A Dense Medial Descriptor for Image Analysis

Matthew van der Zwan[1], Yuri Meiburg[1] and Alexandru Telea[1,2]

[1]*Scientific Visualization and Computer Graphics, University of Groningen, Nijenborgh 9, Groningen, The Netherlands*

[2]*University of Medicine and Pharmacy* Carol Davila*, Bucharest, Romania*

Keywords: Medial Axes, Image Segmentation, Shape Analysis.

Abstract: We present dense medial descriptors, a new technique which generalizes the well-known medial axes to encode and manipulate whole 2D grayvalue images, rather than binary shapes. To compute our descriptors, we first reduce an image to a set of threshold-sets in luminance space. Next, we compute a simplified representation of each threshold-set using a noise-resistant medial axis transform. Finally, we use these medial axis transforms to perform a range of operations on the input image, from perfect reconstruction to segmentation, simplification, and artistic effects. Our pipeline can robustly handle any 2D grayscale image, is easy to use, and allows an efficient CPU or GPU-based implementation. We demonstrate our dense medial descriptors with several image-processing applications.

1 INTRODUCTION

Skeletons, or medial axes, are well-known 2D shape descriptors used in many applications in shape analysis and classification, shape recognition, shape matching, topological analysis, image registration, and path planning. Medial axis structures, augmented with distance information from the medial axis to its corresponding shape, generate the so-called Medial Axis Transform (MAT), which is a true dual for the input shape. In other words, the MAT can be used for the exact reconstruction and also for the simplification of shapes at user-specified levels of detail. 2D skeletons and MATs have been extended to three dimensions to create surface and curve skeletons and their corresponding medial surface transform (MST), which allow processing of 3D shapes analogously to their 2D counterparts. In this paper, we focus on 2D skeletons and MATs.

However powerful, skeletons and MATs have the crucial limitation that they require as input a digital shape, *i.e.* a closed boundary which divides the embedding space into inside and outside regions. This limits their direct application to datasets containing pre-segmented shapes. However, in many applications, one has continuous fields as inputs, such as grayscale or color 2D images or 3D scalar volumes such as CT or MRI scans. Although pre-segmenting such datasets into binary shapes and further using skeletons to analyze such shapes is possible, this is a non-trivial process which requires *a priori* knowledge on the nature and position of the shapes of interest. Moreover, since skeletons require binary shapes, they cannot directly handle fuzzy shapes whose boundaries are defined by a range of scalar values. Eliminating this limitation, *i.e.* enabling skeletal and MST descriptors to directly handle grayscale images, can open new ways for using the analytic power of such descriptors for image segmentation, editing, and classification applications.

In this paper, we present a framework for representing and manipulating 2D images using a new descriptor: Dense medial axes. Our framework operates in three steps. First, we decompose a grayscale image into several so-called threshold sets T_i, *i.e.* pixels whose values exceed a given set of scalar values v_i. Next, we compute a simplified medial axis transform M_i of each threshold set T_i, using a suitable simplification value τ_i. Finally, we use the medial transforms M_i to perform several types of image processing operations on the initial image, ranging from perfect image reconstruction to image simplification, segmentation, editing, and artistic painting effects. The set of threshold-values v_i and medial simplification values τ_i effectively create a two-dimensional scale-space in which we encode the image luminance variations and shapes present in the image, respectively. We propose an efficient GPU-based implementation of our dense medial descriptors, which can compute these and the associated image processing operations in real-time

Zwan M., Meiburg Y. and Telea A..
A Dense Medial Descriptor for Image Analysis.
DOI: 10.5220/0004279202850293
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2013), pages 285-293
ISBN: 978-989-8565-47-1

on mega-pixel images. We demonstrate our framework with several image processing applications.

The structure of this paper is as follows. Section 2 overviews related work on 2D medial descriptors. Section 3 details the three steps of our framework: threshold-set computation (Sec. 3.1), medial transform computation (Sec. 3.2), and image reconstruction (Sec. 3.3). Section 4 presents ways in which we can parameterize the above-described steps of our pipeline to achieve several types of image processing operations, and illustrates these with examples. Section 5 discusses our framework. Section 6 concludes the paper.

2 RELATED WORK

Given a two-dimensional binary shape $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega$, we first define its *distance transform* $DT_{\partial\Omega} : \Omega \to \mathbf{R}_+$ as

$$DT_{\partial\Omega}(\mathbf{x} \in \Omega) = \min_{\mathbf{y} \in \partial\Omega} \|\mathbf{x} - \mathbf{y}\| \qquad (1)$$

The skeleton, or medial axis, of Ω is next defined as

$$S(\partial\Omega) = \{ \quad \mathbf{x} \in \Omega | \exists \mathbf{f}_1, \mathbf{f}_2 \in \partial\Omega, \mathbf{f}_1 \neq \mathbf{f}_2, \qquad (2)$$
$$\|\mathbf{x} - \mathbf{f}_1\| = \|\mathbf{x} - \mathbf{f}_2\| = DT(\mathbf{x})\},$$

where $\mathbf{f_1}$ and $\mathbf{f_2}$ are the contact points with $\partial\Omega$ of the maximally inscribed disc in Ω centered at $\mathbf{x}$, also called *feature transform* (FT) points (Strzodka and Telea, 2004) or *spoke vectors* (Stolpner et al., 2009), where the feature transform is defined as

$$FT_{\partial\Omega}(\mathbf{x} \in \Omega) = \arg\min_{\mathbf{y} \in \partial\Omega} \|\mathbf{x} - \mathbf{y}\|. \qquad (3)$$

The skeleton, together with the distance transform, form the Medial Axis Transform (MAT), which can be used to exactly reconstruct the input shape Ω (Siddiqi and Pizer, 2009; Telea and van Wijk, 2002).

