Research Article Open Access

Joost Koehoorn, André Sobiecki*, Paulo Rauber, Andrei Jalba, and Alexandru Telea

Efficient and Effective Automated Digital Hair Removal from Dermoscopy Images

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, in terms of removing both long and stubble hair of various colors, contrasts, and curvature. We also discuss qualitative and quantitative validations of the produced hair-free images, and show how our method effectively addresses the task of automatic skin-tumor segmentation for hair-occluded images.

Keywords: Digital hair removal, threshold sets, skeletonization, skin tumor segmentation

1 Introduction

Automatic analysis of pigmented skin lesions occluded by hair is a challenging task [8, 17]. Several *digital hair removal* (DHR) methods aim to address this by finding hairs and replacing them by plausible colors based on surrounding skin. However, DHR methods are challenged by thin, entangled, low-contrast, or thick-and-short (stubble) hairs [2, 13, 16, 18, 24, 39].

To address the above problems, we regard DHR in the context of a threshold-set representation [19]. For this, we represent the input skin image as a set of binary images by thresholding its luminance component. Next, we adapt a gap-detection technique to find potential hairs in each threshold layer. Found gaps are merged into a single hair mask, where we find actual hairs by using 2D medial axes or skeletons. Separately, to robustly detect and remove stubble hair, we propose a morphological filter geared to detecting these structures while keeping remaining image details sharp. Finally, we remove detected hairs by standard image inpainting. To implement our approach, we propose a CPU-GPU pipeline that makes the usage of complex image analysis tools such as threshold sets and medial axes practical and computationally efficient.

Previous work has introduced the use of threshold sets to address digital hair removal [19]. The current paper presents three main contributions as compared to [19]:

1. We show how *stubble* hair can be effectively and efficiently removed, by an additional morphological filter, without affecting surrounding image details and while keeping the removal of long hairs;
2. We demonstrate the added value of DHR for the task of robust skin-tumor *segmentation* for images containing occluding hairs;
3. We present a detailed analysis of the *scalability* of the proposed method, showing how its performance depends linearly on image size and number of threshold values, and not on the hair complexity.

Joost Koehoorn: University of Groningen, The Netherlands, E-mail: j. koehoorn@rug.nl
***Corresponding Author: André Sobiecki:** University of Groningen, The Netherlands, E-mail: a.sobiecki@rug.nl
Paulo Rauber: University of Groningen, The Netherlands, E-mail: p.e.rauber@rug.nl
Andrei Jalba: Eindhoven University of Technology, The Netherlands, E-mail: a.c.jalba@tue.nl
Alexandru Telea: University of Groningen, The Netherlands, E-mail: a.c.telea@rug.nl

The structure of this paper is as follows. 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. Section 6 discusses our method. Section 7 concludes the paper.

2 Related Work

In the past decade, many DHR methods have been proposed. The most known ones are outlined next. DullRazor, the first and arguably most famous, finds dark hairs on light skin by morphological closing using three structuring elements that model three line orientations [24]. Different morphological operators were similarly used in [26, 32]. Prewitt edge detection [18] and top-hat filtering [39] help finding low-contrast or thin-and-curled hairs. Once detected, hairs can be removed by bilinear [24] or PDE-based inpainting [38]. Huang *et al.* find hairs by multiscale matched filtering and hysteresis thresholding and remove these by PDE-based inpainting [16]. However, this method is quite slow (minutes for a typical dermoscopy image). Abbas *et al.* find hairs by a derivatives-of-Gaussian (DOG) filter [1, 2]. However, this method has many parameters whose setting is complex.

While filters such as the above ones succeed in finding *locally* linear high-contrast structures, assessing that such structures form together a long-and-thin object requires *global* analyses. Unless this is done, many false-positives will be found, *e.g.* very short disconnected hair-like fragments of various orientations. Their removal affects the skin texture, which may next adversely affect the use of such texture for image analysis and classification. To address this, VirtualShave finds hairs by top-hat filtering, like [39], and uses three density, sphericity, and convex-hull-sphericity metrics to separate true positives (hairs) from other high-contrast details (false positives) [13]. Finding other elongated objects such as arterial vessels and fibers is also addressed by path opening methods [9] and grayscale skeletons [12]. The last method also permits filling thin gaps similar to our hairs. However, such approaches have not been yet demonstrated for DHR aims.

Table 1 captures several aspects of the above DHR methods. As visible, there is little comparison across methods. One salient aspect in this overview is that existing methods are validated on relatively small image sets and/or do not have public implementations on which other researchers could test them (except [16, 24]). As such, exhaustive comparison of existing DHR methods is hard. For our proposed DHR method described next, extensive comparison with other methods and on large image sets will be a main goal.

Table 1. Comparison of existing digital hair removal methods.

Method	Hair detector	Inpainting by	Compared with	# test images	Implementation
DullRazor [24]	generalized morphological closing	bilinear interpolation	–	5	available (binary)
Huang *et al.* [16]	multiscale matched filters	median filtering	DullRazor	20	available (binary)
Fiorese *et al.* [13]	top-hat operator	PDE-based [4]	DullRazor	20	not available
Xie *et al.* [39]	top-hat operator	anisotropic diffusion [28]	DullRazor	40	not available
E-shaver [18]	Prewitt edge detector	color averaging	DullRazor	5	not available
Abbas *et al.* [2]	derivative of Gaussian	coherence transport [5]	DullRazor, Xie *et al.* [39]	100	not available
Our method	long hair detection by multiscale skeletons; stubble detection by morphological operators	fast marching method [36]	DullRazor, Xie *et al.* [39], Huang *et al.* [16], Fiorese *et al.* [13] Abbas *et al.* [2]	**over 300**	**available** (source code, binary)

3 Proposed Method

Most DHR methods find hairs by local luminance analysis (see Tab. 1, column 2). Such methods often cannot to find hairs that have *variable* color, contrast, thickness, or crispness across an image. Hence, the main idea introduced in [19] is to perform a conservative hair detection at all possible luminance values. For this, the following pipeline is proposed. 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). True-positive hairs are next removed by using a classical image inpainting algorithm (Sec. 3.4). Finally, in addition to [19], we detect short and relatively thick hairs (stubble) by morphological analysis and remove these by the same image inpainting method used for long hairs (Sec. 3.5). 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 [40]: 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), in line with the color resolution of typical dermoscopic images. Note that $T_j \subset T_i, \forall j > i$, *i.e.* brighter layers are 'nested' in darker ones. 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 Long Hair Detection

To find typical (long) hairs, we detect thin-and-long shapes in each layer T_i by adapting a recent gap-detection method [34], 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 $\Omega_{co} = (\Omega \bullet H) \circ H$. In detail, given a so-called structuring element H, which is a disk in our case, we consider the dilation of Ω by H, *i.e.*, the union of copies of H_x (H centered at all pixels $x \in \Omega$), *i.e.*

$$\Omega \oplus H = \bigcup_{x \in \Omega} H_x. \tag{2}$$

Similarly, we define the erosion of Ω by H, which keeps only pixels $x \in \Omega$ where H_x fits inside Ω, as

$$\Omega \ominus H = \{x \in \Omega | H_x \subseteq \Omega\}. \tag{3}$$

Next, we define the opening of Ω as erosion followed by dilation, *i.e.*

$$\Omega \circ H = (\Omega \ominus H) \oplus H, \tag{4}$$

and, analogously, the closing of Ω as dilation followed by erosion, *i.e.*

$$\Omega \bullet H = (\Omega \oplus H) \ominus H. \tag{5}$$

In both Ω_{oc} and Ω_{co}, small gaps of the input image Ω get filled; yet, Ω_{co} has more gaps filled than Ω_{oc}, but also fills shallow concavities (dents) along $\partial\Omega$.

