

目の正確な色を考慮した深層学習による カラー参照画像を用いたキャラクタイラスト線画の着色

秋田 健太^{1,a)} 森本 有紀¹ 鶴野 玲治¹

受付日 2020年2月17日, 採録日 2020年10月6日

概要: 近年, 深層学習を用いた自動着色に関する研究がさかんである. カラーイラストは線画を着色することで作成されるが, 着色に非常に時間がかかるため, 自動着色技術によって効率化することが期待される. 既存の研究の多くは, ユーザが線画の領域ごとに色ヒントを入力する必要があるため, 入力が多くなる. また, カラー参照画像による既存の自動着色手法では, 目のような小領域は指定どおりに着色できないことが多い. しかし, 目はキャラクタイラストにおいて, 非常に着目される部分であり, 重要である. そこで本論文では, カラー参照画像によるキャラクタイラストの線画の顔を対象とした自動着色において, 正しく目の色を反映するための深層学習モデルを提案する. 本手法では, カラー参照画像の目の一部をそのまま円形の色ヒントとして, 線画の目の位置に自動で与える. これにより, ユーザ入力なしで入力線画の目に参照画像の目の色を先行手法よりも正しく反映することができる.

キーワード: 自動着色, 線画, Deep learning

Line Drawing Colorization of Character Illustration by Color Reference Image for Accurate Eye Colors based on Deep Learning

KENTA AKITA^{1,a)} YUKI MORIMOTO¹ REIJI TSURUNO¹

Received: February 17, 2020, Accepted: October 6, 2020

Abstract: Recently, automatic colorization of line drawings using deep learning has been studied. Color illustrations are created by colorizing line drawings. Colorization task is tedious and time-consuming. Therefore, the automatic colorization system is expected. Most existing studies require a lot of user inputs as color hints for each region. Previous studies using reference images often fail to realize automatic colorization, especially for small areas, e.g., eye colors in the resulting image may differ from the reference image. However, eyes are important because these are noticeable in the characters' illustrations. In this paper, we propose a method to colorize line drawings of anime characters' face with accurate eye colors of an input color reference image. The proposed method estimates eye positions of line drawing and color reference image. Then round areas centered on these positions are color hints for our colorization network. Comparing with previous methods, our method can accurately colorize eyes of line drawings only with a color reference image.

Keywords: colorization, line drawing, Deep learning

1. はじめに

カラーイラストは多くの場合, 線画を人が着色することにより制作される. この着色作業は非常に時間がかかるため, 近年この着色作業を効率化することを目的とした深層

学習を用いた自動着色手法の提案がされている. 自動着色とは, ユーザの入力が全く必要ない, あるいはユーザが色ドットなどの色ヒントを入力することで, 入力画像を簡単に着色できる技術のことをいう. 自動着色手法の1つとして, カラー参照画像による自動着色手法が提案されている [1], [2]. これらの手法は, カラー参照画像の色合いで入力画像を自動で着色することができる. しかし, 小さいが

¹ 九州大学
Kyushu University, Fukuoka 815–8540, Japan
^{a)} 2DS19064W@s.kyushu-u.ac.jp

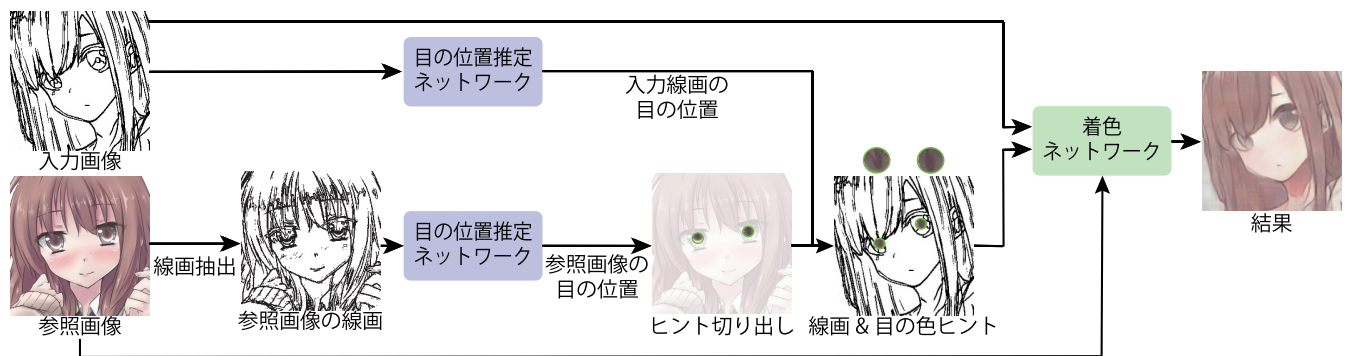


図 1 着色する際の処理の全体図

Fig. 1 Overview of our colorization process.

