Vision, Modeling, and Visualization (2023)
T. Grosch and M. Guthe (Eds.)

Greedy Image Approximation for Artwork Generation via Contiguous Bézier Segments

J. Nehring-Wirxel[1] and I. Lim[1] and L. Kobbelt[1]

[1] RWTH-Aachen University, Visual Computing Institute

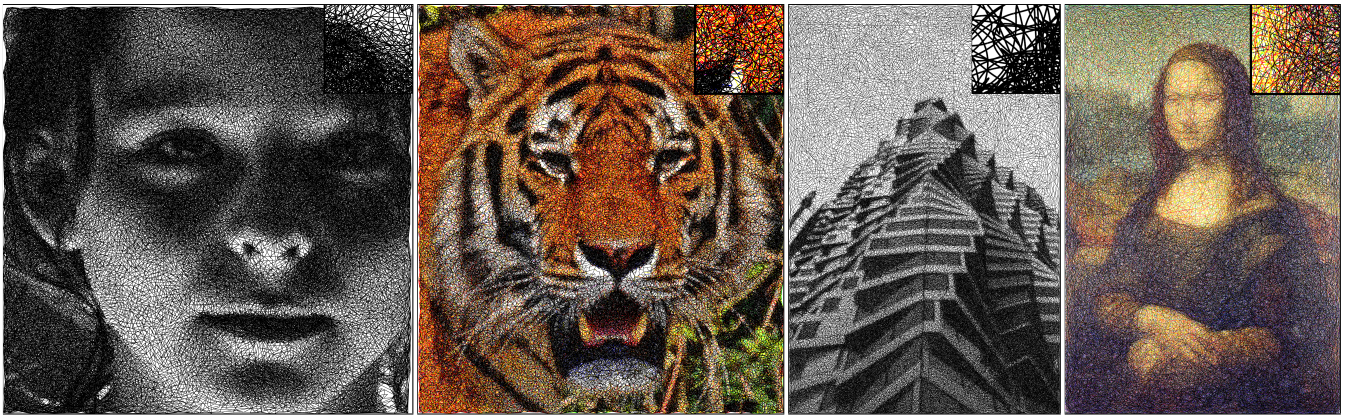


Figure 1: *With our greedy approximation approach we create one or several mono-colored, continuous lines that, when viewed at close range, seem chaotic and random, but reveal meaningful image content when viewed from a distance. The left two images are inspired by work of the artist Kumi Yamashita, while the right two pieces follow the* scribble *art style.* ***Please use the digital version to zoom into the images to observe individual lines.***

Abstract

The automatic creation of digital art has a long history in computer graphics. In this work, we focus on approximating input images to mimic artwork by the artist Kumi Yamashita, as well as the popular scribble art style. Both have in common that the artists create the works by using a single, contiguous thread (Yamashita) or stroke (scribble) that is placed seemingly at random when viewed at close range, but perceived as a tone-mapped picture when viewed from a distance. Our approach takes a rasterized image as input and creates a single, connected path by iteratively sampling a set of candidate segments that extend the current path and greedily selecting the best one. The candidates are sampled according to art style specific constraints, i.e. conforming to continuity constraints in the mathematical sense for the scribble art style. To model the perceptual discrepancy between close and far viewing distances, we minimize the difference between the input image and the image created by rasterizing our path after applying the contrast sensitivity function, which models how human vision blurs images when viewed from a distance. Our approach generalizes to colored images by using one path per color. We evaluate our approach on a wide range of input images and show that it is able to achieve good results for both art styles in grayscale and color.

CCS Concepts

• ***Computing methodologies*** → ***Non-photorealistic rendering;*** • ***Applied computing*** → *Media arts;*

1. Introduction

One of the many aspects of art is that artists will constrain themselves in the creation of their art. For example, the Japanese artist Kumi Yamashita [Yam13, Yam16] creates portraits by placing nails on a white board and winding a single black thread tightly around the nails. By varying the density of the black thread, different regions of the image are perceived to have different shades of gray, allowing the creation of what is essentially a grayscale image. Using multiple continuous strings, each with a different color, Yamashita

DOI: 10.2312/vmv.20231234

has also created a colored string art piece [Yam16]. Some of her artwork is shown at her website http://kumiyamashita.com/portraits.

Similarly, the art style commonly referred to as *scribble art* has analogous constraints. Here, the works are usually created by the artist by placing the pen (or often pencil) on the paper only once and drawing the image with one single, continuous stroke. When viewed from a close range, the strokes are placed seemingly at random, giving it a dynamic, chaotic look. But when viewed from a distance, the image content becomes clear. Both Yamashita's work, and the scribble art style have in common, that they are created using a single, continuous path that locally seems to be almost random.

Although the scribble art style is well established, and a commercial business [Wir23] even fabricating similar pieces for the Yamashita art style, there is a notable relative scarcity of scientific research focused on the automated creation of such works.

In this work, we present how to create paths that approximate the style used for Yamashita's works, as well as the scribble art style. More specifically, the input is a rasterized image, either in grayscale, or color, and the output is one or more mono-colored paths that depict the content of the input image in one of the two styles. This enables the creation of highly customizable artistic pieces even for those with little artistic experience or skill.

