Gap-Sensitive Segmentation and Restoration of Digital Images

A. Sobiecki 1 and A. Jalba 2 and D. Boda 3 and A. Diaconeasa 3 and A. Telea 1,3

1 Johann Bernoulli Institute, University of Groningen, the Netherlands
2 Department of Mathematics and Computer Science, University of Eindhoven, the Netherlands
3 University of Medicine and Pharmacy “Carol Davila”, Bucharest, Romania

Abstract
Many methods exist for removing defects such as gaps, cracks, and disconnections from digital shapes. However, most such methods have several limitations, such as removing both erroneous and important shape details, or requiring non-trivial effort from the end user in the form of manual delineation or parameter setting. In this paper, we propose a technique for removing defects such as internal gaps and cracks from 2D and 3D digital shapes. For this, we first classify gaps as boundary detail (to be preserved) and interior errors (to be removed), based on a heuristic that uses the gap position with respect to the medial axis of the simplified shape. Next, we remove error gaps using an efficient distance-based filling. We illustrate our method on robust segmentation and hair removal tasks for skin imaging, and compare our results with a number of relevant techniques in this area.

1. Introduction
Reconstruction of shapes missing internal information serves a wide range of applications, such as repairing scans of deteriorated images by closing holes, improving shape recognition and shape matching, and connecting shapes that are broken into pieces [BBC*01, JT03, CPT04, CDD*04, Lie03, VCBS03]. Digital shapes missing internal information can often be filled with morphological operators as well as automatically or manually with inpainting techniques. However, morphological operations cannot discriminate between locally identical, but globally different, details, such as gaps close or on the shape boundary (which should not be filled, if we want to preserve boundary detail), and gaps deeper in the shape (which may need to be filled). Separately, inpainting requires extra work to select the areas to be inpainted, which requires manual effort or more involved, and thus more sensitive, image-analysis algorithms [CPT04, Tel04, LNG*97].

We propose a technique to detect and reconstruct 2D and 3D digital shapes that lack some internal information, which we generically call ‘gaps’, while guaranteeing that detail information present on the apparent shape boundary is kept. For this, we first classify gaps into detail (that should be kept) and errors (that should be filled) using a global approach, based on the gap position with respect to the skeleton of a blurred version of the shape. Next, we fill error gaps using the medial axis transform associated to this skeleton. The method generically works for 2D and 3D digital shapes, using respectively 2D and 3D surface-skeletons, and is fast, simple to implement, and easy to use. We present applications for robust detail-preserving image segmentation and hair removal for dermatological images, and compare our method with several segmentation and restoration methods in the same field.

This paper is organized as follows. In Sec. 2, we review the related work. Section 3 presents our proposal. Section 4 presents 2D and 3D shape restoration examples. Section 5 presents an application of our method to the field of dermato-imaging. Section 6 discusses our method. Section 7 concludes the paper.

2. Related work
Many algorithms to segment and reconstruct digital shapes have been proposed. While an exhaustive review of the huge body of work on digital shape restoration is beyond our scope, we review three well-known approaches on segmentation and restoration of digital shapes which relate to our goals.

Filters: Filtering techniques like the median, mean, Laplacian [GL12], and morphological operators like erode, dilate, open, and close [HR98] can restore digital shapes by eliminating small-scale gaps, and are fast and simple to implement. However, most such filters work locally, so they cannot discriminate between gaps deep inside the shape (which we may want to eliminate) and gaps close or on the apparent shape boundary (which we want to keep, as they are part of the border structure).

