Vol.2020-CG-179 No.4
2020/9/19

非顕著な線を重視したカラーイラストからの線画抽出

松田 祐紫[1,a)]　金森 由博[1,b)]　遠藤 結城[1,c)]

概要：イラストなどの画像コンテンツからの線画抽出は，描かれた対象の構造解析に必要な基礎的処理である．しかし，既存手法では周囲の色と似た色の線（本稿では「非顕著な線」と呼ぶ）の抽出に失敗しやすい．本稿では，非顕著な線も抽出可能にするためのニューラルネットワークを用いた教師あり学習手法を提案する．まず，入力カラーイラストに加え，誤検知は多いが検知漏れが少ない既存手法で得た線画も入力する．また，非顕著な線を強調するための重み付けの仕組みを導入する．ベースライン手法と比較し，本研究の有効性を検証した．

Non-Salient-Line-Aware Line Extraction from Color Illustrations

1. はじめに

CG やコンピュータビジョン（CV）の分野では，イラスト，マンガ，アニメに関する研究が近年盛んに行われている．例えば，これらのコンテンツの制作が労働集約的であるために効率化や自動化を目指した研究 [1], [2]，コンテンツを分解し新しいコンテンツとして再構成する研究 [3], [4]，静止画からアニメーションを生成する研究 [5], [6] などである．さらに，これらの技術を応用したサービス [7], [8] も登場している．

上記の研究において，カラーイラストなどの画像コンテンツからの線画抽出は，キャラクターなど描かれた対象の構造解析の基礎となる．例えば，線で囲まれた閉領域同士の関係から奥行きを推定したり [9]，髪の房の重なり順を推定して髪のアニメーションを実現したり [5] といったことが行われている．また，スクリーントーンの施されたマンガ画像から，カラー化のために線画を抽出する研究 [10] もある．印刷されたカラーイラストから線画抽出を行うことで，デジタルリマスタリングを行うという応用も考えられる．さらに最近では，線画の自動着色の訓練データの作成のために，カラーイラストから線画を抽出することも行われている [7]．

1 筑波大学
University of Tsukuba, Tennoudai 1-1-1, Tsukuba, Ibaraki, 305–8573, Japan
a) matsuda@cgg.cs.tsukuba.ac.jp
b) kanamori@cs.tsukuba.ac.jp
c) endo@cs.tsukuba.ac.jp

本研究では，カラーイラストから線画を抽出することを目的とし，これを教師あり学習を用いて実現する．線画抽出の従来手法には，自然画像を対象とした研究が多く存在する [11], [12]．しかし，これらは主に物体の輪郭などの大まかな線しか抽出の対象とせず，イラストの線画のような細かい線は対象としていない．イラストに対する線画抽出の研究 [13], [14] も存在するものの，イラスト中の全ての領域が完全な閉領域になっている画像にしか適用できなかったり，描かれている線の色が周囲の色と似ている部分で抽出に失敗したりするなどの問題がある．

そこで本研究では，従来手法では抽出に失敗しやすい，周囲の色と似た色の線（以降このような線を「非顕著な線」と呼ぶ）も抽出できるようにし，線画抽出の精度向上を図る．まず，入力のカラーイラストに加え，誤検知は多いが検知漏れが少ない既存の線画抽出手法 [15] で得た線画も入力する．また，非顕著な線を強調するためのフィルタを導入し，これによって強調された部分の誤差に重み付けする項を損失関数に追加する．これらの改良によって，抽出しづらい線の抽出を目指す．

2. 提案手法

本手法は，Li らの手法 [10] に基づき，U-Net と ResBlock を組み合わせたネットワークをベースとする．このネットワークは，カラーイラスト I_c と正解の線画 I_l で教師あり学習を行い，I_c を入力として推定された線画 I'_l を出力するものである．図 1 に提案するネットワークの概要を示す．このネットワークに以下の独自の要素 (図 1 中の赤字の部

