# Active Colorization for Cartoon Line Drawings

Shu-Yu Chen†, Jia-Qi Zhang†, Lin Gao*, Yue He, Shihong Xia, Min Shi, Fang-Lue Zhang

**Abstract**—In the animation industry, the colorization of raw sketch images is a vitally important but very time-consuming task. This paper focuses on providing a novel solution that semiautomatically colorizes a set of images using a single colorized reference image. Our method is able to provide coherent colors for regions that have similar semantics to those in the reference image. An active-learning-based framework is used to match local regions, followed by mixed-integer quadratic programming (MIQP) which considers the spatial contexts to further refine the matching results. We efficiently utilize user interactions to achieve high accuracy in the final colorized images. Experiments show that our method outperforms the current state-of-the-art deep learning based colorization method in terms of color coherency with the reference image. The region matching framework could potentially be applied to other applications, such as color transfer.

**Index Terms**—active learning, line drawing colorization, region matching

## 1 Introduction

With the advent of digital media technology, there have been constant improvements in the cartoon industry. Currently, the demand for cartoon entertainment has expanded with the popularity of the internet, making the related industries grow rapidly. In popular cartoon media such as comics, color is particularly important. Coloring line drawings requires a tremendous amount of manual effort. Although there are some popular commercial animation software designed for animation production [1], the colorization step still requires manual effort to produce high-quality animations. If this process could be finished automatically, it would greatly benefit the related cartoon industry.

Inspired by the recent progress made in terms of image synthesis methods and deep generative models, researchers have investigated how to colorize black-and-white pictures and line drawings efficiently. Early research works focused on brush-based interactions [2], [3], [4] to propagate specified colors to similar regions in a single image. The number of interactions will increase linearly with the number of pictures that need to be colorized. Recent cartoon colorization works have focused on using deep learning methods to colorize sketch images automatically [5], [6], [7]. However, artifacts appear along the edges in these research works. Furthermore, the same character in a comic often has the same colors for the same semantic regions across different frames. The above methods fail to obtain satisfactorily coherent results for an image set. Hence, it is essential to develop methods to colorize the same character with consistent colors on the same semantic regions across different frames.

In this paper, we propose a novel method to colorize a given set of cartoon line drawings of a single character. This method is able to provide consistent colors for regions with the same semantics as in the reference image. To achieve this goal, we use an active-learning-based framework to match local regions between target images and the reference image. Then a mixed-integer quadratic programming method (MIQP) is used to refine the matching results by considering the contextual structure. This method could produce highly accurate results while significantly reducing the required user interaction. This method is also able to colorize multiple cartoon characters simultaneously when the individual reference image is given for each character. The final frames of 2D cartoons are normally produced by compositing foreground characters onto a static background, and this method can be easily integrated into the cartoon generation pipeline.

## 2 Related Work

In this section, we will briefly review related research works on image colorization and machine learning.

**Colorization without references:** Previous colorization methods [2], [3], [4] allow users to use brushes to apply the desired colors. Qu et. al. [2] propagated a user's scribbles throughout relevant regions in the image by using the level-set method. Recently, deep learning based methods have been proposed for colorization in either automatic or interactive manners [8], [9], [10], [11]. However, these methods may generate color strings along the border and tend to color the image differently from what the user specified. Many interactions are usually needed to refine the color results. Furthermore, when there are many images to be colorized, much manual effort is needed to perform these operations for each image to achieve consistency for the same character.

**Colorization with reference images:** In an animated scene, the character often wears the same clothes across many frames. One important problem is to colorize the character in the animation consistently with the colored reference image. Sato et al. [12] developed a graph-based method to encode each superpixel as one node. The pairwise similarity between these nodes is computed based on their area and relevant centroid vector. The superpixel-based segmentation needs to specify the number of regions. Moreover, their node representation framework ignores

† *Both authors contributed equally to this research.*
* *Corresponding author.*

- *S. Chen, L. Gao, Y. He and S. Xia are with the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100190, China. E-mail:{chenshuyu, gaolin, heyue, xsh}@ict.ac.cn*
- *J. Zhang and M. Shi are with North China Electric Power University, Beijing 102206, China. E-mail:{zhangjiaqi, shi_ min}@ncepu.edu.cn*
- *F. Zhang is with School of Engineering and Computer Science, Victoria University of Wellington, New Zealand. E-mail:fanglue.zhang@ecs.vuw.ac.nz*

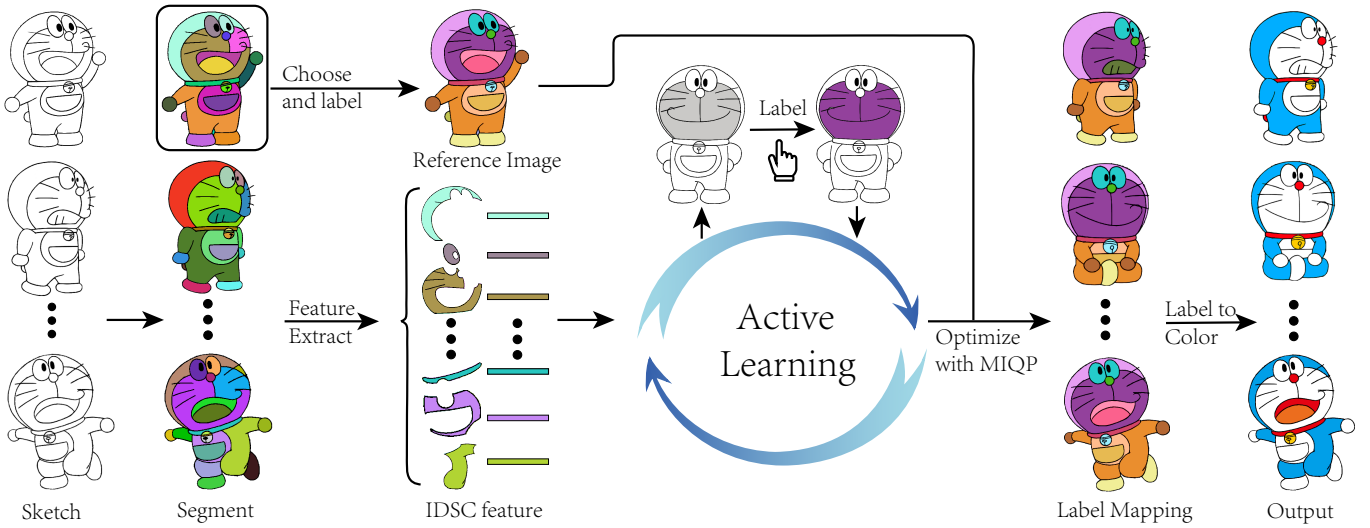



Fig. 1: This figure demonstrates the overview of our approach. We first automatically segment the sketched image, before extracting each segment's IDSC feature. Taking these features as input, we then apply the active colorization algorithm (Algorithm.1). Many more results are shown in Sec.4.2.

the shape information of regions it represents, and they are not entirely sure why the shape information leads to this failure. Thus, their methods fail when the character has large scale deformations and the number of nodes differs greatly between images. Sýkora et al. [13] proposed a method by using path-pasting to colorize continuously animated black and white cartoons. In a continuous sequence, where the cartoon has minimal deformations between frames, their method performs very well. However, when applied to discontinuous sketch images, their method struggles to find accurate correspondences. Researchers have applied deep learning models [5], [6], [7] to colorize cartoon images using reference images. Furusawa et al. [5] first described reference images using color histograms, before using this expression to determine how cartoon characters should be colored in subsequent frames. This method produces results with less detail than those depicted in the reference image while also reducing the amount of shading present in the image. Zhang et al. [6] used deep VGG features as a descriptor for the reference images, but their method produced unclear object boundaries and mixed colors. Hensman et al. [7] used a conditional GAN to colorize grayscale images. They took grayscale images as input and trained neural networks to learn the relationships between region colors and their gray values. However, these methods, which are based on DNNs, perform poorly when applied to cartoon styles that rely on pure color results, and tend to produce color mixing artifacts.

