Based on Boundary Location Technology to Separate an Image Semi-automatically
Peng Ge-gang, Li Xin-yu, Song Ying, Xiang Li-sheng, Shen Qing
(Talkweb Information System Co.Ltd., Changsha, Hunan 410205 China)

Abstract—the key audiences of mobile animation are those high school students, working youth population, etc. Their pursuit of new fashion, fresh and willing hands makes them like to change style frequently to get a pleasurable and a satisfactory time. This paper presents a semi-automatic way to achieve image fragment and synthesis for the creation of mobile cartoon characters. Based on such a technical sketch, a cartoon server may provide a software support to help user satisfy their willing on DIY and create a plenty of animation characters, then transmitted them from mobile to mobile.

To deal with this task, the first and important step is locating a good boundary-line around the target area and separating it automatically, so as to get a sub-image. The second step is using a digital logical operation to synthesize the sub-image into a target image. At last, achieve a “grafting” effect. Moreover, apply pseudo-color on the synthesis’ to get a rather new fancy visual view.

Index Terms—Mobile Cartoon, Digital Image Processing

I. INTRODUCTION

Mobile phone animation products currently include MMS, four cell phones comics, animation clips, and turn-based mobile games, etc. They are different from TV cartoon audiovisual works in three salient features [1-2]: First, it does not require large contiguous time. This feature just to meet those younger people living in the modern city, fast pace of life, large pressure of work, wish to enrich cultural life but have no continuous time to deal with. Secondary, The key audiences of these products are those high school students, working youth population, etc. Their pursuit of fashion, fresh and willing hands makes them very keen on change them self and others’ style to make funny. Third, those high-end entertainment products such as mobile video and mobile game have highly requirements on mobile’s hardware performance, such as processor speed, memory size, color resolution, screen size, etc. and they are more expansive. On the other hand, MMS, four cell phones comics, animation clips, etc. are in lower hardware performance needs, of cause in lower cost, so it is prefer by most mobile users.

This paper provides a semi-automatic way to realize image fragment and synthesis to create mobile cartoon characters. Based on this sketch, a cartoon server may provide a group of tools to help younger to satisfy his/her willing and to create many cartoon characters[3]. It is clear that, this is a new way on mobile users’ own DIY to create a plenty of funny characters, and transmit them from here to there to heighten interesting on mobile cartoon.

To deal with this task, the first and important step is to realize a good boundary location. Then separate it automatically to get a sub-image, inside the boundary. And by add a digital logic operations merging the sub-image with another image and getting a "grafting" effect. Moreover, apply pseudo-color on the synthesis’ to get a rather new fancy visual view.

II. TECHNICAL COURSE

A. Implement Boundary Location and setup marks

A.1 prepare

(a) Establish a temporary file F to store the gray-value \( g(x, y) \) of each pixel of the original color image. The formula is:

\[ l = 0.3B + 0.59G + 0.11R \]

Here R, G, B is the red, green and blue components of the original color pixel \((x, y)\).
(b) Establish a mark file $M$ to store the mark $m(x, y)$ of each pixel for the original color image. And set every mark to "255" (as white) beforehand.

(c) Establish an array $A$ (the size is 256) for gray histogram to record the number of pixels that has the same gray value. For example, if there are 3564 pixels have a same gray value 134, so there is $A[134] = 3564$.

A.2. Calculate the best value as threshold $T_h$ to find out a boundary

(a) Count all pixels in array $A$ and named as Sum.

$$ \sum_{i=0}^{255} A_i \quad \text{here } A_i \text{ is the number of pixels that has same gray value } i.$$  

(b) With formula (1) to calculate the mean $M_{t0}$, $M_{t1}$ and weight $W_{t0}$, $W_{t1}$ for different range $0$-$t$ and $t+1$-$255$ (here variable $t$ may change from $1$ to $254$),

$$ M_{t0} = \frac{\sum_{i=0}^{t} A_i}{\sum A_i}, \quad M_{t1} = \frac{\sum_{i=t+1}^{255} A_i}{\sum A_i}, $$

$$ W_{t0} = \frac{\sum_{i=0}^{t} A_i}{\text{Sum}}, \quad W_{t1} = \frac{\sum_{i=t+1}^{255} A_i}{\text{Sum}}. \quad (1) $$

(c) By select different variable $t$ and compare the variance $\delta_t$ on different range $(0$-$t$ and $t+1$-$255)$ to get a maximal $\delta_t$. This maximal variance is the best value for threshold.

In this way, first, with formula (2) to calculate $\delta_t$

$$ \delta_t = W_{t0} W_{t1} (M_{t1} - M_{t0})^2 \quad (2) $$

then, with formula (3) to get a threshold $T_h$ when $\delta_t$ reach maximum.

$$ T_h = t, \text{ when } \nabla_t = \max(\delta_t), t = 1, 2, \ldots, 254 \quad (3) $$

Here, the threshold $T_h$ is a value calculated from program and user may modify this value to fit his own needs and get shorter or longer boundary-line.

A.3. Compare the gray value of every pixel with $T_h$ in entire image. When one of the gray values is smaller than $T_h$, the program changes the mark of relevant point in file $M$ to "0" (as black).

A.4. Scan the entire map again to find all boundary points. When program discovers a "black" spot is around with at least one "white" adjacent point, then this point is marked as a "boundary point". With similar to the threshold $T_h$, the "white" adjacent points is a modifiable parameter too.

