color:#e3da3d;stop-opacity:0.97;" offset="0"/>
      <stop style="stop-color:#d7c600;stop-opacity:0.48;" offset="1"/>
      <stop style="stop-color:#ebe022;stop-opacity:0.67;" offset="2"/>
      <stop style="stop-color:#e8dd43;stop-opacity:0.00;" offset="3"/>
      <stop style="stop-color:#e3da3d;stop-opacity:0.97;" offset="4"/>
    </linearGradient>
  </defs>
  <path
    id="vectorizeEdgePassCPU1"
    fill="none"
    stroke-width="3.0"
    stroke-linecap="round"
    style="stroke:url(#linearGradientvectorizeEdgePass)"
    d="M 5 1273.5 L 3 1254.5 L 4 1249.5 L 3 1235.5 L 1.5 1223.5 ">
  </path>
</svg>
```

Listing 1: Exemplary structure of a SVG using linear gradients to represent the results of a pencil hatching stylization.

```
for(segment in segments) do
{
 trials = TRIALS;
 shapes[] = generateShapes(numShapes, minSize, maxSize);

 for(shape in shapes) do
 {
  while(trials-- > 0) do
  {
   position = generatePosition()

   if(isInsideSegment(shape, position, segment)
      && !collisionPlacedShapes(shape, position))
   {
    placeShape(shape, position)
   }
  }
 }
}
```

Listing 2: Non-overlapping placement of shapes with varying dimensions in a limited space inside a segment.

Semiotic Aspects. Vectorized pencil hatching uses straight lines and poly-lines as shape type (Table 1). The outline comprises regular poly lines an no filled polygons. These lines are of different lengths and directions, thus the shape size and shape orientation is non-uniform. Since only lines are used, shape type mixing is characterized as uniform. The lines are often overlapping, thus the shape placement is overlapping. The algorithm is applied to the entire image, so the placement approach is global.

4.2 Segment-based Shape Packing

Algorithm. Shape packing defines the arrangement of shapes with or without overlapping each other [Col03]. For a segment-based approach, shape packing is applied locally in a confined space inside a segment (Listing 2). For the image segments, an instance segmentation was performed using the Mask R-CNN model trained on Microsoft Coco dataset (with 80 common object categories) [Lin14]. This segmentation was combined with conventional methods such as Mean Shift [Geo03], Watershed [Vin91], or DBSCAN [Est96] applied to the remaining input image to facilitate the segmentation quality. We tested the algorithm using different shape types, such as circular, rectangular, hearts, and other polygons. Per segment the appearance of the shapes can be different in size, frequency, color, and rotation (Figure 7).

Semiotic Aspects. This effect can also be classified according to the different semiotic aspects (Table 1). We have implemented and tested different shape types such as curved lines, convex polygons, as well as regular polygons. The outline appearance can be regular or none. As shape fill is usually solid or none. Settings such as gradients or textures can only hardly be reproduced by pen plotters. The shapes can be of different sizes or uniform, the same for the orientation and the mixing. The shapes should not overlap and the placement approach is local within the segments.

4.3 Scribbled Line-Art

Algorithm. The basic idea of the Scribbled Line-Art is to overlay multiple lines with on varying luminance (Listing 3). For this, random positions are calculated as the start position for the path. After that in a loop further points are calculated depending on the brightness in the image. This way a new point is created where it is the lowest luminance in the neighborhood of the pixel. This point is then added to the path with a slight rotation transformation. These points of the path result later in a line. Several of these lines stacked on top of each other then have the doodle effect. This approach produces a different result each time and is therefore part of the research field of generative art [Gal03]. Similar to Vectorized Pencil Hatching, we set per node of the path the color as it was in the input image. Thus we get a color gradient along the line (Figure 9). This vectorized result can now serve as the basis for fused stylization techniques, where we mix vector and pixel-based image stylization techniques. This is achieved by applying oil-painting, for example (Figure 9b).

