Marko Rak[1,*], Johannes Steffen[1,*], Anneke Meyer[1], Christian Hansen[1], Klaus-Dietz Tönnies[1]

Combining Convolutional Neural Networks and Star Convex Cuts for Fast Whole Spine Vertebra Segmentation in MRI

Pre-Print

[1] Department of Simulation and Graphics, University of Magdeburg, Germany

[*] Contributed equally

Combining Convolutional Neural Networks and Star Convex Cuts for Fast Whole Spine Vertebra Segmentation in MRI

Marko Rak[1],*, Johannes Steffen*, Anneke Meyer, Christian Hansen, Klaus–Dietz Tönnies

Department of Simulation and Graphics, University of Magdeburg, Germany

Abstract

Background and Objective: We propose an automatic approach for fast vertebral body segmentation in three-dimensional magnetic resonance images of the whole spine. Previous works are limited to the lower thoracolumbar section and often take minutes to compute, which is problematic in clinical routine, for study data sets with numerous subjects or when the cervical or upper thoracic spine is to be analyzed.

Methods: We address these limitations by a novel graph cut formulation based on vertebra patches extracted along the spine. For each patch, our formulation incorporates appearance and shape information derived from a task-specific convolutional neural network as well as star-convexity constraints that ensure a topologically correct segmentation of each vertebra. When segmenting vertebrae individually, ambiguities will occur due to overlapping segmentations of adjacent vertebrae. We tackle this problem by novel non-overlap constraints between neighboring patches based on so-called encoding swaps. The latter allow us to obtain a globally optimal multi-label segmentation of all vertebrae in polynomial time.

Results: We validated our approach on two data sets. The first contains T_1- and T_2-weighted whole spine images of 64 subjects with varying health conditions. The second comprises 23 T_2-weighted thoracolumbar images of young healthy adults and is publicly available. Our method yielded Dice coefficients of $93.8 \pm 2.6\,\%$ and $96.0 \pm 1.0\,\%$ for both data sets with a run time of $1.35 \pm 0.08\,$s and $0.90 \pm 0.03\,$s per vertebra on consumer hardware. A complete whole spine segmentation took $32.4 \pm 1.92\,$s on average.

Conclusions: Our results are superior to those of previous works at a fraction of their run time, which illustrates the efficiency and effectiveness of our whole spine segmentation approach.

Keywords: Magnetic Resonance, Spine Analysis, Vertebra Segmentation, Graph Cuts, Neural Networks

1. Introduction

Due to its soft-tissue contrast, magnetic resonance imaging has become a valuable non-invasive

*Contributed equally

Email address: rak@isg.cs.ovgu.de (Marko Rak)

[1] Department of Simulation and Graphics
University of Magdeburg
Universitätsplatz 2
39106 Magdeburg, Germany

tool for the analysis of the spine both in clinical routine and in study contexts. Potential investigations include measurements of so-called Cobb angles for the rating of kyphosis/scoliosis, assessments related to vertebra morphometry and the identification of compression fractures such as crushed/wedged vertebrae. The rising clinical interest in magnetic resonance-based analysis has led to a number of works on automatic and semi-automatic segmentation of vertebral bodies (simply called vertebrae hereafter), both model- and data-driven. We will now go into some detail on relevant related works. For an in-depth discussion we refer to the comprehensive surveys of [1] and [2].

A first model-driven approach was presented in [3], who use a superquadrics-based parameteric shape model that adapts to a nearby vertebra based on the intensity information in a local neighborhood of the model. Alternatively, [4] employ balloon forces to inflate a surface mesh with smoothness constraints directly inside the vertebra. Both approaches may lead to ambiguities between adjacent vertebrae. To cope with this issue, [5] arranged multiple adjacent vertebrae into a single elastic finite element model, which adapts to the data via forces derived from the nearby image content.

Statistical shape modeling with standard active shape models was used in [6] and later by [7] to fit each vertebra individually. The concept was generalized to part-based models by [8] and [9], who include shape and pose relations between multiple vertebrae to avoid any ambiguities. Both also used non-linear mappings to improve their shape space representation compared to standard active shape modeling. Rather recently, [10] showed encouraging results by linking single vertebra active shape models with vertebra likelihood maps generated from a convolutional neural networks.

Data-driven techniques are most often patch-based, meaning that the segmentation is performed in a small neighborhood around each vertebra. To this end, [11] match a cubically-shaped template deformably to a nearby vertebra via a graph cut optimization framework. A similary strategy was used in [12], where interactive graph cuts were used to carve out the central vertebrae of user-supplied patches. In [13] geodesic active contours and the Chan-Vese intensity model are combined into a level set-based segmentation technique. The same approach was later reused and sped-up in [14].

A data-drive machine learning-based approach is presented in [15]. They combined appearance learned via random forests with shape information estimated via Parzen windows into a vertebra probability map, which is then thresholded. Another learning-based strategy was proposed in [16], who first decompose the image into super-voxels from which appearance and shape features get extracted. These features are used to train a random forest-based super-voxel classifier. A similar strategy was presented in [17], who utilized random forests on different image resolutions to cope with differently-sized vertebrae. [18] demonstrated that also U-Net-like convoluational neural networks can be used for vertebra segmentation with great success.

In purely two-dimensional mid-sagittal settings, data-driven techniques are typically applied to the whole image rather than individual vertebra patches. For instance, [19] use decision trees to combine appearance, shape and pose information

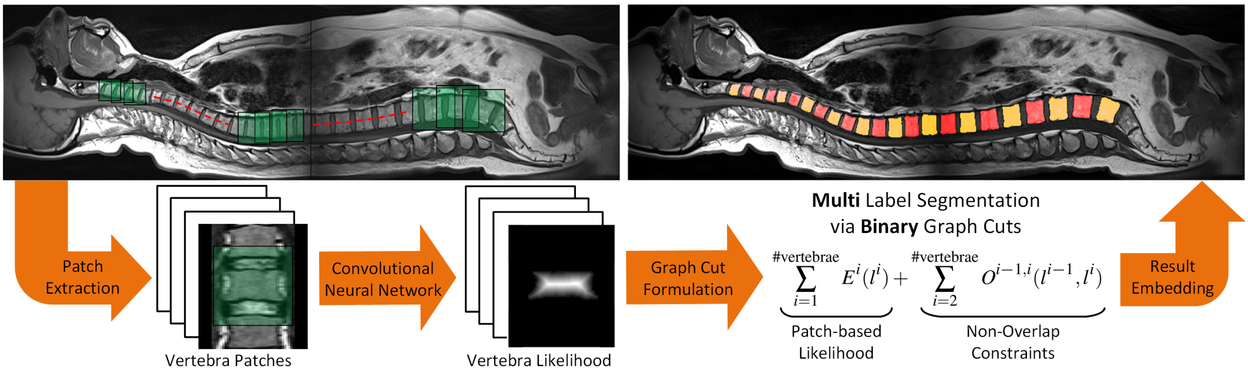
Patch Extraction
Vertebra Patches
Convolutional Neural Network
Vertebra Likelihood
Graph Cut Formulation
Multi Label Segmentation via Binary Graph Cuts
$\sum_{i=1}^{\#\text{vertebrae}} E^i(l^i) + \sum_{i=2}^{\#\text{vertebrae}} O^{i-1,i}(l^{i-1}, l^i)$
Patch-based Likelihood
Non-Overlap Constraints
Result Embedding

Figure 1: Our patch-based framework for whole spine vertebra segmentation. Image patches (green squares) are extracted for each vertebra (top left). Then, a vertebra likelihood map is computed via a task-specific convolutional neural network on patch-level. Afterwards, patch-based segmentation is applied, whereby we use a graph cut formulation to combine the likelihood maps $E^i(\cdot)$ of all vertebra patches i with non-overlap constraints $O^{i,j}(\cdot)$ between any two adjacent vertebra patches i and j to prohibit ambiguities between neighboring vertebrae. This strategy is both effective and efficient, because we are able to solve the multi-label segmentation problem via a binary graph cut formulation. Finally, the resulting segmentation is re-embedded into the image domain (top right). The vertebra segmentation is color-coded to ease the differentiation between neighboring vertebrae. Please note that there is considerable overlap between the neighboring image patches (left). Please also note that for visualization purposes, we varied the patch size from head to foot (left), while for our implementation we actually use fixed-size patches for the whole spine.

into a random field inference task, which is solved approximately by Gibbs sampling. An augmented Lagrangian method is present in [20]. Thereby distributions of vertebrae appearance features are matched to a known reference distribution to differentiate between vertebra and non-vertebra tissue. Normalized cuts were used in [21] to segment multiple vertebrae at once. They also introduced a spatial smoothness term to ensure the compactness of each vertebra. In [22], vertebrae are segmented via a pipeline composed of fuzzy C-means clustering and morphological postprocessing.