ⓒ 2020 Information Processing Society of Japan

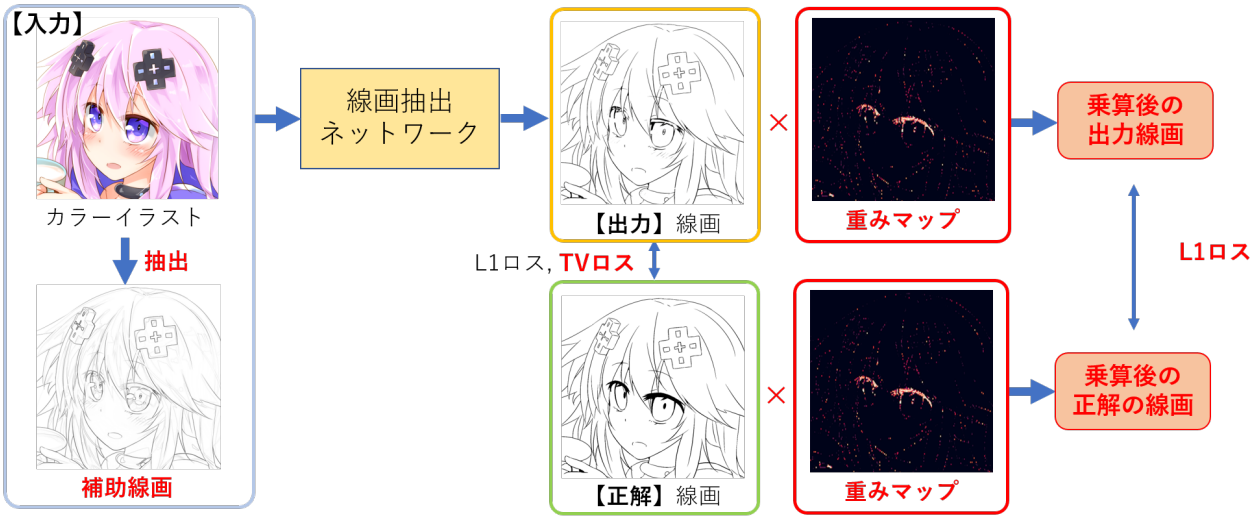


図 1　提案手法のネットワークの概要．赤字で記載されている部分が独自に追加した部分を表す．

分) を追加した．

- 詳細線画の追加による入力の増強 (2.1 節)
- L1 Total Variation ロスの追加 (2.2 節)
- 非顕著な線のための重みマップの導入 (2.3 節)

提案手法で用いる損失関数は次の通りである．

$$\mathcal{L} = w_{\mathcal{L}_1}\,\mathcal{L}_1 + w_{TV}\,\mathcal{L}_{TV} + w_w\,\mathcal{L}_w \qquad (1)$$

ここで $\mathcal{L}_1$ は L1 ロス，$\mathcal{L}_{TV}$ は L1 Total Variation ロス (2.2 節)，$\mathcal{L}_w$ は独自の重みマップを用いたロス (2.3 節) を表す．また，$w_{\mathcal{L}_1}$，w_{TV}，w_w は各項の重みである．以下，本研究の独自の要素について説明する．

2.1　詳細線画の追加による入力の増強

従来の学習ベースの線画抽出手法では，非顕著な線は抽出に失敗しやすいものの，適切な正解データを与えれば，出力結果がそれと大きく離れる可能性は小さく，線が詳細に抽出されすぎるということは起こりにくい．一方で，最近カラー画像からの線画抽出によく用いられている，文献 [15] のモルフォロジー演算を利用した手法では，そのような線は抽出できるものの，一般的に線画では描かれない詳細な線まで抽出されてしまうことや，線の濃さや太さのばらつきが大きい（図 2 参照）という問題がある．そこで学習ベースの線画抽出手法とモルフォロジー演算による線画抽出手法の双方の利点を活かすために，ネットワークの入力にモルフォロジー演算による線画も利用することを試みた．より具体的には，ネットワークの入力には，RGB 成分を持つ 3 チャンネルのカラーイラストに，文献 [15] のモルフォロジー演算を用いた手法によって得られた 1 チャンネルの線画を連結し，4 チャンネルの画像としたものを用いる．

