Automated Digital Hair Removal by Threshold Decomposition and Morphological Analysis

Joost Koehoorn¹, André C. Sobiecki¹(✉), Daniel Boda², Adriana Diaconeasa², Susan Doshi⁴, Stephen Paisey⁵, Andrei Jalba³, and Alexandru Telea¹,²

¹ JBI Institute, University of Groningen, Groningen, The Netherlands
{jo.koehoorn, a.c.telea, a.sobiecki}@rug.nl
² University of Medicine and Pharmacy ‘Carol Davila’, Bucharest, Romania
{daniel.boda, adriana.diaconeasa}@umf.ro
³ Eindhoven University of Technology, Eindhoven, The Netherlands
a.c.jalba@tue.nl
⁴ School of Computer Science and Informatics, Cardiff University, Cardiff, UK
doshisk@cardiff.ac.uk
⁵ School of Medicine, Cardiff University, Cardiff, UK
paiseysj@cf.ac.uk

Abstract. We propose a method for digital hair removal from dermoscopic images, based on a threshold-set model. For every threshold, we adapt a recent gap-detection algorithm to find hairs, and merge results in a single mask image. We find hairs in this mask by combining morphological filters and medial descriptors. We derive robust parameter values for our method from over 300 skin images. We detail a GPU implementation of our method and show how it compares favorably with five existing hair removal methods.

Keywords: Hair removal · Threshold sets · Morphology · Skeletonization

1 Introduction

Automatic analysis of pigmented skin lesions [13,7] occluded by hair is a challenging task. Several digital hair removal (DHR) methods address this by finding and replacing hairs by plausible colors based on surrounding skin. However, DHR methods are challenged by thin entangled, or low-contrast hairs [18,27,14,11,2,12].

We address the above problems by converting the skin image into a threshold-set and adapting a gap-detection technique to find hairs in each threshold layer. Found gaps are merged into a single hair mask, where we find actual hairs by using 2D medial axes, and finally remove them by image inpainting.

Section 2 reviews related work on digital hair removal. Section 3 details our method. Section 4 presents its implementation. Section 5 compares our results with five DHR methods and also shows an extra application for CBCT image restoration. Section 6 discusses our method. Section 7 concludes the paper.

2 Related Work

In the past decade, many DHR methods have been proposed. DullRazor finds dark hairs on light skin by morphological closing using three structuring elements
that model three line orientations [18]. Different morphological operators were used in [22,19]. Hairs are removed by bilinear [18] or PDE-based inpainting [26]. Prewitt edge detection [14] and top-hat filtering [27] help finding low-contrast or thin-and-curled hairs. Huang et al. find hairs by multiscale matched filtering and hysteresis thresholding and remove these by PDE-based inpainting [12]. However, this method is quite slow (minutes for a typical dermoscopy image). VirtualShave finds hairs by top-hat filtering, like [27], and uses three density, sphericity, and convex-hull sphericity metrics to separate true positives (hairs) from other high-contrast details (false positives) [11]. Abbas et al. find hairs by a derivatives-of-Gaussian (DOG) filter [1,2]. However, this method has many parameters whose setting is complex. Finding other elongated objects such as arterial vessels and fibers is also addressed by path opening methods [8] and grayscale skeletons [10]. The last method also permits filling thin gaps similar to our hairs.

Table 1 captures several aspects of the above DHR methods. As visible, there is little comparison across methods. As method implementations are not publicly available (except [18,12]), comparison is hard. Hence, for our new DHR method outlined next, one main aim is to show how it compares to all reviewed methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Hair detector</th>
<th>Inpainting by</th>
<th>Compared with</th>
<th># test images</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DullRazor [18]</td>
<td>generalized morphological closing</td>
<td>bilinear interpolation</td>
<td>DullRazor</td>
<td>5</td>
<td>available</td>
</tr>
<tr>
<td>Huang et al. [12]</td>
<td>multiscale matched filters</td>
<td>median filtering</td>
<td>DullRazor</td>
<td>20</td>
<td>available</td>
</tr>
<tr>
<td>Xie et al. [27]</td>
<td>top-hat operator</td>
<td>anisotropic diffusion [20]</td>
<td>DullRazor</td>
<td>40</td>
<td>not available</td>
</tr>
<tr>
<td>E-shaver [14]</td>
<td>Prewitt edge detector</td>
<td>color averaging</td>
<td>DullRazor</td>
<td>5</td>
<td>not available</td>
</tr>
<tr>
<td>Our method</td>
<td>gap-detection by multiscale skeletons</td>
<td>fast marching method [24]</td>
<td>DullRazor, Xie et al. [27], Huang et al. [12], Fiorese et al. [11], Abbas et al. [2]</td>
<td>over 300</td>
<td>available</td>
</tr>
</tbody>
</table>