**Active Learning:** The active learning framework helps users select the data, and the data need to be tagged by learning from a few labeled samples. In this way, the accuracy of data labeling can be quickly improved by active learning. There have been many works on active learning for image classification. Guo et al. [14] explored the application of active learning in multilabel classification. They argue that a simplified feature labeling method can be used, as the missing information can be inferred from correlations between multiple features. Joshi et al. [15] provided a method that generalizes the application of the margin-based uncertainty to multiple classes. Li et al. [16] approached the problem differently by proposing an adaptive method that combines both the information density and a measurement of points with maximum uncertainty to determine critical instances to classify. Then, Gal et al. [17] developed an active learning framework for high dimensional data by combining Bayesian deep learning into the active learning framework.

## 3 Active Colorization

Given a reference colored image, we aim to colorize line drawings of the same object for the whole image set. The core problem to solve is how to match regions between the target image set and the reference image. We can only rely on the edge/shape similarity among regions of the line drawing image and regions of the reference image, instead of textural features as in other region/object matching tasks.

To address this problem, we build an active learning based framework. We label the regions in the reference image according to the semantic information and painted colors. Then, for all the target regions of the image set, we attempt to assign the corresponding reference label using the active learning method. We first perform superpixel segmentation before matching our regions with the inner-distance shape context (IDSC) (Sec.3.2). A limitation of this approach is that regions with the same semantic features vary significantly as an object deforms or when the object is occluded. Thus, we introduce the active learning method (Sec. 3.3) to find the region that minimizes risk effectively, and this region will be sent to the user for labeling. This significantly improves labeling accuracy without significantly impacting usability. For the remaining matching conflicts in label estimation by active learning, we supplement MIQP with label-adjacent constraints to further rectify ambiguities (Sec.3.4).

### 3.1 Data Collection

Currently, there is no public database focusing on sketches of animated characters. We manually screened the same characters from comic books and cartoons. For our application, we are only concerned about the cartoon characters in the images. Therefore, we segment our database of color images, separating the foreground from the background. We extracted the sketch from the


Fig. 2: (a) Line drawing images, (b-e) are the segmentation results of different methods, where each region is marked in a different color. (b) SEEDS [18], (c) Superpixel method of Liu et al. [19], (d) connected component analysis, (e) trapped-ball segmentation [20].

foreground using Photoshop. This produces a dataset of twelve animated characters taken from seven animated media.

We prepare a labeled sketch image for each character in advance, which specifies a semantic label ID for each region. Then, a list is maintained to store all label IDs for the different semantic parts. We ask the user to mark regions within the same semantic part of the character with the same label. For example, the left and right foot of the character in Fig. 1 will have the same label. By performing this step, we will enable our method to learn the diversity of shapes with the same semantic label and the overall structural information in a better way. However, some regions with the same color in the reference image, such as the pocket and the belly band, will be labeled differently due to their differing semantics.

The labeled reference images provide spatial relationships between the parts represented by labels. The first step in our process is to build an $m$ by $m$ adjacency matrix of labels (where $m$ is the total number of labels) for each image. The contents of the adjacency matrix, $M$, are defined by the relation between image labels, where if the a-th and b-th labels are connected, the (a,b) and (b,a) values of $M$ are set to 1; otherwise, they are set to 0. This adjacency matrix will be used in Equ.6 in Section 3.4. When matching regions, our current framework considers neighborhood relationships between regions and their corresponding labels. An issue we faced is that the labeled reference image may not include all neighborhood relationships. To address this issue, we optimize the neighborhood matrix dynamically in our active interaction step. If the user labels one region and the neighboring regions have been labeled, we will add this spatial relationship to the adjacency matrix $M$.

### 3.2 Segmentation and Region Matching

Our method takes line-drawing images as input, making it difficult to search for matching relationships between the local contents of images using patch-based methods [21] such as PatchMatch [22] or PatchTable [23]. To solve this problem, we segment the image into several small regions according to curvature features, which are then used to extract a descriptor of region features based on the shape.

To segment line drawing sketches properly, we also need to tackle the problem of discontinuous boundaries that occurs in some regions. Thus, we choose trapped-ball segmentation [20] to split target and reference images into small regions. To validate this segmentation method, we also compare it with other common segmentation approaches. As shown in Fig. 2(b)(c), we set 100 segments in SEEDS [18] and [19]. It is obvious that the results in (b) do not preserve the region shape; (c) may combine small parts and sketch lines into the same segment and cannot deal with leaking or connect the region segments. We also perform a simple connected component analysis, as shown in Fig. 2(d), where each connected region is marked with a different color. In Fig. 2(c)(d) we can see that regions with different semantics are merged because of leaking at the boundaries.

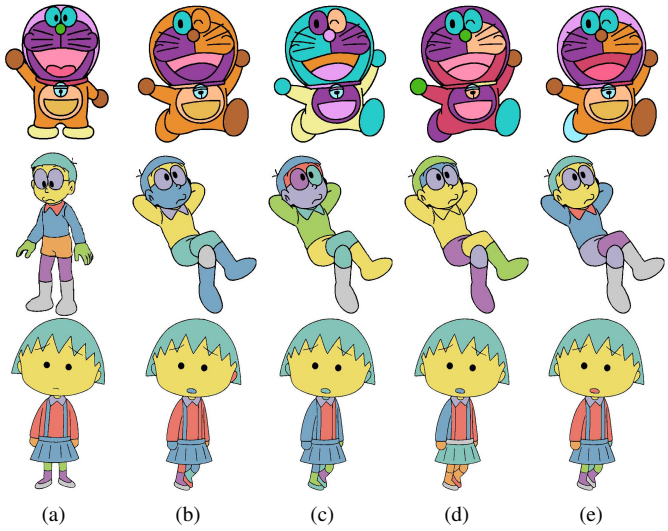



Fig. 3: Comparisons of different feature descriptors. (a) labeled images where the same color indicates the same label; (b) PatchMatch with a patch size of 7×7; (c) SIFT [24]; (d) shape context; (e) IDSC.