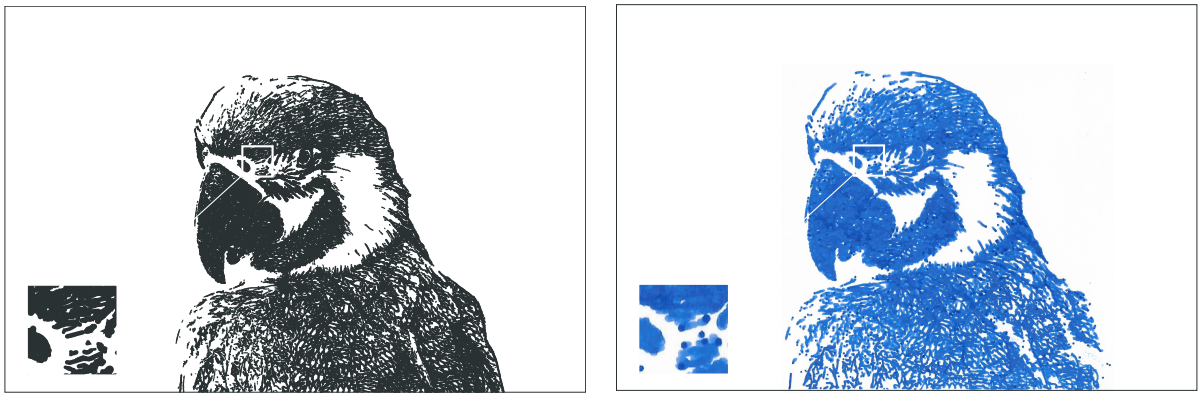
```
1  position = randomPosition();
2  addPositionToPath(position);
3  count = 0;
4
5  while(loopCount-- > 0) do
6  {
7   offset = calculateOffsetBasedOnNeighborsBrightness();
8   rotation = calculateRotationWithPerlinNoise();
9   position += offset * rotation;
10  count++;
11
12  if(count < MAX_COUNT)
13  {
14   addPositionToPath(position);
15   fadePixelsOfPath();
16  } else {
17   addPathAsLineItemAndResetCount();
18   findNextStartPosBasedOnBrightness();
19  }
20 }
```

Listing 3: Sample algorithm for scribbled line-art, which layers lines based on luminance.

Semiotic Aspects. For the semiotic aspects we use as shape type straight lines and poly-lines (Table 1). The outline is regular and the line has no filling. The lengths of the lines vary, therefore shape size is non-uniform. The lines have different rotations, so shape orientation is non-uniform. Only lines are used, so shape type mixing is uniform. The lines are superimposed, so shape placement is overlap. The placement approach is globally applied to the whole image.

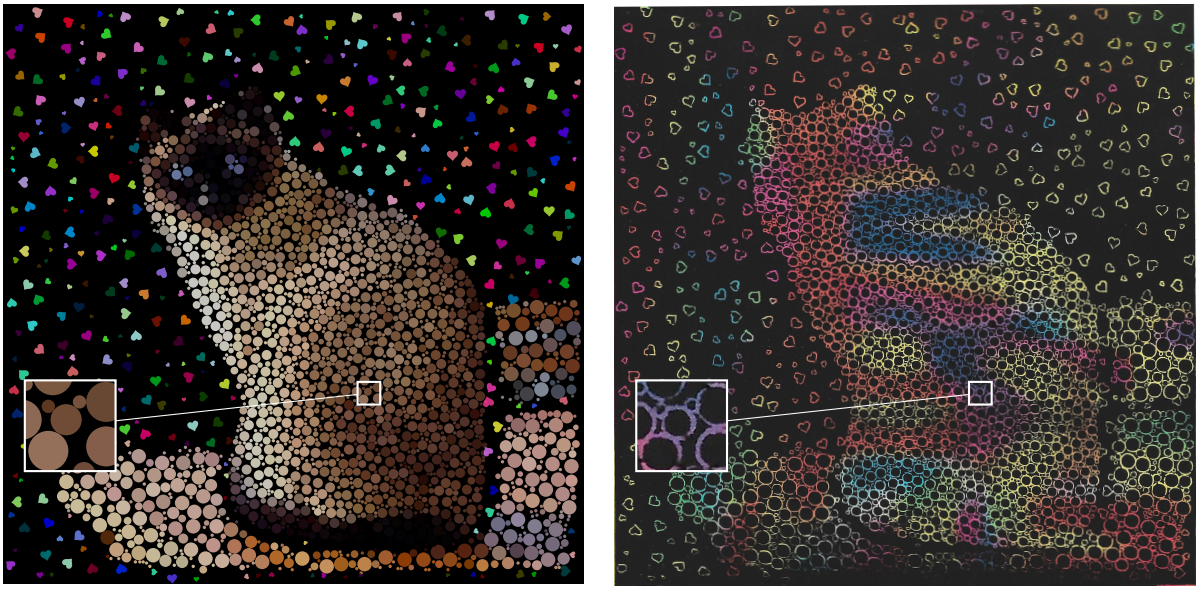
5 DISCUSSION

Evaluation. We categorized different GST according to our concept, classification and semiotic aspects. In our framework, we prototypically implemented different GST that can be combined for FST (Figure 10). In general, the classification worked well to distinguish between different GSTs. However, noticed that some cases the respective GST share the same classification



(a) Mono-colored vectorized pencil-hatching result. (b) Scan of pencil hatching using a Stabilo pen and silhouette plotter.

Figure 6: Comparison of vectorized pencil-hatching as digital and plotted result.



(a) Vectorized shape packing result with different elements. (b) Scan of scratch-art in shape packing style with silhouette plotter.

Figure 7: Comparison of vectorized shape packing as digital and plotted result.

results but the stylization computation show fundamentally different visual appearance. For example, in Table 1, there are the same selected categories for Section 4.1 and Section 4.3.