Two-dimensional MAT techniques can be classified into three groups. *Geometric* methods use a polygonized version of $\partial\Omega$ to compute its Voronoi diagram and the skeleton as a subset thereof (Ogniewicz and Kubler, 1995). *Thinning* methods iteratively remove $\partial\Omega$ pixels while preserving connectivity (Palagyi and Kuba, 1999). Pixel removal in distance-to-boundary order enforces centeredness (Pudney, 1998). Such methods are simpler than geometric methods and they also directly use a pixel-based image representation. *Distance field* methods find the MAT along singularities of $DT_{\partial\Omega}$ (Rumpf and Telea, 2002; Telea and van Wijk, 2002; Wan et al., 2001; Siddiqi et al., 2002; Hesselink and Roerdink, 2008), and can be efficiently implemented on GPUs (Strzodka and Telea, 2004; Sud et al., 2005; van Dortmont et al., 2006; Cao et al., 2010). General-field methods use fields smoother (with less singularities) than distance transforms (Ahuja and Chuang, 1997; Cornea et al., 2005; Hassouna and Farag, 2009), thus are more robust for noisy shapes. Foskey *et al.* compute the θ-SMA, an approximate simplified medial axis, using the angle between feature vectors (Foskey et al., 2003). The θ-SMA can get disconnected along the so-called ligature branches. An accuracy comparison of different field methods for 2D distance and feature transforms is given in (Reniers and Telea, 2007).

Clear, or regularized, skeletons are extracted from noisy shapes by thresholding *importance measures* to prune skeleton pixels caused by small shape details (Shaked and Bruckstein, 1998). One of the simplest, and most effective, such measures is the collapsed boundary length metric $\rho : S \to \mathbb{R}_+$, which ranks skeleton pixels $\mathbf{x}$ by the boundary length, along $\partial\Omega$, between their feature points (Ogniewicz and Kubler, 1995; Costa and Cesar, 2000; Telea and van Wijk, 2002), *i.e.*

$$\rho(\mathbf{x} \in S) = \|\partial\Omega(\mathbf{f_1}, \mathbf{f_2})\|, \qquad (4)$$

where $\mathbf{f_1}$ and $\mathbf{f_2}$ are the feature points of the skeleton point $\mathbf{x}$. The metric ρ increases monotonically on skeletons of genus 0 shapes from their periphery to their center, so thresholding it is guaranteed to directly yield a connected skeleton (Costa and Cesar, 2000; Telea and van Wijk, 2002).

However effective for shape registration (Sundar et al., 2003), matching (Bai and Latecki, 2008; van Eede et al., 2006), classification (Costa and Cesar, 2000), and recognition (Macrini et al., 2008), skeletons can only be computed for *binary* shapes Ω. Given a grayscale image, Ω can be computed using various segmentation methods (Kass et al., 1988; Comaniciu and Meer, 2002; Shi and Malik, 2000; Li et al., 2010). Simpler, but more generic, segmentation methods include classical level-sets and threshold-sets (Sethian, 2002), which are nested structures that capture all image points whose grayvalue is equal to, or respectively larger than, a given value. However, segmentation poses two problems. First, we need to know, in advance, which shapes we are searching for in a given image. Secondly, this approach only delivers skeletons of the image subset captured by the segmentation. In other words, we do not have a medial descriptor for the *entire* image, *e.g.*, we cannot talk about the skeleton of a fuzzy shape.

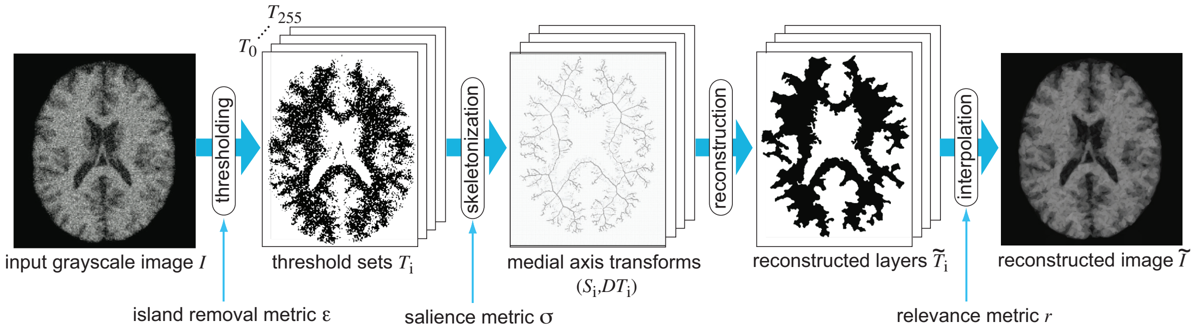

Figure 1: Dense medial descriptor computation pipeline.

3 PROPOSED FRAMEWORK

We propose to join the grayvalue information present in threshold set descriptors with the shape information delivered by medial descriptors in a single new *dense* medial descriptor (DMD), as follows (see also Fig. 1). First, we reduce an image to a set of threshold-sets (Sec. 3.1). Secondly, we compute a simplified MAT for each threshold-set (Sec. 3.2). Finally, we use the threshold-set grayvalues and computed MATs to generate our DMD (Sec. 3.3), which we can use next for various image processing operations (Sec. 4).

