Intervertebral Disc Segmentation Using Mathematical Morphology—A CNN-Free Approach

Edwin Carlinet and Thierry Géraud

EPITA Research and Development Laboratory (LRDE)
`firstname.lastname@lrde.epita.fr`

Abstract. In the context of the challenge of "automatic InterVertebral Disc (IVD) localization and segmentation from 3D multi-modality MR images" that took place at MICCAI 2018, we have proposed a segmentation method based on simple image processing operators. Most of these operators come from the mathematical morphology framework. Driven by some prior knowledge on IVDs (basic information about their shape and the distance between them), and on their contrast in the different modalities, we were able to segment correctly almost every IVD. The most interesting feature of our method is to rely on the morphological structure called the Three of Shapes, which is another way to represent the image contents. This structure arranges all the connected components of an image obtained by thresholding into a tree, where each node represents a particular region. Such structure is actually powerful and versatile for pattern recognition tasks in medical imaging.

Keywords: mathematical morphology · tree of shapes.

1 Introduction

Segmenting intervertebral discs (IVDs) is important to be able to measure automatically their degeneration. Indeed, there is a strong association between such degeneration and low back pain, which is one of the most prevalent health problems amongst population and, consequently, a leading cause of disability that affects work performances and well-being.

The recent trend in medical imaging segmentation is to use convolutional neural networks (CNN), which was not yet the case of the (rather) recent state-of-the-art methods such as [15,1,14,10]. Since many research groups would probably take advantage of the powerful—yet black-boxed–CNNs, we have decided to propose an alternative approach based on *mathematical morphology*. Section 2 explains the morphological tools used in our method, which is described in 3. The result we obtained on the data provided by the challenge "Automatic intervertebral disc localization and segmentation from 3D multi-modality MR images (IVDM3Seg)"[1], that took place at the 21st International Conference on Medical

---

[1] Site of the challenge: `https://ivdm3seg.weebly.com/`

Image Computing & Computer Assisted Intervention (MICCAI) 2018, are given in Section 4. As we advocate reproducible research, the code of the method presented here is available from:
`https://publications.lrde.epita.fr/carlinet.19.csi`

2 Theoretical Background

The method we propose falls into the framework of *mathematical morphology*. This section thus recalls the basic notions that are used in this paper. We will consider that an image, either a 2D digital image or a 3D digital volume, are represented by a function $f : X \to Y$, where X is a subset of $\mathbb{Z}^2$, resp. $\mathbb{Z}^3$, and where Y is a subset of $\mathbb{N}$, typically $[\![0, 255]\!]$ in the case of an 8-bit quantization.

2.1 Operators

An operator φ on images (i.e., taking an image as input and producing an image as output) is:
- increasing iff $f_1 \leq f_2 \Rightarrow \varphi(f_1) \leq \varphi(f_2)$,
- idempotent iff $\varphi \circ \varphi(f) = \varphi(f)$,
- extensive iff $\varphi(f) \geq f$,
- anti-extensive iff $\varphi(f) \leq f$.

In the writing of these properties, we implicitly consider that, for an operator φ, they apply whatever the considered functions. Furthermore, $\varphi \circ \varphi(f) = \varphi(f)$ means that $\forall x \in X$, we have $\varphi \circ \varphi(f)(x) = \varphi(f)(x)$. In the following, we will also use the classical operator *compact* notation, such as $\varphi \circ \varphi = \varphi$, meaning that such a property applies whatever the function. Last, we say that:
- the operators φ and ψ are dual iff $\varphi(f) = -\psi(-f)$,
- the operator φ is self-dual iff $\varphi(f) = -\varphi(-f)$.

2.2 Morphology with Structuring Elements

First let us recall the couple of fundamental operators of mathematical morphology. We call *structuring element*, a set B of vectors having the same discrete coordinate system than X. In the following, we will only consider structuring elements with the two following properties:
- centered, that is, $0 \in B$,
- and symmetrical, that is, $b \in B \Rightarrow -b \in B$.

The structuring element is a parameter for some morphological operators; its shape influences the filtering effect, while its size adjusts the filtering strength. Given a structuring element B, the dilation δ and the erosion ε are operators on images, respectively defined by:

$$\forall x \in X, \ \delta_B(f)(x) = \max_{b \in B} f(x + b) \tag{1}$$

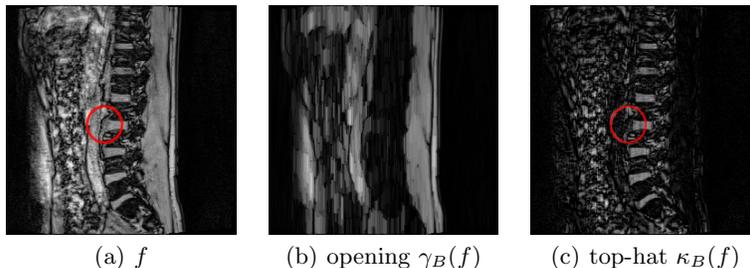$$\varepsilon_B(f)(x) = \min_{b \in B} f(x + b). \tag{2}$$

(a) f (b) opening $\gamma_B(f)$ (c) top-hat $\kappa_B(f)$

Fig. 1. Illustration of the white top-hat effect, with B being a vertical line of 15 pixels.