Image segmentation: A key part of medical imaging is the segmentation of shapes from grayscale or color images. For example, in dermatology, one wants to segment tumors from surrounding healthy skin. Preserving all details on the segmentation border, and in the same time removing small-scale gaps and cracks inside the tumor, is essential for further automated analyses of the segmented image [DHR01, FRK85, PHJ10]. Several such segmentation methods exist [CM02, PHJ10, vdZMT13, FSL04]. However, as we shall see later in Sec. 5, none of these methods
A. Sobiecki & A. Jalba & D. Boda & A. Diaconeasa & A. Telea / Gap-Sensitive Segmentation and Restoration of Images

3. Our Proposal

Summarizing, our major goals are to (a) eliminate thin and long gaps that (nearly) disconnect a shape into several parts. We call these error gaps. In the same time, we want to (b) keep all details, including concave indentations or gaps, present on the shape’s apparent boundary.

For this, we propose a three-step process (see Fig. 1 top). Given a shape \( \Omega \subset \mathbb{R}^n \) with boundary \( \partial \Omega \), stored as a binary image (black=foreground, white=background), we first close all gaps of \( \Omega \) using morphological operations (Sec. 3.1). Next, we use the resulting image \( \Omega_{oc} \) to classify gaps into errors and details, using a topological analysis of \( \Omega_{oc} \) (Sec. 3.2). Finally, we use related topological mechanisms to fill gaps identified as errors in the previous step (Sec. 3.3). These three steps are detailed next.

3.1. Gap Closing

To close all gaps present in our input image \( \Omega \), we use classical morphological operations. In detail, given a so-called structuring element \( H \), we consider the dilation of \( \Omega \) by \( H \), i.e., the union of copies of \( H \), the element \( H \) centered at all pixels \( x \in \Omega \), i.e.,

\[
\Omega \circ H = \bigcup_{x \in \Omega} H_p.
\]

Similarly, we define the erosion of \( \Omega \) by \( H \), which keeps only pixels \( x \in \Omega \) where \( H \) fits inside \( \Omega \), i.e.,

\[
\Omega \setminus H = \{ x \in \Omega | H \subseteq \Omega \}.
\]

Next, we define the opening of \( \Omega \) as erosion followed by dilation, i.e.,

\[
\Omega \circ H = (\Omega \oplus H) \oplus H,
\]

and, analogously, the closing of \( \Omega \) as dilation followed by erosion, i.e.,

\[
\Omega \bullet H = (\Omega \oplus H) \ominus H.
\]

If we use a disk structuring element \( H \) of radius \( \rho \), the result of applying opening and closing, denoted \( \Omega_{oc} = (\Omega \circ \rho) \bullet \rho \), will close all holes in \( \Omega \) whose thickness is smaller than \( \rho \). Additionally, we denote the result of applying closing and opening, by \( \Omega_{oc} = (\Omega \bullet \rho) \circ \rho \). Both \( \Omega_{oc} \) and \( \Omega_{oc} \) will be used next for our error-hole removal, see Secs. 3.2 and 3.3.

3.2. Gap Classification

We now use the image \( \Omega_{oc} \) to classify holes into errors and detail. For this, we first compute the skeleton \( S(\Omega_{oc}) \). For this, we define the distance transform \( DT_{\Omega_{oc}} : \Omega \rightarrow \mathbb{R}^+ \) of any shape \( \Omega \) as

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

The skeleton \( S(\Omega) \), or medial axis, of \( \Omega \) is next defined as

\[
S(\Omega) = \{ x \in \Omega | \exists f_1, f_2 \in \partial \Omega, f_1 \neq f_2, \| x - f_1 \| = \| x - f_2 \| = DT_{\Omega_{oc}}(x) \}.
\]

Figure 1 b shows the shape \( \Omega_{oc} \) (in black) for our test image in Fig. 1 a, and the skeleton \( S(\Omega_{oc}) \) (in white) for the same shape.