Next, we compute the skeleton or medial axis $S_{\Omega_{oc}}$ of the shape Ω_{oc}. Considering the distance transform $DT_{\partial\Omega} : \mathbb{R}^2 \to \mathbb{R}_+$ given by

$$DT_{\partial\Omega}(\mathbf{x} \in \Omega) = \min_{\mathbf{y}\in\partial\Omega} \|\mathbf{x} - \mathbf{y}\|, \tag{6}$$

the skeleton S_Ω of Ω is next defined as

$$S_\Omega = \{\mathbf{x} \in \Omega | \exists \mathbf{f}_1, \mathbf{f}_2 \in \partial\Omega, \mathbf{f}_1 \neq \mathbf{f}_2, \|\mathbf{x} - \mathbf{f}_1\| = \|\mathbf{x} - \mathbf{f}_2\| = DT_{\partial\Omega}(\mathbf{x})\} \tag{7}$$

where $\mathbf{f}_1$ and $\mathbf{f}_2$ are the contact points with $\partial\Omega$ of the maximally inscribed disc in Ω centered at $\mathbf{x}$. From $S_{\Omega_{oc}}$, the algorithm removes branch fragments that overlap with Ω, yielding a set $F = S_{\Omega_{oc}} \setminus \Omega$ that contains skeleton-fragments located in thin *gaps* that cut *deeply* inside 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 [34], this produces an accurate identification of deep-and-thin indentations, or *gaps*, in Ω, while ignoring pixels in shallow dents along $\partial\Omega$.

Hair-detection: 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 [34] are needed. First, we note that [34] uses $DT_{\Omega_{co}}$ as disk-radius values for gap-filling as they argue that Ω_{co} closes more gaps than Ω_{oc}, supported by the observation that $DT_{\Omega_{co}}(\mathbf{x}) \geq DT_{\Omega_{oc}}(\mathbf{x}), \forall \mathbf{x} \in F$. 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 $DT_{\partial\Omega_{co}}$. For this, we define a set of pairs disk-centers $\mathbf{x}$ and corresponding disk-radii ρ as

$$D_\lambda = \{(\mathbf{x}, \rho = (1 - \lambda) DT_{\partial\Omega_{co}}(\mathbf{x}) + \lambda DT_{\partial\Omega_{oc}}(\mathbf{x})) \, | \mathbf{x} \in F\} \tag{8}$$

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

$$G = D \setminus \Omega. \tag{9}$$

We apply Eqn. 9 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 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 search for gaps on *every* threshold-level, we find more gaps than traditional approaches, *e.g.* [16, 18, 24, 39]. Filtering out ‘false positives’ (gaps unlikely to be hairs), is thus necessary. We achieve this in four steps, outlined below.

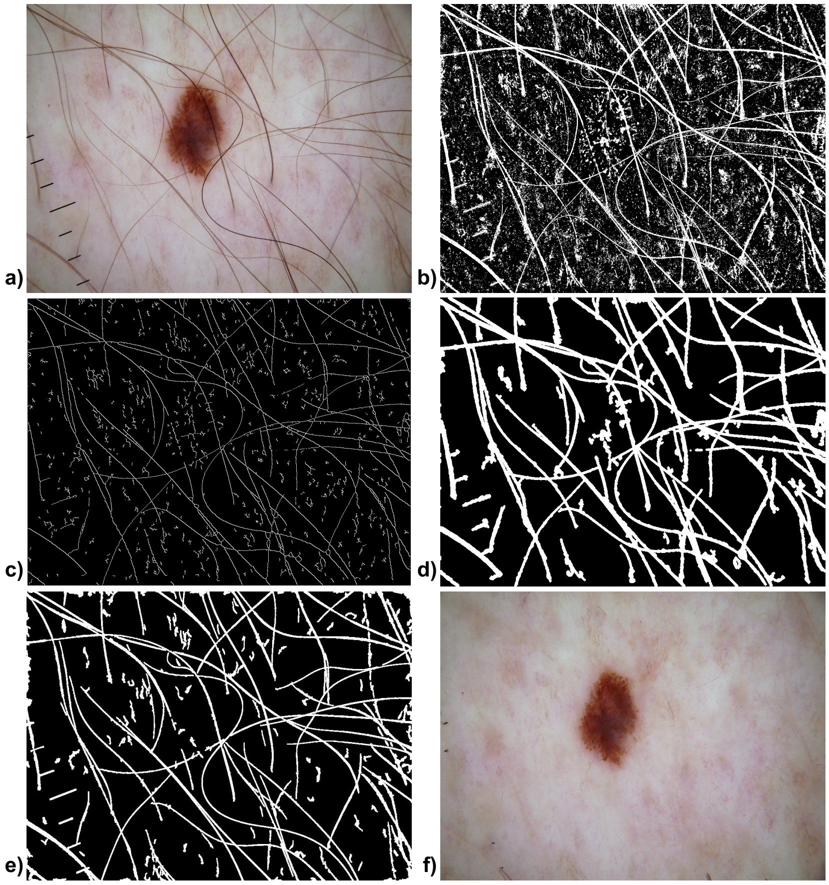

Fig. 1. a) Input image. b) Full hair mask M. c) Simplified mask skeleton S^{τ}_{M}. d) Filtered mask M^f. e) Mask created by [16]. f) Inpainted hair using M^f.

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 low resolution (Fig. 1 b), 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 [37]: 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 τ. By making τ 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 τ_{min} to discard tiny spurious fragments, and a maximum τ_{max} to preserve large branches. Hence, the pruning parameter τ for a component C_i is

$$\tau = \max(\tau_{min}, \min(\|\partial C_i\| \cdot \mu, \tau_{max})) \tag{10}$$

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

Hair detection: In classical 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 [24]; (b) using principal component analysis to find if the major-to-minor eigenvalue ratio exceeds a threshold [23]; and (c) computing an elongation metric comparing a component's skeleton-length with its area [39]. Xie *et al.* argue that (a) and (b) are limited, as they favor mainly straight hairs and yield false-negatives for curled hairs [39]. 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. However, 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 $J_i = \{\mathbf{x} \in S^{\tau}_{\partial C_i}\}$ 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 $d_{max} = \max_{\mathbf{x} \in J_i, \mathbf{y} \in J_i, \mathbf{x} \neq \mathbf{y}} \|\mathbf{x} - \mathbf{y}\|$ 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 to be a thin elongated hair (or a few crossing hairs), so we erase $S^{\tau}_{\partial C_i}$ from the skeleton image. Good preset 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 [16] (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 Long Hair Removal

We remove the detected thin-and-long hairs by using classical inpainting [36] 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 Fig. 1 b. Figure 1 f shows our final DHR result.

3.5 Stubble Detection and Removal

While the above four steps effectively find and remove thin-and-long hairs, they can easily miss thick-and-short hairs (stubble). Such hairs appear in dermoscopy images, *e.g.* in situations where the lesion area was shaved for a better image acquisition. To remove stubble, we propose a post-processing filter on the images generated by the inpainting step (Sec. 3.4), as follows.

Let I^{inp} be the output of the long-and-thin hair inpainting step (note that this is a color image). We compute I_{OC} and I_{CO} by applying open-close and close-open operators respectively to the red, green, and blue channels of I^{inp}. We next compute the absolute difference (grayscale) images $I^d_{OC,I}$ and $I^d_{CO,I}$ of I_{OC} and I_{CO} respectively with the input image I^{inp}. This is related, but not identical to, the top-hat and bottom-hat transforms, where the difference between an image and its opening, respectively closing, is taken. As shown in [34], using the open-close and close-open images instead of basic openings and closings yields better results for gap detection scenarios like our DHR context.