**Algorithm 1:** Framework of active colorization and MIQP.

$W \leftarrow CalculateWeightMatrix(S)$
$(ActiveLearning)$
**for** *each step* **do**
  **for** *each image* $I_i$ *in* $I_{tar}$ **do**
    $(MIQP)$
    $y \leftarrow max\{E = E_{match} + \lambda E_{local}\}$;
    $I_{color} \leftarrow Colorize(L_{color})$;
  **end**
  $S_{chose} \leftarrow FindMinimalRiskSegment(S, y)$
  *Label* $S_{chose}$ *with semantic label*;
  *Update neighbor relation* $M$;
  *Update label dataset* $D_{ref}$
**end**

Each segmentation may have a complex structure and internal holes, so we use IDSC [25] to match regions according to their shape information. The shape descriptor in IDSC can represent both the local features of sampling points and the global characteristics. The global feature information of local points is robust to occlusion and nonlinear distortion.

In IDSC [25], we uniformly sample $N$ points $P = \{p_i\}$ on the contour of a given region. The shape context of each sample point $p_i$ is represented by a histogram $h_i$ of the remaining $N-1$ points in the relative coordinate system. Given two regions and their contour point sets $P = \{p_i\}$ and $Q = \{q_i\}$, their matching cost

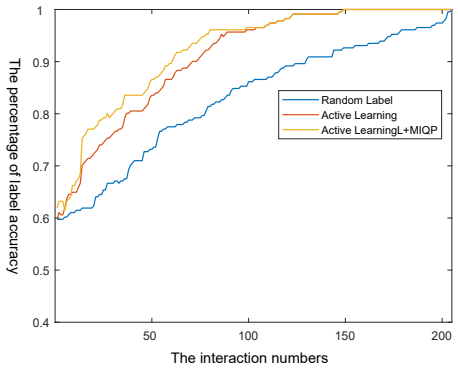
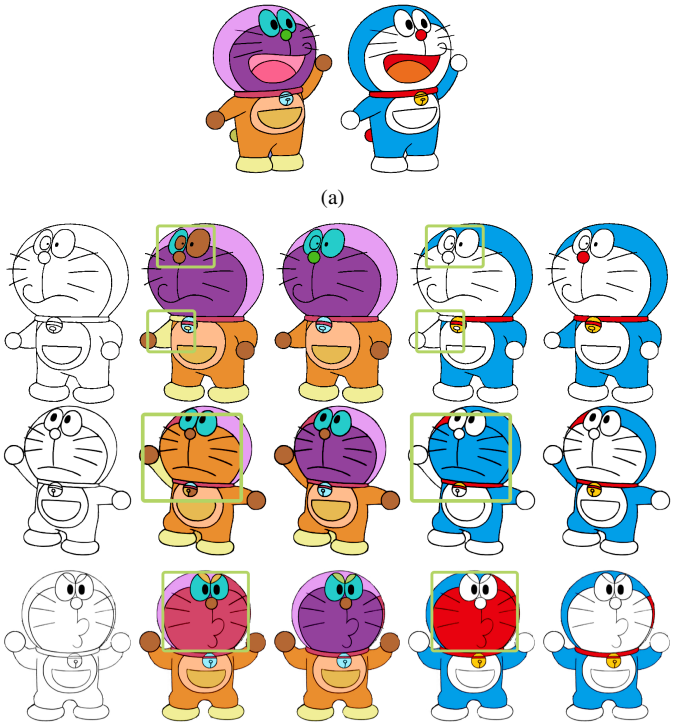



Fig. 4: Accuracy of three different labeling strategies are shown: the red curve is for the strategy using risk estimation, and the blue curve is for the random labeling method. The orange curve is the result from MIQP after active learning.



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

Fig. 5: Labeling results. (a) reference image and its labels. (b) input sketch image. (c) and (e) are the results without MIQP, (d) and (f) are results with MIQP. (c) and (d) are using colors to show the labels. (e) and (f) show the colorization results.

is calculated by the $\chi^2$ statistic. We use dynamic programming to find a matching $\kappa$ from $P$ to $Q$. We use $\kappa(i)$ to represent whether $p_i$ matches $q_{\kappa(i)}$. The total matching error between two shapes is computed as $E(\kappa)$:

$$E(\kappa) = \sum_{1 \le i \le N} e(i, \kappa(i)) \tag{1}$$

### 3.3 Active Learning

We reformulate the task of region matching to a labeling problem, which can be solved using active learning in an interactive way. As an initialization step, the reference image will be labeled manually, according to the semantic information and colors that have been introduced above. We define a real-valued function $f : L \cup U \rightarrow R$ to denote labels on every region, where $U$ and $L$ denote the unlabeled and labeled sets, respectively. In general, we suggest that similar unlabeled regions have the same label. In active learning, each region to be manually marked is returned to improve the classification accuracy of each segment. We use the method introduced in [26] to build our active learning framework. The Gaussian fields and harmonic functions are combined with semisupervised learning and active learning and are used to select the best candidate region to be manually labeled in each iteration to improve the labeling results.

We assume a connected graph $G = (V, E)$ is given with nodes $V$ corresponding to all $n$ regions after image segmentation. The edges $E$ are represented by an $n \times n$ weight matrix $W$. In the previous section, we introduced how to calculate the matching error for two regions, and we use that to build an $n \times n$ weight matrix $W$ for active learning. In each matching operation, we can obtain the matching loss $E_{i,j}(\kappa)$ and the number of matched point features $C_{i,j}$. The weight matrix $W$ used in our active learning method is expressed by

$$W(i,j) = \frac{E_{i,j}(\kappa)}{C_{i,j}} \tag{2}$$

Since we want similar nodes to have similar labels on the graph, we choose a quadratic energy function to represent the smoothness on the graph:

$$E(f) = \frac{1}{2} \sum_{i,j} w_{i,j} (f(i) - f(j))^2 \tag{3}$$

According to [26], the energy minimization function $f = argmin_{y|L=y_L} E(y)$ of the Gaussian random field is harmonic. The harmonic property means that the function value of each labeled region equals $y_L$, and is equal to the mean of the graph neighbor's values in each unlabeled region. Moreover, the harmonic energy minimization function $f$ can be computed with matrix methods by assigning the Laplacian matrix $\triangle = D - W$ to blocks for labeled and unlabeled nodes.

To select the best region to be marked in each iteration, we refer to [26], where they apply active learning with a Gaussian random field model by greedily selecting queries from unlabeled regions. The selected region minimizes the risk of the Bayesian classifier based on the harmonic energy minimization function, and the risk can be computed with matrix methods. In the official code provided by the authors of [26], the multiclass labeling method is supported. It is suitable to our needs, where $y_i$ can be represented by a one-hot vector.

To determine the impact our method has on accuracy, we compare it to a random labeling strategy (as shown in Fig. 4). The comparison clearly shows that our method converged to a more accurate state at a significantly faster rate than the random case.