These two operators are dual, so $\varepsilon_B(f) = -\delta_B(-f)$. The dilation is extensive (the resulting image is brighter than the input image), whereas the erosion is anti-extensive (the result is darker than the input).

From these two operators, we can define two idempotent operators, the closing (extensive) and the opening (anti-extensive), respectively by:

$$\phi_B = \varepsilon_B \circ \delta_B, \tag{3}$$

$$\gamma_B = \delta_B \circ \varepsilon_B, \tag{4}$$

which are dual: $\phi_B(f) = -\gamma_B(-f)$. If we consider that an image f is seen as a landscape, where $f(x)$ is the elevation—height of the landscape—at point x, the effect of the closing ϕ_B is to fill valleys, i.e., image parts surrounded (in the sense of B) by brighter pixels, whereas the opening γ_B has the opposite effect: remove montains, i.e., image parts surrounded by darker pixels. The white top-hat operator is derived from the opening:

$$\kappa_B = \mathrm{id} - \gamma_B, \tag{5}$$

where "id" denotes the identity operator. Since we have $\kappa_B \leq \mathrm{id}$, the top-hat operator is anti-extensive: it removes some bright regions in images.

The behavior of the opening and top-hat operators are illustrated in Fig. 1. The IVD spaces appear as light parts in the original image f (Fig. 1(a)), surrounded *vertically* by some darker regions, that correspond to disks. Therefore the effect of an opening with a vertical structuring element is to remove the bright IVD spaces, as it can be seen in Fig. 1(b). In this image, namely $\gamma_B(f)$, the part of the spine is exclusively dark.

The top-hat is the difference $\kappa_B(f) = f - \gamma_B(f)$, so the removed IVDs now reappear; this result is depicted in Fig. 1(c). When comparing the original image f with $\kappa_B(f)$, we can observe that most of the bright parts/objects of f have been filtered out, and, as a corollary, some IVD regions that were connected in f with some other anatomical parts are now de-connected in $\kappa_B(f)$; see for example the red circle in Fig. 1(a) and 1(c).

In the following, the top-hat operator will thus be used to "clean" the 3D volumes in different modalities, so that:

(a) Gray scale (b) An image f (c) Level lines of f

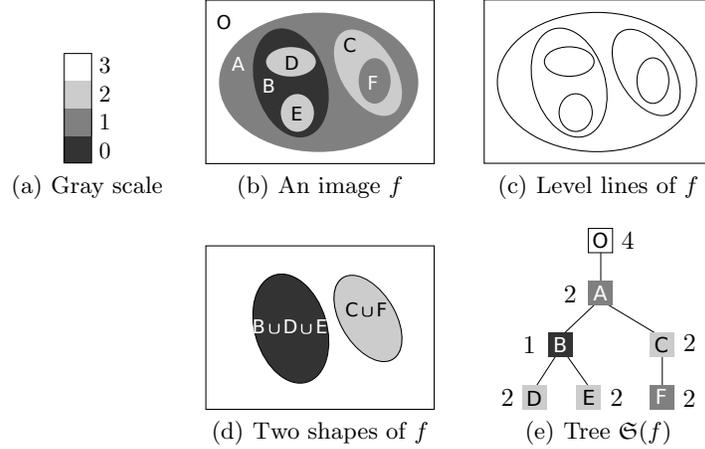(d) Two shapes of f (e) Tree $\mathfrak{S}(f)$

Fig. 2. Toy example of an image, its level lines, its shapes, and its tree of shapes.

- many non-IVD objects are removed in the resulting volumes,
- and IVDs appear more clearly and are de-connected from other objects.

2.3 Tree of Shapes

Given a gray-level image $f : X \to Y$ and any scalar $\lambda \in Y$, the lower level sets are defined as:

$$[f < \lambda] = \{x \in X;\ f(x) < \lambda\}, \tag{6}$$

and the upper level sets as:

$$[f \geq \lambda] = \{x \in X;\ f(x) \geq \lambda\}. \tag{7}$$

We will now consider the connected components (obtained by the operator denoted by $\mathcal{CC}$) of these sets. Let us denote by Sat the cavity-fill-in operator[2]. In the following, we call *shape* the result of the cavity-fill-in operator applied to a connected component of a (lower or upper) level set. In the image f depicted in Fig. 2(b), we have for instance the lower level set $[f < 1] = \mathsf{B}$, and $\mathcal{CC}([f < 1]) = \{\mathsf{B}\}$. Note that B has two holes, namely D and E, so we have the shape $\mathrm{Sat}(\mathsf{B}) = \mathsf{B} \cup \mathsf{D} \cup \mathsf{E}$. An example of upper level set is $[f \geq 2]$, and $\mathcal{CC}([f \geq 2]) = \{\mathsf{C}, \mathsf{D}, \mathsf{E}\}$. C is a component of a level set, so $\mathrm{Sat}(\mathsf{C}) = \mathsf{C} \cup \mathsf{F}$ is a shape. Fig. 2(d) depicts the two shapes $\mathrm{Sat}(\mathsf{B})$ and $\mathrm{Sat}(\mathsf{C})$.

The tree of shapes (ToS) of an image u is classically [13] defined by:

$$\mathfrak{S}(f) \ = \ \{\, \mathrm{Sat}(\Gamma);\ \Gamma \in \mathcal{CC}([f < \lambda]) \cup \mathcal{CC}([f \geq \lambda]) \,\}_\lambda. \tag{8}$$

---

[2] Topological reminder: In 2D, the cavity-fill-in operator is also called hole filling; in 3D, a cavity is just like a bubble within a sphere (whereas a hole is like a tunnel going through a sphere).

An image f and its tree of shapes $\mathfrak{S}(f)$ are depicted respectively in Fig. 2(b) and Fig. 2(e). An element of $\mathfrak{S}(f)$ is called a shape; it is a connected component of X with no cavity, and its boundary is a level line of f. Two shapes of f are displayed in Fig. 2(d). Every shape corresponds to a node of the tree; for instance, in Fig. 2(e) (right), the sub-tree rooted at node "B" corresponds to the shape $\mathsf{B} \cup \mathsf{D} \cup \mathsf{E}$. Keeping the level of every node—such as displayed in Fig. 2(e) (right)—allows to reconstruct the image from its tree. It is thus another way to represent the image contents.
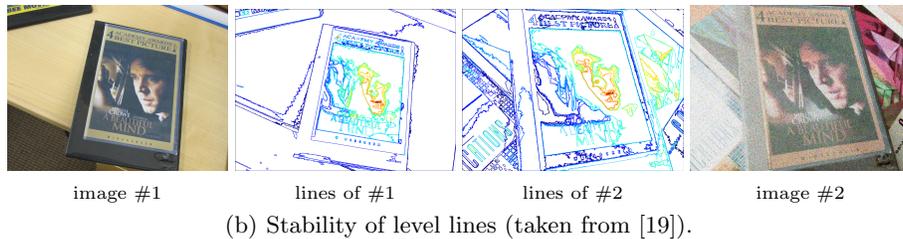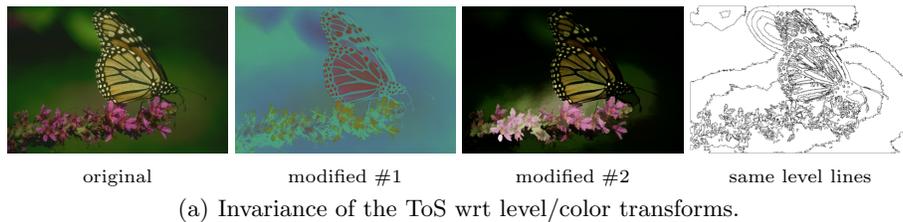

| original | modified #1 | modified #2 | same level lines |

(a) Invariance of the ToS wrt level/color transforms.

| image #1 | lines of #1 | lines of #2 | image #2 |

(b) Stability of level lines (taken from [19]).

Fig. 3. About properties of the tree of shapes and the level lines.

It is worth mentioning that the tree of shapes has also been defined for multi-variate data [5], that is, images whose pixel values are not scalars but vectors, such as it is the case for instance for color images, multi-modality medical images, and multi-band satellite images[3].

The tree of shapes of an image f is a morphological representation of f, which makes it easier to deal with the image contents [9]. For a "classical" image, there is about as many nodes in the tree than pixels in the image. Such a tree thus encodes a lot of shapes (connected components, i.e., regions) and their inclusion relationship. Despite one might think that such a structure should be complex and long to compute, and heavy to store in memory, this is actually not true. Indeed, storing [3] and computing [12,11,6] the tree of shapes can both be done *very efficiently*.

---

[3] Such a multi-variate version of the tree of shapes is used in the illustrations of Fig. 3 and Fig. 4, but not in the segmentation method presented in this paper.