In a mathematical sense, this can be formulated as a challenging optimization problem, where the goal is to minimize the difference between the input image and a rendered image of the output path subject to some style specific constraints. Instead of trying to exactly solve this optimization problem, we propose a greedy approximation. After an initialization, we select the best path continuation from a set of candidate extensions to our path. Each candidate extension is already subject to constraints imposed by the corresponding art style. Because the image content is only recognizable when viewed from a distance, we model the human vision system response by applying the *contrast sensitivity function* (CSF) [MK88, JF03] to both the input image and a rasterization of the current path image including the candidate. This allows us to then compute a pixel-wise distance measure in a suitable color space between both images to determine which candidate offers the best approximation. By creating multiple paths (each with a different corresponding single color), our method is able to also produce colored images, too. To summarize, we contribute:

- An algorithm to greedily approximate a variety of continuous stroke-based art styles
- A constraint-aware sampling strategy to incrementally create constrained paths
- The approximation of both grayscale and colored images

2. Related Work

Halftoning

In the field of computer graphics, there has been extensive work on how to represent images when only a few discreet colors are available. By varying the size, density, or pattern of discreetly colored dots or stripes, halftoning [Uli87, MP92, LAG98, LA18], stippling [PFS03, MARI17], or hatching [ZISS04, PHWF01] methods allow the creation of images that nevertheless appear to have continuous color levels when viewed from a distance. Similarly, in our method a set of discretely colored paths is created that only combine to shape a colored image when viewed from a distance.

Style Transfer

Since we create Yamashita- or scribble-style images, our method is related to the field of *style transfer*. Classical approaches like [EF01, HJO*01, PHWF01] work by recreating an image by placing patches from another — possibly synthetic — image onto another. Nowadays, a large number of works exist that approximate an input image in a particular style using neural networks. An overview of which can be found in [JYF*20]. However, in this paper we propose a method that approximates an input image by sequentially generating a path under a set of constraints. We will therefore focus on previous work that also generate paths for the automatic synthesis of various art-styles.

Path-based Methods

Our proposed method falls into the category of *path-based* methods. Here, a single, continuous path is generated to approximate an input image. Kaplan and Bosch [KB05] propose a method where the path is created as the solution of a traveling salesperson problem (TSP). Nodes of the graph are first distributed according to a dithering algorithm (an overview can be found in [Uli00]), which already produces a reasonable approximation of the input image. Using a standard TSP solver, they then connect the nodes via a single, never intersecting path of linear segments, which is used to represent the input image. Extending this method to produce images in the scribble art-style, Chiu et al. [CLLC15] initially create a TSP path and then transform the resulting linear segments into a scribble image. For this, they model a rotating disk that traverses the TSP path at the disk center with a virtual pen attached to the border of the disk. In addition to the node density, they modulate the radius, velocity, and normal of the disk along the path to approximate the local luminance of the input image. Lo et al. [LLC19] further improve upon this by first selecting a set of representative colors from the target image and then drawing multiple colored layers of the scribbles with [CLLC15] to create scribble art.

More in the context of actual physical fabrication, Birsak et al. [BRWM18] propose a method that creates artworks from a single, continuous string. To this end virtual nails are placed on a circular border. Their algorithm then greedily adds or removes linear segments between the nails on the boundary until the perceived image cannot be approximate the input any better. The resulting path can then be manufactured using a robot. The authors show, however, that the approximation quality cannot be uniformly distributed across the image. Therefore, they propose that saliency maps are used as additional input by the user to specify regions where good approximations are desirable.

Note that there exist commercial products in this field. The *WireStyle GMBH* [Wir23] automatically creates artworks that are similar to the pieces created by the artist Yamashita [Yam13]. They insert a set of nails into a white plastic board and then span a single black thread across these nails to approximate customer images.

Our method differs from these works in several ways. In the

spirit of Yamashita [Yam13] we distribute the nodes that are to be visited uniformly and do not rely on a dithering step to approximate the input image. We are able to do so, because unlike in [KB05, CLLC15, LLC19], our method aims to compute and minimize an image approximation error. Furthermore, the method by Birsak et al. [BRWM18] is restricted to circular placements of nodes. Similarly to Lu et al. [LLC19], our method is able to produce colored images. However, our method does not rely on selecting a different set of colors per input image. Rather, by making use of the image approximation error, our approach is able to reproduce different images with the same set of colors in various art styles.

3. Image Approximation Algorithm

Since our algorithm handles two distinct art styles, we will first give an overview of how it works in general, including a problem definition. We will then go into more detail on how to customize it for the Yamashita and scribble art styles.

3.1. Problem Definition

The general setup can be formulated as a global optimization problem: The goal is to approximate an input image for a given canvas size and viewing distance with a single or multiple paths that are at least C^0 continuous. In the case of the Yamashita style this path consists of contiguous line segments and for the scribble art style it is a smooth curve. For the sake of simplicity, let us start by assuming that we have just a single black path on white background, which is the use-case for grayscale images. The task is then: Find a path that minimizes some error measure between the target image and the image created by rasterizing the path subject to possible additional constraints that vary depending on the art style.