Similarly to the mask construction for long hair detection in the presence of hairs darker, respectively lighter, than skin, we next choose to use the difference image $I^d \in \{I^d_{OC,I}, I^d_{CO,I}\}$ which has the largest intensity value summed over its pixels. This selects $I^d := I^d_{OC,I}$ for images having predominantly dark stubble, and $I^d_{CO,I}$ for images having predominantly light-colored stubble.

We next threshold I^d into a binary stubble mask M^s by using a threshold value defined as

$$t = \frac{\max_{\mathbf{x} \in I^d} I^d(\mathbf{x})}{\gamma}, \tag{11}$$

where γ is a scaling factor. Setting $\gamma = 2$ reliably selected stubble hair in all out test images. After thresholding, we normalize the resulting image to $[0, 255]$. This has the effect of a contrast enhancement

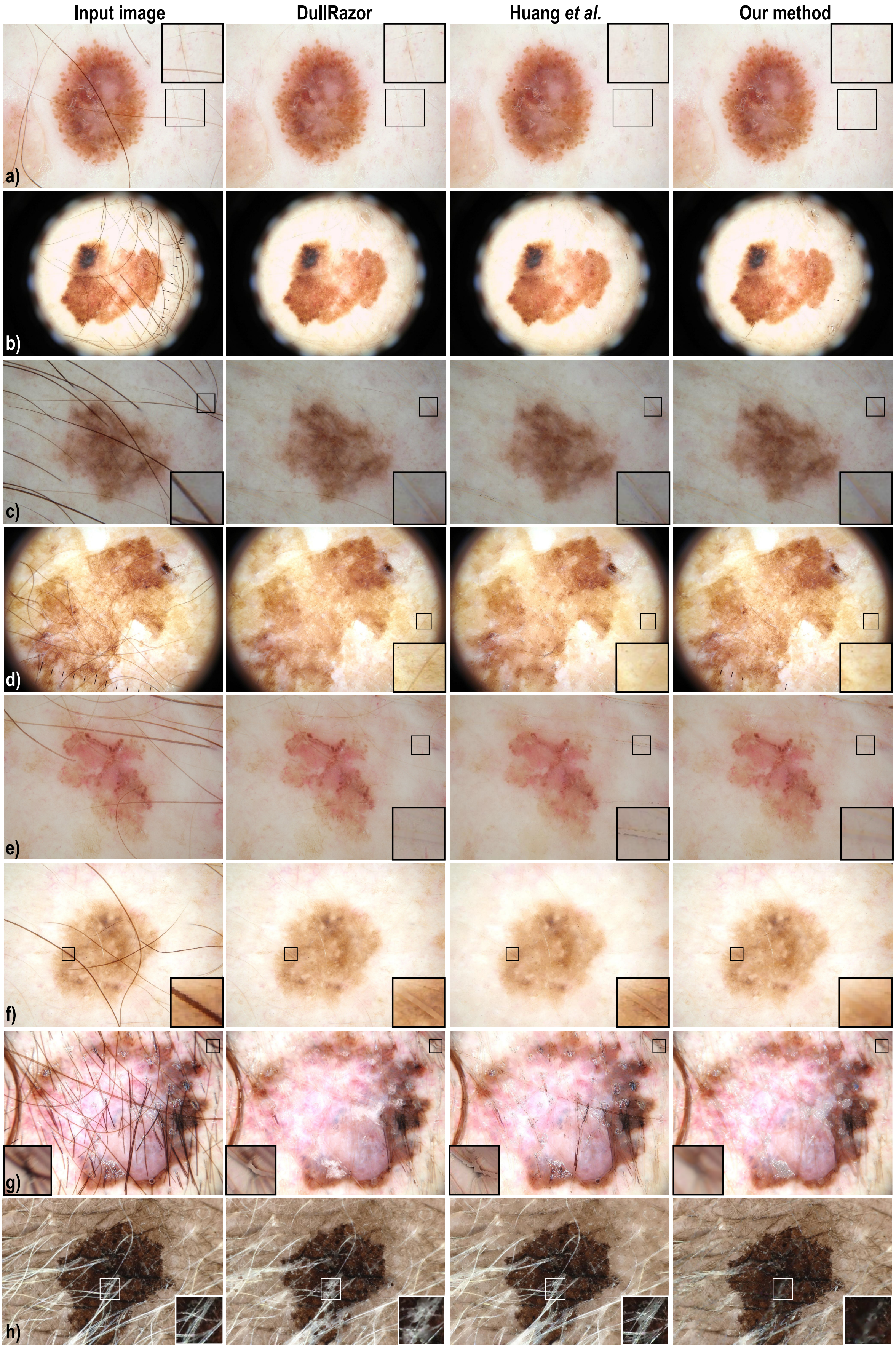

Fig. 2. Comparison of our method with DullRazor [23] and Huang *et al.* [16]. Insets show details.

operation, which makes low-contrast hairs more visible and thus selects them more reliably in the mask M^s. We finally dilate M^s isotropically by a 3×3 square structuring element, and remove stubble from I^{inp} by inpainting it over M^s, analogously to the long-and-thin hair removal (Sec. 3.4).

Figure 3 shows the effects of our stubble removal filter. As visible, stubble is still present in the output of the long-and-thin DHR algorithm pass (Figs. 3 a,c), while it is well detected and removed by our stubble removal filter (Figs. 3 b,d). The stubble filter also removes other small-scale line-like details, such as the ruler annotations introduced by the dermatoscope (Figs. 3 a,c top-row). The two filters (long-and-thin and stubble removal) assist each other, as follows. If an image contains only thin-and-long hair, or only stubble, only one of the filters will actively change the image, while the other one will act as a pass-through. If, however, an image contains both hair types, applying the thin-and-long hair filter *before* stubble removal has the desirable effect of making stubble detection much easier, as complicated structures of entangled hair are already removed.

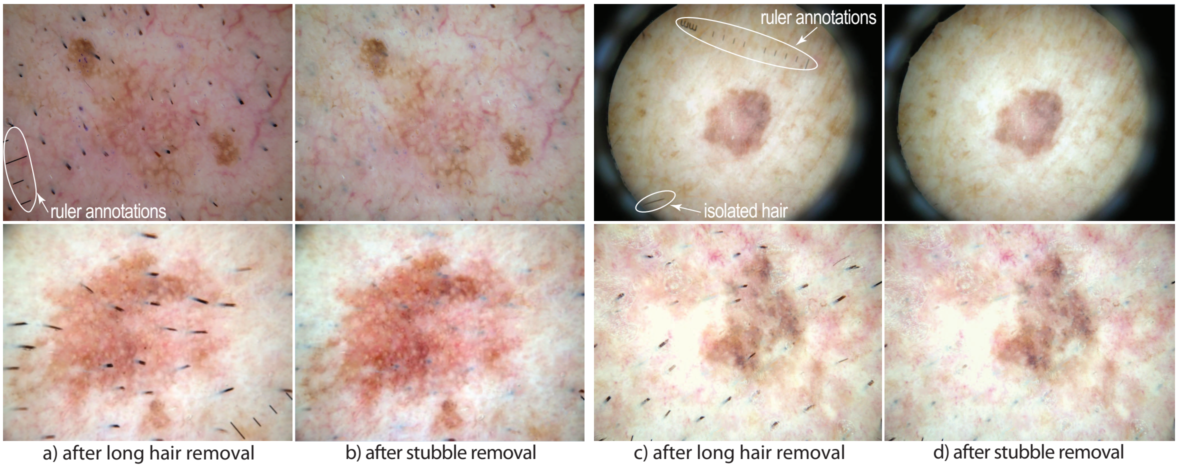

Fig. 3. Stubble removal filter (b,d) filters out stubble from the output of the long-hair removal pass (a,b).