特徴的である目などの領域に指定した色が反映されにくいことが課題としてあげられる。

そこで本論文では、カラー参照画像の目の色を線画の目に反映するための顔を対象とした自動着色手法を提案する。本手法では、目の位置を推定するネットワーク (4.3 節) を用い、入力線画と参照画像の目の位置をそれぞれ推定する。その後、推定された参照画像の目の位置の周囲を複数の色を含んだ、円形パッチとして切り出し色ヒントとする。これを線画に与えることで、着色ネットワーク (4.2 節) により自動で正しい目の色を反映した着色を行う (図 1)。本手法の新規性の要約は以下のとおりである。

- 目の位置推定による、カラー参照画像の目の色の反映が可能な自動着色手法の提案。
- 線画のための目の位置推定ネットワークの提案。

2. 関連研究

2.1 カラーストロークを用いる自動着色

深層学習モデルによる線画の自動着色手法として、ユーザがカラーのストロークやドットを色ヒントとして各領域に描画する手法がある [3], [4], [5]。このような手法では小さい領域であっても指定したとおりの色で着色できる。その反面、領域ごとの指定が必要なため、入力が多くなることもある。対して本手法は、1 枚のカラー参照画像の入力のみによって線画を着色することができる。

2.2 カラー参照画像や自然言語を用いる自動着色

各領域に対する色ヒントを自然言語を用いて行う自動着色の手法がある [6]。この手法では目、髪などの各領域、および、それに対する赤、茶などの色を単語で入力することで自動着色を行う。たとえば、Brown hair などの自然言語を入力した場合、髪の領域を茶色で着色することができる。複数の画像に対して同じ単語を用いる場合、複数枚の線画を同時に着色することが可能である。しかし、学習の際に用いた単語の領域や色でしか着色できないため、ユーザが細かな指定を行えない。

カラー参照画像を用いて自動着色を行う手法 [1], [2] で

は、ユーザが着色したいと考える色合いの画像をカラー参照画像として入力することで着色を行う。参照画像を用いる手法は色転写とも表現されるが、本論文ではすべて着色と表現する。これらの手法は同一のカラー参照画像により、複数枚の画像を同時に着色することが可能である。しかし、目や口などの小さい領域を参照画像の別の領域の色で着色してしまうことがある。本手法では、参照画像の目から色ヒントを切り出して線画に与えることで、より正しく目の色を反映することが可能である。

3. データ作成

本手法では着色と目の位置推定のために 2 種類のネットワークを使用する。2 種類のネットワークはそれぞれ 3.1 節の方法で作成されたデータセットと 3.2 節の方法で作成されたデータセットを用いて学習する。本論文で示す結果を作成するためのデータセットはアニメーションの結果を除いて、Nico-illustr Dataset [7] を用いた。

3.1 着色ネットワークの学習データセット

着色ネットワーク (4.2 節) の学習のために入力となる線画とその対となるカラー画像のデータセットを作成する。カラー画像作成のため、ニコニコ動画 [8] からダウンロードした実際のキャラクターアニメーションから 256×256 pixel 以上の正方形で顔の部分を検出 [9] する。学習のために約 2 万枚の顔画像を切り出し、 256×256 pixel に拡大縮小する。このときロバスト性を高めるため、先行手法 [1] 同様、拡大縮小した画像を切り抜きの位置を画像ごとにランダムに変えた上で 224×224 pixel に切り抜き、50% の確率で左右反転する。これをアニメのカラー画像データセット (C^{anime}) とする。 C^{anime} はキャラクターごとのラベルがあるデータセットであり、250 キャラクターのラベルがある。また、 C^{anime} の他に、イラストのデータセット [10] の 1 人のキャラクターが描かれたイラストのみから、 C^{anime} と同様の方法で学習のために約 37 万枚の顔画像を切り出し、同様の前処理を施したデータセットを作成する。これをイラストのカラー画像データセット $C^{illustr}$ とする。 $C^{illustr}$ は