3.2. Greedy Approximation

Consider the setting (similar to Yamashita) where the possible endpoints of line segments are restricted to a fixed set of *nail* positions. We then have to find a contiguous path that visits these positions in such a manner that the perceptual difference to the input image is minimized. In this case, finding the best path is a challenging combinatorial problem. Since it is not feasible to exactly solve the task, our algorithm employs a greedy strategy by building the path in an iterative manner:

1. Initialize the path with a *good* starting position
2. At the current position, sample a set of path candidates that extend our path according to style specific constraints
3. Evaluate a global error measure for each path candidate
4. Select the best candidate if it improves the global error
5. Repeat steps 2. to 4. until termination criteria are met

3.3. Multi-colored Images

To extend the approximation to use multiple colors, we add one path per used color. We then perform the initialization for every path color and advance the paths in a round-robin fashion. In our setting, we assume that the last drawn curve determines the color at its position. In other words, there is no blending between the top-most curve, and curves that have previously crossed the same position. For the optimization problem, this further increases the space of possible configurations: There is a combinatorial degree of freedom, in which order to draw different lines that cross the same position. For our greedy approximation, we do not explicitly handle this degree of freedom. However, because the curves are extended in a round-robin fashion, it is possible for a curve to visit the same region again after it has been visited by another path color.

3.3.1. Color Choice

For a given point, the perceived color when viewed from a distance is a linear combination of the colors of the surrounding region. This in combination with no per-pixel blending of colors means that if we want to be able to represent the entire RGB cube, we need to be able to represent each of its corners. In practice that means, that for a *true color* mode we have one color represented by the background, and seven paths, each with the color of one of the remaining seven corners. In our experiments, we chose a white background and have colors in red, green, blue, cyan, magenta, yellow, and black.

3.4. Yamashita Style

To imitate the art style used by Yamashita, we must have a closer look at how her artwork is created: For grayscale images, a single, continuous thread is tightly spun around a set of nails that have previously been nailed into a white wooden board that serves as background. The artist also created colored work, where multiple strings, each with a different color, are used in the same manner. Together, they blend into different shades of colors when viewed from a distance. Regarding our method, we can derive at least two constraints:

- We can only connect lines to a discrete set of nail positions
- Each line must be a single, linear segment, since Yamashita tightly wraps the string around two nails to connect them

To synthesize images of the same style, we emulate this manual process.

3.4.1. Nail Generation

We start by generating a set of nails that stay fixed for the remainder of the process. In Yamashita's work, the nails are evenly, but not regularly distributed, similar to Blue noise or low-discrepancy sampling [APC*16]. For similar results, we employ a stratified sampling strategy. This means that we divide the image into a regular grid and sample a nail position per grid cell. The resolution of the grid is a user parameter. In Figure 7, we show the same image with different grid sizes. While placing the nails in a uniform random fashion in each grid cell gives a good pseudo-uniform sampling of the image plane, we found that the image approximation improves, if nails are placed on the pixel with maximal gradient magnitude in each grid cell. The intention is that important features are aligned with the gradient of the image. Placing nails on the gradient maxima allows string connections along feature lines that cross cell boundaries.

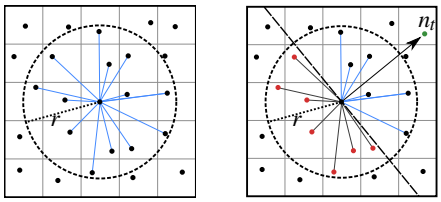


Figure 2: *For the Yamashita style, (a) we evaluate all lines inside the fixed radius r and choose the best one, if it improves the global score. If there is no valid candidate, (b) we choose the line that produces the least amount of error and decreases the distance to the closest nail n_t that must yet be visited by the color of the current path. The red nails do not decrease the distance to n_t and are therefore not considered in this step.*

3.4.2. Sampling

Our general algorithm can now be followed as presented in Section 3.2: In each iteration, we sample a set of candidate edges that connect the path from the current nail with other nails in a local neighborhood as shown in Figure 2a to keep the runtime complexity low. Since we have a discrete set of edges for a given neighborhood r, we query all neighboring nails and create a candidate for each. We ignore edges that have already been drawn to avoid going back and forth between two nails infinitely. We select the candidate edge that improves the global error the most and add the edge to the path.

3.4.3. Forced Color Positions

Our greedy strategy can cause the path search to be stuck in local minima, where no edge within r can be found that decreases the global error. We offer a fallback strategy to escape the minimum: During the algorithm initialization, we determine per colored path P_i a set of nails S_i that this path is forced to visit with the idea that we know that the path's color is required in the region surrounding the nail. Each time no segment candidate, which improves the error, is found, we pick nail $n_t \in S_p$ that is closest to our current nail n_c and has not been visited by the path P_i yet. Among the original segment candidates, we then only consider those, that lie closer to n_t than n_c, ignoring those would increase the distance towards n_t (red nails in Figure 2b) and append the candidate that produces the least amount of error. Note, that it is possible that each fallback step may be enough to move the path out of the local minimum, and we can thus consider all candidates during the next step again.

To decide if nail n should be visited by path P_j, we need to determine if P_i's color c_i is required in the region surrounding n. The Voronoi cell of each nail is a natural choice for its local neighborhood. For each n, we compute the average RGB color c_n of the pixels inside the Voronoi cell corresponding to n. Then, we split the RGB cube into six tetrahedra and determine which of them contains the cell's average color. This gives us four colors $c_{t_j}, j \in 1..4$ (corresponding to the corners of the tetrahedron) that can be used to linearly interpolate the average color. We then compute the corresponding Barycentric coordinates b_j of the average nail color inside the containing tetrahedron. Because we split the color cube into tetrahedra, the Barycentric coordinates are guaranteed to be unique and non-negative. Each coordinate b_j also gives us the ratio of how much color c_{t_j} should appear in the region surrounding P_{t_j}. To translate this into a binary decision, we apply a stochastic approach: For every c_{t_j}, we mark the nail as a necessary target location for the corresponding colored path by drawing a sample $u_j \sim \mathcal{U}(0,1)$, where $\mathcal{U}$ is the uniform distribution. If $u_j \leq b_j$, then the nail has to be visited by the respective color. If no marks were assigned, we force the colored path with the maximal Barycentric coefficient to visit the nail. This ensures that every nail is visited at least once in the spirit of Yamashita.