We now compute the fragments \( F \) of the skeleton \( S(\Omega_{oc}) \) that fall outside our input shape \( \Omega \), i.e.,

\[
F = \{ x \in S(\Omega_{oc}) | x \notin \Omega \}.
\]

Next, we observe that points in \( F \) are inside the error gaps, but outside the detail gaps, of \( \Omega \). Let us explain this. As noted in Sec. 3.1, \( \Omega_{oc} \) closes both error and detail gaps of \( \Omega \), by construction. Additionally, \( \Omega_{oc} \) has a boundary that is smoother than \( \Omega \) (see Fig. 1 b). More precisely, all details on \( \partial \Omega \) whose curvature is larger than \( 1/\rho \) are replaced by the close operation (Eqn. 4) by circle arcs in 2D (and respectively spherical segments in 3D) of radius \( \rho \). We know that branches in \( S(\Omega_{oc}) \) correspond to curvature maxima on \( \partial \Omega_{oc} \) [SP09]. Since \( \partial \Omega_{oc} \) is smoother than \( \Omega \), it follows that branches of \( S(\Omega_{oc}) \), thus also points in \( F \), will never be located inside boundary gaps, or details, of \( \partial \Omega \), since (1) these correspond to curvature minima along \( \partial \Omega \), and (2) \( \Omega_{oc} \) has an absolute curvature smaller than \( \partial \Omega \). On the other hand, since the branches of \( S(\Omega_{oc}) \) are centered in the middle of the salient features of \( \Omega_{oc} \) (by the definition of the skeleton, Eqn. 6), they will also be centered in the middle of the corresponding salient features of \( \Omega \) (compare Figs. 1 b and a). This is so because the open-close operation that constructs \( \Omega_{oc} \) from \( \Omega \) uses a circular disk \( H \), so it does not modify the local shape symmetry. Overall, it follows that points in \( F \) will be located in gaps of \( \Omega \) which protrude deep inside this shape.

3.3. Error Gap Restoration

To close the error gaps identified by the skeleton subset \( F \), we next proceed as follows. For each point \( p \in F \), we find its closest skeleton point being in the input shape \( \Omega \)

\[
C(p) = \operatorname{argmin}_{q \in S(\Omega_{oc}) \cap \Omega} \| p - q \|
\]
and then draw a foreground-disk with radius \( DT_{\partial \Omega_c} (C(p)) \) centered at \( p \). This effectively fills the error gap containing \( p \) using the local shape thickness, which is equal to \( DT_{\partial \Omega_c} (C(p)) \). Let us explain this. First, we use the distance transform of the shape \( \Omega_{oc} \) (see Fig. 1 d) obtained by the close-open operation, rather than the distance transform of \( \Omega_c \), since the former first dilates, then erodes, the input shape. As such, \( \Omega_{co} \) closes gaps better than \( \Omega_{oc} \) (compare Fig. 1 c vs b). Thus, using \( DT_{\partial \Omega_c} \) gives a better estimate of the apparent (filled) shape boundary within gaps than \( DT_{\partial \Omega_{oc}} \). On the other hand, we use the skeleton of \( \Omega_{oc} \) to detect error gaps, and initiate reconstruction from, rather than the skeleton of \( \Omega_c \), since \( \Omega_{oc} \) does not close detail gaps (on the input boundary). If, in contrast, we used the skeleton of \( \Omega_{co} \), this skeleton would have branches that protrude outside \( \Omega \) in boundary areas, and thus using \( F \) defined by Eqn. 7 would fill both error and detail gaps, which is undesired. Figure 2 details the above decision for a simple rectangle shape cut half-way by a vertical gap (Fig. 2 a). Images (b) and (c) show the results of the open-close and close-open operations, respectively. As visible, the close-open operation better fills the gap. Image (d) shows the reconstruction result if we used \( DT_{\partial \Omega_{co}} \). As visible, the gap is not well filled, since \( \Omega_{oc} \) does not fill well the gap (image (b)). Image (e) shows our chosen reconstruction, where we use \( DT_{\partial \Omega_{co}} \). The skeleton \( S(\Omega_{oc}) \), drawn in red, is of course identical. However, the disks drawn atop of the skeleton fragments \( F \) are now larger, since \( \Omega_{oc} \) is larger than \( \Omega_{co} \), and thus, correspondingly, \( DT_{\partial \Omega_{co}} (x) \geq DT_{\partial \Omega_{oc}} (x) \), \( \forall x \in F \).