キャラクタごとのラベルを持たないデータセットである。 \mathcal{C}^{illust} を作成するのは、 \mathcal{C}^{anime} は学習データセットとして小規模なためである。 \mathcal{C}^{illust} は \mathcal{C}^{anime} よりも大規模な学習データであり、 \mathcal{C}^{illust} のデータを用いた学習も行うことで、より参照画像の色を反映した着色が可能になる。

次に \mathcal{C}^{anime} と \mathcal{C}^{illust} を線画化する。線画は petalica paint [5] で用いられている線画抽出法で抽出した後、二値化することで作成する。petalica paint の線画抽出法では、まずカラー画像をグレースケール画像に変換し、グレースケール画像に対するモルフォロジー処理で白い部分を膨張させ、線を細くする。細線化画像とグレースケール画像との差をとると色の濃い領域の縁から数 pixel ほどの領域が白く線状に残る。これを白黒反転することで線画としている。この処理に加えて二値化することで作成した線画をそれぞれデータセット (\mathcal{L}^{anime} , \mathcal{L}^{illust}) とする。

3.2 目の位置推定ネットワークの学習データセット

目の位置推定ネットワーク (4.3 節) の学習のために入力となる線画とその線画の目の位置が対となるデータセットを作成する。学習のための線画データセット ($\mathcal{L}_{train}^{eye}$) と検証データセット (\mathcal{L}_{eval}^{eye}) は主に \mathcal{L}^{anime} とは別のアニメ画像から作成した線画を用いており、 $\mathcal{L}_{train}^{eye}$ は約 2,600 枚、 \mathcal{L}_{eval}^{eye} は約 200 枚作成する。また、これらのデータセットに加えて 6.2 節の評価のためにテストデータセット (\mathcal{L}_{test}^{eye}) としてイラストのデータセット [10] から約 200 枚の線画を作成する。 \mathcal{L}^{anime} とは別に作成した学習のための線画も同様の前処理を行う。それぞれの目の位置の正解データセットは、それぞれのデータセットの画像に対して手動でキャラクタの左右の目の座標を入力して作成する。

4. ネットワークモデル

本手法では入力された線画を参照画像の色で着色するため、着色ネットワークとして Comicolorization [1] と同様のネットワークを用いる。さらに、線画に参照画像の目の色を反映するため、目の位置推定ネットワークを用いてヒントを目の位置に自動的に与える (図 1)。本手法では着色の対象をキャラクタの線画の顔周りとする。

4.1 Comicolorization

Comicolorization のネットワークは、主に Low-level Features Network, Mid-level Features Network, Global Features Network, Colorization Network の 4 つのネットワークからなる。Low-level Features Network は入力画像から着色と分類のために必要な共通の特徴量を抽出する。そのため、Low-level Features Network の重みは共通である。この共通の特徴量からさらに Mid-level Features Network は着色に、Global Features Network は分類に必要な特徴量を抽出する。このとき、Comicolorization では Global Fea-

tures Network の分類のためのラベルとしてキャラクタ名を用いる。Mid-level Features Network と Global Features Network は別々に特徴量を抽出するため、これらの特徴量を中間層の Fusion Layer と呼ばれる層で結合する。Fusion Layer ではこれらの特徴量に加えて、色ヒントとして与えられたカラー参照画像を RGB それぞれ 6 段階に減色し、216 次元のベクトルで表現した特徴量も同時に結合する。

Fusion Layer で着色のための特徴量、分類のための特徴量、色ヒントとなるベクトルを結合した後、結合した特徴量が Colorization Network に入力される。この入力された特徴量に基づき、Colorization Network が入力画像を着色し、着色画像が出力される。

4.2 着色ネットワーク

Comicolorization のネットワークはパレットモデルとヒストグラムモデルの 2 種類がある。そのうち本手法では、ヒストグラムモデルを適用する。Comicolorization のパレットモデルでは、ネットワークへ色ヒントとして入力されるカラー参照画像は 216 次元の色数に減色され、それぞれの色が画像に含まれるかどうかを 0/1 で表現する。このため、各色がどの程度含まれるかの情報が失われてしまう。これに対し、ヒストグラムモデルには各色がどれだけあるかの情報がある。