3.1 Threshold Set Computation

Given a grayscale image $I : \mathbb{R}^2 \to \mathbb{R}_+$ and a grayvalue $v \in \mathbb{R}_+$, we define the threshold-set

$$T(v) = \{\mathbf{x} \in \mathbb{R}^2 | I(\mathbf{x}) \geq v\}. \tag{5}$$

By definition, threshold-sets for increasing grayscale values are nested 2D structures. For a digital n-bit-per-pixel image, we have thus 2^n threshold sets, or layers, $T_i = T(i), 0 \leq i < 2^n$. Here and further, we use a value of $n = 8$. Further, from each layer, we remove foreground and background islands with an area ε smaller than 3% of the layer's area $|T_i|$. This further simplifies our medial descriptors used to encode the layers (Sec. 3.2).

The density of layer borders is proportional with the probability of having an edge in the image: Low-density areas indicate that consecutive layers are far apart, thus the image is relatively flat. High-density areas indicate close consecutive layers, *i.e.* quickly varying grayvalues. We will further use this observation in the reconstruction pass (Sec. 3.3). Threshold-sets are nested, *i.e.* $\forall i < j, T_j \subset T_i$. This observation is important, as it implies that if we remove a pixel from T_j, this pixel will get the (darker) grayvalue of its closest parent layer.

3.2 Simplified Medial Axis

A threshold-set T_i can be seen as an arbitrary 'slice' in the image grayvalue space. Geometrically speaking, T_i can have any shape, *e.g.* a collection of noisy disconnected components. We propose here to use MATs to capture the essence of the shape of a T_i and remove its spurious details.

For this, we compute the distance transform $DT_i = DT(T_i)$ and simplified skeleton $S_i = S(T_i)$, following Eqns. 1 and 3 respectively. We compute DT_i using the GPU-based exact Euclidean method of Cao *et al.* (Cao et al., 2010). This method also computes the feature transform of a shape (Eqn. 3). Hence, it is trivial to modify this method to determine, for each point $\mathbf{x} \in \mathbf{T_i}$, which are its two feature points, and next, following a simple arc-length parameterization of ∂T_i, analogous to (Telea and van Wijk, 2002), the collapsed boundary length at $\mathbf{x}$.

As noted earlier, T_i can contain a large amount of geometrical and topological noise. Once we have S_i and DT_i, we can remove these easily. As skeleton importance, we use the so-called *salience metric*

$$\sigma : S \to \mathbb{R}_+ = \rho / DT, \tag{6}$$

equal to the collapsed boundary length ρ (Eqn. 4) divided by the distance transform (Eqn. 1) (Telea, 2012). This metric has the desirable property that it removes small-scale boundary noise, but it keeps salient features, such as cusps or dents. Figure 2 illustrates this: The input image is an 8-bit noisy human brain CT (a), from which we select the threshold-set corresponding to the level 132 (b). Next, we remove small foreground and background islands (as mentioned in Sec. 3.1), compute the simplified MAT of this threshold-set, regularized by the saliency metric σ, and reconstruct this set from this skeleton. The result (c) captures the main shape described by the threshold-set (b), ignoring small-scale details such as specks, holes, and boundary noise.

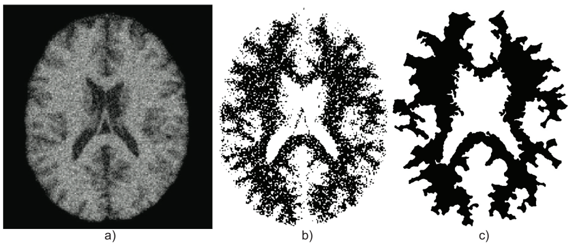

Figure 2: Salience metric for skeleton simplification: (a) grayvalue image; (b) threshold set; (c) threshold set reconstructed from saliency skeleton.

3.3 Image Reconstruction

So far, we have reduced our input grayvalue image I to a set of threshold-sets T_i, each having an MAT S_i. We can reconstruct a simplified version $\tilde{T}_i$ of each T_i (in increasing order of i, *i.e.* from dark to light layers) from its corresponding MAT (S_i, DT_i) by either executing a Fast-Marching-Method (FMM) 'inflation' of S_i outwards until each point $\mathbf{x} \in \mathbf{S_i}$ reaches the distance $DT_i(\mathbf{x})$ (Sethian, 2002; Telea and van Wijk, 2002), or alternatively drawing discs centered at $\mathbf{x}$ with radius $DT_i(\mathbf{x})$, and coloring each layer T_i with the grayvalue i. The first approach (FMM) works best on a CPU, while the second scales better on the GPU. However, this method is basically a nearest-neighbor (zero-order interpolation) reconstruction of I, so it shows intensity-banding effects around the boundaries ∂T_i of the layers T_i. A better way is as follows: For each two consecutive layers i and $i+1$, whrere $i \in [0,255]$, we reconstruct the layers T_i and T_{i+1} as above (either on the CPU or GPU), and next, we set the grayvalue $v(\mathbf{x})$ of each pixel $\mathbf{x}$ located between the boundaries ∂T_i and ∂T_{i+1} to

$$v(\mathbf{x}) = \frac{1}{2}\left[\min\left(\frac{DT_{T_i}}{DT_{T_{i+1}}},1\right)v_i + \max\left(1-\frac{DT_{T_{i+1}}}{DT_{T_i}},0\right)v_{i+1}\right] \tag{7}$$

This achieves a smooth distance-based interpolation between the boundaries ∂T_i (with grayvalue v_i) and ∂T_{i+1} (with grayvalue v_{i+1}) in the Hausdorff sense. Applying Eqn. 7 at all pixels $\mathbf{x}$ of the image space yields our final reconstructed image $\tilde{I}$. Examples of reconstruction are discussed next in Sec. 4.1.

4 APPLICATIONS

We next present several applications of our dense medial descriptor.