図 2　モルフォロジー演算による線画（左）とアーティストによる線画（右）の違い．

2.2　L1 Total Variation ロスの追加

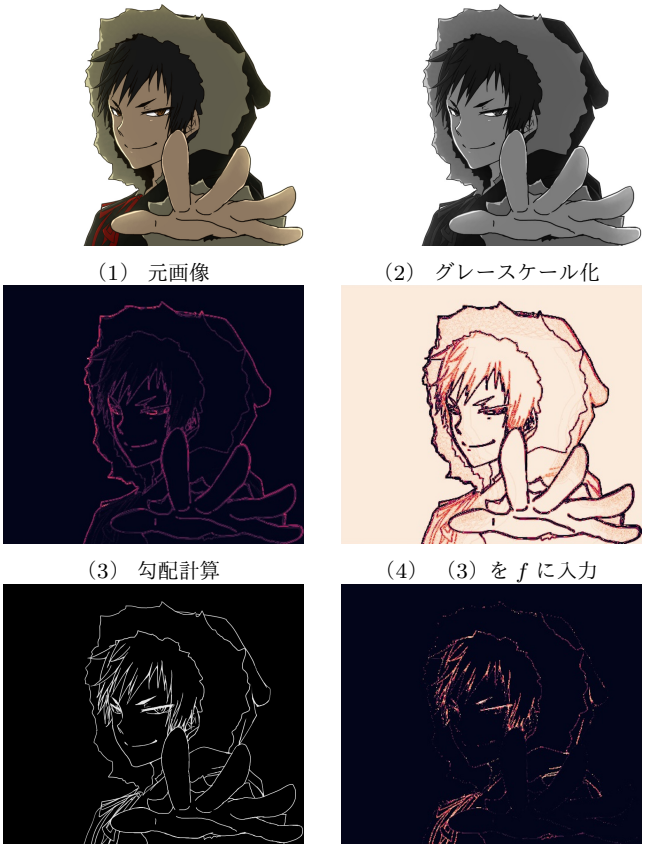
L1 Total Variation ロスは，画素値の勾配に関するロスである．これにより，出力結果において勾配がゼロの画素が多い結果が得られる，すなわち滑らかな画像が得られるという効果がある．画像の縦方向と横方向の勾配を計算する演算子をそれぞれ ∇_x，∇_y とすると，L1 Total Variation ロス $\mathcal{L}_{TV}$ は次のように書ける．

$$\mathcal{L}_{TV} = \|\nabla_x I_l - \nabla_x I'_l\|_1 + \|\nabla_y I_l - \nabla_y I'_l\|_1 \qquad (2)$$

2.3　非顕著な線のための重みマップの導入

実験により，従来の線画抽出手法では非顕著な線の抽出に失敗しやすいことがわかった．非顕著な線の誤差に大きな重みを与えることで，そのような線が抽出できるように学習が進むことが期待される．そこで，非顕著な線を強調し，その部分の誤差に重み付けするためのフィルタを設計する．これによって得られた重みマップを出力の線画に掛けることで，非顕著な線に対して大きな重みを与えることができるようになる．

ⓒ 2020 Information Processing Society of Japan



（1） 元画像　（2） グレースケール化
（3） 勾配計算　（4） （3）を f に入力
（5） 正解線画を白黒反転　（6） （3）と（4）の積

図 3 重みマップの計算手順.

非顕著な線を強調するための重みマップを M とする．以下に M の計算手順を述べる．また，図 3 は下記の各手順で生成される画像を示したものである．

手順 1: 画像をグレースケール化
手順 2: グレースケール画像のラプラシアンを計算
手順 3: 手順 2 の画像を単調減少関数に入力
手順 4: 正解の線画を白黒反転
手順 5: 手順 3 の画像と手順 4 の画像を掛ける

つまり，グレースケール化した画像を I_g，ラプラシアン演算子を Δ，単調減少関数を f，白黒反転した正解の線画を $\hat{I}_l$ とすると，M は次のように表すことができる．

$$M = f(\Delta I_g) * \hat{I}_l \tag{3}$$

ここで $*$ は 2 つの画像の要素積（アダマール積）を表す 2 項演算子である．

前述の重みマップの計算手順について説明する．手順 2 の段階では，勾配が大きい部分の値が大きくなっている．しかし，今着目すべきなのは，線画抽出しづらい部分に相当する，勾配が小さい部分である．そこで，手順 2 の画像を単調減少関数 f に入力することで，勾配が大きい部分の値が小さくなり，勾配が小さい部分の値は大きくなる．ただしこのままでは，勾配が小さい部分よりも白背景の部分，すなわち勾配が 0 だった部分の値の方が大きくなってしまう．白背景部分の重みは大きくすべきではないので，$\hat{I}_l$ を掛けることで該当する部分の値を 0 にしている．

以上により，独自の重みマップを用いたロス $\mathcal{L}_w$ は次式で定義される．

$$\mathcal{L}_w = \|M * I_l - M * I'_l\|_1 \tag{4}$$

3. 実験・結果

実装には Python ライブラリである PyTorch を使用し，NVIDIA GeForce GTX 1080 Ti を用いて学習を行った．実験に用いたデータセットは，pixiv から収集したカラーイラストと線画のペアからなる．全てのカラーイラストは白背景に統一されており，訓練データは 60 組，テストデータは 15 組である．訓練データにはデータ拡張として，各エポック毎にランダムでスケーリング，回転（90 度単位），左右反転を適用した．バッチサイズは 15，エポック数は 20,000，学習時の最適化手法には Adam を使用し，学習率は 0.001，$\beta_1 = 0.9$，$\beta_2 = 0.999$ とした．