The tree of shapes is an operator satisfying two important major properties. First, we have:

$$\mathfrak{S}(-f) \ = \ \mathfrak{S}(f), \tag{9}$$

meaning that this representation does not favor a particular contrast (light regions surrounded by darker ones, or the opposite). This property thus "contrasts" with the morphological operators presented in Sec. 2.3, where dual operators (such as δ and ε, or ϕ and γ) can be useful exactly because they rely on a particular kind of contrast: we choose one of the dual operators so that we process either brighter or darker parts of the images (remind Fig. 1 for instance). Conversely, the tree of shapes is a structure from which we can derive self-dual operators, that are, operators that process "the same way" light objects *and* dark objects. The second property is that, with any non-negative function ℓ acting over gray-levels (that is, a contrast change function), we have:

$$\mathfrak{S}(\ell \circ f) \ = \ \mathfrak{S}(f). \tag{10}$$

This property implies that it is not the gray-level values of the pixels that matters, but only their ordering. Applying a gray-level change (or look-up table) such as $\ell = [0 \mapsto 0, 1 \mapsto 2, 2 \mapsto 4, 3 \mapsto 5]$ to the image in Fig. 2(b) does not change the structure of its tree of shapes: the ToS of the new image is the one of Fig. 2(e). As a direct consequence, the image processing operators that derive from the ToS structure apply the same way on low-contrasted images (or low-contrasted parts of images) than on better contrasted ones.

These properties are illustrated in Fig. 3. The image on the left in Fig. 3(a) has been modified to produce two new images. The modification #1 consists in contrast-change and contrast-inversion on the different color components; we have applied a function ℓ_i (such as in Eq. 10) on each ith component. The modification #2 is a *local* contrast change. In both cases, the original image and the modified ones have exactly the same set of level lines, depicted on the right. In Fig 3(b), two images (on the left and on the right) share partially the same contents—a DVD jacket; yet the point of view, the lighting environment, and the quantity of noise are different. Despite these differences, the "meaningful" level lines extracted from both images are very similar; the lines are depicted on the middle, the colors expressing the depth in their respective tree of shapes. In the grain filter example, depicted on the top-left part in Fig. 4, we can see that both bright and dark objects tiny are filtering out at the same time, thus illustrating the self-dual property, Eq. 9, of the ToS structure.

2.4 Some Applications of the Tree of Shapes

The tree of shapes is a versatile tool to perform image filtering [17], and a very relevant structure to perform some pattern recognition and computer vision tasks [7,2]. For illustration purpose, Fig. 4 shows that many applications can be derived from manipulating—or just using—the tree of shapes.
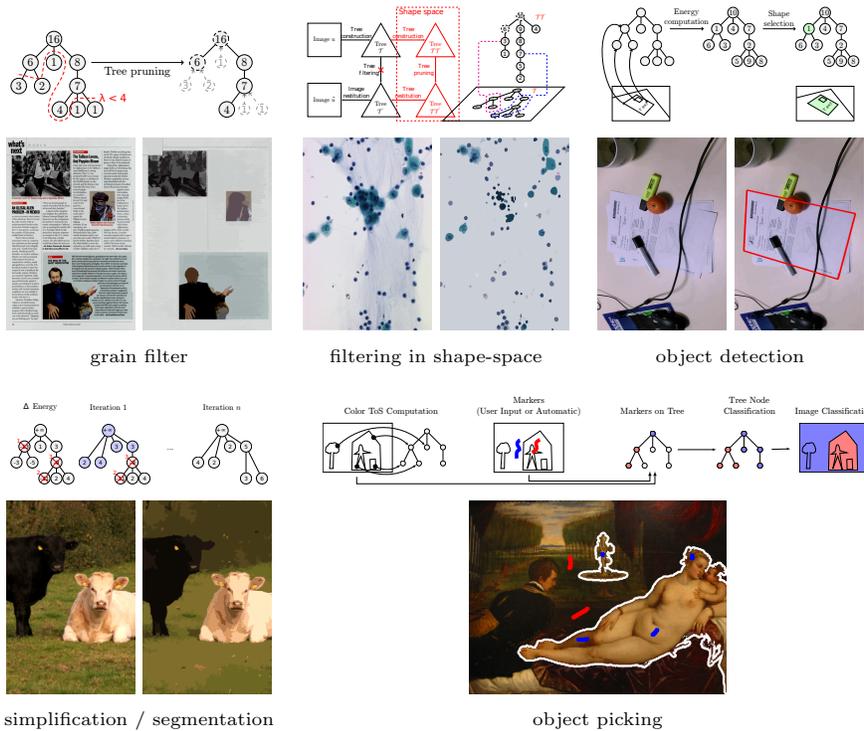
grain filter   filtering in shape-space   object detection

simplification / segmentation   object picking

Fig. 4. Some applications of the ToS: grain filter [8], filtering in shape-space [17], object detection [16], simplification / segmentation [18], object picking [4].