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 [15], processing a single image must be done within milliseconds to yield an acceptable speed. For this, we use the Parallel Banding Algorithm (PBA) for exact Euclidean distance transforms (EDTs) 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 [37]). Computing the skeleton of a shape Ω by [37] only requires the identity of the closest point of $\partial\Omega$ for any point in Ω, or the so-called feature transform of $\partial\Omega$. This information is directly provided by the PBA method, so computing skeletons has virtually no additional cost atop of the distance computation.

Hair masks M (Sec. 3.2) are also computed on the GPU. First, the grayscale image is copied from CPU memory to VRAM, after which each threshold is processed sequentially on the GPU. For each threshold i, the open-close and close-open images are computed from the binary shape Ω_i. Erosions and dilations are computed by thresholding the distance transforms $DT_{\partial\Omega_i}$ and $DT_{\partial\bar{\Omega}_i}$ with the radius of the desired disk structuring element. Open-close images are computed by optimizing $\Omega \oplus H \ominus H \ominus H \oplus H$ into $\Omega \oplus H \ominus H' \oplus H$, where H' has double the radius of H. Similar optimizations are done for close-open.

The distance transforms $DT_{\partial\Omega_i}$ and $DT_{\partial\bar{\Omega}_i}$ are subsequently used to compute the radii ρ of the disks D_λ (Eqn. 8). Next, for each skeleton pixel $\mathbf{x}$ located in a gap (set F in Sec. 3.2), we launch a thread to draw a disk of radius ρ centered at $\mathbf{x}$, which yields the image D. As F does not contain many pixels (hundreds at most), computing D by disk drawing is efficient. The final step in processing a layer is to compute the gap mask G_i by finding all disk pixels outside Ω_i and marking their locations directly in the hair mask M.

After all layers have been processed, the hair mask M is copied from VRAM back to CPU memory. The latter steps of the algorithm – connected component detection, done with union-find [29]; skeleton-based filtering; stubble filtering; and hair inpainting [36] – are implemented in C++ on the CPU, as they are only performed once and thus not performance-critical as the per-layer computations are.

We also ran our method on multi-GPU machines by starting k MPI processes for k GPUs. Each process $p \in \{0, \ldots, k\}$ 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 the algorithm continues on the CPU like outlined above.

Memory-wise, our entire implementation requires only 12 floating-point buffers of the size of the input image I, seven by PBA [6] to compute EDTs and skeletons, and five for the remaining algorithm steps. This allows processing megapixel-size images on even the lowest-range CUDA-capable GPUs having 128 MB VRAM. Feature-wise, we only use CUDA 1.1 capabilities, which makes our implementation run on virtually all existing Nvidia cards, including low-end ones. C++ source code of our full method is available openly for download at [20]. Additional details regarding computational speed are given in Sec. 6.

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, and density; image resolution (between 400^2 and 2448×3264 pixels *i.e.* full Handyscope resolution [15]); and skin pigmentation. Images were acquired by several types of dermoscopes, by three unrelated research groups. Additionally, we tested our DHR method on the skin images reported in the papers of [2, 13, 16]. Some of our test images contain no hair (see *e.g.* Fig. 6c discussed further in this section); they let us see how well can we avoid false positives. This is important, as removing non-hair details may affect subsequent analyses [2, 16].

Methods: We compared our results with five DHR methods, as follows: Where an implementation of the method to compare with was available [16, 24], we ran our full image-set through it. For the other methods [2, 13, 39], we processed images from the respective papers by our method and compared our results with the ones in the respective papers.

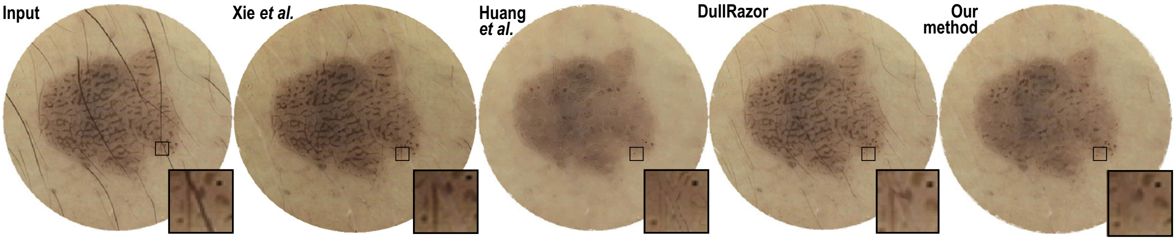

Fig. 4. Comparison between Xie *et al.* [39], Huang *et al.* [16], DullRazor, and our method. Input image from Xie *et al.*

Results: Compared to DullRazor and Huang *et al.* [16] (Fig. 2), we see that DullRazor cannot remove low-contrast hairs (a,d); and both methods create undesired '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

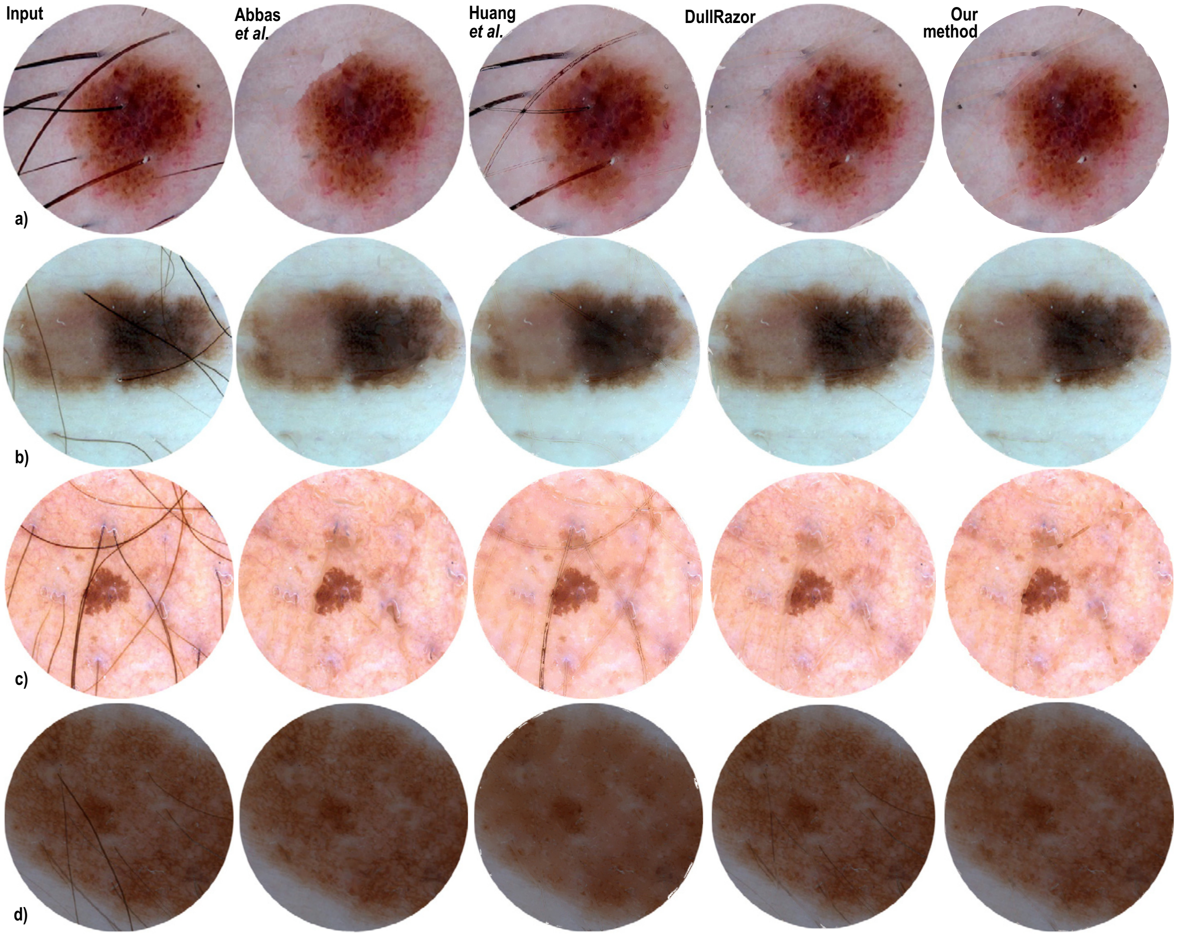

Fig. 5. Comparison between Abbas *et al.* [2], Huang *et al.* [16], DullRazor, and our method. Input images from Abbas *et al.*

