A NEW APPROACH TO AUTOMATIC CLOTHING MATTING FROM MANNEQUINS

Bin Yuan[1], *Zongqing Lu*[1,*], *Jing-Hao Xue*[2], *Qingmin Liao*[1]

[1]Graduate School at Shenzhen, Tsinghua University, China
[2]Department of Statistical Science, University College London, UK
luzq@sz.tsinghua.edu.cn

ABSTRACT

It is crucial to extract retail clothes from images of mannequins when building a database of clothing images for virtual try-on systems. However, clothes often have complex texture and translucent material, such as holes and laces. It is thus difficult to extract clothes as foreground by existing generic natural image matting methods. Hence in this paper, we present a novel approach to automatic clothing matting from mannequins, with auxiliary information from a rough background image of the mannequin only. Experiments show that we can achieve remarkable improvement on the alpha matte near challenging regions of complex texture and translucent material of clothes. Moreover, our approach can automatically generate trimaps to facilitate the development and evaluation of other image matting algorithms.

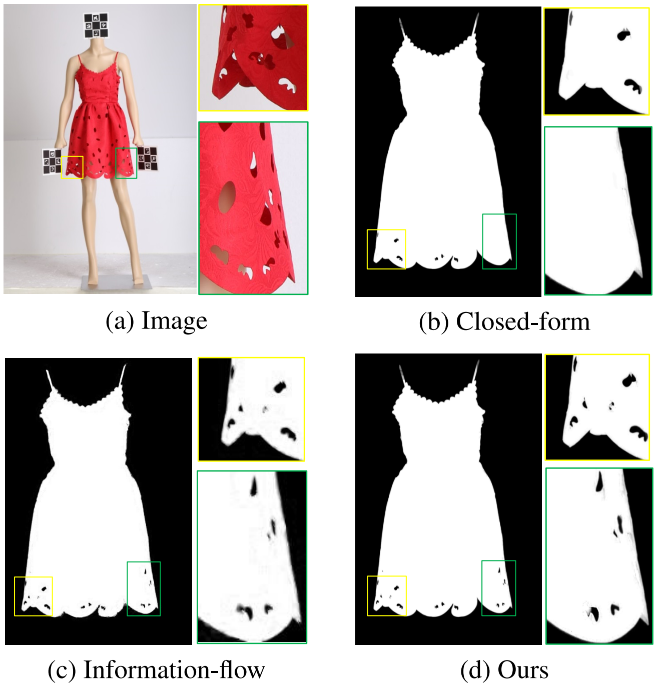
***Index Terms*—** Clothing image matting, Virtual clothing try-on

1. INTRODUCTION

Virtual clothing try-on systems [1] are now in more use by a growing number of prominent retailers in their clothing retail entity shops. A virtual fitting room enables shoppers to try on clothes for checking one or more of size, fit and style, virtually rather than physically. To construct a virtual clothing try-on system, a database with a large number of clothing images is required. In order to build a quality database of clothing images which can make the virtual try-on look realistic, clothes are preferred to be first worn and photographed on mannequins and then automatically extracted from the images. Hence, automatic extraction of clothes as foreground from the images of clothes on mannequins is an important task to build a quality database for the success of a virtual clothing try-on system.

Extracting the opacity information of foreground objects from a natural image is known as natural image matting. Mathematically, image matting assumes that the value of a pixel in the transition regions from foreground to background is a convex combination of its underlying foreground and background values:

$$I_i = \alpha_i F_i + (1 - \alpha_i)\, B_i, \tag{1}$$


(a) Image

(b) Closed-form

(c) Information-flow

(d) Ours

Fig. 1. Comparison of the closed-form matting [2], the information-flow matting [3] and our algorithm for clothing matting.

where α_i represents the pixel's foreground opacity which can take any value in [0, 1], and F_i and B_i are foreground and background pixel values, respectively. If $\alpha_i = 1$ or 0, then the pixel at location i belongs to definite foreground or definite background, respectively. Otherwise the pixel is called a mixed pixel. Since neither the foreground color F_i and the background color B_i nor the opacity α_i is known, estimating the opacity value is an ill-posed problem.