3 Proposed Method

Most DHR methods find hairs by local luminance analysis (see Tab. 1, column 2). Such methods often cannot find hairs that have variable color, contrast, thickness, or crispness across an image. Hence, our main idea is to perform a conservative hair detection at all possible luminance values. For this, we propose the following pipeline. First, we convert the input image into a luminance threshold-set representation (Sec. 3.1). For each threshold layer, we find thin hair-like structures using a morphological gap-detection algorithm (Sec. 3.2). Potential hairs found in all layers are merged in a mask image, which we next analyze to remove false-positives (Sec. 3.3). Finally, we remove true-positive hairs by using a classical image inpainting algorithm (Sec. 3.4). These steps are discussed next.
3.1 Threshold-Set Decomposition

We reduce color images first to their luminance component in HSV space. Next, we compute a threshold-set model of the image [28]: Given a luminance image \( I : \mathbb{R}^2 \rightarrow \mathbb{R}_+ \) and a value \( v \in \mathbb{R}_+ \), the threshold-set \( T(v) \) for \( v \) is defined as

\[
T(v) = \{ x \in \mathbb{R}^2 | I(x) \geq v \}. \tag{1}
\]

For \( n \)-bits-per-pixel images, Eqn. 1 yields \( 2^n \) layers \( T_i = T(i), 0 \leq i < 2^n \). We use \( n = 8 \) (256 luminances). Note that \( T_j \subset T_i, \forall j > i \), i.e. brighter layers are ‘nested’ in darker ones. If \( I(x) \neq i, \forall x \in \mathbb{R}^2 \), we find that \( T_i = T_{i+1} \). In such cases, we simply skip \( T_i \) from our threshold-set decomposition, as it does not add any information. Our decomposition \( \{T_i\} \) will thus have at most \( 2^n \) layers.

3.2 Potential Hair Detection

We find thin-and-long shapes in each layer \( T_i \) by adapting a recent gap-detection method [23], as follows.

Original Gap-Detection. Given a binary shape \( \Omega \subset \mathbb{R}^2 \) with boundary \( \partial \Omega \), we compute the open-close image \( \Omega_{oc} = (\Omega \circ H) \bullet H \) and close-open image \( \Omega_{co} = (\Omega \bullet H) \circ H \). Here, \( \circ \) and \( \bullet \) denote morphological opening and respectively closing with a disk of radius \( H \) as structuring element. In both \( \Omega_{oc} \) and \( \Omega_{co} \), small gaps get filled; yet, \( \Omega_{co} \) has more gaps filled than \( \Omega_{oc} \), but also fills shallow concavities (dents) along \( \partial \Omega \). Next, the skeleton or medial axis \( S_{\Omega_{oc}} \) of \( \Omega_{oc} \) is computed. For this, we first define the distance transform \( DT_{\partial \Omega} : \mathbb{R}^2 \rightarrow \mathbb{R}_+ \) as

\[
DT_{\partial \Omega}(x \in \Omega) = \min_{y \in \partial \Omega} \| x - y \|. \tag{2}
\]

The skeleton \( S_{\Omega} \) of \( \Omega \) is next defined as

\[
S_{\Omega} = \{ x \in \Omega | \exists f_1, f_2 \in \partial \Omega, f_1 \neq f_2, \| x - f_1 \| = \| x - f_2 \| = DT_{\partial \Omega}(x) \} \tag{3}
\]

where \( f_1 \) and \( f_2 \) are the contact points with \( \partial \Omega \) of the maximally inscribed disc in \( \Omega \) centered at \( x \). From \( S_{\Omega_{oc}} \), the algorithm removes branch fragments that overlap with \( \Omega \), yielding a set \( F = S_{\Omega_{oc}} \setminus \Omega \) that contains skeleton-fragments located in thin gaps that cut deeply inside \( \Omega \). To find all pixels in the gaps, the proposed method convolve the pixels \( x \in F \) with disk kernels centered at the respective pixels and of radius equal to \( DT_{\partial \Omega}(x) \). As shown in [23], this produces an accurate identification of deep indentations, or gaps, in \( \Omega \), while ignoring pixels in shallow dents along \( \partial \Omega \).

