Automatic Trimap Generation for Digital Image Matting

Chang-Lin Hsieh and Ming-Sui Lee
Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan, R.O.C.
E-mail: {r98922088, mslee}@csie.ntu.edu.tw Tel: +886-2-33664888

Abstract— **Digital image matting is one of the most popular topics in image processing in recent years. For most matting methods, trimap serves as one of the key inputs, and the accuracy of the trimap affects image matting result a lot. Most existing works did not pay much attention to acquiring a trimap; instead, they assumed that the trimap was given, meaning the matting process usually involved users' inputs. In this paper, an automatic trimap generation technique is proposed. First, the contour of the segmentation result is dilated to get an initial guess of the trimap followed by alpha estimation. Then, a smart brush with dynamic width is performed by analyzing the structure of the foreground object to generate another trimap. In other words, the brush size is enlarged if the object boundary contains fine details like hair, fur, etc. On the contrary, the brush size gets smaller if the contour of the object is just a simple curve or straight line. Moreover, by combining the trimap obtained in step one and downsampling the image, the uncertainty is defined as the blurred region, and the third trimap is formed. The final step is to combine these three trimaps together by voting. The experimental results show that the trimap generated by the proposed method effectively improves the matting result. Moreover, the enhancement of the accuracy of the trimap results in a reduction of regions to be processed, so that the matting procedure is accelerated.**

I. INTRODUCTION

Cutting a targeting object from an image and pasting it to another one is a major application in image/video processing. Image segmentation has been developed to achieve this goal; however, the result of segmentation is simply a binary image that separates the object and background. In that case, objects with hair, fur or semi-transparent part cannot be extracted out perfectly. Later, a technique called image matting was proposed to solve this problem. Instead of a binary image, the map generated by matting is a gray-level image with values between 0 and 1. Each pixel color in the image can be synthesized by the following equation:

$$\mathrm{C} = \alpha\mathrm{F} + (1-\alpha)B \qquad (1)$$

where C represents the pixel color, F and B are foreground and background colors, and α is the foreground opacity.

There are many algorithms proposed for image matting. Most of them adopt a trimap to be the system constraint. Trimap segments the image into three regions: foreground, background, and unknown. According to an appropriate trimap, the alpha value can be estimated more accurately. A good trimap should narrow down the unknown region keeping the known region as large as possible. J. Wang and M. F. Cohen also said that "One of the important factors affecting the performance of a matting algorithm is how accurate the trimap is" [1]. Table 1 and Fig. 1 are part of image matting evaluation results from the website of alpha matting evaluation which is constructed by [5]. Three testing images are considered and three different trimaps are utilized in four different methods. As we can see from the table, the SAD (Sum of Absolute Difference) of the large trimap is almost twice that of the small trimap. In other words, trimap indeed plays a major role for image matting. Although the trimap accuracy is very important, most existing algorithms use dilation to obtain it while others require users to input a precise trimap. In this paper, an automatic trimap generation method is proposed to replace the dilation part in the traditional matting process. With the trimap acquired from the proposed algorithm, the image matting achieves better output quality and the computational complexity is lowered to a certain level without requiring users' involvement.

The rest of this paper is organized as follows. The related work is presented in Section 2. The proposed trimap generation algorithm is described in Section 3. Experimental results are given in Section 4 to demonstrate that the proposed trimap achieves good matting results. Finally, concluding remarks are drawn in Section 5.