A.5. Connect all boundary points to form a boundary-line

B. Manually assist to demarcate a border

When part of the boundary can not be set by program automatically or attempt to strip a body but it is not inside the border, in such case, manually assist is needed to demarcate a clear and closed border.

For example, if the background of an image is in light gray and the cheek is in light yellow, so it is difficult for program to set a valid boundary between the cheek and background. Another example may take a person wears a dark jacket, although there are good boundary between the neck and coat, but if the needs for creation do not want include a necklace, in this case may require to locate a new boundary manually above the necklace and abandon the original closed border.

C. Separate out the target area (inside the border) automatically

There are two stages to finish a separate operation. The first stage is marking a square area just surround the target sub-image. It includes three steps as below:

Step1. Scan the entire image from the very beginning line of whole image. When meet a line has boundary point firstly, this line is marked as a TOP line.
of the cut-off sub-image. Scan line by line again till the BOTTOM line of the sub-image, which means no longer a boundary point below this line.

Step2. Scan a line from left to right. When meet a border point on any column firstly, records it as a left border point (briefly named as LBP) of this line. Scan columns one by one again to find the right boundary point (briefly RBP).

Step3. Turn to next line and do the same operate as step2 does, to find out all the LBPs and RBPs of these lines.

The second stage is used to cut-off the target sub-image. It includes three steps as below:

Step1. Set all lines, which are beyond of the boundary-line, before the top and behind the bottom line, to null (as "white").

Step2. Set all columns which are beyond of the boundary-line (left of LBP and right of RBP) to null (as "white").

\[
    h(x, y) = \begin{cases} 
    a(x, y) = \text{null}, & b(x, y) = \text{null} \\
    a(x, y) \neq \text{null}, & b(x, y) = \text{null} \\
    a(x, y) = \text{null}, & b(x, y) \neq \text{null} \\
    a(x, y) \neq \text{null}, & b(x, y) \neq \text{null}
    \end{cases}
\]

E. Pseudo-color rendering

This is an additional stage of this scheme. It is based on region growing algorithm. The goal is make the merged image a rather new fancy visual view.

Step1: Select a point inside the image as a seed. Usually a seed is one of the points inside the area which wants to be overspread to a larger area from the seed.

Step2: Set a stack and push the seed into the stack.

Step3: Check the gray value of the 8 neighbors of the seed. If the gray difference (between one of the neighbors with the seed) less than the pre-setting threshold, mark this neighbor as a new growing point and turn to step4. Otherwise, it is consider there is no new growing point here and turn step4 too. The range of pre-setting threshold is 0~255 and it may adjust by user in the program.

Step4: If the stack is empty turn to step5, otherwise pop out a new point as a new seed and turn step3.

Step5: Do render to the just grown area. The selected colorized color may chose from color table by user in this program.

III CONCLUSIONS

This scheme is designed to make the image separation and synthesis (similar to Magic Wand in Adobe PhotoShop) in lesser manual intervention and more arbitrary. We regret to say that no human intervention in image separation is unlikely to be done. Because the computer is difficult to predict the separation object of the operator’s mind. And in some shooting conditions obtained images, it is difficult to fully automatic separation.
In addition, although we can not say which kind of border positioning technology is used by PhotoShop's Magic Wand, but we can happily report: As in our program, a user can modify the threshold $T_h$ and the number of "neighbors", so that the boundary location was more controllable, see Figure 1, it is somewhat better than PhotoShop's Magic Wand. And the results of sub-image cut off and pseudo-color rendering shown in Figure 2, 3.

If you have any question or need the source code you are welcome to contact us with E-mail.

The inadequacies of the paper want to be criticized and corrected.

IV ACKNOWLEDGMENT

We would like to thank deeply to the National Technical Supporting Project Foundation of China and the Ministry of Culture of China for their support on this project

References

1. Peng Ge-gang, Li Xinyu, Song Ying, Xiang Lisheng, Shen Qing, A Facial Organs Positioning Method Based on Grayness Mutation[] ICWN’10.

2. Li Xinyu, Song Ying, Xiang Lisheng, Shen Qing, A Facial Organs Positioning Method Based on Grayness Mutation[] ICWN’10.

3. Peng Ge-gang, Li Xinyu, Song Ying, Xiang Lisheng, Shen Qing, A Facial Organs Positioning Method Based on Grayness Mutation[] ICWN’10.

Authors

Peng Ge-gang: Peng is a post-doctor in computer application technology. He is a senior architect and interested in system architecture, mobile communication, etc. E-mail: penggegang@hotmail.com

Li Xinyu: Li is the general manager of Talkweb Information System Corp. He is a senior architect and received a BS degree in Computer and Information Engineering from Huaihai University, Nanjing in 1988. He is interested in system architecture, mobile communication, etc.

Song Ying: Song is the vice general manager of Talkweb Information System Corp. He received a BS degree in Chemistry from China University of Geosciences, Wuhan in 1989. He is interested in mobile communication, multimedia, etc.

Xiang Lisheng: Xiang is a vice technical supervisor in Talkweb Information System Corp. He received a BS degree in computer science from Hunan University in
Shen Qing: She received a BS degree in electronic instrument from Changchun Geological Institute. He was a professor in the Institute of Computer Science at National University of Defense Technology, China. Now he is the first chief technical consultant of Hunan Talkweb Information System Corp. He is interested in pattern recognition, AI, MM, animation, etc. E-mail: sq1950224@yahoo.com.cn