3 Method Description

In the IVDM3Seg challenge, for each patient we have four aligned high-resolution 3D volumes: in-phase, opposed-phase, fat and water images. We only use the three last modalities, abbreviated in the following `opp`, `fat`, and `wat` respectively.

Our method has four main steps, illustrated in Fig. 5:

- Step 1: obtain some prior knowledge about IVDs localization, i.e., get a 2D region of interest (ROI) for each IVD;
- Step 2: prepare a 3D "input" volume from the volumes corresponding to the 3 modalities;
- Step 3: identify shapes that correspond to IVDs in the set of "input" slices, using the ROIs as localization constraints;
- Step 4: regularize the output in 3D.

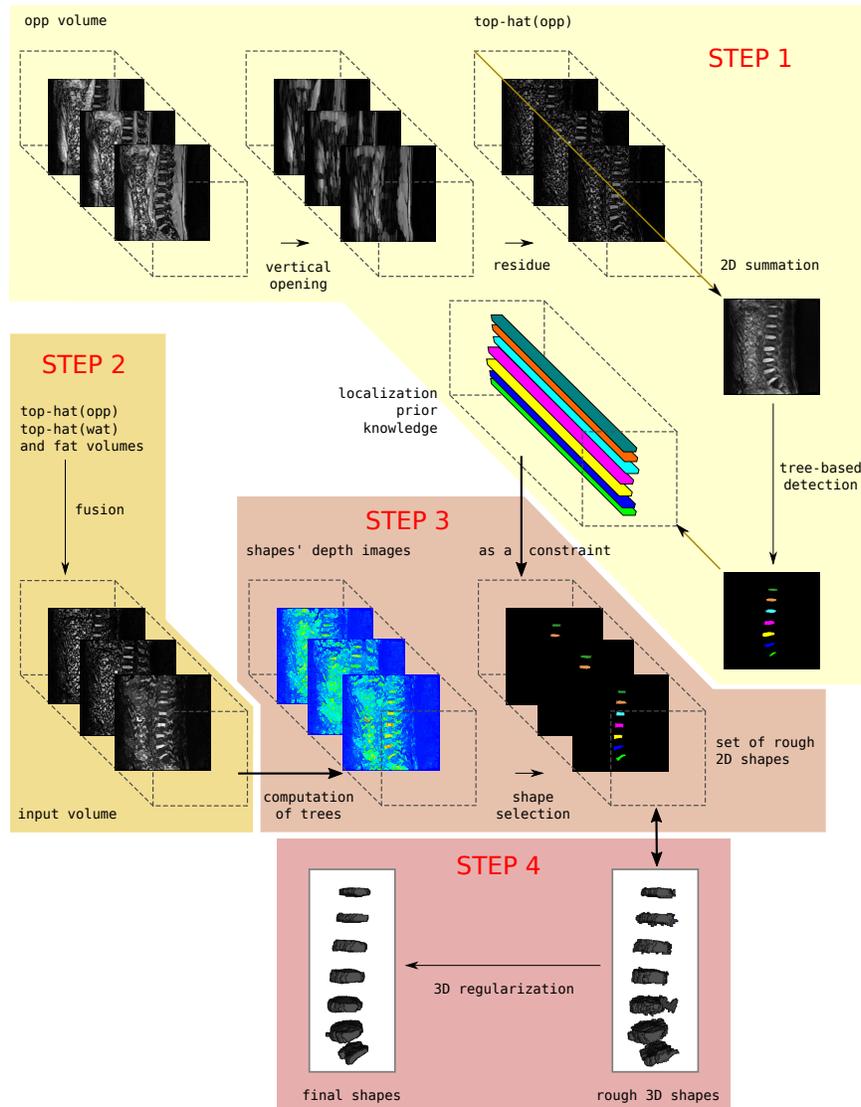These steps are described in the next four sections.

Fig. 5. Scheme of our method.

3.1 Obtaining prior knowledge about IVDs localization

The first step of the method aims at getting a gross estimation of the IVDs in 3D which will be refined later. At this stage, we do not need a precision at pixel level, only the bounding box of the IVDs.

