Automated Digital Hair Removal by Threshold Decomposition and Morphological Analysis

J. Koehoorn[1], A. Sobiecki[1], D. Boda[2], A. Diaconeasa[2], S. Doshi[4], S. Paisey[5], A. Jalba[3], and A. Telea[1,2]

[1] JBI Institute, University of Groningen, the Netherlands
{j.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, the Netherlands
a.c.jalba@tue.nl
[4] School of Computer Science and Informatics, Cardiff University, UK
doshisk@cardiff.ac.uk
[5] School of Medicine, Cardiff University, UK
paiseysj@cf.ac.uk

Abstract. We propose a new approach to 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. Next, we find hairs in this mask by combining morphological filters and medial descriptors. We derive robust parameter settings for our method based on its application on over 300 skin images. We detail a GPU implementation of our method and show how it compares favorably with five existing digital hair removal methods.

Keywords: Hair removal, threshold sets, morphology, skeletonization

1 Introduction

Automatic analysis of pigmented skin lesions [14,7] occluded by hair is a challenging task. Several *digital hair removal* (DHR) methods address this by finding and replacing such hairs by plausible colors based on surrounding skin. Despite much work in this area [19,30,15,12,2,13], DHR methods are challenged by hairs which are thin, entangled, of similar contrast or color to underlying skin, or overlaid on a highly-textured skin structure.

We present a new DHR approach that addresses most above problems. After converting the skin image into a luminance threshold-set, we adapt a recent gap-detection technique to find thin structures that are potential hairs in each threshold layer. All found gaps are next merged into a single hair mask, from which we find actual hairs using a combination of morphological techniques and 2D medial axes, and finally remove these by image inpainting. We implement our pipeline on the GPU, which yields speeds comparable to all DHR algorithms we are aware of, while producing results of similar or higher quality.

Section 2 reviews related work on digital hair removal. Section 3 details our DHR method. Section 4 presents implementation details. Section 5 compares our results with five DHR methods and also shows a separate application for the restoration of CBCT images. Section 6 discusses our method's speed, robustness, and parameters. Section 7 concludes the paper.

2 Related Work

In the past decade, several DHR methods have been proposed. DullRazor, the first and arguably best known method, finds dark hairs on light skin by a generalized morphological closing using three structuring elements that model three line orientations [19]. Different morphological operators were used in [23,20]. DullRazor removes hairs by basic image inpainting (bilinear color interpolation), further extended to PDE-based inpainting [29]. Yet, DullRazor cannot find and remove low-contrast hairs or thin or highly curled hairs. E-shaver improves upon DullRazor by Prewitt edge detection to find light-colored and/or thin hairs [15]. However, the found edges are thresholded, so hair close to skin color is hard to find. Top-hat filtering has also been used to find hairs [30]. Huang *et al.* find hairs by multiscale matched filtering and hysteresis thresholding and remove these by region growing with PDE-based inpainting [13]. However, this method is quite slow (minutes for a typical dermoscopy image). VirtualShave finds hairs by top-hat filtering, similar to [30], and uses three density, sphericity, and convex-hull sphericity metrics to separate thin elongated structures (hairs) from other high-contrast details (false positives) [12]. Abbas *et al.* propose to find hairs by a derivatives-of-Gaussian (DOG) filter [1,2]. While this method shows good results on a few selected images, it 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 summarizes several aspects of the key DHR methods reviewed above. As visible, there is little comparison done across existing methods. As implementations of these methods are not publicly available (except for [19,13]), assessing an existing (or newly proposed) DHR method is hard. Hence, when presenting our new DHR method next, one main aim is to show how it compares to all reviewed methods on a large image set.

Table 1: Comparison of existing digital hair removal methods.

Method	Hair detector	Inpainting by	Compared with	# test images	Implementation
DullRazor [19]	generalized morphological closing	bilinear interpolation	–	5	available
Huang *et al.* [13]	multiscale matched filters	median filtering	DullRazor	20	available
Fiorese *et al.* [12]	top-hat operator	PDE-based [4]	DullRazor	20	not available
Xie *et al.* [30]	top-hat operator	anisotropic diffusion [21]	DullRazor	40	not available
E-shaver [15]	Prewitt edge detector	color averaging	DullRazor	5	not available
Abbas *et al.* [2]	derivative of Gaussian	coherence transport [5]	DullRazor, Xie *et al.* [30]	100	not available
Our method	gap-detection by multiscale skeletons	fast marching method [27]	DullRazor, Xie *et al.* [30], Huang *et al.* [13], Fiorese *et al.* [12] Abbas *et al.* [2]	over 300	available