3.5. Scribble Art Style

With the scribble art style, artists create images that drawn using just a single, contiguous stroke. Commonly, this is done with black ink or pencil on white background. When looking closely at an artwork, the local structure of the strokes may look chaotic, like randomly *scribbled* curves on a sheet of paper. However, similar to the setting of Yamashita, when viewed from a distance, the strokes form a meaningful image by human perception. In this setting we can relax the constraints introduced for the Yamashita style somewhat. We no longer require that the path has to connect a predetermined discrete set of nail positions. Furthermore, the path is no longer restricted to consist of only linear segments.

To model this art style, we use smoothly connected Bézier curves to represent the contiguous strokes. They emulate the shape of manual strokes well, and it is easy to sample new Bézier curves that fulfill possible continuity constraints w.r.t. their predecessor. We choose to use Bézier curves of degree three since they offer a good compromise between complexity and geometric flexibility.

3.5.1. Continuity

Two consecutive Bézier curves $B_i(t)$ and $B_{i+i}(t)$ are connected $\mathcal{C}^n$ continuously, if $\frac{d^k}{dt^k}B_i(1) = \frac{d^k}{dt^k}B_{i+i}(0)$ for $k \in \{0,\dots,n\}$ [Far02]. This means that when $B_i(t)$ is given, some control points of $B_{i+i}(t)$ are already fixed according to the chosen continuity constraint. For two contiguous Bézier curves of degree three with the control points (p_0,p_1,p_2,p_3) and (q_0,q_1,q_2,q_3) the following constraints hold when the p_i are fixed:

- $\mathcal{C}^0$: $q_0 = p_3$
- $\mathcal{C}^1$: $q_1 = 2p_3 - p_2$
- $\mathcal{C}^2$: $q_2 = 4(p_3 - p_2) + p_1$

We found that $\mathcal{C}^2$ are too constrained. Since the first three control points are already fixed, the curves will often be forced to have a large extend, and often leave the canvas. Since hard corners at the connection are undesired for this art style, we do not use $\mathcal{C}^0$ curves Thus, for our approach we employ $\mathcal{C}^1$ curves.

3.5.2. Candidate Sampling

To generate $\mathcal{C}^1$ continuous Bézier curves of degree three, the first two control points q_0 and q_1 are fixed, leaving us to decide where to place q_2 and q_3. Since we do not have the constraint of visiting a discrete set of nail locations as with the Yamashita style, we are free to sample these 2D positions in a continuous manner (see

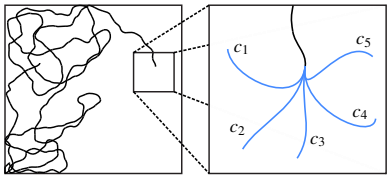
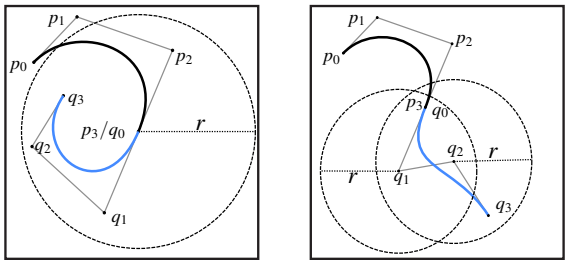


Figure 3: *During each iteration of our algorithm, a set of new candidate segments c_i are generated. Each is evaluated regarding an error function. The candidate with the minimal error is added the image if it improves the global image error.*



(a) **(b)**

Figure 4: *To sample new $\mathcal{C}^1$ continuous Bézier candidates, we offer two sampling strategies. (a) We sample both free control points q_2 and q_3 inside a circle surrounding q_0. This forces the generation of more localized lines that stay in the approximate area its predecessor. (b) Alternatively, we sample each control point in a circle surrounding the previous control point.*

Figure 3). To this end we propose two sampling strategies that are shown in Figure 4 that both produce scribble art curves, but with largely varying style. Both methods have a user parameter r that defines a radius in which new control points can be sampled.

First-Control-Point: In Figure 4a, we uniformly sample both q_2 and q_3 in same disk surrounding q_0 with radius r, i.e. $\|q_2 - q_0\| \leq r$ and $\|q_3 - q_0\| \leq r$. This produces curves that are more localized.

Previous-Control-Point: Alternatively (Figure 4b), we sample each control point uniformly in a disk centered at the previous control point, i.e. $\|q_2 - q_1\| \leq r$ and $\|q_3 - q_2\| \leq r$.

3.5.3. Initialization

We initialize the first curve by randomly sampling Bézier control points without continuity constraints as there is no predecessor. For this, we iteratively sample batches of curves, and take the best curve from the first batch that improves the error metric.