II. RELATED WORK

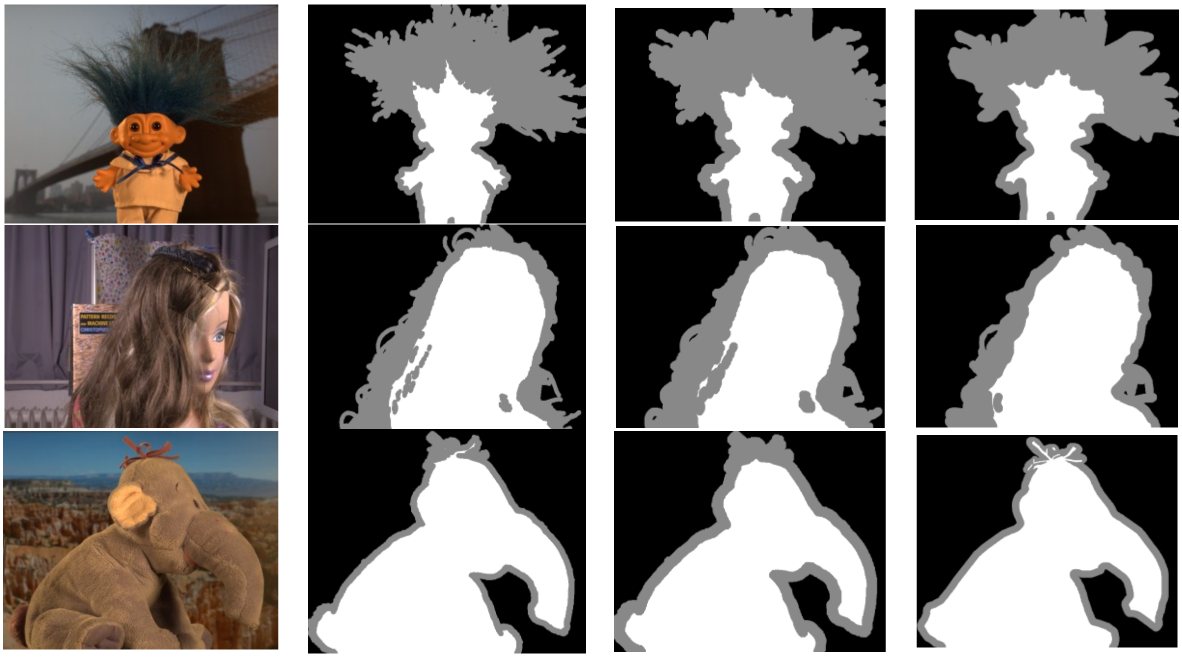
In this section, two major algorithms, Soft Scissors [6] and shared matting [2], referenced in the proposed methods are introduced.

A. *Soft Scissors*

Soft Scissors is an interactive real-time matting algorithm. Every time the user draws a stroke on the object boundary, the system computes an updated region and solves the matting equation in the update region. Because the updated region is small, the system can finish all computations in real-time. One of its features is that it could change the brush width dynamically by predicting the alpha values in front of painting track. We will use this feature in Sec. III.D. to help us generate a trimap.

Table 1. Image matting evaluation results

Sum of Absolute Difference	Troll			Doll			Elephant		
	Small	Large	User	Small	Large	User	Small	Large	User
Shared Matting [2]	10.8	20.5	15.0	7.8	11.6	8.1	2.1	5.8	2.9
Segmentation-based Matting [3]	12.8	23.5	16.6	6.6	8.3	7.3	2.1	3.9	3.1
Improved Color Matting [4]	14.9	24.5	20.0	6.7	9.5	8.5	2.6	5.4	3.4
Shared Matting (real-time) [2]	12.4	21.6	16.3	9.5	13.5	9.9	2.5	6.8	3.2



(a) Image (b) Small Trimap (c) Large Trimap (d) User's Trimap

Fig. 1. Images and trimaps used to evaluate image matting result

B. *Shared Matting*

Shared matting is a nearly real-time matting algorithm. One of the reasons it is fast is because the algorithm designed can be parallelized with the GPGPU to compute the unknown pixels. Another reason is they observe that there are lots of redundancies in estimating alpha values. We use shared matting in Sec. III.C. to help us estimate the alpha values and analyze the structure of foreground object.

III. AUTOMATIC TRIMAP GENERATION

In this section, the system overview of the proposed method as shown in Fig. 2 is introduced, and each part of the algorithm is described in detail. It includes how to keep small structure by dilating uncertainty, and how to combine the final trimap using voting.

A. *System Overview*

The proposed method contains five main parts which are Dilation, Alpha Estimation, Dynamic Width, Dilate Uncertainty, and Voting. The inputs of this method are image itself and its segmentation result. The segmentation result can be generated by any algorithm. Here we use the GrabCut [7] to perform the segmentation.

First, we use Dilation method to get an initial guess of the trimap, naming it T_d. It is then passed to the Alpha Estimation to estimate alpha values α_d and confidence values f_d which are used in Dynamic Width, Dilate Uncertainty, and Voting steps.

In the Dynamic Width step, we analyze α_d and f_d which help us figure out the image structure and use the proposed dynamic brush width technique modified from Soft Scissors to generate the second trimap named dynamic width trimap T_w. Meanwhile, the Dilate Uncertainty step also adopts α_d and f_d to obtain the third trimap named dilate uncertainty trimap T_u; it is able to capture small structures that may be missed out in Dynamic Width step.