3 Proposed Method

Most DHR methods detect hairs by local analysis of the image luminance (see Tab. 1, column 2). Such approaches often have difficulties to 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 their HSV representation. Next, we compute a threshold-set representation of this image [31]: Given a luminance image $I : \mathbb{R}^2 \to \mathbb{R}_+$ and a value $v \in \mathbb{R}_+$, the threshold-set $T(v)$ for v is defined as

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

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), which models commonly-used 8-bit images. Note that $T_j \subset T_i, \forall j > i$, *i.e.* brighter layers are 'nested' within darker ones. Moreover, if $I(\mathbf{x}) \neq i, \forall \mathbf{x} \in \mathbb{R}^2$, 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 structures in each layer T_i by enhancing a recent general-purpose gap-detection algorithm based on morphological openings/closings and skeletonization [24]. We next present the original algorithm, followed by our enhancements for hair detection.

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 Ω_{oc} and Ω_{co}, small gaps get filled; yet, Ω_{co} has more gaps filled than Ω_{oc}, but also fills shallow concavities (dents) along $\partial\Omega$. Next, the skeleton or medial axis $S_{\Omega_{oc}}$ of Ω_{oc} is computed. For this, we first define the distance transform $DT_{\partial\Omega} : \mathbb{R}^2 \to \mathbb{R}_+$ as

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

The skeleton S_Ω of Ω is next defined as

$$S_\Omega = \{\mathbf{x} \in \Omega | \exists \mathbf{f}_1, \mathbf{f}_2 \in \partial\Omega, \mathbf{f}_1 \neq \mathbf{f}_2, \|\mathbf{x} - \mathbf{f}_1\| = \|\mathbf{x} - \mathbf{f}_2\| = DT_{\partial\Omega}(\mathbf{x})\} \tag{3}$$

where $\mathbf{f}_1$ and $\mathbf{f}_2$ are the contact points with $\partial\Omega$ of the maximally inscribed disc in Ω centered at $\mathbf{x}$, known also as *feature transform* (FT) points [25]. The FT itself assigns to any point in $\mathbb{R}^2$ its closest point on $\partial\Omega$, and is defined as

$$FT_{\partial\Omega}(\mathbf{x} \in \Omega) = \arg\min_{\mathbf{y} \in \partial\Omega} \|\mathbf{x} - \mathbf{y}\|. \tag{4}$$

From $S_{\Omega_{oc}}$, the gap-filling algorithm removes branch fragments that overlap with the shape Ω, yielding a set $F = S_{\Omega_{oc}} \setminus \Omega$. The set F thus contains skeleton-fragments located in thin *gaps* that cut *deeply* inside the shape Ω. To find all pixels in the gaps, the proposed method convolve the pixels $\mathbf{x} \in F$ with disk kernels centered at the respective pixels and of radius equal to $DT_{co}(\mathbf{x})$. As shown in [24], this produces an accurate identification of deep indentations, or gaps, in Ω, while ignoring pixels in shallow dents along $\partial\Omega$.

Hair-detection modification: We observe that, in a binary image with hairs in the foreground, hairs are gaps of surrounding background. We next aim to find hairs in all layers T_i. To robustly find such hairs in our images T_i, several changes to [24] are needed. First, we note that [24] uses $DT_{\Omega_{co}}$ as disk-radius values for gap-filling since they argue that Ω_{co} closes more gaps than Ω_{oc}. This is supported by the observation that $DT_{\Omega_{co}}(\mathbf{x}) \geq DT_{\Omega_{oc}}(\mathbf{x}), \forall \mathbf{x} \in F$. However, for our hair-removal context, using $DT_{\partial\Omega_{co}}$ on every layer T_i, and subsequently 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 $\mathbf{x}$ and corresponding disk-radii ρ as

$$D_\lambda = \{(\mathbf{x}, \rho = (1-\lambda)DT_{\partial\Omega_{co}}(\mathbf{x}) + \lambda DT_{\partial\Omega_{oc}}(\mathbf{x}))\,|\mathbf{x} \in F\} \tag{5}$$

where $\lambda \in [0, 1]$ gives the contribution of the two distance transforms $DT_{\partial\Omega_{oc}}$ 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_λ. We next find the gaps G that potentially describe hairs as the difference

$$G = D \setminus \Omega. \tag{6}$$

We apply Eqn. 6 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 $M = \bigcup_{i=0}^{2^n} G_i$.

Morphological closing finds only hairs darker than surrounding 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 $S_{\partial M}$. For example, for the image in Fig. 1 a, which has mainly dark hairs, our method will select to use the mask $M := M^d$ (Fig. 1 b).