4.1 Reconstruction

Overall, the interpretation of our reconstruction technique is simple: Given several layers T_i with corresponding MATs (S_i, DT_i), we can reconstruct a smooth version of the original image I from which T_i have been produced. Thus, I is integrally encoded in our dense MAT (S_i, DT_i). For instance, if we encode *all* layers T_i present in the original image, and do not simplify at all the resulting MATs S_i, the reconstruction presented in Sec. 3.3 is an *exact* copy of the original image, by definition, *i.e.* since we encode all luminance layers and since an unsimplified skeleton exactly preserves the shape of each layer. Thus, our dense medial descriptor (DMD) can encode the full input information, if desired. However, our DMD can be used to simplify the input image I in several ways, as follows.

First, we can only encode the *relevant* layers T_i. A layer is deemed relevant if its removal from the reconstruction (Sec. 3.3) causes a too large difference between the original image I and the reconstructed image (Eqn. 7). Indeed, the advantage of our reconstruction scheme is that it allows to easily remove, or keep, layers T_i in the reconstruction process. As such, given a typical image with 255 layers, we can decide on-the-fly which layers are relevant for the reconstruction or not, depending on application-specific metrics.

The simplest of such metrics is the *relevance* of a layer: Given all layers T_i, we can remove those which contribute less to reconstructing an image close to the input image I. To compare the reconstruction $\tilde{I}$ with the original image I, we use the well-known mean structural similarity index (SSIM) metric (Wang et al., 2004). Figure 3 ilustrates this. Here, we have removed the least relevant layers to the reconstructed image (as according to SSIM) and plotted the SSIM metric. We see that we can remove around 30..50% of the 255 layers of an 8-bit image without a perceptual decrease in image quality: Details such as salient sharp boundaries, highlights, and even global small-scale patterns (such as the mandrill's hair structure) are well preserved. Accordingly, this means we can *compress* an image, by the same layer removal ratio, *i.e.* 60% (a), 23% (b), 78% (c), and 37% (d) respectively, with little *perceptual* loss. Consequently, if we accept the implied perceptual difference, between the input image and our simplification, techniques such as JPEG encoding can be subsequently applied atop of our simplification.

4.2 Segmentation

Segmenting an image into its salient shapes has countless applications in medical imaging, computer vision, and image classification. We show below how

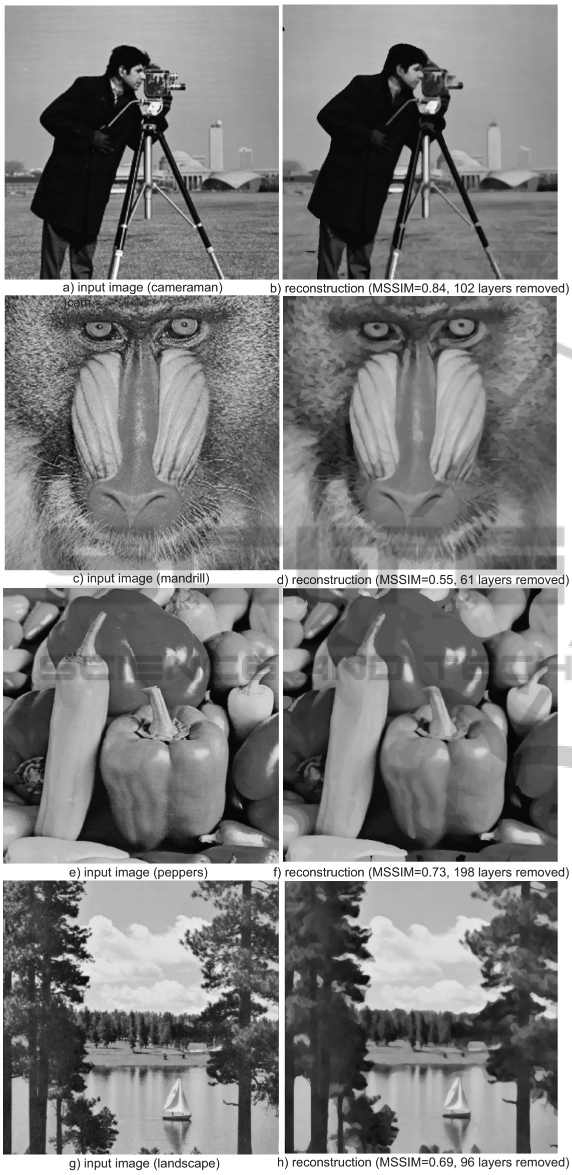
a) input image (cameraman)
b) reconstruction (MSSIM=0.84, 102 layers removed)
c) input image (mandrill)
d) reconstruction (MSSIM=0.55, 61 layers removed)
e) input image (peppers)
f) reconstruction (MSSIM=0.73, 198 layers removed)
g) input image (landscape)
h) reconstruction (MSSIM=0.69, 96 layers removed)

Figure 3: Image reconstruction accuracy (*vs* SSIM) while removing layers.

our DMD representation can be used for image segmentation. Given an image I with $0 \leq i < 256$ grayvalues, we compute, for each layer T_i, its relevance r as being the difference between I and the reconstruction using all layers except T_i. Next, we select for reconstruction only the k most relevant layers, where k is a small user-supplied value. This will keep the most 'salient' shapes present in the image. Moreover, since the shape of each layer is simplified by island removal (Sec. 3.1) and boundary jaggies removal (Sec. 3.2), the resulting reconstruction will have simpler shapes.