After acquiring T_w and T_u, we pass them to the Alpha Estimation part to get their alpha values and confidence values. With these three trimaps: T_d, T_w, and T_u, Voting step will combine them to form a final trimap T_f by considering their alpha values and confidence values.

B. *Dilation*

This step simply uses a dilation method to form an initial guess of the trimap. We first set all pixels along the segmentation contour as unknown, then we dilate each unknown pixels $k_{dilation}$ times to form the trimap T_d, Fig. 3(a) shows an example of this step.

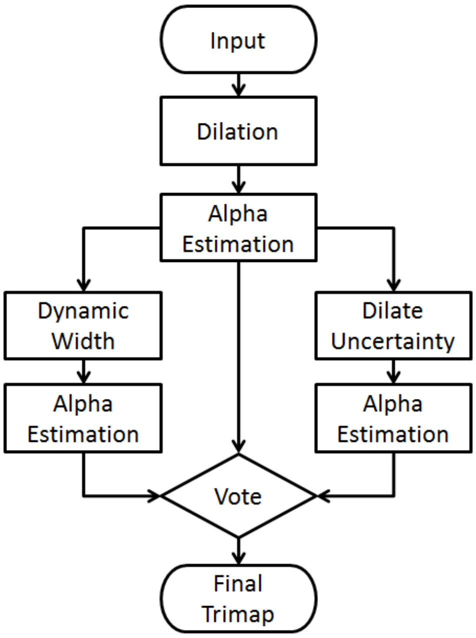


Fig. 2. System flowchart

C. Alpha Estimation

In order to analyze the foreground object structure, we use the initial trimap T_d and the input image to perform image matting. Here we use the shared matting to perform alpha estimation which has been introduced in the related work. To simulate the effect of sparse sampling in dynamic brush width of Soft Scissors and speedup the matting process, we choose to down sample both the image and the trimap k_{sparse} times ($k_{sparse} = 4$ in this paper). The parameters recommended by the shared matting should be adjusted since it is a down-sampled version. The value k_i which is used to control the search range in known region expansion part is changed to k_i/k_{sparse}, as the original k_i is obviously too large. Also, the value k_g which is adopted to define how many search directions in sample gathering part to get more information is doubled. After the matting process is complete, we have two maps available, the alpha matte and the confidence map. Afterwards, bilinear interpolation is performed to up-sample these two maps to get α_d and f_d which will be used in Dynamic Width and Dilate Uncertainty part. After these two parts, this procedure is repeated twice using the trimaps T_w and T_u to get the alpha matte α_w and α_u and the confidence map f_w and f_u, respectively.

D. Dynamic Brush Size

In this step, we modify the dynamic brush width method from the Soft Scissors to generate a content adaptive trimap. In the Soft Scissors, the brush width is adjusted according to 1D distribution to analyze the alpha values. Here we use 2D distribution to analyze the alpha values so we can control the shape and the size of the brush dynamically.

Different from Soft Scissors analyzing and predicting the brush width along the user painting direction, the proposed method does not require users' input; instead, the segment of the foreground object is considered. Along the contours of the segment, we perform this function sparsely at a distance k_{sparse}, and analyze nearby information in a radius $k_{dilation}$. This is because in our initial guess trimap, the pixels out of this range are certainly foreground or background. We grid sample pixels in this area, the distance between two samples is also k_{sparse} and following the directions of segment's contour's tangent and normal as shown in Fig. 3(a). With these samples, their alpha values and confidence values are combined to form a weighting function as:

$$w = c \cdot (0.5 - |\alpha - 0.5|) \tag{2}$$

where c is the confidence value, and α is the alpha value for each sample. Noticed that the weighting value becomes larger as c increases and α gets close to 0.5 since this kind of pixels are what we would like to process in image matting.