3.3 False Positive Elimination

Since we conservatively find gaps that occur on *every* threshold-level, we inherently find more gaps than traditional approaches, *e.g.* [19,30,15,13]. 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 shapes. To measure such properties on our components C_i, we use their skeletons $S_{\partial C_i}$. Yet, hair-like shapes in M may have jagged borders, due to input-image noise, shadows, or resolution limitations (Fig. 1 b). Hence, the skeletons $S_{\partial C_i}$ will have many short spurious branches. We discard such branches and keep those describing each component 'core' by pruning each $S_{\partial C_i}$ following [28]: Given a skeleton $S_{\partial\Omega}$, we produce a skeleton $S^\tau_{\partial\Omega}$ which keeps only points in $S_{\partial\Omega}$ caused by details of $\partial\Omega$ longer than τ. We adapt this pruning by making τ proportional to the component's boundary

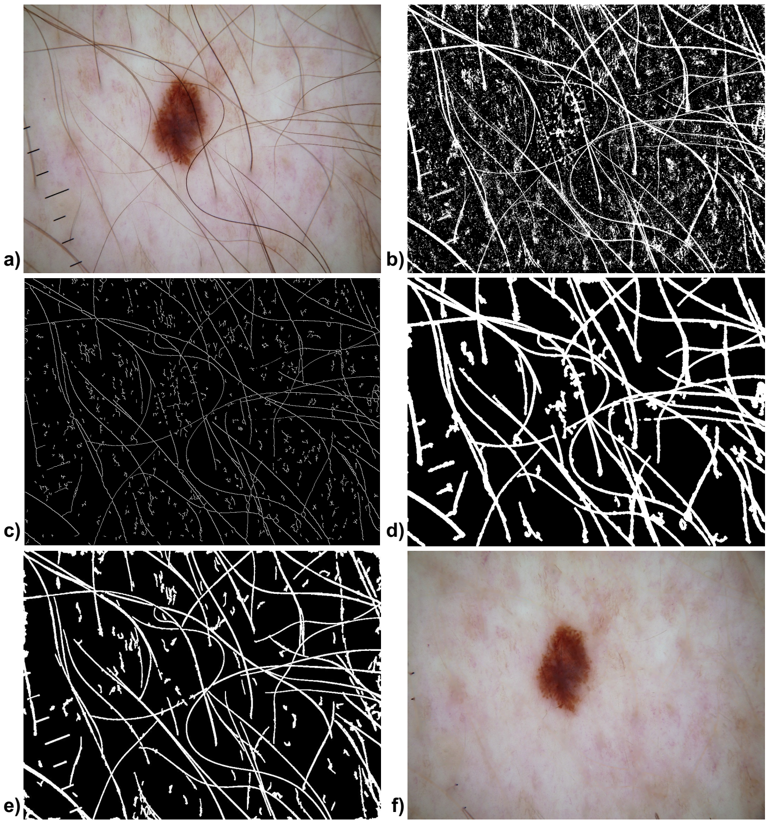

Fig. 1: a) Input image. b) Full hair mask M found by our method. c) Simplified mask skeleton S^{τ}_{M}. d) Filtered mask M^{f}. e) Mask created by Huang *et al.* [13]. f) Inpainted hair using M^{f}.

length $\|\partial C_i\|$, so that longer branches are pruned more than shorter ones. We also impose a minimum τ_{min} to discard tiny spurious fragments, and a maximum τ_{max} so as to preserve large branches. Overall, the pruning parameter τ for a component C_i is given by

$$\tau = \max(\tau_{min}, \min(\|\partial C_i\| \cdot \mu, \tau_{max})) \tag{7}$$

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

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 [19]; (b) using principal component analysis to find if the major-to-minor eigenvalue ratio exceeds a given threshold [18]; and (c) computing an elongation metric comparing a component's skeleton-length with its area [30]. Xie *et al.* argue that (a) and (b) are ineffective, as they favor mainly straight hairs and yield false-negatives for highly curled hairs [30]. To alleviate this, they use an elongation metric equal to the ratio of the area $\|C_i\|$ to the squared length of the 'central axis' of C_i. However, they give no details on how this central-axis (and its length) are precisely 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 overcome such issues, we propose a new metric to find if a thin-and-long shape is likely a hair. Let $J_i = \{\mathbf{x}_i \in S^{\tau}_{\partial C_i}\}$ be the set of junctions of $S^{\tau}_{\partial C_i}$, *i.e.*, pixels where at least three branches of $S^{\tau}_{\partial C_i}$ meet. If the maximum distance $d_{max} = \max_{\mathbf{x} \in J_i, \mathbf{y} \in J_i, \mathbf{x} \neq \mathbf{y}} \|\mathbf{x} - \mathbf{y}\|$ 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 $d_{avg} = \|S_{\partial C_i}\| / \|J_i\|$, *i.e.*, the number of skeleton-pixels divided by the junction count. If either $d_{max} < \delta_{max}$ or $d_{avg} < \delta_{avg}$, then C_i has too many branches with respect to its overall size to be a thin elongated hair (or a few crossing hairs), so we erase $S^{\tau}_{\partial C_i}$ from the skeleton image. Good values for δ_{max} and δ_{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 $\mathbf{x}$ and of radius equal to $DT_{\partial M}(\mathbf{x})$. Figure 1 d shows the mask M^f corresponding to the skeleton image in Fig. 1 c. Comparing it with the hair-mask produced by [13] (Fig. 1 e), 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 a classical inpainting algorithm [27] 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 × 3 square structuring element. This tells why hairs in M^f in Fig. 1 d are slightly thicker than those in *e.g.* in Fig. 1 b. Figure 1 f shows our final DHR result, illustrating the hair removal.