Image preprocessing. The method works with the *opp* volume only. In slices which reveal the IVDs the most, IVDs appear as bright oriented blobs which are
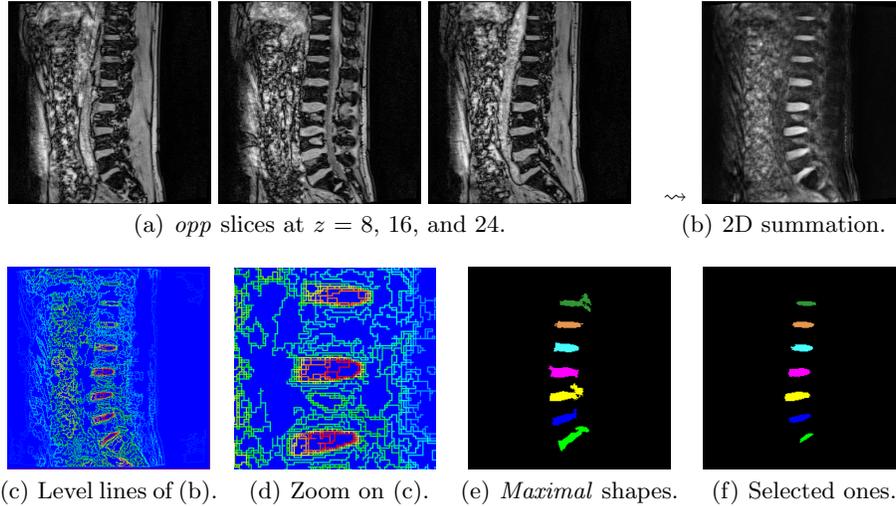
(a) *opp* slices at $z = 8$, 16, and 24.   (b) 2D summation.

(c) Level lines of (b).   (d) Zoom on (c).   (e) *Maximal* shapes.   (f) Selected ones.

Fig. 6. Step 1: Obtaining localization prior knowledge.

at least 7-pixels high. Thus, for each slice, a top-hat (as described in Section 2.2) with a flat vertical structuring element of size 15×1 allows filtering out the background and highlights the IVDs. Then, the slices are summed up (similar to an Average Intensity Projection along the z-azis) to produce a consensus image. The projection serves as a Temporal Noise Reduction to reduce noisy structures that could have passed the top-hat filtering in some frames. Fig. 6 (b) shows the result of the preprocesssing of a volume whose slices are shown in Fig. 6 (a).

IVD selection. The method computes the Tree of Shapes (ToS) on the preprocessed image. The latter enables a hierarchical representation of the inclusion of the hole-filled connected components of the image. The tree is then filtered by some prior-knowledge-based basic criteria:

- bounding box size and position of the shape
- position of the center of the shape
- orientation of the shape
- height of the shape
- average gray level of the shape

Only about 20 maximal (i.e., not included in any other shape) shapes S_i are able to pass these requirements but have non-regular contours. To overcome this problem, we then look for the sub-shapes S_i^* the most compact (i.e., maximizing the ratio of the surface over the enclosing oriented rectangle surface) included in the maximal shapes S_i.

From this set of candidates, we then need to select only 7 of them—because exactly 7 IVDs are expected for the challenge. The candidates are sorted by
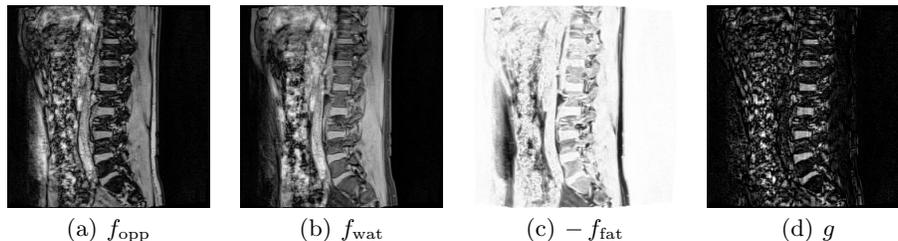
(a) f_{opp} (b) f_{wat} (c) $-f_{\mathrm{fat}}$ (d) g

Fig. 7. Step 2: Creation of a 3D volume from three different modalities.

decreasing average gray value (remind that IVDs appear very bright in the pre-processed image). The brightest shape serves as a reference and is augmented with shapes taken from S_i^* satisfying some relative positioning constraints:

- the y-distance between the shape center and the current bounding box is between 15 and 45 pixels
- the x-distance between the shape center and top/bottom selected shapes is below 15 pixels

Fig. 6 (e) and (f) illustrate the 7 *maximal* and *regular* shapes retained by our shape selection algorithm. From these shapes, we extract the 7 Region of Interests (ROI) as the bounding boxes of the selected shapes. These ROIs and shape center will be used as markers in Step 3.

3.2   Preparing a 3D "input" Volume

The previously detected seeds are used to guide the search in the 2D slices. We are now going to work on an image combining the `opp`, `fat` and `wat` modalities, as IVDs contours may be spread among these images. To that aim, the top-hat filtering is used to enhance the contrast of IVDs. The combination of the 3 volumes is given by:

$$g = \kappa_B(f_{\mathrm{opp}}) + \kappa_B(f_{\mathrm{wat}}) - f_{\mathrm{wat}}, \tag{11}$$

and is illustrated in Fig. 7.

3.3   Identifying Shapes of IVDs in 2D Slices

This step is very similar to the *IDV Selection* process of Step 2 described in Section 3.1. A ToS is computed on each slice of the 3D input volume. In each ROI of the IVDs localized previously, we look for the best *regular* shape passing some basic geometric criteria (min/max size, bounding box, minimum intensity...). Note that for an IVD ROI, there may not exist such shape, as some IVDs might not be visible in some slices.