Image matting algorithms can be categorized into two basic groups, sampling-based methods and affinity-based methods, as well as their combinations. The sampling-based methods sample foreground and background colors from some definitely known regions to estimate alpha mattes within the unknown region [4–7]. The affinity-based methods assume local/nonlocal smoothness of alpha values and propagate the alpha values from the known regions to the unknown regions [2,8–11]. However, to generate the alpha matte from the input image, most of these matting algorithms require predetermined trimap [4] or scribbles [12], which is not practicable when our aim is to build a database with a huge number of clothing images.

In the clothing image matting scenario, the foreground object (i.e. clothes) often has complex textures and translucent material, such as laces and holes, which is largely affected by background (see Fig. 1(a)). This makes the matting even more severely ill-posed than usual, on which the existing generic-purpose natural image matting algorithms cannot work well for the challenging regions (Fig. 1(b)). Fortunately, the background in images of clothes on mannequins is relatively simpler than most outdoor natural images, and often rough background images with mannequins only can be readily obtained under the same scenario as that of the input images.

Therefore in this paper, to facilitate building a database of realistic clothing images, we propose a novel automatic image matting approach, which will avoid the necessity of manually providing a pre-determined trimap or scribbles, by exploiting rough background images as auxiliary information input. Such auxiliary information will render our approach a better image matting solver while addressing the challenging regions of holes and other translucent material on clothes (see Fig. 1(c) for the case of holes).

The main contributions of this paper are twofold:

- Firstly, our system can improve the quality of mattes in the regions with complex textures and translucent material such as laces and holes in clothing matting.
- Secondly, as a by-product of exploiting rough but auxiliary background information, we can automatically generate trimaps, which also in turn facilitate the development and evaluation of various trimap-based image matting algorithms for clothing matting.

The rest of the paper is organized as follows. In Section 2 various relevant image matting algorithms are discussed along with their strengths and limitations. Section 3 describes our proposed algorithm. Section 4 gives the experimental results and analysis of our approach and finally Section 5 concludes the paper.

2. RELATED WORK

Natural image matting methods in the literature can be roughly categorized as sampling-based methods, affinity-based methods, or their combinations. In this section, we briefly review the methods that are most relevant to our work.

Sampling-based image matting methods [4–6] propose strategies to gather samples from the background and foreground regions defined by the trimap, and select the best fitting background-foreground pair according to certain criteria for representing an observed mixed pixel as a mixture of foreground and background. Chuang et al. proposed Bayesian matting [4], which formulates the problem in a well-defined Bayesian framework and solves it using the MAP estimation. The method of shared matting [5] collects samples by shooting rays in different directions: the samples selected are the nearest foreground and background pixels on every ray, and each unknown pixel shares samples with neighboring pixels. Global sampling [6] uses all samples available in the image; in order to handle the ambiguity, the global sampling method defines a simple but effective cost function for sample selection.

Affinity-based image matting methods [2, 3, 8, 9] mainly make use of some pixel similarity metrics dependent on the color similarity or the spatial proximity, and propagate the alpha values from regions with known opacity. The Poisson matting algorithm [8] observes that if the foreground and background colors are locally smooth, the gradient of the matte can be estimated from the gradient of the image. The random Walk matting method [9] defines the affinity matrix via a Gaussian function. The closed-form matting approach [2] assumes that there is a smooth transition between foreground and background layers, and solve the matting problem by identifying these transitions.

Most of these state-of-the-art image matting algorithms require predetermined trimaps [4] or scribbles [12]. However, this is not practicable for building a database with a huge number of clothing images. Our approach is thus distinct from them in the sense that we do not need to input trimaps or scribbles; instead we input a rough background image, which is ready to obtain in our scenario of building virtual clothing try-on systems.

Technically, as with the closed-form matting method, our approach can be regarded as an affinity-based method. However, besides the difference in input, our approach differs from the closed-form matting in that our pixel affinity is not the pixel similarity metric that relies on the color similarity or the spatial proximity, as we shall discuss further in Section 3. To summarize, we propose a novel automatic affinity-based matting algorithm to generates high-quality alpha mattes with rough background images given as auxiliary information.