To support a wide range of applications, vertebra segmentation techniques should apply to different imaging sequences and to the whole spine. They should be reasonably fast, because time may become a critical resource in clinical routine or for study data sets with numerous subjects. These challenges are often overlooked in previous works, which are limited to the lower thoracolumbar section of the spine, easily take minutes to compute and typically apply to a single sequence only. To the best of our knowledge, we are the first to address all named challenges in a single segmentation framework.

2. Method

We contribute a novel binary graph cut formulation, which fuses patch-based star convex vertebra segmentation with non-overlap constraints between adjacent patches, ensuring topological correctness for each and between adjacent vertebrae. Akin to previous works, our formulation involves information about vertebra appearance and shape. This work is based on our earlier work [23], where we showed that engineered appearance and shape features can compete with recent machine learning-based methods if integrated into our graph cut formulation. Within this work, we go beyond that, showing that superior results can be obtained when integrating appearance and shape information by an end-to-end convolutional neural network on patch level. Otherwise put, we replace the handcrafted appearance and shape features of our earlier work [23] by an end-to-end trainable convolutional neural network and demonstrate how this can be integrated effectively into our graph cut formulation to yield a topologically correct segmentations.

Our novel combination of graph cuts and neural networks ranges midway between the two straightforward solutions: (a) an end-to-end trainable network that applies to the image as a whole and (b) a sliding window network for voxel-wise prediction. Solution (a) requires a large training data set to capture the topological relations between multiple adjacent vertebrae. Solution (b) requires rather little training data, but would perform worse since neither the appearance nor the shape of a single vertebra is captured completely. In our method, topological relations do not require any training, because they are handled by the graph cut framework explicitly. Therefore, our patch-based neural network can focus on the appearance and shape of individual vertebrae, which eases the training and requires only a moderately sized training data set. In the remainder of the section, we first outline necessary preprocessing steps. Afterwards, we introduce our novel graph cut formulation and go into detail on the relations between global optimality and topological guarantees. Finally, we show how to set up and integrate a task-specific convolution neural network into the overall problem formulation.

2.1. Preprocessing

Vertebrae segmentation is typically applied only after a vertebra localization, which is either based on user-interaction [3, 8, 5, 9, 13, 10, 14] or on automatic vertebra detectors [6, 4, 15]. Akin to previous works, we interpret vertebrae localization as a preprocessing step, for which many valuable techniques exist. We use our fast whole spine detector [24] and refer to the comprehensive surveys of [1] and [2] for a discussion of other approaches. Our detector utilizes the Kullback-Leibler divergence to model the appearance of neighboring vertebrae, which makes it suitable for localization in T_1- and T_2-weighted images alike. Specifically, we exploit the fact that vertebrae are homogeneous and neighboring vertebrae look similar to each other. Appearance information is complemented by information about the spine geometry, i.e. geometrical relations between neighboring vertebrae. This directly leads to an inference task on a second-order graphical model, which can be solved efficiently via belief propagation. Please see [24] for further details on vertebra localization.

Based on the vertebra localization, we extract cubically-shaped vertebra-centered patches for the whole spine as outlined in Figure 1. Patch-based strategies significantly reduce the problem size compared to whole-image segmentation and ease modeling by focusing on individual vertebrae. However, ambiguities can arise for close targets when patch-wise results are re-embedded into the image domain, which is especially true if vertebrae are not well-separated by intervertebral discs. In what follows, we first detail our patch-wise formulation and show how to combine the patch-wise tasks into a joint ambiguity-free formulation afterwards.

2.2. Patch-wise Formulation

For each extracted vertebra patch, we interpret the segmentation of its central vertebra as an energy minimization problem. In particular, we seek a binary labeling $l \in \{0,1\}^{|\mathcal{P}|}$ of the voxels $p \in \mathcal{P}$ of the patch into foreground ($l_p = 1$), i.e. voxels inside the central vertebra, and background ($l_p = 0$), i.e. voxels outside of it, that minimizes

$$E(l) = \underbrace{\sum_{p \in \mathcal{P}} U_p(l_p)}_{\text{Appearance \& Shape}} + \underbrace{\sum_{(p,q) \in \mathcal{C}} C_{pq}(l_p, l_q)}_{\text{Star-Convexity}} . \quad (1)$$

Our model involves soft priors for vertebra appearance and shape as well as hard constraints that ensure a star convex shaped segmentation of the vertebra. The edge set $\mathcal{C}$ comprises the ordered voxel pairs (p,q) that are linked by star-convexity constraints; this will be discussed later.

Please note that we will design each term such that the resulting energy is graph-representable, in which case the minimization of Equation 1 takes only $\mathcal{O}(\#\text{voxels} \cdot \#\text{edges}^2)$ time, cf. [25]. For the problem to be graph-representable, all pairwise terms $T(l_i, l_j)$, i.e. the star-convexity constraints in Equation 1, have to obey $T(0,0) + T(1,1) \leq T(0,1) + T(1,0)$ [26]. This essentially means that the assignment of different labels should not be cheaper than the assignment of similar ones. We now discuss each term in greater detail.

2.2.1. Appearance and Shape

Recent works encode appearance information by machine learning techniques like decision trees [19], random forests [15, 16] and convolutional neural networks [10]. We apply machine learning too. However, we address both appearance and shape at the same time, which greatly simplifies the modeling compared to previous works. Specifically, we learn only a single unary term to capture appearance and shape information at the same time. Our unary term reads

$$U_p(l_p) = [l_p = 0] \cdot \underbrace{\begin{cases} d_p(\partial\Omega) & d \in \Omega \\ -d_p(\partial\Omega) & d \notin \Omega \end{cases}}_{u_p}, \quad (2)$$

whereby Iverson brackets $[\cdot]$ select only the background label, which implies zero costs for the foreground. The background costs depend on the Euclidean distance d_p of the voxel d to the boundary $\partial\Omega$ of a preliminary segmented region Ω. Term u_p is a so-called signed distance function, which gives increasingly positive values towards the interior of Ω and decreasingly negative values towards the border of the vertebra patch. The values of u_p can be computed efficiently by solving the Eikonal equation with respect to Ω. To this end, we use the fast marching algorithm [27], which takes quasilinear time when implemented with a heap data structure.

Our combined appearance and shape prior is illustrated in Figure 2. The term favors foreground

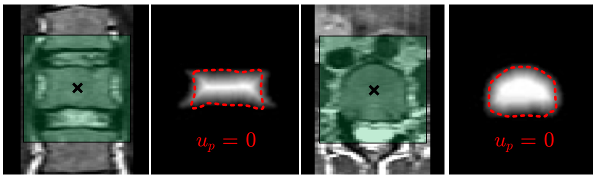

Figure 2: Our combined appearance and shape prior on coronal (first and second image) and axial (third and fourth image) slices of a particular vertebra patch (green squares). Ideally, the unary costs u_p (second and fourth image) are positive in the foreground, i.e. inside the central vertebra, and negative for most of the background. If our combined prior would be considered on its own, then the energy minimization would essentially be a voxel-wise thresholding at the pivot point $u_p = 0$. Please note that we oversimplified the thresholding (red curves) for illustration purposes.

when Equation 2 becomes positive and background otherwise. The pivot point is at a distance of $d_p = 0$, which means that the preliminary segmented region Ω should ideally reflect only the central vertebra of the patch. However, this is hard to guarantee, which will be shown in the experiments. In this work, the presegmentation Ω is computed by our convolutional neural network, which - for the sake of argumentation - will be introduced in the end of this section.