### 3.4 Mixed-Integer Quadratic Programming

Using active learning, we can obtain the best correspondences between regions, but they are only based on the pairwise shape matching results. Thus, we further utilize the contextual relationship between semantic labels by the mixed-integer quadratic programming (MIQP) method. Our optimization objective is to maximize the following energy:

$$E = E_{match} + \lambda E_{local} \tag{4}$$

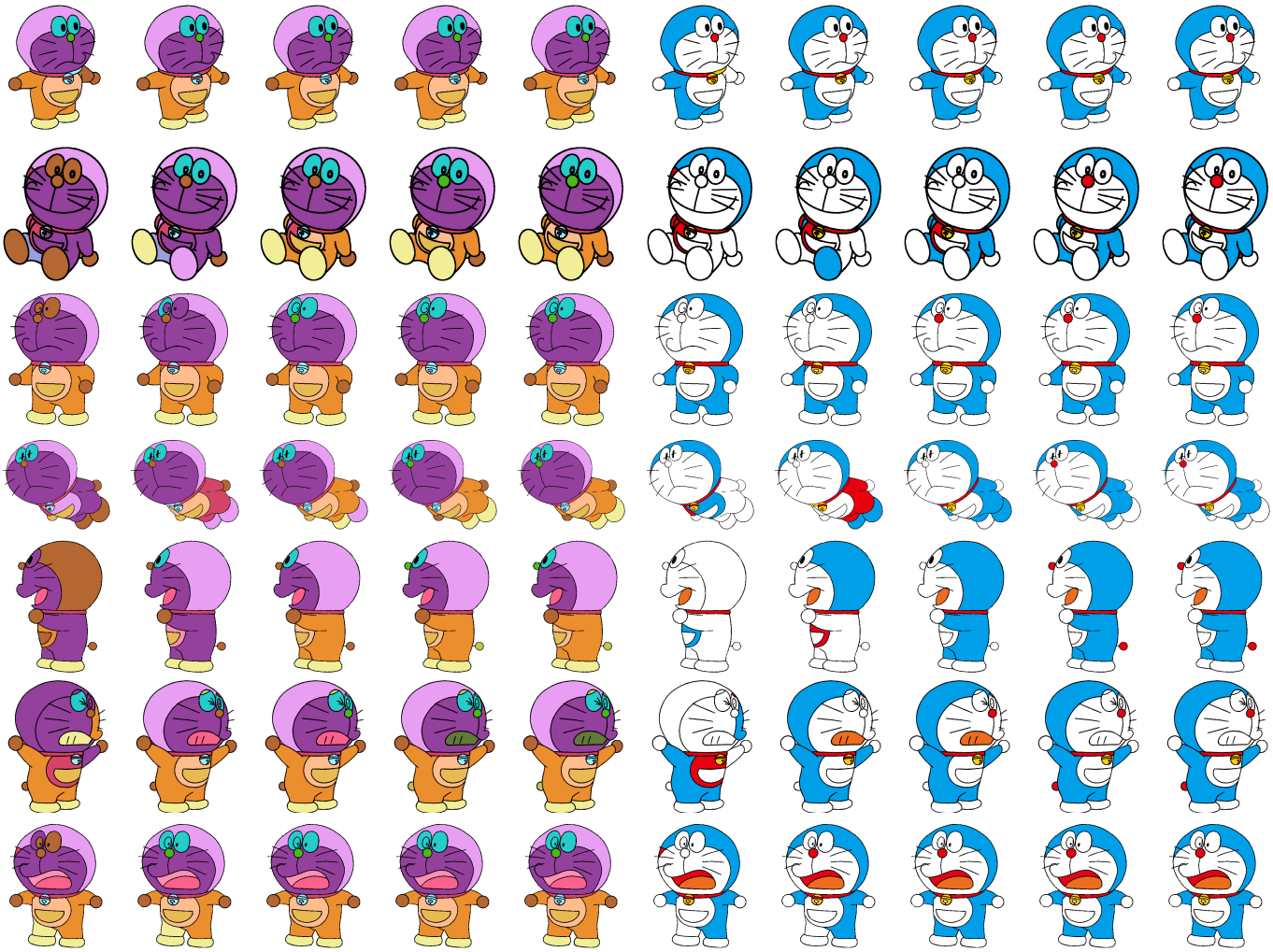



Fig. 6: Optimization results with MIQP. (a)∼(d) and (f)∼(i) show the results for 0, 50, 150, and 450 iterations, respectively; (e) and (j) show the ground-truth images. (a)∼(d) show the colorized images with different label colors, and (f)∼(i) show the re-colorized images with the actual color palette. This experiment uses 50 images with 1164 regions in total.

where

$$E_{match} = \sum_{i}^{N} c_i^T y_i \tag{5}$$

$$E_{local} = \sum_{i}^{N} \sum_{j \in Near(i)} y_i^T M y_j \tag{6}$$

Here, $E_{match}$ is an energy term for shape-based label matching, and $E_{local}$ is a local constraint that rewards regions maintaining the same neighbors between frames. $\lambda$ is used to adjust the impact of $E_{local}$. $c_i(j)$ is the probability of labeling $S_{tar}^i$ with $L_{ref}^j$. $y_i$ is the label of $S_{tar}^i$, which is represented by a one-hot vector. $M$ is the adjacency matrix of $L_{ref}$ and is computed by checking the spatial connectivity of the labeled regions in the reference image. $Near(i)$ represents the set of segments that neighbor $S_{tar}^i$. As shown in Fig. 4, an integer programming method can improve the accuracy based on the results labeled by active learning. The red line is the labeling accuracy using integer programming after each step in active learning. Integer programming is quite helpful to improve the accuracy. In Fig.5, we can see the final labeling results with and without MIQP. Some incorrectly labeled regions are fixed after integrating the spatial contextual information by MIQP, especially for regions with large deformations, but a coherent neighborhood relationship.

## 4 Experimental Results

We perform both a qualitative and quantitative evaluation of our method, before comparing it against the current state-of-the-art methods.

### 4.1 Qualitative Evaluation

Matching results using different numbers of iterations are shown in Fig. 6. The labeled ground-truth images are shown in Fig. 6(e). Fig. 6(a)∼(d) are the results with increasing numbers of manual labeling iterations; 0 iterations can be regarded as the results without active learning. As is clearly shown, the error is reduced as the number of iterations increases.

We also compare against existing methods using the same reference image as the input, as shown in Fig. 7. For the sake of a fair comparison, we do not use any user interaction, which means we run the MIQP without active learning after the initial IDSC matching results. Here, (a) and (f) are the reference images, and (b∼d) and (g∼i) are results from different methods. (b) and