3. OUR METHOD

The framework of our proposed method is illustrated in Fig. 2.

First, we use the input image and a rough background image to generate the trimap. Then we divide the image into small patches in case of prohibitively high memory consumption when it comes to solve high-resolution image matting problem. Finally we solve the alpha mattes of every patch by utilizing information from the image, and merge these results.

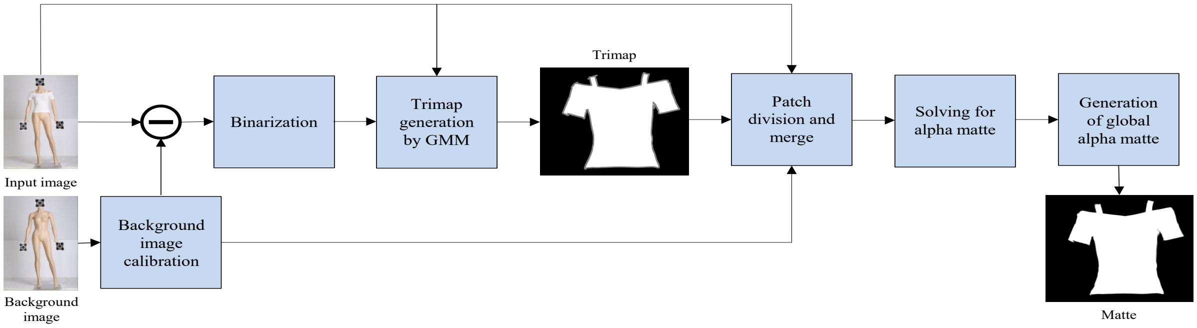


Fig. 2. Framework of our proposed image matting approach. To generate a trimap, we calibrate the background image, binarize the difference image between the input image and the rough background image to initialize the mask, and generate the trimap by using Gaussian mixture models (Section 3.1). Then, we divide the image into appropriate patches to reduce memory consumption in computation (Section 3.2). Finally, we solve for the alpha matte (Section 3.3).

3.1. Trimap generation

As mentioned before, in our scenario a rough background image with mannequin only can be obtained as auxiliary information. We can use this rough background image and the actual input image of clothes on the mannequin to automatically generate a trimap. However, this background image is not the true "background" because the background image is taken at another time with mannequin only. There can be some differences between these two images: for example, the movement of the mannequin, the camera position, and the reflection of light produced by the clothing on the mannequin. These differences will make it impractical to take the rough background image directly as the background of the original image.

Therefore, we develop the following algorithm to generate a trimap: First, we calibrate the background image to match it with the input original image. Then, we binarize the difference image between these two matched images, by using Otsu's thresholding method [13] to generate a mask of rough area of clothing. This mask is used then for image segmentation algorithm GrabCut [14] to establish a Gaussian mixture model (GMM). The segmentation map is finally eroded and dilated to attain a trimap.

3.2. Patch division and merge

Most existing image matting algorithms employ global optimization over the whole set of image pixels to recover alpha matte. However, clothing images are usually of high resolution (e.g. 4480×6720), which results in a prohibitively high memory consumption. Inspired by [15], we propose a patch division and merge strategy for developing the matting algorithm at the patch level, so as to reduce memory consumption of the clothing image matting algorithm. Our strategy is described as follows.

First, according to the distribution of unknown pixels in the trimap, we divide the image into small overlapped patches, with overlap between patches being considered to maintain consistency over the whole image.

To develop a patch-level matting algorithm, we need each match to contain enough information of known foreground or background. However, some patches may be unfortunately short of such information. For instance, it is possible that all pixels in a patch are within the unknown area in the trimap. In order to resolve this problem, we propose to merge several adjacent patches which have enough known information to the fully unknown patch, and then find an appropriate direction to re-divide the merged region to ensure the new patch size being invariant to the old one.