4 Implementation

The most expensive part of our method is the computation of M, which requires computing distance transforms and skeletons from up to 256 binary images (Sec. 3.2). As these images can be large, *e.g.* over 1024^2 pixels for modern dermoscopy devices, processing a single image must be done within milliseconds to yield an acceptable total processing time. To achieve this, we use the efficient GPU-based method for computing exact Euclidean distance and feature transforms in [6]. A simple modification of this method, implemented in CUDA, also allows us to compute dilations and erosions (by thresholding the distance transform with the radius of the desired disk structuring element) and simplified skeletons (by implementing the boundary-collapse in [28]). For implementation details, we refer to [31]. We also implemented our method on multi-GPU machines by starting k MPI processes for k GPUs. Each process $p \in [0, k)$ performs gap-detection on a subset of the threshold-set, further launching CUDA threads to parallelize gap-detection at image block level, see [6]. This yields k separate masks $M_p, 1 \leq p \leq k$. Process 0 collects all masks M_p and merges them into a single mask M, after which it continues with false-positive removal (Sec. 3.3). Connected component detection, done with union-find [22], and hair inpainting [27], are implemented in C++ on the CPU, as they are done on a single image. Performance figures are discussed next in Sec. 5.

5 Results and Comparison

Material: We have tested our method on a set of over 300 skin images. These cover a wide range of skin lesions, hair types (thickness, color, length, density), and skin colors, acquired