Secondly, we note that \( DT_{\partial \Omega_{oc}} (p) \) is typically smaller than \( DT_{\partial \Omega_{co}} (C(p)) \), due to the effect of the close-open operation sequence. Hence, the gap filling done by this operation tends to ‘shrink’ the filled shape towards the middle of the gap. Hence, using \( DT_{\partial \Omega_{co}} (C(p)) \) instead of \( DT_{\partial \Omega_{oc}} (p) \), fills the gap by using a value which is much closer to the real shape thickness, and thus leads to a smoother reconstruction of the boundary of the filled shape across the error gap.

By the above procedure, error gaps which intersect the skeleton \( S_{\Omega_c} \) are thus eliminated. Figure 1 e shows the reconstructed shape \( \Omega = \Omega \cup D \), where \( D \) is the set of pixels filled by the disk-drawing procedure outlined above. For clarity, we marked background pixels of \( \Omega \) as blue, foreground \( \Omega \) pixels as green, pixels in \( D \) as yellow, and pixels in \( F \) as red. Intuitively, our reconstruction procedures implies that gaps which cut deep inside \( \Omega \), to be precise more than half of the local thickness, get filled. In particular, gaps which completely disconnect (cut) \( \Omega \), but which are removed in \( \Omega_{oc} \) by the close operation, are guaranteed to be filled. In contrast, small superficial concavities or indentations of \( \partial \Omega \) that do not intersect \( S_{\Omega_c} \), i.e. are less deep than half the local thickness of \( \Omega \), are never eliminated. This way, concave boundary details of \( \Omega \) are kept (see insets in Fig. 1 e). Separately, note that the removal of convex details in \( \Omega_{oc} \) (as compared to \( \Omega_c \), see Fig. 1 b vs a), due to the open operation (Eqn. 3), does not adversely affect our final result. Indeed, our gap filling only adds foreground pixels to \( \Omega \), but never eliminates pixels from it (see again insets in Fig. 1 e).

### 3.4. Implementation

For 2D skeleton extracting, we use the AFMM method [TvW02], which computes robust, centered, pixel-wide, and topologically correct skeletons for 2D shapes of up to 10242 pixels in subsecond time on a modern PC. For 3D surface skeletons, we use the IMA method [HR09], which shares the same desirable properties. Both the AFMM and IMA implicitly compute, besides skeletons, the exact Euclidean distance transform \( DT_{\partial \Omega} \) of the input shape. This allows us to efficiently implement accurate dilation and erosion (Eqns. 1 and 2) by simply thresholding \( DT_{\partial \Omega} \) with the desired radius of the disk structuring element \( H \). Finally, we efficiently implement the disk-drawing filling in Sec. 3.3 by computing the distance transform \( DT_F \) of the set \( F \) and lower-thresholding it by the values \( DT_{\partial \Omega_{co}} (C(p)) \) for all points \( p \in F \). Both AFMM and IMA methods are implemented in C++, and do not use parallelization. Overall, on a commodity 3.5 GHz PC, our entire pipeline takes subsecond time for 2D images up to 30002 pixels and a few seconds for 3D volumes up to 4003 voxels.

### 4. Results

Figure 3 shows several 2D restoration examples for a set of synthetic shapes, on which gaps were created manually (a,d,j) or by luminance thresholding (g,m). As visible, our gap filling eliminates the complex internal gaps, but keeps the fine boundary details, including all boundary indentations. In contrast, if we were...
to use a naive gap-filling by executing only an open-close operation sequence, the result would indeed fill most of the internal gaps, but also erase much of the (convex) boundary detail (images (b,h,k,n)). Images (m-r) show the effect of varying the structuring-element radius \( \rho \) (Sec. 3.1) for the input shape in image (m). Images (n) and (o) show, for illustration purposes, the open-closed shape \( \Omega_{oc} \) and its skeleton \( S(\Omega_{oc}) \) respectively for the input image and the \( \rho \) value for the result shown in image (q). As we increase \( \rho \), larger internal gaps get progressively filled. However, fine-scale details on the apparent boundary of the input image stay preserved. As such, \( \rho \) can be effectively used to control the thickness of the internal shape gaps to be filled.