2.2.2. Star-Convexity

We employ the star-convexity constraints of [28] to ensure a topologically correct segmentation of the vertebra. In particular, we will restrict the segmentation to a single connected component without holes in the center of the patch. Thus leakage into adjacent vertebrae and other neighboring structures becomes less likely. To this end, we enforce that for any foreground voxel r, every other voxel p on the line segment between r and the center of the patch c is foreground too. The idea is illustrated in Figure 3.

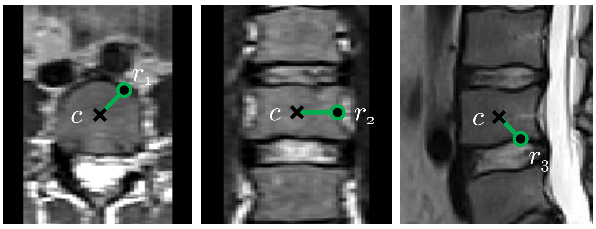

Figure 3: Star-convexity constraints on axial (left), coronal (middle) and sagittal (right) slices. If voxels r_i (dots) were assigned to the foreground, then any other voxel on the line segment (green) to the center voxel c (crosses) would be foreground too. The resulting segmentation consists of a single connected component without holes in the center of the vertebra patch.

Star-convexity may be implemented by rasterization. To this end, the line segment between the center c of the patch and every voxel r is transformed into a sequence of voxels $(c, \ldots, p, q, \ldots r)$. Then each pair of neighboring voxels of the sequence is "tied" together by hard constraints. Given two such voxels p and q the star-convexity constraints read

$$C_{pq}(l_p, l_q) = [l_p = 0 \land l_q = 1] \cdot \infty, \tag{3}$$

where Iverson brackets $[\,\cdot\,]$ assign infinite costs when foreground shall be assigned after some background voxel, cf. [28].[2] The rasterization does not need to be computed "online" during segmentation, because it is independent of the image content and of the voxel size.

We precompute the rasterization and, thus, all pairs (p, q) of voxels via Bresenham's line algorithm [29] on a sufficiently large reference patch. The pairs are then loaded before the segmentation and cropped to the particular image patch, which leads

[2] Please note that ∧ is the logical operation *and*.

to an efficient implementation of star-convexity.

2.3. Joint Formulation

For close targets such as adjacent vertebrae, ensuring star-convexity is challenging. As outlined in Figure 4, naive binary whole image formulations will fuse targets and multi-label formulations are less performant and do not come with the same optimality guarantees as binary tasks. Our patch-based binary formulation has neither problem, but ambiguities may arise when results are re-embedded into the image domain. This is especially true when vertebrae are not well-separated by large intervertebral discs like in the cervical and thoracic section of the spine. Here the patch-based energy could favor to bridge the thin gap to the neighboring vertebrae.

To circumvent the ambiguities and preserve optimality, we combine the patch-wise formulations into a joint binary minimization problem with topological constraints that guarantee a non-overlapping segmentations between adjacent vertebra patches. In particular, we seek a combination of patch-wise labelings that minimizes

$$E\left(l^1,\dots,l^{\#\text{vertebrae}}\right) = \underbrace{\sum_{i=1}^{\#\text{vertebrae}} E^i\left(l^i\right)}_{\text{Patch Energy}} + \underbrace{\sum_{i=2}^{\#\text{vertebrae}} O^{i-1,i}\left(l^{i-1},l^i\right)}_{\text{Non-Overlap}}, \quad (4)$$

where the introduced superscripts enumerate all vertebra patches from head to foot (or vice versa). The first sum pools the already introduced energies of the individual vertebra patches and the second sum handles the regions where neighboring vertebra patches overlap. We now discuss the implementation of the latter in greater detail.

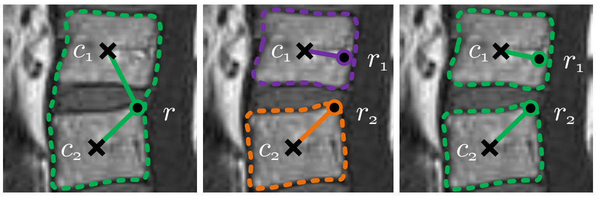

Figure 4: Star-convexity implementation for neighboring vertebrae. A binary whole image formulation (left) will fuse both vertebrae (green region). The rationale is that any voxel r will be "tied" to both centers c_1 and c_2 simultaneously. Multi-label formulations (middle) avoid this issue by imposing star-convexity for each label individually, but optimality guarantees of binary graph cuts are lost. Our binary patch-wise formulation (right) has neither problem, because the vertebra patches are treated individually.

2.3.1. Direct Solution

For any such overlap region, only one of the two involved vertebra patches shall be allowed to assign foreground to a shared voxel, because otherwise ambiguities can occur. Hence, we seek to establish binary terms between the shared voxels of both vertebra patches to prohibit such situations. The non-overlap requirement of the patch-wise labelings is equivalent to the following hard constraints

$$O^{ij}\left(l^i,l^j\right) = \sum_{(p,q)\in\mathcal{O}^{ij}} \underbrace{\left[l^i_p = 1 \wedge l^j_q = 1\right]\cdot\infty}_{T^{ij}\left(l^i_p,l^j_q\right)}, \quad (5)$$

where Iverson brackets $[\cdot]$ assign infinite costs when both patches select foreground for a shared voxel.[3] Set $\mathcal{O}^{ij}$ comprises the voxels shared by both patches. In particular, it contains the ordered pairs (p,q) of voxels p from patch i and voxels q

[3] Please see Footnote 2.

from patch j that represent the same voxel after re-embedding of patches into the image domain.

The constraints presented in Equation 5 cannot be realized with graph cuts directly, because it is not graph-representable in its current form. In particular, the $T^{ij}(\cdot)$ violate $T(0,0) + T(1,1) \leq T(0,1) + T(1,0)$ [26], which essentially means that the assignment of different labels should not be cheaper than the assignment of similar ones. We can, however, derive an equivalent formulation that is indeed graph-representable.

2.3.2. Encoding Swaps

We go back to our representation of foreground and background, which are implemented by $l_p^i = 1$ and $l_p^i = 0$, respectively. Noticing that this is a convention, we could just as well have swapped the meaning of labels. Specifically, we could have encoded foreground with $l_p^{\tilde{i}} = 0$ and background with $l_p^{\tilde{i}} = 1$, whereby the introduced tilde differentiates between the swapped and the standard encoding.

Obviously, swapping encodings for every vertebra patch does not solve the problem, because the resulting non-overlap constraints are not graph-representable either. However, when we swap only every other vertebra patch from head to foot, then the hard constraints of Equation 5 change to

$$T^{i\tilde{j}}\left(l_p^i, l_q^{\tilde{j}}\right) = \left[l_p^i = 1 \wedge l_q^{\tilde{j}} = 0\right] \cdot \infty, \qquad (6)$$

where standard-encoded patch i overlaps with encoding-swapped patch $\tilde{j}$.[4] It is easy to verify that $T^{i\tilde{j}}(\cdot)$ as well as its straightforward counterpart $T^{\tilde{i}j}(\cdot)$ obey $T(0,0) + T(1,1) \leq T(0,1) + T(1,0)$. The patch energies remain unaffected by the encoding swap if the unary terms as well as star-convexity constraints are adjusted (swapped) accordingly. Please note that the argument does not contradict the graph-representability of [26]; it rather exploits the available degrees of freedom. Eventually, we compute the optimal labeling by the algorithm of [25]. In particular, we use the implementation that is provided by the Darwin framework [30].

The concept of encoding swaps is not limited to our particular application. Let each image patch be a node in a graph and let each overlap region between two patches be an edge between their associated graph nodes. In this notation, every bipartite graph, i.e. every graph that has a two-coloring, can benefit from an efficient implementation of non-overlap constraints via encoding swaps. Specifically, chain-like overlap layouts like ours are covered, but also all forms of tree-like layouts and certain grid-like overlap layouts too.