3.4 3D Regularization

Z-axis regularization: In some slices, when no shape can be found for a given IVD, it may be normal but also might be a missed detection. If a pixel (x, y) is labeled at $z = k - 1$ and $z = k + 1$, but not at slice $z = k$, it is likely a miss-detection. As a consequence, the regularization applies:

$$f(z, x, y) = f(z, x, y) \vee (f(z - 1, x, y) \wedge f(z + 1, x, y))$$

3D shape regularization: In each 2D slice, shapes are quite *regular* because of the shape selection algorithm that favors regular contours. On the contrary, back in 3D, the concatenation of 2D results has no 3D coherence. To tackle this problem, a structural opening followed by a structural closing with a small 3D ball allows to remove contour irregularities.

Isolated pixels removal: While the *z-axis regularization* tackles the missed-detection problem, false-detections may appear due to some natural noise (especially at the beginning and the end of the sequence). These shapes are generally disconnected in 3D from the real IVDs. Thus, as a final step, we perform a 3D connected component labeling and only retain the 7 largest ones.

Step 4 in Fig. 5 illustrates the 3D regularization of the shapes performed by our method.

	Dice
mean	0.881
sd	0.025
min	0.852
max	0.927

(a) On the training set.

	IVDs						
	1	2	3	4	5	6	7
1	0.784	—	—	0.858	0.589	0.850	0.659
2	0.899	0.416	—	0.916	0.902	0.892	0.706
3	0.742	—	—	0.787	0.815	0.732	0.601
4	0.813	0.908	0.910	0.837	0.832	0.793	0.763
5	—	0.805	0.911	0.879	0.864	0.875	0.659
6	—	0.723	0.902	0.883	0.882	0.857	0.635
7	0.839	0.902	0.900	0.888	0.872	0.839	0.784
8	0.847	0.896	0.892	0.897	0.890	0.863	—
mean	0.821	0.775	0.903	0.868	0.831	0.838	0.687

(b) Dice values on the 8 cases of the test set.

Fig. 8. Quantitative results on the challenge data sets.

4 Results

First we have run our method on the 16 cases of the challenge training set. We can observe in Fig. 8(a) that the average Dice value of **0.881** is good, with a very low standard deviation. On the 8 cases of the test set, the different rows in
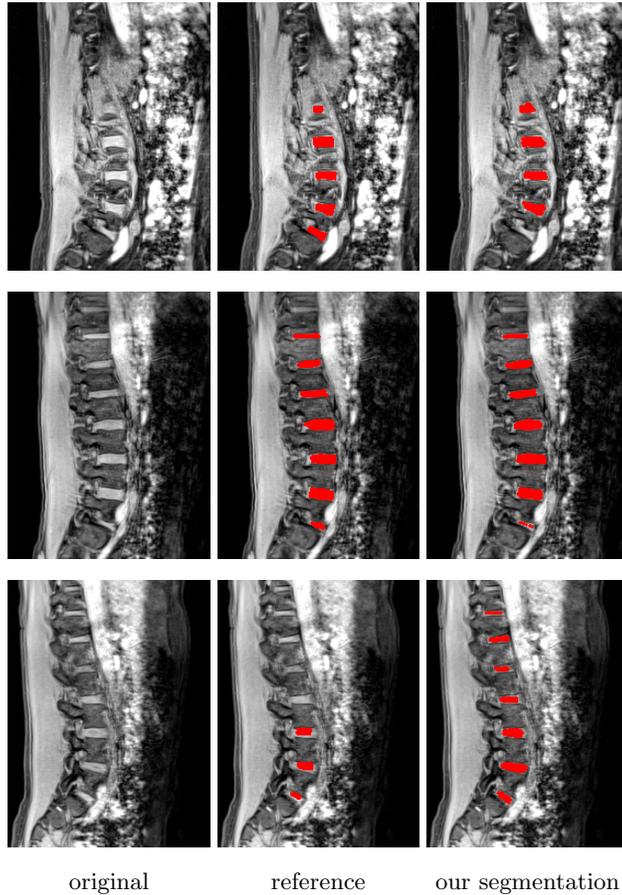
original   reference   our segmentation

Fig. 9. Some qualitative results on selected slices, respectively taken from training samples #6 (top), #14 (middle), and #16 (bottom).

Fig. 8(b), we miss some IVDs (which is symbolized by "—" in the table). Since we do not have access to the data of the test set, we cannot figure out what makes our method fail for these few IVDs. Yet, for the ones we segment, the Dice values are satisfactory, with an overall average Dice of **0.816**. Last, some qualitative results, compared to the reference images provided by the challenge organizers, are depicted in Fig. 9.

