

微分可能レンダラーを用いたロゴ画像生成

山倉 隆太^{1,a)} 柳井 啓司^{1,b)}

概要

本研究では、近年提案された微分可能レンダラー [6] とラスター画像生成における強力な性能を有する Stable Diffusion [11] を用いることで、テキストプロンプトと形状の画像から算出される損失を用いてベクターパラメータの最適化を行い、ベクター形式ロゴ画像生成を行う。本手法により入力テキスト及び形状を維持したロゴ画像を生成することが可能である事が示された一方で、不要なパスが残留することやテキストによる制御が難しいことなどの改善点が残る結果となった。

1. はじめに

画像をパラメトリックな数学的プリミティブにより表現するベクター画像は、拡大・縮小の操作に画質の劣化を伴わない点や、多くの場合同等のラスター画像よりもデータサイズがコンパクトである等の特有の利点から、フォントやアイコンなどの特定のデザイン領域で標準的に用いられている。一般的にベクターが用いられるグラフィックの中でも特に、ロゴ画像は芸術的な表現と特定のアイデンティティが識別されることが必要であるため、高度な創作技能が必要であり、作成が困難である。また、ベクター画像の作成には文法やツールへの習熟などが必要であり、多大な労力と時間を必要とするため、テキストや画像をハンドルとした直感的で柔軟なベクター画像の生成は芸術活動の支援やデザインプロセスの不要な反復の削減につながるものと考えられる。従って、本論文ではベクター形式のロゴ画像に着目して、形状を示す画像とコンテンツを示すテキストプロンプトから、視覚的なアイデンティティと芸術的なグラフィックを反映する多様なベクター画像の生成を行う。

2. 関連研究

2.1 微分可能レンダラー

従来、ベクターグラフィックスのレンダリング工程は一方方向であり、ラスター画像のベクタライズにはエッジのトレースを伴う特殊なメソッド [3], [12] が必要だった。しかし、これらの手法によるベクタライズは元のベクトルメ

トリクスと無関係であり生成されるベクターグラフィックスは劇的に異なる構造を持つため、ラスターベースのアルゴリズムをベクターグラフィックスに適用することができない。Li ら [6] はこのようなベクタライズ手法の問題点に対して微分可能な 3D レンダラー [7] に基づきピクセルの事前フィルタリングに関して、分析プレフィルターとマルチサンプリングアンチエイリアシングの 2 つの手法を用いることで入力ベクトルパラメータに対して勾配を自動的に計算できるレンダラーを提案した。

2.2 微分可能レンダラーによる画像生成

微分可能レンダラーの登場によって、高品質かつ大規模なベクターコンテンツのデータ収集の必要性の制限を回避して、ラスターベースの損失関数や機械学習手法をベクターコンテンツへ直接応用してベクターパラメータを最適化するいくつかの手法が提案された。

CLIPDraw [2] は、OpenAI により発表された画像とテキストの関係性を学習した大規模なモデル CLIP [10] を用いて、入力テキストと出力画像の類似度を計算し、微分可能レンダラーを経てベジェ曲線のパラメータを最適化する手法であり、以後の同様のフレームワークのベースとなった。

さらに、VectorFusion [5] では、学習済みのテキストからの画像生成 (T2I) モデルである Stable Diffusion [11] を用いて、拡散モデルの出力を 3D やベクターパラメータなどの任意のパラメータ空間に変換するようにモデルを蒸留するメソッド Score Distillation Sampling (SDS) [9] をベクターグラフィックス向けに調整して損失関数として利用することによって、より一貫したベクターグラフィックスの生成を可能とした。なお、本研究においても VectorFusion で用いられた SDS Loss を用いる。VectorFusion は CLIPDraw に比べ、拡散モデルの頑強な出力空間を用いたことで大幅に描画能力が向上したが、入力としてテキストのみを受け取るため入力画像の形状を維持する機能が存在しない。

3. 提案手法

3.1 手法概要

手法概要図を図 1 に示す。本手法では、入力画像 I より初期化されたベジェ曲線パラメータ (色、不透明度及び制御点座標) のセット \hat{I}_0 を生成し、微分可能レンダラー R を用いることで勾配発生可能な状態でレンダリングし、クロップ及び透視変換を用いた増強の上で安定拡散モデルに入力する。この過程で 3.3 章に示される Tone Loss、Radiation