Figure 4 shows results for a set of binary shapes obtained from natural grayscale and color images via luminance thresholding. As expected, thresholding creates many disconnected components and/or holes and cracks within the perceived overall shapes. As for the synthetic images discussed earlier, open-close can fill most such gaps, but inherently destroys the boundary detail. In contrast, our method successfully removes gaps inside the apparent shape, but keeps most boundary detail.

Figure 5 shows a variation of our gap-filling technique. We start like in the previous cases, i.e., we produce a binary segmentation (b) by luminance-thresholding of a grayscale CT brain image (a), taken from [Tel12]. The segmentation result shows significant noise and gaps that disconnect the apparent (black) foreground shape. Images (c-e) show the result of our gap-filling method. In contrast to the earlier examples (Figs. 3, 4), we use now the skeleton \( S(\Omega_{oc}) \) instead of \( S(\Omega_{co}) \). The effect is that more, and larger, gaps get filled, as we increase the structuring-element radius \( \rho \). Additionally, instead of using the full skeleton \( S(\Omega_{co}) \), we now threshold it by eliminating branch end-fragments that correspond to fragments of the boundary \( \partial \Omega_{co} \) shorter than \( \tau \) pixels, using the skeleton-importance metric proposed in [TvW02], to which we refer for implementation details. This further smooths the boundary of the reconstructed shape (yellow pixels in Fig. 5). Overall, by increasing \( \rho \) and \( \tau \), we thus obtain a set of progressively simpler segmentations where larger holes are filled by smoother segments. However, as visible in images (c-e), small-scale convex boundary detail are still well preserved.

Finally, Fig. 6 shows our method applied to a 3D brain voxel dataset (a). Since we lacked an actual dataset with gaps, we generated these synthetically by performing several cut cuts (b). Image (c) shows the result of our method. As for the 2D examples, internal gaps are removed, while the brain surface detail is kept.

5. Applications for Skin Imaging

We next present several applications of our gap reconstruction technique in the field of skin imaging. Input images are dermoscopic color images, of resolutions ranging from 500\(^2\) to over 2500\(^2\) pixels, showing skin tumors which can be either naevi (benign) or melanoma (malignant). Several techniques exist for the automatic pre-diagnosis of such tumors, based e.g. on the ABCD criteria [DHR01, FRK85, PHJ10]. However, to automatically evaluate such criteria, an accurate segmentation of the tumor from the surrounding skin is required. This is hard to do, as shown in Fig. 7, where we show the result of six known image segmentation methods on a typical dermatoscopic image (mean shift (MS) [CM02], gradient vector flow (GVF) [PHJ10], graph cuts (GC) [SM00], image foresting transform (IFT) [FSL04], level sets (LS) [LG10], and dense skeletons (DS) [vdZMT13]). Three types of problems occur. First, fuzzy tumor areas create strong ir-
regularities in the segmentation boundary (GC, MS). Methods with an in-built boundary smoothness remove such problems, but create too smooth boundaries missing image details (GVF). Both these issues create problems in evaluating the ABCD criteria [PHJ10]. Secondly, occluding hairs generate boundary artifacts (MS, LS). Finally, several methods are prone to oversegmentation (MS, GC, DS). All in all, this proves that segmentation of such images is a challenging task.

