Improved inclusion matching for animation paint bucket colorization

Yuqin Lei
the Department of Computing, the Hong Kong Polytechnic University, Hong Kong, 999077, China
yuqin.lei@connect.polyu.hk.

ABSTRACT

The celluloid style is usually characterized by clear lines, distinct color blocks, and sharp contrast between light and dark, etc. When it comes to celluloid-style cartoons, it involves colorizing the line-enclosed segments of line art frame by frame. In the past decades, with the popularization of computer technology, practitioners commonly utilize paint bucket tools to perform line art colorization tasks, based on RGB values predetermined by a color designer. Nevertheless, it is still laborious regarding diverse color segments, segment matching and the large number of frames. Concerning that, a number of automated methodologies have been devised. The methodology named inclusion matching proposed by a group in NTU is advanced and practical. To a large extent, it can effectively address issues like occlusion or wrinkles that arise among frames. The inclusion matching pipeline is based on deep neural networks. From coarse to fine, it starts to warp the line art for extracting features and then performs inclusion matching using the attention mechanism. However, this pipeline ignores the global information of line art. Inspired by the vision transformer, the present study introduces a new mechanism to enhance the inclusion matching module. Experiments depict the effectiveness of our techniques.

Keywords: Deep learning, cartoonization, transformer, colorization

1. INTRODUCTION

Celluloid-style works' characteristics make them visually very impactful and recognizable. Till now, celluloid works still attract enthusiasts. First, artists produce clear and concise line drawings. This is the basis of the entire coloring process. Lines mainly serve to define shapes and outlines, and their colors are mostly monochrome or simple strokes. However, in some cases, in order to enhance the visual effect, lines of different colors or thicknesses may be used to distinguish different areas or emphasize specific details, such as highlights in the eyes. Based on the line drawing and keyframes, which are colorized, fill the basic line-enclosed color blocks in the in-between frames with colors. This step is usually performed by applying paint bucket tools. It is a time-consuming endeavor because of hundreds of manual clicks. Software applications like Retas Studio Paintman, OpenToonz, and CLIP Studio Paint have been developed to tackle this problem. Many effective functions are offered for help. Despite these technological advancements, the pursuit of fully automated colorization is ongoing, with various methods being explored and proposed.

Graph-based methodologies [1-5] view individual segments as graph nodes, interconnecting them through edges based on their proximity. But they are computationally demanding. To address this, Casey et al. [6] suggested the Animation Transformer (AnT) and its application Cadmium. AnT streamlines the segment alignment process by standardizing their sizes and leveraging Convolutional Neural Networks (CNNs) for extracting salient features. However, in more complex scenarios, particularly those characterized by occlusion or substantial motion, AnT's performance becomes less robust, struggling to maintain the same level of accuracy. The inclusion matching method raised by Yuekun Dai et al. [7] partially solved the issue. Their approach computes the likelihood of each segment in a target frame being included within a specific region of a reference frame rather than direct segment matching. Although the inclusion matching method has excellent results, the model is still not perfect because of the lack of global information. In this study, we enhanced the inclusion matching module by concatenating a new token with the segments' features. The new token represents "class" and will acquire global information from attention layers.

2. RELATED WORK

2.1 Correspondence matching

To match the correspondence, a number of researchers attempt to match regions of consequent frames or relevant images in feature space. Taking advantage of deep neural networks, it is common and effective to extract usable high-level feature maps of original images.

Fourth International Conference on Advanced Algorithms and Neural Networks (AANN 2024),
edited by Weishan Zhang, Qinghua Lu, Proc. of SPIE Vol. 13416, 134163L
 · 0277-786X · doi: 10.1117/12.3049723

Tasks like Video tracking [8, 9] and exemplar-based colorization [10, 11] adopt this framework. Despite showing promising prospects, there exist innate limitations. Pixel-level operations require a large amount of computational resources, especially referring to high-resolution images. Naturally, region representations via patches [12, 13] or descriptors [14, 15] are being explored.

2.2 Segment matching

In the celluloid-style cartoon process, shapes are clearly defined because lines surround segments. Segment-wise colorization is another point of view and demonstrates its potential.

Graph-based approaches [1-5] transform the segments into graph nodes, trying to solve the matching problem from the perspective of graph optimization. These methods have certain functions and effects, but the computational cost is high.