2.4. Neural Network

2.4.1. General Outline

So far we did not cover the computation of our patch-wise presegmentation Ω. The latter shall capture both appearance and shape information of the central vertebra of each patch. To this end, we follow the U-Net concept [31] and its extension [32] to three-dimensional domains. Other network architectures are reasonable too. For instance, one could use a standard encoder/decoder concept attaching links on particular levels of the hierarchy [33] or use pre-trained networks to improve learning [34] as often done in general computer vision. For

[4] Please see Footnote 2.

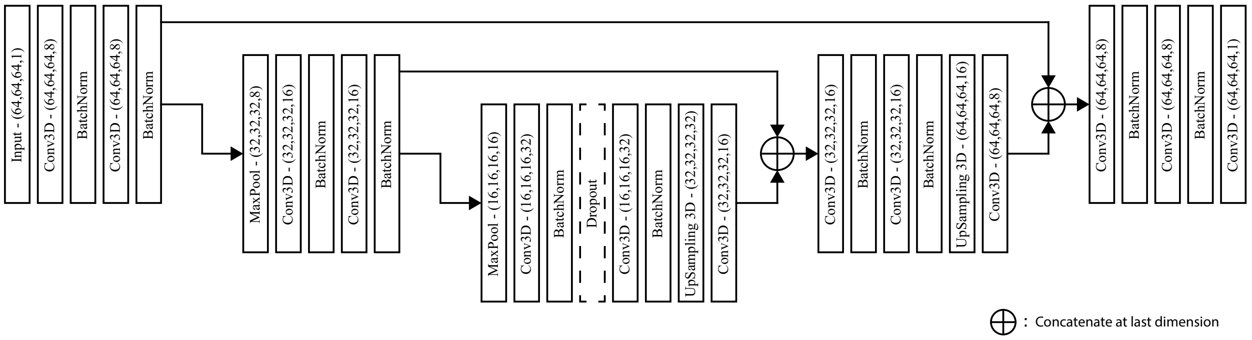

Figure 5: Schematic overview of our presegmentation network, which maps an extracted vertebra patch (first layer) to the corresponding presegmentation result (last layer). The former contains the actual image content and the latter uses a standard binary representation. The architecture closely follows the popular U-Net concept and its extension to three-dimensional domains. We simplified the complexity of our network, making it more suitable for the vertebra segmentation task.

medical applications however, the U-Net architecture is successful in many applications and models are typically trained from scratch [35].

Our end-to-end presegmentation network is depicted in Figure 5. Akin to the standard U-Net, our neural network is fully convolutional and consists of two paths. Firstly, a contraction path, where the information of the vertebra patch is first reduced subsequently by interleaving several convolutional and pooling layers. Secondly, an expansive path, where the generated feature maps are upsampled again to finally obtain the sought vertebra presegmentation. To guide the expansion, additional information from the contraction path is introduced through shortcut connections.

Our rational for developing a task-specific adaptation of the original U-Net is that not only time, but also graphics hardware is a often limited in clinical routine or when analyzing study data sets with numerous subjects. Specifically, a standard three-dimensional U-Net may not be used directly, because of the memory consumption that arises from the large number of network parameters. Moreover, it is unclear whether that many parameters are necessary for our task in the first place, because vertebra appearance and shape varies rather moderately unless pathological cases are considered. We believe that reducing the network complexity is reasonable for our task, which is backed by experimental results.

2.4.2. Architectural Detail

As depicted in Figure 5, our contraction path comprises several contraction blocks, whereby each block consists of zero padding, convolutional operators of size $3 \times 3 \times 3$, batch normalization according to [36], zero padding, convolutional operators of size $3 \times 3 \times 3$ and, again, batch normalization. At the end of the contraction block, a max-pooling operation of size $2 \times 2 \times 2$ is used to downsample the resolution of the arising feature maps for the next depth level. To compensate the loss of information due to the downsampling, the number of feature maps doubles between the resolution levels [31]. We start with an initial number of eight feature maps. There are two contraction blocks in total.

Along the expansion path, the feature maps are processed in expansion blocks, whereby each block consists of upsampling, convolutional operators of size $3 \times 3 \times 3$ that halve the number of feature maps, concatenation with the output of the respective contraction block, convolutional operations of size $3 \times 3 \times 3$, batch normalization, convolutions of size $3 \times 3 \times 3$ and, again, batch normalization. There are two expansion blocks in total. To finally re-obtain the size of the vertebra patch, the last layer uses a convolutional operator of size $1 \times 1 \times 1$. We use rectified linear units as activation functions after each convolutional layer, except for the last layer, where a sigmoid activation is used to be compatible with our loss function, which is the inverse of the fuzzy Dice coefficient.

3. Experiments

3.1. Data Sets and Preprocessing

We carried out experiments on two data sets. The first data set (called DS1) comprises T_1- and T_2-weighted whole-spine images of 64 subjects with varying health conditions from the "Study of Health in Pomerania" [37]. For DS1 ground truth segmentations are available from C3 to L5. The second data set (called DS2) comprises 23 T_2-weighted thoracolumbar images of young healthy adults and is publicly available, cf. [15]. For DS2 ground truth segmentations are available from T11 to L5. Both data sets were acquired by turbo spin echo sequences on Siemens 1.5 Tesla imagers and reconstructed sagitally at 1.12 × 1.12 × 4.4 mm and 1.25 × 1.25 × 2.0 mm, respectively. To simplify the later processing, we upsampled all images with linear interpolation in mediolateral direction, yielding isotropic voxels of 1.12 × 1.12 × 1.12 mm and 1.25 × 1.25 × 1.25 mm, respectively.

After the upsampling, we applied our vertebra localization [24], which correctly detected 96.0 % of the vertebrae in DS1 at an accuracy of 3.45 ± 2.20 mm with respect to the known ground truth centers. For DS2 the detection rate was 98.1 % with 3.07 ± 1.78 mm distance to the known ground truth centers. The difference in localization quality between the data sets is mainly due to the difference in laterolateral resolution (4.4 vs. 2.2 mm). For both data set, the localizations took around one second per vertebra (Intel Core i5 @ 4×3.30 GHz). All falsely detected vertebrae were corrected manually before the actual segmentation, i.e. a user-specified vertebra center was used instead of the found location.

In the remainder of this section, we first go into detail on our patch-based presegmentation network, discussing its training and the obtained results. Afterwards, we introduce our overall results, comparing our novel graph cut-based method to previous works on magnetic resonance-based vertebrae segmentation. We assessed the standard measures of segmentation quality: the Dice coefficient (DC), the average Euclidean inter-surface distance (AD) and the Hausdorff distance (HD).

3.2. Presegmentation Network

3.2.1. Sample Generation

To obtain training samples for each data set, we extracted image patches of size $64 \times 64 \times 64$ around each of the preliminary located vertebra centers. Since the extracted patches contain multiple par-

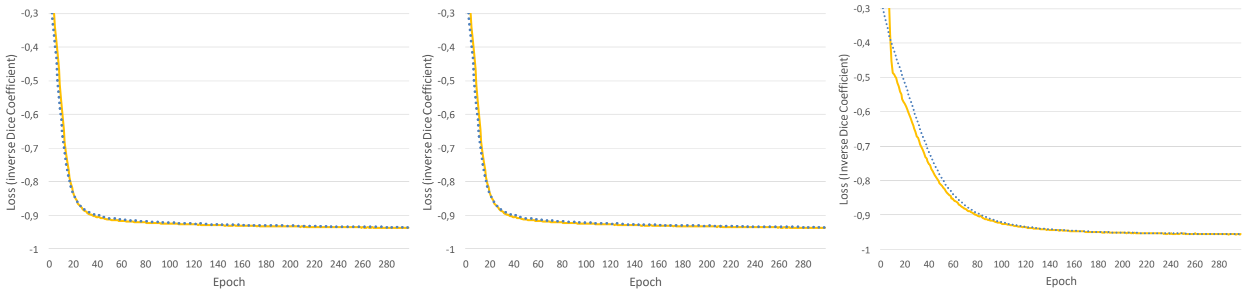

Figure 6: Comparison of fold-averaged training subset (yellow curves) and validation subset (blue curves) losses over 300 epochs for the T_1-weighted (left) and the T_1-weighted (middle) images from DS1 as well as for the T_2-weighted images from DS2 (right).

tially visible vertebrae, the respective ground truth patches were altered such that only the central vertebra is segmented for any given patch. Our rationale is two-fold. Firstly, we are mainly interested in the central vertebra for any given patch, since the adjacent vertebrae are covered by the adjacent patches in our graph cut formulation anyway. Secondly, the neural network is forced to learn not only appearance but also shape information, because the latter is the only source of information that helps discriminating between the central and the other vertebrae partially visible on each patch.