Figure 7 j shows the result of our method, applied to a luminance-based thresholding of the input skin image (Fig. 7 e). As visible, thresholding generates many holes, due to both inherent color variation in the tumor, and to occluding hairs. As visible, our method generates smooth (but also detail-rich) boundaries, does not oversegment the image, and is not sensitive to occluding hairs. The gap-filling effectively removes the latter two issues, but does not remove the fine-scale detail present on the tumor boundary. Figure 7 b shows, for comparison, a manual segmentation performed by a dermatologist. While this segmentation is unavoidably not identical to ours, we notice that our result is, among the set of automatic techniques considered, the closest, both in shape and extent, to the manual segmentation.

Figure 8 shows the result of our method on five other skin-tumor

---

Figure 3: Gap filling for a set of simple shapes. (a,d,g,j,m) Input shapes $\Omega$. (b,e,h,k,n) Result of an open-close operation. (c,f,i,l,p-r) Gap-filling results, with blue=background pixels, green=foreground pixels, yellow=filled pixels, and red=skeleton pixels. See Sec. 4.
images taken with different acquisition devices, of various resolutions, and showing widely different patterns that correspond to different types of skin diseases. As visible, our method improves the binary thresholding results by closing gaps inside the apparent tumor shape but keeping the tumor boundary details.

Finally, Figure 9 shows the use of our method for automatic hair removal in dermoscopic images. The input image (a) show a very complex tumor shape, which is also covered by dense hair. Applying our technique on a luminance-thresholded image (b) yields the segmentation in (c). To remove hairs, we use $\Omega_r \setminus \Omega$ (see image (d)), i.e., the difference between our result $\Omega_r$ (c) and the thresholded image $\Omega$ (b) as a mask for inpainting the input image using the method in [Teo04]. The result (f) shows that all internal hairs have been successfully removed while preserving the tumor texture. Note that, for diagnostic image analysis, accurately segmenting the tumor and removing hairs inside the tumor only, is sufficient: Diagnostic analysis will next only run on the portion of the image inside the tumor, so all hairs (as well as healthy skin) outside the tumor are irrelevant. In contrast, the DullRazor method [LNG+97], with the software provided by the authors, one of the best-known hair-removal techniques in dermato-imaging, fails to remove most hairs (c), as it cannot robustly detect these, and is also considerably slower (16 seconds vs 0.6 seconds for our method on the platform mentioned in Sec. 3.2). Upon closer analysis, this is not surprising: DullRazor detects hairs using a contrast-based edge detector that works well for relatively separated constant-color hairs covering a low-contrast tumor of highly different luminance than the hairs. In our image, however, we have dense, variable-luminance, hairs that overlap a highly textured tumor, so this method fails.

For validation, we showed our skin-image segmentation and restoration results to a dermatology specialist, with over 6 working-experience years in dermato-oncology. We posed a set of questions pertaining qualitative aspects of our results, such as perceived correctness, relative quality with respect to other similar automatic methods, and relative quality with respect to manual segmentation. The test-set included over 30 images (not all present in this paper). The specialist responded very positively, pointing our that our segmentations are, in nearly all cases, superior in terms of boundary smoothness, detail preservation, and ease-of-use, to

---

(c) The Eurographics Association 2014.
any known automatic method (though she indicated that manual segmentation can sometimes perform better in some fuzzy image areas). Additionally, our hair-removal method was found qualitatively better than manual alternatives and much better than DullRazor, for all complex images being tested, and equally good to these methods for the simple (low-hair-occlusion) test images. In particular, it was noted that our method has only two user-parameters (the luminance threshold level and gap-filling radius \( \rho \)), so it is much simpler to learn and use than other methods which expose more, and more complex, parameters.

6. Discussion

Below we discuss several aspects of our method.

Strengths: The main strength of our method is the ability to close gaps which appear inside the input binary shape, and in the same time keep both convex and concave detail present on the apparent boundary of the same shape. The method can handle well gaps of variable position, orientation, and thickness, as demonstrated by the examples shown in this paper. The single user-parameter to control is \( \rho \), the maximal thickness of gaps to be closed, which has an intuitive meaning. Experiments done showed that our method can yield good-quality segmentations and restorations of dermatoscopic images, which are perceived to be better, and more useful, by domain specialists.