5 Conclusion

We have presented a mathematical morpholology-based of the IVD segmentation problem. This method which is a machine-learning free, only relies on a chain of some simple morphological processing blocks. Despite being a learning-free approach, we have shown that it is able to compete with new CNN-based methods

(but still perform worst when looking at the metrics only). On the other hand, the strength of our method lies in its speed. Only few seconds are required to process a whole volume with a single-threaded desktop processor, where CNN-based methods would be several order of magnitude slower. Yet, our implementation does not take benefit neither from a straightforward parallelization of the 2D slices processing, nor from parallel implementations of the tree of shapes [11,6].

References

1. Ben Ayed, I., Punithakumar, K., Garvin, G., Romano, W., Li, S.: Graph cuts with invariant object-interaction priors: Application to intervertebral disc segmentation. Inf Process Med Imaging pp. 221–232 (2011)
2. Cao, F., Lisani, J.L., Morel, J.M., Musé, P., Sur, F.: A Theory of Shape Identification, Lecture Notes in Mathematics, vol. 1948. Springer (2008)
3. Carlinet, E., Géraud, T.: A comparative review of component tree computation algorithms. IEEE Trans. on Image Processing 23(9), 3885–3895 (2014)
4. Carlinet, E., Géraud, T.: Morphological object picking based on the color tree of shapes. In: Proceedings of the 5th International Conference on Image Processing Theory, Tools and Applications (IPTA). pp. 125–130. Orléans, France (November 2015)
5. Carlinet, E., Géraud, T.: MToS: A tree of shapes for multivariate images. IEEE Transactions on Image Processing 24(12), 5330–5342 (2015)
6. Carlinet, E., Géraud, T., Crozet, S.: The tree of shapes turned into a max-tree: A simple and efficient linear algorithm. In: Proceedings of the 24th IEEE International Conference on Image Processing (ICIP). pp. 1488–1492. Athens, Greece (October 2018)
7. Caselles, V., Coll, B., Morel, J.M.: Topographic maps and local contrast changes in natural images. International Journal of Computer Vision 33(1), 5–27 (1999)
8. Caselles, V., Monasse, P.: Grain filters. Journal of Mathematical Imaging and Vision 17(3), 249–270 (2002)
9. Caselles, V., Monasse, P.: Geometric Description of Images as Topographic Maps, Lecture Notes in Mathematics, vol. 1984. Springer-Verlag (2009)
10. Chen, C., Belavy, D., Yu, W., Chu, C., Armbrecht, G., Bansmann, M., Felsenberg, D., Zheng, G.: Localization and segmentation of 3D intervertebral discs in MR images by data driven estimation. IEEE Transactions on Medical Imaging 34(8), 1719–1729 (August 2015)
11. Crozet, S., Géraud, T.: A first parallel algorithm to compute the morphological tree of shapes of nD images. In: Proceedings of the 21st IEEE International Conference on Image Processing (ICIP). pp. 2933–2937. Paris, France (2014)
12. Géraud, T., Carlinet, E., Crozet, S., Najman, L.: A quasi-linear algorithm to compute the tree of shapes of n-D images. In: Proc. of the Intl. Symp. on Mathematical Morphology (ISMM). Lecture Notes in Computer Science, vol. 7883, pp. 98–110. Springer, Uppsala, Sweden (2013)
13. Monasse, P., Guichard, F.: Fast computation of a contrast-invariant image representation. IEEE Transactions on Image Processing 9(5), 860–872 (2000)
14. Neubert, A., Fripp, J., Engstrom, C., Walker, D., Weber, M.: Three-dimensional morphological and signal intensity features for detection of intervertebral disc degeneration from magnetic resonance images. J Am Med Inform Assoc 20(6), 1082–1090 (Novembre-December 2013)

15. Stern, D., Likar, B., Pernus, F., Vrtovec, T.: Automated detection of spinal centrelines, vertebral bodies and intervertebral discs in CT and MR images of lumbar spine. Phys Med Biol. 55(1), 247–264 (January 2010)
16. Xu, Y., Carlinet, E., Géraud, T., Najman, L.: Hierarchical segmentation using tree-based shape spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(3), 457–469 (April 2017)
17. Xu, Y., Géraud, T., Najman, L.: Connected filtering on tree-based shape-spaces. IEEE Trans. on Pattern Analysis and Machine Intelligence 38(6), 1126–1140 (2016)
18. Xu, Y., Géraud, T., Najman, L.: Hierarchical image simplification and segmentation based on Mumford-Shah-salient level line selection. Pattern Recognition Letters 83(3), 278–286 (November 2016)
19. Xu, Y., Monasse, P., Géraud, T., Najman, L.: Tree-based morse regions: A topological approach to local feature detection. IEEE Transactions on Image Processing 23(12), 5612–5625 (December 2014)