Hair-Detection Modification. We observe that, in a binary image with hairs in foreground, hairs are gaps of surrounding background. We next aim to find robustly hairs in all layers \( T_i \). For this, several changes to [23] are needed. First, we note that [23] uses \( DT_{\Omega_{oc}} \) as disk-radius values for gap-filling as they argue that \( \Omega_{co} \) closes more gaps than \( \Omega_{oc} \), supported by the observation that \( DT_{\Omega_{co}}(x) \geq DT_{\Omega_{oc}}(x), \forall x \in F \). Yet, for our hair-removal context, using \( DT_{\partial \Omega_{co}} \) on every layer \( T_i \), and next merging gaps into a single hair-mask, results in too
many areas being marked as hair. The resulting mask proves to be too dense – thus, creates too many false-positive hairs for our next filtering step (Sec. 3.3). Using the smaller $DT_{\partial \Omega_{oc}}$ as disk radius prevents this problem, but fails to find many hair fragments – thus, creates too many false-negatives. To overcome these issues, we propose to use a linear combination of $DT_{\partial \Omega_{oc}}$ and $DT_{\partial \Omega_{co}}$. In detail, we define a set of pairs disk-centers $x$ and corresponding disk-radii $\rho$ as

$$D_\lambda = \{(x, \rho = (1 - \lambda)DT_{\partial \Omega_{co}}(x) + \lambda DT_{\partial \Omega_{oc}}(x)) | x \in F\}$$  

(4)

where $\lambda \in [0, 1]$ gives the effect of $DT_{\partial \Omega_{oc}}$ and $DT_{\partial \Omega_{co}}$ to the disk radius. A value of $\lambda = 0.2$, found empirically (see Sec. 6), avoids finding too many gaps (false-positives), while also preventing too many false-negatives.

Let $D$ be the union of pixels in all disks described by $D_\lambda$. We next find the gaps $G$ that potentially describe hairs as the difference

$$G = D \setminus \Omega.$$  

(5)

We apply Eqn. 5 to compute a gap $G_i$ from every shape $\Omega_i := T_i$. Next, we merge all resulting gaps $G_i$ together into a single hair-mask image $M = \bigcup_{i=0}^{2^n} G_i$.

Fig. 1. a) Input image. b) Full hair mask $M$. c) Simplified mask skeleton $S_M^\tau$. d) Filtered mask $M^f$. e) Mask created by [12]. f) Inpainted hair using $M^f$.

Morphological closing finds only hairs darker than skin. To find hairs lighter than skin, we replace closing by morphological opening. Having the dark-hair
and light-hair masks $M^d$ and $M^l$, we can next either combine the two or select one mask to use further. We observed in virtually all our test images that dark and light hairs do not occur together. So, we use next the mask $M \in \{M^d, M^l\}$ that most likely contains hairs, i.e., which maximizes the length of the longest skeleton-branch in $S_M$. For example, for the image in Fig. 1a, which has mainly dark hairs, our method will select to use the mask $M := M^d$ (Fig. 1b).

### 3.3 False Positive Elimination

Since we search for gaps on every threshold-level, we find more gaps than traditional approaches, e.g. [18,27,14,12]. Filtering out ‘false positives’ (gaps unlikely to be hairs), is thus necessary. We achieve this in four steps, outlined below.

**Component Detection.** First, we extract from $M$ all 8-connected foreground components $C_i \subset M$. We skip components less than 1% of the size of image $M$, as these cannot possibly be elongated hairs. Remaining components are analyzed next to see if they are hairs or not.