損失関数における各項の重みは，$w_{\mathcal{L}_1} = 1$，$w_{TV} = 10$ とし，w_w について $\{1, 2.5, 5, 10, 15\}$ の 5 通りを試した．重みマップを計算するときの単調減少関数 f （2.3 節参照）には，以下の 2 通りを試した．

- linear: $1 - \Delta I_g$
- exp: $\exp\left(\frac{-\Delta I_g}{\sigma}\right)$

ここで σ は ΔI_g の画素値の標準偏差を表す．

本研究のネットワークを用いて，条件を変えて比較実験を行った．実験条件として，2.1 節の入力画像の増強のありなし，$\mathcal{L}_{TV}$ および $\mathcal{L}_w$ のありなし，$\mathcal{L}_w$ ありの場合はどの単調減少関数を使うか，および w_w に $\{1, 2.5, 5, 10, 15\}$ のどれを使うかを変更した．表 1 に各実験条件のオプションの有無を示す．baseline は L1 ロスだけを用いたものである．TV は baseline に TV ロスを加えたものである．4ch は TV に 2.1 節の入力画像の増強を加えたものである．linear と exp は 4ch に重みマップを加えたもので，それぞれ重みマップの計算時に対応する名前の単調減少関数を用いている．加えて，既存手法である *pix2pix* [16] との比較も行った．

表 1 各実験条件のオプションの有無.

	$\mathcal{L}_1$	$\mathcal{L}_{TV}$	入力画像の増強	$\mathcal{L}_w$
baseline	○	×	×	×
TV	○	○	×	×
4ch	○	○	○	×
linear	○	○	○	○
exp	○	○	○	○

ⓒ 2020 Information Processing Society of Japan

表 2　定量評価の結果．RMSE が最小のもの，および linear と exp のそれぞれで weighted RMSE が最小のものを赤字で示す．

	RMSE	weighted RMSE
baseline	32.70	–
TV	33.58	–
4ch	31.94	21.64 (linear), 13.02 (exp)
linear_1	31.43	**22.39**
linear_2.5	31.44	22.97
linear_5	31.52	24.08
linear_10	34.94	28.99
linear_15	39.78	33.54
exp_1	32.11	**14.31**
exp_2.5	31.14	14.35
exp_5	**30.97**	14.79
exp_10	31.96	14.94
exp_15	32.67	14.75
pix2pix	31.06	23.65 (linear), 14.97 (exp)

3.1　定量評価

定量評価には RMSE と weighted RMSE の 2 つの評価指標を用いた．もし重みマップを導入したことで非顕著な線が抽できたとすると，重みマップによって大きく重み付けされた部分の誤差は小さくなっているはずである．このように，重みマップがどれだけ有効に働いているかを確認するために，強調された部分に注目して誤差を計算し，重みマップを用いた場合とそうでない場合で値を比較した．weighted RMSE は重みマップの有効性を表す評価指標であり，出力の線画と正解の線画のそれぞれに重みマップを掛けたものに対して RMSE を計算することで求められる．

表 2 に各実験条件における評価指標の値を示す．linear，exp の後の数字は，w_w の値を表す．4ch と pix2pix の weighted RMSE に値が 2 つ記載されているのは，linear や exp の weighted RMSE と比較する際に，それぞれの方法で作った重みマップを掛けたものと比較するためである．

全ての条件の中で RMSE が最小であったのは，単調減少関数に exp を使用し，$w_w = 5$ とした場合であった．また，pix2pix の RMSE を上回ったのはこの条件のみであった．TV ロスのみを追加した場合では，追加する前である baseline よりも性能が低下した．weighted RMSE に関しては，重みマップの作り方によって値の範囲が異なるため，linear と exp それぞれの中で比較すると，どちらも $w_w = 1$ の時が最小であった．しかし，4ch の結果と比較すると，重みマップを使わない場合よりも性能が低下したことがわかった．

3.2　定性評価

全ての実験条件の中で，目視で顕著な差が確認できた baseline と 4ch，および RMSE が最小であった exp_5 について定性評価を行った．図 4 に各実験条件での出力結果を示す．特に髪の輪郭に注目すると，baseline では多くの部分で抽出に失敗していたが，4ch では線の色はまた薄いものの，正解の線画に近いレベルで抽出ができている．exp_5 でも baseline で抽出に失敗していた部分が抽出できているが，4ch より線がかなり薄れてしまっている．他のテスト画像でも似たような結果となっており，入力にモルフォロジー演算による線画を追加したことは有効に働いているが，重みマップを導入したことによる精度の向上は認められないことがわかった．