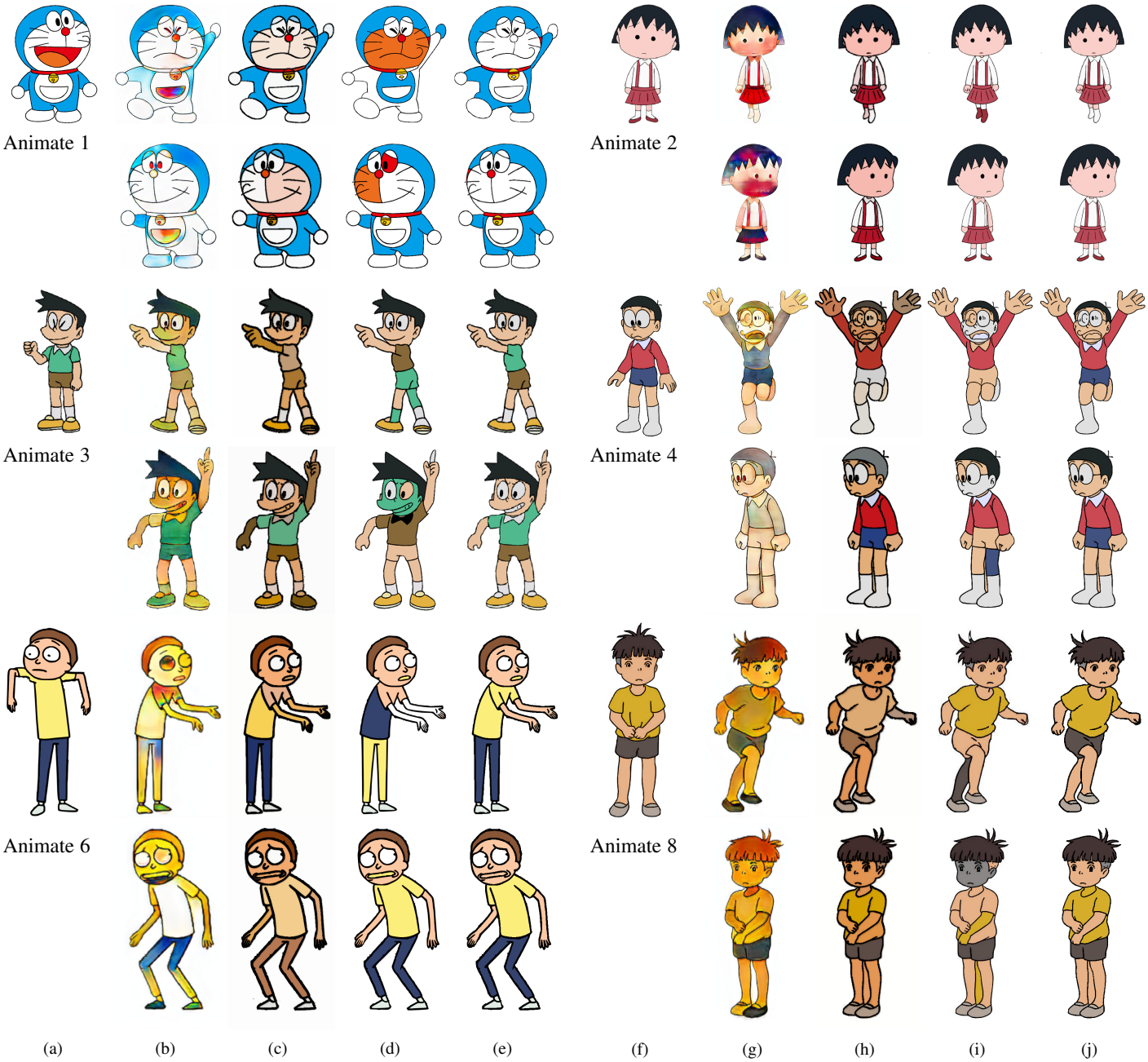



Fig. 7: Comparison with existing methods [6], [7], [12]. (a) and (f) are the reference images; (b) and (g) are the results of [6]; (c) and (h) are the results of [7]; (d) and (i) are the result of [12]; (e) and (j) are our results.

(g) are colorized by method [6] with reference images (a) and (f), respectively; (c) and (h) show the results of method [7], which are obtained by postprocessing; (d) and (i) are the results of method [12]. It can be seen that method [12] cannot handle cases when the character has large deformations or when the numbers of nodes between images differ greatly. Last, (e) and (j) are our results. Our colorization results correlate with the reference image far better than the other approaches.

Since our method is actually generating matching relationships between target regions and the reference region, if we want to re-colorize the whole image set with a new color theme, we just need to change the reference image. This method is far more convenient than other exsiting approaches. For example, method [7] requires a model to be retrained to specify color style. As shown in Fig. 8, we use two different color variants for each reference image to show how the colorization is impacted. It is clear that our method handles changing color much better than the other approaches.

### 4.2 Quantitative Evaluation

In Sec. 3.4, we test a parameter $\lambda$ on ten images to weight the neighbor relation energy impact. As shown in Fig. 9, the accuracy is tested for different $\lambda$ values, where the horizontal axis represents the number of interactions, and the vertical axis indicates the label accuracy percentage. We choose $\lambda = 0.1$ in the experiments, as this value performed best in this preliminary test.

To verify the stability of our method, we randomly select 50/75/100/125/150 images and segment them into 1100/1603/2180/2660/3200 regions to assess the performance. The total number of segmented regions is also the maximum

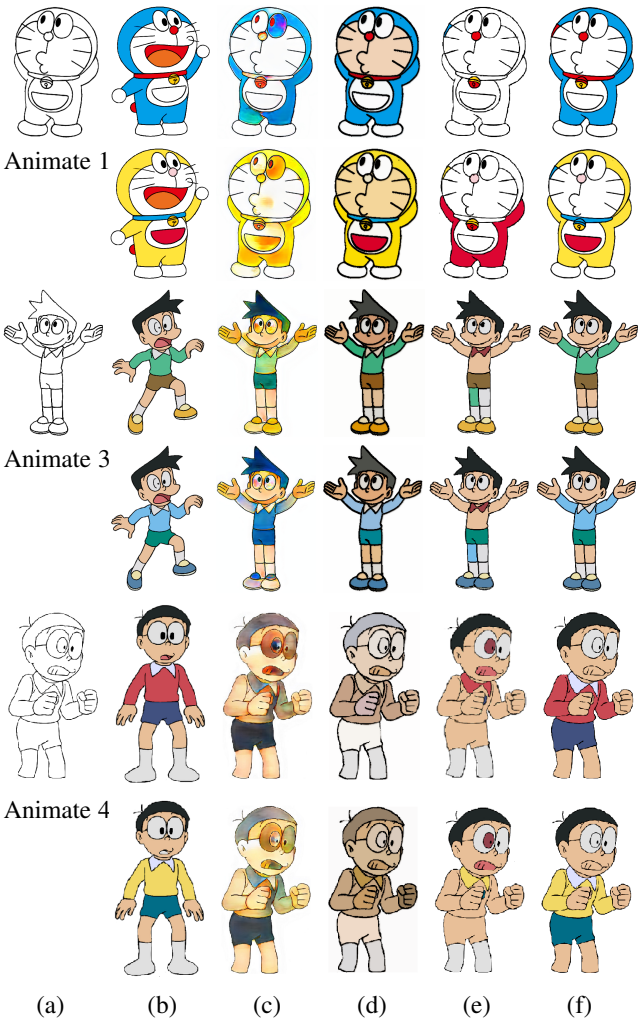



Fig. 8: Comparison against existing methods [6], [7] and [12] using the recolored reference images. (a) sketch images; (b) reference images in different color styles; (c) results of [6] (d) results of [7]; (e) results of [12]; (f) our results.

number of user interactions. Table 1 shows the percentage of user labeled regions needed to achieve colorization accuracies of $70\%/80\%/90\%/99\%$ for different numbers of images. It is clear that the colorization accuracy and required number of user interactions maintain a similar relationship regardless of the number of input images. For example, if a user labels $30\%$ of the regions, the accuracy is approximately $90\%$, which means that this approach could substantially reduce the labeling workload.