¹ 電気通信大学

^{a)} yamakura-r@mm.inf.uec.ac.jp

^{b)} yanai@mm.inf.uec.ac.jp

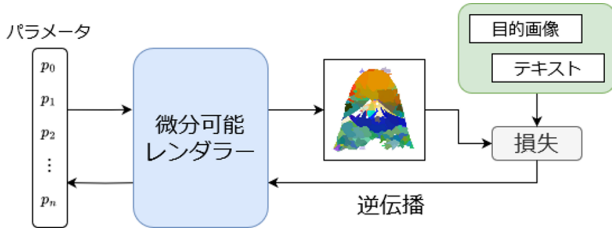


図 1 手法概略図

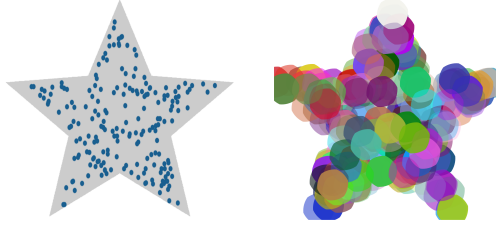


図 2 初期化の例。入力画像に従ってベジェ曲線の中心位置をランダムに取得し (左図)、円形に初期化を行う (右図)。

Loss、SDS Loss を計算し、これを用いてベジェ曲線パラメータを勾配降下法により最適化する。

3.2 表現形式と初期化

ベクター画像を構成するプリミティブは、CLIPDraw [2] のアイデアに基づき、閉じた始点 3 次ベジェ曲線に限定する。ベジェ曲線は制御点群により定義される曲線であり、ベジェ曲線同士を連結して用いることで多様な形状に対して近似が可能であり、表現力を損なわずにシンプルな実装と評価を行うことができる。このとき、 N 本の閉じたベジェ曲線パスを用いてベクターグラフィック $\hat{I} = \{P_0, \dots, P_{N-1}\}$ を形成するものとする。各パスは M 本の 3 次ベジェ曲線セグメントによって構成され、セグメントを構成する 4 つの制御点 p_m により $P_n = \{p_{0,0}, p_{0,1}, p_{0,2}, \dots, p_{M-1,0}, p_{M-1,1}, p_{M-1,2}\}$ と定義される。なお、パスは必ずセグメントの始点制御点を通過し、セグメントの終点制御点は接続のために次のセグメントの始点制御点と共有される。また、各 P_n はそれぞれ単一の色を保有し、制御点座標及び色を個別に最適化することでベクター生成を行う。

ベクターパスの初期化は LIVE [8] のアイデアに従って、すべての制御点を円形に配置することで行う。これによりパスの自己交差問題を事前に防ぐ効果が期待される。さらに Tone Loss の収束速度のためにあらかじめ入力画像 I の範囲内にパスの中心座標を設定する。図 2 に初期化の例を示す。

3.3 損失関数

図 3 に損失関数の導出フローを示す。損失関数には入力画像の領域に形状を制限する Tone Preserving Loss、入力テキストのコンテンツを反映する SDS Loss、ベクターパスの品質を保つ Radiation Loss を導入する。

Tone Preserving Loss. 出力画像を入力画像の形状に

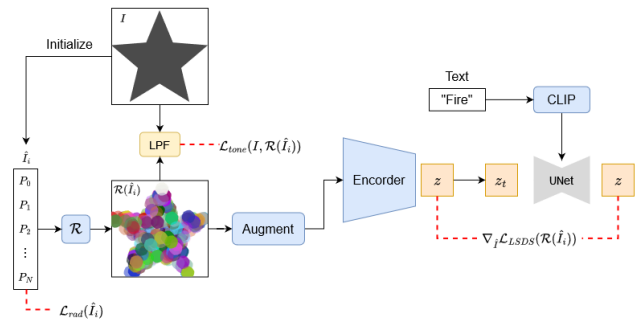


図 3 損失関数

制限するために Tone Preserving Loss [4] を追加する。なお、以下では Tone Loss と呼称する。Tone Loss はターゲット画像の形状に入力画像を近づけるために、両者をローパスフィルタに適用した後に L2 距離をとった損失である。

$$\mathcal{L}_{tone} = \|\text{LPF}(I) - \text{LPF}(\mathcal{R}(\hat{I})_{binary})\|_2^2 \quad (1)$$