Preliminary considerations should be made for the particular GST to achieve optimum quality. Synthesis of shape types is not trivial, e.g., Section 4.1 requires new TAMs to make the line detection work. However, there are clear differences between raster-based pencil hatching and vector-based pencil hatching (Figure 8). For example, vectorized lines with gradients applied do not appear coherent and plausible compared to a TAM based on hand-drawn strokes and matched imprint of pencil on paper. Therefore, the vector file could be improved in post-processing by making it exportable with appropriate settings for different plotters. In addition, the line density due to the vectorized TAMs is not as dense as with the raster-based technique. A possible approach would be to add more layers of lines to fill the gaps.

Some materials and tools did not work well with the plotter for certain GST. For example, using scratch paper was not optimal for scribbled line-art because it overlays different lines. A scratch paper consists of only one black layer that covers a colored background. Additionally, the shapes sizes of Section 4.2 should not be set too small, to achieve a pleasing result.

Further, the plotter tools used have a fixed physical width that limits the production with the plotter. Thus, either the canvas size or the shape size need be to increased. The properties in the SVG representing the paths do not cover every desirable aspect, making the differences between raster-based and vector-based results obvious. To counterbalance this, workarounds that

	Shape Type						Outline			Fill				Shape Size		Shape Orientation		Shape Type Mixing		Shape Placement		Placement Approach	
	point	line			convex polygon	concave polygon	regular polygon	none	sketchy	regular	none	solid	gradient	texture	uniform	non-uniform	uniform	non-uniform	uniform	non-uniform	over-lapping	non-overlapping	global	local
		straight line	polyline	curve																				
4.1 Vectorized Pencil Hatching		•	•							•	•				•	•				•		•	
4.2 Segment-based Shape Packing				•	•		•	•			•	•	•	•	•	•	•	•	•	•		•		•
4.3 Scribbled Line-Art		•	•							•	•				•	•				•		•	

Table 1: Comparison of the presented GST example applications based on the presented semiotic aspects.

are specific for a GST are required. For pencil hatching example, the line segments could be grouped by pencil print, as this would be individually selectable especially for a silhouette printer.