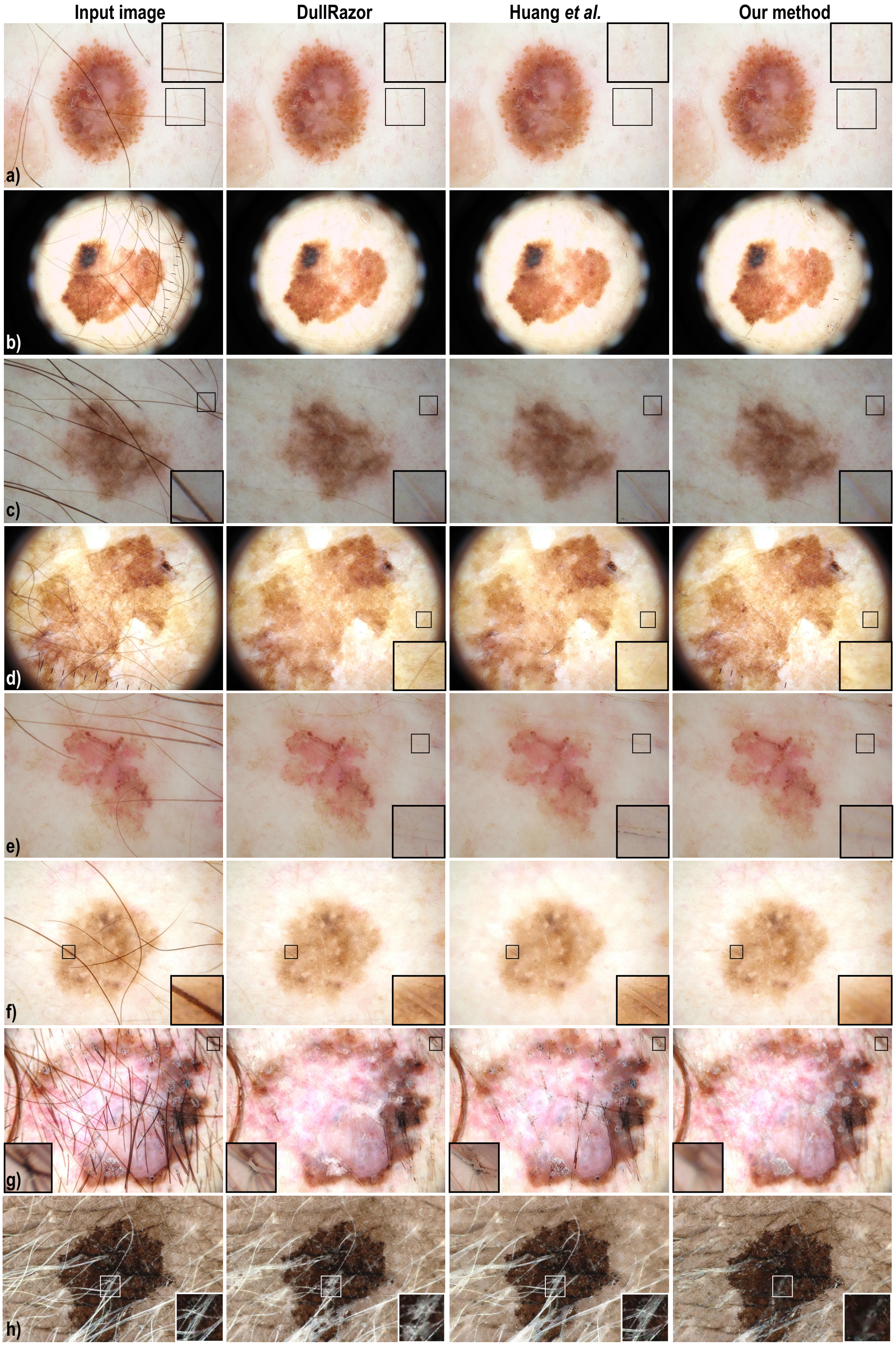

Fig. 2: Comparison of our method with DullRazor [18] and Huang *et al.* [13]. Insets show details.

by several types of dermoscopes, by three different research groups. Some images contain no hair at all; this lets us see how well can our method avoid false positives. This is important, as removing non-hair details may affect subsequent analyses and should be avoided [2,13].

Methods: We compared our results with five existing DHR methods: Where an implementation was available [19,13], we ran our entire image-set through the respective tools. For the other methods [30,2,12], we processed the images from the respective papers by our method.

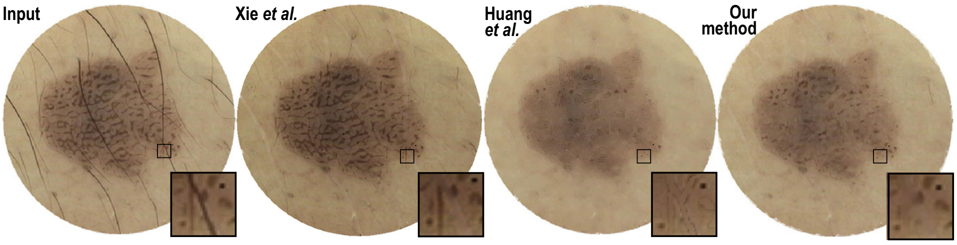

Fig. 3: Comparison between Xie *et al.* [30], Huang *et al.* [13], and our method.