The quality of the presegmentation network and, thus, also that of our overall method may depend on the accuracy of the preliminary vertebra localization. To deal with this issue, we apply data augmentation, increasing the robustness of our network with respect to localization inaccuracies. Specifically, we extract randomly displaced image patches for each vertebra to mimic inaccuracies due to the localization. In total we augmented the training samples by three displaced patches per vertebra, whereby displacements of up to 3 voxels along each dimension of the image domain were sampled from a uniform distribution. Displacements can occur in either direction and are mixed among different dimensions of the image domain to obtain an unbiased sampling.

3.2.2. Cross-Validation

Our presegmentation network was implemented in Keras [38] using Tensorflow [39] as back end. To evaluate the quality of our presegmentation network, we applied a five-fold cross-validation on each of our data sets. Please note that we treat the T_1- and T_2-weighted images from DS1 as separate data sets within the cross-validation context. Moreover, to avoid any subject-related bias, cross-validation splits into training (4 of 5 folds) and validation (1 of 5 folds) subsets were performed on subject level rather than on vertebra level. This prohibits any situation where vertebrae of the validation subset are visible, either partially or fully, on some patch of the training subset. This is necessary because otherwise the presegmentation network could learn something about the validation subset indirectly during its training.

For each cross-validation fold, the network was trained on the four remaining folds for 300 epochs with a batch size of 12 patches utilizing the Adam optimizer [40]. After each epoch all training patches were shuffled randomly to increase robustness of the training and avoid any memorization. The learning rate of the optimizer was set to 0.0001, while all other parameters were set as suggested in [40]. To decrease the risk of over- and co-adaptation, we apply dropout [41] of 50 % in-between the contraction and expansion path; the dashed box in Figure 5. Please note that we did not use early stopping or other strategies that utilize the validation data.

All cross-validations were performed on a NVIDIA Titan Xp with 12 GB of memory, whereby the training took up to 12 hours per fold, depending on the number of subjects in the fold. When considering predictions only, then 100 patches may be processed per batch, which gives run times of only 14.12 ± 0.28 ms per vertebra patch. More importantly, due to the small number of network parameters (about 85 000), our presegmentation network may predict all vertebrae of the whole spine in one batch on any graphics card with 6 GB of memory.

3.2.3. Experimental Results

Our cross-validation results are illustrated in Figure 6. As can be seen, the loss decreases faster for DS1 than for DS2. This can be ascribed to the larger number of samples in DS1, which has almost three times as many subjects and about three times as many vertebrae per subject. Still, even for DS2 the quality has saturated at about 250 epochs, which underpins that our neural network requires only a small training data set. Over-adaptation to the training subset is not an issue either, since losses decrease at approximately the same rate for the training and validation subsets, irrespectively of the data set. We cannot find any noticeably differences between the T_1- and T_2-weighted results for DS1, which means that our network captures both sequences equally well. The obtainable quality is slightly better for DS2, which we ascribe to the fact that DS2 contains only lower thoracolumbar vertebrae, which are easier to segment due to their larger size.

If used on its own, then our presegmentation network would yield Dice coefficients of 93.4 ± 3.7 % and 95.9 ± 1.2 % for DS1 and DS2, respectively. This is quite encouraging, because these results are already better than all previous work on magnetic resonance-based vertebra segmentation. However, as motivated introductory, ambiguities occur between adjacent vertebra even if the network is trained to segment only the central vertebra of each patch. To verify that, we checked any two neighboring vertebra patches of each image and counted the number of ambiguities, i.e. when segmentations of adjacent vertebrae would overlap each other. Regarding DS1, ambiguities occurred 56.1 ± 0.2 % of the time. The occurance of ambiguous segmentations correlates with the size of the vertebrae, meaning that they occur more frequently in the cervical and section, but less frequently in the lumbar section of the spine. This is backed by the results on DS2, for which ambiguities arose in only 11.7 ± 16.0 % of the cases.

3.2.4. Discussion

Given these results, an explicit handling of ambiguities may not be necessary for settings where only the lower thoracolumbar section of the spine is present and only young healthy adults are imaged, which both is true for DS2. However, for more general settings, which include the cervical or upper thoracic section of the spine or cover subjects of varying health statuses - both is true for DS1 - the explicit handling of ambiguities becomes necessary. Our graph cut formulation addresses this challenge naturally, while guaranteeing a globally optimal result. Compared to using the presegmentation network, we expect a statistically significant ($p < 0.01$) gain in result quality at least for DS1. For DS2 the result quality is already very good and ambiguities are rather rare. Hence, we do not expect large effects on the overall result quality here.

3.3. Graph Cut Framework

3.3.1. Experimental Setup

Next we discuss the overall results of our method and put it into context with previous works. To this end, we set up a quantitative comparison in Table 1. The given results for our method represent average values from a large scale experiment, the details of which are listed in Table 2. Within the experiment we evaluated our graph cut approach with the same five-fold cross-validation that was used for the presegmentation network. To be precise, we pooled the subjects into training and validation sets according to the current fold, then trained a presegmentation network on the training set, applied the network to the validation set but inside of our graph cut framework and reported the result quality and run time for that fold. The complete cross-validation was repeated another five times with uniformly drawn configurations of yet unseen vertebra displacements for each subject of each validation set. The combination of cross-validation and unseen displacements ensures the resilience of our results.

3.3.2. Comparison to Related Work

As can be seen from results in Table 1, the presented method clearly improves upon our earlier work [23]. Moreover, it also outperforms all other previous works on magnetic resonance-based vertebra segmentation. This becomes most clear when comparing our thoracolumbar results to those of previous works. Please note that some works were evaluated on the same data (marked by @ DS2), which implies that results are directly comparable. For all other works, the comparison needs to be taken with a grain of salt. For DS2 we yield Dice coefficients of $96.0 \pm 1.0\,\%$. Only the results of [6] and [10] are come close to these values, showing Dice coefficients of $90.8 \pm 1.8\,\%$ and $93.4 \pm 1.7\,\%$, respectively. Both works apply active shape models, which according to [6] take several minutes to fit a single vertebra (Intel Core 2 Duo @ 2×2.0 GHz), casting doubts about the applicability in practice. Performance was not reported in [10], but we expect even longer run times due to their convolutional neural network. Their network predicts a vertebra likelihood map voxel-wise in sliding window manner, which will take several minutes per vertebra, cf. the run time benchmark given in [23].

Our method took only $6.3 \pm 0.21\,\mathrm{s}$ per image on DS2 (Intel Core i5 @ 4×3.30 GHz). This is a considerable speed-up compared to previous works, where the segmentation of a section of the spine may take minutes. For comparison, [15] reported 1.3 min (unknown multi-core system @ 3.0 GHz) for thoracolumbar images with seven vertebrae and

Table 1: Comparison to previous works on magnetic resonance-based vertebra segmentation. Work is categorized into 2D (mid-sagittal) and 3D (volumetric) analysis. For each category, work is sorted chronologically. Please note that we included the 2D analysis techniques for completeness, their results are not comparable to the 3D setting. Abbreviations: DS1 - data set 1; DS2 - data set 2; DC - Dice coefficient; AD - average inter-surface distance; HD - Hausdorff distance; C - cervical; L - lumbar; TL - thoracolumbar; W - whole spine; T_1w - T_1-weighted; T_2w - T_2-weighted; ^{P} provided by author; ^{R} recalculated from results.

