A Fast Image Matting Method based on Interval-line Sampling

Wang Hu
College of Software Engineering
South China University of Technology
Guangzhou 510006, China
billhhh@163.com

Yang Zhongming *
College of Computer Engineering Technical
Guangdong Polytechnic of Science and Technology
Zhuhai 519090, Guangdong
yzm8008@126.com

Ma Congbo
College of Software Engineering
South China University of Technology
Guangzhou 510006, China
201520121828@mail.scut.edu.cn

Wu Hongyue
College of Software Engineering
South China University of Technology
Guangzhou 510006, China
497563913@qq.com

Ouyang Mingting
Electronic Science and Technology
University of Electronic Science and Technology of China
Chengdu Sichuan 610054, China
carlosouyang@outlook.com

***Abstract*—Image matting techniques are a key step of image processing or video editing, which is widely used nowadays. So, it has a highly demanding on processing speed. Since it usually takes a long time to process images using existing image matting algorithms, especially in high-resolution images or videos, this paper proposes a state-of-art image matting speed-up method to solve the slow-processing problem of image matting based on interval-line sampling, and the results of experiments show that it can reduce the processing time considerably.**

Keywords-Image matting; Speed-up; High-resolution; Interval-line

I. INTRODUCTION

With the blooming of the image processing and movie industry, image matting is becoming more and more important and useful. Accurately extracting the foreground from an image is known as Alpha matte problem, which is to estimate the alpha value from a foreground and background combination equation

$$I_z = \alpha_z F_z + (1-\alpha_z) B_z \tag{1}$$

where F_z stands for the foreground color and B_z stands for the background color. The value of α falls between 0 and 1. Here are two special cases, if α equals to 1, it belongs to the foreground and if α equals to 0, it belongs to the background of the given image[1]. It is a highly ill-posed problem because of the attempt to find seven unknowns out of three equations, which represent three channels of the color space. Therefore, with an original image, usually trimap with extra information that roughly indicates the foreground/background and unknown pixels are required.

According to the study of Wang[2], Zhu[3] and other authors, existing matting techniques can be mainly scattered into three categories: 1) Sampling-based matting 2) Propagation-based matting 3) Learning-based matting. There is another matting method called blue screen matting method that places foreground into the purified background to eliminate the complex interference of the background and Learning-based matting techniques are relatively new[4-6], while most researches of the image matting area have been done on Sampling-based and Propagation-based matting methods.

In Sampling-based matting techniques, such as work did by Feng [7], a straightforward idea that sampling foreground and background pairs for all unknown pixels is often referred. Thus, how to figure out the relationships between alpha value and samples or to find the best sample pairs for unknown pixels is the main problem of these methods.

Based on the assumption that images are relatively smooth is Propagation-based matting techniques. These methods mainly focus on the continuity between pixels in a smaller area rather than the foreground and background colors to estimate the alpha value. To constrain the matting equation, Propagation -based matting techniques use affinities of nearby pixels as its basic idea. The work has been done by Chen[8] in KNN matting algorithm and Xiao[9] in closed-form matting and other authors illustrated the point clearly.

Corresponding author: yzm8008@126.com

Sampling-based matting techniques discussed above focus on the distances between known foreground and background sample pairs to an unknown pixel in color space, so these methods can be used to assign data costs to pixels through this trait. On the other hand, Propagation-based matting methods measure the relationships between neighboring pixels, therefore setting neighborhood costs can be used through this feature. Thus, there are several other techniques, such as [10-11], combine Sampling-based matting method and Propagation-based matting methods together, which could generate better results.

These algorithms perform well in the results. However, it is slow and takes a huge amount of time to calculate and estimate the final result, and even too slow to be used for video processing apparently. In order to avoid the time efficiency problem, this paper proposes a general technique that can be used on most of existing matting algorithms to dramatically reduce the computational complexity and time. Conventional wisdom may hold that a larger number of pixels in a picture, the more calculation will be, even though sometimes time efficiency is more emphasized rather than the quality of matting results. To achieve this goal, this paper defines a simple but efficient method to reduce the number of pixels for forming the sample set. The good result both in time efficiency and quality of matting results can be seen in the experiments.

II. PROPOSED METHOD

A. High-Resolution Problem

This subsection will demonstrate the problem caused by high resolution of given images which may lead to a huge amount of time consumption. Higher resolution means more pixels in a specific area of an image. Usually, an image with higher resolution carries more information because much more useful detail can describe the image more specifically, especially in a large screen where this feature can be seen clearly. Due to the high resolution of the image, it may cost a large amount of time to deal with these details, even though these details are not that important sometimes. For instance, in real-time video processing area, time efficiency always comes to the first priority, and to deal with a relatively lower resolution but less time could be more feasible. So, in this case, the high-resolution processing becomes a problem.

