ユーザーによる修正が可能なセルアニメーション彩色のための領域マッチング

Pablo Garcia Trigo[†] Henry Johan[‡] 今給黎 隆[†] 西田 友是[†]
東京大学[†] Nanyang Technological University[‡]

1. Introduction

Traditional 2D animated cartoons are created by drawing and coloring each frame manually. Compared to other ways of making animations, it is a very flexible means, but it is also a very time-consuming process.

The traditional 2D animation pipeline consists of several steps, being the generation of inbetweenings and the coloring the most time-consuming ones (approx. to 60% of the time of the whole process [1], even when using commercial software like Retas or Animo). Thus, previous research has concentrated on automating them.

This paper proposes an interactive method for the coloring step. The coloring is done gradually and the user can fix coloring mistakes as soon as they appear. In doing so, mistakes do not propagate to the rest of the animation frames nor generate other mistakes, thus reducing the total number of them and reducing the total effort needed until having a correctly colored cartoon.

2. Related work

There are several papers that attempt to automate the coloring process using a matching algorithm: they try to identify what regions in one frame correspond to what regions in the other frames comparing its features and then painting them with the same color. Since matching can be a very difficult task due to the possible lack of coherence between frames, approaches like [2] and [3] increase the matching accuracy by using graphs, building a hierarchy of regions or inserting skeletons.

While the above approaches can color automatically certain kinds of animations, they may fail almost completely in more difficult ones, leaving the artist with the task of correcting lots of wrongly colored regions manually. This happens because a wrongly matched region in a frame in the middle of the animation usually keeps being wrongly matched in the successive frames. Even worse, that wrong match is used later as information for other matches, worsening their accuracy.

「User-guided matching for animated cartoon coloring」

† Pablo Garcia Trigo, Takashi Imagire, Tomoyuki Nishita
The University of Tokyo
‡ Henry Johan
Nanyang Technological University

In our method the user fixes the matching mistakes as soon as they appear. Thus, the previous situation does not happen and, on the whole, it takes less effort for the animator to color the frames.

3. Proposed method

3.1. General overview

Our coloring algorithm goes as follows:

1. Scan the hand-drawn frames
2. Apply filters to reduce noise and holes
3. Segment each input frame into its closed regions
4. For each region extract the its features
5. User-guided matching

Note that we work with raster images and we do not vectorize the input frames.

3.2. Region features

For each region we extract the following features: Area, Position, Neighbor Regions and Dominant Points. Dominant Points are points in a boundary that have a high curvature. They are also called Character Points in related works.

3.3. The matching algorithm

We have implemented a forward matching algorithm starting from the first frame. Inside each frame, we try to match all regions, starting from the biggest to the smallest. Given a region r in a frame f, we compare r with all the non-matched regions in the frame $f+1$ and pick up the most similar as the match. When done with all regions in frame f, we advance to frame $f+1$ and repeat the process until we reach the last frame.

3.4. The comparison function

For comparing two regions we use a comparison function *comp* that given two regions returns a score indicating how similar they are. It does so by comparing the region features one by one and normalizing them into one final score.

3.5. Interactive matching

The user interface consists of a set of panels that display the frames to be matched.

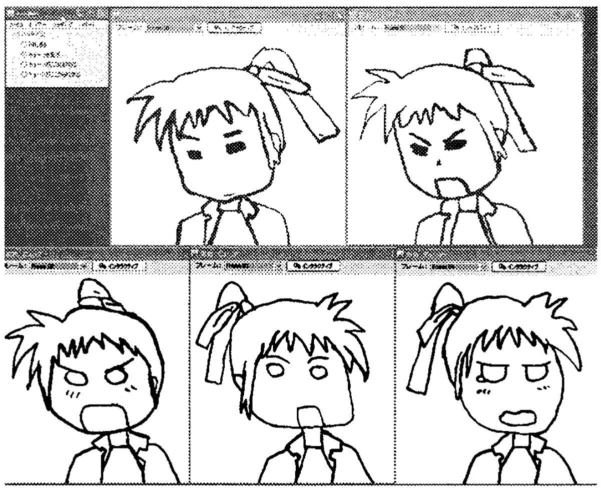

Figure 1: The user interface showing 5 frames

Initially, the user has only to click the "Interactive matching" button (The button in the upper part of the panels of figure 1). At each press, it picks up the first non-matched region, matches it and shows the result on screen. The matched regions appear colored with the same color.