ここで、 \mathcal{R} は微分可能レンダラーであり、 $\mathcal{R}(\hat{I})_{binary}$ はパスの存在する領域をマスクした 2 値画像である。

Score Distillation Sampling Loss. Score Distillation Sampling (SDS) Loss は、DreamFusion [9] で提案された、拡散モデルのヤコビアン項を省略した場合におけるサンプリング結果を用いた損失関数であり、最小化することで条件に適合するようにパラメータを調整する損失である。本手法ではテキストに示されるコンテンツを反映するために、VectorFusion [5] でベクター画像向けに調整された \mathcal{L}_{LSDS} を用いる。

Radiation Loss. 以上の 2 つのロスに加えて、Ma ら [8] によって指摘された、パスの自己交差問題の解決のために、同研究で提案された Xing Loss を拡張して Radiation Loss を提案する。Radiation Loss ではパスが自己交差しないための条件として、パスの制御点が回転方向に順番に配置される制約を設ける。すなわち、各セグメントの始点制御点 c を中心とベジェ曲線パス P_n を構成するすべての制御点 p_n を用いて以下の式 2 のように定義する。

$$\mathcal{L}_{rad} = \sum_n \text{ReLU}(\angle p_n c p_{n+1} - \angle p_n c p_{n+2}) \quad (2)$$

Total Loss. 以上 3 つの損失関数を重みを伴って加算することで以下の式 3 で Total Loss を定義する。

$$\mathcal{L}_{total} = \lambda_{tone} \mathcal{L}_{tone} + \lambda_{rad} \mathcal{L}_{rad} + \lambda_{LSDS} \mathcal{L}_{LSDS} \quad (3)$$

3.4 パスの削除

本手法ではラスタライズ後のラスタ画像を用いて SDS Loss を算出するため、Loss が小さくなるようにパスの縮小、透明化が生じる。これらのパスの曖昧化はレンダリング後のラスタ画像では問題にならないが、ベクター画像にとってはデータサイズおよびレンダリング処理に負の影響を与える。従って、最適化の過程でレンダリング画像に対する影響を考慮したパスの削除を行うことでより簡潔なベクター画像の生成を行う。閾値を τ として、以下の式 5

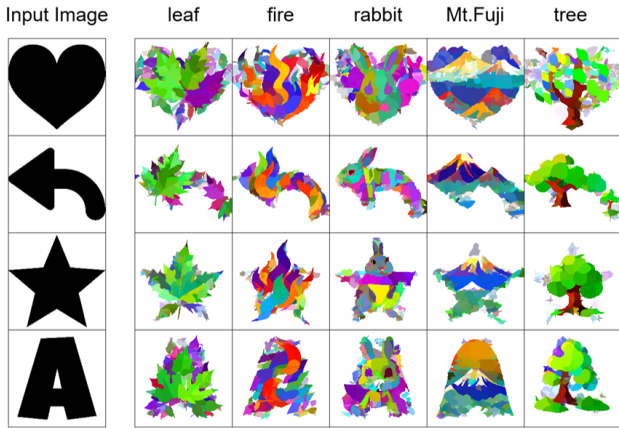


図 4 本手法による生成例

に示される条件を満たす場合に P_n を取り除く。

$$\hat{I}_{i,n} = \mathcal{R}(\{P_n, \dots, P_N\}) - \mathcal{R}(\{P_{n+1}, \dots, P_N\}) \quad (4)$$

$$\frac{\sum_{x,y} \text{alpha}(\hat{I}_{i,n})}{w \times h} < \tau \quad (5)$$

ここで \mathcal{R} はレンダラであり、 $\text{alpha}(\hat{I}_{i,n})$ は画像の不透明度、 w と h は出力画像サイズである。式 4 は P_n がどの程度レンダリング後のラスター画像に表示されているのかを表しており、実験では $\tau = 5.0 \times 10^{-4}$ とした。