removes considerably more hair than both methods. Figure 6 shows supplementary comparisons for four complex images. Image (a) contains several crossing and very low-contrast hairs. We see that DullRazor can remove several, but not all, such hairs. Also, both DullRazor and Huang *et al.* create high-contrast edges from small non-hair pigmentation details, such as the ones shown in the insets, an effect of their use of local edge-detection filters. In contrast, our method removes most such hairs and also correctly preserves pigmentation details. Image (b) contains a few hair-like details (dermoscope markers in top-left corner), but no hairs. The markers are successfully removed by all methods, including ours. Image (c) shows a few crossing very low-contrast hairs. DullRazor cannot remove these. Huang *et al.* remove them, but also significantly blurs the skin line-like pattern. Our method removes the hairs and keeps the skin pattern, since its line-like structures are not sufficiently long to be seen as hairs by our skeleton-based analysis (Sec. 3.3). Finally, image (d) contains no hairs, but a number of bubbles formed by contact gel placed between the dermoscope lens and the skin for better contact (see inset). Like hairs, such artifacts are not part of the tumor texture proper, and can confuse subsequent image analyses, and as such should be removed, if possible. We see that both DullRazor and Huang *et al.* cannot remove these structures. In contrast, our method detects the thin-and-curly bubble structure and removes most of it.

Figure 4 compares our results with Xie *et al.* [39] 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 5 compares our method with Abbas *et al.* [2], Huang *et al.*, and DullRazor, on a set of images from Abbas *et al.* These images cover a wide gamut of skin and lesion pigmentations and hair thicknesses and contrasts. Our method shows comparable results to Abbas *et al.* Huang *et al.* has issues with thick

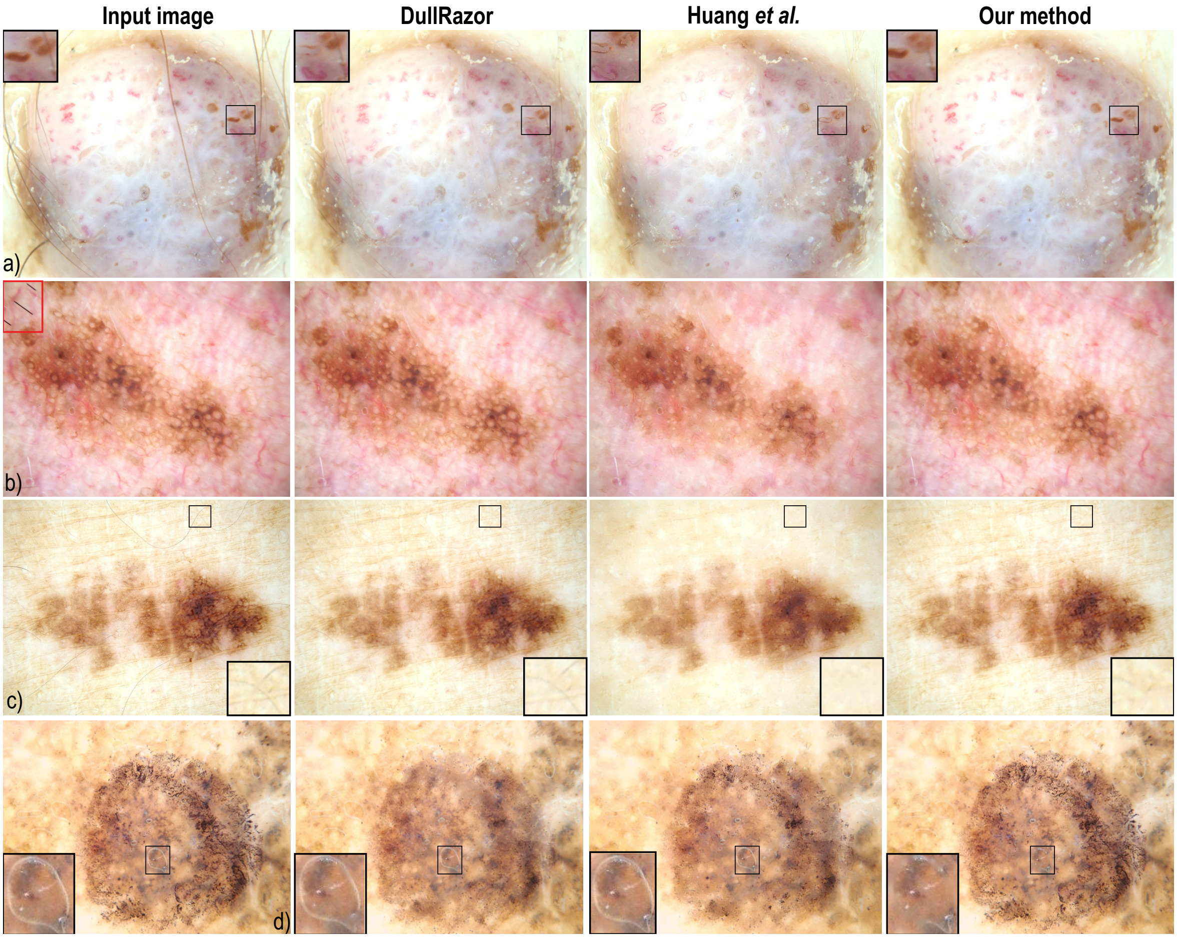

Fig. 6. Comparison of our method with DullRazor [23] and Huang *et al.* [16] for several complex skin images.

hairs (a); creates undesired hair halos (c); and also blurs the fine-grained typical network texture present in image (d). This last effect is highly undesired, since typical network texture is, among other image features, an important indicator for the malignancy assessment of skin tumors [22]. Separately, we see that DullRazor cannot remove most of the low-contrast hairs for the dark lesion (d).

Compared to Fiorese *et al.* [13], we show a similar ability in removing both stubble and elongated hairs (Fig. 7). For images (a,b), Fiorese *et al.* strikingly changes the hue of the input image, which is undesired, as this can affect both manual and automatic lesion assessment. Our method correctly preserves the hue of the image. For the same images, showing both stubble hair (Fig. 7 a) and long curly hair (Fig. 7 b,c), our method performs very similarly to DullRazor and Huang *et al.*, and also creates less halos around removed hairs (see insets).

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.