3.5.4. Termination

Similar to Section 3.4.3, it can happen that none of the sampled curve candidates improve the global approximation error. In this case, we increase the sample radius r and resample. We continue to increase r until the entire image plane is covered by the sampling circle or a suitable candidate is found. In the latter case, r is reset to its original value. This procedure allows us to escape local configurations where the approximation is already good, and move to areas further away where improvements can still be made. The algorithm terminates if no candidates where added after a fixed number of iterations.

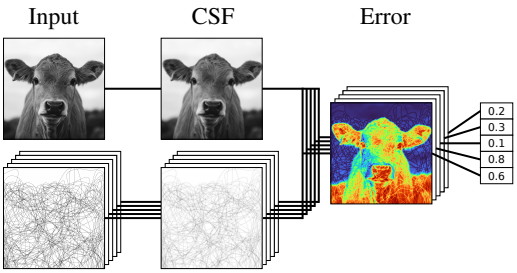


Figure 5: *To append to an existing path, we generate a set of candidates c_i. Each candidate is added to the current image state. Then the CSF is applied to both the input image and each candidate image to model human perception. Finally, we evaluate a global approximation error to determine the quality of each candidate.*

4. Error Function

For both art styles we need to determine the quality of the input image approximation in order to evaluate and select the best path continuation from a set of candidates. To this end, we rasterize the candidates using the same resolution as the target image. Since the portrayed objects only become recognizable when viewed from a distance, our error computation should take into account the input image resolution, the physical size of the produced images, and the distance from which they are meant to be viewed. Therefore, we need to emulate how the human vision system processes images of a certain size when viewed from a given distance. Similar to the work by Andersson et al. [ANAM*20], we make use of the *contrast sensitvity function* (CSF) (see [MK88, JF03]), to model the system response of the human vision system, which allows us to consider the various image and viewing parameters mentioned above. For our implementation, the evaluation of the CSF if based on ꟻLIP [ANAM*20].

Thus, to evaluate the approximation error (cf. Figure 5) for a curve candidate we apply the CSF to both the input image and a rasterized image containing all curves up to this point including the candidate curve. Then, we convert both images to the CIELab [JF03] color space, as it is specifically designed such that Euclidean distances in this space are equivalent to perceived distances between colors. Finally, we compute the mean absolute color difference between both images as the global approximation error.

Finally, we need to address gamma correction: Since the colors of the paths are corners of the RGB cube, i.e. all RGB pixel values are either 0 or 1, applying gamma correction to our output image has no effect. To account for that, we instead approximate the gamma corrected target image.

5. Evaluation

We evaluate our method on a wide range of input images as shown in Figure 6. Unless stated otherwise, we create 100 candidate seg-

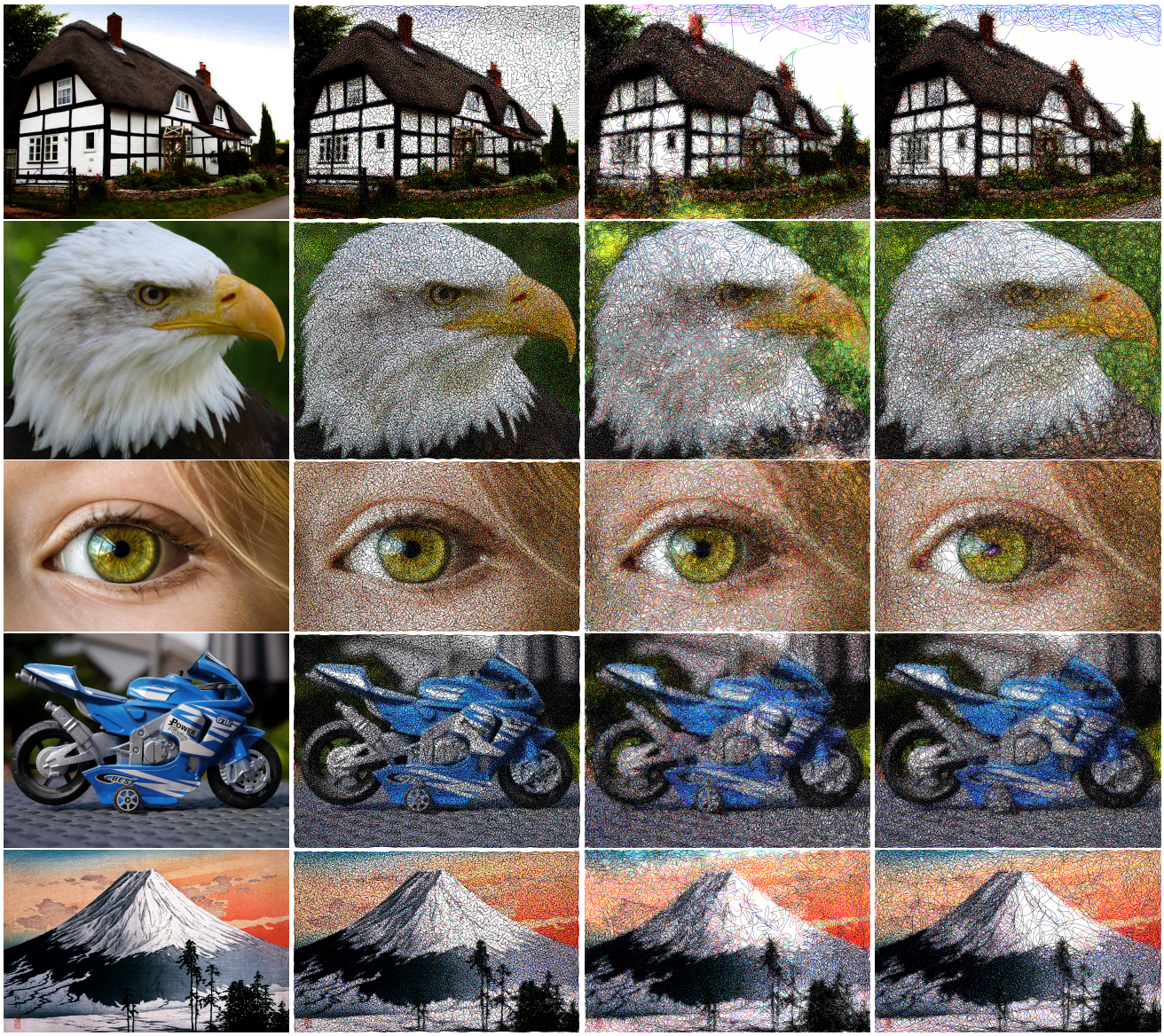
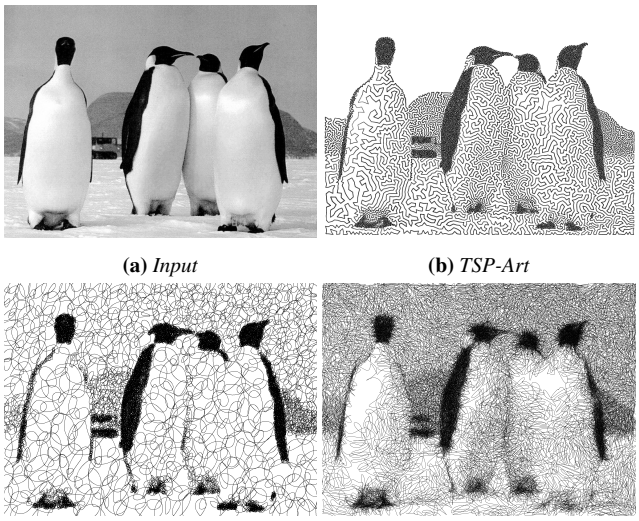