Another proposal [16] is to generate feature maps by computing Hu moments [17] and then feeding the moments to U-Net [18]. After mapping segments to the feature dimension, calculating distance/loss among segments contributes to matching.

Recently, Transformers [19] are getting more and more popular. Transformer-based architectures are proven to be significantly successful in several domains. On this basis, Casey et al. [6] introduce an effective architecture "The Animation Transformer" (AnT). They leverage CNN for extraction and apply a multiplex transformer architecture for feature aggregation and segment matching.

3. METHOD

3.1 Inclusion matching

Inclusion matching [7] is an advanced methodology suggested by a research group from NTU (Nanyang Technological University). Its architecture shows excellent performance and can better deal with occlusion, wrinkles, and large movements. Those are the problems that are usually hard to solve. The present study mainly incorporates this approach into our research.

As shown in Figure 1, the architecture can be decomposed into several modules.

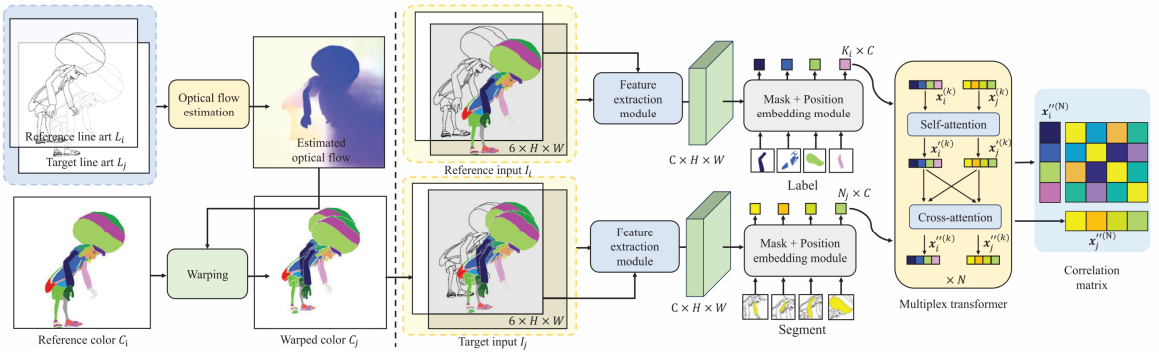

Figure 1. Paint bucket colorization architecture based on inclusion matching. (from [7]).

First, in the Color Warping Module, reference line art L_i and target line art L_j are utilized to estimate the optical flow through the optical flow estimation model RAFT [20]. Then the segments are recolorized and encoded with index labels, transforming reference line art L_i into a color image C_i. Accordingly, a new warped color image C_j is produced using the reference color image C_i and the estimated optical flow. Second, the Feature Extraction Module employs deformable convolution kernels [21] to align images, a CLIP [22] encoder to obtain textual information, and a lightweight U-Net to encode and decode features. It outputs new representations which are then fed to the Mask + Position Embedding Module. With the segment position input, feature maps are tokenized through the Mask + Position Embedding Module. Each token represents one segment, containing color information, semantic information, textual information, and positional information. To be specific, a set of tokens (also called a descriptor) $X_i \in R^{K_i \times C}$ denotes the reference line art and descriptor $X_j \in R^{N_j \times C}$denotes the target line art. K_i and N_j is the number of tokens/segments. C is the feature dimension. They can be written as follows:

$$X_i = \left(x_{i,1}, x_{i,2}, \cdots, x_{i,K_i}\right) \tag{1}$$

$$X_j = \left(x_{j,1}, x_{j,2}, \cdots, x_{j,N_j}\right) \tag{2}$$

At last, a Multiplex Transformer built upon AnT [6] and SuperGlue [14] will process tokens. There are N blocks inside the Transformer, each of which has a self-attention layer and a cross-attention layer. For each attention layer, the output is added to the original X_i and X_j and then sent to an MLP. Repeat N times and the final result is $\widehat{X_i} \in R^{K_i \times C}$ and $\widehat{X_j} \in R^{N_j \times C}$.