Next, we analyze these weights' distribution and use a Gaussian distribution to draw an ellipse to cover as many pixels with large weighting values as possible. The data center $(\bar{x}, \bar{y})$ is first computed by a weighted average:

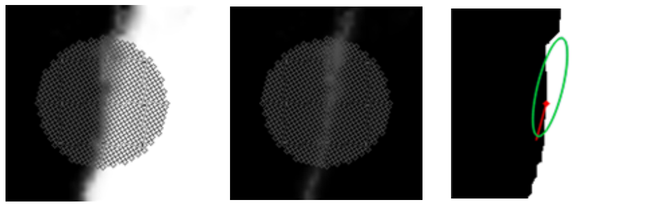
$$\bar{x} = \left(\sum w_p \cdot x_p\right) / \sum w_p; \quad \bar{y} = \left(\sum w_p \cdot y_p\right) / \sum w_p \tag{3}$$

where x_p and y_p are the distance between sample p and the pixel sampled on the segment contour. Then we compute the variance to be a reference of the ellipse brush's two axes:

$$a = k_{scalar} \cdot \sqrt{\left(\sum w_p \cdot (x_p - \bar{x})^2\right) / \sum w_p}$$

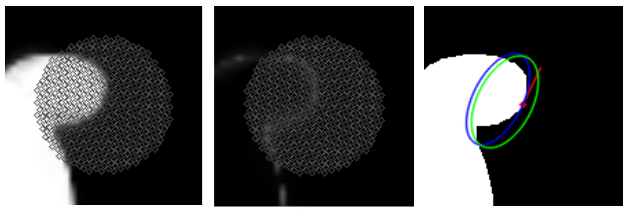
$$b = k_{scalar} \cdot \sqrt{\left(\sum w_p \cdot (y_p - \bar{y})^2\right) / \sum w_p} \tag{4}$$

where $k_{scalar} = 1.5$ is a scalar that makes sure we can cover all important pixels. Finally we draw an ellipse using $\bar{x}$, $\bar{y}$, a, and b as shown in Fig. 3(c).



(a) (b) (c)

Fig. 3. An example of the Dynamic brush width. (a) sampling on alpha values, (b) sampling on weighting values (c) the final computed brush.



(a) (b) (c)

Fig. 4. An example of the brush refinement. (a) sampling on alpha values, (b) sampling on weighting values (c) the final computed brush.

Sometimes, if the unknown region in T_d is too wide, matting miscalculation may happen, which means the ellipse brush may deviate from the segment contour. In such situation, the location of the brush ellipse should be refined to be closer to the segment contour. As an example shown in Fig. 4(c), the original brush ellipse is too far away from the segment contour and the size is not appropriate. Let the displacement represents the distance between the data center $(\bar{x}, \bar{y})$ and the segment contour. The goal is to make sure that the ellipse can cover the segment contour so we check if the displacement is bigger than the axis length. Moreover, it is better to avoid having the ellipse right on the contour edge. If the displacement is too large, both the displacement and axis length are modified by decreasing the displacement and increasing the axis length to make a bigger ellipse to cover the desired region.

E. Dilate Uncertainty

In some images, the foreground object may contain small structures like hair, fur, or something small and thin. Because we down-sampled the image and the trimap, the PSF (point spread function) will cause these structures to be blurred. These blurred structures should have alpha values between 0 and 1 after matting and we call these pixels uncertainty. To capture these uncertainties, we parse whole alpha matte α_d and set all pixels with alpha value between 0 and 1 to unknown and then dilate these pixel $k_{uncertainty}$ times (We use $k_{uncertainty} = 5$ in this paper). This process also helps us keep as much foreground and background pixels as possible. This step will generate another trimap T_u which should cover all small structures found by matting. Fig. 5(c) shows an example of this step.

F. Voting

Now we have three trimaps available: the dilation trimap T_d which covers all possible foreground object, the dynamic width trimap T_w which uses Gaussian distribution to cover most of the area of interest, and the dilate uncertainty trimap T_u which covers all captured small structures. The next step is to combine these three trimaps to form a final trimap T_f. For each pixel, if all three trimaps agree it belongs to unknown area then it is set to be unknown. If only one or two trimaps agree with it, we will check their alpha and confidence values. If the confidence value is high enough ($c > 0.9$ in practice) and the alpha value is not exactly 0 or 1, pass this pixel to matting to recalculate so it is set to be unknown. Other pixels are set to be either foreground or background based on the segmentation result. Fig. 5(f) shows a final trimap.