Figure 6: *From left to right: Input image, Yamashita style, scribble sampling around the first control point, scribble sampling around the previous control point.*

(a) *Input* **(b)** *10880 nails* **(c)** *2688 nails* **(d)** *1176 nails* **(e)** *672 nails*

Figure 7: *For the Yamashita style, estimating the same picture with a decreasing number of nails leads to much poorer approximation quality.*

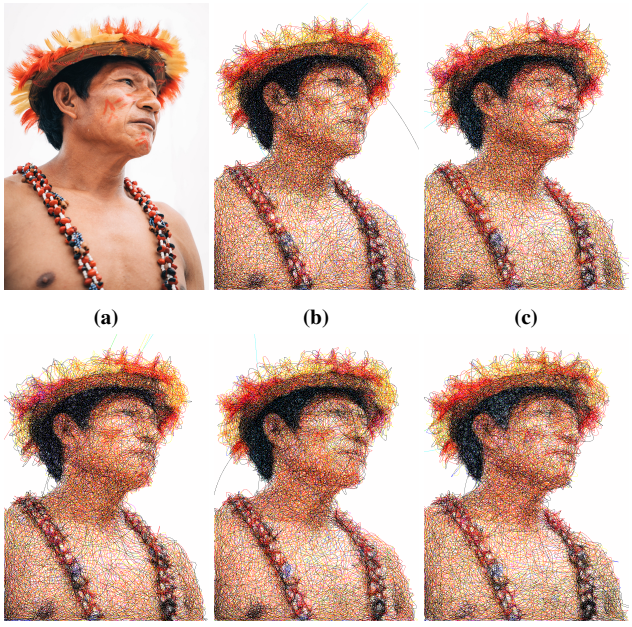


(a) *Input* **(b)** *TSP-Art* **(c)** *Circular Scribble* **(d)** *Ours Scribble*

Figure 8: *The same image (a) as input to (b) TSP-Art [KB05] (with the background removed), (c) Circular Scribble Art [CLLC15], and our approach using the scribble variation.*

ments per path per iteration for the scribble style, with a search radius of 2.5% of the image diagonal. For the Yamashita style, we choose the search radius to be five times as large as the grid resolution in which the nails are placed. The corresponding search circle covers the area of approximately 78 grid cells which also corresponds to the average number of candidates per path per iteration. The Yamashita style generally offers the best approximation quality. In some cases, especially in white regions (see the sky in the top row in Figure 6) the approximation is slightly too dark, as we force each nail to be visited by at least one path. For both our scribble styles the input images are well approximated with a few suboptimal patches. For example, the chimney in the top row for the scribble-first style is poorly approximated. The reason here is that the nature of the Bézier curves makes it more difficult to escape undesirable local minima when compared to the Yamashita style (cf. Section 3.4.3).