For fixing matching mistakes the toolbox (the upper left dialog in the user interface) presents the following options:

1. Connect two regions: Requires 2 clicks for indicating the regions. They become matched and get the same color. Matchings in successive frames are recalculated.
2. Region disappears: Requires 1 click for indicating the disappearing region. Any match in successive frames becomes non-matched.
3. Region with new color: Requires 1 click for indicating the region. Assigns a new random color and looks for matches in successive frames.

4. Results and Discussion

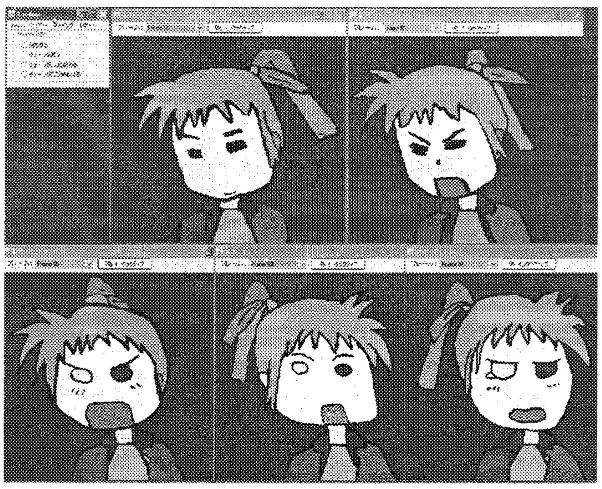

Figure 2: Correctly matched regions with random colors

We implemented our system in Java. We ran the tests on an Intel Core2 Quad CPU Q6700 @ 2.66 GHz, with 2 GB of RAM. After the segmentation and the extraction of features, all the interaction is in real-time. The animation consists of 5 frames, each of 507x446 pixels. Each frame has respectively: 15, 16, 17, 18 and 18 regions. In total, there are 84 regions.

Table 1: Results

Algorithm	Number of clicks	Wrongly colored regions
100% Manually	144 clicks	0
User-guided matching	38 clicks	0 (16 corrected by the user, 10 by the algorithm)
Non-interactive	0 clicks	43

We counted the number of mouse clicks on the panels (for connecting regions) and on the toolbox (choosing an option).

The 5 frames used as a test contain some difficult parts: the pony tail, the ear and the collar. That is because they change their shape, position and/or disappear.

Our non-interactive algorithm colored incorrectly 43 regions. Fixing that with our software would require 86 clicks (option 1). On the other hand, with the interactive matching, the user finished in just 38.

This shows how, for difficult cases, an approach that involves the user can be more efficient that a non-interactive approach.

5. Conclusions and Future Work

As seen in the results, fixing the coloring mistakes as soon as they appear has allowed us to stop their propagation and the generation other mistakes, thus reducing the total number of them and reducing the total effort needed until having a correctly colored cartoon.

As for future work, right now our comparison function's accuracy is low. We want to implement better ways of comparing the dominant points and neighbors. Regarding the user interface, sometimes small colored regions are difficult to see. It would be ideal to add visual indicators that help the user.

References

[1] J. Qiu, H. Seah, F. Tian, Z. Wu, and Q. Chen. Feature- and region-based auto painting for 2D animation. Visual Computer, 21:928-944, Oct 2005.

[2] J. Madeira, A. Stork, and M. Gross. An approach to computer-supported cartooning. Visual Computer, 12:1-17, 1996.

[3] J. Qiu, H. S. Seah, F. Tian,Q. Chen, Z. Wu, and M. Konstantin, Auto coloring with character registration. Proceedings of the International Conference on Game Research and Development (CyberGames '06), vol. 223, pp. 25–32, Perth, Australia, December 2006.