IV. Experimental Results

We show the experimental results of the proposed algorithm and compare it with the trimap generated by dilation. Because there is no standard to evaluate a trimap, we perform the shared matting to the trimap and compare the matting results. The images used in the experiments are from the website of alpha matting evaluation since it has ground truth in its database. As shown in Fig. 6, nineteen images with various content complexities are tested as they have different challenges. The sum of absolute difference (SAD) are computed between the matting result and ground truth for objective comparison. Fig. 7 shows the SAD values of all test images. As we can see, the matting result with the proposed trimap outperforms that with dilation trimap except image #15. The experimental results of image #15 are shown in Fig. 8. Even though most details of the hair are captured by the proposed trimap, some hair tips are missing in the final matting result. This is because image #15 contains large area of long hairs which makes the proposed dynamic brush difficult to adjust the brush width accordingly.

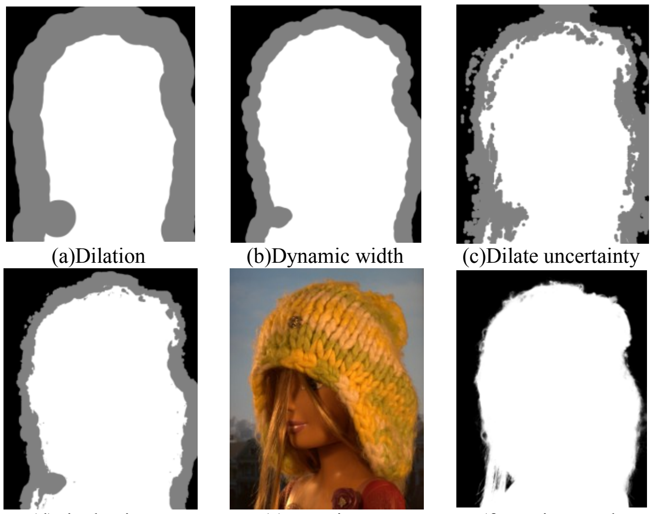


Fig. 5. An example of generated trimap using proposed method.

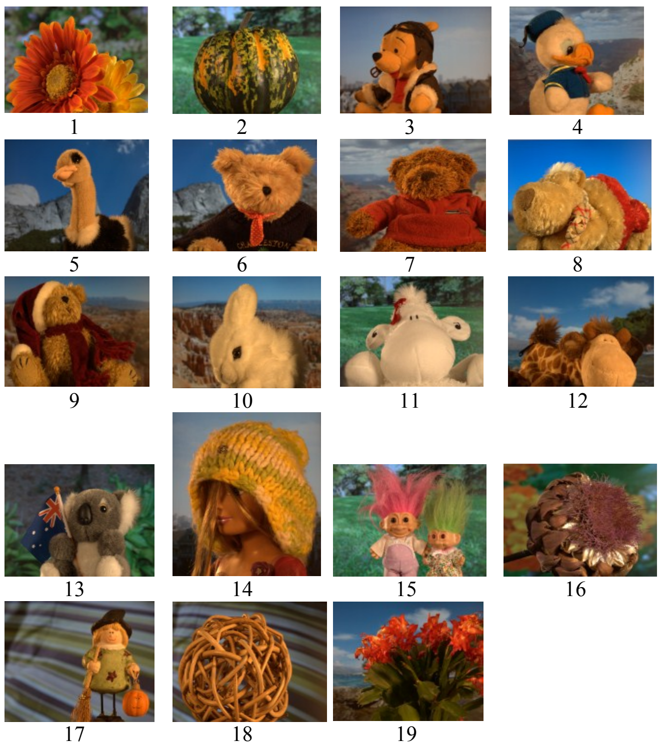


Fig. 6. Nineteen test images.

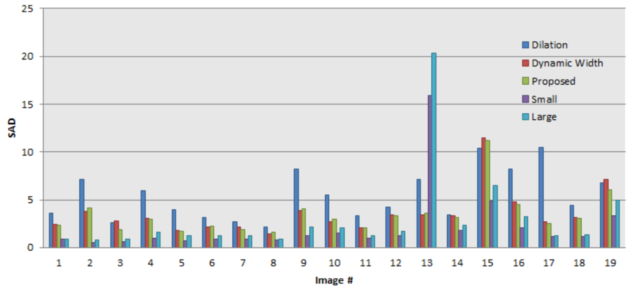


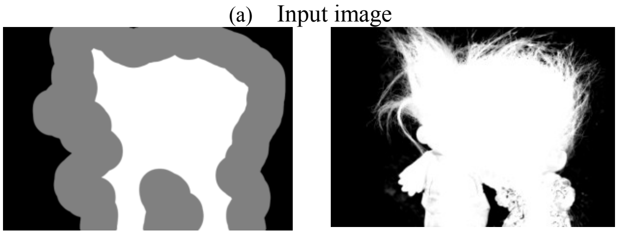
Fig. 7. The SAD values of nineteen test images.