Nevertheless, one would then have to draw each group subsequently to achieve the desired quality. However, that approach increases the reproduction time required. In general, there are various plotters on the consumer market, including home-built ones, that usually offer a smaller range of options. It would be desirable to specify different settings per GST for different plotter types.

Limitations & Future Research. As already mentioned, the plotter hardware with its tools and the material used is a major limiting factor with respect to the quality of the reprodcution. Therefore, some GSTs are not suitable for all types of production with the plotter. One solution here was to divide the file into several small parts. Further, we observed that FSTs are not suitable in all combinations.

In order to combine different GSTs and enhance the user experience, a separated stylization and layering of GSTs with parallel processing would be a possible extension. To improve or evaluate the described semiotic aspects of GSTs, further techniques could be implemented. Additionally, further FST can be implemented to imitate geometric abstraction artists. More plotters (manufacturers) should be tested to define settings in each case, so that the production can then provide the best result.

6 CONCLUSIONS

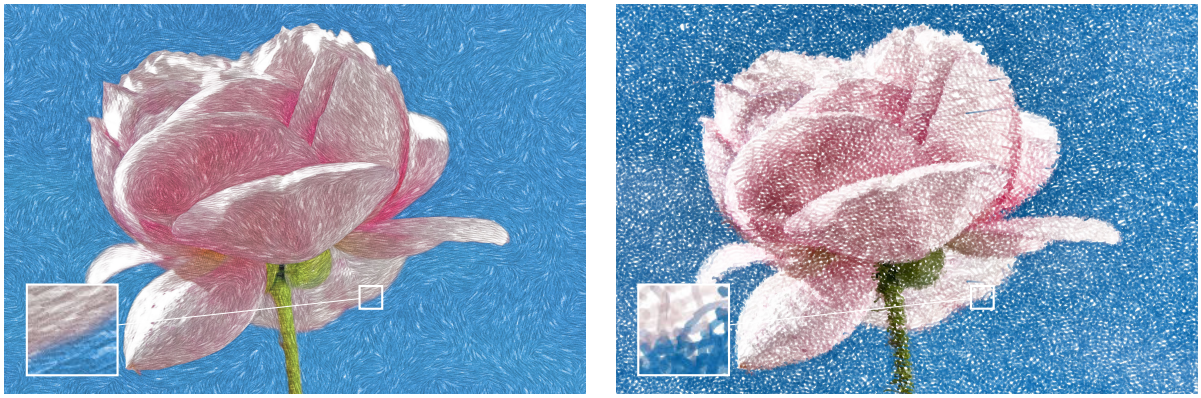
This paper presents the concept and semiotic aspects of geometry-based stylization techniques. To demonstrate its feasibility we developed a prototypical framework for implementation. We evaluate this framework by means of different application examples for geometry-based stylization techniques. By integrating image-based stylization techniques, the present approach enables the implementation of fused stylization technique, which denotes the combination of geometry-based and raster-based stylization techniques. The frame represent a basis for future research in image stylization.

ACKNOWLEDGMENTS

We thank the reviewers for their insightful comments. The project was supported by the Federal Ministry of Education and Research (BMBF), Germany (mdViPro, 01IS18092).