Through division and merge, we are able to generate patches which contain enough information of known foreground or background pixels. The pseudo-code of our patch division and merge strategy is listed in Algorithm 1.

3.3. Solving for the alpha matte with background given

3.3.1. The color line assumption

As mentioned earlier, we assume that background B is roughly known in this work. The matting problem, however, is still massively ill-posed. Therefore, some assumptions on the nature of F are needed, one of which is the color line assumption for the foreground colors in a local neighborhood in many natural images [2, 16].

That is, we assume that within a small window Ω_j around each pixel j, the set of foreground colors is convex in the RGB space:

$$F_i = \beta_i F_1 + (1 - \beta_i) F_2, \tag{2}$$

where F_1 and F_2 are two end points of a line in the RGB space, and the foreground color F_i lies on this straight line with β_i representing relative similarity of F_1 to F_i (or equivalently the proportional distance from F_2 to F_i). The color

Algorithm 1 Patch division and merge

```
Inputs: image: $\Gamma$; patch size: $n$; overlap width: $m$; threshold: $\sigma$
Outputs: output patch: $\phi$
Divide image $\Gamma$ into patches of $n \times n$ pixels with $m$ pixels width of overlap:
    $\psi \leftarrow InitialDivide\left(\Gamma, n, m\right)$
Calculate the proportion of foreground $F$, background $B$ and unknown $U$ regions in each patch:
    $\theta \leftarrow CalculateFBU\left(\Gamma, \psi\right)$
for $j = 0$ to $\psi.size$ do
    if $CurrentBlockIsGoodEnough\left(\Gamma, \psi, \theta, j, \sigma\right)$ then
        continue
    else
        Find the directions of merge (Horizontal/Vertical):
            $k \leftarrow FindDirections\left(\Gamma, \psi, \theta, j, \sigma\right)$
        Do the combination and get the merged patch:
            $rect \leftarrow MergePatch\left(\Gamma, \psi, k, j, \sigma\right)$
        Re-Divide the merged patch to ensure the new patch size being invariant as before:
            $\phi \leftarrow ReDivisionPatch\left(rect, k\right)$
    end if
end for
```

line assumption is roughly valid locally in a small window, as illustrated in Fig. 3.

3.3.2. *Rewriting the alpha matte problem*

Plugging the color line model (2) into the matting problem (1), we can obtain

$$I_i = \alpha_i \left[\beta_i F_1 + \left(1 - \beta_i\right) F_2\right] + \left(1 - \alpha_i\right) B_i, \tag{3}$$

and then rearrange the terms in (3) into

$$I_i - B_i = \begin{bmatrix} F_1 - F_2 & F_2 - B_i \end{bmatrix} \begin{bmatrix} \alpha_i \beta_i \\ \alpha_i \end{bmatrix}. \tag{4}$$

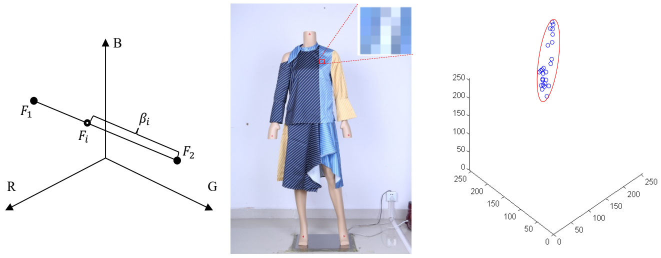
We denote the matrix $[F_1 - F_2, F_2 - B_i]$ by Γ_j, and multiply both sides by $\left(\Gamma_j^T \Gamma_j\right)^{-1} \Gamma_j^T$ with the two rows of which denoted by row vectors r_{j1} and r_{j2}:

$$\begin{aligned} \left(\Gamma_j^T \Gamma_j\right)^{-1} \Gamma_j^T \left(I_i - B_i\right) &= \begin{bmatrix} \alpha_i \beta_i \\ \alpha_i \end{bmatrix}, \\ \begin{bmatrix} r_{j1} \\ r_{j2} \end{bmatrix} \left(I_i - B_i\right) &= \begin{bmatrix} \alpha_i \beta_i \\ \alpha_i \end{bmatrix}. \end{aligned} \tag{5}$$