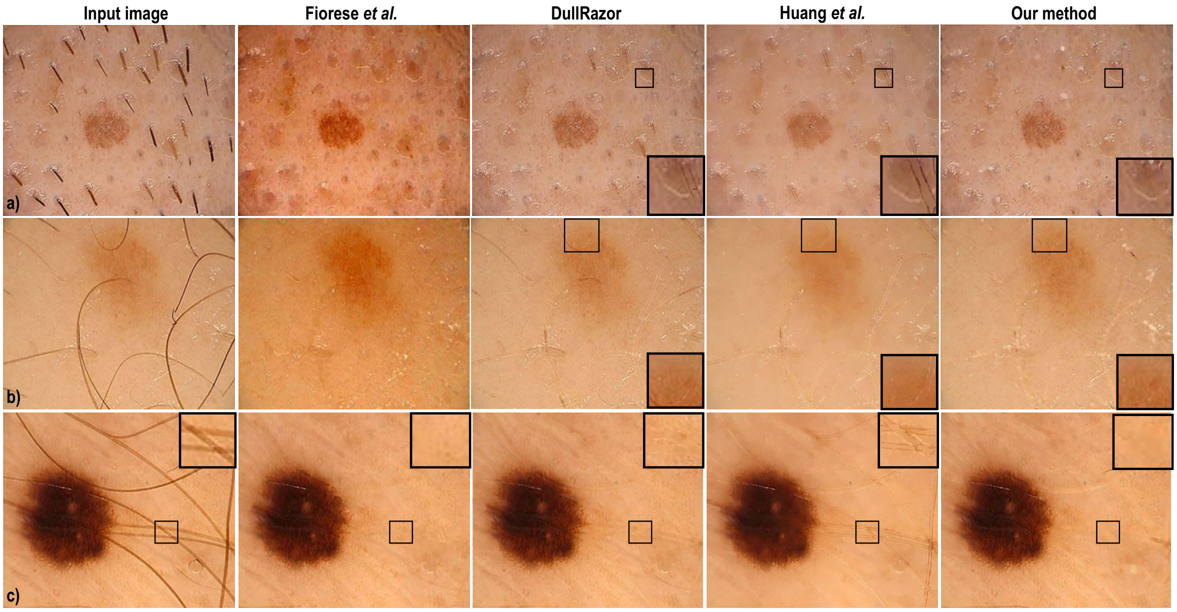

Fig. 7. Comparison of Fiorese *et al.* [13], Huang *et al.* [16], DullRazor, and our method. Input images from Fiorese *et al.*

Table 2. Empirically established parameter values.

	Description	Definition	Value
H	Structuring element radius	Section 3.2	5.0 pixels
λ	Gap detection parameter	Equation 8	0.2
μ	Skeleton simplification parameter	Equation 10	0.05
τ_{min}	Minimum skeleton pruning	Equation 10	3.0 pixels
τ_{max}	Maximum skeleton pruning	Equation 10	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

Robustness: We reliably remove hairs regardless of thickness, curvature, length, color, or underlying skin pattern. Very thin and low-contrast hairs 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, the fact that such hairs are not detected (and thus not removed) does not affect further usage of the skin images, since they are almost invisible in the first place.

Speed: We compute an open-close, a close-open, a skeletonization, and a skeleton-to-shape reconstruction step for all threshold layers T_i found in an image. For a 1024^2 pixel image densely populated by hairs, 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. As such, our method is the third-fastest from the set of methods we compared against.

The complexity of our method is $O(\|T\| \cdot \|I\|)$, *i.e.*, it is linear in the size of the input image I and the number of threshold layers that we decompose I into (Sec. 3.1). This is due to the fact that all core operations in our pipeline (morphological filters, inpainting, distance transforms, and skeletonization) are linear in the number of processed pixels, and we process $\|T\|$ such images, one for each threshold layer. We next analyze how our implementation scales with respect to the number of threshold layers $\|T\|$, as this is the parameter that dominates the processing time. For this, we fix the input image resolution at 1024 × 768, and run our DHR method on 100 images which have a wide variation of the remaining parameters (type and density of hairs and skin color). Figure 8 (left) shows the measured execution timings *vs* the number of different threshold images $\|T\|$ found in each input image. We notice a good linear correlation of the execution time with number of thresholds. Note that a maximum of

512 threshold images are being processed, as we compute two masks M^l and M^d, and each mask is determined by maximally $2^8 = 256$ thresholds.

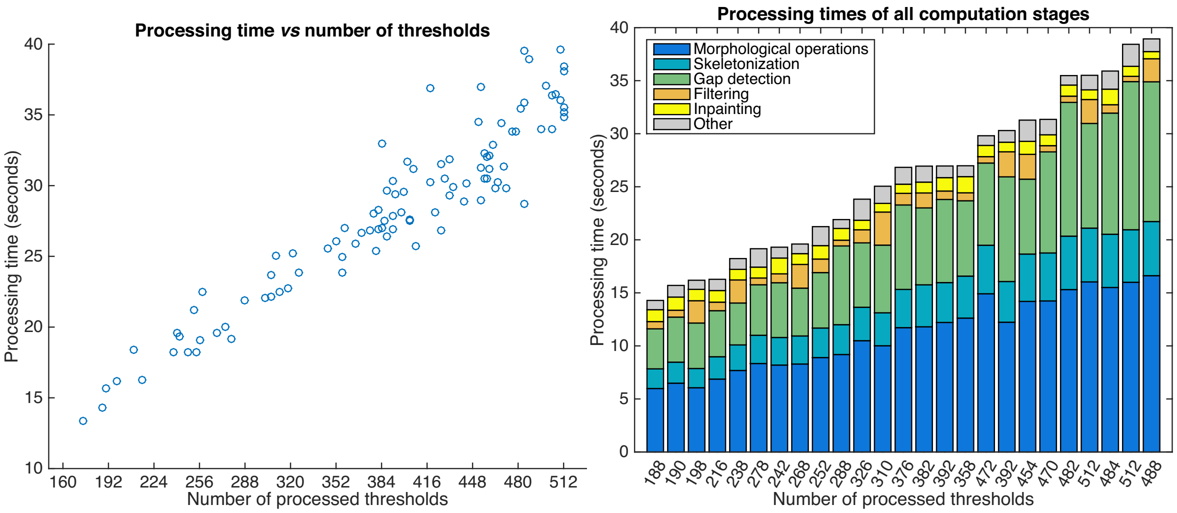

Fig. 8. Left: Total processing time as a function of $\|T\|$. Right: Relative cost of computation stages, sorted on total time.

Figure 8 (right) shows the distribution of relative costs of the various stages of our pipeline, for the same set of images as in Fig. 8 (left). Several points can be made here, as follows. First, we note that the *relative* costs of all stages are largely independent on the number of processed thresholds – or, in other words, that the total cost is indeed dominated by the number $\|T\|$ of processed threshold images. Secondly, we note that total cost is dominated by morphological operations (opening and closing) and the gap-detection (computation of image D by disk drawing, see Sec. 4). Interestingly, computing exact Euclidean skeletons, which is often perceived as an expensive operation, accounts for only 10 up to 15% of the total processing time. Inpainting also has a very low cost, which justifies our implementation thereof on the CPU. Overall, this analysis tells that significant speed-ups can be obtained by optimizing our implementation of the morphological operations and disk-drawing used to detect the hair gaps.

Per image threshold, we obtained an average processing time $\bar{\tau} \in [60, 90]$ milliseconds, following a distribution with mean 73.8 and standard deviation of 31.7 respectively. Furthermore, we tested for correlation of $\bar{\tau}$ with various image features such as the amount of hair pixels detected in an image, amount of hair crossings, and average hair length. No significant correlations were found. This strengthens the earlier observation that our method's throughput is dominated by number of processed image thresholds (for a given image resolution) and not by the type and/or amount of hair to remove. In turn, this indicates that, if our current morphological operations and gap detection implementations were further optimized, significant performance can be consistently gained. Separately, this tells that our method can be trivially accelerated by using newer GPUs that offer more processing cores.

Tumor segmentation use-case: A practical way to measure the quality (and usefulness) of our DHR method is to see how different the results of *tumor segmentation* are for images with hair and with hairs removed by our method. Tumor segmentation is a crucial step in the computation of image descriptors used for skin lesion classification, since such descriptors need to be assessed only over the lesion area and not over surrounding healthy skin [7, 22, 27, 33].