REFERENCES

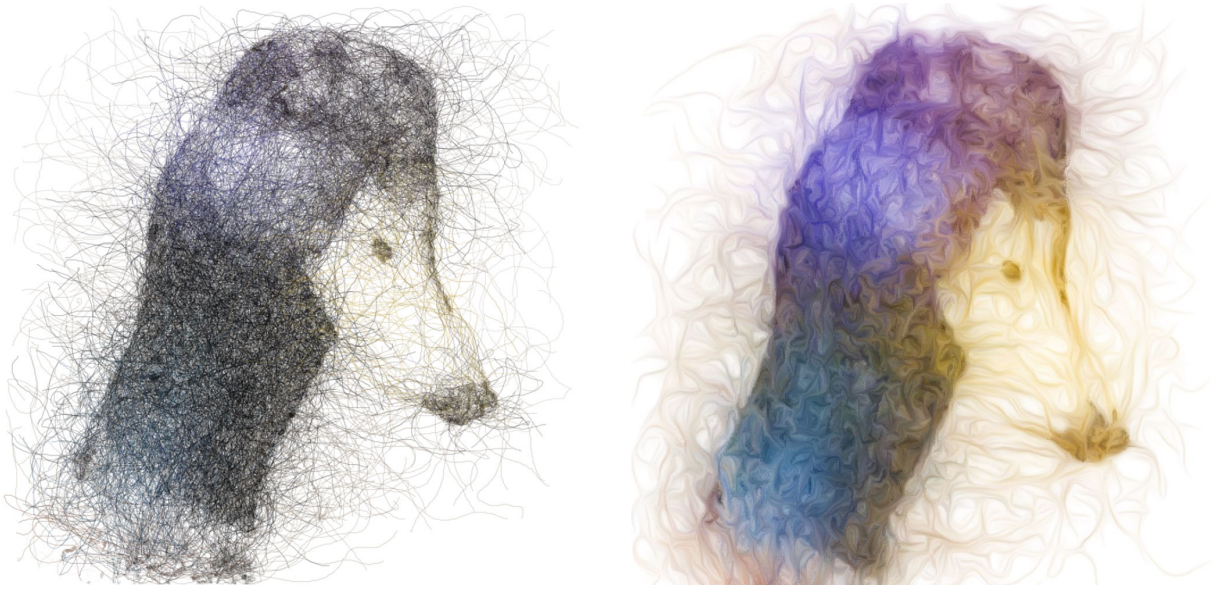
[Ber67] Jacques Bertin. “Sémiologie Graphique - Les diagrammes, les reseaux, les cartes”. In: *Geographical Journal* 135 (Jan. 1967). DOI: 10.2307/1795660.



(a) Raster-based Pencil Hatching Result.

(b) Vectorized Pencil Hatching Result.

Figure 8: Comparison of raster-based and vectorized Pencil Hatching. The vectorized lines with gradient are less realistic and the line density due to the optimized TAMs is not as dense as with the raster-based technique.



(a) Vectorized result of scribbled line-art of a duck.

(b) Applied Oil-painting on the vectorized scribble result.

Figure 9: Example of a FST that combines a scribbled line-art GST and an oil-painting image-based stylization technique.

[Bou06] Adrien Bousseau et al. “Interactive Watercolor Rendering with Temporal Coherence and Abstraction”. In: *Proceedings of the 4th International Symposium on Non-Photorealistic Animation and Rendering*. NPAR ’06. Annecy, France: Association for Computing Machinery, 2006, pp. 141–149. ISBN: 1595933573. DOI: 10.1145/1124728.1124751.

[Col03] Charles R. Collins and Kenneth Stephenson. “A circle packing algorithm”. In: *Computational Geometry* 25.3 (2003), pp. 233–256. ISSN: 0925-7721. DOI: https://doi.org/10.1016/S0925-7721(02)00099-8.

[Dev13] Kapil Dev. “Mobile Expressive Renderings: The State of the Art”. In: *IEEE Computer Graphics and Applications* 33.3 (2013), pp. 22–31. DOI: 10.1109/MCG.2013.20.

[Est96] Martin Ester et al. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: vol. 96. Jan. 1996, pp. 226–231.

[Gal03] Philip Galanter. “What is generative art? Complexity theory as a context for art theory”. In: Jan. 2003.

[Geo03] Bogdan Georgescu, Ilan Shimshoni, and Peter Meer. “Mean Shift Based Clustering

in High Dimensions: A Texture Classification Example". In: vol. 1. Nov. 2003, 456–463 vol.1. ISBN: 0-7695-1950-4. DOI: 10.1109/ICCV.2003.1238382.

[Glö20] D.-Amadeus J. Glöckner et al. "Intermediate Representations for Vectorization of Stylized Images". In: *J. WSCG* 28.1-2 (2020), pp. 187–196.