Figure 4 shows an example. Statistics of the input image (a) are displayed in Fig. 4e. The area $|T_i|$ of a layer is computed as the pixels that have precisely grayvalue i. We notice, as expected, that the darkest 20% layers are empty and brighter layers have increasingly less pixels (highlights are smaller than darker zones). The number of shapes (connected components) per layer is relatively large for the layers having a large area, which indicates that the image is non-trivial to segment. The relevance r shows several local peaks: These are layers which are (1) globally significant for the image representation and (2) more significant than layers having similar grayvalues. We select the $k = 6$ most significant such layers as shown in Figure 4b. Here, we do not use the linear interpolation of consecutive layer grayvalues (Eqn. 7), so the result shows a luminance-quantization-like segmentation of the image. In contrast, using linear interpolation (Fig. 4c) blurs the reconstruction where the original image has low contrast (since, as explained in Sec. 3.1, layers in such zones have far-apart boundaries) but keeps sharp luminance edges visible (since these correspond to high-density layer boundaries). This yields a fuzzy segmentation of the input image. For comparison purposes, Fig. 4d shows the result of mean-shift segmentation (Comaniciu and Meer, 2002) applied on the input image. Although the produced segments are not identical to ours (Fig. 4b), the overall segmentation impression is similar.

The exact selection of the k most important layers is not critical for the segmentation results. Figure 5 shows an example. Here, we reconstructed the input image (a) by using the 60% most relevant layers. The result (b) is quite similar with the results of mean shift segmentation (c), see *e.g.* the segments corresponding to the house walls, window panes, and bush. Our segments are less jagged than the ones produced by mean shift *and* still preserve their salient sharp corners, see *e.g.* the areas marked in red on the figure. The reason for this is the working of the salience regularization metric for medial axes, which, as explained, eliminates small boundary jaggies but keeps salient corners unchanged. On the other hand, our result (b) has a slightly more fuzzy segmentation aspect than mean shift (c). If a more clear-cut segmentation is desired (less segments), fewer most-relevant layers can be selected, as shown in the earlier example (Fig. 4).

Figure 6 shows a final segmentation example. The input image (a) shows a skin lesion (naevus) photograph taken with a Handyscope mobile dermatology device in the framework of a digital dermatology skin-cancer screening project. A typical network naevus structure is visible herein. The central part of the naevus has a slightly darker, and denser, network pattern, which is only visible on the original high-resolution 1936 by 2592 pixels image. The marked

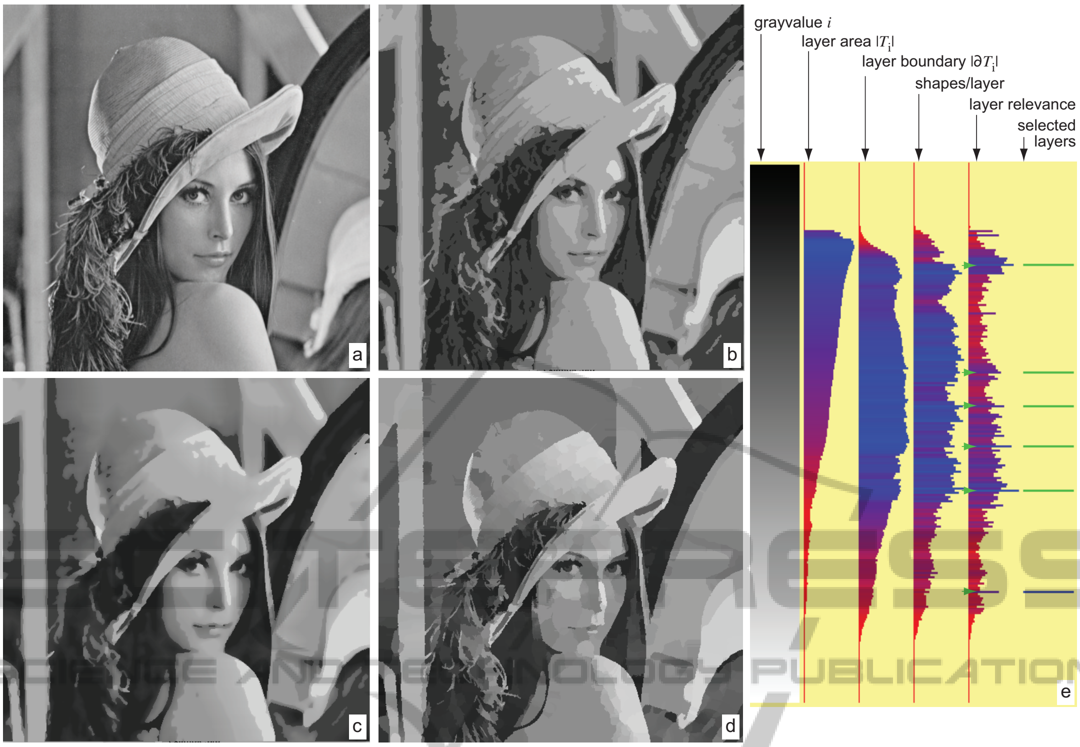

Figure 4: Segmentation example. (a) input image; (b) 5 most relevant layers selected; (c) reconstruction; (d) mean shift segmentation comparison; (e) layer statistics visualization (see Sec. 4.2).