本手法では学習時に入力線画の各領域に対し、同じ領域内では類似色が割り当てられるような学習が行われると考えられるため、パレットモデルよりヒストグラムモデルの方が適している。

着色ネットワークの学習の際、入力線画は正解画像から抽出された線画を、参照画像は正解画像を用いる。そのため、学習時は参照画像の各色の領域面積と、入力線画の対応する領域面積が一致する。しかしながら、テスト時は学習時と異なり、ネットワークへ入力される参照画像の各色の領域面積と入力線画の対応する領域面積は一致しないことがほとんどである。このため、テスト時の参照画像と入力線画の面積の不一致に対するネットワークのロバスト性を高める必要がある。そこで、本手法では学習時に参照画像を 216 次元のベクトルに変換した後、各ベクトルのスカラーをランダムに増減させ、参照画像の各色の領域面積が入力線画の対応する領域面積に一致しないようにすることでロバスト性を高める。

本手法ではヒントとして参照画像の目の色情報を自動で与えることで、目の色を反映した着色を行う。このヒントを与える着色を可能にするため着色ネットワークの学習の際に、あらかじめ学習した目の位置推定ネットワークを用いて正解画像の目の位置を推定する。この推定された目の位置に切り出した目の色情報をヒントとして与えた画像も入力する。

学習時のヒントは参照カラー画像から切り出した複数の

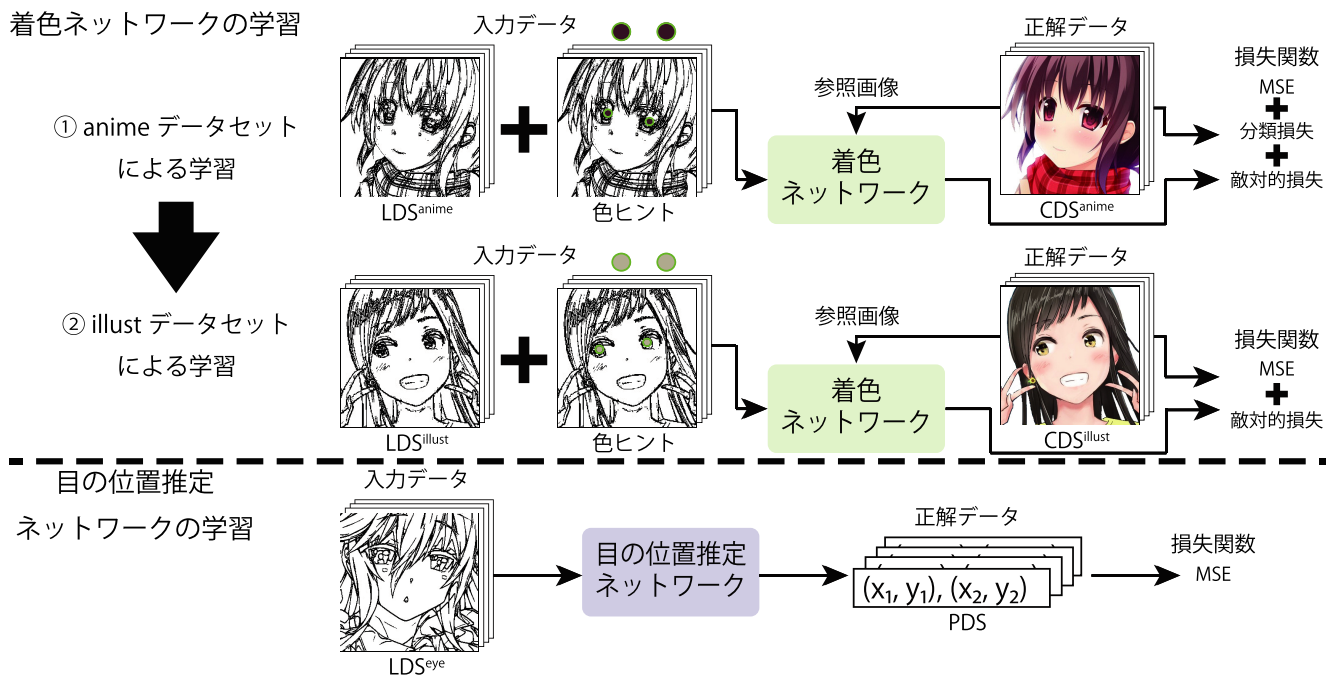


図 2 学習の概要図. 着色ネットワークは \mathcal{L}^{anime} と \mathcal{C}^{anime} での学習を行った後①, $\mathcal{L}^{illustr}$ と $\mathcal{C}^{illustr}$ の学習②を行う