Fig. 8 is another challenging case where the object contains lots of holes standing in front of a complex background. Compare to the trimap generated by dilation, the matting result with the proposed trimap can perfectly recover the doll's legs and the broom.

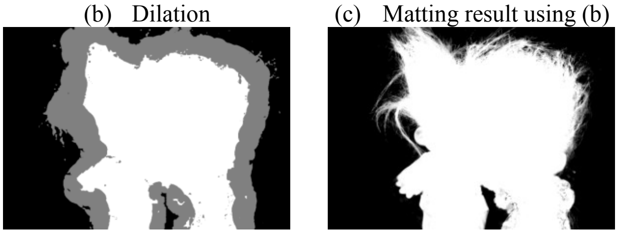
V. CONCLUSIONS AND DISCUSSIONS

In this paper, instead of improving the matting algorithm directly, we choose to find a new direction to improve matting quality by generating a better trimap automatically. Experimental results demonstrate that the trimap generated by the proposed method is much better than the dilation result. Moreover, no user's involvement is required. Another advantage of this work is it provides flexibility to make it applicable to any trimap-based matting algorithms. The system can be determined as an automatic or a user interactive one according to the chosen segmentation method. However, the proposed method still has limitation since the segmentation result is needed to obtain the trimap. As can be seen from Fig. 10, if the target foreground object contains large semi-transparent area, the proposed method may not get a good trimap.

(a) Input image



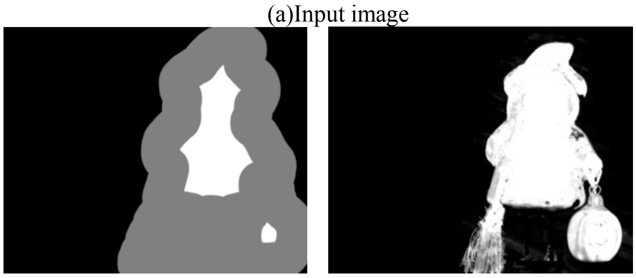
(b) Dilation (c) Matting result using (b)



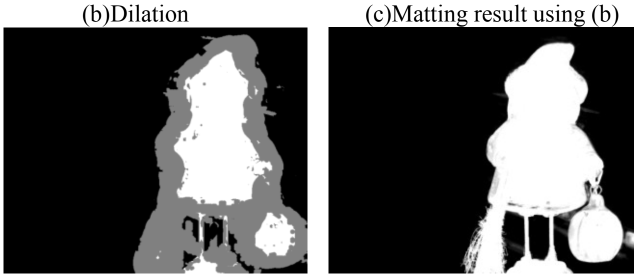
(d) proposed method (e) Matting result using (d)

Fig. 8. Experimental results of image #15.

(a)Input image



(b)Dilation (c)Matting result using (b)



(d)proposed method (e)Matting result using (d)

Fig. 9. Experimental results of image #17.

(a)input image (b) Trimap (c) Matting result

Fig. 10. An example of a failed case.

REFERENCES

[1] J. Wang and M. Cohen, "Image and video matting : A survey," *Found. Trends. Comput. Graph. Vis.*, 2007.

[2] E. S. L. Gastal and M. M. Oliveira, "Shared sampling for real-time alpha matting," *Eurographics*, 2010.

[3] P. Kohli C. Rhemann, C. Rother and M. Gelautz, "A spatially varying psf-based prior for alpha matting," *CVPR*, 2010.

[4] C. Rother C. Rhemann and M. Gelautz, "Improving color modeling for alpha matting," *BMVC*, 2008.

[5] J. Wang M. Gelautz P. Kohli C. Rhemann, C. Rother and P. Rott, "A perceptually motivated online benchmark for image matting," *CVPR*, 2009.

[6] M. Agrawala J. Wang and M. Cohen, "Soft scissors : An interactive tool for realtime high quality matting," *SIGGRAPH*, 2007.

[7] V. Kolmogorov C. Rother and A. Blake, ""grabcut" interactive foreground extraction using iterated graph cuts," *SIGGRAPH*, 2004.