Limitations: The key heuristic of our method is the classification of error gaps (to be filled) as being those which intersect the skeleton of a simplified (open-close) version \( \Omega_{oc} \) of the input shape \( \Omega \). The main rationale behind this heuristic is that (a) open-close simplifies \( \Omega \) by removing details but keeping its main structure, and its parameter \( \rho \) allows specifying the maximal thickness of gaps to be filled (e.g., allows users to specify that large gaps are important details, so should not be filled); (b) hence, the skeleton of the simplified shape \( \Omega_{oc} \) captures the main part-whole structure of the original \( \Omega \); (c) gaps in the original \( \Omega \) that cut this skeleton affect thus more critically the ‘structure’ of \( \Omega \), and thus should be removed, than gaps far from the skeleton, which can be safely regarded as details of \( \Omega \). Clearly, there can exist application contexts where step (c) of our heuristic would fail. In such cases, our method would fill less gaps than desired. However, in the over 120 examples tested so far, we have noticed that our heuristic works as expected, i.e. discriminates between relevant gaps (far from the shape skeleton, and thus should be kept) and error gaps (which locally cut the shape more than half, and thus should be removed) in the desired way. However, we fully agree that our heuristic needs more testing before being able to state its value in a strong sense.

Comparison: In our presented examples, we make a number of
simulations. First, we only use basic luminance thresholding for creating the input binary images for our gap-filling and restoration process. Clearly, more advanced techniques can be used. However, we chose a simple technique precisely to be able to demonstrate the added-value of our method on poor-quality input images. Secondly, the comparison against the six segmentation methods in Sec. 5 is surely limited, as more such methods exist. However, as stated, it is noteworthy that our (simpler) method performs qualitatively better than this range of very different segmentation methods. Thirdly, our inpainting examples only use a simple technique [TvW02]. We do this to clearly separate the inpainting effects from the added-value of our method. End applications can improve this skeleton. We efficiently implement our method by means of these artifacts but fully preserve the shape boundary. To achieve the error-gap detection (Sec. 3.2) and gap filling (Sec. 3.3) would not correctly work.

Acknowledgments
This work was financially supported by the PN-II-RU-TE-2011-3-2049 “Image-assisted diagnosis and prognosis of cutaneous melanocytic tumors” (ANCS, Romania) and grant 202535/2011-8 (CNPq, Brazil).

7. Conclusion
We have presented a method for reconstruction of binary 2D and 3D shapes that miss internal information in the form of holes, disconnections, and cracks. In contrast to local filtering methods, which can remove such artifacts, but also smooth our relevant details on the shape boundary, our method can successfully remove these artifacts but fully preserve the shape boundary. To achieve this, we propose a heuristic to classify gaps in terms of their position to the shape skeleton, and next remove deep gaps which intersect this skeleton. We efficiently implement our method by means of distance transform and skeletonization algorithms for both the 2D and 3D cases. Finally, we present a concrete application of our technique for robust image segmentation and hair removal in dermatological applications, and compare our results with a number of known segmentation and one restoration technique in this field.

Future work can target a number of directions. Technique-wise, our method could be extended to the area of hole and crack filling in 3D surface meshes [Lie03, VCBS03]. Application-wise, we can adapt our method for the reconstruction of 3D scalar and/or vector fields, such as CT and MRI scans, by removal and restoration of low-quality and low-certainty areas [DMGL02].

References
[HR09] HESSELINK W., ROERDINK J.: Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. IEEE TPAMI 12, 30 (2009), 2204–2217. 3, 8
[LXGF10] Li C., Xu C., Gu C., Fox M. D.: Distance regularized level set evolution and its application to image segmentation. IEEE TPAMI 32, 12 (2010), 2343–2354. 4