**Hair Skeletons.** Hair fragments are long and thin. To measure such properties on our components $C_i$, we use their skeletons $S_{\partial C_i}$. Yet, components $C_i$ may have jagged borders, due to input-image noise, shadows, or resolution limits (Fig. 1b), so $S_{\partial C_i}$ have many short spurious branches. We discard these and keep each component ‘core’ by pruning each $S_{\partial C_i}$ as in [25]: From $S_{\partial \Omega}$, we produce a skeleton $S^\tau_{\partial \Omega}$ which keeps only points in $S_{\partial \Omega}$ caused by details of $\partial \Omega$ longer than $\tau$. By making $\tau$ proportional to the component’s boundary length $\|\partial C_i\|$, we ensure that longer branches are pruned more than shorter ones. We also impose a minimum $\tau_{\text{min}}$ to discard tiny spurious fragments, and a maximum $\tau_{\text{max}}$ to preserve large branches. Hence, the pruning parameter $\tau$ for a component $C_i$ is

$$
\tau = \max(\tau_{\text{min}}, \min(\|\partial C_i\| : \mu, \tau_{\text{max}}))
$$

where $\mu \in [0,1]$ is used as a scaling parameter. Figure 1c shows the simplified skeleton $S^\tau_{\partial M}$ obtained from the mask $M$ in Fig. 1b.

**Hair Detection.** In DHR, finding if a component is thin and long is done by e.g. (a) fitting lines in a finite number of orientations and checking the length of the longest such line [18]; (b) using principal component analysis to find if the major-to-minor eigenvalue ratio exceeds a threshold [17]; and (c) computing an elongation metric comparing a component’s skeleton-length with its area [27]. Xie et al. argue that (a) and (b) are limited, as they favor mainly straight hairs and yield false-negatives for curled hairs [27]. They alleviate this by an elongation metric equal to the ratio of the area $\|C_i\|$ to the squared length of the ‘central axis’ of $C_i$; but they give no details on how this central-axis (and its length) are computed. In particular, for crossing hairs, i.e., when the skeleton of $C_i$ has multiple similar-length branches, multiple interpretations of the notion of a ‘central axis’ are possible. We also found that (c) also yields many false-negatives, i.e., marks as hair shapes which do not visually resemble a hair structure at all.

To address such issues, we propose a new metric to find if a thin-and-long shape is likely a hair. Let $J_i = \{x_i \in S^\tau_{\partial C_i}\}$ be the set of junctions of $S^\tau_{\partial C_i}$, i.e., pixels where at least three $S^\tau_{\partial C_i}$ branches meet. If the maximum distance
\[ d_{\text{max}} = \max_{x, y \in J_i, x \neq y} \| x - y \| \]

between any two junctions is small, then \( C_i \) is too irregular to be a hair. We also consider the average branch-length between junctions \( d_{\text{avg}} = \| S_{\partial C_i} \| / \| J_i \| \), i.e., the number of skeleton-pixels divided by the junction count. If either \( d_{\text{max}} < \delta_{\text{max}} \) or \( d_{\text{avg}} < \delta_{\text{avg}} \), then \( C_i \) has too many branches to be a thin elongated hair (or a few crossing hairs), so we erase \( S_{\partial C_i} \) from the skeleton image. Good values for \( \delta_{\text{max}} \) and \( \delta_{\text{avg}} \) are discussed in Sec. 6.

Mask Construction. We construct the final mask \( M^f \) that captures hairs by convolving the filtered skeleton-image (in which false-positives have been removed) with disks centered at each skeleton-pixel \( x \) and of radius equal to \( DT_{\partial M}(x) \). Figure 1d shows the mask \( M^f \) corresponding to the skeleton image in Fig. 1c. Comparing it with the hair-mask produced by [12] (Fig. 1e), we see that our mask succeeds in capturing the same amount of elongated hairs, but contains fewer small isolated line-fragments (thus, has fewer false-positives).

3.4 Hair Removal

We remove hairs by using classical inpainting [24] on the hair-mask \( M^f \). To overcome penumbras (pixels just outside \( M^f \) are slightly darker due to hair shadows), which get smudged by inpainting into \( M^f \), we first dilate \( M^f \) isotropically by a \( 3 \times 3 \) square structuring element. This tells why hairs in \( M^f \) in Fig. 1d are slightly thicker than those in Fig. 1b. Figure 1f shows our final DHR result.