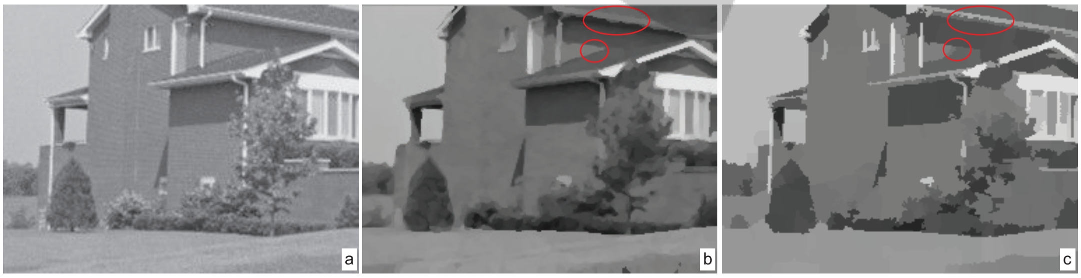

Figure 5: Segmentation example. (a) input image; (b) our method (60% layers); (c) mean shift segmentation (see Sec. 4.2).

boundary (in green) shows the segmentation of the lesion as manually drawn by a dermatology expert atop of this image. We processed the input image, without the manually-drawn segmentation, to obtain the result in Figure (b). Here, we see the three most relevant layers segmented from the input image, *i.e.*, the lesion's extent atop of the healthy skin (A), and two regions corresponding to the darker and denser central area (B,C). This figure was obtained with relatively low salience and island-removal values, $\varepsilon = 0.03$ and $\sigma = 2$ (Secs. 3.1 and 3.2). If we increase these values to $\varepsilon = 0.05$ and $\sigma = 5$, more small-scale islands and also jaggies on the layers' boundaries get removed. Figure (c) shows the result: The lesion's outer boundary has now been considerably smoothed. Note, also, that this layer is indeed by far the most relevant from all the image's layers, as indicated by its large relevance value (Fig. 6d (A)), and its shape is quite similar to the manual segmentation. The lesion's inner layers are also simplified, but to a lesser extent.

The target users (dermatology medical specialists) noted that the tool can be very useful as a guided aid to their manual work rather than an automatic segmentation techinque: The relevance values suggest salient structures in the input images. Seeing such values, users select them in the relevance metric-bar, visualize the corresponding structures, and decide whether these are useful segments of the case under analysis.

4.3 Artistic Editing

Our method can also be used to generate painting-

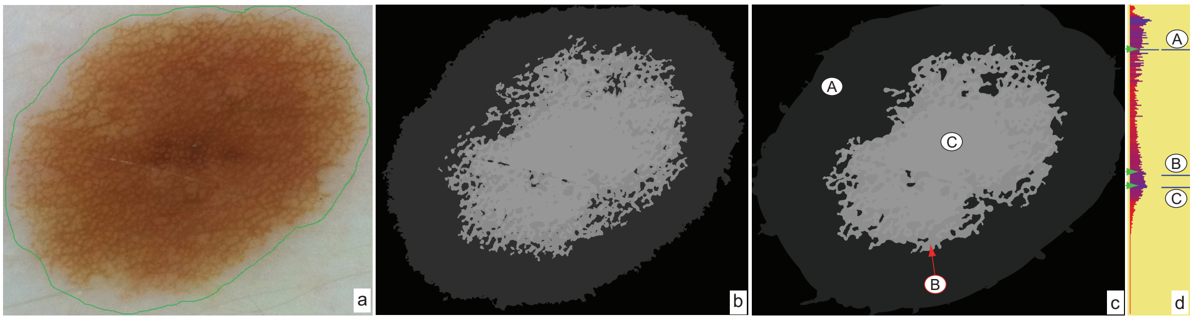

Figure 6: Skin image segmentation. (a) original image showing manual segmentation; (b) detail-preserving automatic segmentation; (c) simplified automatic segmentation with corresponding relevance metric (d).

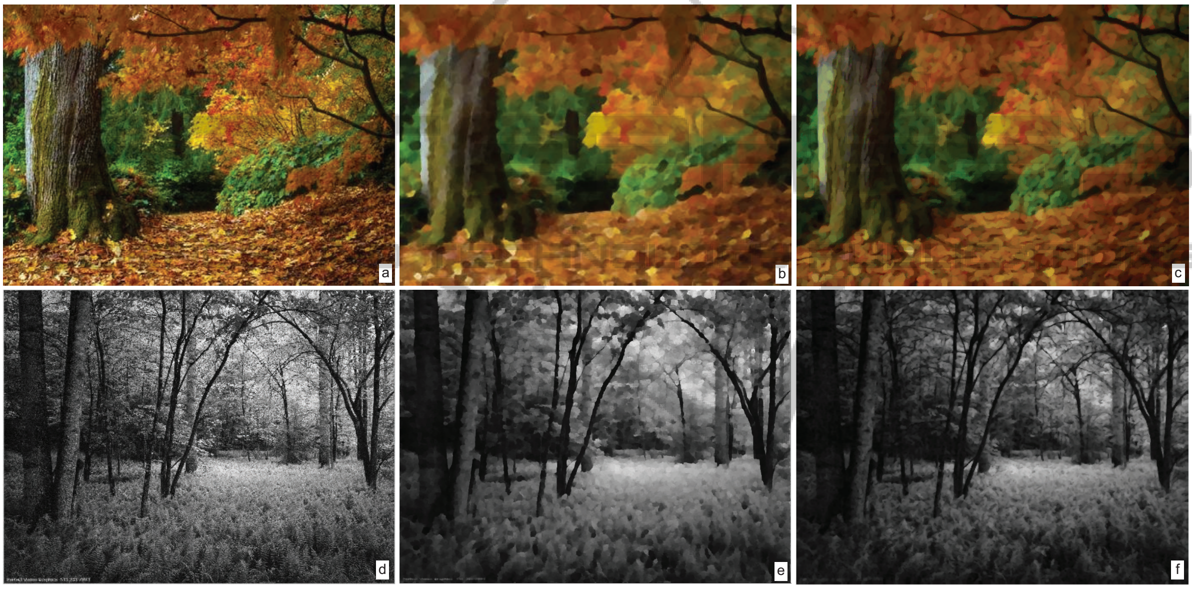

Figure 7: Original images: color example (a) and grayscale example (d). Painting-like effects obtained using the method of (Papari et al., 2007) (b,d) and our method (c,f).