Fig. 2 Overview of our training. After training with \mathcal{L}^{anime} and \mathcal{C}^{anime} , the colorization network is trained with $\mathcal{L}^{illustr}$ and $\mathcal{C}^{illustr}$.

色を持つパッチではなく、推定した位置のピクセルの色を用い、図 2 の色ヒントのように入力線画の目の位置に 1 色の色を半径 9 pixel の円形にして与える。学習時にヒントとして参照カラー画像から切り出したパッチをそのまま用いる場合、学習中は参照カラー画像に正解画像を用い、線画は正解画像から抽出されているため、パッチのハイライトの位置や描かれている詳細な形状などの特徴が入力線画のものとは完全に一致する。このような場合、ネットワークは入力線画にかかわらず、パッチそのものを出力するように学習するため、テスト時にヒントと入力線画の目の特徴が異なると、ヒントのハイライトなどの位置や形状が反映され入力線画の目の情報が消えてしまう。よって本手法では学習時に単色パッチを用いることで、ヒントと入力線画の対応する位置の特徴が一致しないようにし、入力線画の目の情報を削除しないようにする。一方で、学習時と異なり、テスト時には複数色のパッチを与える。これは、単色よりもパッチに含まれる色が考慮され、目の推定位置の誤差にロバストになるためである。

自動着色のための処理の順序は、最初に入力線画と参照画像を 3.1 節と同様の線画抽出法で線画化した線画画像を目の位置推定ネットワークに入力し、それぞれの画像の目の位置を推定する。その後、参照画像の推定された目の位置から着色ネットワークにヒントとして与える目の色情報を切り出す。ヒントとして切り出す領域は推定された目の位置を中心として、半径 9 pixel の円形のパッチとする。このパッチを入力線画の推定された目の位置に張り付けた画

像、パッチの貼り付けられていない元の線画、そして参照画像を着色ネットワークに入力する。これにより、本手法では自動で着色を行うことができる (図 1)。

4.3 目の位置推定ネットワーク

本ネットワークは入力層と全結合層の最終層を除いて、ResNet-18 [11] と同様のネットワーク構造を用いている。入力層は入力が線画であるため 1 チャネルにし、出力が右目と左目の x 座標と y 座標の合計 4 つの座標であるため、全結合層の最終層のユニット数を 4 にした。これにより、画像の左右の目の x 座標と y 座標を得ることができる。目の位置推定は入力画像が線画である方が、推定精度が高いため、参照画像は 4.2 節でも述べたように 3.1 節の線画抽出手法を用いて線画に変換したうえで位置推定を行う。

5. 学習

ネットワークの学習には、着色ネットワーク、目の位置推定ネットワークともに最適化手法として Adam [12] を用いる。パラメータは Comicolorization [1] と同様、 $\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ に設定して学習した。重みの更新のための GPU は、NVIDIA Tesla P100 を用いた。本手法では、着色ネットワークはラベルのあるデータセットとないデータセットの 2 種類があるため、Low-level Features Network の重みは共有しない。着色ネットワークは最初、入力を \mathcal{L}^{anime} 、出力を \mathcal{C}^{anime} として学習する。最初の学習の際は着色ネットワークの損失関数は Comicolorization

と同様であり、平均二乗誤差 (MSE) と敵対的損失、分類損失を用いた。それぞれの損失は式 (1) のように重み付けした。

$$\text{MSE} : \text{敵対的損失} : \text{分類損失} = 1 : 0.001 : 0.003 \quad (1)$$

敵対的損失は Generative Adversarial Networks (GANs) [13] によるものである。着色ネットワークの学習にこの敵対的損失を導入することでより正解画像に近い鮮やかな着色が可能となる [14]。分類損失のためのラベルは 250 のキャラクターラベルを用いた。 \mathcal{L}^{anime} と \mathcal{C}^{anime} を用いた学習は、バッチサイズを 64 とし、約 86,000 イテレーションさせた。学習には約 23 時間かかった。