B. Sampling every few lines

How to solve the High-Resolution Problem? We propose a method to take care of the problem by sampling every few lines rather than using the whole picture in matting process. This method will be dramatically fewer the number of pixels to be processed and then lower the complexity of the calculation. The technique is given as follows:

$$\begin{cases} R_{obj} = \dfrac{R_{org}}{\omega} \\ C_{obj} = \dfrac{C_{org}}{\omega} \end{cases} \tag{2}$$

where R_{org} represents the row number of the original image matrix and C_{org} represents the column number of the original image matrix, while R_{obj} represents the row number of the object image matrix after processing and C_{obj} represents the column number of the object image matrix after processing, parameters ω stands for sampling every ω lines given by users. And then segment the image matrix of the original image into smaller submatrix as the example shows below:

$$I_{org} = \begin{pmatrix} \begin{pmatrix} I_{11} & \cdots & I_{1(n/\omega)} \\ \vdots & \ddots & \vdots \\ I_{(m/\omega)1} & \cdots & I_{(m/\omega)(n/\omega)} \end{pmatrix} & \cdots & \begin{pmatrix} I_{1(n(\omega-1)/\omega)} & \cdots & I_{1n} \\ \vdots & \ddots & \vdots \\ I_{(m/\omega)(n(\omega-1)/\omega)} & \cdots & I_{(m/\omega)n} \end{pmatrix} \\ \vdots & \ddots & \vdots \\ \begin{pmatrix} I_{(m(\omega-1)/\omega)1} & \cdots & I_{(m(\omega-1)/\omega)(n/\omega)} \\ \vdots & \ddots & \vdots \\ I_{m1} & \cdots & I_{m(n/\omega)} \end{pmatrix} & \cdots & \begin{pmatrix} I_{(m(\omega-1)/\omega)(n(\omega-1)/\omega)} & \cdots & I_{(m(\omega-1)/\omega)n} \\ \vdots & \ddots & \vdots \\ I_{m(n(\omega-1)/\omega)} & \cdots & I_{mn} \end{pmatrix} \end{pmatrix} \tag{3}$$

So, the new matrix of the object image can be formed by averaging all the elements of every submatrix into its elements.

$$I_{obj} = \begin{pmatrix} OI_{11} & \cdots & OI_{1n} \\ \vdots & \ddots & \vdots \\ OI_{m1} & \cdots & OI_{mn} \end{pmatrix} \tag{4}$$

$$OI_{mn} = \frac{I_{(m(\omega-1)/\omega)\,(n(\omega-1)/\omega)} + \cdots + I_{mn}}{(m / \omega)\,(n / \omega)} \tag{5}$$

Pseudo-code of interval sampling algorithm is given as below.

Algorithm 1: Downsize the original image matrix by averaging

Input: Original image matrix **IOrg** and the number of intervals **Interval**

Output: The result of downsized image matrix **DszImg**

```
Get the number of row, col, layer from IOrg
if mod(row, Interval) ≠ 0 then
    //add rows
    for i ←1 to Interval-mod(row, Interval) do
        Add (row+i)th row for IOrg
    end for
end if
if mod(col, Interval) ≠ 0 then
    //add columns
    for i ←1 to Interval-mod(col, Interval) do
        Add (col+i)th column for IOrg
    end for
end if
for i ←1 to row step Interval do
    for j ←1 to col step Interval do
        for k ←1 to layer do
```

```
                pixel_avr ← (IOrg (i,j,k)+...
                + IOrg (i+Interval-1,j+ Interval-1,k)) /( Interval^2)
                DszImg((i+1)/Interval,(j+1)/Interval,k) ←pixel_avr
            end for
        end for
    end for
```

C. Revert the object image

After the matting process, we will revert the object image matrix back to the same size of the original image matrix by sharing the same pixel value in a submatrix.

$$\begin{pmatrix} I_{(m(\omega-1)/\omega)\,(n(\omega-1)/\omega)} & \cdots & I_{(m(\omega-1)/\omega)n} \\ \vdots & \ddots & \vdots \\ I_{m(n(\omega-1)/\omega)} & \cdots & I_{mn} \end{pmatrix} = OI_{mn} \tag{6}$$

Pseudo-code of reverting process is given as below.