2/3D	Works	Section	Weighting	#Images	#Vertebrae	DC [%]	AD [mm]	HD [mm]
2D	Huang [42]	C, L, W	T_2w	?	52	96±?	?	?
	Ayed [20]	L	T_2w	15	75	85 ± 5.1	?	?
	Zheng [21]	L	T_1w, T_2w	5	?	96.6 ± 0.3	?	1.7 ± 0.2
	Ghosh [19]	L	T_2w	13	?	84.4 ± 3.8	?	?
	Athertya [22]	TL	T_1w	16	?	86.7 ± 4.1	?	5.40 ± 1.12
3D	Stern [3]	TL	T_2w	9	75	?	1.85 ± 0.47	?
	Neubert [6]	TL	T_2w	14	132	90.8 ± 1.8^R	0.67 ± 0.17^R	4.08 ± 0.94^R
	Kadoury [8]	TL	T_1w	8	136	?	2.93 ± 1.83^R	?
	Schwarzenberger [11]	L	T_2w	2	10	81.3 ± 5.1	?	?
	Suzani [9]	L	T_1w	9	45	?	3.02 ± 0.82^R	9.20 ± 2.43^R
	Zukic [4]	TL	T_1w, T_2w	17	153	79.3 ± 5.0^P	1.76 ± 0.38	11.89 ± 2.65^P
	Chu [15] @ DS2	TL	T_2w	23	161	88.7 ± 2.9	1.5 ± 0.2	6.4 ± 1.2
	Hille [13]	TL	T_1w	6	34	84.8±?	1.29 ± 0.42	6.55±?
	Korez [10] @ DS2	TL	T_2w	23	161	93.4 ± 1.7	0.54 ± 0.14	3.83 ± 1.04
	Gaonkar [16]	TL	T_1w, T_2w	23	?	79 ± 5.0	?	?
	Hille [14] @ DS2	TL	T_2w	23	161	88.2 ± 1.9	1.66 ± 0.28	6.01 ± 1.01
3D	Rak [23] @ DS1	W	T_1w, T_2w	128	1412	85.2 ± 4.1	1.39 ± 0.34	5.39 ± 1.56
	Rak [23] @ DS2	TL	T_2w	23	161	90.3 ± 2.0	1.23 ± 0.24	5.20 ± 1.04
	This work @ DS1	W	T_1w, T_2w	128	1412	93.8 ± 2.6	1.06 ± 0.23	4.06 ± 1.14
	This work @ DS2	TL	T_2w	23	161	96.0 ± 1.0	0.79 ± 0.25	3.85 ± 2.20

[9] segments lumbar images in “less than two minutes” (Intel Core i5 @ 4×2.5 GHz). The difference becomes most clear when considering the run time per vertebrae. For instance in [13] the computation “never exceeded 60 s” (unknown system), which got improved in [14] to 21.9 s (unknown system) on average per vertebra. In [3] and [6] even several minutes per vertebra (Intel Core 2 Duo @ 2×2.0 GHz and 2.83 GHz, respectively) were reported. In general, all previous works state run times well above ten seconds, while ours took only 0.9 ± 0.03 s per vertebra (Intel Core i5 @ 4×3.30 GHz) for DS2.

3.3.3. Discussion

The results shown in Table 1 indicate that DS1 is more challenging than DS2. Specifically, the Dice coefficients decrease to $93.8 \pm 2.6\,\%$, which is still better than what was reported in any previous work. Although the improvement is smaller, we think that these results are encouraging too. The rationale is that DS1 not only contains lumbar vertebrae, but also cervical and upper thoracic vertebrae, which are harder to segment due to their smaller size. Exemplary results for all data sets are depicted in Figure 7. When considering the T_1- and T_2-weighted subsets of DS1, we observe that the latter sequence gives slightly better results. Specifically, the T_1- and T_2-weighted images yield Dice coefficients of $93.6 \pm 3.0\,\%$ and $94.0 \pm 2.3\,\%$, respectively. Presumably, the difference can be explained by the improved contrast to intervertebral discs for the latter sequence. For DS1 the run time per vertebra is 1.35 ± 0.08 s, which means that whole spine images can be segmented in as little as 32.4 ± 1.92 s (Intel Core i5 @ 4×3.30 GHz) on average.

Comparing the results of our graph cut framework to those of the presegmentation network, we observe that Dice coefficients rise by $0.4\,\%$ on average for DS1. According to a one-sided Welch t-test, the improvement is statistically significant with $p = 2.3359 \times 10^{-11}$. Not only the overall quality increases, but also its spread decreases. Specifically, the standard deviation of the Dice coefficients reduce by $1.1\,\%$ on average for DS1. There is also a measurable increase in result quality (by $0.06\,\%$) and a decrease in spread (by $0.17\,\%$) for DS2, but both are one order of magnitude smaller than for DS1 and thus not statistically significant. These results underpin that the handling of ambiguities may not be beneficial for lumbar settings and young healthy adults. However, they also underpin that for more general settings, which include the cervical or thoracic section of the spine or cover subjects of varying health statuses, explicit handling of ambiguities is clearly beneficial.

4. Conclusion

We proposed an automatic approach for fast vertebral body segmentation in three-dimensional magnetic resonance images of the whole spine. To this end, we employ a novel combination of a task-specific convolutional neural network with a graph cut formulation based on encoding swaps, which enable the segmentation of multiple vertebrae in an efficient binary problem formulation without risking ambiguous segmentations of adjacent vertebrae. Our approach grounds on our earlier work [23], where we showed that engineered appearance and shape features can compete with recent machine