Taking the second equation in (5), we see that:

$$\alpha_i = a_j^T \left(I_i - B_i\right), \forall i \in \Omega_j, \tag{6}$$

where Ω_j is a small image window (e.g. 5×5) around pixel j and contains pixel i, α_i is a scalar, a_j is a 3×1 column vector, and the same $a_j^T = r_{j2}$ is applied to every pixel in the window Ω_j as Γ_j can be regarded as constant in the window. That is, we can compute opacity α_i for each pixel i in the window Ω_j as a linear combination of the RGB values of that pixel.



(a) (b) (c)

Fig. 3. (a) Color line model: in a small window, each color lies on a straight line between two colors in the RGB space. (b) Taking a small window in a given image. (c) Pixels in the small window in the RGB space.

3.3.3. *Optimization problem*

As with [16], the relationship (6) between α_i and colors I_i and B_i inside a window leads to a natural cost function for our image matting problem:

$$J\left(\alpha, a\right) = \sum_{j \in I} \left\{ \sum_{i \in \Omega_j} \left[\alpha_i - a_j^T \left(I_i - B_i\right)\right]^2 \right\}, \tag{7}$$

where Ω_j is a small window around pixel j, $\alpha = (\alpha_1, \ldots, \alpha_N)^T$ is an $N \times 1$ vector that collects all the N pixels' values in the image I, and a represents the collections of coefficients a_j for all windows. Our goal is thus to find the α and a that minimize the cost function.

This cost function expresses the total squared error over the image, under the linearity assumption in (6) for each window. We want to minimize J to find α_i at each pixel as well as the coefficients a_j for every window Ω_j around pixel j. Since we place a window around each pixel, the windows between adjacent pixels overlap. This enables the propagation of information between neighboring pixels, and thus the αs estimated at pixels are not independent.

The cost function is quadratic in α and a, with $2N$ unknowns for an image with N pixels. To reduce the number of unknowns to N, we follow a trick introduced in [16].

We can rewrite (7) as

$$\begin{aligned} J\left(\alpha, a\right) &= \sum_{j \in I} \left\| \begin{bmatrix} (I_j^1 - B_j^1)^T \\ (I_j^2 - B_j^2)^T \\ \vdots \\ (I_j^\omega - B_j^\omega)^T \end{bmatrix} a_j - \begin{bmatrix} \alpha_j^1 \\ \alpha_j^2 \\ \vdots \\ \alpha_j^\omega \end{bmatrix} \right\|_2^2 & (8) \\ &= \sum_{j \in I} \left\| G_j a_j - \vec{\alpha}_j \right\|^2, & (9) \end{aligned}$$

where ω denotes the number of pixels in the window Ω_j; $\left\{I_j^1, I_j^2, \cdots, I_j^\omega\right\}$, $\left\{B_j^1, B_j^2, \cdots, B_j^\omega\right\}$ and $\left\{\alpha_j^1, \alpha_j^2, \cdots, \alpha_j^\omega\right\}$ represent the ordered list of image colors, background colors and α values for all pixels inside window Ω_j; and G_j and $\vec{\alpha}_j$ are compact representations of corresponding terms in (8).

For each pixel j, when the matte $\vec{\alpha}_j$ is known, a_j can be solved via least squares estimation:

$$a_j^* = \left(G_j^T G_j\right)^{-1} G_j^T \vec{\alpha}_j. \tag{10}$$

This means that, if we denote $\bar{G}_j = I - G_j \left(G_j^T G_j\right)^{-1} G_j^T$, we can rewrite (9) as

$$J(\alpha) = \sum_{j \in I} \vec{\alpha}_j^T \bar{G}_j^T \bar{G}_j \vec{\alpha}_j, \tag{11}$$

and some further algebra shows that the (i, j)-th entry of $\bar{G}_j^T \bar{G}_j$ may be expressed as

$$\delta_{i,j} - \frac{\left(I_i - B_i\right)^T \left(I_j - B_j\right)}{\sum_{k \in \Omega_j} \left(I_j^k - B_j^k\right)^T \left(I_j^k - B_j^k\right)}. \tag{12}$$