To assess this difference, we considered several skin images having high-contrast hairs. Such hairs adversely influence most automatic segmentation methods that try to separate the tumor from surrounding skin (see *e.g.* [35], Sec. 9.4.2, Fig. 9.9). We considered next two segmentation methods which are applicable to skin tumors: superpixel graphs based on the image foresting transform [31] and the more specific normalized-cut method in [14], which claims to be robust for skin lesions occluded by hairs. We also tried other known segmentation methods, such as the active contour approach used in skin tumor segmentation in [27], the mean shift method [10], and the level-set approach in [25]. However, these additional methods showed much larger sensitivity to input image characteristics, including hairs but also lesion details, as compared to [31] and [14]. As such, we deemed them less suitable candidates for skin segmentation in general, and eliminated them from further detailed inspection.

Focusing on the two most robust segmentation methods of hair-occluded tumors [14, 31], we see that both methods still have significant problems for images containing long high-contrast hairs. These problems manifest themselves in terms of creating segments which either contain large parts of skin outside the lesion or have boundaries that follow hairs that intersect the lesion (Fig. 9 b,c, red markers). Such suboptimal segmentations create *major* problems for *e.g.* the computation of reliable image descriptors that should characterize the precisely delimited tumor area, to be used in automatic lesion classification [7, 33]. After removing hairs by our DHR method (Fig. 9 d), both considered segmentation methods achieve a very good segmentation result that closely follows the apparent skin tumor boundary without being distracted by crossing hairs (Fig. 9 e,f). This shows that our DHR method can be used as an automatic preprocessing filter for robust skin-tumor segmentation in tumor classification pipelines.

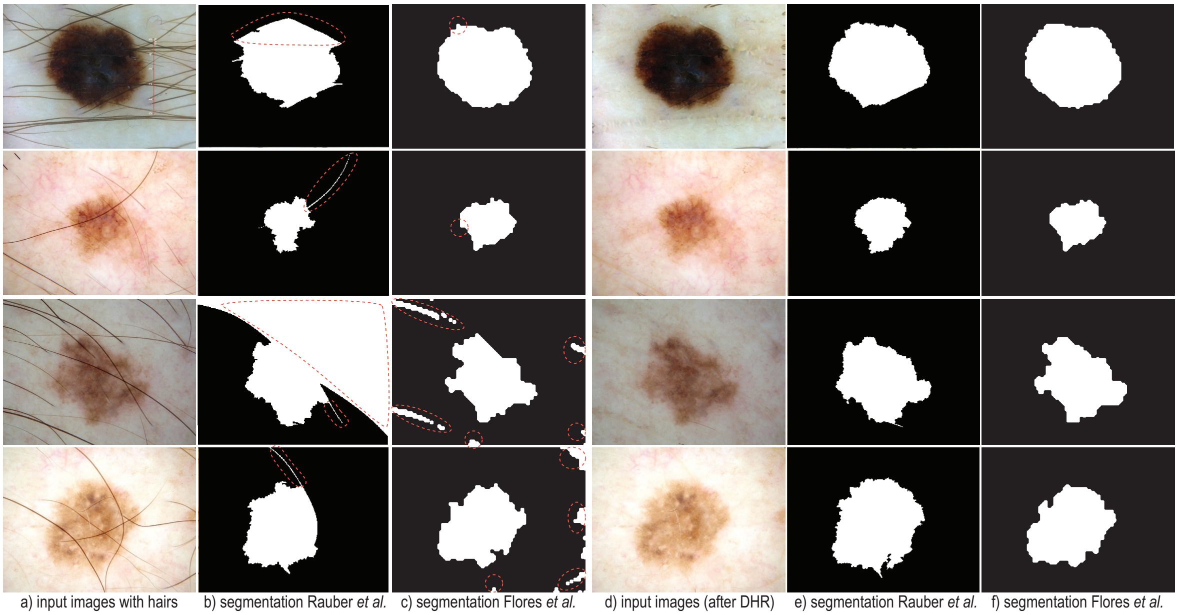
a) input images with hairs
b) segmentation Rauber *et al.*
c) segmentation Flores *et al.*
d) input images (after DHR)
e) segmentation Rauber *et al.*
f) segmentation Flores *et al.*

Fig. 9. Tumor segmentation: (a) Input images with hairs, and corresponding segmentations using the methods of (b) Rauber *et al.* [31] and (c) Flores *et al.* [14]. Segmentation artifacts are marked in red. (d-f) Results of segmenting the same images with the same methods after hair removal.

Qualitative validation: We have shown all our input images, and obtained DHR results, to two dermatologists having over 11 years of clinical experience, in blind mode – that is, the two specialists did not know of each other's assessment results, nor did they know about the aims of the evaluation or specifics of the DHR method being used to remove hairs. We asked whether the raw *vs* DHR-processed images would lead them to different interpretations, diagnoses, or insights. For all images, the answer was negative. While a more formal measurement would bring additional insights, this test already tells that our DHR method does not change the images in *undesirable* ways from the perspective of specialist users who assess them. Separately, hair removal is obviously *desirable*, *e.g.* when using images in automated image analysis and classification procedures [2, 16, 27, 30], such as the tumor segmentation use-case discussed above.

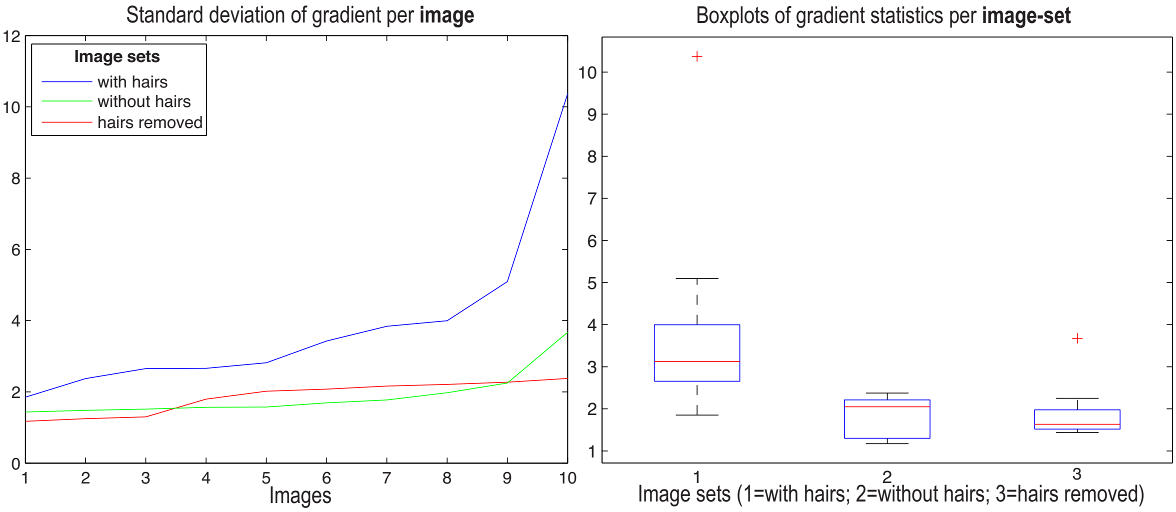

Fig. 10. Left: Standard deviation of per-image gradients in three considered image-sets. Right: Minimum, maximum, average, and standard deviation, per-image-set, of image gradients.