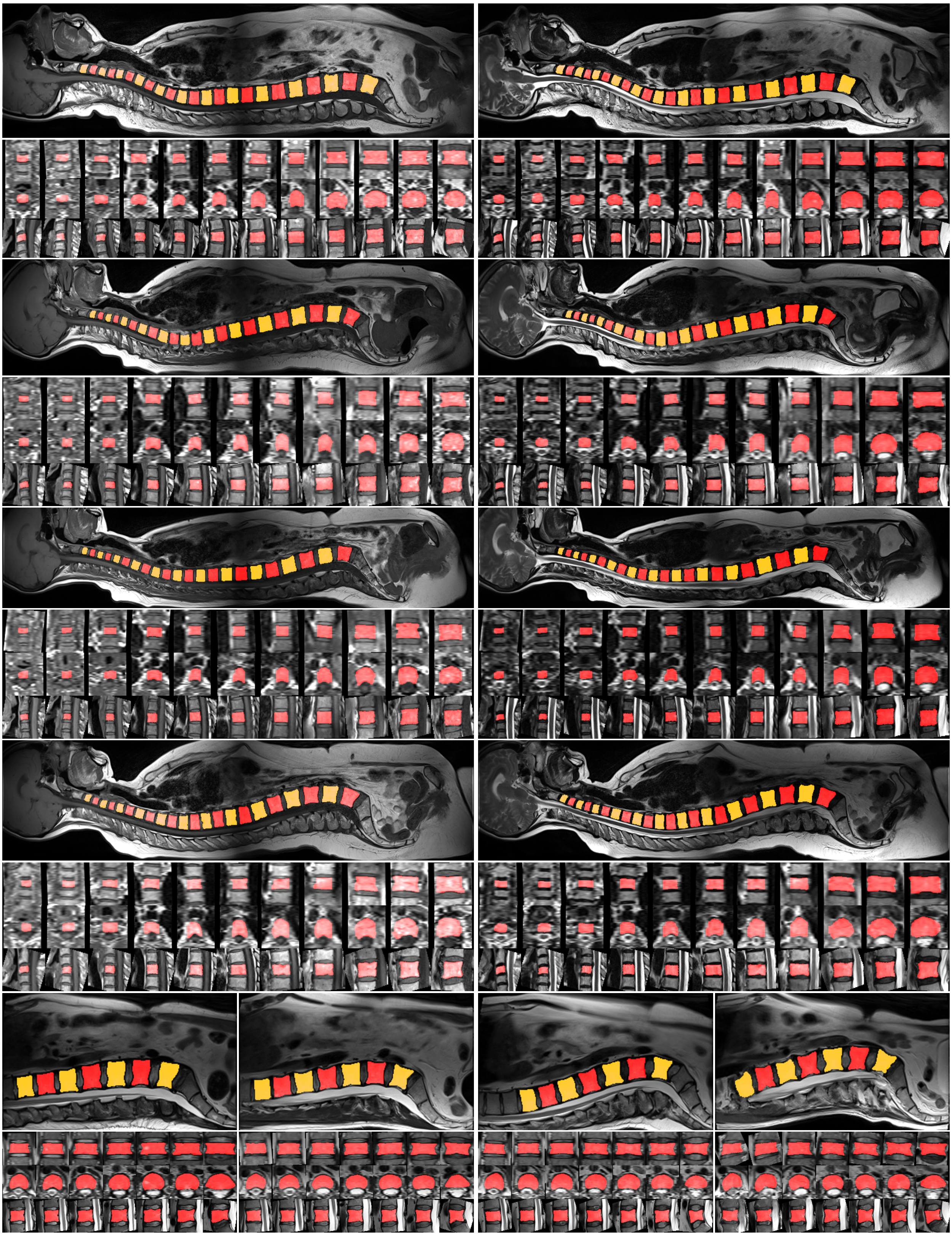

Figure 7: Eight topmost rows: segmentation results on T_1- (left column) and T_2-weighted (right column) images of four subjects from DS1. Two bottommost rows: segmentation results on T_2-weighted images of four different subjects from DS2. Odd rows: mid-sagittal slices after re-embedding of patch-wise segmentation results into the image domain. Vertebra coloring reflects encoding swaps. Even rows: particular coronal, transverse and sagittal views of every other vertebra from head to foot.

learning-based methods if integrated into our graph cut formulation. Within this work, we replace the hand-crafted appearance and shape features by an end-to-end trainable convolutional neural network and demonstrate how this can be integrated effectively into our graph cut formulation to yield a topologically correct segmentations.

On the application side, our work is the first that applies to different imaging sequences as well as to the whole spine, which we demonstrated on two data sets. The first data set contains T_1- and T_2-weighted whole-spine images of 64 subjects. The second data set comprises 23 T_2-weighted thoracolumbar images and is publicly available. Compared to our earlier work [23], the segmentation quality rose by a significant 5.7 % to 8.6 %, yielding Dice coefficients of $93.8 \pm 2.6\,\%$ and $96.0 \pm 1.0\,\%$ for both data sets, respectively. Our results are also superior to those of previous works, while our method takes only a fraction of their run time. In particular, run times were $1.35 \pm 0.08\,$s and $0.90 \pm 0.03\,$s per vertebra for both data sets, respectively. A complete whole spine segmentation took $32.4 \pm 1.92\,$s on average.

Our work has limitations. In case severe pathologies alter the appearance or shape of a vertebra, e.g. metastases or burst fractures, the segmentation quality may be poor. Any available appearance or shape modeling technique will struggle to capture the large diversity accompanied with such cases. To be precise, we are not aware of any work addressing this issue. Neural network-based techniques like ours should be able to cover even severe pathological cases, but only after a large training data set becomes available, which is not the case as of today. In the future, we want to address this challenge. Moreover, we plan to incorporate intervertebral discs into our binary problem formulation, enabling the segmentation both structures at the same time. Besides that, another direction of future work would be to somehow the merge the localization and segmentation step into a unified framework to overcome the need for user interaction in case of erroneous localizations.

Conflict of Interest

The authors confirm that they do not have any financial or personal relationships with any other person or organization that could inappropriately influence (bias) their work.

Acknowledgments

We thank all parties and participants of the Study of Health in Pomerania for providing us with such a comprehensive image data set. This research was partially funded by the project Visual Analytics in Public Health (TO 166/13-2), which is part of the Priority Program 1335: Scalable Visual Analytics of the German Research Foundation. Additional funding was provided by the European Union and the federal state of Saxony-Anhalt, Germany (FuE 74/16). We also acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp graphics card that was used for this research.

References

[1] R. S. Alomari, S. Ghosh, J. Koh, V. Chaudhary, Vertebral column localization, labeling, and segmentation,

Table 2: Dice coefficients of our overall approach on both data sets. The columns enumerate the folds of our cross-validation. The rows enumerate different uniformly sampled configurations of vertebra displacements. The first row of each block represents a configuration without displacement. Abbreviations: DS1 - data set 1; DS2 - data set 2; T_1w - T_1-weighted; T_2w - T_2-weighted

Dice coefficient [%]			Folds					
			1	2	3	4	5	All
DS1 - T_1w	Displacements	0	94.3 ± 1.8	94.9 ± 1.3	95.1 ± 1.4	95.1 ± 1.5	91.7 ± 4.4	94.2 ± 2.1
		1	93.5 ± 2.3	94.0 ± 3.0	94.6 ± 1.7	94.2 ± 2.3	90.8 ± 6.7	93.4 ± 3.2
		2	93.6 ± 2.1	93.7 ± 5.4	94.5 ± 1.8	94.2 ± 2.4	90.2 ± 9.2	93.2 ± 4.2
		3	93.6 ± 1.9	94.0 ± 2.8	94.4 ± 1.9	94.4 ± 1.9	91.5 ± 4.6	93.6 ± 2.6
		4	93.6 ± 2.0	94.2 ± 1.8	94.3 ± 2.4	94.2 ± 2.2	90.2 ± 8.2	93.3 ± 3.3
		5	93.6 ± 2.1	94.1 ± 2.2	94.5 ± 1.8	94.4 ± 2.2	91.5 ± 4.6	93.6 ± 2.6
		All	93.7 ± 2.0	94.2 ± 2.7	94.6 ± 1.8	94.4 ± 2.1	91.0 ± 6.3	93.6 ± 3.0
DS1 - T_2w	Displacements	0	94.6 ± 1.7	94.8 ± 1.4	95.4 ± 1.1	95.0 ± 1.2	92.8 ± 2.5	94.5 ± 1.6
		1	93.5 ± 4.0	94.1 ± 2.8	94.6 ± 3.3	94.5 ± 1.5	92.4 ± 2.8	93.8 ± 2.9
		2	93.9 ± 2.3	93.9 ± 2.4	94.7 ± 2.5	94.5 ± 1.4	92.5 ± 2.7	93.9 ± 2.3
		3	93.6 ± 3.8	94.2 ± 1.8	94.7 ± 1.7	94.5 ± 1.5	92.0 ± 4.5	93.8 ± 2.7
		4	94.0 ± 2.2	94.4 ± 1.7	94.7 ± 1.6	94.5 ± 1.5	92.3 ± 3.1	94.0 ± 2.0
		5	93.8 ± 2.8	94.1 ± 1.9	94.8 ± 1.5	94.5 ± 1.7	92.4 ± 2.8	93.9 ± 2.1
		All	93.9 ± 2.8	94.2 ± 2.0	94.8 ± 1.9	94.6 ± 1.5	92.4 ± 3.1	94.0 ± 2.3
DS2 - T_2w	Displacements	0	96.9 ± 0.6	97.2 ± 0.4	96.7 ± 0.5	96.6 ± 0.4	93.8 ± 2.7	96.2 ± 0.9
		1	96.6 ± 0.5	96.6 ± 1.3	96.4 ± 0.5	96.2 ± 0.5	94.4 ± 1.2	96.0 ± 0.8
		2	96.6 ± 0.5	96.2 ± 2.4	96.3 ± 0.7	96.2 ± 0.5	94.4 ± 1.3	95.9 ± 1.1
		3	96.6 ± 0.5	96.4 ± 1.7	96.3 ± 0.6	96.3 ± 0.5	94.3 ± 1.4	96.0 ± 0.9
		4	96.3 ± 1.7	96.4 ± 2.1	96.3 ± 0.6	96.1 ± 0.7	94.4 ± 1.3	95.9 ± 1.3
		5	96.4 ± 1.7	96.8 ± 0.4	96.4 ± 0.5	96.3 ± 0.4	94.2 ± 2.0	96.0 ± 1.0
		All	96.6 ± 0.9	96.6 ± 1.4	96.4 ± 0.6	96.3 ± 0.5	94.2 ± 1.7	96.0 ± 1.0

in: Spinal Imaging and Image Analysis, 2015, pp. 193–229.