Then by defining the matting Laplacian $L_{I-B} = \bar{G}_j^T \bar{G}_j$, we can rewrite $J(\alpha)$ as

$$J(\alpha) = \alpha^T L_{I-B} \alpha. \tag{13}$$

3.3.4. Solving the alpha matte

To extract an alpha matte matching the trimap, the optimization problem now becomes

$$\begin{aligned} \min_{\alpha} \ & \alpha^T L_{I-B} \alpha \\ s.t. \ & \alpha_i = 1 \ \text{ if } i \in F \\ & \alpha_i = 0 \ \text{ if } i \in B \end{aligned} \tag{14}$$

We can rewrite (14) as

$$\min_{\alpha} \ \alpha^T L_{I-B} \alpha + \lambda \left(\alpha - \alpha_k\right)^T D \left(\alpha - \alpha_k\right), \tag{15}$$

where α_k is an $N \times 1$ vector equal to 1 at known foreground pixels and 0 everywhere else; D is a diagonal matrix whose diagonal elements are equal to 1 when in the trimap the pixel belongs to F or B; and λ is set to be a large number (e.g., 100) such that the solution is forced to agree closely with the trimap. Setting the derivative of (15) to 0 results in

$$\left(L_{I-B} + \lambda D\right) \alpha = \lambda \alpha_k, \tag{16}$$

which will solve the matte.

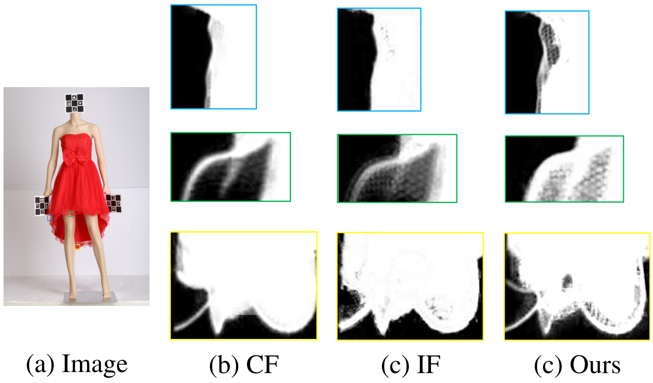


Fig. 4. Comparison of the closed-form (CF) matting, the information-flow (IF) matting and our algorithm.

4. EXPERIMENTAL STUDIES

4.1. Experimental setting

We usually define the matting Laplacian L_{I-B} using small windows of 3×3 or 5×5. When the foreground and background color distributions are not very complex, using bigger windows is helpful, although it will increase computation time. We solve the matting system using C++'s solver for a 4480×6720 image on a 2.6GHz CPU. Because of the memory limitations, it is impossible to solve the alpha matte of the whole image at one time. We propose a patch division and merge strategy (see Section 3.2) for developing the matting algorithm at the patch level.

4.2. Results and analysis

We compared our algorithm with the closed-form matting [2] and the information-flow matting [3].

Fig. 1 shows a visual comparison for an image with complex structure such as holes. In the two zoomed panels (for the yellow and blue boxes in the original image, respectively), the closed-form matting method wrongly fills in some holes, while our method is able to preserve these details and produce a cleaner matte than the information-flow matting method.

A more challenging image is shown in the Fig. 4 in which the foreground object (lace) is translucent. In the three zoomed panels, our method is able to preserve more complex details and sharpness even around translucent areas than the closed-form matting method and the information-flow matting method, while producing a clean matte.

These results indicate that taking into account some auxiliary knowledge of background can be more efficient and accurate than the affinity-based optimization used in the closed-form matting and the information-flow matting.