4. 実験

4.1 実験設定

入出力画像のサイズは 600×600 である。拡散モデル前の増強では $\mathcal{R}(\hat{I}_i)$ を 512×512 にクロップする。デフォルトで、ベジェ曲線の本数を 200、各セグメントの数を 6、パラメータの更新回数 i を 1000、またロスの重みをそれぞれ、 $\lambda_{tone} = 200$ 、 $\lambda_{rad} = 1$ 、 $\lambda_{LSDS} = 1$ 、さらに Tone loss 用いるローパスフィルタのカーネルサイズを 101、 σ を 30 とした。なお、これらの値は経験的に設定したものである。また、 $i = 800$ の時にパスの削除を行った。入力には VectorFusion [5] を参考にプロンプトエンジニアリングを用い、“a logo of { コンセプト }. minimal flat 2d vector. lineal color. trending on artstation.” とした。

4.2 実験結果

本手法による実験結果を図 4 に示す。入力画像の形状にテキストに示される構造が生成されていることが分かる。例えば、“Mt.Fuji” の入力の場合、富士山のような構造体が出現している。最も大きい構造体としてそれぞれ炎や富士山の形状が現れている一方で入力画像領域内を満たすようにあまり意味を感じられないパスが生成されている。加えて、“rabbit” の例のように入力テキストから直感的に想像される色から離れた配色がされている。

4.3 アブレーション研究

4.3.1 Tone Loss による影響

Tone Loss に異なる重みをかけた結果を図 5 に示す。重

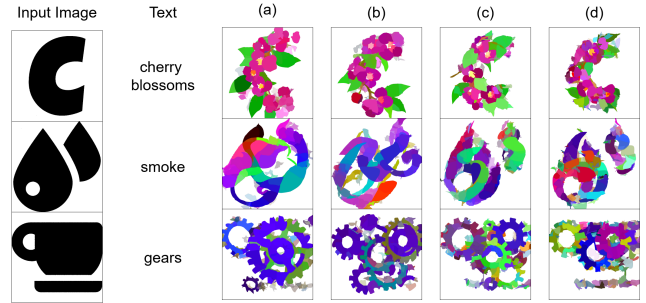


図 5 Tone Loss の重み λ_{tone} を変化させた例。(a) $\lambda_{tone} = 1.0$ 、(b) $\lambda_{tone} = 1.0 \times 10$ 、(c) $\lambda_{tone} = 1.0 \times 10^2$ 、(d) $\lambda_{tone} = 3.0 \times 10^2$

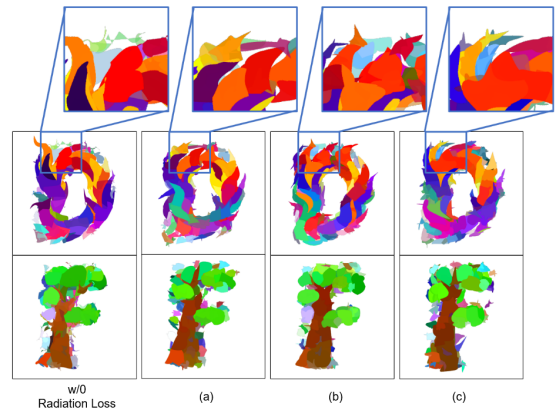


図 6 Radiation Loss の重み λ_{rad} を変化させた例。(a) $\lambda_{rad} = 1.0$ 、(b) $\lambda_{rad} = 1.0 \times 10$ 、(c) $\lambda_{rad} = 1.0 \times 10^2$

みを大きくするほど入力画像に近い形状を持つ出力が得られていることが分かる。従って、このロスを調整することによって、出力画像の形状を制御することが可能であると考えられる。

4.3.2 Radiation Loss による影響

Radiation Loss に異なる重みをかけた際の結果を図 6 に示す。Radiation Loss を取り除いた場合には自己交差による複雑なパスが生成され滑らかさが失われている一方で、Radiation Loss を追加した場合にはその重みが小さい場合でもパスが完全に凸包になっていることが分かる。ただし Radiation Loss を追加した場合にも始点制御点以外の 2 点が大きくなることによる棘状のアーティファクトが依然として生成されている。

また、Xing Loss と Radiation Loss の比較を図 7 に示す。Xing Loss の場合に比べて Radiation Loss ではパスの自己交差が低減されており、各パスがより大きい構造を形成していることが分かる。