like effects from a given (sharp) photograph, similar to the artistic edge and corner preserving smoothing effect of Papari *et al.* (Papari et al., 2007). By increasing the skeleton saliency metric σ (Eqn. 6), we eliminate small-scale jaggies of *all* threshold-sets, *i.e.* isophote contours, while keeping their sharp corners. The reconstruction (Sec. 3.3) interpolates between these simplified contours, yielding effects akin to painting. Figure 7 (c,f) illustrates this on two complex, fine-grained detail, images. The resulting images, where MSTs have been simplified by a saliency value of $\sigma = 0.4$ and we kept 65% of the original threshold-sets, show a painting-like effect of the input forest images, where small-scale details are ‘clustered’ into larger shapes (due to the skeleton simplification), but the contrast is not unnecessarily blurred (due to keeping a significant number of the original grayvalues, or threshold-sets). As such, salient details such as the dark thin trees and light spots are well preserved, but small-scale *and* weak-contrast details such as the foliage, are simplified. The painting effect is strikingly similar with the results produced by the method of Papari *et al.*, see Fig. 7 (b,e).

5 DISCUSSION

Below we discuss several aspects of our method.

Robustness. We use medial axes for saliency-based simplification and encoding of image layers. Although medial axes are known to be unstable and not robust to noise, we should stress that this does not affect our method. Indeed, we use *regularized* medial axes, *i.e.* eliminate noisy branches by means of the salience metric (Eqn. 6). As explained in detail

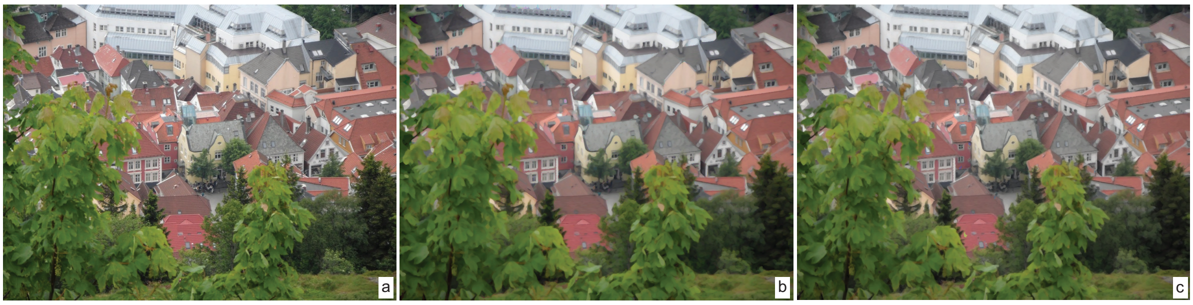

Figure 8: Original color image (a). Simplified representation using our method in the RGB space (b) and HSV space (c).

in (Telea and van Wijk, 2002; Telea, 2012), this regularization produces medial axes which are robust to arbitrary boundary noise for shapes of arbitrary genus. Also, we should note that the medial axes are exact, *i.e.* precisely centered in their shapes and pixel-thin, by construction, given the exact Euclidean DT we use (Cao et al., 2010) and the underlying skeletonization algorithm (Telea and van Wijk, 2002).

Speed. Our method relies on the fast computation of distance transforms and skeletons (Secs. 3.2,3.3). On the CPU, we have used for this the method presented in (Telea and van Wijk, 2002), which is worst-case $O(n\ log(\sqrt{n}))$ for an image of n pixels. On the GPU, using the method in (Cao et al., 2010), we achieve a complexity of $O(n)$. For images of 512^2 pixels, our CPU method takes about 1 minute on a PC at 2.5 GHz, while on an Nvidia 330 GTM GPU, we take 1..2 seconds. The memory complexity is $O(n)$, as we only need to store a fixed set of 256 MSTs per image.

Parameters. Our method selects a subset of relevant threshold-sets from the 256 possible sets, and then computes simplified MSTs for each such threshold-set according to the specified saliency. Hence, saliency and relevance (σ, r) create a two-dimensional scale-space for the input image. Selecting less threshold-sets (high r) emphasizes fewer high-relevance structures in the image (Sec. 3.3. Simplifying each MST (high σ) reduces the border-detail of such contours (Sec. 3.2). The third and final parameter is the size ε of the foreground and background islands to be removed (Sec. 3.1). For typical applications, setting ε to values between 3 and 5% of the area of a layer achieves the desired effect, *i.e.* removal of small isolated bright or dark specks.

Color Images. Applying our DMD representation to color images is trivial. For this, we apply the entire pipeline (threshold sets, medial axes, and simplified reconstruction) to each channel of a color image. Figure 8 illustrates this. As visible, choosing either an RGB or HSV color space does not create significant differences, as both hues and luminances are well preserved. Computing DMDs for color images is three times slower than for grayscale images, given that we process each color channel independently.

Applications. We have illustrated our method with applications in image segmentation, simplification, and artistic manipulation. For all such use-cases, there exist obviously more specialized methods which yield better results. Our purpose in selecting these use-cases was mainly to illustrate the versatility of our framework, *i.e.* the fact that the proposed DMD representation can be seen as a potential, simple, alternative for a wide spectrum of image processing tasks. As such, we see the DMD as a low-level descriptor atop of which more advanced manipulations can be built, and not as an end-user instrument by itself.

6 CONCLUSIONS

We have presented dense medial descriptors, a new representation that encodes shape and luminance information in grayvalue images. To allow using medial descriptors for such images, we first decompose an image into all its possible threshold-sets, and then encode each such set using classical medial axes regularized by a corner-preserving saliency metric. The resulting descriptor allows an exact reconstruction of the initial image using distance-based interpolation techniques, and also an application-dependent simplification by eliminating shapes, or shape details, of low interest or relevance. We have implemented our descriptor using GPU-based techniques to achieve near-real-time performance. We demonstrate our proposal with applications in image simplification, segmentation, and artistic painting effects.