また他の問題点として，元画像で線が太めに描かれている場合（図 5），どの実験条件においても抽出される線が掠れてしまった．

4.　おわりに

本研究では，従来手法では抽出に失敗する傾向にある，非顕著な線も抽出できるようにするための，カラーイラストからの線画抽出手法を検討した．ResBlock を含む U-Net をベースとしたネットワークに以下の工夫を行った．

- 3 チャンネルの入力カラーイラストにモルフォロジー演算による線画を結合し，4 チャンネル画像としてネットワークに入力した．
- 非顕著な線を強調するための重みマップを計算し，これによって強調された部分の誤差に重み付けする項を損失関数に追加した．

今後の課題として，重みマップの計算方法やネットワークへの組み込み方の改善が挙げられる．図 3（6）のように計算した重みマップを可視化したところ，従来では抽出しづらい部分を強調できていることは確認できた．しかし，現状では重みマップによる精度の向上は見られないため，ネットワークやロスへの組み込み方に工夫の余地があると考えている．また，現在重みマップは手動で設計しているが，アテンションマップによる自動的な重み付けの仕組みを導入することも検討している．加えて，既存の線画抽出手法との比較も行う予定である．

参考文献

[1] Lvmin Zhang, Chengze Li, Tien-Tsin Wong, Yi Ji, and Chunping Liu. Two-stage sketch colorization. *ACM Transactions on Graphics (SIGGRAPH Asia 2018 issue)*, Vol. 37, No. 6, pp. 261:1–261:14, 2018.

[2] Haichao Zhu, Xueting Liu, Tien-Tsin Wong, and Pheng-Ann Heng. Globally optimal toon tracking. *ACM Transactions on Graphics (SIGGRAPH 2016 issue)*, Vol. 35, No. 4, pp. 75:1–75:10, 2016.

[3] Yuki Koyama and Masataka Goto. Decomposing images into layers with advanced color blending. *Computer Graphics Forum*, Vol. 37, No. 7, p. 397–407, 2018.

[4] Lvmin Zhang, Chengze Li, Yi Ji, Chunping Liu, and Tien tsin Wong. Erasing appearance preservation in optimization-based smoothing. In *European Conference on Computer Vision (ECCV)*, 2020.

[5] Chih-Kuo Yeh, Pradeep Kumar Jayaraman, Xiaopei Liu, and Chi-Wing Fu. 2.5D cartoon hair modeling and ma-

ⓒ 2020 Information Processing Society of Japan

（1） 元画像　（2） 正解の線画　（3） baseline　（4） 4ch　（5） exp_5

図 4 定性評価の結果.

ⓒ 2020 Information Processing Society of Japan

（1） 正解の線画　　（2） 推定結果

図 5 太めに描かれる線が掠れる例.（線画: 綾海しろ, Twitter : @shiroayami）

nipulation. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 21, No. 3, pp. 304–314, 2015.

[6] Talking Head Anime from a Single Image. https://pkhungurn.github.io/talking-head-anime/. (最終アクセス日 2020-7-22).

[7] style2paints.github.io — sketch + style = paints ! https://style2paints.github.io/. (最終アクセス日 2020-7-21).

[8] Photo, illustration and video editor AI tool : cre8tiveAI. https://cre8tiveai.com/. (最終アクセス日 2020-7-21).

[9] Xueting Liu, Xiangyu Mao, Xuan Yang, Linling Zhang, and Tien-Tsin Wong. Stereoscopizing cel animations. *ACM Transactions on Graphics (SIGGRAPH Asia 2013 issue)*, Vol. 32, No. 6, pp. 223:1–223:10, 2013.

[10] Chengze Li, Xueting Liu, and Tien-Tsin Wong. Deep extraction of manga structural lines. *ACM Transactions on Graphics (SIGGRAPH 2018 issue)*, Vol. 36, No. 4, pp. 117:1–117:12, 2017.

[11] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. *International Journal of Computer Vision*, Vol. 125, pp. 1–16, 2017.

[12] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Zhijiang Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[13] Ming-Ming Cheng. Curve structure extraction for cartoon images. In *Joint Conference on Harmonious Human Machine Environment*, pp. 13–20, 2009.

[14] Xueting Liu, Tien-Tsin Wong, and Xuemiao Xu. Region-based structure line detection for cartoons. *Computational Visual Media*, Vol. 1, pp. 69–78, 2015.

[15] カラー画像から線画を作る [OpenCV python] - MATHGRAM. https://www.mathgram.xyz/entry/cv/contour. (最終アクセス日 2020-7-21).

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, 2017.

ⓒ 2020 Information Processing Society of Japan