In Figure 8, we show a comparison with the TSP-based scribble algorithms [KB05, CLLC15]. Please note that for the TSP-Art, the background sky has been removed. We also compare the approximation via colored scribbles [LLC19] to our approach in Figure 12. In Table 1, we show how our CSF-based error function rates results from [LLC19] (the only other relevant method to produce color images) and our methods on the same set of images. Especially for the Yamashita style, the error is lower in most cases. The reason it is higher in some cases is due to poor approximations in regions near the border as shown in Figure 10. It can happen that the scribble approximation terminates too early for images with only a sparse set of foreground pixels. An example of this is shown in Figure 11. In this case, we can still produce a good approximation by assigning a weighting to the approximation error of each pixel. Here, black pixels that mainly define the contour are assigned a much larger weight than the white pixels. Generally, our approach reliably produces good results. In Figure 9, we show multiple outcomes for the same input image for the scribble style sampling around the previous control point. The chosen seeds for the random number generator are the successive numbers 0 to 4. To show the effect of the grid resolution, the same image is approximated with a varying grid resolution for the initial nail placement in Figure 7. The shown grid resolutions are 10, 20, 30, and 40 pixels per grid cell edge, resulting in the given number nails. It is clearly important to have high nail density for a good image approximation.

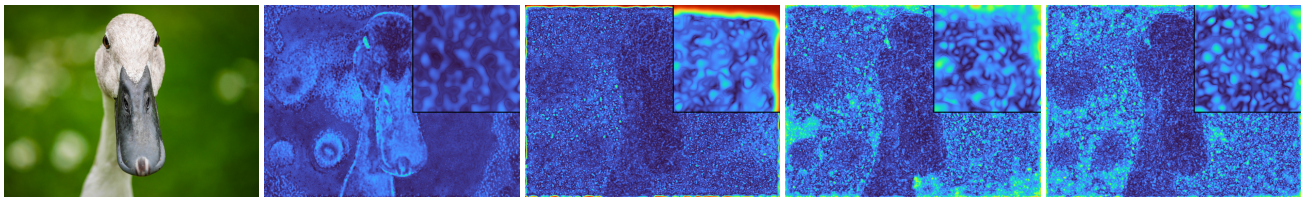


(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 9: *For the input (a), seeding the pseudo-random number generator with seeds from 0 to 4 for images (b) to (f) results in different, but overall robust approximations of the input image.*

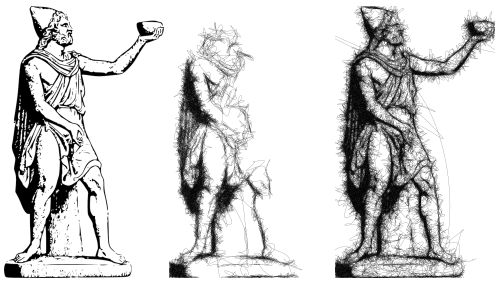
	[LLC19]	Yamashita	scribble first	scribble previous
eagle	14.1	**8.3**	14.8	13.2
duck	**11.3**	13.0	17.7	17.4
rocks	12.9	**6.7**	10.7	10.5
motor	9.5	**7.5**	10.8	10.1
eye	17.0	**8.3**	10.0	10.2
scotland	12.6	**10.7**	12.7	14.0
cottage	10.7	**8.6**	12.9	10.8
elephant	**4.5**	4.8	6.6	5.3
paprika	15.5	**11.0**	19.7	16.0
pizza	18.0	**10.9**	17.4	14.6
fish	23.0	**11.5**	16.4	14.1
rose	14.2	**10.8**	20.7	15.7

Table 1: *Results of our CSF-based error function on the approximations with [LLC19] and our method. The lowest error per image is marked in bold.*



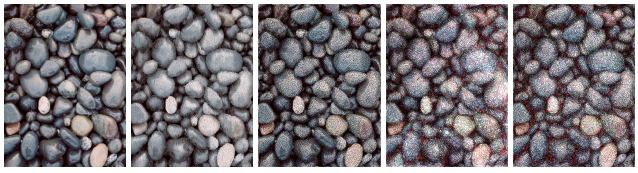
(a) (b) (c) (d) (e)

Figure 10: *Comparison of the error of [LLC19] (b) and our results (Yamashita (c), scribble-first (d), scribble-previous (e)). The input image is shown in (a). Dark blue indicates a small error and red a large error. The main cause of error in our approach are the boundary regions. The top-right corner is enlarged.*



(a) (b) (c)

Figure 11: *For inputs such as (a) it can happen that the error introduced to bridge the white regions is too large, causing the algorithm to terminate early (b). A good output (c) can still be achieved by assigning white regions a lower weight than black regions. Now, improvements to black regions can outweigh the cost of having to cross large white sections.*



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

Figure 12: *Image comparison of the input image (a), [LLC19] (b), and our approaches Yamashita-style (c), scribble-first (d), and scribble-last (e).*

Performance wise, our approach is limited by the evaluation of the CSF and the subsequent error computation. We have implemented these steps using OpenGL to utilize GPU acceleration, and together they only take about 0.87 ms for a 1000×1000 pixel image when using a NVidia 2080 RTX GPU. Since our method requires many iterations to converge, each colored path adds approximately 1 hour of compute time for an image resolution of roughly 1 million pixels.

6. Conclusion

Our approach is able to approximate images in styles similar to those of the artist Kumi Yamashita or the common scribble art. By modeling strokes or threads as contiguous paths, our approximation method is able to greedily minimize a global error function, whilst satisfying all constraints depending on the art style. Since we employ a perceptive error function with a CSF, our approach considers human visual perception. Finally, our algorithm for both styles works well on a wide range of input images. Furthermore, our scribble style is competitive with previous state of the art.