\mathcal{L}^{anime} と \mathcal{C}^{anime} を用いて学習した後、入力を $\mathcal{L}^{illustr}$ 、出力を $\mathcal{C}^{illustr}$ に変更して学習する。この学習で用いる損失関数は $\mathcal{C}^{illustr}$ にラベルがないため、MSE と敵対的損失である。そのため、Global Features Network は Fusion layer に接続するための最終層を除いて学習は行わない。MSE と敵対的損失の重み付けは式 (1) と同様である。 $\mathcal{L}^{illustr}$ と $\mathcal{C}^{illustr}$ を用いた学習はバッチサイズを 64 とし、約 23,000 イテレーションさせた。学習には約 5 時間かかった。

目の位置推定ネットワークは入力を $\mathcal{L}^{eye}_{train}$ 、出力を $\mathcal{P}^{eye}_{train}$ とし、座標推定のため、損失関数として MSE を用いた。学習はバッチサイズを 64 とし、約 73,000 イテレーションさせた。学習には約 3 時間かかった。学習の概要図を図 2 に示す。

6. 結果と評価

6.1 アンケート調査

本手法を評価するため、本手法と先行手法に関するアンケート調査を行った。アンケート調査の参加者は 17 人で、以下の評価項目についてそれぞれ 0 から 10 までの 11 段階で評価してもらった。

- 自然さ：ある程度熟練したイラストレータが着色したように見えるか
- 目の色：目の色が反映できているか
- 目以外の全体的色：目以外の全体的色を反映できているか

自然さについては基準画像を示し、その基準に沿って評価してもらった。自然さの項目は Comicolorization [1] と本手法を比較するために設けた。本手法では基準画像は 2 つの手法による結果のうち、どちらの手法で作成したものかを隠した上で最も良いものと悪いものを著者らが選んだ。

比較する手法は、本手法とユーザ入力を用いない先行手法である。これらの手法で 30 枚の線画を 40 枚の参照画像を用いて着色する。各手法で生成した 1,200 枚の画像から、10 枚ずつ合計 20 枚の画像をランダムに抽出し、被験者に評価してもらう。それぞれの項目について、評価の平均を

表 1 アンケート評価の結果。0 から 10 の 11 段階評価の平均を示す
Table 1 The result of questionnaire evaluation. These values are means of scores in the range of [0, 10].

評価項目	Comicolorization [1]	本手法
自然さ	5.82	5.67
目の色	3.94	6.11
目以外の全体的色	5.97	6.23

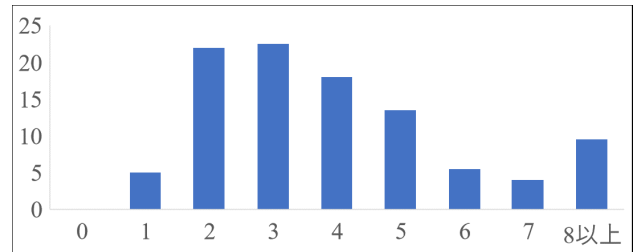


図 3 目の位置の推定精度。横軸が誤差 d_{max} 、縦軸が割合である
Fig. 3 Accuracy of our eye positions estimation. Horizontal axis is error d_{max} , and vertical axis is proportion.

取った結果を表 1 に示す。

表 1 から目の色の項目は、先行手法と比較して本手法の方が約 2 段階評価が高く、本手法により先行手法と比較してより目の色を正しく着色することができたことが分かる。自然さと目以外の全体的色の項目では本手法と先行手法で差が小さい。これは、自然さは画像全体の色合いに違和感があるかを評価する質問であるため、目の色が参照画像と異なる色で着色されていたとしても、評価への影響が小さかったためと考えられる。また、目以外の全体的色の項目は、本手法では目に色ヒントを与えることで参照画像の目の色を反映した着色をしており、他の領域にはほとんど影響がないためであると考えられる。