4 Implementation

The most expensive part of our method is computing \( M \), which requires distance transforms and skeletons from up to 256 binary images (Sec. 3.2). As these images can be over \( 1024^2 \) pixels for modern dermoscopes, processing a single image must be done within milliseconds to yield an acceptable speed. For this, we use the GPU-based method for exact Euclidean distance transforms in [6]. A simple modification of this method allows us to compute dilations and erosions (by thresholding the distance transform with the radius of the disk structuring element) and simplified skeletons (by implementing the boundary-collapse in [25]). For implementation details, we refer to [28]. We also tested our method on multi-GPU machines by starting \( k \) MPI processes for \( k \) GPUs. Each process \( p \in [0, k) \) does gap-detection on a subset of the threshold-set by launching CUDA threads to parallelize gap-detection at image block level [6]. The \( k \) separate masks \( M_p, 1 \leq p \leq k \) are merged by process 0 into a single mask \( M \), after which it continues with false-positive removal (Sec. 3.3). Connected component detection, done with union-find [21], and hair inpainting [24], are implemented in C++ on the CPU, as they are done on a single image.

5 Results and Comparison

Material. We have tested our method on over 300 skin images. These cover a wide range of skin lesions; hair thickness, color, length, density; and skin colors, acquired by several types of dermoscopes, by three research groups. Some images contain no hair; they let us see how well we can avoid false positives. This is important, as removing non-hair details may affect subsequent analyses [2,12].
Fig. 2. Comparison of our method with DullRazor [17] and Huang et al. [12]. Insets show details.
Methods. We compare our results with five DHR methods: Where an implementation was available [18, 12], we ran our full image-set through it. For the other methods [27, 2, 11], we processed images from the respective papers.

Results. Compared to DullRazor and Huang et al. [12] (Fig. 2), we see that DullRazor cannot remove low-contrast hairs (a,d); and both methods create 'halos' around removed hairs (c,f,e,f). Images (g,h) show two complex lesions, with hair of variable tints, opacity, thickness, and density. For (g), we create less halos around removed hairs than both DullRazor and Huang et al. For (h), our method removes considerably more hair than both methods. Figure 3 compares our results with Xie et al. [27] and Huang et al. We remove more hairs than Xie et al., but also remove a small fraction of the skin. Huang et al. removes all hairs but also massively blurs out the skin. This is undesirable, since such patterns are key to lesion analysis. Figure 4 compares our method with Abbas et al. [2] and Huang et al. We show comparable results to Abbas et al. Huang et al. has issues with thick hairs (a) and also creates undesired hair halos (c). Compared
to Fiorese et al. [11], we show a similar ability in removing both stubble and elongated hairs (Fig. 5). Strikingly, Fiorese et al. changes the hue of the input image, which is undesired. Our method correctly preserves the hue of the image.

**Validation.** We have shown the input images, and obtained DHR results, to two dermatologists having over 11 years of clinical experience. We asked whether the two images would lead them to different interpretations or diagnoses. In all cases, the answer was negative. While a more formal, quantitative, test would bring additional insight, this test tells that our DHR method does not change the images in undesirable ways. Separately, hair removal is obviously desirable, e.g. when using images in automated image-analysis procedures [2,12].

**Other Applications.** Our method can be used beyond DHR. Figure 6 shows an use-case for cone-beam computed tomography (CBCT) images. Positron emission tomography (PET) is a functional imaging modality used to deduce the spatial distribution of a radio-labelled substance injected into a subject. To put PET data in spatial context, high-resolution CBCT images can be acquired and co-registered with PET data. Two types of sensors are inserted into the subject (a mouse under physiological monitoring): soft plastic tubes (S) and hard metal wires (H). H sensors cause streak artifacts, making the CBCT reconstruction (onto which the PET data is overlaid) unusable. Hence, we want to automatically remove them. Doing this by using the CBCT volume is possible but quite expensive and complex. We remove such artifacts directly from the 2D X-ray images used to create the CBCT volume. Our DHR method is suitable for this, since the H implements appear as thin, elongated, and dark 2D shapes in such projections (see Figs. 6 a-c). Figures 6 e-g show the H implement-removal results. As visible, the H implements present in the input images have been successfully detected and removed. In contrast, the S implements, which have lower contrast and are thicker, are left largely untouched. Figure 6 d shows the 3D reconstruction done from the raw X-ray images (without our artifact removal). In the lower part, the image is massively affected by streak artifacts. Figure 6 h shows the reconstruction done from our DHR-processed images. As visible, most streak...
Fig. 6. Artifact removal from CBCT images. (a-c) Input images, with soft (S) and hard metal-wire (H) artifacts. (e-g) Reconstructed images with removed wires. 3D volumetric reconstructions from original images (d) vs our images (h).