Algorithm 2: revert the object image matrix

Input: Object image matrix **ObjI,** Original image matrix **IOrg** and the number of intervals **Interval**

Output: The result of downsized image matrix **RvtImg**

```
Get the number of row, col, layer from ObjI
for i ←1 to row do
    for j ←1 to col do
        for k ←1 to layer do
            RvtImg(i*(Interval-1)+1,j*( Interval-1)+1,k) ←I(i,j,k);
            ...
            RvtImg(i* Interval,j*Interval,k) ←I(i,j,k);
        end for
    end for
end for
Get the number of Irow, Icol from IOrg
if mod(Irow, Interval) ≠ 0 then
    //delete rows
    for i ←1 to row*Interval-Irow do
        Delete (row*interval-i+1)th row of RvtImg
    end for
end if
if mod(Icol, Interval) ≠ 0 then
    // delete columns
    for i ←1 to col*Interval- Icol do
        Delete (col*interval-i+1)th column of RvtImg
    end for
end if
```

Next let us discuss the basic idea of this method.

D. The analysis of computational complexity

The equation (2) above shows the basic process of the method in which using a given ω to achieve the goal that lower the algorithm complexity by sampling every n lines. By using this technique, the pixels in a submatrix share a same value, and it leads to the reduction of dimensions in mathematics directly.

Sampling every ω lines both on rows and columns ensures the object image after the sampling process has the same scale as the original image. Thus, the scale of the number of pixels to be processed will be reduced considerably, which can lead to less time consumption, especially in high-resolution image matting. The analysis of computational complexity is given below.

If the original picture has N pixels, so the computational complexity is:

$$T_{org} = O(N)$$

But, since the pixels of the object image has been reduced to $\frac{N}{\omega^2}$, the computational complexity of the object image is:

$$T_{obj} = O(\frac{N}{\omega^2})$$

For instance, if the ω is 2, which means sampling every other line to form object image matrix. And the original image matrix can be segmented as below:

$$I_{org} = \begin{pmatrix} \begin{pmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{pmatrix}\begin{pmatrix} I_{13} & I_{14} \\ I_{23} & I_{24} \end{pmatrix} & \cdots & \begin{pmatrix} I_{1(n-1)} & I_{1n} \\ I_{2(n-2)} & I_{2n} \end{pmatrix} \\ \vdots & \ddots & \vdots \\ \begin{pmatrix} I_{(m-1)1} & I_{(m-1)2} \\ I_{m1} & I_{m2} \end{pmatrix}\begin{pmatrix} I_{(m-1)3} & I_{(m-1)4} \\ I_{m3} & I_{m4} \end{pmatrix} & \cdots & \begin{pmatrix} I_{(m-1)\,(n-1)} & I_{(m-1)n} \\ I_{m(n-1)} & I_{mn} \end{pmatrix} \end{pmatrix}$$

Therefore, the scale of the object image matrix becomes one fourth of the original ones. Through this method, we can reduce the computational complexity of original images so that the time cost can be reduced.

III. EXPERIMENTAL RESULTS

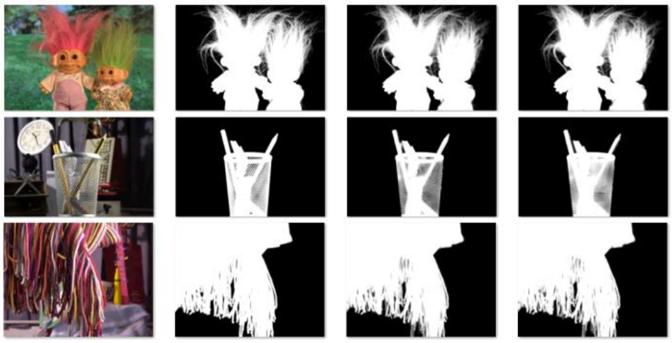
A. Object of experiments

The proposed method of this paper targets on time efficiency of matting algorithm. If the result reduced time considerably as the analysis of computational complexity given in Chapter 2, that means we hit the target; otherwise, it means we miss it.

B. Setup of experiments

We choose an existing matting algorithm[12] and add the proposed method on it to experiment. We can find if our method can work or not through comparing the results from the proposed method and original matting results. The test set is 27-picture training dataset on www.alphamatting.com. And the results of the proposed method (ω=2) and the original matting results both will be evaluated by time logs.

C. *Description of experimental results*



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

Figure 1. Example of proposed method. (a) original images, (b) ground truth images, (c) original matting results. (d) results from the proposed method.