4.4 その他のハイパーパラメータの影響

テキストプロンプトの操作による出力結果への影響を調べるために形容詞に色を追加した場合やプロンプトのプレフィックス、サフィックスを変更して実験を行った。図 8 にテキストプロンプトを“{色} {コンセプト}”として生成を行った結果を示す。有彩色の入力による出力例では色を反映している一方で、“black”や“white”などの無彩色の場合には色が無視されるか一部にのみ適用されているこ

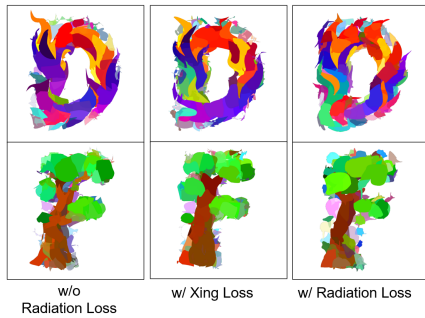


図 7 Xing Loss と Radiation Loss の比較

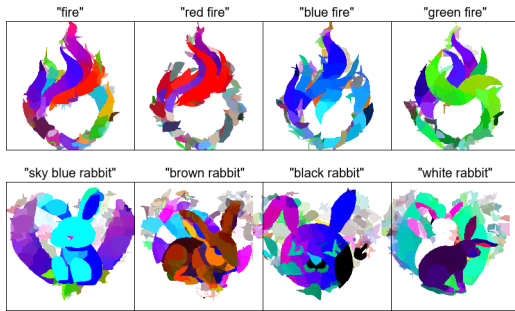


図 8 形容詞に色を追加した場合の出力例

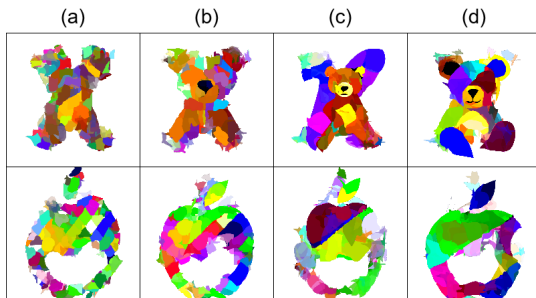


図 9 パラメータの更新回数 i を変更した場合の出力例。(a) $i = 500$ 、(b) $i = 1000$ 、(c) $i = 1500$ 、(d) $i = 2000$ 、上段の入力テキストが “bear”、下段が “apple” である。

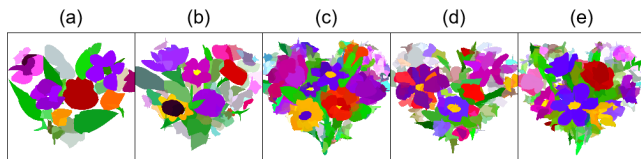


図 10 パスの本数 N を変化させた場合の出力例。パスの削除を適用することによって以下のようにパスの本数が変化する。(a)32本から32本、(b)64本から52本、(c)128本から116本、(d)256本から141本、(e)512本から166本。

と分かる。

図 9 は、パラメータの更新回数を変更した場合の出力例である。 $i = 1000$ で大部分の形状が形成されており、 $i = 1500$ にはほとんど完全に安定した形状が出力されていることが分かる。従って、 $1000 < i < 1500$ の間で適切な値を選択することが望ましい。

また、パスの本数 N を変更した場合の生成例を図 10 に示す。パスの削除によって、過度なパスの本数を指定した場合に不要なパスが削除されていることが分かる。

5. 考察

5.1 出力結果について

出力結果のサンプル (図 4) を観察すると、一部にテキストに示されるような構造体が出現する一方で、その他の部分には形状を満たすようにあまり意味のないパスが出現している。これは SDS Loss で得られるサンプル結果と形状画像の乖離を Tone Loss によって制限しているために起こる減少だと考えられる。また、一部のサンプルでテキストから想像される色相から乖離した配色が見られるが、Tone Loss による形状の制限が生成される形状を破壊していることが考えられる。これらは Tone Loss による形状の制限では SDS Loss によるサンプリング結果の形状を変化させることが不可能であることから、入力画像の形状にクロップしているのと同じような効果になっているのではないかと考えられる。従って、入力画像を拡散モデルの条件付けに用いるなどの方法によって、サンプリング結果そのものがある程度入力形状を維持できるようにする拡張が必要である。