Based on that, a correlation/similarity matrix $S \in R^{N_j \times K_i}$ is calculated:

$$S_{mn} = \frac{\exp(\widehat{x_{im}} \cdot \widehat{x_{jn}})}{\sum_{m=1}^{N_j} \exp(\widehat{x_{im}} \cdot \widehat{x_{jn}})} \tag{3}$$

Where m is the index of segments in the target frame and n is the index of segments in the reference frame. This matrix can be regarded as the color probability. Each element is interpreted as the likelihood of the segment tokens matching. $\widehat{y_m} \in R^{K_i}$ indicates the matching probability of the segment m in the target frame and one certain segment in a reference frame. Finally, we choose cross-entropy loss as the loss function to optimize parameters:

$$L_{CE} = -\sum_{m=1}^{K_i} y_m \log(\widehat{y_m}) \tag{4}$$

3.2 Improvement

We notice that the token processing progress only considers relationships among segments. As animation frames are sequential and frames look similar somehow, it can be said that the reference line art and target line art should have similar or even the same global information.

Inspired by ViT (Vision Transformer) [23], in this study, a class token is added to each of the descriptors X_i (reference line art) and X_j (target line art) before passing through self-attention layers in the Transformer:

$$X_i' = \left(x_{i,class}, x_{i,1}, x_{i,2}, \cdots, x_{i,K_i}\right) \tag{5}$$

$$X_j' = \left(x_{j,class}, x_{j,1}, x_{j,2}, \cdots, x_{j,N_j}\right) \tag{6}$$

$x_{i,class}$ and $x_{j,class}$ are randomly initialized C-dimensional vectors which are considered parameters and will be optimized. After going through self-attention layers, $x_{i,class}$ and $x_{j,class}$ are able to learn global representations of all other tokens of the respective descriptors. Then, $x_{i,class}$ and $x_{j,class}$ will be extracted to avoid sending to cross-attention layers because the only goal of them is to obtain each descriptor's global information. Since the Multiplex Transformer consists of N blocks, the process of $x_{i,class}$ and $x_{j,class}$ repeats N times. In the end, only $\left(x_{i,0}, x_{i,1}, \cdots, x_{i,K_i-1}\right)$ and $\left(x_{j,0}, x_{j,1}, \cdots, x_{j,N_j-1}\right)$ are leveraged to generate the similarity matrix. As mentioned, $x_{i,class}$ and $x_{j,class}$ should have the same value because global information is considered consistent. The last extracted output of $x_{i,class}$ and $x_{j,class}$ are used to calculated another loss L_{CLS}(in this study, two kinds of loss function are tested. They are interchangeable):

1. Squared Difference Loss. It calculates the squared differences between $x_{i,class}$ and $x_{j,class}$ after propogation:

$$L_{CLS} = \left(x_{i,class} - x_{j,class}\right)^2 \tag{7}$$

2. Cosine Loss. The dot product of two vectors is divided by the product of their moduli to obtain a value between -1 and 1, which represents the similarity between the two vectors. Larger values indicate higher similarity between the two vectors, and smaller values indicate lower similarity. The purpose of taking the complement of the cosine similarity is usually to convert the measure of similarity into a loss value, so that minimizing the loss function corresponds to maximizing the similarity. Specifically::

$$L_{CLS} = 1 - \frac{x_{i,class} \cdot x_{j,class}}{\|x_{i,class}\| \|x_{j,class}\|} \tag{8}$$

In order to control the proportion of two losses, a multiplication coefficient α is added to L_{CLS}. α is a hyperparameter. Therefore, the new loss function is illustrated below:

$$L_{total} = L_{CE} + \alpha L_{CLS} \tag{9}$$

The whole procedure is shown in Figure 2. The class token forms a general expression of the line art during training. Moreover, while keeping the class token unchanged, the model is forced to learn the capability of avoiding the loss of global information.

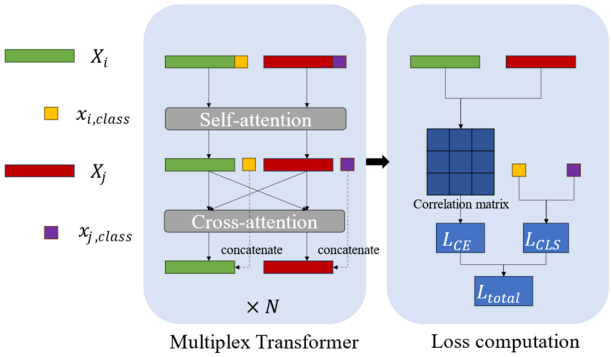

Figure 2. Improved Transformer architecture and diagram of loss computation.