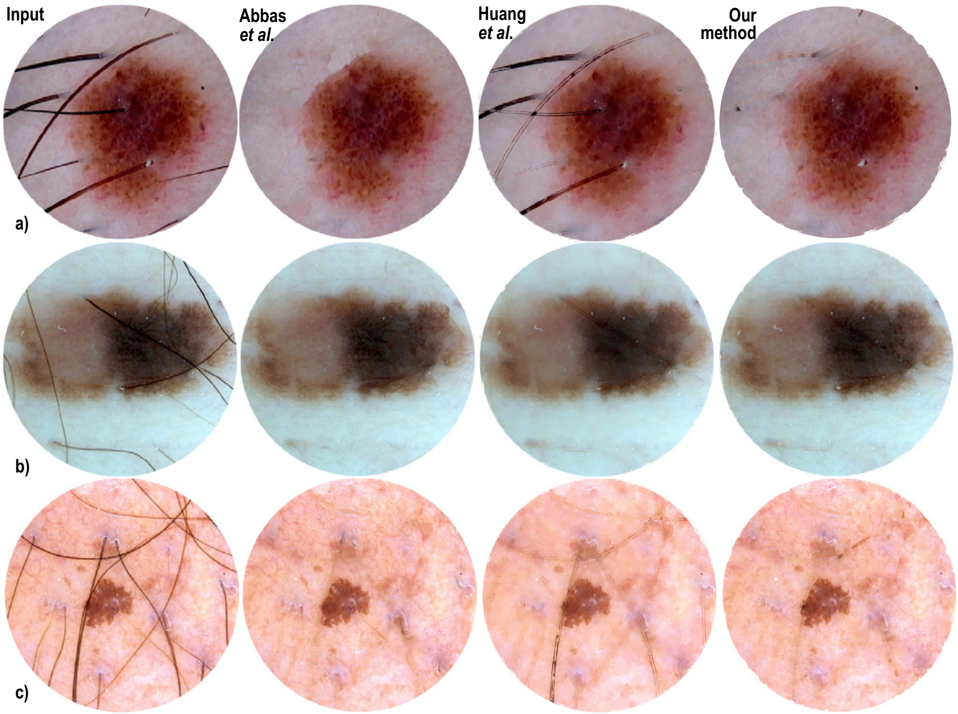

Fig. 4: Comparison between Abbas *et al.* [2], Huang *et al.* [13], and our method.

Results: We first compare with DullRazor and Huang *et al.* [13] (Fig. 2). We see that DullRazor cannot remove low-contrast hairs (a,d) and/or creates 'halos' around removed hairs (c,f). Halos are also created by Huang *et al.* (e,f). Images (g,h) show two very challenging lesions, where overlapping hair has variable tints, opacity, thickness, and density. For image (g), our method creates less halos around removed hairs than both DullRazor and Huang *et al.* For image (h), our method removes considerably more hair than both previous methods.

Figure 3 compares our results with Xie *et al.* [30] and Huang *et al.* While our method removes more hairs than Xie *et al.*, it also removes a small fraction of the skin texture (which is suboptimal). In contrast, Huang *et al.* removes all hairs but also massively blurs out the skin pattern. This is highly undesirable, since such patterns are crucial for further lesion analysis. Figure 4 compares our method with Abbas *et al.* [2] and Huang *et al.* We see that we produce comparable results to Abbas *et al.* In contrast, Huang *et al.* has problems with thick hairs (a) and also produces undesired hair halos (c). Finally, Figure 5 compared our method with Fiorese *et al.* [12]. Our results show a similar ability in removing both stubble and elongated hairs. Interestingly, Fiorese *et al.* changes the overall hue of the input image, which is undesired. In contrast, our method correctly preserves the hue of the image.

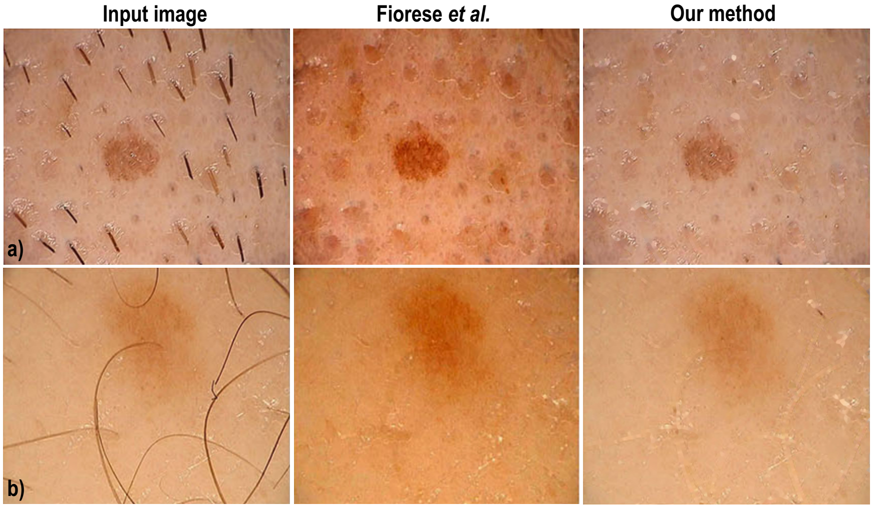

Fig. 5: Comparison between Fiorese *et al.* [12] and our method.

Validation: To get more insight into the quality of our method, we have shown the input images, and corresponding DHR results produced by our tool, to two dermatologists having both 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 confidence, this assessment 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 [2,13].

Other applications: Our method can be used in contexts 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 pharmaceutical 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 elongated sensors are inserted into the subject (a mouse undergoing physiological monitoring): soft plastic tubes (S) and hard metal wires (H). H implements cause streak artifacts, making the CBCT reconstruction (onto which the PET data is superimposed) unusable.

Hence, we would like to automatically remove them. Doing this by processing the CBCT volume is possible but quite expensive and complex. We remove such artifacts on-the-fly, before reconstruction, 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 artifacts have been removed. In contrast, the thick soft (S) tubes have been preserved by our DHR method and the resulting reconstruction.