5.2 Radiation Loss の効果

図 6 に示されるように Radiation Loss は自己交差問題を抑え、パスの形状の安定化に一定の効果を示された。その一方で完全に滑らかなパスの生成には未だ不十分であり、特定のセグメントが棘状に伸びたアーティファクトが形成されている。これは、Radiation Loss の制約の中で不要なセグメントによる描画への影響をおさえるために描画面積を少なくするように最適化されたためだと考えられる。また、Radiation Loss は各セグメントの始点制御点に関する情報が含まれているため、始点制御点の位置に不必要な制限が設けられ、不可能なパスの形状が生じる問題がある。本手法ではレンダリング後のラスター画像によりコンテンツを示す SDS Loss を使用しているため、複数のベジェ曲線による複合の効果で不可能な形状へ対処されているが、これは適切に整理されたベクターパスという目的にはそぐわない。従って Radiation Loss における中心点 c に相当する適切な基準位置の設定が必要である。

6. まとめ

本論文では、微分可能レンダラー [6] の機能と SDS Loss [9] 及び Tone Loss [4] を組み合わせることで形状を維持したベクターグラフィックスの生成手法を提案した。さらに、Xing Loss [8] を拡張した Radiation Loss を導入することで自己交差問題に対処した。実験結果より本手法では入力画像の形状を維持したままテキストの内容をベクターグラフィックスに反映することが可能である一方、入力の条件に出力結果が左右されるため依然改良の余地が残る。

今後の展望として、形状を直接制御するのではなく Stable Diffusion の Image-to-Image による条件付け等を用いてサンプリング結果そのものを入力画像に近づける方法が考えられる。また、Radiation Loss における中心座標 c の代わりに中心軸変換 [1] を用いる拡張を行うことで始点制御点に対する不要な制限を緩和する。

参考文献

- [1] Blum, H.(ed.): *A Transformation for Extracting New Descriptors of Shape*, pp. 362–380, MIT Press (1967).
- [2] Frans, K., Soros, L. and Witkowski, O.: CLIPDraw: Exploring Text-to-Drawing Synthesis through Language-Image Encoders, *Advances in Neural Information Processing Systems*, Vol. 35, pp. 5207–5218 (2022).
- [3] Hertzmann, A.: Painterly rendering with curved brush strokes of multiple sizes, *Procs. of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98*, pp. 453–460 (1998).
- [4] Iluz, S., Vinker, Y., Hertz, A., Berio, D., Cohen-Or, D. and Shamir, A.: Word-As-Image for Semantic Typography, *ACM Transactions on Graphics*, Vol. 42, No. 4, pp. 1–11 (2023).
- [5] Jain, A., Xie, A. and Abbeel, P.: VectorFusion: Text-to-SVG by Abstracting Pixel-Based Diffusion Models, *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1911–1920 (2023).
- [6] Li, T., Lukáč, M., M, G. and Ragan-Kelley, J.: Differentiable Vector Graphics Rasterization for Editing and Learning, *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, Vol. 39, No. 6, pp. 193:1–193:15 (2020).
- [7] Li, T.-M., Aittala, M., Durand, F. and Lehtinen, J.: Differentiable Monte Carlo Ray Tracing through Edge Sampling, *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, Vol. 37, No. 6, pp. 222:1–222:11 (2018).
- [8] Ma, X., Zhou, Y., Xu, X., Sun, B., Filev, V., Orlov, N., Fu, Y. and Shi, H.: Towards layer-wise image vectorization, *Procs. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16314–16323 (2022).
- [9] Poole, B., Jain, A., Barron, J. T. and Mildenhall, B.: DreamFusion: Text-to-3D using 2D Diffusion, *The Eleventh International Conference on Learning Representations* (2022).
- [10] Radford, A., Kim, J., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. et al.: Learning transferable visual models from natural language supervision, *International Conference on Machine Learning*, pp. 8748–8763 (2021).
- [11] Rombach, R., Blattmann, A., Lorenz, D., Esser, P. and Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models, *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685 (2022).
- [12] Selinger, P.: Potrace : a polygon-based tracing algorithm, <http://potrace.sourceforge.net/potrace.pdf> (2003).