6.2 目の位置推定の精度評価

本手法の目の位置推定 (4.3 節) の精度の評価を行う。3.2 節で作成した \mathcal{L}^{eye}_{test} に対して目の位置推定を行い、正解座標である \mathcal{P}^{eye}_{test} との誤差を計測した。評価のためのキャラクターは 4.3 節のネットワークの学習で用いたものとは異なるキャラクターを選んだ。正解座標と推定座標の誤差の計算はユークリッド距離を用いた。このとき、推定座標は整数ではなく小数であり、またユークリッド距離であるため、計算結果に小数が含まれる。小数以下の値は評価を簡単にするために切り捨て、整数にしたものを正解座標との距離とする。

また、片目でも推定位置が大幅にずれれば両目に正確に色の反映をすることができない。そのため、左右の目でそれぞれ正解との誤差を求めた後、誤差が大きい方の距離を d_{max} とし、その値を最終的な誤差として評価する。正解との誤差と d_{max} の割合を示したのが図 3 である。図 3 から、誤差は約 90% 以上が 7 以下であることが分かる。推定



図 4 結果の比較. (a) は入力線画, (b) は参照画像, (c) はヒントを与えた画像, (d) は本手法で着色した画像. (e) は先行手法 [1] でユーザ入力を行わなかった場合の結果画像

Fig. 4 Comparisons with the previous method. (a) input line drawing, (b) reference image, (c) input image with hint patches, (d) our result, (e) a result by Ref. [1] without post processing.

誤差が7以下であれば、ほとんどの場合、正確に目を着色することができた。

6.3 計算時間と結果の考察

本手法の着色結果と先行手法による結果を比較したものを図4に示す。テスト時のGPUはNVIDIA GTX 1070を用いた。テスト時に要した時間は入力画像が1枚のときは1.53秒であり、複数枚(64枚)のときは1.96秒であった。テスト時の時間は、目の位置推定と着色ネットワークの計算時間を合わせた時間であり、5回計測した結果の平均である。

アンケート評価(表1)からも分かる通り、ユーザの修正なしの先行手法と比較して、参照画像の目の色をより正しく反映した着色をすることができた。また、目の色のヒントをネットワークに与えているため、図4中段のような先行手法では反映が難しい、左右の目の色が異なる参照画像を用いた場合であっても目の色を反映することができた。

本手法はユーザの入力が参照画像以外必要なく、複数枚の線画を1枚の参照画像のみの入力でも着色することができ

る。複数枚の線画の自動着色の例として、同一キャラクターのアニメーションに関して、本手法と既存の参照画像を用いる手法の比較を行った(図5)。比較の結果から、本手法では、特に目に関してより参照画像に近い複数枚の線画の着色が可能であることが分かった。

6.4 今後の課題

本手法の課題として、与えたパッチの詳細な色合いを線画に反映できないことがある。これは学習中に用いる目の色ヒントが1色である場合、参照画像から切り出した複数の色を含んだ色ヒントで学習した場合と比較して、パッチに含まれる複数の色を線画の目に反映できないためである。しかし、4.2節でもあげたように、参照画像から切り出した色ヒントを用いて学習した場合、色ヒントに含まれるハイライトなどの色以外の詳細を反映するといった問題が起こることがある。そのため、このような問題が起こらず、与えたパッチの詳細な色合いをより反映できるような色ヒントへ改良したい。このほか、キャラクターの全身画像などの着色、片目しか映っていない場合などへの対応もあげられる。

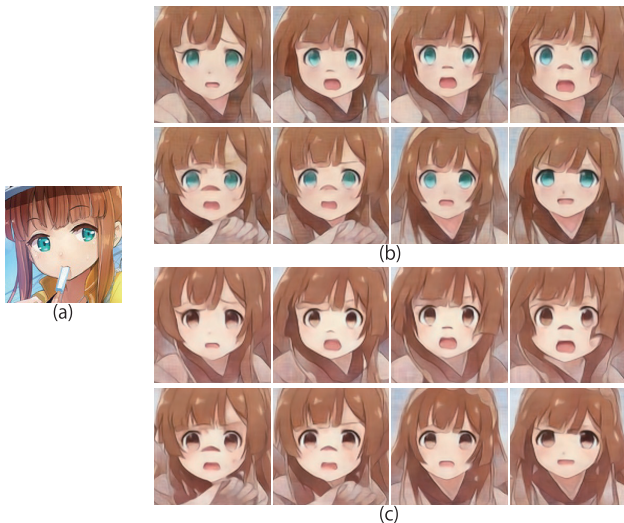


図 5 1 枚の参照画像を用いて複数枚の線画を着色した結果. (a) は参照画像, (b) 本手法の着色結果, (c) は先行手法 [1] の着色結果. ©SSS/STL/WAO