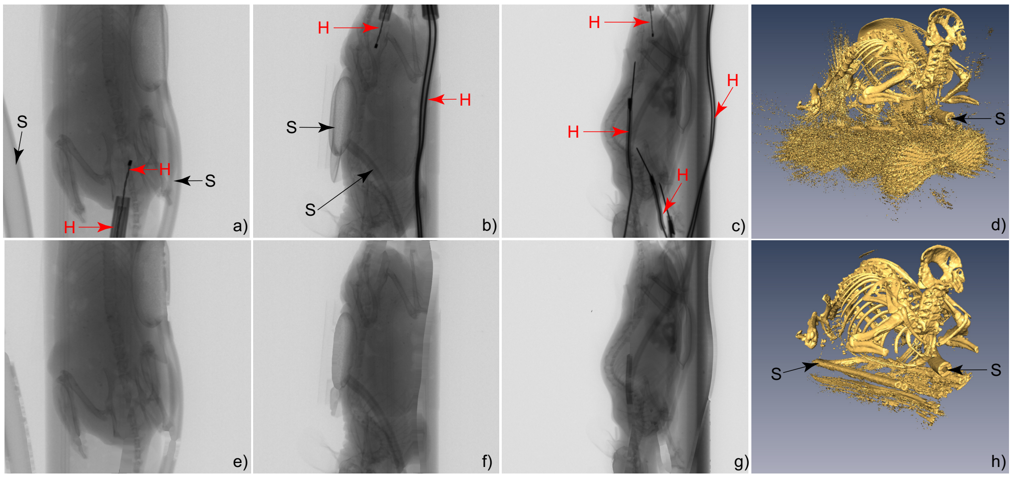

Fig. 6: Artifact removal from CBCT X-ray images. (a-c) Input images, with soft (S) and hard-and-thin metal-wire (H) implements. (e-g) Reconstructed images with removed wires. 3D volumetric reconstructions from original images (d) *vs* our images (h).

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.

Robustness: Our method reliably removes 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 the mask M^f or do not meet the elongation criteria (Sec. 3.3). Yet, we argue that such leftover 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

Table 2: Empirically established parameter values.

	Description	Definition	Value
H	Structuring element radius	Section 3.2	5.0 pixels
λ	Gap detection parameter	Equation 5	0.2
μ	Skeleton simplification parameter	Equation 7	0.05
$\tau_{\min}$	Minimum skeleton pruning	Equation 7	3.0 pixels
$\tau_{\max}$	Maximum skeleton pruning	Equation 7	40.0 pixels
δ_{max}	Hair detection parameter	Section 3.3	20.0 pixels
δ_{avg}	Hair detection parameter	Section 3.3	10.0 pixels

on a MacBook Pro Core i7 laptop with a GT 750M GPU, and 18 seconds on a comparable desktop PC with a GTX 690 GPU. 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.* needs about 10 minutes (following our tests and/or timings given in the respective papers). Our method is thus close to the average speed of the competitor methods.

Implementation: We use the method in [6] to compute feature and distance transforms of binary shapes on the GPU in linear time with the foreground pixel count. This method can be easily adapted to compute multiscale skeletons with the same complexity and compute morphological opening and closing by thresholding the distance transform [31]. For inpainting, we use the simple method in [27]. C++ source code of our full method is available at [16].

Limitations: For very dense hairs of varying color, overlaid on a high-contrast skin texture (*e.g.* Fig. 2 h), our method cannot fully eliminate all hairs. However, this type of image is extremely atypical – we found very few such images in all dermoscopic collections we had access to. Moreover, other methods [19,13] remove significantly less hairs in such cases.

7 Conclusions

We have proposed a new approach for digital hair removal (DHR) by decomposing an image into its threshold-set and detecting gaps in all layers. We remove false-positives by a new filtering that uses 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 – which, to our knowledge, is the broadest DHR method comparison published so far.

Machine learning techniques (k-NN [3], LVQ [17] or SVM [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.

Acknowledgements

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

References

1. Abbas, Q., Fondon, I., Rashid, M.: Unsupervised skin lesions border detection via two-dimensional image analysis. Comp. Meth. Prog. Biom. 104, 1–15 (2011)
2. Abbas, Q., Celebi, M.E., García, I.F.: Hair removal methods: A comparative study for dermoscopy images. Biomedical Signal Processing and Control 6(4), 395–404 (2011)
3. Altman, N.: An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician 46(3), 175–185 (1992)
4. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proc. ACM SIGGRAPH. pp. 417–424 (2000)