Many possible extensions of our proposal exist. First, we can exploit the topological information present in our dense medial axes, *e.g.* branching or looping structures, to perform higher-level image analysis tasks such as fuzzy object recognition. Secondly, we can exploit the spatial and topological relations of medial axes of consecutive image layers to perform new types of image editing, *e.g.* fuzzy object deformation, and also to study new methods for image compression. Finally, generalizing our method to 3D scalar volumes is an interesting avenue to explore.

ACKNOWLEDGEMENTS

This project was co-financed by the research grant PN-II-RU-TE-2011-3-2049 "Image-assisted diagnosis and prognosis of cutaneous melanocitary tumors" offered by ANCS, Romania.

REFERENCES

Ahuja, N. and Chuang, J. (1997). Shape representation using a generalized potential field model. *IEEE TPAMI*, 19(2):169–176.

Bai, X. and Latecki, L. (2008). Path similarity skeleton graph matching. *IEEE TPAMI*, 30(7):1282–1292.

Cao, T., Tang, K., Mohamed, A., and Tan, T. (2010). Parallel banding algorithm to compute exact distance transform with the GPU. In *Proc. SIGGRAPH I3D Symp.*, pages 134–141.

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

Cornea, N., Silver, D., Yuan, X., and Balasubramanian, R. (2005). Computing hierarchical curve-skeletons of 3D objects. *Visual Comput.*, 21(11):945–955.

Costa, L. and Cesar, R. (2000). *Shape analysis and classification*. CRC Press.

Foskey, M., Lin, M., and Manocha, D. (2003). Efficient computation of a simplified medial axis. In *Proc. Shape Modeling*, pages 135–142.

Hassouna, M. and Farag, A. (2009). Variational curve skeletons using gradient vector flow. *IEEE TPAMI*, 31(12):2257–2274.

Hesselink, W. and Roerdink, J. (2008). Euclidean skeletons of digiral image and volume data in linear time by the integer medial axis transform. *IEEE TPAMI*, 30(12):2204–2217.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *IJCV*, 1(4):321–331.

Li, C., Xu, C., Gui, C., and Fox, M. (2010). Distance regularized level set evolution and its application to image segmentation. *IEEE TIP*, 19(12):32433254.

Macrini, D., Siddiqi, K., and Dickinson, S. (2008). From skeletons to bone graphs: Medial abstraction for object recognition. In *Proc. CVPR*, pages 324–332.

Ogniewicz, R. L. and Kubler, O. (1995). Hierarchic voronoi skeletons. *Patt. Recog.*, (28):343– 359.

Palagyi, K. and Kuba, A. (1999). Directional 3D thinning using 8 subiterations. In *Proc. DGCI*, volume 1568, pages 325–336. Springer LNCS.

Papari, G., Petkov, N., and Campisi, P. (2007). Artistic edge and corner preserving smoothing. *IEEE TIP*, 16(10):2449–2462.

Pudney, C. (1998). Distance-ordered homotopic thinning: A skeletonization algorithm for 3D digital images. *CVIU*, 72(3):404–413.

Reniers, D. and Telea, A. (2007). Tolerance-based feature transforms. In *Advances in Comp. Graphics and Comp. Vision (eds. J. Jorge* et al.*)*, pages 187–200. Springer.

Rumpf, M. and Telea, A. (2002). A continuous skeletonization method based on level sets. In *Proc. VisSym*, pages 151–158.

Sethian, J. (2002). *Level Set Methods and Fast Marching Methods*. Cambridge Univ. Press.

Shaked, D. and Bruckstein, A. (1998). Pruning medial axes. *CVIU*, 69(2):156–169.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905.

Siddiqi, K., Bouix, S., Tannenbaum, A., and Zucker, S. (2002). Hamilton-Jacobi skeletons. *IJCV*, 48(3):215–231.

Siddiqi, K. and Pizer, S. (2009). *Medial Representations: Mathematics, Algorithms and Applications*. Springer.

Stolpner, S., Whitesides, S., and Siddiqi, K. (2009). Sampled medial loci and boundary differential geometry. In *Proc. IEEE 3DIM*, pages 87–95.

Strzodka, R. and Telea, A. (2004). Generalized distance transforms and skeletons in graphics hardware. In *Proc. VisSym*, pages 221–230.

Sud, A., Foskey, M., and Manocha, D. (2005). Homotopy-preserving medial axis simplification. In *Proc. SPM*, pages 103–110.

Sundar, H., Silver, D., Gagvani, N., and Dickinson, S. (2003). Skeleton based shape matching and retrieval. In *Proc. SMI*, pages 130–138.

Telea, A. (2012). Feature preserving smoothing of shapes using saliency skeletons. *Visualization in Medicine and Life Sciences*, pages 155–172.

Telea, A. and van Wijk, J. J. (2002). An augmented fast marching method for computing skeletons and centerlines. In *Proc. VisSym*, pages 251–259.

van Dortmont, M., van de Wetering, H., and Telea, A. (2006). Skeletonization and distance transforms of 3D volumes using graphics hardware. In *Proc. DGCI*, pages 617–629. Springer LNCS.

van Eede, M., Macrini, D., Telea, A., and Sminchisescu, C. (2006). Canonical skeletons for shape matching. In *Proc. ICPR*, pages 542–550.

Wan, M., Dachille, F., and Kaufman, A. (2001). Distance-field based skeletons for virtual navigation. In *Proc. IEEE Visualization*, pages 239–246.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE TIP*, 13(4):600–612.