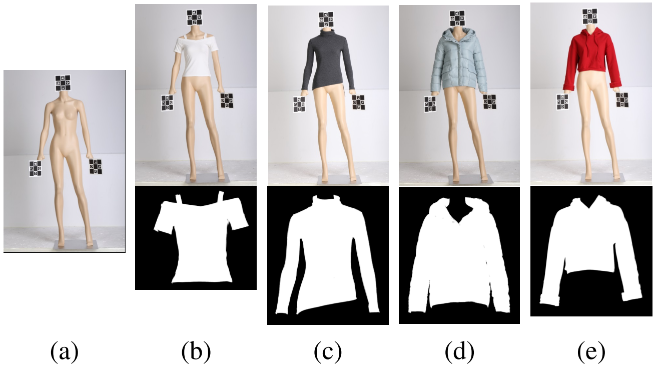


Fig. 5. Examples of clothing image matting results. For a given background image (a), we can get many alpha mattes (b-e) efficiently.

4.3. Discussion

To further evaluate our approach, we constructed a test dataset of four examples as shown in Fig 5, from which we can see that only one background image with the mannequin is needed to extract several alpha mattes efficiently. In fact, when constructing a database of a large number of clothing images for a virtual clothing try-on system, many clothes are often worn by the same mannequin, and thus we do not need to take a rough background image for each input image.

5. CONCLUSION

In this paper, we have presented a novel approach to automatically generating high quality image mattes with auxiliary information from rough background images. As a by-product, our approach can automatically generate trimaps. Also, experiments have shown that with such auxiliary information about background, our algorithm can improve the quality of mattes in challenging regions with a significant portion of complex texture and translucent material.

6. REFERENCES

[1] Yuwei Meng, PY Mok, and Xiaogang Jin, "Interactive virtual try-on clothing design systems," *Computer-Aided Design*, vol. 42, no. 4, pp. 310–321, 2010.

[2] Anat Levin, Dani Lischinski, and Yair Weiss, "A closed-form solution to natural image matting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 228–242, 2008.

[3] Yagiz Aksoy, Tunç Ozan Aydin, and Marc Pollefeys, "Designing effective inter-pixel information flow for natural image matting," in *CVPR*, 2017, pp. 228–236.

[4] Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski, "A Bayesian approach to digital matting," in *CVPR 2001*. IEEE, 2001, vol. 2, pp. II–II.

[5] Eduardo SL Gastal and Manuel M Oliveira, "Shared sampling for real-time alpha matting," *Computer Graphics Forum*, vol. 29, no. 2, pp. 575–584, 2010.

[6] Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun, "A global sampling method for alpha matting," in *CVPR 2011*. IEEE, 2011, pp. 2049–2056.

[7] Ehsan Shahrian and Deepu Rajan, "Weighted color and texture sample selection for image matting," in *CVPR 2012*. IEEE, 2012, pp. 718–725.

[8] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum, "Poisson matting," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 315–321, 2004.

[9] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann, "Random walks for interactive alpha-matting," in *Proceedings of VIIP*, 2005, vol. 2005, pp. 423–429.

[10] Yongfang Shi, Oscar C Au, Jiahao Pang, Ketan Tang, Wenxiu Sun, Hong Zhang, Wenjing Zhu, and Luheng Jia, "Color clustering matting," in *Multimedia and Expo (ICME), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.

[11] Xiaowu Chen, Dongqing Zou, Steven Zhiying Zhou, Qinping Zhao, and Ping Tan, "Image matting with local and nonlocal smooth priors," in *CVPR 2013*, 2013, pp. 1902–1907.

[12] Jue Wang and Michael F Cohen, "An iterative optimization approach for unified image segmentation and matting," in *ICCV 2005*. IEEE, 2005, vol. 2, pp. 936–943.

[13] Nobuyuki Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[14] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[15] Guangying Cao, Jianwei Li, Zhiqiang He, and Xiaowu Chen, "Divide and conquer: a self-adaptive approach for high-resolution image matting," in *Virtual Reality and Visualization (ICVRV), 2016 International Conference on*. IEEE, 2016, pp. 24–30.

[16] Richard J Radke, *Computer vision for visual effects*, Cambridge University Press, 2013.