[2] M. Rak, K. D. Tönnies, On computerized methods for spine analysis in MRI: a systematic review, International Journal of Computer Assisted Radiology and Surgery (2016) 1–21.

[3] D. Štern, B. Likar, F. Pernuš, T. Vrtovec, Parametric modelling and segmentation of vertebral bodies in 3D CT and MR spine images, Physics in Medicine and Biology 56 (2011) 7505–7522.

[4] D. Zukić, A. Vlasák, J. Egger, D. Hořínek, C. Nimsky, A. Kolb, Robust detection and segmentation for diagnosis of vertebral diseases using routine MR images, Computer Graphics Forum 33 (2014) 190–204.

[5] M. Rak, K. Engel, K. D. Tönnies, Closed-form hierarchical finite element models for part-based object detection, in: Proceedings of the International Symposium on Vision, Modeling and Visualization, 2013, pp. 137–144.

[6] A. Neubert, J. Fripp, C. Engstrom, R. Schwarz, L. Lauer, O. Salvado, S. Crozier, Automated detection, 3D segmentation and analysis of high resolution spine MR images using statistical shape models, Physics in Medicine and Biology 57 (2012) 8357–8376.

[7] D. Gaweł, P. Główka, T. Kotwicki, M. Nowak, Automatic spine tissue segmentation from MRI data based on cascade of boosted classifiers and active appearance model, BioMed Research International 2018 (2018) 7952946.

[8] S. Kadoury, H. Labelle, N. Paragios, Spine segmentation in medical images using manifold embeddings and higher-order MRFs, IEEE Transactions on Medical Imaging 32 (2013) 1227–1238.

[9] A. Suzani, A. Rasoulian, S. Fels, R. N. Rohling, P. Abolmaesumi, Semi-automatic segmentation of vertebral bodies in volumetric MR images using a statistical shape+pose model, in: Proceedings of SPIE Medical Imaging, 2014, p. 90360P.

[10] R. Korez, B. Likar, F. Pernuš, T. Vrtovec, Model-based segmentation of vertebral bodies from MR images with 3D CNNs, in: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, 2016, pp. 433–441.

[11] R. Schwarzenberg, B. Freisleben, C. Nimsky, J. Egger, Cube-cut: vertebral body segmentation in MRI-data through cubic-shaped divergences, PloS one 9 (2014) e93389.

[12] S. Kim, W. C. Bae, K. Masuda, C. B. Chung, D. Hwang, Semi-automatic segmentation of vertebral bodies in MR images of human lumbar spines, Applied Sciences 8 (2018) 1586.

[13] G. Hille, S. Glaßer, K. Tönnies, Hybrid level-sets for vertebral body segmentation in clinical spine MRI, Procedia Computer Science 90 (2016) 22–27.

[14] G. Hille, S. Saalfeld, S. Serowy, K. Tönnies, Vertebral body segmentation in wide range clinical routine spine MRI data, Computer Methods and Programs in Biomedicine 155 (2017) 93–99.

[15] C. Chu, D. L. Belavỳ, G. Armbrecht, M. Bansmann, D. Felsenberg, G. Zheng, Fully automatic localization and segmentation of 3D vertebral bodies from CT/MR images via a learning-based method, PloS one 10 (2015) e0143327.

[16] B. Gaonkar, Y. Xia, D. S. Villaroman, A. Ko, M. Attiah, J. S. Beckett, L. Macyszyn, Multi-parameter ensemble learning for automated vertebral body segmentation in heterogeneously acquired clinical MR images, IEEE Journal of Translational Engineering in Health and Medicine 5 (2017) 1–12.

[17] F. Fallah, S. S. Walter, F. Bamberg, B. Yang, Simultaneous volumetric segmentation of vertebral bodies and intervertebral discs on fat-water MR images, IEEE Journal of Biomedical and Health Informatics (2018) epub.

[18] J. T. Lu, S. Pedemonte, B. Bizzo, S. Doyle, K. P. Andriole, M. H. Michalski, R. G. Gonzalez, S. R. Pomerantz, Deep Spine: automated lumbar vertebral segmentation, disc-level designation, and spinal stenosis grading using deep learning, in: Proceedings of the Machine Learning for Healthcare Conference, 2018, pp. 403–419.

[19] S. Ghosh, M. R. Malgireddy, V. Chaudhary, G. Dhillon, A supervised approach towards segmentation of clinical MRI for automatic lumbar diagnosis, in: Proceedings of the Workshop on Computational Methods and Clinical Applications for Spine Imaging, 2014, pp. 185–195.

[20] I. B. Ayed, K. Punithakumar, R. Minhas, R. Joshi,

G. J. Garvin, Vertebral body segmentation in MRI via convex relaxation and distribution matching, in: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, 2012, pp. 520–527.

[21] Q. Zheng, Z. Lu, Q. Feng, J. Ma, W. Yang, C. Chen, W. Chen, Adaptive segmentation of vertebral bodies from sagittal MR images based on local spatial information and Gaussian weighted chi-square distance, Journal of Digital Imaging 26 (2013) 578–593.

[22] J. Athertya, G. S. Kumar, Fuzzy clustering based segmentation of vertebrae in T1-weighted spinal MR images, International Journal of Fuzzy Logic Systems 6 (2016) 23–34.

[23] M. Rak, K. D. Tönnies, Star convex cuts with encoding swaps for fast whole-spine vertebra segmentation in MRI, in: Proceedings of the International Symposium on Vision, Modeling and Visualization, 2017, pp. 145–152.

[24] M. Rak, K. D. Tönnies, A learning-free approach to whole spine vertebra localization in MRI, in: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, 2016, pp. 283–290.

[25] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 1124–1137.

[26] V. Kolmogorov, R. Zabin, What energy functions can be minimized via graph cuts?, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 147–159.

[27] J. A. Sethian, Fast marching methods, SIAM Review 41 (1999) 199–235.

[28] O. Veksler, Star shape prior for graph-cut image segmentation, in: Proceedings of the European Conference on Computer Vision, 2008, pp. 454–467.

[29] J. E. Bresenham, Algorithm for computer control of a digital plotter, IBM Systems Journal 4 (1965) 25–30.

[30] S. Gould, DARWIN: A framework for machine learning and computer vision research and development, Journal of Machine Learning Research 13 (2012) 3533–3537.

[31] O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation, in: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241.

[32] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, O. Ronneberger, 3D U-Net: learning dense volumetric segmentation from sparse annotation, in: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, 2016, pp. 424–432.

[33] W. Yang, Q. Zhou, Y. Fan, G. Gao, S. Wu, W. Ou, H. Lu, J. Cheng, L. J. Latecki, Deep context convolutional neural networks for semantic segmentation, in: Proceedings of CCCV Communications in Computer and Information Science, 2017, pp. 696–704.

[34] Q. Liu, X. Lu, Z. He, C. Zhang, W. S. Chen, Deep convolutional neural networks for thermal infrared object tracking, Knowledge-Based Systems 134 (2017) 189–198.

[35] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. van der Laak, B. van Ginneken, C. I. Sanchez, A survey on deep learning in medical image analysis, Medical Image Analysis 42 (2017) 60–88.

[36] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 448–456.

[37] H. Völzke, D. Alte, C. O. Schmidt, D. Radke, R. Lorbeer, N. Friedrich, N. Aumann, K. Lau, M. Piontek, G. Born, et al., Cohort profile: The Study of Health in Pomerania, International Journal of Epidemiology 40 (2011) 294–307.

[38] F. Chollet, et al., Keras, `https://keras.io/` (2015).

[39] M. Abadi, et al., TensorFlow: large-scale machine learning on heterogeneous systems, `https://tensorflow.org/` (2015).

[40] D. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[41] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv:1207.0580.

[42] S. H. Huang, Y. H. Chu, S. H. Lai, C. L. Novak, Learning-based vertebra detection and iterative normalized-cut segmentation for spinal MRI, IEEE Transactions on Medical Imaging 28 (2009) 1595–1605.