We also calculate the arithmetic mean of the MSE and PSNR over images in the test set from Fig. 7, as shown in Table 2. Our quantitative comparison results show that our method obtains better results without interaction than the other approaches and achieves significantly better results when interactions are used. The colorization result of method [6] can be improved semiautomatically by introducing user interactions. Unlike our method, it could not control the generated colorization results accurately. We conducted a user experiment to quantitatively compare our method with method [6]. We invite 20 participants to interactively color the same images within 6 minutes, and we quantitatively analyze the coloring results. For each animation, we randomly select 15 images. To test the method of [6] in a semiautomatic way, we first use automatic coloring to obtain the initial results, then we allow

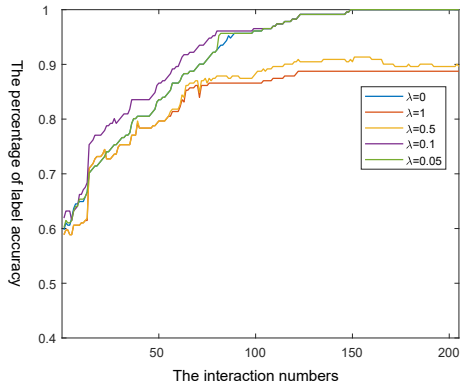



Fig. 9: The accuracy for different values of the balance parameter $\lambda$. By changing the value of parameter, we can adjust the strength of an adjacency relationship between segments, which in turn affects the MIQP optimization results.

| Image number | Segment number($\pm$10%) | Accuracy 70% | 80% | 90% | 99% |
|---|---|---|---|---|---|
| 50 | 1100 | 5.15 | 19.6 | 32.2 | 56.8 |
| 75 | 1603 | 6.8 | 20.0 | 30.9 | 52.6 |
| 100 | 2180 | 7.3 | 18.6 | 30.6 | 54.0 |
| 125 | 2660 | 6.6 | 17.5 | 29.9 | 52.8 |
| 150 | 3200 | 6.3 | 17.4 | 29.2 | 50.7 |

TABLE 1: Stability experiment with MIQP. We randomly choose different numbers of images to colorize and record the percentage of iterations with different coloring accuracies. As the number of images increases, the proportion of the number of interactions required to achieve certain accuracy levels remains the same.

participants to interactively refine the results using method [6]. We count the numbers of images that can be interactively refined within 6 minutes in Table 3. The interaction in [6] is complicated and it is not easy for users to complete an interactive refinement for a large number of images. As shown in Table 3, our method can improve the colorization results faster and more accurately than method [6].

### 4.3 Experiments on Scalability

#### *4.3.1 User Study*

As shown in Fig.10, a user interface is developed for users to perform an interaction required by this method. The label map

| | mMSE [6] | [7] | [12] | our | mPSNR [6] | [7] | [12] | our |
|---|---|---|---|---|---|---|---|---|
| Anime 1 | 0.053 | 0.042 | 0.045 | **0.042** | 13.09 | 14.04 | 15.39 | **15.69** |
| Anime 2 | 0.029 | 0.039 | 0.023 | **0.018** | 16.18 | 14.98 | 17.29 | **20.73** |
| Anime 3 | 0.016 | 0.025 | **0.005** | 0.008 | 17.98 | 16.13 | **24.83** | 22.24 |
| Anime 4 | 0.046 | 0.059 | 0.031 | **0.019** | 14.06 | 12.81 | 15.95 | **19.39** |
| Anime 5 | 0.010 | 0.031 | 0.012 | **0.009** | 20.06 | 15.34 | 20.18 | **20.66** |
| Anime 6 | 0.020 | 0.015 | 0.015 | **0.006** | 17.13 | 18.71 | 20.68 | **24.23** |
| Anime 7 | 0.017 | 0.024 | 0.015 | **0.013** | 17.95 | 16.30 | **20.07** | 19.39 |
| Anime 8 | 0.015 | 0.023 | **0.004** | 0.008 | 18.51 | 16.99 | **26.61** | 22.02 |
| Anime 9 | 0.049 | 0.026 | 0.033 | **0.009** | 13.68 | 15.95 | 18.23 | **23.15** |
| Anime 10 | 0.010 | 0.009 | 0.011 | **0.002** | 19.90 | 20.64 | 24.23 | **28.58** |
| Anime 11 | 0.024 | 0.025 | 0.017 | **0.013** | 16.38 | 16.78 | 20.51 | **21.26** |
| Anime 12 | 0.017 | 0.016 | 0.012 | **0.003** | 18.26 | 18.25 | 20.43 | **24.77** |

TABLE 2: Performance validation in Fig.7. We perform a quantitative evaluation of [6], [7], and [12] using the arithmetic mean of the MSE and PSNR. The results for the remaining animated characters are shown in the supplementary materials.

| | Image Nums | | mMSE | | mPSNR | |
|---|---|---|---|---|---|---|
| | [6] | our | [6] | our | [6] | our |
| Anime 1 | 5 | 15 | 0.0399 | **0.0015** | 16.16 | **29.69** |
| Anime 2 | 4 | 15 | 0.0322 | **0.0144** | 17.31 | **36.40** |
| Anime 3 | 5 | 15 | 0.0124 | **0.0020** | 19.61 | **29.55** |
| Anime 4 | 6 | 15 | 0.0268 | **0.0065** | 17.97 | **24.73** |
| Anime 5 | 4 | 15 | 0.0154 | **0.0096** | 18.34 | **21.03** |
| Anime 6 | 6 | 15 | 0.0096 | **0.0028** | 20.94 | **38.61** |
| Anime 7 | 5 | 15 | 0.0106 | **0.0105** | 20.68 | **22.15** |
| Anime 8 | 6 | 15 | 0.0089 | **0.0006** | 21.04 | **56.00** |
| Anime 9 | 8 | 15 | 0.0216 | **0.0003** | 18.57 | **37.48** |
| Anime 10 | 9 | 15 | 0.0053 | **0.0001** | 23.46 | **42.09** |
| Anime 11 | 4 | 15 | 0.0188 | **0.0032** | 18.03 | **33.24** |
| Anime 12 | 5 | 15 | 0.0121 | **0.0023** | 19.94 | **27.11** |

TABLE 3: Comparison of the effectiveness and accuracy of our method against semiautomatic method [6]. We evaluate the accuracy using the arithmetic mean of the MSE and PSNR, and we evaluate the effectiveness using the number of images refined by user interactions within the allowed time.

and true colors of the regions are shown on the upper-right corner. Users can click different buttons to switch between adding new labels, labeling regions and other operations. We conducted a 2-phase user study to verify its validity. Twenty participants were invited to colorize the sketch images manually, and using our method separately within two six-minute periods. Ten sketches are randomly selected for each user. To facilitate manual colorization, images are segmented into regions in advance, and users only need to specify the color of each segment. We compare the region-level colorization accuracy of two different methods by the percentage of correctly colored segments. As shown in Fig. 12, this method produces a more accurate result than the other method. To evaluate the required time to reach a given accuracy, we design a second phase of our user study.We invited another group of 20 participants and randomly selected 20 sketches for each participant. Fig. 13(a) shows the resulting pixel-level accuracy, and Fig. 13(b) shows the region-level accuracy. From these two images, our method obtains much more accurate results within the same amount of time.