4. EXPERIMENTS

4.1 Dataset and implementation details

PaintBucket-Character [7] is a dataset used for training and testing, which consists of 11,345 training images and 3,200 test images. This dataset only focuses on characters and is made from 3D character animation rendered in a flat color style. It has already shown good performance in previous work [7].

For implementation, two NVIDIA GeForce RTX 3080ti GPUs are emploied. The total iterations is 300,000 and each card carries one batch during training. So, for each iteration. 2 batches are processed.

Basically, to better compare with the inclusion matching model [7], the improved whole structure adopts the same hyperparameters, except α which is newly added and is set to 0.1. Concretely, we use the Adam optimizer with a learning rate of 10-4 and no weight decay. The pre-trained optical flow estimation module and the CLIP encoder are frozen, which means their parameters will not be updated. The feature dimension C extracted by the U-Net is assigned to 128. As for the multiplex transformer, the number of blocks (denoted as N) is configured as 9, and the number of heads is defined as 4.

4.2 Comparison

We make a comparison between the original inclusion matching model and ours. In addition, squared difference loss and cosine loss are both taken into consideration. The comparison relies on the test set of the PaintBucket-Character dataset.

Segment-wise accuracy is indicated by the terms "Acc" and "Acc-Thres," which offer information about possible task reductions for digital painters. In ‘Acc-Thres’, Segments smaller than 10 pixels are thresholded out. The visualization performance is represented by the terms "PixAcc," "Pix-F-Acc," and "Pix-B-MIoU," which stand for pixel-wise accuracy, foreground pixel-wise accuracy, and pixel-wise background MIoU, respectively.

Table 1. Quantitative comparison of our method with original inclusion matching methods.

Method	Acc	Acc-Thres	Pix-Acc	Pix-F-Acc	Pix-B-MIoU
Inclusion matching (data from [7])	0.8266	0.8726	0.9905	0.9724	0.9948
Inclusion matching (on our device)	0.8256	0.8642	0.9877	0.9678	0.9913
Ours(with squared difference loss)	0.8350	0.8724	0.9889	0.9739	0.9917
Ours(with cosine loss)	0.8347	0.8720	0.9896	0.9730	0.9934

Referring to the data in Table 1, comparing the results of our method with those in the reference paper, it can be clearly seen that our improvements are effective in the two indicators of Acc and Pix-F-Acc.

In addition, the first two rows compare the data in the reference paper with the data running on the device used in this experiment. They are not the same, because, on different devices, even if the same seed is used, the results may be different.

This is because there may be differences among different devices (hardware differences, software versions, parallel computing).

If the comparison is based on the results on the device used in this experiment (that is, comparing the last three rows), the improved method has shown enhancements in all indicators. (Besides, due to time and financial reasons, the value of the α coefficient has not been perfectly adjusted.) It can be reasonably inferred that our improved method is absolutely effective for inclusion matching.

Besides, the visual comparison shown in Figure 3 confirms our method's improvement.

Figure 3. Visual comparison of different methods. From left to right, they are: Inclusion matching (on our device), Ours (with squared difference loss), and Ours (with cosine loss).

5. CONCLUSION

Automated colorizing celluloid-style cartoons is a promising task that can form a more efficient cartoon production process. We propose a new method derived from inclusion matching [7]. In our method, a token (regarded as a class token) is introduced into the Multiplex Transformer. During training, the class token creates a broad expression of the line art. Furthermore, the model is compelled to acquire the ability to prevent the loss of global information while maintaining the class token stable. The outcomes of the experiment prove the success of our technique.

However, our research also has certain limitations. For example, the whole optimization details (such as initialization, loss function, or hyperparameters) may be carefully revised after the change of the architecture. Besides, when the image character approaches the boundary or changes its posture, it may result in significant changes in the shape of the segments or the formation of unexpected segments. There may be errors in the matching during colorizing. In addition, incorrect matches might also occur in areas with small areas but large quantities. Future studies can further explore this field more comprehensively.