Fig. 5 Colorization results of many line drawings using one reference image. (a) reference image, (b) our results, (c) results by Ref. [1]. ©SSS/STL/WAO

7. 結論

本論文では, キャラクターの線画をカラー参照画像によって自動着色する際に, 目の色をより正確に表現する手法を提案した. 本手法では, 目の位置推定ネットワークを用い, 線画に自動でカラー参照画像の目の色ヒントをパッチとして与えることによってこれを実現した. 今後, 本手法を目以外の領域に対しても自動でより詳細な着色を行える手法へ発展させたい.

謝辞 本研究は九州大学情報基盤研究開発センター研究用計算機システムの一般利用を利用した.

参考文献

- [1] Furusawa, C., Hiroshiba, K., Ogaki, K. and Odagiri, Y.: Comicolorization: Semi-automatic Manga Colorization, *SIGGRAPH Asia 2017 Technical Briefs* (2017).
- [2] Zhang, L., Ji, Y., Lin, X. and Liu, C.: Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier GAN, *ACPR* (2017).
- [3] Zhang, L., Li, C., Wong, T.-T., Ji, Y. and Liu, C.: Two-stage Sketch Colorization, *ACM Trans. Graph* (2018).
- [4] Ci, Y., Ma, X., Wang, Z., Li, H. and Luo, Z.: User-Guided Deep Anime Line Art Colorization with Conditional Adversarial Networks, *MM* (2018).
- [5] Yonetsuji, T.: petalica paint (2017), available from https://petalica-paint.pixiv.dev/index_ja.html.
- [6] Kim, H., Jhoo, H.Y., Park, E. and Yoo, S.: Tag2Pix: Line Art Colorization Using Text Tag With SECat and Changing Loss, *ICCV* (2019).
- [7] Ikuta, H., Ogaki, K. and Odagiri, Y.: Blending Texture Features from Multiple Reference Images for Style Transfer, *SIGGRAPH Asia Technical Briefs* (2016).
- [8] ニコニコ動画, 入手先 <https://www.nicovideo.jp/>.

- [9] Nagadomi: lbpcascade_animeface (2011), available from https://github.com/nagadomi/lbpcascade_animeface.
- [10] Anonymous, The Danbooru Community, Branwen, G. and Gokaslan, A.: Danbooru2018: A Large-Scale Crowdsourced and Tagged Anime Illustration Dataset (2019), available from <https://www.gwern.net/Danbooru2018>.
- [11] He, K., Zhang, X., Ren, S. and Sun, J.: Deep Residual Learning for Image Recognition, *CVPR* (2016).
- [12] Kingma, D.P. and Adam, B.J.: Adam: A Method for Stochastic Optimization, *ICLR* (2015).
- [13] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y.: Generative Adversarial Nets, *NIPS* (2014).
- [14] Sangkloy, P., Lu, J., Fang, C., Yu, F. and Hays, J.: Scribbler: Controlling Deep Image Synthesis with Sketch and Color, *CVPR* (2017).



秋田 健太

2019年九州大学芸術工学部卒業. 同年より同大学大学院芸術工学府に在学中. コンピュータグラフィックスに関する研究に従事. ACM 会員.



森本 有紀 (正会員)

2008年九州大学芸術工学府修了, 同年東京大学にて日本学術振興会特別研究員, 2009年独立行政法人理化学研究所研究員, 2012年東京電機大学未来科学科講師, 芝浦工業大学助教等を経て, 2016年より九州大学助教.

コンピュータグラフィックスに関する研究に従事. ACM SIGGRAPH 会員. 博士 (芸術工学).



鶴野 玲治 (正会員)

九州大学大学院芸術工学研究院教授. 1987年大阪府立大学大学院修了 (情報科学). 博士 (工学). コンピュータグラフィックス, インタラクショナルデザインの研究に従事. ACM, Eurographics, 芸術科学会等の会員.