Figure 1 shows the result of the comparison of the proposed method and original matting results. And we log the time of processing on each picture, the result shows as below:

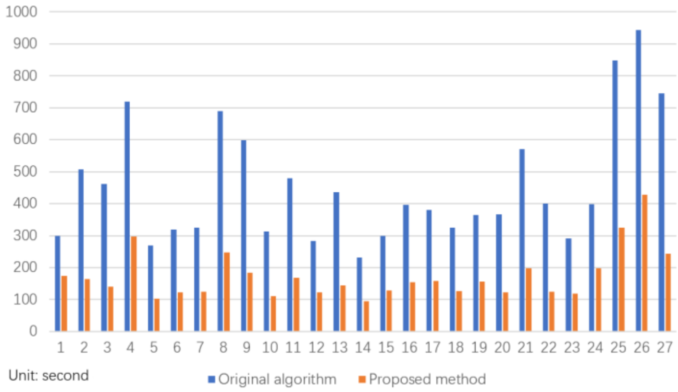


Figure 2. Time logs for each picture both in the original algorithm and the proposed method.

From the chart shows in Figure 2 we can see that the method proposed by this paper can reduce the processing time dramatically, even though may lose accuracy slightly of the image compare to the original image because of the sharing value in certain submatrix described in Chapter 2.

IV. Conclusion and Future Work

This paper presents a speed-up method for image matting based on sampling every few lines and sharing pixel value in a small region. This method performs well and can effectively reduce the processing time, especially in ultra-high-resolution images. And it can be widely used in the scenarios highly demanding on processing speed or even in real-time video processing.

Nevertheless, there are still many works can be done to improve the result in this method. This method can be further extended on the methods of interval-line other than average to have less accuracy loss or can be a Pre-Processing step for matting techniques to achieve better image matting results.

V. Acknowledgements

This work is supported by National Natural Science Foundation of China (61370102), Guangdong Province outstanding young teacher training plan (Yq2014187), Guangdong Natural Science Funds for Distinguished Young Scholar (2014A030306050), the Ministry of Education - China Mobile Research Funds (MCM20160206) and Guangdong High-level personnel of special support program(2014TQ01X664).

VI. References

[1] Chen, Xiaowu, Dongqing Zou, Steven Zhiying Zhou, Qinping Zhao, and Ping Tan. "Image matting with local and nonlocal smooth priors." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1902-1907. 2013.

[2] Wang, Jue, and Michael F. Cohen. "Image and video matting: a survey." Foundations and Trends® in Computer Graphics and Vision 3, no. 2 (2008): 97-175.

[3] Zhu, Qingsong, Ling Shao, Xuelong Li, and Lei Wang. "Targeting accurate object extraction from an image: A comprehensive study of natural image matting." IEEE transactions on neural networks and learning systems 26, no. 2 (2015): 185-207.

[4] Cho, Donghyeon, Yu-Wing Tai, and Inso Kweon. "Natural image matting using deep convolutional neural networks." In European Conference on Computer Vision, pp. 626-643. Springer International Publishing, 2016.

[5] Xu, Ning, Brian Price, Scott Cohen, and Thomas Huang. "Deep Image Matting." arXiv preprint arXiv:1703.03872 (2017).

[6] Hu, Hong, Liang Pang, and Zhongzhi Shi. "Image matting in the perception granular deep learning." Knowledge-Based Systems 102 (2016): 51-63.

[7] Feng, Xiaoxue, Xiaohui Liang, and Zili Zhang. "A Cluster Sampling Method for Image Matting via Sparse Coding." In European Conference on Computer Vision, pp. 204-219. Springer International Publishing, 2016.

[8] Chen, Qifeng, Dingzeyu Li, and Chi-Keung Tang. "KNN matting." IEEE transactions on pattern analysis and machine intelligence 35, no. 9 (2013): 2175-2188.

[9] Xiao, Chunxia, Meng Liu, Donglin Xiao, Zhao Dong, and Kwan-Liu Ma. "Fast closed-form matting using a hierarchical data structure." IEEE Transactions on Circuits and Systems for Video Technology 24, no. 1 (2014): 49-62.

[10] Jin, Meiguang, Byoung-Kwang Kim, and Woo-Jin Song. "Adaptive propagation-based color-sampling for alpha matting." IEEE Transactions on Circuits and Systems for Video Technology 24, no. 7 (2014): 1101-1110.

[11] Chen, Xiao, and Fazhi He. "A propagation matting method based on the Local Sampling and KNN Classification with adaptive feature space." arXiv preprint arXiv:1605.00732 (2016).

[12] Shahrian, Ehsan, Deepu Rajan, Brian Price, and Scott Cohen. "Improving image matting using comprehensive sampling sets." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 636-643. 2013.