Quantitative validation: To quantitatively assess the effect of hair removal, we performed the following experiment. We created three image databases, each having 10 different images (within each database and across databases), all images having the same resolution. Database 1 contains skin-tumor images without occluding hairs. Database 2 contains images with significant amounts of occluding hairs. Database 3 contains images created by our DHR method by removing hairs from a set of images with occluding hairs (not present in database 2). For each image in the three databases, we next computed the gradient magnitude at each pixel and its standard deviation over each image. Separately, we computed the standard deviation of the per-image gradients over all images in each database. These metrics give an overall characterization of image features relevant for a wide range of tasks such as image classification [3, 11, 21], since most image descriptors such as color histograms, tumor boundaries, edge histograms, and texture descriptors used by such techniques strongly depend on local image gradients [22, 33]. Plotting the gradient standard deviation for images in all three databases, we see that the DHR images are very similar to the *different* hair-free images, while the hair-occluded images clearly stand apart (Fig. 10). This supports the hypothesis that our DHR method creates, on average, images which have the same statistical characteristics as hair-free images, *i.e.* images which could be used with the same success as hair-free images in various automatic analyses. While more accurate comparisons of the actual image features extracted from the raw *vs* DHR-processed images could be performed, such as considering texture descriptors, this simple test already indicates a good statistical match between our results and typical naturally hair-free images.

Limitations: For very dense hairs of varying color on high-contrast skin (*e.g.* Fig. 2 h), we cannot fully remove all hairs. Yet, this image type is extremely atypical – it actually is a skin lesion of a Labrador canine subject, which has massively more hairs than typical humans; whose hairs are significantly thicker than human hair, half-transparent, and hollow; and whose underlying skin texture shows complex high-contrast striations. Also, other methods [16, 24] remove significantly less hairs in such cases. Separately, while our method's speed is around the average of the tested competitors, faster (albeit lower-quality) methods exist [13, 24]. Using a more conservative method to select a subset of layers from the entire threshold set of 256 binary images to further process to detect hairs, in line with similar layer-selection procedures used for image compression [40], would accelerate our method up to one order of magnitude. Indeed, as discussed earlier in this section, our speed is chiefly influenced by the number of processed layers. This would make our approach (compete with) the fastest DHR method published so far.

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. In this respect, our method can better remove long curly hair and short stubble hair than its competitors. We show how our method effectively improves skin tumor segmentation in the case of hair-occluded tumors, an important asset for automatic skin lesion processing. Performance analysis of our method show its linear dependence on the input image size and number of threshold sets identified in the image. Qualitative and quantitative validations support the claim that our method produces images which are perceptually and also quantitatively very similar to the original hair-occluded images.

Future work can target several directions. Machine learning techniques [3, 11, 21] could be used to 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. Application-wise, our method can be straightforwardly incorporated into skin tumor classifiers for melanoma detection in order to make such techniques directly applicable to hair-occluded images too.

Acknowledgments

This work was funded by the grants 202535/2011-8 (CNPq, Brazil) and PN-II RU-TE 2011-3-0249 (CNCS, Romania). We acknowledge the author Eliezer Flores of [14] for testing their skin-tumor segmentation method upon our image collection.

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. Biomed Signal Proc 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] Celebi, M., Kingravi, H., Uddin, B., Iyatomi, H., Aslandogan, A., Stoecker, W., Moss, R.: A methodological approach to the classification of dermoscopy images. Comput Med Imaging Graph 31(6), 362–373 (2007)

[8] 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)

[9] Cokelaer, F., Talbot, H., Chanussot, J.: Efficient robust d-dimensional path operators. IEEE J. Selected Topics in Signal Processing 6(7), 830–839 (2012)

[10] Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE TPAMI 24(5), 603–619 (2002)

[11] Cortes, C., Vapnik, V.: Support-vector networks. Mach Learn 20(3), 273–297 (1995)

[12] Couprie, M., Bezerra, F.N., Bertrand, G.: Topological operators for grayscale image processing. J. Electronic Imag. 10(4), 1003–1015 (2001)

[13] Fiorese, M., Peserico, E., Silletti, A.: VirtualShave: automated hair removal from digital dermatoscopic images. In: Proc. IEEE EMBS. pp. 5145–5148 (2011)

[14] Flores, E., Scharcanski, J.: Segmentation of pigmented melanocytic skin lesions based on learned dictionaries and normalized graph cuts. In: Graphics, Patterns and Images (SIBGRAPI), 2014 27th SIBGRAPI Conference on. pp. 33–40 (Aug 2014)

[15] FotoFinder: Handyscope mobile dermatoscope specifications (2015), www.handyscope.net

[16] 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)

[17] 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)

[18] Kiani, K., Sharafat, A.: E-shaver: An improved dullrazor for digitally removing dark and light-colored hairs in dermoscopic images. Comput Biol Med 41(3), 139–145 (2011)

[19] Koehoorn, J., Sobiecki, A., Boda, D., Diaconeasa, A., Doshi, S., Paisey, S., Jalba, A., Telea, A.: Automated digital hair removal by threshold decomposition and morphological analysis. In: Proc. ISMM. pp. 324–335 (2015)

[20] 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

[21] Kohonen, T.: Learning vector quantization. In: Self-Organizing Maps, pp. 203–217. Springer (1997)

[22] Korotkov, K., Garcia, R.: Methodological review: Computerized analysis of pigmented skin lesions: A review. Artif. Intell. Med. 56(2), 69–90 (2012)

[23] Lee, H.Y., Lee, H.K., Kim, T., Park, W.: Towards knowledge-based extraction of roads from 1m-resolution satellite images. In: Proc. SSIAI. pp. 171–178 (2000)

[24] 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)

[25] Li, C., Xu, C., Gui, C., Fox, M.D.: Distance regularized level set evolution and its application to image segmentation. IEEE TPAMI 19(12), 3243–3254 (2010)

[26] 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)

[27] Parolin, A., Herzer, E., Jung, C.: Semi-automated diagnosis of melanoma through the analysis of dermatological images. In: Proc. SIBGRAPI. pp. 71–78. IEEE Press (2010)

[28] Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE TPAMI 12(7), 629–639 (1990)

[29] Rahimi, A.: Fast connected components on images. alumni.media.mit.edu/~rahimi/connected (2014)

[30] Rauber, P., da Silva, R., Feringa, S., Celebi, M., Falcao, A., Telea, A.: Interactive image feature selection aided by dimensionality reduction. In: Proc. EuroVA. pp. 322–328 (2015)

[31] Rauber, P., Falcao, A., Spina, T., De Rezende, P.: Interactive segmentation by image foresting transform on superpixel graphs. In: Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI - Conference on. pp. 131–138 (Aug 2013)

[32] Saugeon, P., Guillod, J., Thiran, J.: Towards a computer-aided diagnosis system for pigmented skin lesions. Comput. Med. Imag. Grap. 27, 65–78 (2003)

[33] Scharkanski, J., Celebi, M.: Computer Vision Techniques for the Diagnosis of Skin Cancer. Springer (2014)

[34] Sobiecki, A., Jalba, A., Boda, D., Diaconeasa, A., Telea, A.: Gap-sensitive segmentation and restoration of digital images. In: Proc. EG GVC. pp. 136–144 (2014)

[35] Telea, A.: Data Visualization – Principles and practice. CRC Press (2014), 2^{nd} edition

[36] Telea, A.: An image inpainting technique based on the fast marching method. J. Graphics, GPU, & Game Tools 9(1), 23–34 (2004)

[37] Telea, A., van Wijk, J.J.: An augmented fast marching method for computing skeletons and centerlines. In: Proc. VisSym. pp. 251–259 (2002)

[38] Wighton, P., Lee, T., Atkins, M.: Dermascopic hair disocclusion using inpainting. In: Proc. SPIE Med. Imaging. pp. 144–151 (2008)

[39] 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)

[40] van der Zwan, M., Meiburg, Y., Telea, A.: A dense medial descriptor for image analysis. In: Proc. VISAPP. pp. 285–293 (2013)