REFERENCES

[1] Liu, Shaolong, Xingce Wang, Zhongke Wu, and Hock Soon Seah.. "Shape correspondence based on Kendall shape space and RAG for 2D animation." The Visual Computer 36 (2020): 2457-2469.

[2] Liu, Shaolong, Xingce Wang, Xiangyuan Liu, Zhongke Wu, and Hock Soon Seah. "Shape correspondence for cel animation based on a shape association graph and spectral matching." Computational Visual Media 9, no. 3 (2023): 633-656.

[3] Maejima, Akinobu, Hiroyuki Kubo, Takuya Funatomi, Tatsuo Yotsukura, Satoshi Nakamura, and Yasuhiro Mukaigawa.. "Graph matching based anime colorization with multiple references." In ACM SIGGRAPH 2019 Posters, pp. 1-2. 2019.

[4] Zhang, Lei, Hua Huang, and Hongbo Fu.. "EXCOL: An EXtract-and-COmplete layering approach to cartoon animation reusing." IEEE transactions on visualization and computer graphics 18, no. 7 (2011): 1156-1169.
[5] Zhu, Haichao, Xueting Liu, Tien-Tsin Wong, and Pheng-Ann Heng. "Globally optimal toon tracking." ACM Transactions on Graphics (TOG) 35, no. 4 (2016): 1-10.
[6] Casey, Evan, Víctor Pérez, and Zhuoru Li.. "The animation transformer: Visual correspondence via segment matching." In Proceedings of the IEEE/CVF international conference on computer vision, pp. 11323-11332. 2021.
[7] Dai, Yuekun, Shangchen Zhou, Qinyue Li, Chongyi Li, and Chen Change Loy.. "Learning Inclusion Matching for Animation Paint Bucket Colorization." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 25544-25553. 2024.
[8] Vondrick, Carl, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murph. "Tracking emerges by colorizing videos." In Proceedings of the European conference on computer vision (ECCV), pp. 391-408. 2018.
[9] Lai, Zihang, and Weidi Xie. "Self-supervised learning for video correspondence flow." In BMVC, 2019..
[10] Meyer, Simone, Victor Cornillère, Abdelaziz Djelouah, Christopher Schroers, and Markus Gross.. "Deep video color propagation." In Proceedings of the British Machine Vision Conference BMVC, 2018.
[11] Zhang, Bo, Mingming He, Jing Liao, Pedro V. Sander, Lu Yuan, Amine Bermak, and Dong Chen.. "Deep exemplar-based video colorization." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 8052-8061. 2019.
[12] Jabri, Allan, Andrew Owens, and Alexei Efros. "Space-time correspondence as a contrastive random walk." Advances in neural information processing systems 33 (2020): 19545-19560.
[13] Bertasius, Gedas, Heng Wang, and Lorenzo Torresani. "Is space-time attention all you need for video understanding?." In ICML, vol. 2, no. 3, p. 4. 2021.
[14] Sarlin, Paul-Edouard, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. "Superglue: Learning feature matching with graph neural networks." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4938-4947. 2020.
[15] Luo, Zixin, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan.. "Contextdesc: Local descriptor augmentation with cross-modality context." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 2527-2536. 2019.
[16] Dang, Trung DQ, Thien Do, Anh Nguyen, Van Pham, Quoc Nguyen, Bach Hoang, and Giao Nguyen.. "Correspondence neural network for line art colorization." In ACM SIGGRAPH 2020 Posters, pp. 1-2. 2020.
[17] Hu, Ming-Kuei. "Visual pattern recognition by moment invariants." Proc. IRE 49 (1961): 1428.
[18] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." In Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18, pp. 234-241. Springer International Publishing, 2015.
[19] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." Advances in neural information processing systems 30 (2017).
[20] Teed, Zachary, and Jia Deng. "Raft: Recurrent all-pairs field transforms for optical flow." In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, pp. 402-419. Springer International Publishing, 2020.
[21] Dai, Jifeng, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. "Deformable convolutional networks." In Proceedings of the IEEE international conference on computer vision, pp. 764-773. 2017.
[22] Cherti, Mehdi, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. "Reproducible scaling laws for contrastive language-image learning." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2818-2829. 2023.
[23] Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." In International Conference on Learning Representations. 2020.