[Kum19] Mp Kumar et al. "A comprehensive survey on non-photorealistic rendering and benchmark developments for image abstraction and stylization". In: *Iran Journal of Computer Science* 2 (Sept. 2019). DOI: 10.1007/s42044-019-00034-1.

[Kyp13] Jan Eric Kyprianidis et al. "State of the "Art": A Taxonomy of Artistic Stylization Techniques for Images and Video". In: *IEEE Transactions on Visualization and Computer Graphics* 19.5 (2013), pp. 866–885. DOI: 10.1109/TVCG.2012.160.

[Lin14] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: vol. 8693. Apr. 2014. ISBN: 978-3-319-10601-4. DOI: 10.1007/978-3-319-10602-1_48.

[Lo19] Yi-Hsiang Lo, Ruen-Rone Lee, and Hung-Kuo Chu. "Generating Color Scribble Images using Multi-layered Monochromatic Strokes Dithering". In: *Computer Graphics Forum* 38 (May 2019), pp. 265–276. DOI: 10.1111/cgf.13636.

[Mee20] Jayananden Meenakchisundaram. *Review on Building A Cost Efficient Pen Plotter*. Oct. 2020.

[Mou17] David Mould and Paul Rosin. "Developing and applying a benchmark for evaluating image stylization". In: *Computers & Graphics* 67 (June 2017). DOI: 10.1016/j.cag.2017.05.025.

[Mue15] Stefanie Mueller and Patrick Baudisch. "Laser cutters". In: *interactions* 22 (Aug. 2015), pp. 72–74. DOI: 10.1145/2811292.

[Orz08] Alexandrina Orzan et al. "Diffusion Curves: A Vector Representation for Smooth-Shaded Images". In: *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*. Vol. 27. 2008.

[Pra01] Emil Praun et al. "Real-Time Hatching". In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 581. ISBN: 158113374X. DOI: 10.1145/383259.383328.

[Sec03] Adrian Secord. "Weighted Voronoi Stippling". In: *Proc. of the 2nd Int. Symp. on Non-photorealistic Animation and Rendering* (Mar. 2003). DOI: 10.1145/508530.508537.

[Sel03] Peter Selinger. "Potrace : a polygon-based tracing algorithm". In: 2003.

[Sem161] Amir Semmo et al. "Image stylization by interactive oil paint filtering". In: *Computers & Graphics* 55 (2016), pp. 157–171. ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2015.12.001.

[Sem162] Amir Semmo et al. "Interactive Image Filtering with Multiple Levels-of-Control on Mobile Devices". In: *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications*. SA '16. Macau: Association for Computing Machinery, 2016. ISBN: 9781450345514. DOI: 10.1145/2999508.2999521.

[Sem20] Amir Semmo and Sebastian Pasewaldt. "Graphite: Interactive Photo-to-Drawing Stylization on Mobile Devices". In: *ACM SIGGRAPH 2020 Appy Hour*. SIGGRAPH '20. Virtual Event, USA: Association for Computing Machinery, 2020. ISBN: 9781450379656. DOI: 10.1145/3388529.3407306.

[Son08] Yi-Zhe Song et al. "Arty Shapes." In: Jan. 2008, pp. 65–72. DOI: 10.2312/COMPAESTH/COMPAESTH08/065-072.

[Vin91] Luc Vincent and Pierre Soille. "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 13 (Jan. 1991), pp. 583–598.

[Win06] Holger Winnemöller, Sven C. Olsen, and Bruce Gooch. "Real-time Video Abstraction". In: *ACM Trans. Graph.* 25.3 (July 2006), pp. 1221–1226. ISSN: 0730-0301. DOI: 10.1145/1141911.1142018.

Figure 10: Results produced with the proposed system for geometry stylization techniques: vectorized pencil hatching, segment-based shape packing, scribbled line-art, and combinations. The input images are obtained from the benchmark of Mould and Rosin.