#### 4.3.2 Multiple Characters

This active-learning-based colorization method for scenes could also be applied to scenes with multiple characters. Given individual reference images for each character, our method can find correspondences between regions in the target image and the reference image set. The results shown in Fig. 14 are generated without active learning iterations. It demonstrates that our colorization framework is still robust and can find correspondences effectively even in complex scenarios with multiple characters.

#### 4.3.3 Colorization with Gradients

In the previous sections, this method is mainly focused on providing coherent region-matching results across all cartoon frames. While this approach is not restricted to coloring regions with color, it could also be applied to color gradients. We refer to the nonrigid body registration method [27] to convert the gradient color of the reference region to that of the target.

More specifically, we first obtain region correspondences by our proposed region-matching method. Then, we use method [27] to register the source and target shapes. The registration is optimized by minimizing the corresponding point distance. The boundary correspondence from our region-matching step is used as a constraint condition to improve the accuracy of the registration distance calculated by the shape registration method. Then, the positions of grid control points are determined according

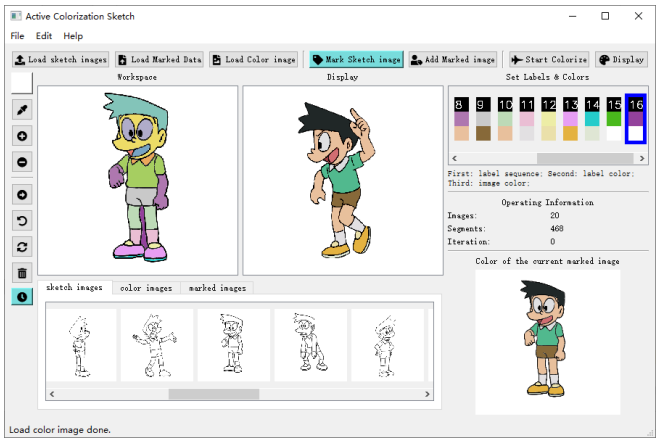
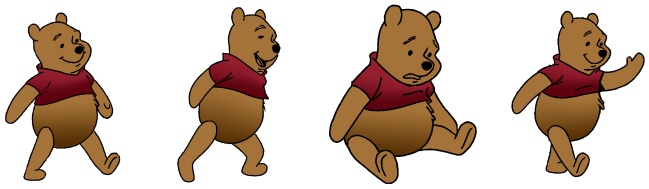



Fig. 10: User interface for colorizing cartoon line images with the proposed methods.



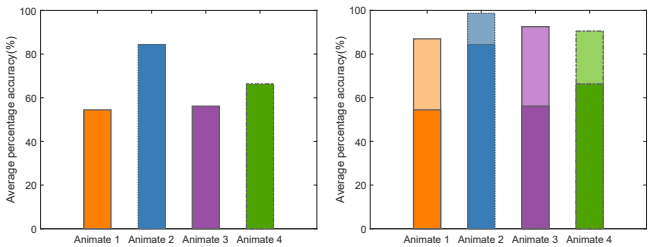
(a) (b) (c) (d)

Fig. 11: (a) Reference color image, (b-d) are the results of converting the reference image color to the target image.

to the registration method. Finally, we transform the reference to the target region by using these values. Fig. 11 shows one sequence of gradient coloring results.

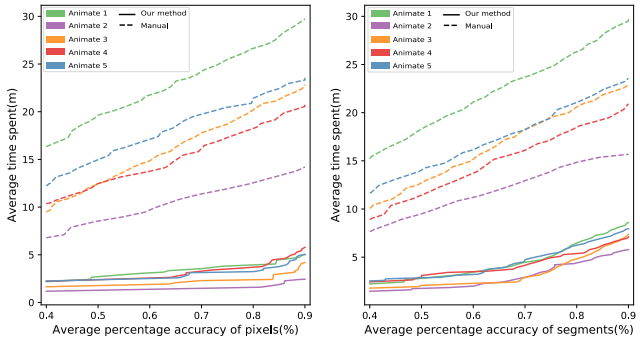
#### 4.3.4 Application of Region Matching

To test the capability of our proposed region matching framework, we apply it to natural scenes. We extract regions in natural images with similar semantic configurations, and use the region matching method to calculate region correspondences by using segment maps. Then, we perform color transfer with the matching results only. The color transfer operations are performed in $Lab$ color space. We transfer the average values of channels $a$ and $b$ of the reference region to all the pixels of the target region while keeping the value of the $L$ channel to maintain the textural information. Some of the final results are shown in Fig.15. This figure demonstrates that this approach is also able to obtain reasonable and meaningful color transfer results for natural images.




(a) (b)

Fig. 12: The accuracy of the results for two different colorization approaches: (a) manual and (b) our method. The users perform better using our method over the fully manual approach.




(a) (b)

Fig. 13: The relationship between user time and accuracy for different approaches: (a) the pixel-level and the (b) region-level accuracy.

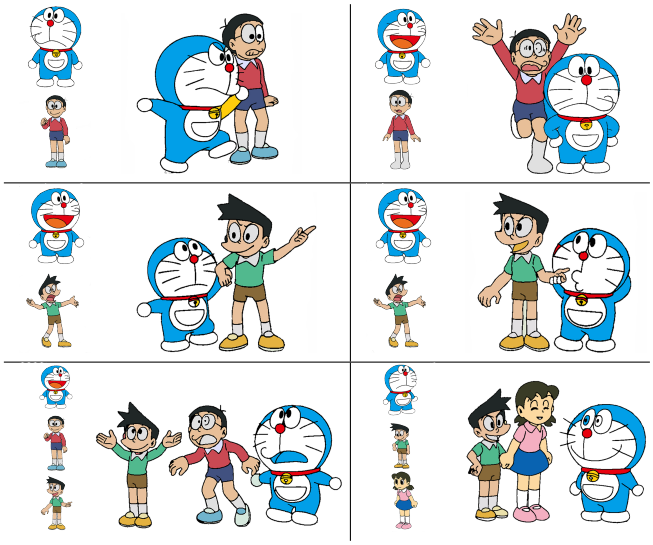


Fig. 14: Optimization results without active learning. Each image has multiple cartoon characters, and the reference images are shown on the left side of each subfigure.

### 4.4 Limitations

Our system is designed for the colorization of line drawings of cartoon characters, not for dealing with the background of scenes specifically. The large variety of region shapes in the backgrounds of cartoon sequences is too difficult to address with the current approach. Our method has another limitation when addressing multiple characters. Because our method do not have a character localization and identification module, this method sometimes failed at the initial region matching stage if the characters occlude each other. Solving this problem will require many more user interactions to obtain correct coloring results.

## 5 Conclusion

In this paper, we propose a novel colorization algorithm for line drawing images. Our method is built upon an active learning framework designed to match the target region to regions from the colored reference image. We build a weight matrix for the active learning algorithm using the similarity in terms of the shape features. By providing the best region to users for labeling, this approach could achieve a high region-matching accuracy. In addition, we introduce an integer programming method to further refine the labeling results. In future work, we will enlarge the image dataset and research the application of deep learning to the active learning method to improve the algorithm performance further.