artifacts have been removed. In contrast, the thick soft (S) tubes have been preserved by our DHR method and the resulting reconstruction.

6 Discussion

Parameters. To obtain full automation, we ran our method on several tens of skin images (at resolution $1024^2$), varying all its parameters, and selected those values which visually yielded the best results (most true-positive and least false-positive hairs). Next, we computed final parameters by averaging, and tested that these values give good results on our full image test-set. Table 2 presents the final parameter values, used to produce all images in this paper.

<table>
<thead>
<tr>
<th>Description</th>
<th>Definition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>Structuring element radius</td>
<td>Section 3.2</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Gap detection parameter</td>
<td>Equation 4</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Skeleton simplification parameter</td>
<td>Equation 6</td>
</tr>
<tr>
<td>$\tau_{\text{min}}$</td>
<td>Minimum skeleton pruning</td>
<td>Equation 6</td>
</tr>
<tr>
<td>$\tau_{\text{max}}$</td>
<td>Maximum skeleton pruning</td>
<td>Equation 6</td>
</tr>
<tr>
<td>$\delta_{\text{max}}$</td>
<td>Hair detection parameter</td>
<td>Section 3.3</td>
</tr>
<tr>
<td>$\delta_{\text{avg}}$</td>
<td>Hair detection parameter</td>
<td>Section 3.3</td>
</tr>
</tbody>
</table>

Robustness. We reliably remove hairs regardless of thickness, curvature, color, or underlying skin pattern. Very thin and low-contrast hairs or stubble may not get (fully) removed, as they are either not found in $M^f$ or do not meet the elongation criteria (Sec. 3.3). Yet, such details do not influence further analysis tasks.

Speed. We compute an open-close, a close-open, a skeletonization, and a skeleton-to-shape reconstruction step for all 256 thresholds. For a $1024^2$ pixel image, this takes 28 seconds on a MacBook Pro Core i7 with a GT 750M GPU,
and 18 seconds on a comparable desktop PC with a GTX 690. For the same image and desktop PC, DullRazor needs 4 seconds, Fiorese et al. 7 seconds, Abbas et al. 40 seconds, Xie et al. 150 seconds, and Huang et al. about 10 minutes.

**Implementation.** We use [6] to compute distance transforms on the GPU in linear time with the pixel count, and also multiscale skeletons and morphological openings and closings [28]. For inpainting, we use the simple method in [24]. C++ source code of our full method is available at [15].

**Limitations.** For very dense hairs of varying color on high-contrast skin (e.g. Fig. 2h), we cannot fully remove all hairs. Yet, this image type is extremely atypical. Also, other methods [18,12] remove significantly less hairs in such cases.

7 Conclusions

We have proposed a new approach for digital hair removal (DHR) by detecting gaps in all layers of an image threshold-set decomposition. We find false-positives by using medial descriptors to find thin and elongated shapes. We compared our method against five known DHR methods on a set of over 300 skin images – to our knowledge, is the broadest DHR method comparison published so far.

Machine learning techniques [3,16,9] could improve false-positive filtering. Further false-negative avoidance can be improved by extending our method to use additional input dimensions besides luminance, such as hue and texture.

**Acknowledgement.** This work was funded by the grants 202535/2011-8 (CNPq, Brazil) and PN-II RU-TE 2011-3-0249 (CNCS, Romania).

**References**

Mathematical Morphology and Its Applications to Signal and Image Processing
Benediktsson, J.A.; Chanussot, J.; Najman, L.; Talbot, H. (Eds.)
2015, XIV, 754 p. 324 illus., Softcover
ISBN: 978-3-319-18719-8