5. Bornemann, F., März, T.: Fast image inpainting based on coherence transport. J. Math. Imaging Vis 28, 259–278 (2007)
6. Cao, T., Tang, K., Mohamed, A., Tan, T.: Parallel banding algorithm to compute exact distance transform with the GPU. In: Proc. ACM I3D. pp. 83–90 (2010)
7. Christensen, J., Soerensen, M., Linghui, Z., Chen, S., Jensen, M.: Pre-diagnostic digital imaging prediction model to discriminate between malignant melanoma and benign pigmented skin lesion. Skin Res. Technol. 16 (2010)
8. Cokelaer, F., Talbot, H., Chanussot, J.: Efficient robust d-dimensional path operators. IEEE J. Selected Topics in Signal Processing 6(7), 830–839 (2012)
9. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
10. Couprie, M., Bezerra, F.N., Bertrand, G.: Topological operators for grayscale image processing. J. Electronic Imag. 10(4), 1003–1015 (2001)
11. Criminisi, A., Perez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Process. 13(9), 1134–1144 (2004)
12. Fiorese, M., Peserico, E., Silletti, A.: VirtualShave: automated hair removal from digital dermatoscopic images. In: Proc. IEEE EMBS. pp. 5145–5148 (2011)
13. Huang, A., Kwan, S., Chang, W., Liu, M., Chi, M., Chen, G.: A robust hair segmentation and removal approach for clinical images of skin lesions. In: Proc. EMBS. pp. 3315–3318 (2013)
14. Iyatomi, H., Oka, H., Celebi, G., Hashimoto, M., Hagiwara, M., Tanaka, M., Ogawa, K.: An improved internet-based melanoma screening system with dermatologist-like tumor area extraction algorithm. Comp. Med. Imag. Graph. 32(7), 566–579 (2008)
15. Kiani, K., Sharafat, A.R.: E-shaver: An improved dullrazor for digitally removing dark and light-colored hairs in dermoscopic images. Comput. Biol. Med. 41(3), 139–145 (Mar 2011)
16. Koehoorn, J., Sobiecki, A., Boda, D., Diaconeasa, A., Jalba, A., Telea, A.: Digital hair removal source code (2014), www.cs.rug.nl/svcg/Shapes/HairRemoval
17. Kohonen, T.: Learning vector quantization. In: Self-Organizing Maps, pp. 203–217. Springer (1997)
18. Lee, H.Y., Lee, H.K., Kim, T., Park, W.: Towards knowledge-based extraction of roads from 1m-resolution satellite images. In: Proc. IEEE SSIAI. pp. 171–178 (2000)
19. Lee, T., Ng, V., Gallagher, R., Coldman, A., McLean, D.: Dullrazor®: A software approach to hair removal from images. Comput. Biol. Med. 27(6), 533–543 (1997)
20. Nguyen, N., Lee, T., Atkins, M.: Segmentation of light and dark hair in dermoscopic images: a hybrid approach using a universal kernel. In: Proc. SPIE Med. Imaging. pp. 1–8 (2010)
21. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE TPAMI 12(7), 629–639 (1990)
22. Rahimi, A.: Fast connected components on images. alumni.media.mit.edu/~rahimi/connected (2014)
23. Saugeon, P., Guillod, J., Thiran, J.: Towards a computer-aided diagnosis system for pigmented skin lesions. Comput. Med. Imag. Grap. 27, 65–78 (2003)
24. Sobiecki, A., Jalba, A., Boda, D., Diaconeasa, A., Telea, A.: Gap-sensitive segmentation and restoration of digital images. In: Proc. CGVC. pp. 136–144. Eurographics (2014)
25. Strzodka, R., Telea, A.: Generalized distance transforms and skeletons in graphics hardware. In: Proc. VisSym. pp. 221–230 (2004)
26. Telea, A.: GPU skeletonization source code (2014), www.cs.rug.nl/svcg/Shapes/CUDASkel
27. Telea, A.: An image inpainting technique based on the fast marching method. J. Graphics, GPU, & Game Tools 9(1), 23–34 (2004), source code: www.cs.rug.nl/svcg/Shapes/Inpainting
28. Telea, A., van Wijk, J.J.: An augmented fast marching method for computing skeletons and centerlines. In: Proc. VisSym. pp. 251–259 (2002)
29. Wighton, P., Lee, T., Atkins, M.: Dermascopic hair disocclusion using inpainting. In: Proc. SPIE Med. Imaging. pp. 144–151 (2008)
30. Xie, F., Qin, S., Jiang, Z., Meng, R.: PDE-based unsupervised repair of hair-occluded information in dermoscopy images of melanoma. Comp. Med. Imag. Graph. 33(4), 275–282 (2009)
31. van der Zwan, M., Meiburg, Y., Telea, A.: A dense medial descriptor for image analysis. In: Proc. VISAPP. pp. 285–293 (2013)