Reference image Original image Color transfer

Fig. 15: Application of region correspondences. Changing the colors of objects in the original image using the colors of the reference image.

## Acknowledgement

This work was supported by Royal Society Newton Advanced Fellowship (No. NAF\R2\192151), National Natural Science Foundation of China (No. 61872440), CCF-Tencent Open Fund, Tencent AI Lab Rhino-Bird Focused Research Program (No.JR202024), Beijing Municipal Natural Science Foundation (No. L182016), Youth Innovation Promotion Association CAS and Victoria Early-Career Research Excellence Award. We thank Simon Finnie (CMIC, Victoria University of Wellington) for proofreading the article.

## References

[1] Toon Boom Animation. www.toonboom.com, 2020.
[2] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. Manga colorization. *ACM Trans. Graph.*, 25(3):1214–1220, July 2006.
[3] Daniel Skora, John Dingliana, and Steven Collins. Lazybrush: Flexible painting tool for hand-drawn cartoons. *Computer Graphics Forum*, 28(2):599–608, 2009.
[4] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, August 2004.
[5] Chie Furusawa, Kazuyuki Hiroshiba, Keisuke Ogaki, and Yuri Odagiri. Comicolorization: Semi-automatic manga colorization. In *SIGGRAPH Asia 2017 Technical Briefs*, SA '17, pages 12:1–12:4, New York, NY, USA, 2017. ACM.
[6] Lvmin Zhang, Chengze Li, Tien-Tsin Wong, Yi Ji, and Chunping Liu. Two-stage sketch colorization. *ACM Trans. Graph.*, 37(6), December 2018.
[7] Paulina Hensman and Kiyoharu Aizawa. cgan-based manga colorization using a single training image. *CoRR*, abs/1706.06918, 2017.
[8] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. *CoRR*, abs/1612.00835, 2016.

[9] Lvmin Zhang, Chengze Li, Tien-Tsin Wong, Yi Ji, and Chunping Liu. Two-stage sketch colorization. *ACM Trans. Graph.*, 37(6):261:1–261:14, December 2018.

[10] Yuanzheng Ci, Xinzhu Ma, Zhihui Wang, Haojie Li, and Zhongxuan Luo. User-guided deep anime line art colorization with conditional adversarial networks. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, pages 1536–1544, New York, NY, USA, 2018. ACM.

[11] Domonkos Varga, Csaba Attila Szabó, and Tamás Szirányi. Automatic cartoon colorization based on convolutional neural network. In *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, CBMI '17, pages 28:1–28:6, New York, NY, USA, 2017. ACM.

[12] Kazuhiro Sato, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. Reference-based manga colorization by graph correspondence using quadratic programming. In *SIGGRAPH Asia 2014 Technical Briefs*, SA '14, pages 15:1–15:4, New York, NY, USA, 2014. ACM.

[13] Daniel Sýkora, Jan Buriánek, and Jiří ára. Colorization of black-and-white cartoons. *Image Vision Comput.*, 23(9):767–782, September 2005.

[14] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, and Hong-Jiang Zhang. Two-dimensional active learning for image classification. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[15] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2372–2379. IEEE, 2009.

[16] Xin Li and Yuhong Guo. Adaptive active learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 859–866, 2013.

[17] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1183–1192. JMLR. org, 2017.

[18] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VII*, ECCV'12, pages 13–26. Springer-Verlag, 2012.

[19] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy rate superpixel segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2097–2104. IEEE, IEEE, 2011.

[20] Song-Hai Zhang, Tao Chen, Yi-Fei Zhang, Shi-Min Hu, and Ralph R. Martin. Vectorizing cartoon animations. *IEEE Transactions on Visualization and Computer Graphics*, 15(4):618–629, July 2009.

[21] Connelly Barnes and Fang-Lue Zhang. A survey of the state-of-the-art in patch-based synthesis. *Computational Visual Media*, 3(1):3–20, 2017.

[22] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (ToG)*, volume 28, page 24. ACM, 2009.

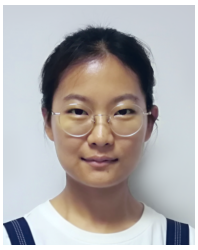
[23] Connelly Barnes, Fang-Lue Zhang, Liming Lou, Xian Wu, and Shi-Min Hu. Patchtable: Efficient patch queries for large datasets and applications. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, August 2015.

[24] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

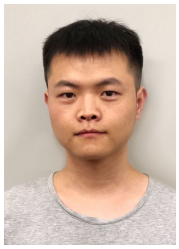
[25] Haibin Ling and David W. Jacobs. Shape classification using the inner-distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):286–299, February 2007.

[26] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, volume 3, 2003.

[27] Mohammad Rouhani and Angel D Sappa. Non-rigid shape registration: A single linear least squares framework. In *European Conference on Computer Vision*, pages 264–277. Springer, 2012.

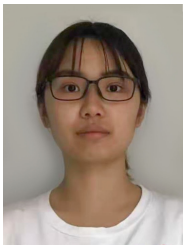



**Shu-Yu Chen** received the BS degree in computer science and technology from Zhengzhou University. She is currently working toward the PhD degree in Institute of Computing Technology, Chinese Academy of Sciences. Her research interests include computer graphics and geometry processing.



**Jia-Qi Zhang** received the BS degree in software engineering from North China Electric Power University. He is currently working toward the master degree in North China Electric Power University. His research interests include computer graphics.

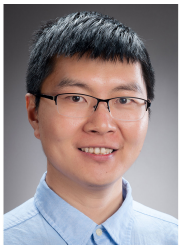
**Lin Gao** received the bachelor's degree in mathematics from Sichuan University and the PhD degree in computer science from Tsinghua University. He is currently an Associate Professor at the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include computer graphics and geometry processing. He received the Newton Advanced Fellowship award from Royal Society in 2019.



**Yue He** received the BS degree in computer science and technology from University of Chinese Academy of Sciences. She is currently working toward the master degree in Institute of Computing Technology, Chinese Academy of Sciences. Her research interests include computer graphics.

**Shihong Xia** is a professor associated with the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences. He received the BS degree in Mathematics from the Sichuan Normal University, China in 1996 and the PhD degree in Computer Software and Theory from University of Chinese Academy of Sciences in 2002. His research interests include computer graphics and virtual reality.

**Min Shi** is an associate professor in the school of Control and Computer Engineering, North China Electric Power University. She received her Ph.D. degree in computer science and technology from Chinese Academy of Sciences in 2013. Her research interests include cloth simulation, data analysis and visualization and virtual reality.



**Fang-Lue Zhang** is currently a lecturer with Victoria University of Wellington, New Zealand. He received the Bachelors degree from Zhejiang University, Hangzhou, China, in 2009, and the Doctoral degree from Tsinghua University, Beijing, China, in 2015. His research interests include image and video editing, computer vision, and computer graphics. He is a member of IEEE and ACM. He received Victoria Early-Career Research Excellence Award in 2019.