Since our algorithm is a greedy optimization, it can run into non-optimal configurations where a poor decision early on can have a negative impact on the overall approximation. Especially for our scribble style, we have no robust mechanism to escape local minima to force the path to visit regions where the approximation quality can still be improved. We believe this can be solved by either employing a similar strategy to our Yamashita approach, i.e. forcing the path to visit specific regions, or by using some kind of lookahead strategy. Additionally, we believe that performance can be significantly improved. Currently, for each candidate the CSF and error is computed for the entire image, even though only small parts of the image have been changed. By localizing the evaluation, performance can be improved greatly.

Acknowledgments

This work was partially funded by the German Federal Ministry of Education and Research within the Higher Education Pact and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 513808244. We thank Mika Specht and Johannes Schulten for their excellent Bachelor theses that helped to lay the foundation for this work.

References

[ANAM*20] ANDERSSON P., NILSSON J., AKENINE-MÖLLER T., OSKARSSON M., ÅSTRÖM K., FAIRCHILD M. D.: Flip: A difference evaluator for alternating images. *Proc. ACM Comput. Graph. Interact. Tech. 3*, 2 (2020), 15–1. 5

[APC*16] AHMED A. G., PERRIER H., COEURJOLLY D., OSTROMOUKHOV V., GUO J., YAN D.-M., HUANG H., DEUSSEN O.: Low-discrepancy blue noise sampling. *ACM Transactions on Graphics (TOG) 35*, 6 (2016), 1–13. 3

[BRWM18] BIRSAK M., RIST F., WONKA P., MUSIALSKI P.: String art: towards computational fabrication of string images. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 263–274. 2, 3

[CLLC15] CHIU C.-C., LO Y.-H., LEE R.-R., CHU H.-K.: Tone-and feature-aware circular scribble art. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 225–234. 2, 3, 7

[EF01] EFROS A. A., FREEMAN W. T.: Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), pp. 341–346. 2

[Far02] FARIN G.: *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 2002. 4

[HJO*01] HERTZMANN A., JACOBS C. E., OLIVER N., CURLESS B., SALESIN D. H.: Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), pp. 327–340. 2

[JF03] JOHNSON G. M., FAIRCHILD M. D.: A top down description of s-cielab and ciede2000. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur 28*, 6 (2003), 425–435. 2, 5

[JYF*20] JING Y., YANG Y., FENG Z., YE J., YU Y., SONG M.: Neural style transfer: A review. *IEEE Trans. Vis. Comput. Graph. 26*, 11 (2020), 3365–3385. 2

[KB05] KAPLAN C. S., BOSCH R.: Tsp art. In *Renaissance Banff: Mathematics, music, art, culture* (2005), pp. 301–308. 2, 3, 7

[LA18] LAU D. L., ARCE G. R.: *Modern digital halftoning*, vol. 1. CRC Press, 2018. 2

[LAG98] LAU D. L., ARCE G. R., GALLAGHER N. C.: Green-noise digital halftoning. *Proceedings of the IEEE 86*, 12 (1998), 2424–2444. 2

[LLC19] LO Y.-H., LEE R.-R., CHU H.-K.: Generating color scribble images using multi-layered monochromatic strokes dithering. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 265–276. 2, 3, 7, 8

[MARI17] MARTÍN D., ARROYO G., RODRÍGUEZ A., ISENBERG T.: A survey of digital stippling. *Computers & Graphics 67* (2017), 24–44. 2

[MK88] MOVSHON J. A., KIORPES L.: Analysis of the development of spatial contrast sensitivity in monkey and human infants. *JOSA A 5*, 12 (1988), 2166–2172. 2, 5

[MP92] MITSA T., PARKER K. J.: Digital halftoning technique using a blue-noise mask. *JOSA A 9*, 11 (1992), 1920–1929. 2

[PFS03] PASTOR O. M., FREUDENBERG B., STROTHOTTE T.: Real-time animated stippling. *IEEE Computer Graphics and Applications 23*, 4 (2003), 62–68. 2

[PHWF01] PRAUN E., HOPPE H., WEBB M., FINKELSTEIN A.: Real-time hatching. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), p. 581. 2

[Uli87] ULICHNEY R.: *Digital halftoning*. MIT press, 1987. 2

[Uli00] ULICHNEY R.: Review of halftoning techniques. In *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts V, San Jose, CA, USA, January 22, 2000* (2000), Eschbach R., Marcu G. G., (Eds.), vol. 3963 of *SPIE Proceedings*, SPIE, pp. 378–391. 2

[Wir23] WIRESTYLE GMBH: Wirestyle, 2023. URL: https://wirestyle.de/. 2

[Yam13] YAMASHITA K.: Constellation - mana no.2. http://kumiyamashita.com/portraits/, 2013. [Online; accessed 10-May-2023]. 1, 2, 3

[Yam16] YAMASHITA K.: Strings. http://kumiyamashita.com/portraits/, 2016. [Online; accessed 10-May-2023]. 1, 2

[ZISS04] ZANDER J., ISENBERG T., SCHLECHTWEG S., STROTHOTTE T.: High quality hatching. In *Computer Graphics Forum* (2004), vol. 23, Wiley Online Library, pp. 421–430. 2